САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Кафедра вычислительной физики

В.А. Градусов, А.В. Цыганов

ПРАКТИКУМ ПО КОМПЬЮТЕРНЫМ СРЕДСТВАМ И СИСТЕМАМ ДЛЯ СТУДЕНТОВ ВТОРОГО КУРСА

Учебно-методическое пособие

Санкт-Петербург
2015 г.

Рецензенты:
доцент кафедры вычислительной физики СПбГУ,
кандидат физ.-мат. наук **Е.А. Яревский**,
профессор кафедры высшей математики и математической физики СПбГУ,
доктор физ.-мат. наук **М.А. Лялинов**.

Печатается по решению Ученого совета физического факультета СПбГУ.

В.А. Градусов, А.В. Цыганов
ПРАКТИКУМ ПО КОМПЬЮТЕРНЫМ СРЕДСТВАМ И СИСТЕМАМ ДЛЯ СТУДЕНТОВ ВТОРОГО КУРСА.—
СПб., 2015. — 85 с.

Учебно-методическое пособие содержит начальные сведения о системе компьютерной верстки LaTeX и системе компьютерной алгебры Mathematica. Этот материал предназначен для изучения студентами второго курса бакалавриата «Физика» и «Прикладные математика и физика» на практических занятиях по курсу «Компьютерные средства и системы».

Оглавление

Работа № 1

Основы LaTeX

Напомним, что LaTeX- это компьютерная система для создания профессиональных научных документов высокого качества. Основные сведения об этой системе можно почерпнуть из ридеров лекций, учебных пособий и дополнительных материалов по дисциплине «Компьютерные средства и системы», развернутой в системе Blackboard СПбГУ по адресу https://bb.spbu.ru/

Данное пособие предназначено для проведения практических занятий по данной дисциплине. Итак, начнём с создания нового документа `.tex`, в который записывается исходный код LaTeX. Если вы используете редактор WinEdt или TeXworks, то для этого нужно в меню File выбрать пункт New. В остальных редакторах действия практически аналогичны.

Исходный код содержит команды и обычный текст. Названия команд начинаются с обратной косой черты "\\". Команды, как правило, имеют аргументы. Обязательные аргументы заключаются в фигурные скобки, необязательные — в квадратные. Строка комментария начинается с символа процента "%".

Наберем следующий исходный код:

```
\documentclass[12pt,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[english,russian]{babel}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}
Привет, мир!
\end{document}
```

Здесь мы использовали строку с "%", чтобы отделить преамбулу от начала документа.

В команде `\documentclass` в качестве обязательного аргумента указывается класс или тип документа, который мы создаём. В данном случае это article или статья. В качестве необязательных аргументов рекомендуется указывать основной размер шрифта (для класса article может равняться 10, 11 или 12pt) и формат листа бумаги A3, A4 и т.д.

Команда `\usepackage` загружает дополнительные пакеты. В нашем примере это пакеты для поддержки русского языка. Аргумент utf8 указывает кодировку документа `.tex`. Вам, возможно, понадобится указать другую кодировку, например cp1251 или koi8-r — это зависит от редактора, в котором вы создали документ.

Пакет поддержки языков babel гарантирует корректный перенос слов, добавляет специальные символы и т.п.

Часть документа, которая расположена до окружения `\begin{document}...\end{document}`, называется преамбулой документа, а часть, расположенная внутри этого окружения — телом документа.

Теперь нужно компилировать наш код в документ `.dvi` или `.pdf`. В редакторе WinEdt для создания `.dvi` в меню Accessories нужно выбрать пункт TeXify (или нажать комбинацию клавиш Shift+Ctrl+x). Команда TeXify является макросом, который нужное число раз запускает программы Latex, BibTeX и MakeIndex, которые собственно и занимаются компиляцией исходных кодов LaTeX в документ `.dvi`. Если компиляция прошла успешно, выбираем в меню Accessories пункт DVI Preview (или нажимаем комбинацию клавиш Shift+Ctrl+v) для просмотра созданного документа.

В TeXworks нужно нажать на стрелочку в зелёном кружке, предварительно выбрав список команд, который будет выполнен при этом действии. Окно с файлом `.dvi` появится автоматически.

Изначально LaTeXсоздавался именно для подготовки математических статей, поэтому его возможности в данной области практически безграничны, охватывая все аспекты математической науки. Так как в каждой области знаний и каждом научном издательстве свои традиции в оформлении статей, то необходимо ознакомиться с примерами статей и описанием стилей для авторов, которые можно найти на сайтах издательств. Там же можно найти шаблон статьи — `.tex` документ, который, как правило, сопровождается файлом класса документа `.cls` или стилевым файлом `.sty`. Они определяют внешний вид статьи в соответствии с требованиями издательства или конкретного журнала. Все эти файлы следует скачать на свой компьютер и использовать как заготовку статьи, которую следует наполнить содержанием. Вот как может выглядеть шаблон статьи (создайте новый документ `.tex`, наберите приведенный ниже код и скомпилируйте его):

```
\documentclass[12pt,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[english,russian]{babel}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}

\author{И.~Иванов \and П.~Петров\thanks{СПбГУ}}
\title{Как в Латех \\ набирать статьи}
\date{}

\maketitle

%\begin{titlepage}
%Оформляем как хотим
%\end{titlepage}
```

```
\begin{abstract}
текст текст текст текст
\end{abstract}

\tableofcontents{}

\section{Первый раздел}

текст текст текст

\subsection{его подраздел}

текст текст текст

\subsubsection{подподраздел}

текст текст текст

\paragraph{Важный параграф}

текст текст текст

\section{Второй раздел}

текст текст текст

\section*{Заключение}
%\addcontentsline{toc}{section}{Заключение}

текст текст текст

\appendix
%начинается приложение к документу

\section{Название приложения}

текст текст текст

\end{document}
```

Всюду вместо слов “текст текст текст текст текст” вставьте свои собственные слова. Их можно скопировать, например, из Интернета или pdf файлов научных статей из архива xxx.lanl.gov. С помощью команд `\author` и т.п. в начале тела документа вво-

дится информация для создания заголовка статьи, сам же заголовок вставляется командой `\maketitle`. Имейте ввиду, что если не подать команду `\date`, LaTeX напечатает дату на английском языке. Альтернативно, титульный лист с заголовком можно создавать вручную, что делается с помощью окружения `titlepage`. После заголовка статьи, как правило, идет аннотация — окружение `abstract`, а дальше оглавление, которое создается автоматически командой `\tableofcontents`. Дальше идет основной текст статьи, который принято делить на разделы, подразделы и т.д. (команды `\section`, `\subsection` и т.п.). Нам не нужно заботиться о нумерации разделов, LaTeX делает это автоматически. Звёздочка после команды создания раздела заставляет LaTeX не нумеровать этот раздел. Обратите внимание на то, что раздел Заключение не был включен в оглавление — побочный эффект стоящей после команды `\section` звёздочки. Чтобы исправить это, раскомментируйте строчку с командой `\addcontentsline` после команды `\section*`.

Чтобы можно было создавать в тексте ссылки на раздел документа, нужно добавить метку с помощью команды `\label`, например

```
\section{Второй раздел}
\label{teoremy}
```

Теперь можно создавать ссылки на этот раздел:

```
В разделе~\ref{teoremy} на странице~\pageref{teoremy} мы доказали теорему.
```

Для создания сноски в заголовке используется команда `\thanks`, но в основном тексте вместо нее нужно использовать команду

```
\footnote{текст сноски}
```

Набирайте в одном из разделов документа следующий код и компилируйте его:

```
Когда нужно использовать спецсимволы:
Дефис используем в как-то, короткое тире в диапазоне 10--15,
а обычное тире это --- "длинное"тире.
 Можно использовать кавычки ``лапки'' и <<ёлочки>>.
 Многоточие \ldots
А вот как набрать Schr\"odinger equation.
% Это называется диакритический знак
\begin{center}
Поместим посередине
\end{center}
% попробуйте также окружения flushleft и flushright
Дальше печатаем \textit{курсивом}, или
 \textbf{жирным шрифтом},
или шрифтом \textsf{без засечек} (sans serif),
а текст программ шрифтом\texttt{пишущей машинки}.
% обычный прямой шрифт вставляется командой \textrm
```

```
Размер шрифта {\small маленький}, {\large большой},
{\Large больше} и {\LARGE ещё больше}.
% нормальный \normalsize, ещё \Huge
В И.\,И.~Иванов лучше поставить пробел вручную и неразрывный пробел.
Разрыв \\ строки.

Абзацный отступ.
% для создания абзацного отступа просто оставляем пустую строчку
Отступы по \hspace{1.5cm} горизонтали и

\vspace{0.3cm}

вертикали.
Разрыв
\newpage
страницы.
Создадим нумерованный
\begin{enumerate}
  \item три
  \item два
  \item и вложенный не нумерованный список
    \begin{itemize}
      \item один и три четверти
      \item один с половиной
    \end{itemize}
\end{enumerate}
Буквальное воспроизведение
\begin{verbatim}
some code
\end{verbatim}
%используется для печати кода программ
%текст печатается шрифтом пишущей машинки, команды Tex внутри окружения
%не исполняются и печатаются как есть
```

Попробуем менять преамбулу нашего документа. Изменим класс документа на scrartcl. Посмотрите, как это повлияет на внешний вид документа. Затем вернёмся к article, но добавим опцию twocolumn

```
\documentclass[12pt,a4paper,twocolumn]{article}
```

Посмотрите, к чему это приведет. Заменим опцию twocolumn на draft. Эта опция используется при отладке внешнего вида документа. Она заставляет LaTeX отмечать черным прямоугольником слишком разреженные строки и строки с переполнением. Чтобы убедиться в этом, можете набрать строку

```
Введем длинный путь
```

```
/usr/local/Wolfram/Mathematica/9.0/Documentation
```

Вернёмся теперь к исходной команде

```
\documentclass[12pt,a4paper]{article}
```

и установим поля документа вручную:

```
\textheight 24cm \textwidth 15cm \topmargin -.5in
\oddsidemargin 0in
```

Эти команды задают высоту, ширину и расположение на странице прямоугольника с текстом.

1.1 Задание

Создайте файл .tex и вставьте в него две части текста - на русском и английском. Тексты можно взять из Интернета.
Откомпилируйте файл, проверьте правильность расстановки переносов. Сделайте так, чтобы различные предложения были набраны различными стилями. Измените преамбулу так, чтобы текст был набран в две колонки.
Творческий подход к заданию будет оценён дополнительно.

Работа № 2

Набор формул в LaTeX

Формулы в LaTeX бывают строчные и выключенные. Чтобы понять, что это такое, наберите в теле документа следующий код и компилируйте документ:

```
Напишем строчную формулу  $r=\sqrt{x^2+y^2+z^2}$.
А теперь выключенную
\[
\Gamma(z)=\int\limits_0^{+\infty} \mbox{d}t\,  e^{-t}  t^{z-1}.
\]
```

Символы “_” и “^” создают нижние и верхние индексы. В выключенной формуле мы использовали команду `\limits`, которая заставляет LaTeX рисовать пределы интегрирования внизу и вверху знака интеграла, как принято в российской традиции, а не сбоку от него. Команда `\mbox` создаёт бокс с текстом. В данном случае она используется для того, чтобы нарисовать знак дифференциала (прямым) шрифтом основного текста. Дело в том, что в формуле по умолчанию латинскую букву LaTeX считает переменной и рисует курсивом. Обратите также внимание на созданный вручную (\,) пробел перед подынтегральным выражением (обычные пробелы между символами в формулах игнорируются). Именно так принято набирать интегралы в LaTeX.

Очень часто бывает необходимо сослаться на какую-то из имеющихся в тексте формул. Поэтому части формул в тексте присваиваются номера. Нумерованные формулы создаются с помощью окружения equation:

```
Вот пример нумерованной формулы:
\begin{equation}
\lim_{n\to\infty} (1+\frac{1}{n})^n = e.
\end{equation}
```

LaTeX нарисовал слишком маленькие круглые скобки, и формула выглядит некрасиво. Чтобы размер скобок автоматически подгонялся под размер того, что находится между ними, нужно перед открывающей и закрывающей скобками поставить команды `\left` и `\right`:

```
\lim_{n\to\infty} \left(1+\frac{1}{n}\right)^n = e.
```

Чтобы можно было в текст вставить ссылку на только что набранную нами формулу, нужно ещё добавить метку с помощью команды `\label`:

```
\begin{equation}
\label{e-def}
```

Ссылку создает команда `\ref`, при этом принято ставить скобочки вокруг номера, что приходится делать вручную:

```
Определение основания натурального логарифма~(\ref{e-def}).
```

Набирайте дальше:

```
В формулах можно использовать различные замысловатые шрифты
\begin{equation}
\mbox{Обычный} \  \mathrm{text} \ \mathit{text} \quad
\mathcal{T} \quad \mathbb{R} \quad \mathfrak{G}
\end{equation}
```

Имейте ввиду, что из всех использованных здесь команд кириллицу можно использовать только в команде `\mbox`. Команда `\mathcal` отрисует букву, которую у нас скорее всего назовут “T красивое”. Чтобы использовать команды `\mathbb` и `\mathfrak` — ажурное и готическое написания букв — нужно загрузить дополнительный пакет amssymb, для чего в преамбулу документа добавляем

```
\usepackage{amssymb}
```

Пакеты amssymb и amsmath, которым мы воспользуемся ниже, созданные Американским математическим сообществом (AMS), рекомендуется загружать в любом документе, содержащем формулы. Пакет amssymb также добавляет новые символы

```
\begin{equation}
\blacktriangle \varpropto \geqslant \mbox{сравните с} \gg
\end{equation}
```

и многие другие. В редакторе WinEdt список доступных символов можно посмотреть в меню, которое вызывается нажатием на кнопку с символом суммы на панели инструментов и позволяет вставлять команды и символы нажатием на соответствующие кнопки.

Окружение array используется для набора массивов, в частности матриц

```
Наберем матрицу
\begin{equation}
A= \left(
\begin{array}{ccc}
1000 & -1 & 0 \\
1 & -2000 & -1 \\
0 & 1 & 3000
\end{array}
\right)
\end{equation}
```

Разделителем элементов в строке служит символ &, строки разделяются двумя обратными косыми чертами. Аргумент ссс задает количество столбцов матрицы и одновременно определяет, что в каждом из столбцов элементы будут выровнены по центру столбца (l — по левому краю, r — по правому). Массивы также используются для набора таких вот определений:

```
Более сложный массив
\begin{equation}
\theta(x)=\left\{
\begin{array}{l}
0\quad\mbox{при}\ x<0, \\
1\quad\mbox{при}\ x\ge0
\end{array}
\right.
\end{equation}
```

Здесь размер открывающей скобки подгоняется с помощью команды `\left`, но парной закрывающей скобки в этой формуле нет. В таких случаях после команды `\right` ставится точка.

Как вы уже поняли, много дополнительных возможностей добавляют в LaTeX всевозможные специальные пакеты. Рассмотрим пакет для создания диаграмм и графов XY-pic, который мы, как обычно, загрузим командой:

```
\usepackage[all]{xy}
```

```
Пример графа
\[
\xymatrix{
 & & 1 \ar@/^/[ddl]^f \ar@/_/[dl] \\
2 \ar@{.}[r] \ar[dr]|g & 3 & \\
4 \ar@{--}[u] & 5 &
}
\]
```

Диаграммы и графы строятся в виде прямоугольной матрицы, каждый элемент которой является вершиной графа. Стрелки, идущие от данной вершины, описываются с помощью команды `\ar`, которая позволяет задать стиль и направление стрелки, а также снабдить стрелку текстом или другими объектами. Например, команда `\ar@/^/[ddl]^f` означает: нарисовать выпуклую вверх относительно направления стрелки дуговую стрелку (`@/^/`) по направлению вниз-вниз-влево (`[ddl]`) и разместить сверху относительно направления стрелки метку f (`^f`).

Формулы иногда бывают очень длинными и не умещаются на одной строчке, так что приходится делать их многострочными. Классическим способом набора многострочной формулы является использование окружения eqnarray:

```
Многострочная формула
```

```
\begin{eqnarray}
y(x) & = & \mbox{очень длинное выражение} \nonumber \\
 & + & \mbox{его продолжение} \nonumber \\
 & + & \mbox{его окончание}
\end{eqnarray}
```

Окружение eqnarray очень похоже на окружение array. Поскольку оно используется для набора формул, в строках должно быть по три элемента. Команда `\nonumber` подавляет нумерацию каждой строки в отдельности.

Окружение eqnarray — далеко не единственная возможность набирать многострочные формулы в LaTeX. Например, можно воспользоваться окружением multline, которое входит в уже упомянутый нами выше пакет amsmath. После загрузки этого пакета

```
\usepackage{amsmath}
```

наберите и скомпилируйте код

```
\begin{multline}
\text{Ну} \\
\text{очень}\\
\text{длинная}\\
\text{формула}
\end{multline}
```

По умолчанию первая строчка выравнивается по левому краю листа, последняя — по правому, а остальные — по середине. Считается, что именно таким образом отрисованная многострочная формула выглядит красиво. Обратите внимание, что для отрисовки текста внутри формулы мы использовали здесь команду `\text`, которая также входит в пакет amsmath. В аргументе этой команды можно использовать кириллические символы. Использование `\text` предпочтительнее, чем использование команды `\mbox`.

В LaTeX можно создавать и собственные команды. Мы не будем подробно описывать технологию создания команд, приведем здесь только простой пример использования команды `\newcommand{\name}[n][default]{definition}`: создание короткого варианта названия стандартной команды (вспомните, что именно лень является двигателем прогресса). Для этого в преамбулу документа напишем:

```
\newcommand{\bq}{\begin{equation}}
\newcommand{\eq}{\end{equation}}
```

После этого нумерованную формулу можно создать вот так:

```
\bq
2+2=4
\eq
```

2.1 Задание

Набрать десять формул из списка в формате "статья" , - номера формул по указанию преподавателя, набирать без номера спереди.

Название статьи "Первое задание" , автор статьи - Ваши ФИО, благодарность СПбГУ. Этот файл будет затем использован в последующих заданиях по курсу.

1.
$$S = \left(\bigcup_{j=1}^{p} S_j\right) \bigcup \left(\bigcup_{i=1}^{i=p} \bigcup_{j=1}^{j=p} \Gamma_i \bigcap \Gamma_j\right)$$

2.
$$\begin{aligned} \overrightarrow{r'_u}(u,v) &= \varphi'_u(u,v)\vec{i} + \psi'_u(u,v)\vec{j} + \chi'_u(u,v)\vec{k}, \\ \overrightarrow{r'_v}(u,v) &= \varphi'_v(u,v)\vec{i} + \psi'_v(u,v)\vec{j} + \chi'_v(u,v)\vec{k}. \end{aligned}$$

3.
$$(a+b)^n = \sum_{k=0}^{n} C_n^k \, a^{n-k} \, b^k$$

4.
$$\begin{cases} (x^2+y^2)^2 = a^2(x^2-y^2) \\ x^2+y^2+z^2 \le a^2 \\ x \ge 0, y \ge 0, z \ge 0 \end{cases}$$

5.
$$\lim_{x\to a} \lim_{n\to\infty} f_n(x) = \lim_{n\to\infty} \lim_{x\to a} f_n(x)$$

6.
$$\underline{\lim_{n\to\infty}} x_n = \inf P(\{x_n\}), \ \overline{\lim_{n\to\infty}} x_n = \sup P(\{x_n\}), \ \underline{\lim_{n\to\infty}} x_n \le \overline{\lim_{n\to\infty}} x_n$$

7.
$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle f}\downarrow & \swarrow_{jB} & \downarrow{\scriptstyle g} \\ B & \xrightarrow[g]{} & C \end{array}$$

8.
$$a_0 = -\frac{2}{\pi}\int_0^{\pi} \ln\sin\frac{x}{2}\,dx = -\frac{2}{\pi}\int_0^{\pi/2} \ln\sin\frac{x}{2}\,dx - \frac{2}{\pi}\int_{\pi/2}^{\pi} \ln\sin\frac{x}{2}\,dx =$$
$$= \ln 2 - \frac{2}{\pi}\int_0^{\pi/2} \ln\sin x\,dx = \ln 2 - \frac{1}{\pi}\int_0^{\pi} \ln\sin\frac{t}{2}\,dt = \ln 2 + \frac{a_0}{2}$$

9.
$$(f\,\varphi)^{(n)} = \sum_{k=0}^{n} \frac{n!}{k!(n-k)!} f^{(n-k)}\varphi^{(k)}, \text{ где } f^{(0)} = f$$

10.
$$\int \frac{dx}{\sqrt{x^2+a^2}} = \ln(x + \sqrt{x^2+a^2}) + C \ (a \neq 0)$$

11.
$$\int \frac{Ax+B}{x^2+px+q}\,dx = = \frac{A}{2}\ln(x^2+px+q) + \frac{2B-Ap}{2\sqrt{q-\frac{p^2}{4}}} \arctan \frac{x+\frac{p}{2}}{\sqrt{q-\frac{p^2}{4}}} + C$$

12.
$$\iint_S z\,ds = \iint_D \varphi(x,y)\sqrt{1+\varphi'^2_x(x,y)+\varphi'^2_y(x,y)}\,dx\,dy$$

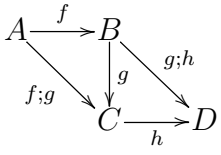
13.
$$\iiint\limits_G f(x,y,z)\,dx\,dy\,dz = \int\limits_a^b dx \iint\limits_{P_x} f(x,y,z)\,dy\,dz$$

14.
$$\iint_{S^+} P\,dy\,dz + Q\,dz\,dx + R\,dx\,dy = \iiint_G \left(\frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial R}{\partial z}\right) dx\,dy\,dz$$

15.
$$\int_a^b f(x)g(x)\,dx = g(a)\int_a^c f(x)\,dx + g(b)\int_c^b f(x)\,dx$$

16.
$$\int_L f\,ds = \int_\alpha^\beta (f\circ\varphi)(u)\sqrt{\sum_{k=1}^{n}(\varphi'_k(u))^2}\,du$$

17.
$$\iint_\Omega \left(\frac{\partial Q}{\partial x} - \frac{\partial}{\partial y}\right) dx\,dy = \int_{\partial\Omega^+} P\,dx + Q\,dy$$

18.


19.
$$\frac{\partial f}{\partial x_j}(a) = \lim_{t\to a_j} \frac{f(a_1,\ldots,a_{j-1},t,a_{j+1},\ldots,a_n) - f(a)}{t-a_j}$$

20.

$$\left\| \frac{\partial f_j}{\partial x_k}(a) \right\|_{j=1,k=1}^{p,n} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(a) & \frac{\partial f_1}{\partial x_2}(a) & \dots & \frac{\partial f_1}{\partial x_n}(a) \\ \frac{\partial f_2}{\partial x_1}(a) & \frac{\partial f_2}{\partial x_2}(a) & \dots & \frac{\partial f_2}{\partial x_n}(a) \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_p}{\partial x_1}(a) & \frac{\partial f_p}{\partial x_2}(a) & \dots & \frac{\partial f_p}{\partial x_n}(a) \end{pmatrix}$$

21.

$$J_{\Phi_s}(r,\varphi,\psi) = \begin{vmatrix} \cos\varphi\cos\psi & -r\sin\varphi\cos\psi & -r\cos\varphi\sin\psi \\ \sin\varphi\cos\psi & r\cos\varphi\cos\psi & -r\sin\varphi\sin\psi \\ \sin\psi & 0 & r\cos\psi \end{vmatrix} = r^2\cos\psi$$

22.

$$\frac{\partial^s f}{\partial x_{j_s}\,\partial x_{j_{s-1}}\dots\partial x_{j_1}}(a) = \frac{\partial}{\partial x_{j_s}}\left(\frac{\partial^{s-1} f}{\partial x_{j_{s-1}}\dots\partial x_{j_1}}\right)(a)$$

23.

$$\int\limits_a^b \sum_{n=1}^{\infty} f_n(x)\,dx = \sum_{n=1}^{\infty} \int\limits_a^b f_n(x)\,dx$$

24.

$$R = \sup\left\{ |x| : \text{ряд} \sum_{n=0}^{\infty} a_n x^n \text{сходится} \right\}$$

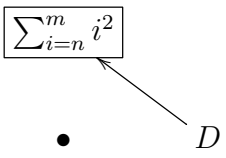
25.

$$\widetilde{I}\,(y) = \int\limits_{\alpha(y)}^{\beta(y)} f(x,y)\,dx,\ \ y \in [c,d]$$

26.

$$\Gamma^{(n)}(\alpha) = \int\limits_0^{+\infty} x^{\alpha-1}e^{-x}\ln^n x\,dx,\ \ \forall\alpha\in(0,+\infty),\ \ \forall n = 0,1,2,\dots$$

27.



28.

$$B(\alpha,\beta) = \int\limits_0^{+\infty} \frac{t^{\alpha-1}}{(1+t)^{\alpha+\beta}}dt,\ \ \alpha>0,\ \ \beta>0$$

29.

$$\int\limits_a^b \varphi_n(x)\cdot\varphi_m(x)\,dx = \begin{cases} 0, & \text{если } n\neq m, \\ \gamma_n > 0, & \text{если } n = m. \end{cases}$$

30.

$$\min_{Q_m} \int_a^b (f(x) - Q_m(x))^2\,dx = \int_a^b (f(x) - S_m^f(x))^2\,dx$$

31.

$$f(x) \sim \frac{\alpha_0}{2} + \sum_{k=1}^{\infty} \left(\alpha_k \cos\frac{k\pi x}{T} + \beta_k \sin\frac{k\pi x}{T} \right)$$

32.

$$d(f \cdot \psi)_a(\Delta x) = df_a(\Delta x) \cdot \psi(a) + f(a) \cdot d\psi_a(\Delta x)$$

33.

$$\int_0^{\pi} \frac{f(x_0+t) - f(x_0+0)}{\pi}\, D_n(t)\,dt = \\ = \int_0^{\pi} \frac{f(x_0+t) - f(x_0+0)}{2\pi\,\sin\frac{t}{2}}\, \sin\left(n+\frac{1}{2}\right) t\,dt$$

34.

$$\int_{\Gamma_S^+} P\,dx + Q\,dy + R\,dz = \iint_{S^+} \begin{vmatrix} dy\,dz & dz\,dx & dx\,dy \\ \partial/\partial x & \partial/\partial y & \partial/\partial z \\ P & Q & R \end{vmatrix}$$

35.

$$\Gamma_f = \{(x, f(x)) \in X \times Y : x \in X\}$$

36.

$$\overrightarrow{n}^{\pm}(u,v) = \pm \frac{\overrightarrow{r'_u} \times \overrightarrow{r'_v}}{|\overrightarrow{r'_u} \times \overrightarrow{r'_v}|} = \frac{A(u,v)\vec{i} + B(u,v)\vec{j} + C(u,v)\vec{k}}{\sqrt{A^2(u,v) + B^2(u,v) + C^2(u,v)}}$$

37.

$$\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \varphi'_u & \psi'_u & \chi'_u \\ \varphi'_v & \psi'_v & \chi'_v \end{vmatrix} = \vec{i} \begin{vmatrix} \psi'_u & \chi'_u \\ \psi'_v & \chi'_v \end{vmatrix} - \vec{j} \begin{vmatrix} \varphi'_u & \chi'_u \\ \varphi'_v & \chi'_v \end{vmatrix} + \vec{k} \begin{vmatrix} \varphi'_u & \psi'_u \\ \varphi'_v & \psi'_v \end{vmatrix} = \\ = A(u,v)\vec{i} + B(u,v)\vec{j} + C(u,v)\vec{k}. \tag{2.1}$$

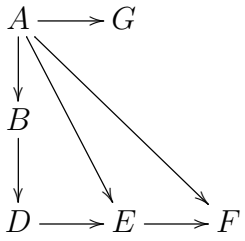
38.

$$\begin{aligned} \iint_{(S,\vec{n}^+)} P\,dy\,dz &= \iint_S P(M)\cos(\vec{n}^+(M), \vec{i})\,ds; \\ \iint_{(S,\vec{n}^+)} Q\,dz\,dx &= \iint_S Q(M)\cos(\vec{n}^+(M), \vec{j})\,ds; \\ \iint_{(S,\vec{n}^+)} R\,dx\,dy &= \iint_S R(M)\cos(\vec{n}^+(M), \vec{k})\,ds. \end{aligned} \tag{2.2}$$

39.

$$\int_{\Gamma^+} P\,dx + Q\,dy + R\,dz = \iint_{(S,\vec{n}^+)} \Big(\frac{\partial R}{\partial y} - \frac{\partial Q}{\partial z}\Big)\,dy\,dz + \Big(\frac{\partial P}{\partial z} - \frac{\partial R}{\partial x}\Big)\,dz\,dx + \Big(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\Big)\,dx\,dy. \quad (2.3)$$

40.



Работа № 3

Библиография, таблицы, рисунки и цвет в LaTeX

Список литературы создаётся с помощью окружения thebibliography

```
\begin{thebibliography}{99}

\bibitem{humberston}
Charlton M, Humberston J W 2001 \textit{Positron Physics}
(Cambridge: Cambridge University Press)

\bibitem{dirac30}
Dirac P A M 1930 \textit{Proc. Roy. Soc. Lond. A}
\textbf{126} 360

\end{thebibliography}
```

Обязательный аргумент окружения thebibliography определяет номер элемента списка литературы, который займет больше всего места при отрисовке. Чаще всего число элементов списка определяется двузначным числом, тогда традиционно в качестве аргумента используют 99. Теперь в основном тексте мы можем сослаться на литературные источники с помощью команды `\cite`

```
Существование позитрона было предсказано
теоретически~\cite{humberston,dirac30}.
```

В каждом уважающем себя журнале есть свой определённый стиль оформления элементов списка литературы. Например, только что созданный нами список литературы соответствует требованиям журнала Journal of Physics A. Из-за этого одни и те же литературные источники в разных публикациях приходится оформлять по-разному. Занятие это очень трудоемкое и малопривлекательное. Поэтому была создана программа BibTeX, которая умеет формировать окружение thebibliography по записям в специальном файле с расширением .bib. Создайте в текстовом редакторе файл References.bib следующего содержания:

```
@BOOK{humberston,
    author = "M. Charlton and J. W. Humberston",
```

```
    title = "Positron Physics",
    publisher = "Cambridge University Press",
    adress = "Cambridge",
    year = "2001"
}

@ARTICLE{dirac30,
    author        = "P. A. M. Dirac",
    year          = "1930",
    journal       = "Proc. Roy. Soc. Lond. A",
    volume        = "126",
    pages         = "360",
}
```

Теперь для создания списка литературы в теле основного документа вместо окружения thebibliography достаточно написать команду

```
\bibliography{References}
```

а в преамбулу добавить команду

```
\bibliographystyle{plain}
```

Список литературы будет создан автоматически и будет содержать те источники, которые мы указали в основном тексте с помощью команды `\cite`. Остальные записи в файле `.bib` в список литературы включаться не будут. Аргумент команды `\bibliographystyle` определяет стиль оформления элементов списка литературы — в соответствии с требованиями того или иного журнала. Попробуйте поменять стиль plain на siam, затем на alpha.

Теперь создадим таблицу. Это делается с помощью двух окружений: table и tabular. Сама таблица создаётся с помощью окружения tabular. Окружение table позволяет делать таблицу плавающим объектом. Это означает, что размещение её в конкретном месте конкретной страницы будет определяться соседними с ней текстом и другими объектами. Повлиять на размещение пользователь может через ключи размещения. Например, ключи t!hp означают: настоятельно прошу разместить таблицу наверху страницы (t — top, ! усиливает пожелание), если не получится, то здесь же в тексте (h — here), если и так не получится, то на отдельной странице с плавающими объектами (p — page). Окружение table также позволяет добавить подпись к таблице с помощью команды `\caption` и создать метку с помощью команды `\label`.

```
\begin{table}[t!hp]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
 & точка перегиба & нет точки перегиба \\
\hline
```

```
$f''(x)$ & $+-$ & $-+$ & $--$ & $++$ \\
\hline
 & вогн. вып. & вып. вогн. & выпукл. & вогн. \\
\hline
\end{tabular}
\caption{Правило нахождения точек перегиба кривой}
\label{pravilo}
\end{table}
```

Окружение tabular похоже на окружение array, так как таблица по сути является массивом ячеек. Ячейки в строке разделяются символом &, строки разделяются двумя обратными косыми чертами. Аргумент |c|c|c|c|c| задает количество столбцов, выравнивание данных в столбцах по центру ячеек, а также разделительные линии между столбцами. Горизонтальные разделительные линии создаются с помощью команды `\hline`.

Пока что на печати мы получили не совсем то, что надо: заголовки в первой строке должны относиться к столбцам 2 и 3, 4 и 5 соответственно. Чтобы этого добиться, нужно объединить соответствующие ячейки в строке заголовков. Это делается с помощью команды `\multicolumn`. Изменим первую строку таблицы:

```
 & \multicolumn{2}{|c|}{точка
    перегиба} & \multicolumn{2}{|c|}{нет
точки перегиба} \\
```

Хотелось бы еще объединить две верхние ячейки в первом столбце. Для этого вместо команды `\hline`, которая рисует горизонтальную линию между первой и второй строками, напишем команду

```
\cline{2-5}
```

Создадим еще одну таблицу, в которой в качестве разделителей столбцов используем точку, чтобы выровнять числа в столбце по десятичному разделителю. Используем инструкцию @{}, которая вставляет между столбцами любой символ, указанный в качестве ее аргумента.

```
\begin{table}[t!hp]
\centering
\begin{tabular}{cr@{.}l}
\hline
$n$ & $n$-й член ряда Тейлора \\
\hline
0 & 1 & 000000 \\
1 & -15 & 00000 \\
2 & 112 & 5000 \\
3 & -562 & 5000 \\
\hline
\end{tabular}
```

```
\caption{Вычисление значений экспоненты с помощью ряда Тейлора}
\label{exp-Taylor}
\end{table}
```

Чтобы выровнять заголовок второго столбца таблицы, вновь воспользуемся командой `\multicolumn`:

```
$n$ & \multicolumn{2}{c}{$n$-й член ряда Тейлора} \\
```

Рисунки в документы LaTeX вставляются с помощью пакета graphicx. При загрузке этого пакета нужно указать драйвер, который будет использоваться для отрисовки документа. Поскольку мы будем использовать стандартный LaTeX-конвейер dvi→PostScript, мы укажем драйвер dvips:

```
\usepackage[dvips]{graphicx}
```

При таком выборе драйвера можно использовать только графические файлы с расширением `.eps`. Вставим в документ три графических файла e_singl.eps, e_tripl.eps, p.eps. Их можно скопировать у преподавателя, или использовать вместо них любые другие файлы `.eps`. Эти файлы удобно хранить в отдельной папке, которую назовем img и поместим в ту же папку, в которой находится и документ `.tex`. Путь к этой папке (относительно папки с документом `.tex`) указывается в преамбуле документа с помощью команды

```
\graphicspath{{img/}}
```

Рисунок вставляется в документ с помощью окружения figure и команды `\includegraphics`. Окружение figure предназначено для работы с рисунками как с плавающими объектами, и во всем похоже на окружение table. Вставим рисунок e_singl.eps:

```
\begin{figure}[t]

\center{\includegraphics[width=1\textwidth]{e_singl}}

\caption{Фазовые сдвиги рассеяния электрона и позитрона на водороде}
\label{phsh}

\end{figure}
```

Размер рисунка в документе задаётся указанием его ширины, при этом сохраняются пропорции исходного рисунка. Мы растянули рисунок по ширине строки текста (переменная `\textwidth` содержит ширину текста).

Рисунки удобно размещать группами, для этого используется окружение minipage, которое просто создаёт мини страницу (бокс) указанной ширины. Вместо строки, содержащей команду `\includegraphics` в предыдущем примере, напишем такой код:

```
\begin{minipage}[h]{0.49\textwidth}
\center{\includegraphics[width=1\textwidth]{e_singl}} \\ (а)
\end{minipage}
\hfill
\begin{minipage}[h]{0.49\textwidth}
\center{\includegraphics[width=1\textwidth]{e_tripl}} \\ (б)
\end{minipage}
\hfill
\begin{minipage}[h]{0.49\textwidth}
\center{\includegraphics[width=1\textwidth]{p}} \\ (в)
\end{minipage}
```

Указывая ширину мини страницы чуть меньше половины ширины текста, мы размещаем две мини страницы в одной “строке”. Обратите внимание, что ширина рисунка по-прежнему равняется ширине строки текста, но теперь это строка текста на мини странице и ее ширина равняется примерно половине ширины строки страницы документа. Команда `\hfill`, которая создаёт бесконечный горизонтальный промежуток, используется здесь для того, чтобы раздвинуть рисунки по краям страницы.

Рассмотрим работу с цветом на примере пакета color, который загрузим командой

```
\usepackage[dvips]{color}
```

Теперь наберем

```
Воспользуемся \textcolor{magenta}{цветовой моделью named},
а теперь \textcolor[RGB]{0,100,255}{моделью RGB},
далее \textcolor[gray]{0.5}{модель gray - оттенки серого},
```

Пакет color позволяет задавать собственные цвета. Например, определим в преамбуле документа цвет seawave

```
\definecolor{seawave}{RGB}{51,204,255}
```

и воспользуемся им

```
а еще \textcolor{seawave}{своим цветом}.
```

Цветной бокс создается командой `\colorbox`

```
\colorbox{yellow}{
    \textcolor{red}{Красным по желтому.}
}
```

Команда `\fcolorbox` дополнительно обводит этот бокс цветной рамкой, причем толщина рамки задается переменной `\fboxrule`

```
\setlength{\fboxrule}{4pt}
\fcolorbox{red}{yellow}{В красной рамочке.}
```

3.1 Задание

Набранные ранее формулы оформите в виде статьи со ссылками. Например

"В книге Ландау, Лифщица [1] в 1963 году в рамках квазиклассического приближения была получена следующая формула

Ваша формула 1

затем в статье Ньютона [2] для данной задачи было получено дополнительное соотношение

Ваша формула 2

используя эти выражения, после ряда кропотливых вычислений, мы можем получить

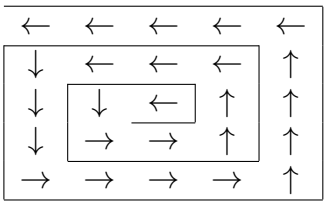
Ваша формула 3

и.т.д."

Всего должно быть не менее 4 реальных ссылок - на монографию, на 2 статьи и на доклад на конференции. Данные найти в Интернете.
Используйте цвет и включите в Вашу статью таблицу и рисунок с подписью. Рисунок и пример таблицы преподаватель может взять из интернета. Например,

2×2	$=$	4
2×4	$=$	8
4×8	$=$	32
8×32	$=$	256
32×256	$=$	$8\ 192$
256×8192	$=$	$2\ 097\ 152$

t	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$	2π
$\cos t$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0	$\frac{\sqrt{2}}{2}$	1
$\sin t$	0	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0



Работа № 4

Создание презентаций в пакете Beamer

В этой работе мы создадим небольшую презентацию с помощью пакета Beamer. Вот как выглядит документ класса beamer, в который пока что не добавлены слайды:

```
\documentclass[serif,professionalfonts]{beamer}

\usepackage[T1]{fontenc}
\usepackage{concmath}

\usepackage[utf8]{inputenc}
\usepackage[english,russian]{babel}

\usetheme{Boadilla}

\begin{document}

\end{document}
```

Пакеты fontenc и concmath задают шрифты основного текста и формул. Команда `\usetheme` задает тему презентации, которая определяет ее внешний вид (цветовая гамма, расположение элементов на слайдах, вид и расположение колонтитулов слайдов и др.).

Создадим титульный слайд. Поместите в папку /img графический файл logo.png с эмблемой Университета, который можно найти в Интернете. Не забудьте в преамбуле указать путь к папке с графическими файлами

```
\graphicspath{{img/}}
```

В тело документа пишем:

```
\title[Презентация в Beamer]{Создание презентации в пакете Beamer}
\author[Иванов]{И.И. Иванов}
\institute[СПбГУ]{Санкт-Петербургский Государственный Университет,
  Физический факультет}
\date[14 марта 2014]{Семинар по КСС \\ 14 марта 2014 }
```

```
\titlegraphic{\includegraphics[width=1.5cm]{logo.png}}

\maketitle
```

В командах в качестве необязательного аргумента мы указываем сокращенные варианты выносимой на титульный слайд информации. Эти сокращенные варианты LaTeX поместит на колонтитулы (если, конечно, они предусмотрены темой документа).

Обычный слайд создается с помощью окружения frame. Строго говоря, это окружение создает некоторый объект, который в терминологии создателей Beamer как раз и называется фрейм. Фрейм может содержать несколько слайдов, что используется для создания анимации. Однако пока наши фреймы будут содержать только один слайд, можно считать, что это одно и то же.

```
\begin{frame}
\frametitle{Введение}

Здесь нужно описать подход, цель работы и какие решались задачи.

\vspace{1cm}

А в конце презентации должен быть слайд с результатами!

\end{frame}
```

На слайд помещено не очень много информации, и он казался бы пустым, если бы мы не создали вертикальный промежуток нужного размера с помощью команды `\vspace`.

Следующий слайд:

```
\begin{frame}
\frametitle{Как выделять переменные, слова и мысли}

\textcolor{cyan}{Умную мысль следует выделить особо}

\begin{block}{Энергия системы}
Энергия есть сумма кинетической энергии и потенциальной энергии
\[
E = \textcolor{red}{T} +\textcolor{blue}{U}
\]
\end{block}

\vspace{0.7cm}

Можно важную формулу нарисовать в рамочке:
\[
```

```
\boxed{T = \frac{mV^2}{2}}
\]
% используется команда \boxed из пакета amsmath

\end{frame}
```

Такое выделение части содержимого презентации — хороший способ привлечения внимания слушателей к содержимому Вашего доклада (презентации).

Другим хорошим способом добиться той же цели является использование небольшого (!!!) количества анимации на слайдах. Добавим в нашу презентацию эффект перехода между слайдами. Для этого сразу же после команды `\frametitle` вставим одну из команд перехода:

```
\begin{frame}
\frametitleВведение{}
\transboxin[duration=0.1]
%\transwipe[direction=90]
%\transdissolve[duration=0.2]
%\transglitter[direction=315]
```

Попробуйте каждую из приведенных здесь команд перехода в действии.

Следующий слайд будет содержать рисунки basis.png, ex1.png, ex2.png (получите у преподавателя или используйте любые другие `.png` файлы), которые поместим в папку img в папке, которая содержит файл `.tex`. В преамбуле укажем, где искать графические файлы:

```
\graphicspath{{img/}}
```

Текст и рисунки на слайдах удобно размещать в нескольких колонках. Для создания колонок на слайде в классе beamer определено специальное окружение columns. Содержимое каждой из колонок заключается в окружение column.

```
\begin{frame}
\frametitle{Немного картинок и анимации}

\begin{columns}

\begin{column}{0.65\textwidth}
\begin{itemize}
\item 4 базисных эрмитовых сплайна
\item Единичные значения или производные в 0 или 1
\end{itemize}
Кубический эрмитов сплайн — линейная комбинация базисных сплайнов.
\end{column}

\begin{column}{0.35\textwidth}
\begin{figure}
```

```
\includegraphics[width=4cm]{basis.png}
\end{figure}
\end{column}

\end{columns}

Кубический эрмитов сплайн используется для интерполяции функций.

\end{frame}
```

Добавим анимацию на предыдущий фрейм. Для этого в Beamer фрейм разбивается на несколько слайдов, которые называются оверлеи. Пользователю необходимо указать, на каких оверлеях будет присутствовать тот или иной объект фрейма. Это делается с помощью спецификаций оверлеев. Например, поменяем содержимое первой колонки нашего фрейма на следующее:

```
\begin{column}{0.65\textwidth}
\begin{itemize}
\item<1-> 4 базисных эрмитовых сплайна
\item<2-> Единичные значения или производные в 0 или 1
\end{itemize}
\uncover<3->{Кубический эрмитов сплайн — линейная комбинация
базисных сплайнов.}
\end{column}
```

Спецификации оверлеев указываются в угловых скобках. Например, <2-> в нашем примере означает, что второй элемент списка будет виден на втором и всех последующих оверлеях. Спецификации оверлеев можно применять к некоторым стандартным командам LaTeX, например `\item`, а также к специально определенным в классе beamer командам, таким как `\uncover`. Эта команда отображает свой аргумент только на тех оверлеях, которые заданы спецификацией оверлеев. На других оверлеях фрейма аргумент невидим, но занимает место на фрейме.

Теперь сделаем так, чтобы последнее предложение нашего фрейма сменялось рисунками с примерами. Для размещения группы рисунков на слайде на этот раз воспользуемся не окружением minipage, а пакетом subfigure, который подключим в преамбуле

```
\usepackage{subfigure}
```

Для спецификации оверлеев в данном случае используем специально определенную в beamer команду `\only`.

```
\only<4>{
Кубический эрмитов сплайн используется для интерполяции функций.
}
\only<5>{
  \begin{figure}
```

```
    \subfigure{
    \includegraphics[width=0.3\textwidth]{ex1.png}
    }
    \hspace{1.0cm}
    \subfigure{
    \includegraphics[width=0.3\textwidth]{ex2.png}
    }
    \caption{Примеры эрмитовой интерполяции}
  \end{figure}
}
\transdissolve<4-5>[duration=0.2]
```

Отличие `\only` от `\uncover` состоит в том, что на всех оверлеях, кроме заданных спецификацией оверлеев, ее аргумент будет выброшен и т.о. не будет занимать место на фрейме. Команда `\transdissolve` со спецификацией оверлеев задает эффект перехода к указанным оверлеям. Чтобы команда `\only` не приводила к сдвигам содержимого слайда по вертикали, опцией [t] выравниваем содержимое по верхнему краю слайда:

```
\begin{frame}[t]
```

Вернемся к внешнему виду нашей презентации. Попробуйте использовать другие темы презентации, например, CambridgeUS и PaloAlto. Вернемся к исходной теме и займемся настройкой отдельных тем и элементов презентации, которые в совокупности и определяют главную тему презентации. В преамбулу добавим

```
\usecolortheme[RGB={150,150,0}]{structure}
%цвет презентации
\useoutertheme{shadow}
%внешнее обрамление
\setbeamertemplate{items}[squares]
\setbeamertemplate{blocks}[rounded][shadow=true]
%настройка конкретных элементов
```

Наконец, добавим логотип университета на слайды, для чего в преамбулу добавим команду

```
\logo{\includegraphics[width=1cm]{logo.png}}
```

4.1 Задание

Оформите набранную Вами ранее статью в виде презентации с помощью пакета Beamer. Стиль либо по выбору преподавателя, либо по Вашему выбору.

Использование герба СПбГУ на первой станице презентации обязательно!

Перед тем как использовать изображение герба, почитайте инструкцию по применению герба
http://pr.spbu.ru/index.php/simvolika/gerb
там же можно скачать и файл с изображением в разных форматах.

Презентация должна содержать математические формулы, “выпадающие” пункты, выделение части текста или формул цветом, боксы, рисунок и другие динамические эффекты по Вашему желанию.

Работа № 5

Curriculum Vitae

В западных странах вместо привычного нам слова “resume” используют аббревиатуру CV. CV расшифровывается как Curriculum Vitae, что в переводе с латыни обозначает “ход жизни”. В Интернете можно найти огромное количество ресурсов, посвящённых тому, как “правильно” написать резюме. Если Вы подаёте CV в какой-либо конкретный Университет, то на его сайте как правило можно найти примеры и правила написания CV - например, можно посмотреть сайт Stanford University `https://studentaffairs.stanford.edu/cdc/resumes` .

Работодатель обычно не затрачивает много времени на прочтение Вашего резюме. Поэтому оно должно быть чётким, профессионально оформленным и не превышать одной - двух страниц. Однако не пытайтесь максимально наполнить эти две страницы информацией. В резюме важно суметь показать, что Вы умеете выделять самое главное. Лучше размещать информацию в порядке убывания её важности.

Существует определённая система построения резюме:

1. **Личная информация**

В первую очередь необходимо указать ФИО, адрес, телефон, дату рождения. Можно написать о семейном положении и национальности. Хоть такая информация и не обязательна, порой она важна для некоторых должностей. К примеру, если Вы хотите работать в международной компании, где требуется знание языка, являющегося для Вас родным, лучше тогда указать в резюме Вашу национальность.

2. **Образование**

Эта часть резюме одна из самых важных. Ведь именно из этого раздела у работодателя складывается первое впечатление о Вас. Самое сложное – правильно подобрать выражения на английском языке для описания Вашей специализации. Эта сложность обусловлена разницей в образовательных системах мира. Лучше всего написать Вашу специализацию латиницей, а затем в скобках дать эквивалент на английском языке. Указывать оценки также не стоит, т.к. оценочные системы разных стран различаются.

3. **Опыт работы**

Этот раздел может быть разделен на несколько подразделов, особенно если у Вас богатый опыт работы. Не бойтесь указывать все обязанности, которые Вы выполняли на всех работах, особенно если они имеют отношение к той должности, на которую Вы претендуете.

4. **Профессиональные навыки**

Здесь Вы можете указать те навыки и преимущества, которые, по вашему мнению, пригодятся в работе, например: наличие водительского удостоверения, знание иностранных языков, а также компьютерная грамотность.

5. **Интересы**

Конечно, работодатель не возьмёт Вас на работу только потому, что и Вы и он любите заниматься спортом. Но из этого раздела он сможет лучше понять, что Вы за человек. Но не переусердствуйте. Укажите только самое главное.

6. **Рекомендации**

Всегда хорошо иметь в резюме имена нескольких человек, готовых дать Вам рекомендацию. В идеале один должен быть Вашим преподавателем из университета, а другой - с прежней работы. Всегда указывайте полное имя и должность того, кто готов дать рекомендацию.

Стандартное Curriculum Vitae содержит:

- Фотография
- Личные данные (ФИО, возраст, пол, национальность, дата и место рождения, семейное положение)
- Адрес, номер контактного телефона и e-mail
- Сведения о полученном образовании (школа, колледж, высшее/неоконченное высшее)
- Зарубежные стажировки
- Название дипломной работы и имя научного руководителя
- Грамоты, награды (включая полученные гранты и стипендии)
- Опыт научной деятельности
- Опыт работы
- Профессиональные навыки
- Публикации или презентации
- Исследовательские работы/диссертации (с кратким описанием)
- Сертификаты
- Знание иностранных языков
- Членство в профессиональных организациях
- Общественная работа, членство в общественных организациях

- Интересы (не забудьте упомянуть о путешествиях, если они были осуществлены в страну работодателя)
- Рекомендации

5.1 Стандартные советы по написанию CV.

A curriculum vitae (singular), meaning "course of one's life, is a document that gives much more detail than does a resume about your academic and professional accomplishments. Curricula vitae (plural) are most often used for academic or research positions, whereas resumes are the preferred documents in business and industry. Note about plural / singular forms: "Curricula vitae"(vee-tie) is the plural form; "curriculum vitae"is singular. The informal shortened form, "vita"standing alone, meaning a brief autobiographical sketch (Webster's), is singular, while "vitae,"is plural. The abbreviation is often used: CV or CVs.

Curricula vitae are commonly used in applying for the following:

- Admission to graduate school or as part of an application packet for a graduate assistantship or scholarship.
- Grant proposals.
- Teaching, research, and upper-level administrative positions in higher education.
- Academic departmental and tenure reviews.
- College or university service appointments.
- Professional association leadership positions.
- Speaking engagements.
- Publishing and editorial review boards.
- Research and consulting positions in a variety of settings.
- School administration positions at the superintendent, principal, or department head level.

While your resume - even for most graduate students - should be kept to one page, vitae are usually two pages at the shortest, and can be many pages in length. Common lengths for curricula vitae are one to three pages for bachelor's and master's degree candidates; two to five pages for doctoral candidates; and five or more pages for an experienced academician or researcher. Even though it's a longer document, write it concisely and give it a clean, easy-to-read layout.

- Students completing a bachelor's degree rarely need a resume longer than one page. An exception might be for those who have extensive professional experience prior to completing the bachelor's degree.
- Graduate students may choose a curriculum vitae format, particularly doctoral students seeking positions in academia. Vitae typically extend to several pages, but are still written concisely. Common lengths for curriculum vitae are one to three pages for master's degree candidates; one to five pages for doctoral candidates; and five or more pages for an experienced academician or researcher. However, graduate students who are pursuing industry employment should do a concise one-page resume for that purpose.
- If your background justifies a two-page resume, keep in mind that the most important information should be contained on the first page, and that the second page should include your name and page number, in case the pages become separated once they are out of your hands.

A curriculum vitae includes information about professional publications, presentations, committee work, grants received, and other details based on each person's experience. You can include:

- Education
- Master's thesis or project
- Dissertation title or topic
- Course highlights or areas of concentration in graduate study
- Teaching experience and interests
- Research experience and interests
- Consulting experience
- Internships or graduate practice
- Fieldwork
- Publications
- Professional papers and presentations
- Grants received
- Professional association and committee leadership positions and activities
- Certificates and licensure
- Special training

- Academic awards, scholarships, and fellowships
- Foreign study and travel abroad
- Language competencies
- Technical and computer skills

Although curricula vitae are often similar to resumes, the preferred style, format, and content varies from discipline to discipline. Before writing a CV, you should become familiar with the requirements of your academic field by asking faculty members in your department and consulting professional associations for additional guidelines and examples. Career Services advisors can review your curriculum vitae and make suggestions.

5.2 Задание

Используя примеры с сайта CTAN – файлы `RaphaelPinson.tex`, `Template_en.tex`, `CVCTAN.tex`, `CV.tex` написать CV на английском языке.
На русском языке написать резюме либо в формате LaTeX, либо в пакете Word - включив либо свою фотографию, либо произвольное изображение из Интернета, подходящего размера.
Отчёты принимаются в pdf формате.

Работа № 6

Основы Mathematica

Предполагается, что мы будем работать с версией системы Mathematica не ниже 9. После запуска программы в главном окне выберем в меню Help пункт Function Navigator. Откроются страницы справки по функциям системы. Самостоятельно изучите, как они устроены.

- При работе с системой Mathematica мы будем иметь дело с алгебраическими выражениями, которые состоят из чисел, переменных и стандартных операций +, -, *, /, ^(возведение в степень), ! (факториал).

- Язык программирования системы Mathematica является интерпретируемым. Это означает, что в основном работа с системой производится следующим образом: пользователь вводит команду в область Input, система ее интерпретирует, производит необходимые вычисления, выдает результат в области Output и ждет ввода следующей команды в следующий Input.

- При наборе новой команды часто приходится пользоваться результатом ранее выполненной команды, и чтобы не перепечатывать его вручную, можно напечатать нужное число символов %. Например, % означает результат предыдущей выполненной команды, %%% — 3 с конца команды и т.д. Однако не стоит слишком увлекаться использованием символов % при создании последовательностей команд, это может привести к ошибкам при изменении порядка их следования или добавлении новых команд. Надежнее сохранять результаты выполнения команд в отдельных переменных.

- Система Mathematica работает с целыми числами, поэтому и в результаты выполнения команд входят только целые числа. Чтобы Mathematica выдала ответ в виде приближенного десятичного числа, ее об этом надо попросить. В частности, функция N[...] выдает десятичную запись числа, являющегося ее аргументом, с количеством значащих цифр по умолчанию. Функция N[...,n] выдает n значащих цифр.

- Константы π, e имеют обозначения Pi, E. Мнимая единица обозначается I. Вообще в Mathematica названия встроенных переменных и функций начинаются с большой буквы. Поэтому названия пользовательских переменных и функций должны начинаться с маленькой буквы.

- Греческие буквы набираются комбинациями клавиш типа Esc →a →Esc, Esc →b →Esc и т.д.
- Для того, чтобы набранная команда выполнилась, следует нажать комбинацию клавиш Shift+Enter (Enter — всего лишь переход на новую строчку).
- Наконец, отметим чрезвычайно полезную комбинацию клавиш Ctrl+L — она вставляет предыдущую выполненную команду в новый Input.

Теперь можно перейти непосредственно к работе с системой Mathematica. Создайте новый рабочий документ (notebook).

1. Вычислите $\sqrt[5]{112}$ и результат представьте в виде десятичного числа с точностью 20 значащих цифр.
2. Разделите 156 на 24. Вы получите ответ в виде несократимой дроби.
3. Представьте предыдущий результат в виде десятичного числа (используйте %).
4. Выполните команды

```
Head[3/4]
Rationalize[3.14]
```

   Функция Head, примененная к числу, выдает тип этого числа. Функция Rationalize представляет свой аргумент в виде несократимой дроби. Попробуйте заранее предположить, какой будет результат выполнения команды

```
Head[Rationalize[1.0]]
```

   Та же самая команда может быть записана в виде

```
Rationalize[1.0]//Head
```

   Постфиксная форма записи выражений (оператор "//") удобна в ситуациях, когда стоящую справа функцию нужно применить ко всему стоящему слева выражению.

5. Присвойте переменной v1 значение -3, используя знак равно.
6. Присвойте переменной v2 значение v1+5, затем v1 значение 15.
7. Посмотрите, изменилось ли значение v2: просто наберите v2 и нажмите Shift+Enter.
8. Как видите, значение v2 осталось прежним, хотя v1 уже не равно своему прежнему значению. Произошло так потому, что при определении v2 мы использовали так называемое мгновенное присваивание (при таком присваивании Mathematica не запоминает, что v2 равно v1+5).

9. Очистите значение переменной v2:

```
Clear[v2]
```

10. Чтобы заставить систему запомнить определение переменной, нужно использовать отложенное присваивание (:=)

```
v2:=v1+5
```

11. Посмотрите, чему равно v2.

12. Присвойте v1 значение x и вычислите v2. После этого, заставьте систему забыть определения всех переменных:

```
ClearAll["Global`*"]
```

13. Важную роль в системе Mathematica играют списки. Введите список {1,2,3,4,5} и присвойте его переменной numbers.

14. Введите список {2,3,5} и присвойте его переменной primes.

15. Извлечем третий элемент primes

```
primes[[3]]
```

16. Объединим списки функцией

```
Join[primes,numbers]
```

17. Добавим элемент 7 в primes

```
primes=Append[primes,7]
```

18. Исследуйте функции Prepend, Insert, Delete и Length, используя справку в режиме командной строки, например

```
?Prepend
```

    При нажатии на гиперссылку “»” открывается полноценная справка с примерами.

19. С помощью функции Insert из списка primes создайте список {2,3,11,5,7}.

20. Исследуйте функцию Table. Сгенерируйте список кубов натуральных чисел от 2 до 10:

```
Table[i^3,{i,2,10}]
```

21. Используйте Table, чтобы сгенерировать список степеней x от 0 до 10.

22. Векторы и матрицы в Mathematica являются списками и списками списков. Множества тоже задаются в виде списков. Вот некоторые функции для работы со списками, представляющими из себя множества:

```
Union[numbers, primes]
Intersection[numbers, primes]
```

    Упомянем еще одну полезную функцию Flatten, которая позволяет раскрывать вложенные списки:

```
Flatten[{{1, 2, 3}, {4, 5}}]
```

23. Теперь определим функцию. Правильное определение функции имеет вид

```
f[x_]:=x^2+3
```

    Здесь x_ — так называемый шаблон, который здесь обозначает формальный аргумент функции, вместо которого будет подставлено какое-то выражение при вызове этой функции.

24. Вычислите f[4].

25. Переопределите функцию в точке 4:

```
f[4]=100
```

26. Вычислите функцию в точках 4 и 5. Посмотрите определение функции

```
?f
```

27. Определите функцию двух переменных

```
z[x_,y_]:=Sin[x]Cos[y]
```

28. Вычислите значение функции z в точке 5, 3.

29. В языке программирования системы Mathematica нет типов, все объекты являются функциями вида f[x,y,...]. Чтобы убедиться в этом, выполните последовательность команд

```
expr=x*y+y
FullForm[expr]
TreeForm[expr]
```

Функция FullForm показывает внутреннее представление объекта в системе Mathematica (в данном случае алгебраического выражения expr), в котором он на самом деле хранится в системе. Функция TreeForm рисует это представление в удобной для восприятия пользователем форме дерева.

30. Исследуйте представление списка numbers

```
FullForm[numbers]
```

31. Представление объекта в Mathematica имеет иерархическую структуру, что видно на примере представления expr в виде дерева. Функция Head выдает функцию, которая находится на вершине этой иерархии (head объекта). Доступ же к объектам, которые находятся на разных уровнях этой иерархии, осуществляется с помощью функции [[...]], где в скобочках через запятую указывается путь по вершинам дерева к тому или иному объекту. Пример:

```
Head[expr]
expr[[1]]
expr[[2]]
expr[[2,1]]
```

32. Функция Map обозначается /@ и применяет функцию, написанную слева, к каждому объекту первого уровня объекта, стоящего справа. Пример:

```
f/@numbers
```

33. Не всегда хочется создавать отдельную функцию, чтобы использовать ее в функции Map. Вместо этого можно использовать анонимную функцию. Аргумент анонимной функции обозначается символом #, если он один, в случае же нескольких аргументов они обозначаются #1, #2 и т.д. (## — все аргументы). Запись анонимной функции завершается символом &. Вот пример того, как ту же самую функцию x^2+3 можно применить к каждому элементу списка numbers, используя анонимную функцию:

```
#^2+3 &/@numbers
```

34. Теперь с помощью анонимной функции и функции Map получите выражение, которое содержит сумму синусов от слагаемых выражения expr.

35. Функция Thread применяет функцию к указанным в виде списков наборам аргументов. Пример:

```
Thread[#1-#2+#3^2 &[{a,b},{x,y},{4,5}]]
```

36. Рассмотрим пример работы с объектами в системе Mathematica:

```
sq=Table[x[i],{i,10}]
TreeForm[sq]
sq2=Select[sq,#[[1]]<9  &]
```

Первым аргументом команды Select может быть только список. На выходе она выдает список, состоящий из тех элементов исходного списка, для которых значением логической функции — ее второго аргумента, является ИСТИНА.

```
sq3=#^#[[1]]  &/@sq2
Plus@@@sq3
```

Функция Apply, которая обозначается @@, заменяет head объета, который стоит справа, функцией, которая стоит слева. В данном случае мы заменили head списка sq3, т.е. List, на Plus, в результате вместо списка элементов получили сумму этих элементов.

6.1 Задания

Пример выполнения задания: Напишите функцию fourierCosFromList[l], которая по данному списку коэффициентов Фурье $\{c_0, c_1, \ldots, c_k\}$ строит тригонометрический ряд Фурье для четных функций $\sum_{n=0}^{k} c_n \cos nx$.

```
fourierCosFromList[l_]:=
Plus@@Table[l[[i]]Cos[(i-1)x],{i,1,Length[l]}]]
```

1. С помощью команды RandomInteger[10000000,1000000] сгенерируйте список из миллиона псевдослучайных целых чисел в диапазоне [0,10000000]. Выясните, имеются ли среди них биномиальные коэффициенты C_{20}^k?

2. Изучите функцию MapThread и с ее помощью напишите функцию replaceColumn[m,v,n], которая заменяет в квадратной матрице m n-й столбец вектором v.

3. Создайте функцию toepl[n], которая строит тёплицеву матрицу размера $n \times n$:

$$\begin{pmatrix} x[0] & x[-1] & x[-2] & \ldots & \ldots & x[-n+1] \\ x[1] & x[0] & x[-1] & \ddots & & \vdots \\ x[2] & x[1] & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & x[-1] & x[-2] \\ \vdots & & \ddots & x[1] & x[0] & x[-1] \\ x[n-1] & \ldots & \ldots & x[2] & x[1] & x[0] \end{pmatrix} \tag{6.1}$$

Вычислите toepl[10].

4. Изучите функцию Tuples и с ее помощью создайте функцию orbitalProjTriples[m,l1,l2,l3], где m — целое и l_j — целые неотрицательные числа. Функция должна возвращать список из всех последовательностей $\{m_1, m_2, m_3\}$ таких, что $m_1 + m_2 + m_3 = m$, где целые числа $m_j \in [-l_j, l_j]$. Для проверки выражений на равенство используйте оператор ==. Вычислите orbitalProjTriples[-3,5,2,3].

5. Вычислите с точностью 10 значащих цифр $\prod_i \left(\cos\left(i\pi/6\right) + 4/3\right)$, где умножение ведется по тем индексам i из диапазона [0,100], для которых выполняется $\sin(i\pi/10) > 1/2$.

Работа № 7

Полиномы, рациональные функции и упрощение выражений в системе Mathematica

Создадим новый рабочий документ, в котором вначале будем работать с полиномами.

1. Введите полином одной переменной

   ```
   (y^3−3y^2+5y−7)(2y^4−7y^3+16y^2−35y+30)
   ```

   и нажмите Shift+Enter. Обратите внимание, что при наборе выражений не нужно использовать символ умножения *, если это не приводит к недоразумениям.

2. Если вы ожидали, что Mathematica раскроет скобки в введенном полиноме, то вы ошиблись: ничего не произошло. Чтобы система раскрыла скобки, ее об этом нужно попросить, для чего существует специальная функция Expand:

   ```
   Expand[%]
   ```

   Заметим, что Mathematica выводит степени в порядке их возрастания слева направо, то есть ровно наоборот по сравнению со способом записи, к которому все привыкли еще в школе.

3. Теперь попросим Mathematica разложить результат предыдущего вычисления на множители, для чего опять же существует специальная функция Factor:

   ```
   Factor[%]
   ```

   Почему же две последние скобки не были разложены на линейные множители? Дело в том, что функция Factor по умолчанию осуществляет разложение полиномов на множители над полем целых чисел. Но корни полиномов (алгебраические числа), как известно, далеко не исчерпываются целыми числами. Например, вещественный корень полинома в последней скобке содержит радикалы.

4. В функции Factor предусмотрена возможность расширения поля чисел, над которым производится разложение полинома на множители. Пример:

```
Factor[x^2-8,Extension->Sqrt[2]]
```

5. Теперь посмотрим, какие в Mathematica существуют возможности работы с полиномами нескольких переменных. Введем полином нескольких переменных и назовем его p

```
p=(x+y+z)^3(x^2-3y)(z+x)
```

6. Функция Exponent позволяет определить степень полинома по одной из переменных

```
Exponent[p,x]
```

7. Функция CoefficientRules выдает список мономов, которые содержит полином, в виде {5,1,0}→3. Такая запись обозначает моном $\ldots + 3x^5y + \ldots$

```
CoefficientRules[p,{x,y,z}]
Expand[p]
```

8. Еще одна функция для преобразования записи полинома называется Collect

```
Collect[%,{z,y,x}]
```

   В данном примере функция Collect переписывает полином в виде полинома по z, коэффициентами которого являются полиномы по y и x, причем эти коэффициенты в свою очередь записаны как полиномы по y с коэффициентами, которые являются полиномами по x.

9. Перейдем к работе с рациональными функциями (отношения двух полиномов). Введите функцию $(x^3 - 1)/(x^2 - 1)$.

10. Ничего не произошло, хотя упрощение напрашивается само собой. Чтобы заставить Mathematica упростить эту дробь, нужно использовать специально для этого предназначенную функцию Cancel

```
Cancel[%]
```

11. Попробуем упростить дробь

```
f=Cancel[(x^3+4x^2-11x+3)/(x^2-9)]
```

12. Никаких сокращений не произошло. Действительно, в "большинстве" рациональных функций числитель и знаменатель все-таки не содержат общих множителей (ну разве только в школьном учебнике по математике). Разделим числитель на знаменатель дроби с остатком и проверим ответ:

```
p=Numerator[f]
q=Denominator[f]
qr=PolynomialQuotientRemainder[p,q,x]
qr[[1]]*q+qr[[2]]
Expand[%]
```

13. Введите $8/(x^2+3x-10)-x/(x^2-4)$.

14. Вновь с введенным выражением ничего не произошло. Используем специальную функцию для приведения дробей к общему знаменателю:

```
Together[%]
```

15. Хотелось бы в полученном результате раскрыть скобки в знаменателе. И для этого предусмотрена специальная функция

```
ExpandDenominator[%]
```

16. А почему для раскрытия скобок в знаменателе мы не воспользовались уже знакомой нам функцией Expand? Давайте попробуем:

```
Expand[%%]
```

    Как видите, это привело вовсе не к тому результату, который мы ожидали. Функция Expand в этой ситуации просто раскладывает числитель на слагаемые и представляет выражение в виде суммы дробей (такое тоже иногда нужно). Надеемся, этот пример убедит вас в том, как важно при работе с системой символьных вычислений представлять, что с конкретным объектом делает конкретная функция.

17. Наконец, функция Apart осуществляет разложение дроби на простейшие (что бывает очень полезно, например, при интегрировании рациональных дробей). В определенном смысле это функция, обратная к функции Together. Примените ее к дроби $(x-3)/(x^2+4x+4)$.

18. Для работы с тригонометрическими выражениями предусмотрены специальные функции. Попробуем вначале разложить тригонометрическое выражение на слагаемые с помощью хорошо знакомой нам функции Expand:

```
e=Sin[x+y]
Expand[e]
```

    Как видите, это ни к чему не привело. Expand умеет только раскрывать скобочки в выражениях, перемножая входящие в них слагаемые. Вместо нее в данном случае нужно воспользоваться специальной функцией TrigExpand, а также обратной к ней функцией TrigFactor:

```
TrigExpand[e]
TrigFactor[%]
```

Изучите самостоятельно важные команды TrigToExp и TrigReduce.

19. Наконец, существуют функции, предназначенные для работы с некоторыми другими элементарными и специальными функциями, такие как FunctionExpand:

```
l=Log[x  y]
FunctionExpand[l]
FunctionExpand[l,x>0 && y>0]
```

Обратите внимание на пробел между x и y в аргументе логарифма — без него Mathematica восприняла бы этот аргумент как одну переменную xy, а не как произведение двух переменных.

20. Теперь научимся делать подстановки в выражениях. Определим:

```
p=x^2+y^3
```

21. Чтобы сделать подстановку, выполним команду

```
p/.y->3
```

Функция /. (“slash dot”) использует правило подстановки, которое стоит справа, в выражении, которое стоит слева.

22. Проверьте, чему теперь равны p и y.

23. Перед тем, как выполнить следующие подстановки, попробуйте предположить, каков будет результат в первом и втором случаях:

```
p/.{x->y,y->x}
p/.x->y/.y->x
```

24. Теперь рассмотрим пример того, как можно использовать подстановки для упрощения выражений. Выполните команды:

```
p=(x+y)^2+1/(x+y)^2
Together[p]
```

Как видите, после приведения дробей к общему знаменателю мы получили более сложное выражение, чем можно было ожидать. Значительно эффективнее здесь действовать следующим образом:

```
ps=p/.x+y->z
Together[ps]
%/.z->x+y
```

25. До сих пор при упрощении выражений мы пользовались функциями "точечного" действия, которые делают конкретное действие с конкретным объектом (например, раскладывают дробь на простейшие). Существует, однако, универсальная функция Simplify, которая "просто упрощает выражение", причем выражение может быть тригонометрическим, содержащим логарифмы и другие элементарные функции. Однако эта универсальность, к сожалению, ведет к потере предсказуемости результата. Важно понимать, как работает функция Simplify. А работает она очень "просто": перебирает всевозможные преобразования выражения, применяя известные системе правила преобразования выражений данного типа (например, тригонометрические тождества), и выбирает самый короткий (по количеству символов) вариант. Важным аспектом использования функции Simplify является использование предположений. Выполните команды:

```
s=Sqrt[x^2]
Simplify[s]
```

Simplify не произвело никаких упрощений, поскольку по умолчанию аргумент x считается комплексным, и в зависимости от положения точки x на комплексной плоскости s может равняться как x, так и -x, что связано с многозначностью функции квадратный корень. Однако, ограничив область изменения аргумента x вещественной осью с помощью предположения, мы добьемся упрощения:

```
Simplify[s,x>0]
```

или

```
Simplify[s,x∈Reals]
```

где символ $\in$ набирается как Esc →el →Esc.

26. Как уже было отмечено выше, функция Simplify использует тригонометрические тождества для преобразования тригонометрических выражений. Пример:

```
Simplify[2Tan[x]/(1+Tan[x]^2)]
```

7.1 Задания

Пример выполнения задания: Докажите тригонометрическое тождество

$$1 + 2\cos x + 2\cos 2x + 2\cos 3x + \ldots + 2\cos nx = \frac{\sin(n+\frac{1}{2})x}{\sin\frac{x}{2}}.$$

при $n = 10$.

```
n=10
Plus@@Table[Cos[2 i  x],{i,n}]
(1+2*%)Sin[x]
res1=TrigExpand[%]
res2=TrigExpand[Sin[(2n+1)x]]
res2-res1
```

1. Напишите функцию remLeadingCoeff[n,p,x,y,z], которая выкидывает из данного полинома p трех переменных x,y,z все мономы, степень которых больше или равна n (например, степень монома $7x^2yz^2$ равняется 5). Вам понадобится функция FromCoefficientRules. С помощью remLeadingCoeff выкиньте из полинома $(x+y+z^2)^3(x^2-3y)(z+x^3)$ все мономы степени больше 5.

2. Напишите функцию dividedDiffPol[p,k], которая для данного полинома p степени n вычисляет разделенную разность k-го порядка

$$p(x_0;\ldots;x_k)=\sum_{j=0}^{k}\frac{p(x_j)}{\prod_{i\neq j}(x_j-x_i)}$$

в точках x[0], x[1], ..., x[k]. Добейтесь того, чтобы при k>n функция dividedDiffPol выдавала 0, а при k=n — старший коэффициент полинома p. Для проверки используйте полином $ax^4-x^3+bx-17$.

3. С помощью команд Mathematica преобразуйте выражение

$$x^2+2x+1+\frac{1}{x^2+2x+1}$$

в выражение

$$\frac{(x+1)^4+1}{(x+1)^2}$$

и наоборот. Используйте подстановки и встроенные функции, присваивания использовать нельзя!

4. Преобразуйте выражение

$$\frac{1-\frac{(x-b)^2}{(x-a)^2}}{1-\frac{(x-c)^{3/2}}{(x-a)^{3/2}}}$$

к наиболее простому виду. Присваивания использовать нельзя, только подстановки и встроенные функции Mathematica.

5. С помощью команд Mathematica покажите, что

$$2\ln\left(\frac{\sqrt{i}\ \sin(x/2+\pi/4)}{\sin(\pi/4-x/2)}\right)=\operatorname{arcch}\left(-i\ \operatorname{tg}(x)\right)-\operatorname{arcsh}\left(\operatorname{tg}(x)\right)$$

при $\pi/2<x<\pi$. Для этого вам понадобятся функции для работы с тригонометрическими выражениями, такие как `TrigExpand` и `TrigToExp`.

Работа № 8

Графика и решение уравнений в системе Mathematica

1. Изучите функцию Plot, которая предназначена для построения двумерных графиков, и ее опции:

```
?Plot
Options[Plot]
?AxesLabel
```

2. Построим график функции $x^8 \sin x$ на интервале [0,4]:

```
Plot[x^8Sin[x],{x,0,4},AxesLabel->{"x","y"}]
```

Обратите внимание, что если написать названия осей без кавычек, Mathematica вместо нужных нам букв подставит значения соответствующих переменных. Почему же Mathematica не построила график на всем интервале [0,4], хотя мы ясно ее об этом попросили? Система автоматически подбирает отображаемый интервал значений, иначе мы не различили бы экстремум (максимум) функции, т.к. при стремлении аргумента к значению 4 значения функции становятся очень большими отрицательными. Чтобы все-таки заставить систему отобразить весь интервал значений, нужно в предыдущей команде в функции Plot через запятую добавить опцию PlotRange->All.

3. Теперь изучим опцию PlotStyle, которая задает цвет и тип линии на графике

```
?PlotStyle
```

Для задания цвета на графиках можно использовать различные цветовые модели. Исследуйте команды:

```
?RGBColor
?Hue
```

Hue — команда, определяющая цвет графика с помощью цветового тона, который задается числами от 0 до 1, при этом соответствующие цвета пробегают по спектру красный–оранжевый–желтый–...–фиолетовый–красный. Пример:

```
colors=Table[Hue[i],{i,0,1,.1}]
sl=Table[Sin[i*t],{i,0,10}]
Plot[sl,{t,-Pi/4,Pi/4},PlotStyle->colors]
```

4. Функция Animate используется для создания анимации:

```
Animate[Plot[E^(-x^2)Sin[k*Pi*x],{x,-Pi,Pi}],{k,1,20},
                        AnimationRunning->False]
```

5. В случае зависимости между x и y, заданной в параметрическом виде $x(t)$ и $y(t)$, для построения графика используется функция ParametricPlot

```
ParametricPlot[{4Cos[t]-Cos[4t],4Sin[t]-Sin[4t]},{t,-Pi,Pi}]
```

   Получившаяся кривая является эпициклоидой. Это траектория, которую описывает точка на поверхности шара радиуса 1, который катится по шару радиуса 3.

6. Для построения графиков зависимостей, заданных в полярных координатах в виде $r(\theta)$, используется функция PolarPlot

```
PolarPlot[Cos[5t],{t,0,2Pi}]
```

   Попробуйте менять числовой коэффициент в аргументе косинуса, и вы убедитесь, что именно он определяет количество лепестков получившегося "цветочка".

7. Для построения графиков неявно заданных зависимостей y от x можно использовать функцию ContourPlot

```
ContourPlot[x^3+y^3-5x  y==0,{x,-3,3},{y,-3,3}]
```

   Мы получили самопересекающуюся кривую, которая называется лист Декарта. Начало координат является особой точкой этой кривой, в ней частные производные по x и y равны нулю. Обратите внимание, что в функции ContourPlot для задания уравнения кривой мы использовали не обычное, а двойное равно (==). Это логический оператор. Пример:

```
15==4
```

8. Для построения графиков функций двух переменных используется функция Plot3D:

```
Plot3D[x^2+y^4,{x,-2,2},{y,-2,2}]
```

9. Посмотрите, какие у этой функции имеются опции:

```
Options[Plot3D]
```

Добавьте опцию BoxRatios->{1,1,1}, которая определяет отношение сторон параллелепипеда, в котором содержится трехмерный график. По умолчанию используются значения {1,1,0.4}, и график выглядит сплюснутым по z. Отметим еще опцию PlotPoints, которая позволяет задавать количество точек сетки, по которой строится график, по каждой из переменных (увеличение количества точек улучшает гладкость графика, но увеличивает время его построения).

10. График неявно заданной зависимости z от x и y строится с помощью функции ContourPlot3D

```
ContourPlot3D[x^2-z*y^2==0,{x,-1,1},{y,-1,1},{z,-1,1}]
```

Эта самопересекающаяся по оси z поверхность называется зонтиком Уитни. Она также включает отрицательную часть оси z (ручка зонтика), но последняя на графике не отобразилась.

11. Для решения уравнений в системе Mathematica используется функция Solve:

```
?Solve
Solve[x^2+3(a+3)x-8a^2==23,x]
```

Как видите, Solve выдает ответ в виде правила подстановки.

12. Неудивительно, что системе удалось найти корни полинома второго порядка. А как насчет полинома четвертого порядка?

```
Solve[5x^4-7x^3+x^2+2x-13==0]
```

Скорее всего, результат превзошел все ваши ожидания. На самом деле мы получили пару комплексно сопряженных корней и два вещественных корня, в чем можно убедиться, вычислив их приближенные значения:

```
N[%]
```

13. Теперь примените функцию Solve для нахождения корней полинома пятого порядка $3x^5 - x^4 + 11x^3 - 8x^2 + 10x - 11$. Как видите, Mathematica на этот раз не удалось выразить корни полинома в радикалах. Попробуйте теперь вместо Solve использовать функцию NSolve, которая предназначена для нахождения приближенных решений уравнений. Замечание: вместо NSolve, в принципе, можно было применить функцию N к выводу функции Solve. Так удобнее управлять точностью приближений.

14. Но и функция NSolve, к сожалению, не всегда сработает. Попробуйте применить ее к уравнению $x \operatorname{tg} x - 1 = 0$, и увидите, что Mathematica не может найти все корни этого уравнения. Это, однако, не означает, что Mathematica не в состоянии найти никакой приближенный корень этого уравнения. Изучите функцию

```
?FindRoot
```

Как видите, функция FindRoot устроена так, что ей нужно указать начальное приближение для поиска корня уравнения. Постройте график нашей функции в интервале $x \in [-6, 6]$, по графику определите начальное приближение, близкое к одному из корней, и уточните это приближение с помощью функции FindRoot. Пример:

```
FindRoot[x  Tan[x]-1,{x,1}]
```

15. Допустим, мы хотим найти пересечения двух кривых

```
p1=x^2+x^3-y^2
p2=(x-1)^2+y^3-1
ContourPlot[{p1==0,p2==0},{x,-3,3},{y,-3,3}]
```

Для этого нужно решить систему двух уравнений высоких порядков. Это делается с помощью базиса Гребнера, нахождение которого в Mathematica осуществляется с помощью функции GroebnerBasis. На выходе этой функции мы получаем новую систему уравнений, которая имеет те же корни, что и исходная система. При этом в первое уравнение новой системы входит только одна переменная, во второе уравнение — та же переменная и еще одна переменная и т.д., так что новая система является в определенном смысле “треугольной”

```
g=GroebnerBasis[{p1,p2},{x,y}]
```

Теперь остается последовательно найти корни уравнений из новой системы

```
xv=Solve[g[[2]]==0,x]
Solve[g[[1]]==0]
yv=NSolve[g[[1]]==0]
xv/.yv
```

Для данной системы корни можно было найти и с помощью команд Solve и NSolve:

```
Solve[{p1==0,p2==0}]
NSolve[{p1==0,p2==0}]
```

8.1 Задания

Пример выполнения задания: Изучите функцию Eliminate, предназначенную для исключения переменных в системе уравнений. Пусть дан полином $x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$ с корнями x_1, x_2, x_3, x_4. Выразите коэффициенты полинома с корнями $x_1^2, x_2^2, x_3^2, x_4^2$ через коэффициенты исходного полинома.

```
?Eliminate
p=Times@@Table[x-x[i],{i,4}]
p2=Times@@Table[x-x[i]^2,{i,4}]
cl=CoefficientList[p,x]
cl=Delete[cl,5]
cl2=CoefficientList[p2,x]
cl2=Delete[cl2,5]
eqs=Join[Thread[cl==Table[c[i],{i,0,3}]],
                        Thread[cl2==Table[d[i],{i,0,3}]]]
eqs=Append[eqs,x^4+Plus@@Table[d[i]x^i,{i,0,3}]==0]
Eliminate[eqs,Join[Table[x[i],{i,4}],Table[d[i],{i,0,3}]]]
Collect[%[[2]]-%[[1]],x]
```

1. Постройте на одном графике в координатах P–V десять изотерм одного моля реального газа, который описывается уравнением Ван-дер-Ваальса

$$\left(P + \frac{a}{V^2}\right)(V - b) = RT.$$

   Изотерма, соответствующая более высокой температуре, должна быть окрашена в более "теплый" цвет. Используйте значения для аргона $a = 0,132 \frac{\text{Па}\cdot\text{м}^2}{\text{моль}^2}$, $b = 32 \cdot 10^{-6} \frac{\text{м}^3}{\text{моль}}$; диапазон объемов 33–1000 мл/моль и температур 100–300 K. $R = 8,31 \frac{\text{Дж}}{\text{моль}\cdot\text{K}}$.

2. Создайте 3D анимацию движения планеты радиуса 1 по эллиптической орбите с длинами большой и малой полуосей 10 и 5.

3. С помощью Mathematica перепишите уравнение кардиоиды

$$(x^2 + y^2 + 2x)^2 - 4(x^2 + y^2) = 0$$

   в виде зависимости $r(\theta)$ в полярных координатах. Полученную зависимость используйте для построения графика кардиоиды.

4. Энергия одномерного ангармонического осциллятора выражается формулой

$$E = \frac{mx'(t)^2}{2} + \frac{mw_0^2}{2}x(t)^2 + \frac{m\alpha}{3}x(t)^3 + \frac{m\beta}{4}x(t)^4.$$

   Исследуйте с помощью графика зависимость максимального отклонения от точки $x = 0$ от параметра w_0 при значениях $m = 1$, $E = 1$, $\alpha = 1$, $\beta = 1$.

5. В теории эллиптических функций и тета-функций важную роль играет специальная функция $q(m)$, которая называется параметр тета-функции, в системе Mathematica `EllipticNomeQ`[m]. Эта функция и ее производные выражаются через полные эллиптические интегралы первого и второго рода $K(m)$ и $E(m)$ (и Mathematica умеет это делать), в системе Mathematica `EllipticK`[m] и `EllipticE`[m]. Производная функции в Mathematica вычисляется с помощью функции D, например, D[EllipticNomeQ[m], m, m] — вторая производная. Составьте дифференциальное уравнение третьего порядка, которому удовлетворяет функция $q(m)$. Ответом считается выражение, содержащее пользовательскую функцию `q`[m] и ее производные до порядка 3 (но не содержащее $K(m)$ и $E(m)$!), такое, что "выражение = 0" — искомое уравнение.

Работа № 9

Дифференцирование, интегрирование и дифференциальные уравнения в системе Mathematica

1. Для дифференцирования функций в системе Mathematica предусмотрены две функции. Функция D предназначена для вычисления обыкновенных и частных производных произвольного порядка:

```
expr=Exp[Cos[x^2]]
D[expr,x]
D[expr,{x,3}]
D[x*y*z,x,y]
```

   Здесь последовательно были вычислены производные $\frac{\mathrm{d}}{\mathrm{d}x}$, $\frac{\mathrm{d}^3}{\mathrm{d}x^3}$, $\frac{\partial}{\partial x}\frac{\partial}{\partial y}$. Результатом на выходе функции D является выражение.

2. Функция Derivative обозначается символом ' и позволяет дифференцировать функцию одной переменной произвольное число раз.

```
f[x_]:=Evaluate[expr]
f'''
```

   Здесь мы вначале использовали функцию Evaluate для преобразования выражения expr в функцию f. Результатом на выходе функции Derivative является функция, так что можно вычислить ее значение в какой-нибудь точке:

```
f'''[1]
```

3. Для интегрирования функций, как неопределенного, так и определенного, в системе Mathematica предназначена функция Integrate

```
integr=(x^2+1)/(x^5-1)
Integrate[integr,x]
```

Проверим, что функция, которую мы получили, действительно является первообразной исходной функции:

```
res=D[%,x]
```

Что-то не очень похоже на исходное выражение! Однако не стоит спешить с выводами. Здесь мы столкнулись с общей бедой современных систем символьных вычислений — "раздуванием" выражений. На самом деле, integr и res — одно и то же выражение, в чем мы сейчас убедимся, упростив последнее из них. Однако прежде, чем пытаться свести одно выражение к другому, разумно сделать простой тест на их равенство. Сравним значения этих выражений при каком-то случайно выбранном значении переменной x:

```
r=RandomReal[]
integr/.x->r
res/.x->r
```

Теперь имеет смысл заниматься упрощением res:

```
Simplify[res]
```

4. Теперь используем Integrate для выполнения определенного интегрирования

```
Integrate[1/t^3,{t,1,z}]
```

Mathematica выдала ответ в виде условного выражения, поскольку этот ответ верен не при всех комплексных z. Например, при отрицательных z интеграл не существует из-за расходимости в точке t=0. Условного выражения можно избежать, используя предположение:

```
Integrate[1/t^3,{t,1,z},Assumptions->z>1]
```

5. Конечно, существуют такие функции, от которых Mathematica не сможет вычислить ни неопределенный, ни определенный интегралы:

```
Integrate[x^x,x]
Integrate[x^x,{x,1,2}]
```

Однако система всегда сможет посчитать приближенное значение определенного интеграла:

```
N[%]
```

6. С помощью функции Integrate можно вычислять многомерные интегралы, сведенные к повторным. Например, вычислим площадь единичного круга:

```
Integrate[1,{y,-1,1},{x,-Sqrt[1-y^2],Sqrt[1-y^2]}]
```

7. Суммирование рядов осуществляется с помощью функции Sum. Например, $\sum_{i=m}^{n} i$:

```
Sum[i,{i,m,n}]
```

8. Для разложения функций в ряды Тейлора существует функция Series. Например, разложим функцию

```
q=z/(1-h[z]-a*z^2)
```

где h[z] — какая-то неизвестная функция, a — параметр, в ряд Тейлора в окрестности точки z=0, до 4-го порядка включительно:

```
ts=Series[q,{z,0,4}]
```

Функция SeriesCoefficient выдает на выходе заданный коэффициент разложения в ряд Тейлора. Например, вычислим коэффициент при z^{10} разложения q в ряд Тейлора в окрестности точки z=0:

```
SeriesCoefficient[q,{z,0,10}]
```

Можете себе представить, сколько времени и усилий потребовалось бы для вычисления этого коэффициента на бумаге, Mathematica же выдает ответ мгновенно.

9. На выходе функции Series мы получаем специальный объект SeriesData (ряд):

```
Head[ts]
```

Ряды можно складывать и перемножать:

```
ts^2(1+ts)
```

При осуществлении операций с рядами удерживаются только члены тех порядков, знание которых обеспечивается точностью исходного ряда. Обычный полином из ряда можно получить с помощью функции Normal

```
Normal[ts]
```

10. Для решения дифференциальных уравнений предназначена функция DSolve. В качестве примера рассмотрим уравнение колебаний математического маятника, которое будем хранить в переменной pend:

```
pend=l θ''[t]+g Sin[θ[t]]==0
```

Попробуем найти линейно независимые решения этого уравнения:

```
DSolve[pend,θ[t],t]
```

Эти решения не выражаются через элементарные функции, а являются некоторыми специальными функциями. Дополним теперь уравнение начальными условиями (начальными отклонением и угловой скоростью маятника)

```
inits={θ[0]==Pi/6,θ'[0]==0}
ivp=Prepend[inits,pend]
```

и попробуем использовать DSolve для решения задачи Коши:

```
DSolve[ivp,θ[t],t]
```

Не удалось! Ничего не остается, кроме как попросить Mathematica решить уравнение приближенно. Чтобы получить численное решение задачи, нужно, конечно, присвоить параметрам l и g определенные значения:

```
ivp1=ivp/.{l->1,g->10}
```

Для приближенного решения задачи Коши на интервале $[0, 2\pi]$ применим функцию NDSolve

```
s=NDSolve[ivp1,θ,{t,0,2Pi}]
```

Mathematica выдает ответ в виде правила подстановки, в котором фигурирует InterpolatingFunction (функция, значения которой находятся интерполяционной процедурой). Это правило подстановки можно использовать для вычисления значения решения в точке или для построения графика:

```
θ[1]/.s
Plot[θ[x]/.s,{x,0,2Pi}]
```

9.1 Задания

Пример выполнения задания: Найдите решения радиального уравнения Шредингера с кулоновским потенциалом

$$-\frac{\mathrm{d}^2}{\mathrm{d}r^2}f(r) - \frac{1}{r}f(r) = Ef(r)$$

с начальными условиями $f(0) = 0$, $f'(0) = 1$ при значениях энергии $E = -1/4$, $E = -1/2$ и $E = -1$. Постройте графики решений на интервале $r \in [0, 8]$.

```
se=-f''[r]-(1/r)f[r]==e f[r]
bc={f[0]==0,f'[0]==1}
bvp=Prepend[bc,se]
DSolve[bvp/.e->-1,f[r],r]
DSolve[bvp/.e->-1/2,f[r],r]
DSolve[bvp/.e->-1/4,f[r],r]
```

```
rwf={%[[1]],%%[[1]],%%%[[1]]}
Plot[f[r]/.rwf,{r,0,8}]
```

1. Проверьте справедливость универсального соотношения

$$\left(\frac{\partial P(V,T)}{\partial V}\right)_T \left(\frac{\partial V(P,T)}{\partial T}\right)_P \left(\frac{\partial T(P,V)}{\partial P}\right)_V = -1$$

для уравнения Ван-дер-Ваальса состояния реального газа

$$\left(P + \frac{a}{V^2}\right)(V - b) = RT.$$

2. Вычислите объем, ограниченный поверхностями $z = \frac{1}{x^2+y^4+1}$, $x^2+y^2 = 1$, $z = 0$. Ответ с точностью 10 значащих цифр.

3. На интервале $[0, 7]$ построить график первых 10 приближений функции $\frac{1}{x^2+25}$ отрезком ряда Тейлора в точке 0.

4. Функция $t(x)$ на интервале $x \in [-1, 1]$ определена дифференциальным уравнением

$$(1 - x^2)t''(x) - xt'(x) + 25^2 t(x) = 0$$

и начальными условиями $t(0) = 0$, $t'(0) = \pi$. Убедитесь, что это полином (Чебышева).

5. Найти в явном виде приближенное решение задачи Коши

$$x''(t) + x(t) + \varepsilon x^3(t) = 0,\ x(0) = 0,\ x'(0) = 1$$

(осциллятор Дуффинга) в виде разложения в ряд $x(t) = \sum_{i=0}^{2} \varepsilon^i x_i(t)$ по малому параметру ε. Возьмите $x_0'(0) = 1$. Сравните полученное приближение с численным решением при $\varepsilon = 1/10$.

Работа № 10

Основы программирования в системе Mathematica

Главная особенность программирования в Mathematica состоит в том, что в языке программирования этой системы нет подпрограмм-процедур, все подпрограммы и управляющие конструкции (циклы, условные операторы и т.д.) являются функциями.

1. Создадим рекуррентную функцию для вычисления чисел Фибоначчи. Для этого выполним команду

```
fib[n_]:=fib[n-1]+fib[n-2]
fib[1]=fib[2]=1
```

   Вторая строка команды содержит начальные значения для вычисления последовательности. Попробуем вычислить 7-е и 30-е числа Фибоначчи:

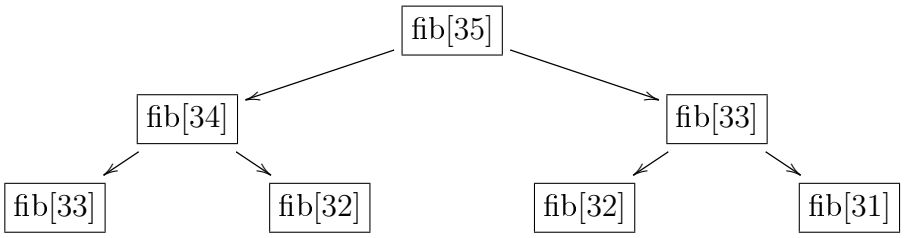
```
fib[7]
fib[30]
```

   Наверное, вы заметили, что для вычисления 30-го числа Фибоначчи компьютеру пришлось немного задуматься. Сколько же времени понадобится компьютеру для вычисления 35-го числа Фибоначчи? Чтобы точно ответить на этот вопрос, воспользуемся функцией Timing, которая помимо результата вычисления своего аргумента выдает также время, за которое это вычисление произошло:

```
fib[35]//Timing
```

   Во времена написания этого методического пособия вполне среднестатистическому компьютеру для выполнения команды fib[35] понадобилось около 14 секунд. Сколько же понадобится времени для вычисления 100-го числа с помощью функции fib? Очень много, и даже не пытайтесь ответить на этот вопрос с помощью компьютерного эксперимента. В чем же дело? Ведь при определенном усердии 100-е число Фибоначчи можно вычислить и на бумаге!

Из графа вызовов функции fib



видно, что количество арифметических операций, которые выполняет функция fib с данным аргументом, растет степенным образом с увеличением аргумента. Суть проблемы заключается в том, что функция fib не умеет запоминать уже вычисленные ею значения, в чем можно убедиться, посмотрев ее определение:

```
?fib
```

Создадим другую функцию newfib, которая запоминает вычисленные ею значения, командой

```
newfib[n_]:=newfib[n]=newfib[n-1]+newfib[n-2]
newfib[1]=newfib[2]=1
```

Вычислим 7-е число Фибоначчи и после этого проверим определение функции

```
newfib[7]
?newfib
```

Функция запомнила все числа Фибоначчи вплоть до 7-го. Теперь можно смело вычислять числа Фибоначчи с большими номерами

```
newfib[35]//Timing
newfib[100]
```

2. Создадим функцию, известную как ступенька: она равна единице на интервале [0,1] и нулю вне этого интервала. Это можно сделать с помощью функции If:

```
hat[x_]:=If[x<0,0,If[x>1,0,1]]
```

Построим ее график и вычислим производную

```
Plot[hat[x],{x,-1,3}]
D[hat[x],x]
```

К сожалению, производная вычислена неправильно. Другим способом написания ступеньки является использование встроенной функции Piecewise, предназначенной для создания кусочно непрерывных функций:

```
newhat[x_]:=Piecewise[{{0,x<0},{0,x>1}},1]
Plot[newhat[x],{x,-1,3}]
D[newhat[x],x]
```

Последним аргументом Piecewise является значение по умолчанию. На этот раз функция D определила, что в начале координат значение производной не определено.

3. Аргументом и значением функции также могут являться функции. Напишем функцию newton, которая своему аргументу (функции $f(x)$) сопоставляет функцию $x - f(x)/f'(x)$, которая используется в итерационном процессе $x_{n+1} = x_n - f(x_n)/f'(x_n)$ метода Ньютона нахождения корней нелинейных уравнений

```
newton[f_]:=#-f[#]/f'[#] &
```

Мы использовали анонимную функцию. Теперь можно заданной функции быстро сопоставить функцию для итерирования по методу Ньютона:

```
g[x_]:=Cos[x]-1/2
gstep:=newton[g]
gstep[x]
```

Возьмем в качестве начального приближения к корню уравнения $\cos x - 1/2 = 0$ значение 0.4 и произведем нужное количество итераций

```
gstep[0.4]
gstep[%]
gstep[%]
gstep[%]
```

4. Напишем функцию, которая дифференцирует полином произвольного порядка. Подразумевается, что полином может быть записан в произвольной форме, например, $(3x^2 + (x^3 + 1)^3)^2(1 - x)^2$, а также содержать разные переменные. Сначала воспользуемся более привычной парадигмой структурного программирования. Изучите, как устроена функция Which:

```
?Which
```

Используя Which, создадим интересующую нас функцию, которая дифференцирует поданное на вход (предположительно) полиномиальное выражение p по переменной x. Обратите внимание, что аргументы функций отделяются друг от друга запятой, при этом аргумент может быть последовательностью команд, которые отделяются точкой с запятой:

```
df[p_,x_]:=Module[{u,v},
Which[NumberQ[p],0,
Head[p]===Symbol, If[p===x,1,0],
Head[p]===Plus, df[#,x] &/@p,
Head[p]===Times, u=p[[1]]; v=p/u; df[u,x]*v+df[v,x]*u,
MatchQ[p, Power[_,_Integer]], u=p[[1]]; v=p[[2]];
                    v*df[u,x]*u^(v-1),
True, df::nopol="Vyrazhenie `1` ne polinom.";
                    Message[df::nopol,p]]
]
```

В функции df выражение p последовательно проверяется на соответствие разным шаблонам. Функция Module используется просто для объявления переменных u,v локальными переменными функции df. Во второй строке мы пользуемся справочной функцией NumberQ, которая выдает значение истина, если ее аргумент имеет head Complex, Integer, Rational или Real, т.е. является числом. Если p — число, то производная равна 0, если нет, идем дальше. В третьей строке проверяется, является ли p переменной. Если p — переменная x, производная по x равна 1, если другая переменная — производная 0, иначе идем дальше. Здесь мы воспользовались функцией ===, предназначенной для проверки на точное соответствие (а не математическое равенство, как ==) левой и правой частей. Эта функция всегда выдает True или False (функция == может остаться не вычисленной). В четвертой строке проверяется, является ли p суммой нескольких слагаемых. Если является, функция df возвращает сумму производных этих слагаемых, для чего функция df рекурсивно применяется к каждому из слагаемых с помощью функции Map (/@). В пятой строке выясняем, является ли p произведением двух или более сомножителей. Если является, находим первый сомножитель и применяем к нему и произведению оставшихся сомножителей правило дифференцирования произведения $(uv)' = u'v + uv'$, опять используя рекурсию. В шестой строке с помощью шаблонов, о которых пойдет речь чуть ниже, проверяем, является ли p выражением вида u^v с целым v. Если это так, применяем правило дифференцирования степени $(u^v)' = (v-1)u'u^{v-1}$. Наконец, если никакая из проверок не дала True, остается стандартным образом вывести сообщение об ошибке df::nopol — выражение p не является полиномом.

5. Ту же самую функцию в Mathematica более естественно написать, используя парадигму функционального программирования. Но для этого нам понадобится более подробно изучить использование шаблонов в Mathematica. Шаблон — это обозначение для выражений определенного типа. В обозначении самих шаблонов применяется символ нижнего подчеркивания _, который означает “какое угодно выражение”. Если нужно в дальнейшем использовать выражение, которое заменяет собой символ нижнего подчеркивания, слева от этого

символа пишется название этого выражения. Пример использования шаблона для замены в списке всех элементов вида “x в степени выражение” на другое выражение:

```
l={1,x,x^2,a/x,x^(a+b),x^x}
l/.x^n_->r[n]
```

Обратите внимание на то, что при сравнении с шаблоном x^n_ имеет значение внутреннее представление объекта — то, которое выдает функция FullForm. В предыдущем примере третий элемент списка a/x был преобразован, поскольку имеет внутренне представление Times[a,Power[x,-1]]. А вот второй элемент x не подпал под шаблон, поскольку имеет внутреннее представление Symbol["x"]. Если после символа нижнего подчеркивания написан head объекта, это означает, что выражение, которое заменяет собой нижнее подчеркивание, должно иметь именно этот head. В качестве примера, используя функцию Cases, которая отбирает элементы списка, подпадающие под указанный шаблон, отберем элементы списка l, которые имеют вид “что-то в степени целое число”

```
Cases[l,_^_Integer]
```

А теперь элементы l вида “что-то в степени целое число или сумма нескольких слагаемых”

```
Cases[l,_^(_Integer|_Plus)]
```

А еще “что-то в степени оно же само”

```
Cases[l,x_^x_]
```

Имейте ввиду, что по умолчанию функция Cases ищет соответствие шаблону на первом уровне внутреннего представления элементов списка. Оператор /; используется для обозначения шаблонов с условием:

```
fac[n_/;n>0]:=n!
fac[5]+fac[-1]
```

Двоеточие в шаблоне позволяет задавать значение по умолчанию:

```
f[x_,y_:1]:=x+y
f[z]
```

В некоторых встроенных функциях аргументы имеют значения по умолчанию. Для указания того, что аргумент встроенной функции может иметь значение по умолчанию, после символа нижнего подчеркивания ставится точка. Например, в функции x_+y_. значение по умолчанию y=0, в функциях x_y_. и x_^y_. значения по умолчанию y=1. Вернемся к примеру, в котором мы заменяли в списке l элементы вида “x в степени выражение”, но теперь укажем возможность значения по умолчанию показателя степени:

```
l /. x^n_.->r[n]
```

Теперь второй элемент списка также подпадает под шаблон.

6. Теперь мы можем создать функцию d для дифференцирования полиномов, используя парадигму функционального программирования. Зададим правило дифференцирования суммы:

```
d[y_+z_,x_]:=d[y,x]+d[z,x]
```

Научим нашу функцию дифференцировать произведения:

```
d[y_ z_,x_]:=z d[y,x]+y d[z,x]
```

Дифференцирование константы:

```
d[c_,x_]:=0/;FreeQ[c,x]
```

Справочная функция FreeQ проверяет вхождение x в выражение c. Остается научить функцию d дифференцировать степень, а также само x:

```
d[x_,x_]:=1
d[y_^n_,x_]:=n y^(n-1) d[y,x]
```

Проверим определение d

```
?d
```

Сравним результаты, которые выдают написанные нами функции:

```
d[(3x^2+(x^3+1)^3)^2(1-x)^2,x]
df[(3x^2+(x^3+1)^3)^2(1-x)^2,x]
```

10.1 Задания

Пример выполнения задания: Пусть функции $s(x)$ и $c(x)$ связаны соотношением $s^2(x) + c^2(x) = 1$. Напишите функцию replaceC[expr], которая в данном выражении expr вида $\sum_k \left(\pm c(x)^k \right)$ заменяет функцию c на функцию s.

```
replaceC[a_+b_]:=replaceC[a]+replaceC[b]
replaceC[Times[-1,a_]]:=-replaceC[a]
replaceC[m_Integer]:=m
replaceC[c[x_]^n_./;IntegerQ[n]]:=(1-s[x]^2)^(n/2)
expr=-1-c[x]+c[x]^4-c[x]^5
replaceC[expr]
```

1. Полиномы Чебышева определяются следующими рекуррентными соотношениями:
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),\ T_0(x) = 1,\ T_1(x) = x.$$
Вычислите $T_{100}(1/3)$.

2. Напишите функцию hermiteSplineInterp[f,xl], значением которой является функция — кубический эрмитов сплайн, интерполирующий значения и производные функции f в наборе точек x_k, заданном списком xl. На интервале (x_k, x_{k+1}) этот сплайн является кубическим полиномом вида

$$h_{00}(t)f(x_k) + h_{10}(t)(x_{k+1} - x_k)f'(x_k) + h_{01}(t)f(x_{k+1}) + h_{11}(t)(x_{k+1} - x_k)f'(x_{k+1}),$$

где $t = (x - x_k)/(x_{k+1} - x_k)$ и базисные сплайны

$$h_{00}(x) = 2x^3{-}3x^2{+}1,\ h_{10}(x) = x^3{-}2x^2{+}x,\ h_{01}(x) = -2x^3{+}3x^2,\ h_{11}(x) = x^3{-}x^2.$$

На одном графике постройте графики функции $e^{-x}\cos x$ и интерполирующего ее на отрезке $[0, 2\pi]$ по четырем точкам эрмитова кубического сплайна.

3. Напишите функцию delInfinitesimal[e,x,n], которая в данном выражении e отбрасывает все слагаемые, являющиеся в окрестности начала координат $o(x^n)$ с целым $n \geq 0$ (то есть убывающие быстрее, чем x^n). Предполагается, что все слагаемые допускают разложение в ряд Тейлора в окрестности $x = 0$. Вам, возможно, понадобится функция While. Удалите из выражения $\operatorname{tg} x \sin x/(x^2 + 25) + x + 1/(x - 1) + x(e^x - 1)$ все слагаемые порядка $o(x)$.

4. Напишите функцию fourierODE[ode,x], которая в данном однородном обыкновенном дифференциальном уравнении по переменной x ode с коэффициентами вида степени x заменяет все производные на степени, а степени на производные по правилу $x^k \mathrm{d}^n/\mathrm{d}x^n \to x^n \mathrm{d}^k/\mathrm{d}x^k$. Например, уравнение $a^2 g''''(t) - 5t^3 g''(t) + t^4 g'(t) - tg(t) = 0$ должно перейти в $a^2 t^4 g(t) - 5t^2 g'''(t) + tg''''(t) - g'(t) = 0$.

5. Используя удобную вам парадигму программирования — структурное или функциональное программирование, напишите функцию integrate, которая умеет интегрировать линейную комбинацию степеней $ax + b$, например, такую функцию:
$$2x^2 - 1 + (8x - 1)^2 - \frac{1}{1 + 5x} + 2(3(c + e)x + d)^3 + 3y(x - 1).$$

Работа № 11

Поля направлений и фазовые портреты ОДУ в Mathematica

Обыкновенное дифференциальное уравнение (ОДУ) определяет поле направлений. Это такое векторное поле, которое задает касательные к интегральным кривым ОДУ в каждой точке. Полем направлений ОДУ первого порядка с неизвестной функцией $y(x)$ является поле векторов с координатами $(1, y')$. В Mathematica векторные поля строятся с помощью функции VectorPlot. Например, поле направлений уравнения $y' = x/(2 + \sin y)$ можно построить командой

```
VectorPlot[{1,x/(2+Sin[y])},{x,-10,10},{y,-10,10}]
```

Добавление опций позволяет получить более наглядный график:

```
slopeField=VectorPlot[{1,x/(2+Sin[y])},{x,-10,10},{y,-10,10},
  Axes->True, VectorScale->{Automatic,Automatic,None}]
```

Опция VectorScale заставляет Mathematica рисовать все векторы одинаковой длины (все равно нас интересуют только их направления). Построим несколько интегральных кривых, задавая диапазон начальных условий от $y(0) = -8$ до $y(0) = 8$:

```
ySolutions=Table[y[x]/.NDSolve[
        {y'[x]==x/(2+Sin[y[x]]),y[0]==y0},y[x],{x,-10,10}],
              {y0,-8,8,2}]
plotSolutions=Plot[ySolutions,{x,-10,10},PlotStyle->Red]
```

А теперь поместим поле направлений и интегральные кривые на один график:

```
Show[slopeField,plotSolutions]
```

Рассмотрим системы ОДУ. Системы уравнений в Mathematica удобно задавать с помощью векторов и матриц. Как Вы помните, в Mathematica это просто списки и списки списков:

```
{{1,2,3},{3,5,2},{7,2,0}}//MatrixForm
```

Функция MatrixForm выводит заданную списками матрицу в привычной нам форме прямоугольной таблички. Вводить матрицы также удобнее в виде табличек. Для

этого нужно вызвать контекстное меню нажатием на правую кнопку мыши и выбрать пункт `Insert Table/Matrix`, а затем задать количество строк и столбцов. Произведение матрицы на вектор записывается с помощью оператора . (“dot”):

```
A={{2, 4}, {6, 8}}; B={{-3, 4}, {-5, 2}}
A.B//MatrixForm
```

Зададим и решим систему дифференциальных уравнений

```
mA={{1, -2}, {2, 1}}; vX[t_]={x[t], y[t]}
system=Thread[vX'[t]==mA.vX[t]]
vSolution[t_]=vX[t]/.DSolve[system, {x, y}, t]
```

Построим поле направлений нашей системы

```
directionField=VectorPlot[mA.{x, y}, {x, -10, 10}, {y, -10, 10},
  Axes->True, VectorScale->{Automatic, Automatic, None}]
```

Мы получили график на плоскости xy. Для построения на этой плоскости интегральных кривых (траекторий) воспользуемся параметрической зависимостью $x(t)$ и $y(t)$. Например, построим траекторию, проходящую через точку (2,-1):

```
vSolution[t_]=vX[t]/.DSolve[{system, Thread[vX[0]=={x0, y0}]},
                                    {x, y}, t]//Flatten
x0=2; y0=-1
ParametricPlot[vSolution[t], {t, 0, 2Pi}]
```

Функция Flatten убирает все внутренние скобочки во вложенных списках. Теперь построим набор интегральных кривых и поместим их вместе с полем направлений на один график:

```
y0=1; Clear[x0]
solutionCurves=Table[vSolution[t], {x0, -5, 5, 1}]
solutionPlot=ParametricPlot[solutionCurves, {t, -2Pi, 2Pi},
                                    PlotStyle->Red]
Show[directionField, solutionPlot]
```

В дополнение к траектории (графику в координатах xy), интересно также построить графики зависимостей $x(t)$ и $y(t)$, которые показывают развитие системы во времени:

```
x0=2; y0=-1
ParametricPlot[vSolution[t], {t, 0, 5}]
xTimePlot=Plot[vSolution[t][[1]], {t, 0, 5}, PlotStyle->Red]
yTimePlot=Plot[vSolution[t][[2]], {t, 0, 5}, PlotStyle->Green]
Show[xTimePlot, yTimePlot]
```

Рассмотренная выше система дифференциальных уравнений являлась автономной (правые части не зависели явно от независимой переменной t). Рассмотрим теперь систему

```
Clear[mA, vX, x, y, t, x0, y0, vSolution, vC]
mA={{1, -2}, {2, 1}}; vX[t_]={x[t], y[t]}; vC={t, t^2};
system=Thread[vX'[t]==mA.vX[t]+vC]
```

При повторном использовании переменных всегда лучше заставить Mathematica забыть их прежние значения с помощью команды Clear. Поле направлений новой системы зависит от t, то есть меняется с течением времени. Посмотрим на эту зависимость, построив набор графиков поля направлений в моменты времени $t = 0, 1, \ldots, 10$

```
Table[VectorPlot[Evaluate[mA.{x, y}+vC], {x, -10, 10}, {y, -10, 10},
  Axes->True, VectorScale->{Automatic, Automatic, None}], {t, 0, 10, 1}]
```

Эти графики мало полезны: по ним невозможно представить себе вид траектории (по мере продвижения радиус-вектора по траектории поле направлений постоянно меняется). Однако построение траектории с помощью параметрической зависимости $x(t)$ и $y(t)$ все так же просто:

```
vSolution[t_]=vX[t]/.DSolve[{system, Thread[vX[0]=={x0, y0}]},
                                        {x, y}, t]//Flatten
x0=2; y0=-1;
ParametricPlot[vSolution[t], {t, 0, 2Pi}]
```

До сих пор мы рассматривали только линейные системы дифференциальных уравнений. Рассмотрим теперь нелинейную автономную систему $x' = f(x, y)$, $y' = g(x, y)$ с

```
f[x_, y_]:=x^2+2y^2-2; g[x_, y_]:=-4x^2+3y^2
```

Особыми точками этой системы ОДУ называются такие точки, в которых одновременно $f(x, y) = 0$ и $g(x, y) = 0$. Найдем их:

```
critPts=Solve[{f[x, y]==0, g[x, y]==0}]//N
```

Для исследования характера поведения интегральных кривых в окрестности особых точек дифференциальные уравнения линеаризуют, то есть приближают линейными уравнениями. Матрица линеаризованного уравнения имеет вид матрицы первых производных правых частей системы, вычисленных в особой точке. В случае первой особой точки имеем

```
jacob[f_, g_]:= {{D[f, x], D[f, y]}, {D[g, x], D[g, y]}}
jacob[f[x, y], g[x, y]]
jacob[f[x, y], g[x, y]]/.critPts[[1]]//MatrixForm
```

Характер же поведения интегральных кривых линейного уравнения в окрестности особой точки определяется собственными значениями матрицы правой части:

```
Eigenvalues[jacob[f[x,y],g[x,y]]]/.critPts[[1]]
```

Определим собственные значения во всех особых точках:

```
Eigenvalues[jacob[f[x,y],g[x,y]]]/.critPts//TableForm
```

Отсюда следует, что в первом и четвертом случаях особые точки являются фокусами, а траектории — спиралями, а во втором и третьем случаях особые точки называются седло, а траектории являются семейством гипербол. Чтобы убедиться в справедливости этих утверждений, отобразим на одном графике поле направлений, критические точки и изоклины, соответствующие вертикальному и горизонтальному уклону поля направлений

```
vField=VectorPlot[{f[x,y],g[x,y]},{x,-2,2},{y,-2,2},Axes->True,
   VectorStyle->Green,VectorScale->{Automatic,Automatic,None}]
critPtsPlot=ListPlot[{x,y}/.critPts,
                              PlotStyle->{Red,PointSize[0.02]}]
nullClines={f[x,y]==0,g[x,y]==0}
nullClinePlot=ContourPlot[Evaluate[nullClines],
                                        {x,-2,2},{y,-2,2}]
Show[vField,nullClinePlot,critPtsPlot]
```

Остается добавить на график траектории. Это было бы совсем несложно сделать, если бы как в предыдущих примерах Mathematica нашла аналитическое решение системы. Но в данном случае ей это не удается:

```
DSolve[{x'[t]==f[x[t],y[t]],y'[t]==g[x[t],y[t]],x[0]==x0,
                                                  y[0]==y0},{x,y},t]
```

Поэтому придется довольствоваться численным решением. Построим траекторию, проходящую через точку (1,0)

```
ParametricPlot[Evaluate[{x[t],y[t]}/.
  First[NDSolve[{x'[t]==f[x[t],y[t]],y'[t]==g[x[t],y[t]],x[0]==1,
                              y[0]==0},{x,y},{t,0,2}]]],{t,0,2}]
```

А теперь через точку (1,-1)

```
ParametricPlot[Evaluate[{x[t],y[t]}/.
  First[NDSolve[{x'[t]==f[x[t],y[t]],y'[t]==g[x[t],y[t]],x[0]==1,
                              y[0]==-1},{x,y},{t,0,2}]]],{t,0,2}]
```

Mathematica сообщает нам, что при вычислении решения правее точки $t \approx 0,54$ возникли проблемы. Числа на осях абсцисс и ординат графика дополнительно сообщают нам, что проблемы эти серьезные. Простым выходом из ситуации может быть уменьшение интервала по t:

```
ParametricPlot[Evaluate[{x[t],y[t]}/.
  First[NDSolve[{x'[t]==f[x[t],y[t]],y'[t]==g[x[t],y[t]],x[0]==1,
                                  y[0]==-1},{x,y},{t,0,0.5}]]],{t,0,0.5}]
```

Автоматизируем процесс подбора приемлемых интервалов по t и построения графиков для списка начальных точек с помощью функцию TrajectoryPlot (уж очень утомительно делать это для каждой точки вручную!)

```
TrajectoryPlot[system_,initialPts_,{time_,tMin_,tMax_},
 {xVar_,xMin_,xMax_},{yVar_,yMin_,yMax_}]:=Module[{},
 Off[NDSolve::ndsz];
 Off[InterpolatingFunction::dmval];

 getSolution[k_]:=First[
  NDSolve[Join[system,{x[0]==initialPts[[k,1]],
          y[0]==initialPts[[k,2]]}],{x,y},{t,tMin,tMax}]];

 getEstDomain[funct_]:=
  {t,Flatten[funct[[1]][[1]]]+0.2,Flatten[funct[[1]][[2]]]-0.2};

 Block[{$DisplayFunction=Identity},
  solutionList=
   Table[ParametricPlot[Evaluate[{x[t],y[t]}/.getSolution[k]],
                        Evaluate[getEstDomain[x/.getSolution[k]]],
    PlotRange->{{xMin,xMax},{yMin,yMax}},AspectRatio->Automatic,
                    PlotPoints->100],{k,1,Length[initialPts]}];];
 On[NDSolve::ndsz];
 On[InterpolatingFunction::dmval];

 Show[solutionList]
]
```

Теперь можно задать набор точек, построить траектории, проходящие через эти точки с помощью функции TrajectoryPlot, и поместить их на график с полем направлений

```
points={{1,1},{1,-1},{-1,1},{-1,-1}}
solCurves=TrajectoryPlot[{x'[t]==f[x[t],y[t]],
    y'[t]==g[x[t],y[t]]},points,{t,-100,100},{x,-2,2},{y,-2,2}]
Show[vField,nullClinePlot,critPtsPlot,solCurves,
                                  PlotRange->{{-2,2},{-2,2}}]
```

Добавим еще точки

```
points={{1,1},{1,-1},{-1,1},{-1,-1},{0,.25},{-1,-2},{0,-2},
   {0,.8},{-1,1.5},{1.5,1.5},{-.5,1},{.5,-2},{-1,1.2},{0.5,0}};
```

```
solCurves=TrajectoryPlot[{x'[t]==f[x[t],y[t]],
    y'[t]==g[x[t],y[t]]},points,{t,-100,100},{x,-2,2},{y,-2,2}]
Show[vField,critPtsPlot,solCurves,
                                    PlotRange->{{-2,2},{-2,2}}]
```

Наконец, сгенерируем большой набор начальных точек

```
points=Flatten[Table[{x,y},{x,-2,2,0.5},{y,-2,2,0.5}],1]
solCurves=TrajectoryPlot[{x'[t]==f[x[t],y[t]],
    y'[t]==g[x[t],y[t]]},points,{t,-100,100},{x,-2,2},{y,-2,2}]
Show[vField,critPtsPlot,solCurves,
                                    PlotRange->{{-2,2},{-2,2}}]
```

Работа № 12

Метод ВКБ для одномерного уравнения Шредингера в Mathematica

Рассмотрим реализацию метода ВКБ для решения одномерного уравнения Шредингера с помощью системы Mathematica. Состояние квантовой частицы массы m в поле потенциала $V(x)$ описывается волновой функцией $\psi(x)$, являющейся решением уравнения Шредингера

$$\frac{\hbar^2}{2m}\psi''(x) + \big(E - V(x)\big)\psi(x) = 0. \tag{12.1}$$

Здесь $\hbar$ — постоянная Планка, универсальная физическая константа. Пусть для удобства $2m = 1$, что соответствует выбору системы единиц. Для нахождения так называемых связанных состояний частицы решают задачу Штурма-Лиувилля: ищут решения уравнения (12.1), дополненного нулевыми граничными условиями на бесконечности, и набор допустимых значений параметра E (энергия частицы), при которых эти решения существуют.

Метод ВКБ — метод приближенного решения уравнения (12.1), основанный на наличии в нем малого параметра — постоянной Планка $\hbar$. Анзац (подстановка)

$$\psi(x) = \frac{1}{\sqrt{S'(x)}} e^{S(x)/\hbar}$$

приводит к уравнению на функцию $S(x)$

$$S'^2(x) + \big(E - V(x)\big) - \frac{1}{2}\hbar^2 \left\{ \frac{S'''(x)}{S'(x)} - \frac{3}{2}\left(\frac{S''(x)}{S'(x)}\right)^2 \right\} = 0. \tag{12.2}$$

Решения этого уравнения ищутся в виде ряда по четным степеням $\hbar$

$$S(x) = \sum_{m=0}^{\infty} S_m(x)\hbar^{2m}. \tag{12.3}$$

Производные $S'_m(x)$ находятся подстановкой ряда (12.3) в уравнение (12.2) и приравниванием нулю коэффициентов при степенях $\hbar^{2m}$ с $m = 0, 1, 2, \ldots$. В случае

$m = 0$ это дает $S'_0(x) = \sqrt{V(x) - E}$. При $m > 0$ получим соотношения, которые выражают производные $S'_m(x)$ через производные с меньшими индексами; по ним функции $S'_m(x)$ находятся рекуррентно. Можно показать [1], что для того, чтобы волновая функция $\psi(x)$ являлась решением задачи Штурма-Лиувилля, функции $S'_m(x)$ должны удовлетворять условию (квантования)

$$\frac{1}{\hbar \mathrm{i}} \sum_{m=0}^{\infty} \hbar^{2m} \int_{x_1}^{x_2} \mathrm{d}x \, S'_m(x) = \pi(n + 1/2), \tag{12.4}$$

где n — положительное целое число, x_1 и x_2 — точки поворота, то есть решения уравнения $V(x) = E$. Из этого условия находятся подходящие значения энергии E.

С помощью Mathematica получим выражения для функций $S'_m(x)$. Подстановкой в уравнение

$$S'^4(x) + \big(E - V(x)\big)S'^2(x) - \frac{1}{2}\hbar^2 S'''(x)S'(x) + \frac{3}{4}\hbar^2 S''^2(x) = 0,$$

которое получается домножением обеих частей равенства (12.2) на $S'^2(x)$, ряда (12.3) и обнулением коэффициента при $\hbar^{2m}$ приходим к формуле

$$S'_m(x) = \frac{1}{2S'_0(x)\Big(2S'^2_0(x) - \big(V(x) - E\big)\Big)} \Bigg(- \sum_{\substack{i,j,k,l=0 \\ i+j+k+l=m}}^{m-1} S'_i(x)S'_j(x)S'_k(x)S'_l(x) + \\ + \big(V(x) - E\big) \sum_{\substack{i,j=0 \\ i+j=m}}^{m-1} S'_i(x)S'_j(x) + \frac{1}{2} \sum_{\substack{i,j=0 \\ i+j=m-1}}^{m-1} S'''_i(x)S'_j(x) - \frac{3}{4} \sum_{\substack{i,j=0 \\ i+j=m-1}}^{m-1} S''_i(x)S''_j(x) \Bigg).$$

Реализуем эту формулу в виде рекурсивной функции для вычисления функций $S'_m(x)$:

```
s[0]=Sqrt[(v[x]-e)]
s[n_Integer/;n>0]:=s[n]=Expand[
((-Sum[s[n-l-k-j]s[j]s[k]s[l],{l,0,n-1},{k,0,Min[n-l,n-1]},
                {j,If[l+k==0,1,0],Min[n-l-k,n-1]}]+
(v[x]-e)If[EvenQ[n],2Sum[s[n-j]s[j],{j,1,n/2}]-s[n/2]^2,
                                2Sum[s[n-j]s[j],{j,1,(n-1)/2}]]+
(1/2)Sum[D[s[n-1-j],{x,2}]s[j],{j,0,n-1}]-(3/4)If[EvenQ[n-1],
  2Sum[D[s[n-1-j],x]D[s[j],x],{j,0,(n-1)/2}]-D[s[(n-1)/2],x]^2,
                2Sum[D[s[n-1-j],x]D[s[j],x],{j,0,(n-2)/2}]])/
(s[0](4s[0]^2-2(v[x]-e))))/.{(v[x]-e)->q}
]/.{q->(v[x]-e)};
s[2]
```

Вычисление двойных сумм оптимизировано за счет симметрии слагаемых.

Если подставить полученные выражения для $S'_m(x)$ в условие квантования (12.4), получим интегралы от входящих в них слагаемых. Эти интегралы будут расходящимися из-за слишком сильных особенностей в точках x_1 и x_2. На самом деле,

в формуле (12.4) стоят регуляризованные версии этих интегралов. Мы не будем здесь обсуждать, что такое регуляризация, а лишь опишем некоторый конкретный алгоритм [4] преобразования этих интегралов к сходящимся. На первом этапе, входящие в выражения для $S'_m(x)$ слагаемые интегрируются по частям. Согласно процедуре регуляризации, внеинтегральные члены при этом можно отбрасывать. Вначале проинтегрируем по частям те слагаемые, которые содержат производные от потенциала $V'(x)$, внося под знак дифференциала множитель $(V(x)-E)\hat{}(\ldots)$. При каждом таком интегрировании степень знаменателя будет понижаться на единицу. К сожалению, стандартная функция Integrate в данном случае не сработает:

```
Integrate[v'[x]v''''[x]/(v[x]-e)^(7/2),x]
```

Поэтому нам придется написать собственную функцию

```
(*Интегрирование по частям слагаемого с V'(x)*)
IntegrationOverV[a_. v'[x]^n_.(v[x]-e)^p_/;p<-1]:=
        Expand[-D[a v'[x]^(n-1),x](v[x]-e)^(p+1)/(p+1)]

(*Применение к отдельным слагаемым*)
IntegrationOverV[x_Plus]:=IntegrationOverV/@x

(*Слагаемое без V'(x) не меняем*)
IntegrationOverV[x_/;FreeQ[Numerator[x],v'[x]]]:=x

IntegrationOverV[v'[x]v''''[x]/(v[x]-e)^(7/2)]
```

Далее, можно проинтегрировать по частям те слагаемые, в которых числители являются полными дифференциалами. Однако при каждом таком интегрировании по частям степень знаменателя будет увеличиваться на единицу. Поэтому эту операцию мы будем применять только к тем слагаемым в выражении для $S'_m(x)$, в которых степень знаменателя не является максимальной среди всех входящих в выражение для данного $S'_m(x)$ знаменателей. Таким образом максимальная степень знаменателей увеличиваться не будет. Напишем вспомогательную функцию IntegrationByParts, которая совершает интегрирование по частям слагаемых вне зависимости от степеней их знаменателей:

```
IntegrationByParts[x_Plus]:=IntegrationByParts/@x

IntegrationByParts[a_. (v[x]-e)^p_]:=Module[{int},
  int=Integrate[a,x];
(*Интегрируем по частям только если числитель — полный дифференциал*)
  If[FreeQ[int,Integrate], Expand[-int p (v[x]-e)^(p-1)v'[x]],
        a (v[x]-e)^p]
]

IntegrationByParts[s[2]]
```

Теперь создадим функцию IntegrationByPartsTogether, которая применяет функцию IntegrationByParts только к слагаемым с меньшей, чем максимальная, степенью знаменателя

```
IntegrationByPartsTogether[y_Plus]:=Module[{y1,max,toIntegrate},
  (*список степеней знаменателей*)
  y1=(Denominator/@(List@@y))/.{_. _^m_->m};
  max=Max[y1];
  (*список слагаемых со степенью, меньшей максимальной*)
  toIntegrate=Cases[y,_. (v[x]-e)^p_/;p>=max];
  (*применяем IntegrationByParts только к этим слагаемым*)
  (y-(Plus@@toIntegrate))+
    Plus@@(IntegrationByParts/@toIntegrate)
]

(*Обработка случая единственного слагаемого*)
IntegrationByPartsTogether[x_]:=x

IntegrationByPartsTogether[s[2]]
```

Для приведения к общему знаменателю слагаемых, содержащих одинаковые степени $V(x)-E$ в знаменателях, напишем вспомогательную функцию

```
myTogether[x_Plus]:=
Plus@@((ExpandNumerator[
  (*применяем Together только к слагаемым с одинаковыми степенями
    V(x)-E*)
  Together[Plus@@(Cases[x,_. #^-1]/.{a_. #^-1->a})] #^-1])&/@
    (*формируем список (V(x)-E)^(...) с разными степенями*)
    Union[Flatten[Cases[#,_^_,{0,Infinity}]&/@
                  (Denominator/@List@@x)]])

(*Обработка случая единственного слагаемого*)
myTogether[x_Times]:=x

myTogether[IntegrationByPartsTogether[s[2]]]
```

На втором этапе преобразования интегралов степени знаменателей подынтегральных выражений понижаются за счет представления этих выражений в виде производных по энергии E. Производные в нужном количестве выносятся за знак интеграла, после чего интегралы становятся сходящимися. Напишем вспомогательную функцию:

```
(*для данного слагаемого возвращает список, содержащий степень производной
  и выражение под знаком производной*)
finalFormAux[a_. (v[x]-e)^p_]:=
  {-(p+1/2),a*
```

```
    (*числовой коэффициент из-за производных по E*)
    Product[-1/i,{i,p+1,-1/2}](v[x]-e)^(-1/2)}

finalFormAux[v'[x]v'''[x]/(v[x]-e)^(5/2)]
```

Теперь напишем функцию finalForm, которая формирует окончательное выражение для интеграла от $S'_m(x)$ с суммой производных по E от сходящихся интегралов. Для наглядности избавляемся от числовых коэффициентов в знаменателях, вынося общий множитель за скобку:

```
finalForm[y_]:=Module[{pre,lcm},
  (*применяем finalFormAux к каждому слагаемому*)
  If[Head[y]===Plus,
    pre=finalFormAux/@List@@y,
    pre={finalFormAux[y]}]
  (*НОК числовых коэффициентов знаменателей*)
  lcm=LCM@@((Denominator[#[[2]]]/.{n_.  (v[x]-e)^_->n})&/@pre);
  (*формируем окончательное выражение*)
  1/lcm  Plus@@Table[HoldForm[
                              D[Integrate[expr,{x,x1,x2}],{e,u}]]/.
          (*подставляем вывод finalFormAux для i-го слагаемого*)
          {expr->lcm
              pre[[i,2]],u->pre[[i,1]]},{i,Length[pre]}]
]

finalForm[0]:=0

finalForm[myTogether[IntegrationByPartsTogether[s[2]]]]
finalForm[5v'[x]v'''[x]/(8(v[x]-e)^(5/2))]
```

Использованная здесь функция HoldForm оставляет свой аргумент невычисленным (без нее Mathematica вычисляла бы производные по E).

Объединим все этапы преобразования интегралов от $S'_m(x)$ в функции

```
WKBCorrection[ord_Integer/;ord>0]:=finalForm[myTogether
  [FixedPoint[IntegrationByPartsTogether,
                FixedPoint[IntegrationOverV,s[ord]]]]]
WKBCorrection[1]
WKBCorrection[2]
WKBCorrection[3]
WKBCorrection[4]
WKBCorrection[5]
```

Функция FixedPoint, стартуя с выражения, указанного вторым аргументом, применяет функцию — свой первый аргумент до тех пор, пока результат не перестанет

меняться. Таким образом, функция WKBCorrection применяет функции IntegrationOverV и IntegrationByPartsTogether максимально возможное число раз.

Определим приближенное значение энергии E в случае потенциала $V(x) = x^4$. Точками поворота являются точки $x_{1,2} = \pm E^{\frac{1}{4}}$. Подставим эти выражения в полученные выше интегралы и вычислим их:

```
tab=Table[Expand[DeleteCases[WKBCorrection[i]/.{x1->-e^(1/4),
  x2->e^(1/4),Derivative[n_][v][x]:>D[x^4,{x,n}],v[x]->x^4}/.
  Integrate[f_,range_]:>Integrate[f,range,
                                  Assumptions->e>0&&e^(1/4)>0],
    HoldForm,Infinity,Heads->True]],{i,5}]
int0=Integrate[s[0]/.v[x]->x^4,{x,-e^(1/4),e^(1/4)},
                                  Assumptions->e>0&&e^(1/4)>0]
```

В случае использования оператора :> вместо оператора ->, стоящий справа от него аргумент вычисляется только после совершения подстановки. Функция DeleteCases здесь убирает из внутреннего представления выражений функцию HoldForm для того, чтобы вычислились интегралы и производные. Теперь создадим функцию, которая проверяет выполнение условия квантования (12.4) с точностью m слагаемых:

```
quantizationCondition[n_,h_,m_]:=
  (1/I)int0+(1/I)Sum[h^(2k)tab[[k]],{k,1,m}]-Pi(n+1/2)h
```

Возьмем $n = 3$, $\hbar = 1/100$ и определим графически значение энергии E, при котором условие квантования (12.4) выполняется, следя за сходимостью по количеству слагаемых:

```
Plot[Evaluate[Table[quantizationCondition[3,1/100,m],{m,5}]],
  {e,0,0.03},PlotRange->{-3/10,+3/10},
        PlotStyle->Table[Hue[i/6],{i,5}]]
```

Из графика видно, что искомое значение энергии приближенно равняется 0,025. Более точное значение определяется хорошо известными методами приближенного решения нелинейных уравнений. Это может быть сделано читателем в качестве упражнения.

Литература

[1] С. Ю. Славянов. *Асимптотика решений одномерного уравнения Шредингера.* Издательство ЛГУ, 1990.

[2] Е. М. Балдин. *Компьютерная типография LaTeX.* БХВ-Петербург, 2008.

[3] С. М. Львовский. *Набор и верстка в пакете LaTeX.* МЦНМО, 2014.

[4] M. Trott. *The Mathematica GuideBook for Symbolics.* Springer, 2006.