

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

*Факультет физико-математических и естественных наук
Институт физических исследований и технологий
Направление «Физика»*

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Обработка результатов измерений, их погрешностей
и построение научной графики

по курсу:

Базовые пакеты

Раздел:

Оформление научно-исследовательских материалов

Выполнил(а) студент(ка) группы НФЗбд-01-20

Иванов(а) Александр(а)

Москва
2022

Содержание

Цель работы	3
Оборудование	3
Теоретическая часть	3
Результаты	4
Анализ экспериментальных результатов	17
Выводы	17
Список литературы	18
Приложение	20

Цель работы: Рассмотреть некоторые возможности языка программирования \LaTeX для подготовки отчётов, научных статей и др. Привести примеры с небольшим пояснением, в том числе работа с математическими формулами, графиками, таблицами, листингом и блок-схемами.

Оборудование: Издательская система \LaTeX 2_ϵ , Gnuplot.

Теоретическая часть: \LaTeX — научная издательская система, в первую очередь предназначенная для подготовки научных публикаций, что выражается в наличии развитого аппарата для быстрого набора математических выражений, диаграмм, физических и химических формул, средств внедрения в текст графических файлов [6].

В \LaTeX очень удобно реализован процесс нумерации объектов: формул, рисунков, таблиц, литературы и т.д., основанный на использовании уникальных меток, присваиваемых пользователем соответствующим объектам. Эта особенность системы особенно ценится пользователями системы, так как добавление или удаление в тексте документа объекта, нарушающего сквозную нумерацию, например, формулы, не требует ручной корректировки нумерации по всему документу, всю рутинную работу \LaTeX выполнит автоматически в процессе компиляции окончательного файла документа.

Ещё одно существенное преимущество \LaTeX — это особая структура документа, позволяющая пользователю иметь уверенность, что при наборе текста и последующем редактировании не сойдётся форматирование: формулы, рисунки, таблицы и фрагменты текста останутся такими же и в тех местах документа. Эту особенность используют редакции серьёзных научных изданий, широко применяя \LaTeX при верстке журналов, когда к статьям, набранным разными авторами, применяется одинаковый набор стилевых инструкций, автоматически приводящий эти статьи к единому оформлению.

Как издательская система \LaTeX представляет собой набор компьютерных программ (пакетов), однако с точки зрения конечного пользователя этой системы \LaTeX — это свод правил и команд, по которым исходный текст до-

кумента преобразуется в окончательную печатную версию, в этом смысле \LaTeX — это язык программирования, которым пользователь системы должен владеть. Первую версию \LaTeX создал Лесли Лэмпорт, который в 1985 г. представил набор макросов (макрокоманд) как надстройку над системой подготовки печатной документации \TeX , разработанной Дональдом Кнутом в 1979 г. Система команд \TeX фактически была первым языком разметки гипертекста, а её исполняемая программа — «`tex.exe`», выполнявшая преобразование размеченного текста в документ, пригодный для печати, была одной из первых программ парсеров (parser). Первый перевод [5] классической книги Дональда Кнута «The \TeX book» [1] с наиболее полной информацией об этой системе, опубликован в России в 1993 г.

Язык \TeX относится к языкам программирования низкого уровня и достаточно сложен в использовании, поэтому, несмотря на то, что \TeX до сих пор общепризнанно является наиболее качественной системой подготовки научных печатных публикаций, «чистый» \TeX сейчас используется главным образом при разработке классов и пакетов для \LaTeX .

Мы аоиспользуемся этой издательской системой в редакции $\text{\LaTeX} 2_{\epsilon}$.

Результаты:

Для начала следует установить \TeX -систему, настоятельно рекомендуется поставить `texlive` (есть на трёх платформах *linux*, *windows*, *mac*, см. <https://tug.org/texlive>), представляющий собой самый полный набор `tex`-программ¹ и возможности настройки.

Исходный код представляет собой текст набранный, как упоминалось, на специальном языке, сохранённый в обычном текстовом файле с расширением «`tex`», например `default.tex`

Для того чтобы скомпилировать `tex`-файл, необходимо вызвать в консоли (терминале, командной строке) программу `pdflatex` и передать в качестве параметра имя файла (расширение указывать необязательно), также рекомен-

¹В отличие от `miktex`, **не рекомендуется** ставить последний, как например в [6]

дуются включать режим синхронизации с исходным кодом ключом `synctex`. Тогда `pdflatex --synctex=1 default`.

Любой \LaTeX овский (или \TeX овский) документ состоит из двух частей: из преамбулы (`preamble`) и тела документа (закл \ddot{y} ч \ddot{e} нного в окружении `document`). Вся стилистика, то есть правила форматирования тела документа задаются набором команд из которых и образована преамбула документа. Например для начала работы с мультязычным вводом (русский и английский), вводом математических формул и вставки графики достаточно в преамбуде подключить всего несколько пакетов:

Пример `tex`-файла \LaTeX овского документа

```
1 % преамбула документа
2 \documentclass[12pt,a4paper]{article}
3 \usepackage[left=3.0cm, right=1.5cm, top=2.0cm,
4   bottom=2.0cm]{geometry} % геометрия страницы, в т.ч. поля
5 \usepackage[utf8]{inputenc} % распознавание кодировки текста в tex-файле
6 \usepackage[T2A]{fontenc} % распознавание шрифтов
7 \usepackage[english, main=russian]{babel} % пакет поддержки орфографий
8 \usepackage{amsmath,amsfonts,amssymb} % пакеты ввода матформул
9 \usepackage{graphicx} % пакеты поддержки вставки рисунков
10
11 \begin{document}
12   % тело документа
13   {\selectlanguage{russian}
14     Здесь может быть ваш собственный текст.
15   } % END \selectlanguage
16
17   {\selectlanguage{english}
18     Your own text can be here.
19   } % END \selectlanguage
```

```
\end{document}
```

Преамбула всегда начинается командой:

```
\documentclass [<параметры_класса_документа>] {<имя_класса_документа>}
```

однако рекомендуется использовать преамбулу в исходном tex-файле данного документа, при этом подключаемый файл `define.tex` командой `include: \include{define};` необязателен.

Отметим, что фигурные скобки `{ }` у команд отвечают за обязательные параметры, квадратные `[]` — за необязательные. Например, любое окружение в \LaTeX вводится командами `begin` и `end`: `\begin{<имя_окружения>}` ... `\end{<имя_окружения>}`. Фигурные скобки используются также для выделения объекта, например, для указания области действия некоторых команд; допустим, используя команду `bfseries` или `itshape` чтобы изменить начертание текста на **полужирное** или *курсивное* соответственно, впишем их внутри фигурных скобок, тогда действие этих команд будет ограничено в пределах данных скобок: `{\bfseries полужирный текст}, {\itshape курсивный текст}`. Важно помнить, что язык \TeX — регистрозависимый.

Подробнее введение в \LaTeX см. в [6].

Допустим, мы хотим ввести уравнения Максвелла в переменных Майорана

$\xi = E + iH$ и $\eta = E - iH$ для электромагнитного поля вне источников. Обратим внимание что строчные формулы для переменных ξ и η вводятся в *математической моде*, для входа в которую используется символ `$` (доллара). Для выхода из неё, возвращаясь в *текстовую моду*, повторно вводится символ `$`. Уравнения же введём в виде «формулы-выключки» (на выделенной строке), используя окружение `equation`:

Источник:

	Пример окружения <code>equation</code>
1	<code>\begin{equation}</code>
2	<code>\label{eq:uravneniya_maksvella_v_premennyyh_majorana}</code>

```

3 \begin{cases}
4 \frac{\partial}{\partial t} \xi = (s, \nabla) \xi, & \operatorname{div} \xi = 0,
5 \frac{\partial}{\partial t} \eta = (s, \nabla) \eta, & \operatorname{div} \eta = 0,
6 \end{cases}
7 \end{equation}

```

Вывод:

$$\begin{cases} \frac{\partial}{\partial t} \xi = (s, \nabla) \xi, & \operatorname{div} \xi = 0, \\ \frac{\partial}{\partial t} \eta = (s, \nabla) \eta, & \operatorname{div} \eta = 0, \end{cases} \quad (1)$$

где s - оператор спина фотона (см. [2]), представляющий собой вектор-матрицу со слагающими $s_j = iA_j \forall j = \overline{1, 3}$. Здесь A_j — матрицы бесконечно малых поворотов вокруг соответствующих осей трёхмерного евклидового пространства [4].

Нумерация формул автоматическая, и согласно установке в преамбуле, нумеруются те формулы на которые есть хотя бы одна ссылка в тексте. Прежде нам необходимо установить *уникальную метку* через команду `label` в качестве её аргумента. Далее чтобы сослаться на метку достаточно воспользоваться командой `ref`, но поскольку мы хотим сослаться на формулу, воспользуемся специально отведённой для этого командой `eqref`. Для уравнений выше мы напишем:

`\eqref{eq:uravneniya_maksvella_v_peremennyx_majorana}`,

где "eq" обозначает то что мы ссылаемся на объект окружения `equation`, затем следует имя метки после ":". Рекомендуется следовать этому регламенту. Разница между командами `ref` и `eqref` состоит в том что последняя выводит метку (номер) в круглых скобках. Так отметим что выражения справа в (1) представляют собой специальные условия на дивергенцию решений уравнений Максвелла.

Отметим что «формулы-выключки» можно также ввести заключив между `"\[` и `"\]` но нумеровать такие формулы невозможно с помощью команды `label`, при этом в \LaTeX сохраняется \TeX овский вариант ввода таких формул, а именно заключив описание формулы между `"$$` и `"$$`.

Нам необходимо ввести и оформить таблицу. Для этого можно подключать совокупность пакетов (см. преамбулу данного документа) по необходимости, однако мы здесь рассмотрим один мощный пакет `tabularray`, например, мы хотим оформить большую таблицу на несколько страниц следующим образом:

Источник:

Пример окружения `longtblr`

```

1 \begin{longtblr}[
2 %   theme = fancy,
3   caption = {Большая таблица, но на самом деле короткая.},
4   entry = {Большая короткая таблица},
5   label = {tblr:bolshaya_tablica_no_na_samom_dele_korotkaya},
6   note{a} = {Это первая сноска.},
7   note{${\dag}} = {Это вторая длинная сноска.},
8   remark{Примечание} = {Некоторые общие примечания.},
9   remark{Источник} = {Сделано силами авторов.},
10 ]{
11   colspec = {l|l|l|},
12   row{odd} = {bg = azure8},
13   row{1} = {c, bg = azure3, fg = white, font = \sffamily},
14   rowhead = 1,
15   rowfoot = 0,
16 }
17 \hline
18 ${\alpha}$ & ${\beta}$ & ${\gamma}$ \\
19 \hline

```


20	δ & ϵ & ζ \\\
21	\hline
22	η & θ & ι \\\
23	\hline
24	κ & λ & μ \\\
25	\hline
26	ν ξ \omicron & π ρ σ & τ υ ϕ \\\
27	\hline
28	\end{longtblr}

Вывод:

Таблица 1: Большая
таблица,
но на
самом
деле ко-
роткая.

α	β	γ
δ	ϵ	ζ
η	θ	ι
κ	λ	μ
$\nu \xi o$	$\pi \rho \sigma$	$\tau \upsilon \phi$

^a Это первая сноска.

[†] Это вторая длинная
сноска.

Примечание: Неко-
торые общие при-
мечания.

Источник: Сделано
силами авторов.

Здесь мы воспользовались окружением `longtblr`, если же наша таблица помещается на одной странице, то достаточно использовать окружение `tblr`. Заметьте, что метку `label` мы указали в необязательных параметрах (т.е. в `[]` скобках), а не в теле окружения, как это сделано в окружении `equation`. Пример результата вывода многостраничной таблицы см. в прил.

Для листинга программы воспользуемся окружением `verbatim`, например подключив пакет `fancyvrb`. В необязательных параметрах окружения `Verbatim`

выберем шрифт `courier`, размер табуляции равным трём, а номера строк расположим слева:

```
\begin{Verbatim}[fontfamily = courier, tabsize = 3, numbers = left, go
...
\end{Verbatim}%
```

В теле окружения прописываем текст исходного кода программы, кроме того можно непосредственно подключать сам файл исходного кода, воспользовавшись командой `VerbatimInput`, например: `\VerbatimInput[fontfamily=courier, numbers=left, numbersep=6pt]{src/testx3p.f08}`. Данный пакет хоть и прост в освоении, всё же довольно грубо приводит листинг. Для этих целей лучше подходит пакет `listings`, более специализированный. Им и воспользуемся для листинга программы далее.

Рассмотрим вставку готового рисунка. Для этого воспользуемся командой `includegraphics` из пакета `graphicx`, но поскольку нам необходимо ещё подписать к нему и возможность сослаться, то ещё воспользуемся окружением `figure`.

```
1 \begin{figure}[H]
2   \center{\includegraphics[width=0.7\linewidth]{c96e01c894dab21d52bb0bd6565f033
3   \caption{Восстановленная фотография тени чёрной дыры.}
4   \label{fig:vosstanovlennaya_fotografiya_chyornoj_dyry}
5 \end{figure}
```

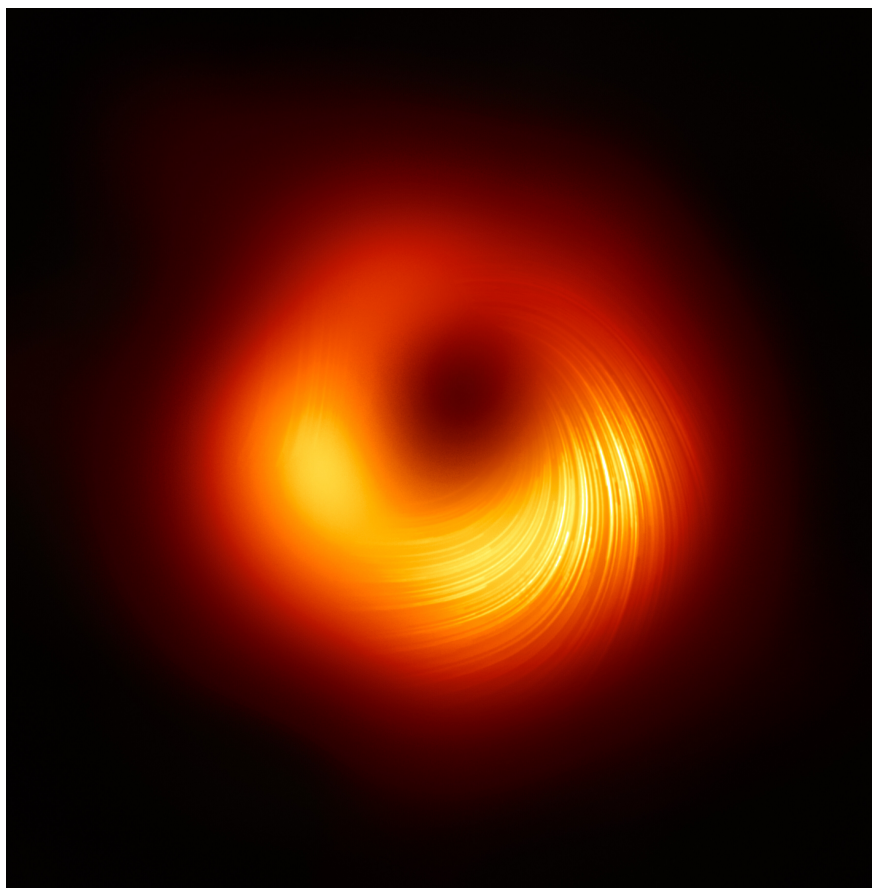


Рис. 1: Восстановленная фотография тени чёрной дыры.

Обратим внимание на параметр «[H]» окружения `figure`. Этот параметр определяет место рисунка в тексте: разрешить алгоритмам TeXa принять решение исходя из заполненности страницы [h] "хотелось бы картинку здесь"; настойчиво просить разместить после текста [h!] "очень хочу картинку здесь"; и ударить кулаком по столу — картинку тут и точка [H] "ХОЧУ картинку здесь и баста"; а с прибавлением буквы "p" мы заставляем поместить LaTeX картинку отдельно на страницу так [pH] — см. https://mydebianblog.blogspot.com/2008/12/latex_15.html.

Теперь попытаемся построить рисунок (графики) с помощью `gnuplot` и одноимённого с ним окружения из пакета `gnuplottex`.

```

1  \begin{figure}[h]%
2    \centering%
3    \begin{gnuplot}[terminal = epslatex, terminaloptions = {color dashed}]
4      set key box top left
5      set key width 4

```

```

6      set sample 1000
7      set xr [-5:5]
8      set yr [-1:1]
9      set xlabel '$x$-label'
10     set ylabel '$y$-label'
11     plot sin(x) w l lc 1 t '$\sin(x)$',\
12         cos(x) w l lc 2 t '$\cos(x)$',\
13         tan(x) w l lc 3 t '$\tan(x)$',\
14         tanh(x) w l lc 4 t '$5\tanh(x)$'
15     \end{gnuplot}
16     \caption{This is a simple example using the latex-terminal.}%
17     \label{fig:latex}%
18     \end{figure}

```

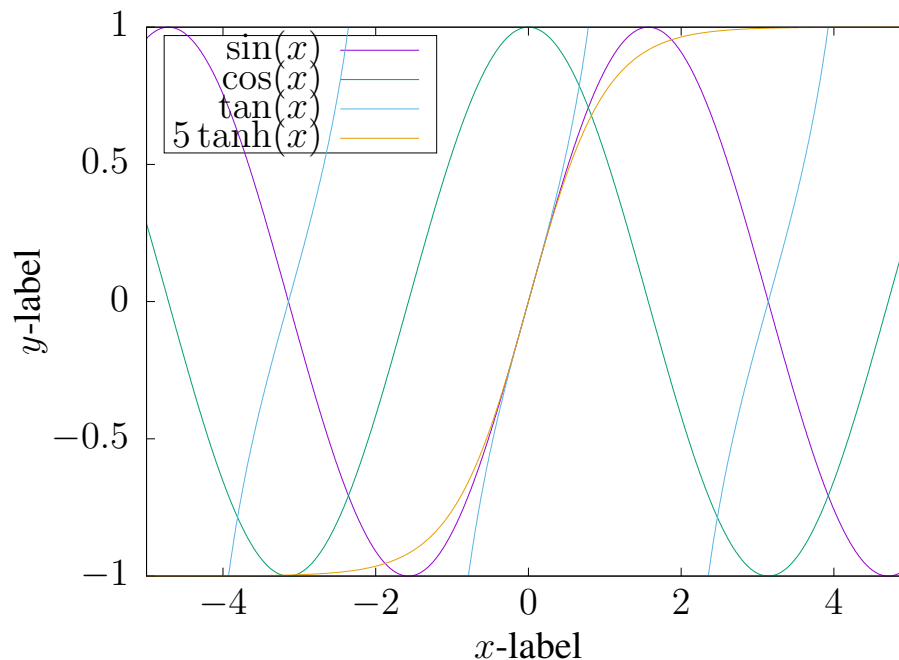


Рис. 2: Пример работы gnuplot.

Для автоматического построения графика средствами gnuplot необходимо также добавить ключ `shell escape` в команде компиляции tex-файла: `pdflatex --synctex=1 --shell-escape default`.

Рассмотрим пример построения блок-схем с использованием пакета `tikz` с подключёнными библиотеками `arrows` и `shapes` перечислив их в `{ }`-скобках команды `usetikzlibrary`, для построения графов же — `graphs`. Описание блок-схем помещаются в тело окружения `tikzpicture`: узлы вводятся и описываются

командой `node`, а пути - командой `path`. Пусть нам нужно описать алгоритм программы `testx3p`, написанной на фортране (см. директорию *app*):

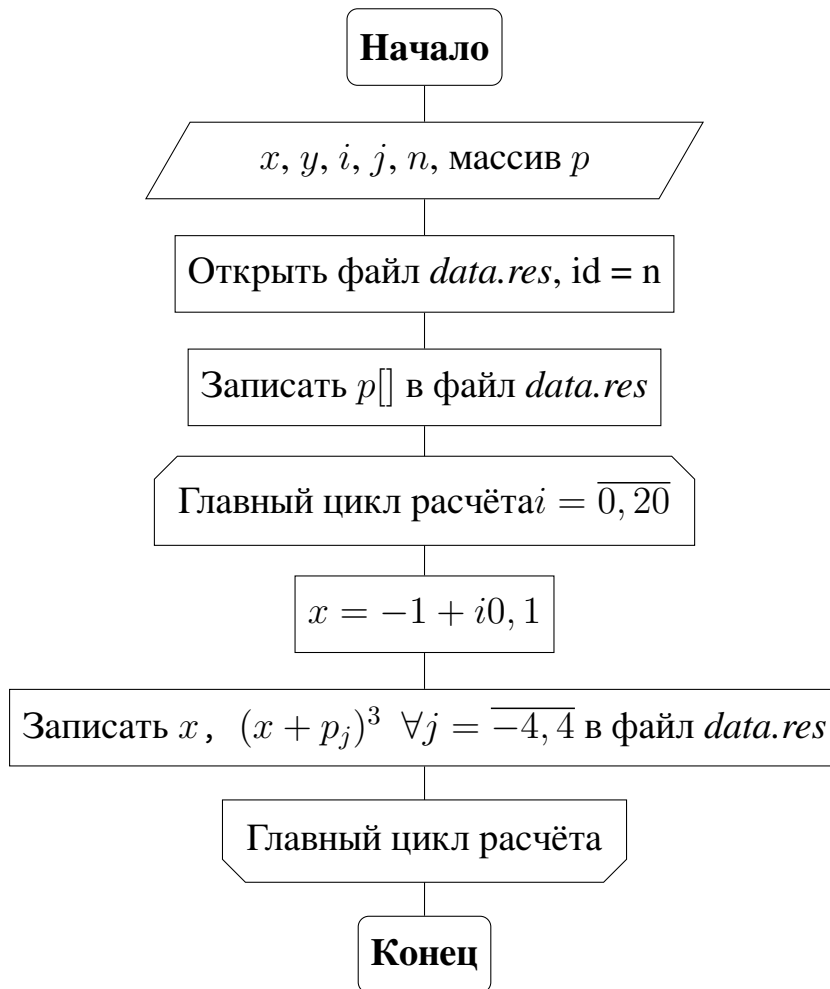
```
1 program testx3p
2   implicit none
3
4   real x, y, p(-4:4) /-0.9,-0.5,-0.3,-0.1, 0.0, 0.1, 0.3, 0.5, 0.9/
5   integer i, j, nres / 10 /
6
7   open (unit=nres,file='data/testx3p.res')
8   write(nres,1000) (p(j),j=-4,4)
9   do i=0,20
10      x=-1+i*0.1; write(nres,1001) x,((x+p(j))*3,j=-4,4)
11   enddo
12   1000 format(1x,'# p ', 9f7.2,/1x,'# x ')
13   1001 format(10f7.2)
14 end
```

Для удобства опишем несколько *tikz*-стилей согласно **ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»** — см. [3], с помощью команды `tikzset`:

`\tikzset{<имя_стиля>/ .style = {<параметры_стиля>}}`.

Используемые стили см. в исходном `tex`-файле настоящей статьи.

Таким образом,



Для случая блок-схемы вызова внешней подпрограммы (см. <https://pro-prof.com/archives/1462>) мы используем следующий стиль:

```

1 \tikzset{subroutine/.style = {% блок вызова внешней подпрограммы
2     rectangle split, rectangle split horizontal,
3     rectangle split parts = 3,
4     draw, text centered,
5     minimum width = 5em,
6     minimum height = 2em,
7     outer sep = 0
8 }
9 }%
10 \begin{tikzpicture}
11 \node [subroutine] at (0,0) (subroutine) {\nodepart{two}\textbf{Подпрограмма}}
  
```

12 \end{tikzpicture}%

	Подпрограмма	
--	---------------------	--

Обратите внимание, что для корректной работы перед содержимым блока в node необходимо прописать \nodepart{two}.

Описание tikz-стилей (да и других) рекомендуется сохранять в отдельном tex-файле и подключать его в преамбуле, или в начале тело документа с помощью команды input.

Построим график из данных расчёта программы testx3p.

Результаты расчёта записываются программой в файл *testx3p.res* в поддиректорию *data*. После небольшой обработки содержимого файла под нужды пакета tabularray и записи результата в файл *tbl1.tex* в поддиректорию *tables* мы можем представить эти данные в табличной виде:

с

Таблица 2: Результаты расчёта программы testx3p.

$\rho \backslash x$	-0,90	-0,50	-0,30	-0,10	0,00	0,10	0,30	0,50	0,90
-1,00	-6,86	-3,38	-2,20	-1,33	-1,00	-0,73	-0,34	-0,12	-0,00
-0,90	-5,83	-2,74	-1,73	-1,00	-0,73	-0,51	-0,22	-0,06	0,00
-0,80	-4,91	-2,20	-1,33	-0,73	-0,51	-0,34	-0,12	-0,03	0,00
-0,70	-4,10	-1,73	-1,00	-0,51	-0,34	-0,22	-0,06	-0,01	0,01
-0,60	-3,38	-1,33	-0,73	-0,34	-0,22	-0,12	-0,03	-0,00	0,03
-0,50	-2,74	-1,00	-0,51	-0,22	-0,12	-0,06	-0,01	0,00	0,06
-0,40	-2,20	-0,73	-0,34	-0,12	-0,06	-0,03	-0,00	0,00	0,12
-0,30	-1,73	-0,51	-0,22	-0,06	-0,03	-0,01	0,00	0,01	0,22
-0,20	-1,33	-0,34	-0,12	-0,03	-0,01	-0,00	0,00	0,03	0,34

Продолжение на следующей странице

Таблица 2: Результаты расчёта программы `testx3p`. (Продолжение)

$\begin{matrix} \backslash & p \\ x \end{matrix}$	-0,90	-0,50	-0,30	-0,10	0,00	0,10	0,30	0,50	0,90
-0,10	-1,00	-0,22	-0,06	-0,01	-0,00	0,00	0,01	0,06	0,51
0,00	-0,73	-0,12	-0,03	-0,00	0,00	0,00	0,03	0,12	0,73
0,10	-0,51	-0,06	-0,01	0,00	0,00	0,01	0,06	0,22	1,00
0,20	-0,34	-0,03	-0,00	0,00	0,01	0,03	0,13	0,34	1,33
0,30	-0,22	-0,01	0,00	0,01	0,03	0,06	0,22	0,51	1,73
0,40	-0,12	-0,00	0,00	0,03	0,06	0,12	0,34	0,73	2,20
0,50	-0,06	0,00	0,01	0,06	0,12	0,22	0,51	1,00	2,74
0,60	-0,03	0,00	0,03	0,12	0,22	0,34	0,73	1,33	3,38
0,70	-0,01	0,01	0,06	0,22	0,34	0,51	1,00	1,73	4,10
0,80	-0,00	0,03	0,13	0,34	0,51	0,73	1,33	2,20	4,91
0,90	0,00	0,06	0,22	0,51	0,73	1,00	1,73	2,74	5,83
1,00	0,00	0,12	0,34	0,73	1,00	1,33	2,20	3,38	6,86

Путь к исходному файлу с данными *testx3p.res* мы пропишем в сам скрипт `gnuplot` для построения графиков, который сохраним в отдельном файле, а затем подключим его в `TEX` документ с помощью команды `gnuplotloadfile`, например

`\gnuplotloadfile[<параметры>]{scr/testx3p1.gp}`. Тогда

Подробнее этот пример программы на *фортране* и взаимодействия с `gnuplot` см. [7].

Анализ экспериментальных результатов: Некоторый анализ. . .

Выводы: В заключении отметим, что данное руководство в большей степени пример; для полного введения лучше обратиться к справочникам и к указанной литературе. По-началу потребуется много практики для освоения

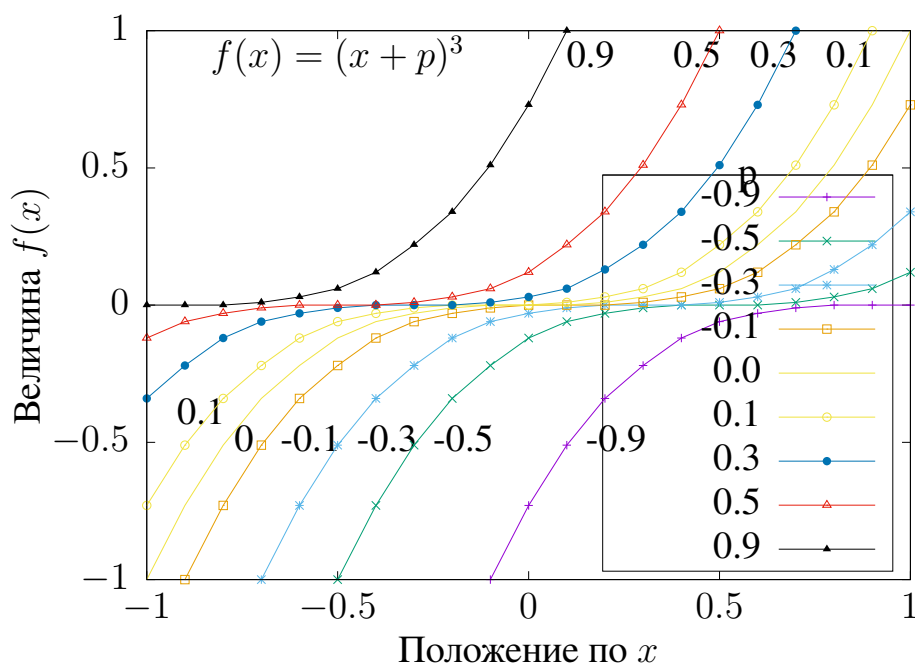


Рис. 3: Семейство кривых функции $f(x) = (x + p)^3$.

языка программирования \LaTeX . Вместе с тем `gnuplot` был выбран из-за простоты в его освоении, и хорошей связки с \LaTeX . Тем не менее, можно строить графики средствами самого \LaTeX а, а именно при помощи того же пакета `tikz`, использованного для построения блок-схем. Однако рекомендуется сначала освоиться с базовым \LaTeX ом.

Список литературы:

Список литературы

- [1] Knuth D. E. The \TeX book. — Reading, Massachusetts : Addison-Wesley Publishing Company, 1984.
- [2] Ахиезер , Берестецкий Квантовая электродинамика / Под ред. Л. П. Рукавой. — 4-е изд. — М. : Наука, 1981.
- [3] ГОСТ 19.701–90 (ИСО 5807–85) «Единая система программной документации» : Сб. ГОСТов. : Отчет : МКС 01.080.50 / Стандартинформ ; исполн.: А. А. Мкртумян, А. Л. Щерс, А. Н. Сироткин и др. — М. : 2010. — янв.

- [4] Гельфанд , Минлос , Шапиро Представление группы вращений и группы Лоренца, их применения. — М. : Физматгиз, 1972.
- [5] Кнут Все про \TeX . — Протвино : R \TeX , 1993.
- [6] Насыров Пакеты прикладных программ для физиков: \LaTeX : учебное пособие. — Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2019.
- [7] Шнейвайс Азы GNUPLOTa. — СПб. : Изд-во СПбГУ, 2016.

Приложение:

Таблица 3: Результаты моделирования
движения частицы.

Время t , с	Координаты		Скорость	
	x , м	y , м	v_x , м/с	v_y , м/с
0,001	0,005	-0,005	4,781	-5,486
0,002	0,005	-0,008	0,159	-1,155
0,003	-0,019	-0,012	-8,105	-1,505
0,004	-0,005	-0,022	3,658	-2,444
0,005	0,021	-0,006	5,137	3,219
0,006	-0,026	-0,029	-7,844	-3,795
0,007	-0,054	-0,094	-3,950	-9,375
0,008	-0,084	-0,023	-3,840	8,925
0,009	-0,171	-0,059	-9,670	-4,042
0,010	-0,195	-0,076	-2,376	-1,628
0,011	-0,151	-0,145	3,968	-6,328
0,012	-0,259	-0,262	-8,983	-9,745
0,013	-0,299	-0,256	-3,022	0,456
0,014	-0,256	-0,168	3,037	6,276
0,015	-0,221	-0,047	2,361	8,065
0,016	-0,367	0,102	-9,124	9,339
0,017	-0,277	0,148	5,282	2,678
0,018	-0,369	0,216	-5,116	3,819
0,019	-0,452	0,328	-4,353	5,891

Продолжение на следующей странице

Таблица 3: Результаты моделирования
движения частицы. (Продол-
жение)

Время t , с	Координаты		Скорость	
	x , м	y , м	v_x , м/с	v_y , м/с
0,020	-0,402	0,236	2,464	-4,593
0,021	-0,291	0,333	5,320	4,596
0,022	-0,481	0,414	-8,654	3,707
0,023	-0,405	0,547	3,323	5,783
0,024	-0,551	0,628	-6,082	3,369
0,025	-0,702	0,402	-6,048	-9,053
0,026	-0,508	0,604	7,459	7,754
0,027	-0,715	0,685	-7,672	3,029
0,028	-0,560	0,545	5,551	-5,028
0,029	-0,558	0,761	0,042	7,452
0,030	-0,537	0,671	0,721	-3,002
0,031	-0,368	0,882	5,433	6,835
0,032	-0,105	0,996	8,241	3,535
0,033	-0,118	1,232	-0,413	7,149
0,034	0,174	0,920	8,581	-9,163
0,035	0,109	1,206	-1,847	8,186
0,036	0,230	0,932	3,351	-7,629
0,037	-0,092	1,175	-8,694	6,571
0,038	-0,238	1,057	-3,849	-3,101

Продолжение на следующей странице

Таблица 3: Результаты моделирования
движения частицы. (Продол-
жение)

Время t , с	Координаты		Скорость	
	x , м	y , м	v_x , м/с	v_y , м/с
0,039	-0,004	0,704	6,004	-9,059
0,040	0,391	0,822	9,889	2,955
0,041	0,420	1,161	0,691	8,269
0,042	0,217	1,309	-4,816	3,521
0,043	0,197	1,395	-0,473	1,998
0,044	0,599	1,251	9,142	-3,277
0,045	0,629	1,294	0,649	0,958
0,046	0,752	1,626	2,675	7,230
0,047	0,930	1,768	3,804	3,010
0,048	1,389	1,777	9,552	0,199
0,049	1,834	1,328	9,091	-9,161
0,050	1,359	1,336	-9,511	0,157
0,051	1,674	0,829	6,186	-9,941
0,052	2,185	0,794	9,827	-0,688
0,053	2,285	1,222	1,889	8,089
0,054	1,761	1,747	-9,706	9,712
0,055	1,347	1,848	-7,537	1,834
0,056	1,190	1,565	-2,802	-5,048
0,057	1,172	1,296	-0,321	-4,714

Продолжение на следующей странице

Таблица 3: Результаты моделирования
движения частицы. (Продол-
жение)

Время t , с	Координаты		Скорость	
	x , м	y , м	v_x , м/с	v_y , м/с
0,058	0,744	1,259	-7,367	-0,637
0,059	0,574	0,779	-2,881	-8,135
0,060	0,739	1,189	2,754	6,826
0,061	0,555	0,881	-3,025	-5,044
0,062	0,831	0,595	4,449	-4,614
0,063	1,001	1,129	2,700	8,478
0,064	0,547	0,571	-7,100	-8,719
0,065	0,672	0,279	1,937	-4,500
0,066	1,282	0,349	9,238	1,069
0,067	1,483	0,524	2,996	2,603
0,068	2,003	0,485	7,651	-0,570
0,069	2,012	0,183	0,134	-4,371
0,070	1,467	-0,431	-7,790	-8,780
0,071	2,057	-0,356	8,309	1,062
0,072	1,405	-0,364	-9,054	-0,108
0,073	1,387	-0,054	-0,254	4,241
0,074	1,347	-0,573	-0,539	-7,014
0,075	1,403	-0,564	0,755	0,120
0,076	1,505	-0,277	1,339	3,778

Продолжение на следующей странице

Таблица 3: Результаты моделирования
движения частицы. (Продол-
жение)

Время t , с	Координаты		Скорость	
	x , м	y , м	v_x , м/с	v_y , м/с
0,077	0,799	-0,228	-9,174	0,635
0,078	0,977	0,215	2,284	5,686
0,079	0,552	-0,551	-5,381	-9,707
0,080	1,084	-0,627	6,650	-0,947
0,081	1,190	-0,875	1,311	-3,062
0,082	0,819	-0,950	-4,525	-0,906
0,083	0,657	-0,552	-1,955	4,785
0,084	0,827	0,199	2,030	8,948
0,085	0,845	-0,404	0,208	-7,098
0,086	0,204	-0,189	-7,451	2,503
0,087	0,887	0,142	7,845	3,805
0,088	1,658	-0,062	8,770	-2,319
0,089	1,164	-0,849	-5,557	-8,840
0,090	0,590	-0,495	-6,380	3,931
0,091	1,179	-1,093	6,475	-6,576
0,092	0,547	-0,549	-6,866	5,914
0,093	0,098	-0,362	-4,832	2,011
0,094	0,215	-0,965	1,248	-6,418
0,095	-0,177	-1,288	-4,131	-3,391

Продолжение на следующей странице

Таблица 3: Результаты моделирования
движения частицы. (Продол-
жение)

Время t , с	Координаты		Скорость	
	x , м	y , м	v_x , м/с	v_y , м/с
0,096	0,020	-2,069	2,060	-8,145
0,097	0,665	-2,317	6,646	-2,553
0,098	0,296	-2,182	-3,766	1,376
0,099	0,609	-2,790	3,166	-6,140
0,100	1,329	-3,064	7,193	-2,740