

# Краткое введение в $\text{\LaTeX}$ и основы работы с Gnuplot в примерах

Николаев Н. Э.<sup>1,2)</sup>, Гоним Н.<sup>1,3)</sup>

<sup>1)</sup>Институт физических исследований и технологий  
Российский университет дружбы народов  
Россия, 117198, Москва, ул. Миклухо-Маклая, 6

<sup>2)</sup>`<nikolaev_ne@pfur.ru>`, <sup>3)</sup>`<gonim_m@pfur.ru>`

Лекция

Май  
2022

## Аннотация

Рассматриваются некоторые возможности языка программирования  $\text{\LaTeX}$  для подготовки отчётов, научных статей и др. Приводятся примеры с небольшим пояснением, в том числе работа с математическими формулами, графиками, таблицами, листингом и блок-схемами.

**Ключевые слова:** издательская система, язык программирования, построение графиков, построение таблиц, построение блок-схем,  $\text{\LaTeX}$ , листинг, оформление математических выражений.

# Содержание

- 1 Введение
- 2 Начало работы
- 3 Ввод формул и их нумерация
- 4 Работа с таблицами и листингом
- 5 Работа с графикой и блок-схемами
- 6 Заключение

# Введение

$\text{\LaTeX}$  — 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]. Особо отметим:

- ▽ автонумерация объектов (формул, рисунков и т.д.) на основе использования пользовательских уникальных меток;
- ▽ структура документа разбита на преамбулу — набор правил форматирования, и тело документа — его содержание.

## $\text{\LaTeX}$ — это язык программирования

Как издательская система  $\text{\LaTeX}$  представляет собой набор компьютерных программ (пакетов), однако с точки зрения конечного пользователя этой системы  $\text{\LaTeX}$  — это свод правил и команд, по которым исходный текст документа преобразуется в окончательную печатную версию.

# История

- система команд  $\text{T}_{\text{E}}\text{X}$  (Д. Кнут, 1979) — система подготовки печатной документации, — фактически был первым языком разметки гипертекста;
- исполняемая программа — «tex», выполняющая преобразование размеченного текста в документ, пригодный для печати — одна из первых программ парсеров (parser);
- Первая версия  $\text{\LaTeX}$  (Л. Лэмпорт, 1985) — набор макросов (макрокоманд) — надстройка над  $\text{T}_{\text{E}}\text{X}$ ;
- Первый перевод [5] классической книги Дональда Кнута «The  $\text{T}_{\text{E}}\text{X}$  book» [1] на русский — 1993 г.

## $\text{T}_{\text{E}}\text{X}$ — ассемблер систем подготовки печатной документации

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

# Строение $\text{\LaTeX}$ -документа

Для начала необходимо установить соответствующее программное обеспечение —  $\text{\TeX}$ -систему, настоятельно рекомендуется *texlive* (*linux*, *windows* или *mac*, см. <https://tug.org/texlive>). Уточним некоторые моменты:

- исходный код представляет собой текст набранный на специальном языке, сохранённый в обычном текстовом файле с расширением «*tex*», например *default.tex*;
- для компиляции *tex*-файла достаточно вызвать в консоли (терминале или командной строке) одну из программ *tex*, *latexmk*, *pdflatex* и др., и передать в качестве параметра имя файла (расширение указывать необязательно).
- по умолчанию  $\text{\LaTeX}$  (и  $\text{\TeX}$ ) компилируют исходный *tex*-файл в собственный формат «*dvi*» (программы просмотра, например, «*xdvi*» или «*dviout*» в *windows*), и только затем, например, с помощью программы *dvipdf* формируется *pdf*-файл;
- для компиляции по умолчанию сразу в «*pdf*» можно использовать программу *pdflatex*;
- рекомендуется включать режим синхронизации с исходным кодом ключом *src-specials* или *synctex* (в случае «*pdf*»);
- **программа *latexmk* не работает с растровой графикой (*png*, *jpg* и тд.), используйте *pdflatex*.**

# Строение $\text{[A]T}_\text{E}^X$ -документа

## Примеры консольных команд

- > `latexmk -bibtex -dvi -e '$latex=q/latex --src-specials %O --shell-escape %S/'`  $\leftrightarrow$   
default % — выдаёт «dvi»;
- > `latexmk -bibtex -dvi -pdf -e '$latex=q/latex --src-specials %O --synctex=1`  $\leftrightarrow$   
`--shell-escape %S/'` default % — выдаёт «pdf», сохраняя «dvi»;
- > `pdflatex -bibtex -output-format dvi --src-specials --shell-escape default %` —  
выдаёт «dvi».
- > `pdflatex -bibtex -output-format pdf --synctex=1 --shell-escape default %` — выдаёт  
«pdf» (по умолчанию: указывать формат вывода явно необязательно).

Ключ `shell-escape` в команде компиляции `tex`-файла необходим для автоматического построения графика средствами `gnuplot`, к которому вернёмся при рассмотрении работы с графикой.

# Строение $\text{\LaTeX}$ -документа

Любой  $\text{\LaTeX}$ овский (или  $\text{\TeX}$ овский) документ состоит из двух частей: из преамбулы (preamble) и тела документа (заключённого в окружении document). Правила форматирования тела документа задаются набором команд из которых и образована преамбула документа.

## Листинг 1: Пример tex-файла $\text{\LaTeX}$ 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 тч.. поля
5  usepackage[utf8]{inputenc} % распознавание кодировки текста в texфайле–
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
20 end{document}
21
```



# Строение $\text{\LaTeX}$ -документа

- Преамбула всегда начинается командой:  
`\documentclass[<параметры_класса_документа>]{<имя_класса_документа>}`.
- Рекомендуется использовать преамбулу в исходном `tex`-файле документа-статьи, при этом подключаемый файл `define.tex` командой `INCLUDE: \include{define}`; необязателен.
- Фигурные скобки `{ }` у команд отвечают за обязательные параметры, а квадратные `[ ]` — за необязательные. Например, любое окружение в  $\text{\LaTeX}$  вводится командами `BEGIN` и `END`: `\begin{<имя_окружения>} ... \end{<имя_окружения>}`.
- Фигурные скобки используются также для выделения объекта, например, для указанные области действия некоторых команд; допустим чтобы изменить начертание текста на **полужирное** или *курсивное* соответственно, впишем их внутри фигурных скобок: `{\bfseries полужирный текст}`, `{\itshape курсивный текст}`.
- **Язык  $\text{\TeX}$  — регистрозависимый.**
- Подробнее введение в  $\text{\LaTeX}$  см. в [6].

# Ввод формул и их нумерация

- ▽ Ввод текста осуществляется в *текстовой моде*, а математических формул — в *математической моде*.
- ▽ Строчные формулы заключаются между символами  $\$$  (доллара).
- ▽ «Формулы-выключки» вводятся в окружении `EQUATION`.
- ▽ Нумерация формул автоматическая, и согласно установке в преамбуле, нумеруются те формулы на которые есть хотя бы одна ссылка в тексте.
- ▽ Для нумерации необходимо установить *уникальную метку* через команду `LABEL` в качестве её аргумента.
- ▽ Чтобы сослаться на метку достаточно воспользоваться командой `REF`, для уравнений — командой `EQREF`.

Отметим что формулы-выключки можно также ввести заключив между “[” и “]”, но нумеровать такие формулы невозможно с помощью команды `LABEL`, при этом в  $\text{\LaTeX}$  сохраняется  $\text{\TeX}$ овский вариант ввода таких формул, а именно заключив описание формулы между “ $\$$ ” и “ $\$$ ”.

# Уравнения Максвелла в переменных Майорана

Допустим, мы хотим ввести уравнения Максвелла в переменных Майорана  $\xi = E + iH$  и  $\eta = E - iH$  для электромагнитного поля вне источников. Введём в виде «формулы-выключки» (на выделенной строке), используя окружение `EQUATION`:

Листинг 2: Пример окружения `EQUATION`

```

1 begin{equation}
2   \label{eq:uravneniya_maksvellya_v_peremennyx_majorana}
3   \hfill
4   \begin{cases}
5     \frac{\partial}{\partial t} \xi + (s, \nabla) \xi = 0, \ \& \operatorname{div} \xi = 0, \\
6     \frac{\partial}{\partial t} \eta - (s, \nabla) \eta = 0, \ \& \operatorname{div} \eta = 0, \\
7   \end{cases}
8   \hfill
9 end{equation}
10
```

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

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 евклидова пространства [4].

# Работа с таблицами и листингом

Для оформления таблиц можно подключать совокупность пакетов (см. преамбулу данного документа) по необходимости, однако здесь рассмотрим один мощный пакет `TABULARARRAY`, например, мы хотим оформить большую таблицу на несколько страниц. Здесь мы воспользуемся окружением `LONGTBRLR`, если же наша таблица помещается на одной странице, то достаточно использовать окружение `TBLR`.

Для листинга программы можно подключить пакет `FANCYVRB`. Но для этих целей лучше подходит специализированный пакет `LISTINGS`. В теле соответствующего окружения прописывается текст исходного кода программы, кроме того можно непосредственно подключать сам файл исходного кода.

Таблица 1: Большая таблица, но на самом деле короткая.

$\alpha$	$\beta$	$\gamma$
$\delta$	$\epsilon$	$\zeta$
$\eta$	$\theta$	$\iota$
$\kappa$	$\lambda$	$\mu$
$\nu \xi o$	$\pi \rho \sigma$	$\tau \upsilon \phi$

<sup>a</sup> Это первая сноска.

<sup>†</sup> Это вторая длинная сноска.

*Примечание:* Некоторые общие примечания.

*Источник:* Сделано силами авторов.

# Работа с таблицами

## Листинг 3: Пример окружения LONGTBRL

```

1  begin{longtblr}[caption = Большая{ таблица, но на самом деле короткая.}, entry = Большая{ ←
    короткая таблица}, label = {tblr:bolshaya_tablica_no_na_samom_dele_korotkaya}, note{a} ←
    = Это{ первая сноска.}, note{${\dag$}} = Это{ вторая длинная сноска.}, remarkПримечание{} ←
    = Некоторые{ общие примечания.}, remarkИсточник{} = Сделано{ силами авторов.}]{hlines, ←
    colspec = {|l|l|l|l|}, row{odd} = {bg = azure8}, row{1} = {c, bg = azure3, fg = white, font = ←
    \sffamily}, rowhead = 1, rowfoot = 0}
2  $\alpha$ & $\beta$ & $\gamma$ \\
3  $\delta$ & $\epsilon$ & $\zeta$ \\
4  $\eta$ & $\theta$ & $\iota$ \\
5  $\kappa$ & $\lambda$ & $\mu$ \\
6  $\nu$ & $\xi$ & $\omicron$ & $\pi$ & $\rho$ & $\sigma$ & $\tau$ & $\upsilon$ & $\phi$ \\
7  end{longtblr}
8

```

Заметьте, что метку `label` мы указываем в необязательных параметрах (т.е. в `[ ]` скобках), а не в теле окружения, как это сделано в окружении `EQUATION`.

# Работа с графикой и блок-схемами

Рассмотрим вставку готового рисунка. Для этого воспользуемся командой `INCLUDEGRAPHICS` из пакета `GRAPHICX`, но поскольку нам необходимо ещё подпись к нему и возможность сослаться, то ещё воспользуемся окружением `FIGURE`.

## Листинг 4: Пример вставки рисунка

```
1 begin{ figure }[H]  
2   \center {\ includegraphics [width=0.7\linewidth ]{  
3     c96e01c894dab21d52bb0bd6565f0331.jpg}  
4   }  
5   \caption{ Восстановленная фотография ←  
6     тени чёрной дыры. }  
7   ←  
8   \label { fig : vosstanovlennaya _ fotografiya _ chyornoj _ dyry }  
9 end{ figure }
```

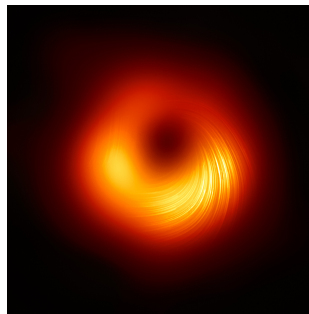


Рис. 1. Восстановленная фотография тени чёрной дыры.

Заметим параметр «[H]» окружения `FIGURE`. Этот параметр определяет место рисунка в тексте: разрешить алгоритмам `TeX` принять решение исходя из заполненности страницы.

# Работа с графикой

Данный параметр у окружения `FIGURE` может принимать следующие значения:

- `[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 "р" мы заставляем поместить `LaTeX` картинку отдельно на страницу так `[pH]` —  
см. [https://mydebianblog.blogspot.com/2008/12/latex\\_15.html](https://mydebianblog.blogspot.com/2008/12/latex_15.html).

# Работа с Gnuplott

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 size ←
      3.0in,2.25in font ',8' }]
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 }
19

```



# Пример графика в Gnuplot

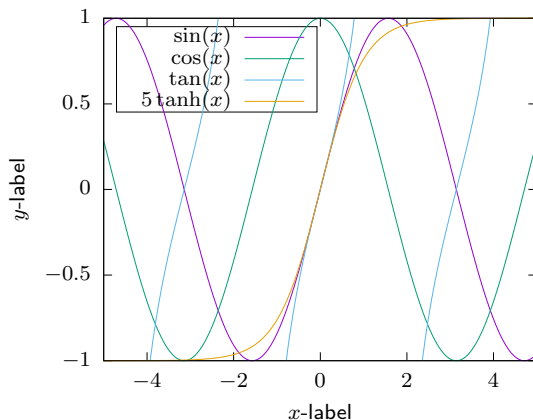


Рис. 2. Пример работы gnuplot.

## Пример программы на языке Fortran

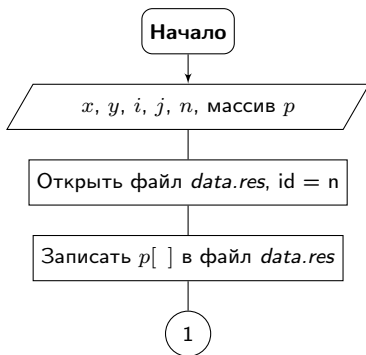
Рассмотрим пример построения блок-схем с использованием пакета `TIKZ` с подключёнными библиотеками `arrows` и `shapes` перечислив их в `{ }`-скобках команды `usetikzlibrary`, для построения графов же — `graphs`. Описание блок схем помещаются в тело окружения `TIKZPICTURE`: узлы вводятся и описываются командой `node`, а пути — командой `path`. Пусть нам нужно описать алгоритм программы `testx3p`, написанной на фортране (см. директорию `app`):

src/testx3p.f08

```
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
```

# Работа с блок-схемами

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,



# Пример стиля блок-схемы

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

## Листинг 5: Пример стиля TIKZ

```

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 }\%
13
```



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.

# Графики результата работы программы `testx3p` I

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 (см. табл. 2).

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}.
```

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

# Графики результата работы программы testx3p II

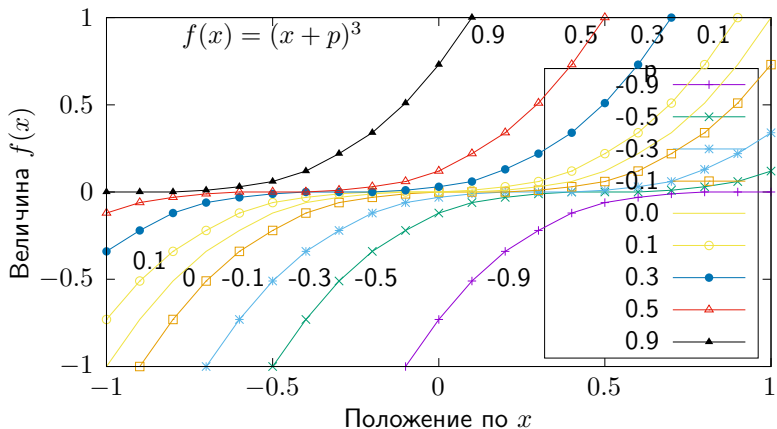


Рис. 3. Семейство кривых функции  $f(x) = (x + p)^3$ .

# Gnuplot-скрипт для графиков программы testx3p

scr/testx3p1.gp

```

1  # set terminal postscript eps color dashed #renhanced # установка типа терминала и
2  # set output 'testx3p1.eps'                # --"---"--- имени .epsфайла--;
3  set size 0.7, 0.7
4  set xlabel 'Положение по $x$'
5  set ylabel 'Величина $f(x)$'
6  set key right bottom sample 5 spacing 1.5 width 5 title "p" box
7  set label "$f(x) = (x + p)^{3}$" at -0.5,0.9 center font "Helvetica",24"
8  set label "-0.9" at 0.15,-0.5; set label "-0.5" at -0.25,-0.5
9  set label "-0.3" at -0.45,-0.5; set label "-0.1" at -0.65,-0.5
10 set label "0" at -0.77,-0.50; set label "0.1" at -0.92,-0.40
11 set label "0.1" at 0.78, 0.90; set label "0.3" at 0.58,0.90
12 set label "0.5" at 0.38, 0.90; set label "0.9" at 0.10,0.90
13 plot [-1:1] [-1:1] 'data/testx3p.res' using 1:2 title "-0.9" w lp lc 1,\
14 'data/testx3p.res' using 1:3 title "-0.5" w lp lc 2,\
15 'data/testx3p.res' using 1:4 title "-0.3" w lp lc 3,\
16 'data/testx3p.res' using 1:5 title "-0.1" w lp lc 4,\
17 'data/testx3p.res' using 1:6 title " 0.0" w l, \
18 'data/testx3p.res' using 1:7 title " 0.1" w lp lc 5,\
19 'data/testx3p.res' using 1:8 title " 0.3" w lp lc 6,\
20 'data/testx3p.res' using 1:9 title " 0.5" w lp lc 7,\
21 'data/testx3p.res' using 1:10 title " 0.9" w lp lc 8

```

# Таблица результатов работы программы testx3p

Таблица 2: Результаты расчёта программы testx3p.

$x \backslash p$	-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
-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



# Заключение

В заключении отметим, что данное руководство в большей степени пример; для полного введения лучше обратиться к справочникам и к указанной литературе. По-началу потребуются много практики для освоения языка программирования  $\text{\LaTeX}$ . Вместе с тем `gnuplot` был выбран из-за простоты в его освоении, и хорошей связки с  $\text{\LaTeX}$ . Тем не менее, можно строить графики средствами самого  $\text{\LaTeX}$ а, а именно при помощи того же пакета `tikz`, использованного для построения блок-схем. Однако рекомендуется сначала освоиться с *базовым*  $\text{\LaTeX}$ ом.

# Список литературы I

- [1] Knuth D. E. The  $\text{\TeX}$ book. —  
Reading, Massachusetts : Addison-Wesley Publishing Company, 1984.
- [2] Ахиезер А. И., Берестецкий В. Б. Квантовая электродинамика / Под ред.  
Л. П. Русаковой. —  
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 / Стандартиформ ; исполн.: А. А. Мкртумян,  
А. Л. Щерс, А. Н. Сироткин и др. —  
М. : 2010. — янв.
- [4] Гельфанд И. М., Минлос Р. А., Шапиро З. Я. Представление группы вращений и  
группы Лоренца, их применения. —  
М. : Физматгиз, 1972.
- [5] Кнут Д. Е. Все про  $\text{\TeX}$ . —  
Протвино : R $\text{\TeX}$ , 1993.

## Список литературы II

- [6] Насыров В. В. Пакеты прикладных программ для физиков:  $\text{\LaTeX}$  : учебное пособие. — Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2019.
- [7] Шнейвайс А. Б. Азы GNUPLOTa. — СПб. : Изд-во СПбГУ, 2016.