

# Создание иллюстраций в MetaPost. (из серии компьютерные TeXнологии)

© Балдин Е.М.

3 июня 2006 г.

## Аннотация

Эта статья была опубликована в 6ом февральском номере русскоязычного журнала Linux Format (<http://www.linuxformat.ru>) за 2006 год. На сайте <http://www.inp.nsk.su/~baldin/> статья размещена с разрешения редакции журнала и до августа месяца все вопросы с размещением статьи в других местах следует решать с редакцией Linux Format.

Текст, представленный здесь, не является точной копией статьи в журнале. Все вопросы по содержанию следует задавать мне по электронной почте [E.M.Baldin@inp.nsk.su](mailto:E.M.Baldin@inp.nsk.su).

## Содержание

|          |                             |           |
|----------|-----------------------------|-----------|
| <b>1</b> | <b>Введение в MetaPost</b>  | <b>3</b>  |
| 1.1      | Здравствуй мир . . . . .    | 4         |
| 1.2      | MetaPost-конвейер . . . . . | 8         |
| 1.3      | Среда разработки . . . . .  | 9         |
| 1.4      | Чуть-чуть о МЕТА . . . . .  | 11        |
| 1.5      | Литература . . . . .        | 14        |
| <b>2</b> | <b>Базовые элементы</b>     | <b>16</b> |
| 2.1      | Рисуем по точкам . . . . .  | 16        |
| 2.2      | Пути . . . . .              | 19        |
| 2.3      | Вставка текста . . . . .    | 23        |
| 2.4      | Заливка . . . . .           | 25        |
| 2.5      | Цвета . . . . .             | 27        |

|          |                                          |           |
|----------|------------------------------------------|-----------|
| <b>3</b> | <b>Начала автоматизации</b>              | <b>29</b> |
| 3.1      | Объекты picture . . . . .                | 29        |
| 3.2      | Трансформация . . . . .                  | 32        |
| 3.3      | Циклы и условные операторы . . . . .     | 35        |
| 3.4      | Макросы . . . . .                        | 38        |
| 3.5      | Стандартные функции . . . . .            | 41        |
| <b>4</b> | <b>Графики и диаграммы</b>               | <b>43</b> |
| 4.1      | Графики в MetaPost . . . . .             | 44        |
| 4.2      | Работа с файлами данных . . . . .        | 47        |
| 4.3      | Гистограммы в MetaPost . . . . .         | 50        |
| 4.4      | Круговые диаграммы . . . . .             | 52        |
| 4.5      | ИТЭХ рисует с помощью MetaPost . . . . . | 56        |
| 4.6      | gnuplot . . . . .                        | 58        |
| 4.7      | Подведём итоги . . . . .                 | 59        |
| <b>5</b> | <b>Дополнительные главы</b>              | <b>60</b> |
| 5.1      | Пакет boxes . . . . .                    | 60        |
| 5.2      | Фейнмановские диаграммы . . . . .        | 63        |
| 5.3      | Фракталы . . . . .                       | 66        |
| 5.4      | Увеличительное стекло . . . . .          | 68        |
| 5.5      | Штриховка . . . . .                      | 70        |
| 5.6      | Вставка eps . . . . .                    | 72        |
| 5.7      | Большие числа . . . . .                  | 73        |
| 5.8      | Макрос ТЭХ . . . . .                     | 75        |
| <b>6</b> | <b>Заключение</b>                        | <b>77</b> |

# 1 Введение в MetaPost

Каждая иллюстрация — это целая история.

Люди делятся на тех, кто умеет рисовать, и тех, кто не умеет. Причём вторых большинство. В этом нет ничего плохого или хорошего — такова жизнь. Хороший иллюстратор — редкость.

Компьютерные технологии дают возможность создавать высококачественные документы особенно не концентрируясь на проблеме оформления. Всю черновую работу делает за вас компьютер. Например, система  $\text{\LaTeX}$  фактически заменяет собой электронную типографию, но, к сожалению, создание иллюстраций выносится за рамки этого процесса. Умение иллюстрировать свой текст является необходимым навыком для тех, кто хочет делать свои книги самостоятельно от начала и до конца. Часто бывает что основной смысл несут в себе именно рисунки.

Создание иллюстраций это длительный и тяжёлый процесс, но если целью стоит совершенство вашей книги, то потраченное время того стоит. Обучение в изостудии в младших классах подняло мой уровень рисования с «никуда не годного» до «терпимого, если отойти на сто метров». Как художник я совершенно безнадёжен, но иллюстрации мне приходится делать довольно часто. В основном это не сложные картинки к задачам по физике. Когда мне говорят, что мои рисунки к задачам вызывают чувство зависти — мне с одной стороны приятно, но с другой стороны очень тяжело объяснить как же я этого достиг. Цель этого текста — популяризовать создание иллюстраций с помощью MetaPost.

Желание контролировать всё в процессе создания книги в случае Дональда Э. Кнута (Donald E. Knuth) привело к созданию программ  $\text{\TeX}$  и METAFONT.

Изначально METAFONT предназначался для создания шрифтов и результатом его работы был растр с изображением шрифта. Позже аспирант Д.Э. Кнута Джон Хобби (John Hobby) модифицировал METAFONT таким образом, что результатом работы программы стала картинка в формате EPS (Encapsulated PostScript).

MetaPost — это программа для тех, кто может объяснить компьютеру что он хочет. MetaPost чрезвычайно полезен для случаев, когда картинку проще описать логически, нежели образно. Даже если Вы не умеете рисовать, результат может получиться вполне приличный, потому что

«виртуальный карандаш» под управлением компьютера дрожать не будет.

MetaPost проектировался как простая программа, которую можно относительно быстро настроить под свои нужды. Он вполне обозрим и компактен. Многие «правильные» программы, работающие с векторной графикой, такие как `gnuplot` и `xfig`, умеют экспортировать в MetaPost.

## 1.1 Здравствуй мир

Когда изучается новая программная технология первое что надо сделать, это сказать: «Здравствуй мир!». Что и сделаем, правда, пока на английском. Для этого следует создать файл `helloworld.mp` со следующим содержанием:

---

```
%Для просмотра
prologues := 1;
%Простой Hello World
beginfig (1);
    label ("Hello_World!~~~~~" ,(0,0));
endfig;
% <<конец обработки>> – необходимо вставить в конце файла
end.
```

---

Далее скомпилируем этот код:

```
> mpost helloworld.mp
```

В результате полученный файл `helloworld.1` можно посмотреть с помощью любой программой понимающей `PostScript`, например, с помощью `gv`:



Рис. 1. Первый Hello World на MetaPost

Разнообразим пример. Создадим файл `linuxformat.mp` со следующим содержанием:

---

*% А теперь мы его заворачивали и раскрасили*

```
beginfig(1);  
  for alpha:=90 step -3 until 0:  
    label(btex Linux Format etex  
      scaled (5*(1-alpha/100)) rotated alpha,(0,0))  
    withcolor (max(1-alpha/45,0)*red+  
      min(alpha/45,2-alpha/45)*green+  
      max(alpha/45-1,0)*blue);  
  endfor;  
endfig;  
end.
```

---

Скомпилируем код и результат преобразуем в pdf:

```
> mpost linuxformat.mp  
> pstopdf linuxformat.1
```

Полученный в результате этой операции файл `linuxformat-1.pdf` можно посмотреть, например, с помощью программы `acroread` или `xpdf`.

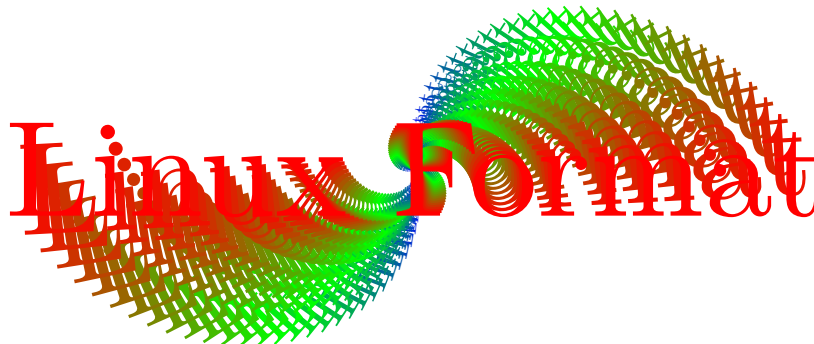


Рис. 2. «Продвинутый» Hello World на MetaPost

Всё, что находится между `btex` и `etex` обрабатывается внешней программой. По умолчанию это `TeX`. Чтобы воспользоваться возможностями `LaTeX` надо в самом начале `linuxformat.mp` строчки вида:

---

```
verbatimtex \documentclass{article}  
\begin{document}  
etex;
```

---

Эти строчки указывают `mpost`, что при обработке любых текстовых вставок им должны предшествовать команды `\documentclass{article}` и `\begin{document}`.

Добавим в `linuxformat.mp` ещё один рисунок. Для этого скопируем уже имеющийся код первого рисунка, вставим его до заключительной команды «`end.`» и немного подправим первые три строчки:

---

```
%Математика вместо Hello World
beginfig(2);
  for alpha:=90 step -9 until 0:
    label(btex \(\mathrm{f}(x)=\frac{1}{\sqrt{2\pi}}\int\limits_{-\infty}^{\infty}e^{-\frac{x^2}{2\sigma^2}}dx\) etex
  ...
```

---

Создание новых картинок на базе уже имеющегося кода в MetaPost — обычное дело. Со временем накапливается *своя* библиотечка примитивов, что сильно облегчает создание новых иллюстраций. Кажущаяся сложность выражений компенсируется исключительной гибкостью языка, позволяющая повторно использовать набранный код.

Скомпилируем код ещё раз:

```
> mpost -tex=latex linuxformat.mp
> pstopdf linuxformat.2
```

Таким образом можно отобразить любую сколь угодно сложную математическую конструкцию.

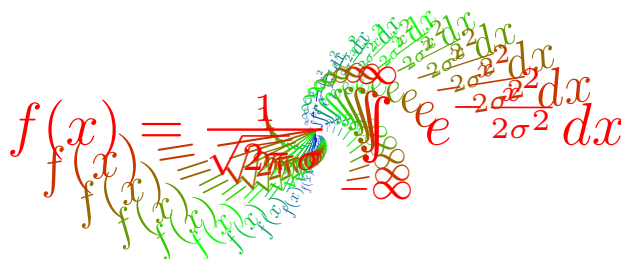


Рис. 3. Математическая вставка

Вместо того чтобы запускать `mpost` с опцией `-tex=latex`, можно установить переменную окружения `TEX` равную `latex`. В `bash` это можно сделать так:

```
> export TEX=latex
```

Как видно из предыдущего примера, MetaPost и L<sup>A</sup>T<sub>E</sub>X умеют неплохо работать вместе. Усовершенствуем немного `linuxformat.mp`. Заменяем первые три строчки в файле на следующий код:

---

```
verbatimtex \input{preheader}  
\begin{document}  
etex;
```

---

и создадим файл `preheader.tex` с примерно следующим содержанием:

---

```
\documentclass[12pt]{article}  
%простешая кириллизация  
\usepackage[koi8-r]{inputenc}  
\usepackage[english,russian]{babel}  
\usepackage[indentfirst]{first paragraph indent}  
\usepackage{graphicx}
```

---

Это стандартный заголовок для файлов L<sup>A</sup>T<sub>E</sub>X. Если Вы набираете свои тексты в L<sup>A</sup>T<sub>E</sub>X, то можете поместить в `preheader.tex` всю свою преамбулу и включить её с помощью команды `\input`. Это позволит создавать текстовые вставки в картинках MetaPost тем же шрифтом, что и в обычном тексте, а также позволит пользоваться любыми предопределёнными вами командами.

Следует обратить внимание, что чем больше будет разрастаться заголовок, тем дольше будет обрабатываться МЕТА-код. Для увеличения скорости лучше всего воспользоваться классом `minimal`, вместо `article` и убрать все ненужные для рисования картинок пакеты.

Добавим третью картинку в `linuxformat.mp`, чуть изменив первоначальный код:

---

```
verbatimtex \input{preheader}  
\begin{document}  
etex;
```

...

```

%изменим направление вращения
beginfig(3) ;
  for alpha:=-90 step 3 until 0:
    label(btex LinuxFormat в России etex
      scaled (5*(1+alpha/100)) rotated alpha,(0,0))
    withcolor (max(1+alpha/45,0)*red+
      min(-alpha/45,2+alpha/45)*green+
      max(-alpha/45-1,0)*blue);
  endfor;
endfig ;
end.

```

---

Теперь мы можем говорить по русски.

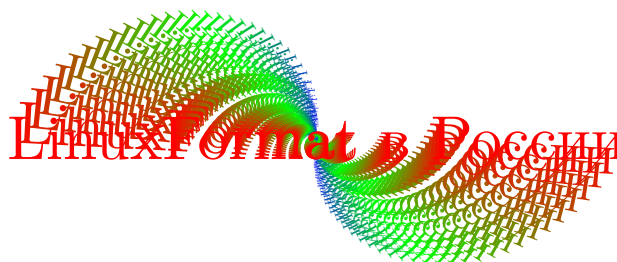


Рис. 4. Говорим по русски

## 1.2 MetaPost-конвейер

Чуть подробнее остановимся на том что же происходит. На вход программы `metapost` подаётся «МЕТА-картинка». «МЕТА-картинка» — это текстовый `mp`-файл с инструкциями на языке МЕТА. В одном `mp`-файле можно хранить несколько описаний картинок. При компиляции с помощью `mpost` создаются файлы с тем же именем, что и у исходного файла, но с расширениями в виде чисел, которые указываются в декларации `beginfig`. Результирующие файлы сразу можно вставлять в ЛАТЭХ-тексты с помощью обычного `\includegraphics`. Для этого достаточно в заголовок `tex`-файла добавить команду из ЛАТЭХ-пакета `graphics`:

```
\DeclareGraphicsRule{*}{eps}{*}{}{}
```



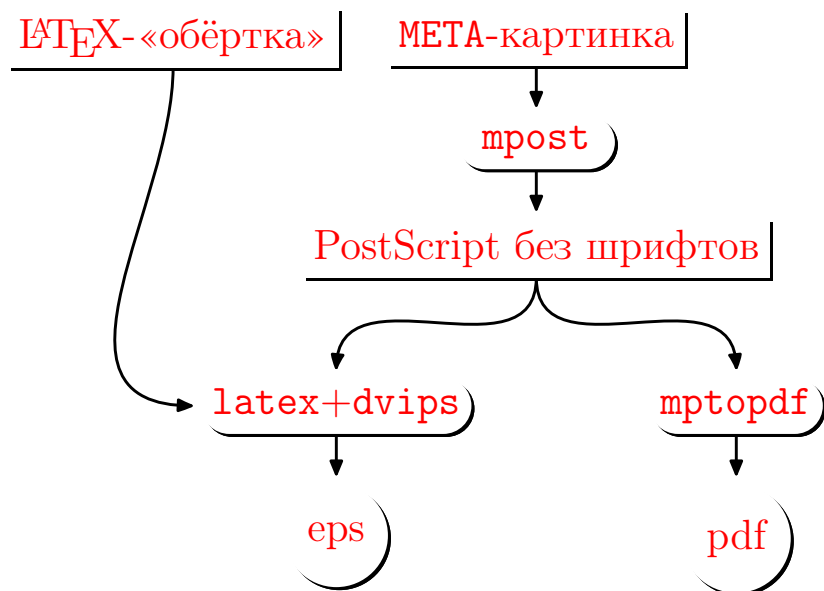


Рис. 5. MetaPost-конвейер

От «правильных» eps-файлов они отличаются только тем, что в них *не* «внедрены» шрифты, поэтому просмотреть их без дополнительной обработки не удастся.

Шрифты можно внедрить посредством программ `latex` и `dvips` с результатом в виде eps-файла или скрипта `mptopdf` с результатом в виде pdf-файла. Эти картинки уже можно использовать независимо любой программой, которая поддерживает эти векторные форматы.

### 1.3 Среда разработки

Удобная среда разработки понятие весьма относительное. Удобство зависит исключительно от Ваших предпочтений. Здесь будет представлена среда удобная для *меня*. Подробности настройки и установки упоминаемых здесь пакетов выходят за рамки этого теста.

В качестве базовой операционной системы я использую Linux. Это позволяет мне использовать для своей работы стандартные средства обработки текстовых данных, которые поставляются с этой системой. Например, ниже будет упомянута утилита `make`, которая позволяет автоматизировать ряд стандартных действий.

В качестве базового дистрибутива  $\text{\LaTeX}$  я использую дистрибутив  $\text{\TeX Live}$ . На сегодня этот дистрибутив является наиболее полным из дистрибутивов  $\text{\LaTeX}$ . Установочный образ для CD можно взять на любом из CTAN-архивов (Comprehensive  $\text{\TeX}$  Archive Network).

В качестве базового редактора я использую редактор `emacs`. Если Вы работаете с текстами в  $\text{\LaTeX}$ -формате, то при настройке этого редактора следует включить пакеты `auctex`<sup>1</sup> и `refTeX`<sup>2</sup>. `emacs` имеет простейшую встроенную поддержку для редактирования файлов в формате MetaPost.

Для автоматизации создания eps-картинок я использую Makefile примерно следующего вида:

---

```
#временный файл
tmp_file := tmp_file
#программы
LATEX := latex
MPOST := mpost -tex=latex
DVIPS := dvips

all:

    @echo "run:_make_mpfile.n.eps"

%.eps: % preheader.tex
    @echo "\input{preheader}">$(tmp_file).tex
    @echo "\DeclareGraphicsRule{*}{eps}{*}{}">>\
        $(tmp_file).tex
    @echo "\nfiles">>$(tmp_file).tex
    @echo "\begin{document}">> $(tmp_file).tex
    @echo "\thispagestyle{empty}">> $(tmp_file).tex
    @echo "\includegraphics{$(basename_$$$)}">>\
        $(tmp_file).tex
    @echo "\end{document}">> $(tmp_file).tex
    @$(LATEX) $(tmp_file)
    @$(DVIPS) -E -o $$$ $(tmp_file)
    @rm $(tmp_file).*
```

---

<sup>1</sup>Продвинутая система для работы с проектами  $\text{\LaTeX}$ .

<sup>2</sup>Автоматизация работы со ссылками.

```

clean:
    @rm -f mpx* *~ *.log *.mpx
    @rm -f $(tmp_file).*

#Зависимости для mpost-картинок.
#По одной для каждого числа из beginfig
%.1: %.mp preheader.tex preheader.mp
    $(MPOST) $<
...
%.64: %.mp preheader.tex preheader.mp
    $(MPOST) $<

```

---

Чтобы на выходе получить готовую eps-картинку с уже «внедрёнными» шрифтами, которую можно вставить уже куда угодно достаточно выполнить следующую команду:

```
> make <имя mp-файлы>.<номер картинки>.eps
```

Обычно, mp-файлам даются короткие имена.

Если в качестве результирующего формата картинок вас интересует pdf-формат, то в **Makefile** можно добавить что-то вроде:

---

```

MPTOPDF := mptopdf
MV := mv

%.pdf: % preheader.tex
    @$(MPTOPDF) $<
    @$(MV) 'echo $< | sed -e \
        "s/\.\.([0-9]\+)\$\$/-1.pdf/" ' $<.pdf

```

---

## 1.4 Чуть-чуть о META

В качестве базового языка, инструкции которого подаются на вход программы MetaPost используется язык **META**.

Язык **META** поддерживает следующие типы данных:

**boolean** объект булева типа, то есть либо true, либо false,

**numeric** обычное число (по умолчанию, если переменная не описана то она считается numeric),

**pair** точка — пара чисел  $(x,y)$  в случае декартовых координат или  $R \cdot \text{dir } \alpha$  в случае полярных координат,

**pen** перо — то, чем компьютер рисует (в подавляющем большинстве случаев используется круглое перо **pencircle**),

**color** цвет — тройка чисел  $(r,g,b)$ ,

**path** путь — совокупность точек с описанием соединения между ними,

**picture** картинка — совокупность путей и точек,

**string** строка — ASCII строка,

**transform** преобразования — линейные преобразования, которые можно применять к объектам типа **pair**, **pen**, **path** и **picture**.

Имена переменных в **META** могут состоять из нескольких лексем. Лексемы могут быть либо буквенными, либо числовыми. Например, переменная **x11** состоит из трёх лексем. Её можно переписать более понятным способом **x[1].1**, то есть числовая лексема по сути указывает на номер элемента в массиве, а следующая за ней буква уточняет элемент структуры. Возможность упускать «**[]**» в написании имён переменных упрощает в некоторых случаях восприятие кода ( $x_{11}$  — это  $x$ -координата границы линии слева по направлению движения для первой точки пути **z[]**) и сокращает объём программы. В замен, если Вам нужны просто переменные без подобных особенностей, то Вам придётся ограничиться только *буквенными* комбинациями.

Все переменные необходимо объявлять перед использованием. Исключением являются переменные типа **numeric**. Массивы объявляются и используются следующим образом:

```
pair w[];  
w1:=(10,5);  
w[2]=w[1];
```

Взаимодействие переменных, чисел и операторов вполне естественно, но достаточно нетривиально. Описание этого достойно отдельного раздела. В любом случае следует действовать по правилу: если сомневаетесь, то расставляйте скобки в нужных местах.

В **META** можно упускать некоторые из операторов для сокращения записей, например, **2\*x** соответствует записи **2x**. Но будьте осторожны,

так как  $1/2x$  это  $0.5x$ , что более естественно с точки зрения математики, но не программирования. В МЕТА сначала обрабатываются числовые лексемы.

Набор стандартных вычислительных операций расширен с учётом специализации языка. В частности поддерживаются операции пифагорова сложения  $a++b = \sqrt{a^2 + b^2}$ , пифагорова вычитания  $a+-+b = \sqrt{a^2 - b^2}$ , целочисленное деление `div` и возведение в степень  $x**y = x^y$ .

В языке присутствуют операторы цикла, условных переходов и тому подобное. Отличительной особенностью МЕТА является возможность решать линейные уравнения. Например, выражения вида  $C = 1/2[A, B]$  означает, что точка  $C$  находится ровно посередине между точками  $A$  и  $B$ .

Программу `metapost` можно использовать в режиме калькулятора для вычислений на языке МЕТА. Это позволяет проверить правильность ваших предположений относительно языка. Пример сеанса представлен ниже:

---

```

baldin@evgueni:~$ mpost
This is MetaPost, Version 0.901 (Web2C 7.5.5)
**\relax

*a:=10;

*b:=8;

*c:=a+-+b;

*show c;
>> 6
*show (3-sqrt 5)/2;
>> 0.38197
*show angle(1,sqrt 3);
>> 60.00008
*show 2**10;
>> 1024.00003
*show infinity;
>> 4095.99998
*show epsilon;
>> 0.00002

```

```
*show infinity-infinity;
>> 0
*end
Transcript written on mpout.log.
```

---

После вывода приглашения **\*\*** следует набрать команду `\relax`. Далее можно вводить команды MetaPost. Делать это надо аккуратно, так как этот режим не поддерживает «истории команд». В начале не предполагалось, что MetaPost можно использовать и так тоже. С помощью команды `show` можно вывести результат на экран. Закончить сеанс можно с помощью команды `end`.

Обратите внимание, что на просьбу вывести бесконечность (`infinity`) MetaPost выдал 4095.99998 — это максимальное значение, которое может принимать переменная типа `numeric`. Причём в процессе вычисления результат может привывать «бесконечность», но ответ должен быть меньше или равен её, иначе будет выдана ошибка. Минимальный шаг изменения типа `numeric` равен `epsilon`, или точнее  $1/256/256$ . При создании рисунка эти ограничения не существенны, так как диапазон изменения чисел вполне велик, чтобы вместить все элементы. Но в любом случае это тоже необходимо учитывать.

Если Вы хотите вычислить однострочное выражение, то на первоначальное приглашение **\*\*** можно ввести `expr`. В этом случае `mpost` считает файл `expr.mf` и на любое ваше действие будет выдаваться ответ:

---

```
baldin@evgueni:~$ mpost
This is MetaPost, Version 0.901 (Web2C 7.5.5)
**expr
(/usr/local/texlive/2005/texmf-dist/metafont/base/expr.mf
gimme an expr: 2(a+3b)-2b
>> 4b+2a
gimme an expr: 1/3[a,b]
>> 0.33333b+0.66667a
```

---

## 1.5 Литература

Язык МЕТА, который используется в MetaPost за некоторыми исключениями полностью соответствует диалекту МЕТА, который используется в программе создания шрифтов METAFONT.

Основной книгой по языку МЕТА является «Всё про МЕТАFONT» Дональда Э. Кнута. В 2003 году издательством Вильямс был выпущен русский перевод этой классической книги (ISBN 5-8459-0442-0). Исходники англоязычного оригинала «The METAFont book» можно найти на любом CTAN архиве. Эта книга, как и другие произведения Д.Э. Кнута имеет несколько уровней сложности. Даже с нулевым начальным уровнем знания предмета Вы можете прочитать книгу полностью — её строение это позволяет, но для дальнейшего продвижения эту книгу придётся перечитывать не один раз. Каждое прочтение приносит новое понимание. Если Вы имеете хоть какое-то отношение к программированию, то книги Д.Э. Кнута *надо* читать. Именно благодаря подобным людям информация может претендовать на фундаментальность.

На русском языке информацию о MetaPost можно найти в «Путеводителе по пакету L<sup>A</sup>T<sub>E</sub>X и его графическим расширениям» М.Гуссенса, С.Ратца и Ф.Миттельбаха от издательства Мир (ISBN 5-03-003388-2).

Все остальные источники в основном англоязычные. Прежде всего это «A User's Manual for MetaPost» Джона Хобби (Jhon D. Hobby) — файл `mpman.pdf`. Этот текст можно найти в документации к дистрибутиву MetaPost. Этому тексту предшествовало несколько «основополагающих» статей, которые при желании можно легко найти в интернете. Очень качественным руководством пользователя отметился Андрэ Хек (André Heck) <http://remote.science.uva.nl/~heck/Courses/mptut.pdf>. Так же представляет интерес книга «metafun» от Ганса Хагена (Hans Hagen) — она находится в открытом доступе, имя файла `metafun-p.pdf`.

В интернете можно найти интересный ресурс под названием «Metapost : exemples» от Vincent Zoonekynd <http://zoonek.free.fr/LaTeX/>. Это страничка с огромным количеством простых примеров. Небольшое неудобство состоит в том, что это франкоговорящий ресурс, с другой стороны текста там немного. Зеркало примеров расположено, например, здесь <http://tex.loria.fr/prod-graph/zoonekynd/metapost/metapost.html>.

Ко всей перечисленной электронной документации можно получить доступ через страничку CTAN, посвящённую MetaPost: <http://www.tug.org/metapost.html>