

Создание иллюстраций в MetaPost. (из серии компьютерные ТеХнологии)

© Балдин Е.М.

28 сентября 2006 г.

Аннотация

Эта статья была опубликована в 10ом (июньском) 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 декабря 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. Текущий текст в отличии от журнального варианта корректор не просматривал. Все вопросы по содержанию, а так же замечания и предложения следует задавать мне по электронной почте <mailto:E.M.Baldin@inp.nsk.su>.

Содержание

1	Введение в MetaPost	3
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
2	Базовые элементы	16
2.1	Рисуем по точкам	16
2.2	Пути	19

2.3	Вставка текста	23
2.4	Заливка	25
2.5	Цвета	27
3	Начала автоматизации	29
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
4	Графики и диаграммы	43
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	ЭТЭХ рисует с помощью MetaPost	56
4.6	gnuplot	58
4.7	Подведём итоги	59
5	Дополнительные главы	60
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 TEX	75
6	Заключение	77

5 Дополнительные главы

Всё описать невозможно, но ничто не запрещает попробовать.

Возможности META в то или иной мере уже изложены. В этой статье будут разобраны полезные приёмы, которые можно применять при кодировании картинки и описаны некоторые из стандартных пакетов, поставляемых с MetaPost.

Исходники стандартных MetaPost пакетов, обычно находится в директории $\$(\text{TEXMF})/\text{texmf-dist}/\text{metapost}/$, а документация к ним в директории $\$(\text{TEXMF})/\text{texmf-dist}/\text{doc}/\text{metapost}/$, где $\$(\text{TEXMF})$ — корневая директория для дистрибутива ЛАТЭХ. Сказанное верно для Т_ЕXLive.

5.1 Пакет boxes

`boxes` — один из самых первых пакетов общего назначения, появившихся в MetaPost. Его предназначение — это рисовать простые диаграммы. Он проектировался как более изощрённая замена для пакета Брайана Кернигана `pic` (`man pic`).

С другой стороны функциональности этого пакета достаточно для реализации автоматической генерации достаточно сложных зависимостей. Для примера разберём как была реализована диаграмма объясняющая действие MetaPost-конвейер во введении в цикл статей по MetaPost.

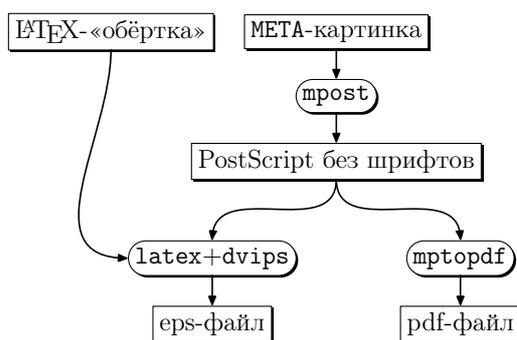


Рис. 32. MetaPost-конвейер

```

%в отличие от boxes здесь ещё определено окружение rboxit
input rboxes;
%определяем ещё один вид box'а
%параметры можно передавать и так
%expr — изолированные выражения
%text — абсолютно всё, что передаётся
vardef drawshadowed(expr dx,dy)(text t) =
  fixsize(t);
  forsuffixes s=t:
    fill bpath.s shifted (dx,dy);
    unfill bpath.s;
    drawboxed(s);
% можно было напечатать только текст
% и не рисовать рамку
% draw pic(s) withcolor red;
  endfor;
enddef;

%пример boxes
beginfig(2) ;
  numeric u;u:=1mm;
  %определяем box'ы
  %определяем box с прямыми углами
  boxit.a(btex \meta-картинка etex);
  %определяем box с закруглёнными углами
  rboxit.b(btex \texttt{metapost} etex);
  %определяем жёсткую связь между a и b
  b.n = a.s - (0,5u);
  boxit.c(btex PostScript без шрифтов etex);
  c.n = b.s - (0,5u);
  rboxit.e(btex \texttt{latex}\(+\)\texttt{dvips} etex);
  e.n=c.s-(20u,10u);
  boxit.d(btex \LaTeX—<<обёртка>> etex);
  d.e=a.w-(5u,0);
  boxit.g(btex eps-файл etex);
  g.n=e.s-(0,5u);
  rboxit.h(btex \texttt{mptopdf} etex);
  h.n=c.s-(-20u,10u);

```

```

boxit .i (btex pdf-файл etex);
i.n=h.s-(0,5u);
%разрешаем зависимости и рисуем boxes
drawshadowed(1/3u,-1/3u,a,b,c,d,e,g,h,i);
pickup pencircle scaled 0.3u;
%рисуем стрелки (сдвигка вида (0u,-1/3u) появилась,
%так как было определено новое окружение drawshadowed)
drawarrow a.s — b.n-(0u,-1/3u);
drawarrow b.s — c.n-(0u,-1/3u);
drawarrow c.s{dir -90} .. {dir -90}e.n-(0u,-1/3u);
drawarrow c.s{dir -90} .. {dir -90}h.n-(0u,-1/3u);
drawarrow e.s — g.n-(0u,-1/3u);
drawarrow h.s — i.n-(0u,-1/3u);
drawarrow d.s{dir -90} .. {dir 0}e.w-(1/3u,0u);
endfig;

```

Обратите внимание на ещё один вид цикла `forsuffixes`. Эта запись позволяет разбирать произвольное число передаваемых аргументов.

Алгоритм создания диаграммы следующий:

- С помощью команд `boxit`, `rboxit` или `circleit` объявляем `box`. Имя `box`'а добавляется к команде через точку как суффикс. В данном примере определено 8 `box`'ов от «a» до «i» включительно.
- Составляем уравнения связей. Через точку к именам `box`'ов можно добавить один из восьми суффиксов `n` — север (верхняя точка), `s` — юг (нижняя точка), `w` — запад (крайняя точка рамки слева), `e` — восток (крайняя точка рамки справа) и `sw`, `bw`, `se`, `sw` — комбинации уже перечисленных суффиксов, которые соответствуют углам рамки. Уравнение связи `b.n = a.s - (0,5u)`; можно описать следующим образом: северная (верхняя) точка `box`'а «b» находится ниже южной (нижней) точки `box`'а «a» на `5u`. При составлении уравнения связей следует пользоваться только знаком равенства. Это не присваивание, а именно уравнение, которое требуется разрешить.
- С помощью команды `drawboxed` нарисовать «закодированную» диаграмму. При исполнении этой команды разрешается система уравнений. По умолчанию, если не указать специально, `box` располагается в точке `(0,0)`.

- Нарисовать стрелки между элементами диаграммы. Можно пользоваться именами box'ов со стандартными суффиксами.

Прежде чем использовать этот пакет необходимо прочитать соответствующий раздел в руководстве пользователя по MetaPost Джона Хобби `mpman.pdf`.

5.2 Фейнмановские диаграммы

Простота MetaPost позволяет использовать как базу для построений более высокого порядка. Существует несколько подобных ЛАТЭХ-пакетов. О пакете `mfpic` уже упоминалось. В этом разделе описывается ещё один старейший пакет созданный по этой технологии.

В 1995 году Торстен Охл (Torsten Ohl) представил пакет `feynmp` для рисования фейнмановских диаграмм. Фейнмановские диаграммы используются для вычисления сумм большого числа вкладов от элементарных процессов. В своё время эта технология довольно сильно продвинула технику вычислений в физике высоких энергий. Её можно использовать везде, где сложный процесс можно описать с помощью элементарных приближений.

Пакет `feynmp`, это пакет ЛАТЭХ, который для рисования диаграмм использует MetaPost.

Интересующий меня с целью извлечения $\Gamma_{J/\psi \rightarrow e^+e^-}$ процесс имеет в Борновском приближении следующий вид:

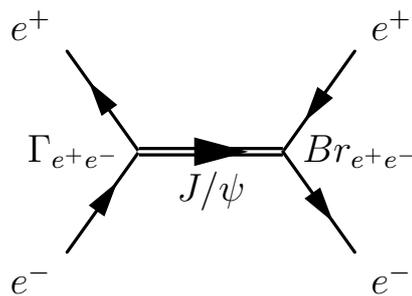


Рис. 33. Простейшая фейнмановская диаграмма

Это не сложная диаграмма и если не считать метки, то для её описания требуется всего пять операторов:

```

%Файл eepsiee.tex
%пакет для рисования фейнмановских диаграмм
\usepackage{feynmp}
...
\begin{fmffile}{ee-psi-ee} %имя тр-файл
\begin{fmfgraph*}(110,62) %размер диаграммы
\fmfleft{ei, pi} %вершины-источники
\fmfright{eo, po} %исходящие вершины
\fmflabel{\$e^-\$}{ei} %метка источника e^-
\fmflabel{\$e^+\$}{pi} %метка источника e^+
\fmflabel{\$e^+\$}{po} %метка исходящей вершины
\fmflabel{\$e^-\$}{eo} %метка исходящей вершины
%линия соединяющая источники
\fmf{fermion}{ei, Ji, pi}
%линия соединяющие исходящие вершины
\fmf{fermion}{po, Jo, eo}
%Метка для начальной вершины промежуточной частицы
\fmflabel{\$\Gamma_{\ee}\$}{Ji}
%Метка для конечной вершины промежуточной частицы
\fmflabel{\$Br_{\ee}\$}{Jo}
%Соединительная линия
\fmf{heavy, label=\$J/\psi\$}{Ji, Jo}
\end{fmfgraph*}
\end{fmffile}

```

Окружение `\beginfmffile` в качестве параметра требует имя тр-файла в который будут писать команды `META`. В данном примере имя файла определено как `ee-psi-ee.mp`. Для того чтобы получить диаграмму, описанную в файле `eepsiee.tex` были проделаны следующие действия:

```

> latex eepsiee.tex
> mpost ee-psi-ee.mp
> latex eepsiee.tex

```

После выполнения этих команд результат можно посмотреть с помощью программы `xdvi` или преобразовать `dvi`-файл в `PostScript` или `pdf`.

Следующая простейшая диаграмма не имеет особого смысла. Она просто демонстрирует, возможности пакета: «Сотрудничество» с `MetaPost`

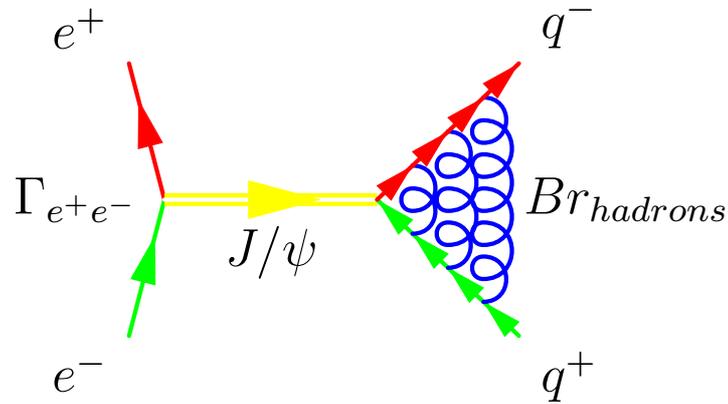


Рис. 34. Раскрашенная фейнмановская диаграмма

даёт возможность `feynmp` получить доступ к цвету. Цвет можно определять точно так же, как он определяется в `MetaPost`.

```

%Файл eepsihad.r.tex
\begin{fmffile}{ee-psi-hadr}
\begin{fmfgraph*}(110,62)
\fmfleft{i1,i2}
\fmfright{o1,o2}
\fmflabel{\$e^-\$}{i1}
\fmflabel{\$e^+\$}{i2}
\fmflabel{\$q^+\$}{o1}
\fmflabel{\$q^-\$}{o2}
\fmf{fermion,foreground=green}{i1,v1}
\fmf{fermion,foreground=red}{v1,i2}
\fmf{fermion,foreground=green}{o1,v2,v3,v4,v5}
\fmf{fermion,foreground=red}{v5,v6,v7,v8,o2}
%изменяем натяжение (расталкиваем вершины)
\fmf{heavy,label=\$J/\psi$,tension=1/3,
foreground=red+green}{v1,v5}

\fmffreeze
\fmf{gluon,foreground=blue}{v2,v8}
\fmf{gluon,foreground=blue}{v3,v7}
\fmf{gluon,foreground=blue}{v4,v6}
\fmfv{label=\$Gamma_{ee}\$}{v1}

```

```

\fmfv{label=$Br_{\hadr}$,label.dist=0.3w}{v5}
\end{fmfgraph*}
\end{fmffile}

```

При создании диаграммы MetaPost пытается оптимально расположить вершины, минимизируя взвешенную сумму расстояний между ними:

$$L(v_1, \dots, v_n) = \frac{1}{2} \sum_{i,j} t_{i,j} (v_i, v_j)^2,$$

где параметр $t_{i,j}$ — «натяжении» между вершинами v_i и v_j . По умолчанию натяжение равно 1. С помощью опции `tension` натяжение можно изменить. Чем меньше натяжение, тем сильнее расталкиваются вершины.

Обычно, полученная диаграмма со значениями по умолчанию на требует вмешательства в код. В случае если результат не устраивает, как правило, достаточно изменить натяжение для одного, максимум для двух соединений.

Подробнее документацию по пакету `feynmp` можно найти в файле `$(TEXMF)/texmf-dist/doc/latex/feynmf/manual.ps.gz`, где `$(TEXMF)` — корень дерева ЛАТЭХ (заведомо верно для Т_ЕXLive).

5.3 Фракталы

MetaPost, естественно, поддерживает рекурсию. Пользуясь этим, а так же генераторами случайных `uniformdeviate` или `normaldeviate` можно организовать фрактальную «лесопосадку».

```

%фрактальная лесопосадка
beginfig(1) ;
  u:=lmm; branchrotation := 50;
  offset := 180-branchrotation;
  thinning := 0.7;
  shortening := 0.8;
  def drawit(expr p, linethickness, col) =
    draw p withpen pencircle scaled linethickness withcolor col;
  enddef;
  %A — основание B — направление роста, n — число ветвей,
  %size — толщина дерева, col — цвет

```



Рис. 35. Фрактальный «лес»

```

vardef tree(expr A,B,n,size,col) =
  save C,D,thickness; pair C,D;
  thickness := size;
  C := shortening[B, A rotatedaround(B,
    offset+uniformdeviate(branchrotation))];
  D := shortening[B, A rotatedaround(B,
    -offset-uniformdeviate(branchrotation))];
  if n>0:
    drawit(A—B, thickness, col);
    thickness := thinning*thickness;
    tree(B, C, n-1, thickness, col);
    tree(B, D, n-1, thickness, col);
  else:
    drawit(A—B, thickness, col);
    thickness := thinning*thickness;
    drawit(B—C, thickness, col);
    drawit(B—D, thickness, col);
  fi;
enddef;

numeric nbr,nx,ny,ell,size;
color col;
nx:=10;ny:=5;
pair A;
for ix:=1 upto nx:

```

```

for iy:=1 upto ny:
  nbr:=4+uniformdeviate 5;
  ell:=nbr*u;
  x:=ix*(1+1/20*normaldeviate);
  y:=iy*(1+1/20*normaldeviate);
  A:=(20u*(x+y*sqrt(2)/2),20u*y*sqrt(2)/2);
  size:=ell/5;
  col:=(uniformdeviate 1,uniformdeviate 1,uniformdeviate 1);
  show ix , iy , A, ell , nbr , size , col ;
  tree(A, A+(0,ell) , nbr , size , col );
endfor ;
endfor ;
endfig ;

```

Увлечение автоматически создаваемыми картинками влечёт за собой опасность переполнения памяти. MetaPost создавался в то время, когда к используемой памяти относились исключительно бережно.

Если при компиляции картинки будет выдана ошибка `MetaPost capacity exceded`, то необходимо поправить файл конфигурации `texmf.cnf`. Обычно, этот файл можно найти в директории $\$(\text{TEXMF})/\text{web2c}/\text{texnf.cnf}$ ⁴.

За выделяемый объём памяти для MetaPost отвечает переменная `main_memory.mpост`. Значение этой переменной должно быть меньше значения `main_memory`, которая ставит верхнюю границу по использованию памяти для всех Т_ЕX-подобных программ. Обратите внимание на то, что в конфигурационном файле может быть несколько переменных с одним и тем же именем — используется последнее значение.

В моём случае ограничение по памяти стояло в 15 Мб. Очевидно, что используемую память можно безболезненно увеличить. С другой стороны, если происходит переполнение, то это возможно значит, что в алгоритм закралась ошибка.

После исправления значения переменных следует регенерировать форматные файлы, например, с помощью команды `texconfig init`.

5.4 Увеличительное стекло

Бывает в процессе создания картинки необходимо увеличить участок для лучшей детализации. MetaPost предоставляет средство для этого в

⁴В Т_ЕXLive можно создавать локальные настройки

виде команды `clip`, которая позволяет обрезать картинку по любому замкнутому пути. Ниже идёт код, который позволил увеличить кусок циферблата корабельных часов.

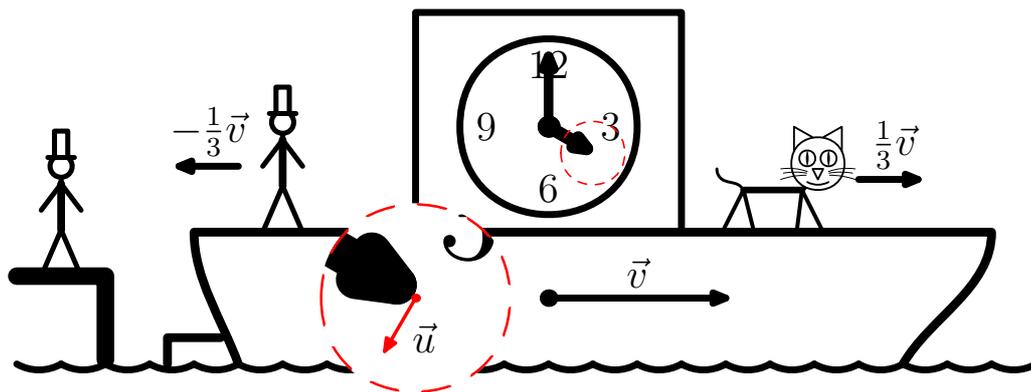


Рис. 36. Преобразование Галилея

```

%Файл dop.tr
R:=3.6u;
path q;
%определяем форму области которую хотим вырезать
q:=(-R,0)..(R,0)..cycle;
%определяем центр вырезаемой области (orig)
%и местоположение увеличенного участка (copy)
pair orig,copy;
orig=(65u,24u);copy=(45u,7.5u);
%отмечаем вырезаемую область
draw q shifted orig dashed evenly scaled 1/2u
      withpen pencircle scaled 0.2u withcolor red;
picture p;
%сохраняем текущую картинку в переменной p
p:=currentpicture;
%коэффициент увеличения
numeric scale;scale:=3;
%Обрезаем картинку p по замкнутому пути q
clip p to (q shifted orig);
%чистим область, где будет нарисована увеличенная копия
fill q scaled scale shifted copy withcolor white;

```

```

%рисуем копию
draw p shifted -orig scaled scale shifted copy;
%рисуем дополнительную стрелку и метку
%на уже увеличенной копии
draw copy withpen pencircle scaled 1u withcolor red;
drawarrow copy--(copy+7u*dir -120)
      withpen pencircle scaled 0.4u withcolor red;
label .lrt (btex \(\vec{u}\) etex ,
          1/3[copy ,(copy+7u*dir -120)]);

```

Пользуясь этим приёмом можно создать любой фон для какой-угодно фигуры, сделать любую штриховку.

5.5 Штриховка

clip-технология, которая упоминалась выше, позволяет создать любой вид штриховки. В стандартной поставке MetaPost идут пакеты, которые этим пользуются.



Рис. 37. Пакет hatching.

Пакет hatching.mp представляет из себя обычный «хак»:

```

%Файл dop.mp
%пример использования пакета hatching
input hatching;
beginfig(3) ;
  numeric u;u=1mm;
  path q;
  q=(10u,0)..{dir -135}(0u,0u){dir 135}..
    (-10u,0)..{dir 90}(0u,0u){dir -90}..cycle;
  hatchfill q withcolor red withcolor (25,1u,-1);
  draw q;
endfig;

```

Функция `hatchfill` берёт информацию о штриховке из данных, которые следуют с декларацией `withcolor`. Сигналом, что эти данные предназначены именно для `hatchfill` является то, что голубая компонента тройки чисел меньше 0. Красная компонента соответствует углу наклона штриховки, зелёная расстоянию между штрихами. Подробности можно узнать в краткой документации к пакету.

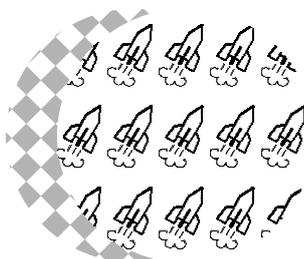


Рис. 38. Пакет `mpattern`

Более развитым (возможно излишне) является пакет `mpattern`:

```

%Файл dor.tex
%пример использования пакета mpattern
input mpattern;

%создаём "обои" ракета
beginpattern(rocket);
  begingroup; save u;
  u:=1mm;
  draw Rocket scaled 2/3u rotated -30;
  patternbbox(-5u,-8u,5u,8u);
  endgroup;
endpattern;

%создаём "обои" повёрнутая клетка
beginpattern(rotated_checker);
  fill unitsquare scaled 4mm rotated 45 withcolor .7white;
endpattern;

beginfig(4);
  numeric u; u:=1mm;

```

```

path p;
z1=(10u,0u);
p=fullcircle scaled 50u;
%рисует фигуру в клетку
fill p withpattern rotated_checker;
unfill p shifted z1;
%рисует фигуру "в ракету"
fill p shifted z1 withpattern rocket;
endfig;

```

Здесь уже определены свои команды для создания штриховки. Пакет не свободен от недостатков, но он достаточно прост и можно легко «довести» код до необходимой кондиции. Простота — это общее свойство пакетов MetaPost. МЕТА **вынуждает** писать кратко. Подробности можно найти в документации к пакету.

5.6 Вставка eps

Пакет `exteps.mp` позволяет включить eps-картинку как единый объект. Краткая документация доступна в файле `exteps.pdf`, поставляемом с этим пакетом.

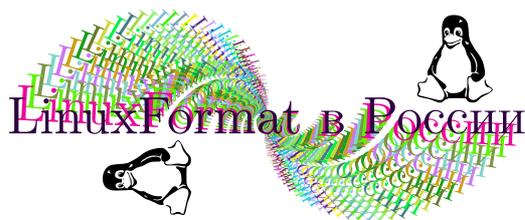


Рис. 39. Добавляем чёрно-белых Туксов

```

%Файл dop.mp
input exteps;
%Включение eps-картинки
beginfig (6) ;
    numeric u; u:=1mm;

% базовая надпись

```

```

for alpha:=-90 step 3 until 0:
  label(btex LinuxFormat в России etex
    scaled (5*(1+alpha/100)) rotated alpha ,(0 ,0))
  withcolor
    (uniformdeviate 1,uniformdeviate 1,uniformdeviate 1);
endfor ;

% посадим Тукса (penguin.eps) справа
begineps "penguin.eps" ;
% ширина картинка
width:=30u;
% сдвиг картинка от начала координат
base:= (60u,5u);
% можно нарисовать решётку на картинка
% grid := true;
% обрезание по bounding box
% clip := true;
endeps ;

% посадим Тукса (penguin.eps) слева
begineps "penguin.eps" ;
% поворот
angle:=90;
width:=30u;
base:= (-30u,-40u);
endeps ;
endfig ;

```

5.7 Большие числа

Когда обсуждалась вставка меток в качестве примера был представлен треугольник Паскаля. Из-за ограничений `META` на максимальный размер числа треугольник Паскаля отрисовывался до 14ой строчки.

Это ограничение можно обойти воспользовавшись пакетом `sarith.mp`. Этот пакет был создан специально под пакет `graphics.mp`. Что вполне естественно, так как если для рисования небольших рисунков больших

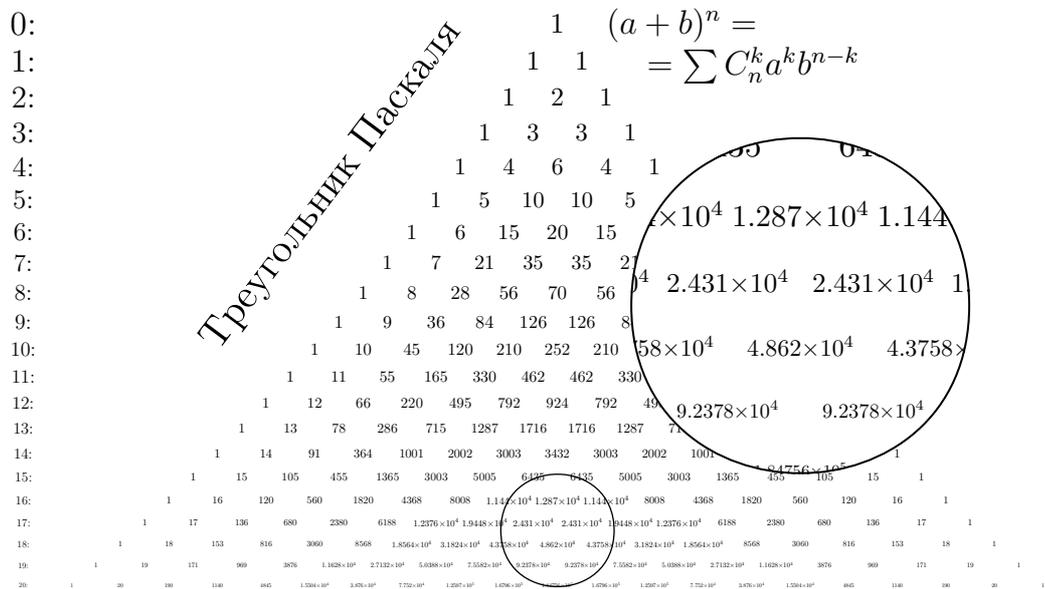


Рис. 40. Треугольник Паскаля.

чисел быть не может, то при анализе данных это вполне рядовая ситуация.

```

%Файл dop.m
%Macros for arithmetic on strings that represent big numbers
input sarith;
%Треугольник Паскаля с большими числами
beginfig(5);
  numeric u;
  u = 1mm;
  numeric dy, dx, x, y, i, j, sy, ds, nlast; dy:=5u;
  dx:=6u; x=0; y=0; nlast:=20;
  %для хранения больших чисел нужна строка
  string n[][];
  ds=0.04; sy=0.032;
  picture z;
  for i:=0 upto nlast:
    dy:=dy*(1-sy);
    y:=y-dy;
    for j:=0 upto i:

```

```

if (j=0) or (j=i):
  n[i][j]:=Sabs 1; %сложение
else :
  n[i][j]:=n[i-1][j-1] Sadd n[i-1][j]; %сложение
fi
%формат вывода
z:=thelabel(format("%6g", n[i][j]),(0,0));
x:=dx*(j-i/2);
label(z scaled (1-ds*i),(x,y));
endfor
z:=thelabel.lft(decimal(i)&" :", (0,0));
label(z scaled (1-ds*i),(dx*(-nlast/2-1),y));
endfor
endfig ;

```

Для представления больших чисел используется встроенный тип **string** — строка.

Операция	действие
Scvnum <number>	конвертация в numeric
Sabs <number>	абсолютное значение
<number> Sadd <number>	сложение
<number> Ssub <number>	вычитание
<number> Smul <number>	умножение
<number> Sdiv <number>	деление
<number> Sleq <number>	сравнение <=
<number> Sneq <number>	сравнение <>

Таблица 1. Операции над большими числами

Документация к пакету **sarith.mp** является частью документации к **graphics.mp** (**mpgraph.pdf** от Джона Хобби).

5.8 Макрос **TEX**

Вы наверное уже обратили внимание, что метки в MetaPost являются статическими. Всё, что между **btex** и **etex** отдаётся L^AT_EX для обработки. Это значит, что метку нельзя скомпоновать из различных кусков.

Решением является немного модифицированный макрос из стандартного пакета `TEX.mp`:

```
%Файл macros.mp
vardef TEX primary s =
  write "verbatim" to "mptextmp.mp";
  write "\input{preheader-base}" to "mptextmp.mp";
  write "\begin{document}" to "mptextmp.mp";
  write "etex" to "mptextmp.mp";
  write "btex_&_etex" to "mptextmp.mp";
  write EOF to "mptextmp.mp";
  scantokens "input_mptextmp"
enddef;
```

В процессе вызова макроса `TEX`, всё что находится внутри него записывается в `mptextmp.mp`, а затем этот файл включается в основной файл прямо во время компиляции. То есть, происходит модификация программы прямо во время компиляции. Пример использования макроса идёт ниже:

```
%Файл coord.mp
numeric u;u:=2mm;
for i:=-15 step 5 until 15:
  label.lft (TEX("\("&decimal(i)&"\)"),(-10u,i*u));
endfor;
```

При формировании ASCII строчки использовался оператор `&` для объединения. Хотя, в данном случае можно было обойтись только статическими записями, но иногда этот приём может пригодиться для целей автоматизации, где скорость компиляции не главное.

Основной минус этого способа в том, что он очень «мееедленный». Для ускорения следует оставить в `preheader-base.tex` только самые необходимые инструкции. Что-то вроде:

```
\documentclass[12pt]{article}
\usepackage[warn]{mathtext}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\usepackage[english,russian]{babel}
```

6 Заключение

Естественно, в таком кратком рассказе невозможно изложить *всё*. Основные понятия и приёмы уже рассмотрены, но множество вещей выпало из обзора. В частности, совсем не рассмотрены 3D объекты, создание геометрических чертежей, параметризация пути и векторные поля в MetaPost. Это не смертельно, так как есть весьма качественная свободная литература посвящённая этим предметам.

MetaPost достаточно легко использовать после обучения, но обучение перед использованием просто необходимо. Это свойство всех стоящих технологий, которые позволяют делать то, для чего компьютеры и существуют: *автоматизировать* рутинные действия, для выполнения которых не требуется задумываться. Думать же, в любом случае, прерогатива человека.