

An introduction to MetaPost

--- A programming language to generate
vector-based graphics

Fuxiao Xin
Oct 5, 2011

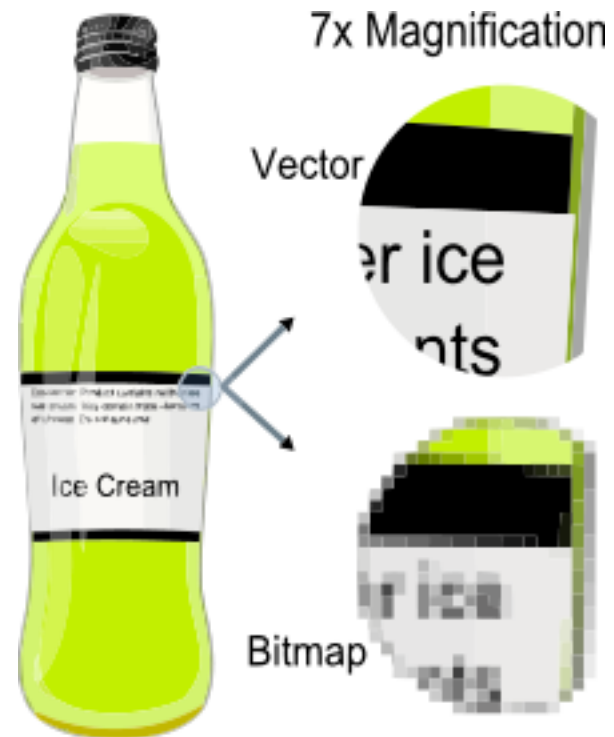
Reference: <http://www.tug.org/docs/metapost/mpman.pdf>

Bitmap VS. Postscript

Bitmap: Higher the resolution, larger the file

Postscript: scalability without loss of resolution

Pixels versus Vectors based



History



John D. Hobby

[Computing and Software Principles Research](#)
Bell Labs, Alcatel-Lucent

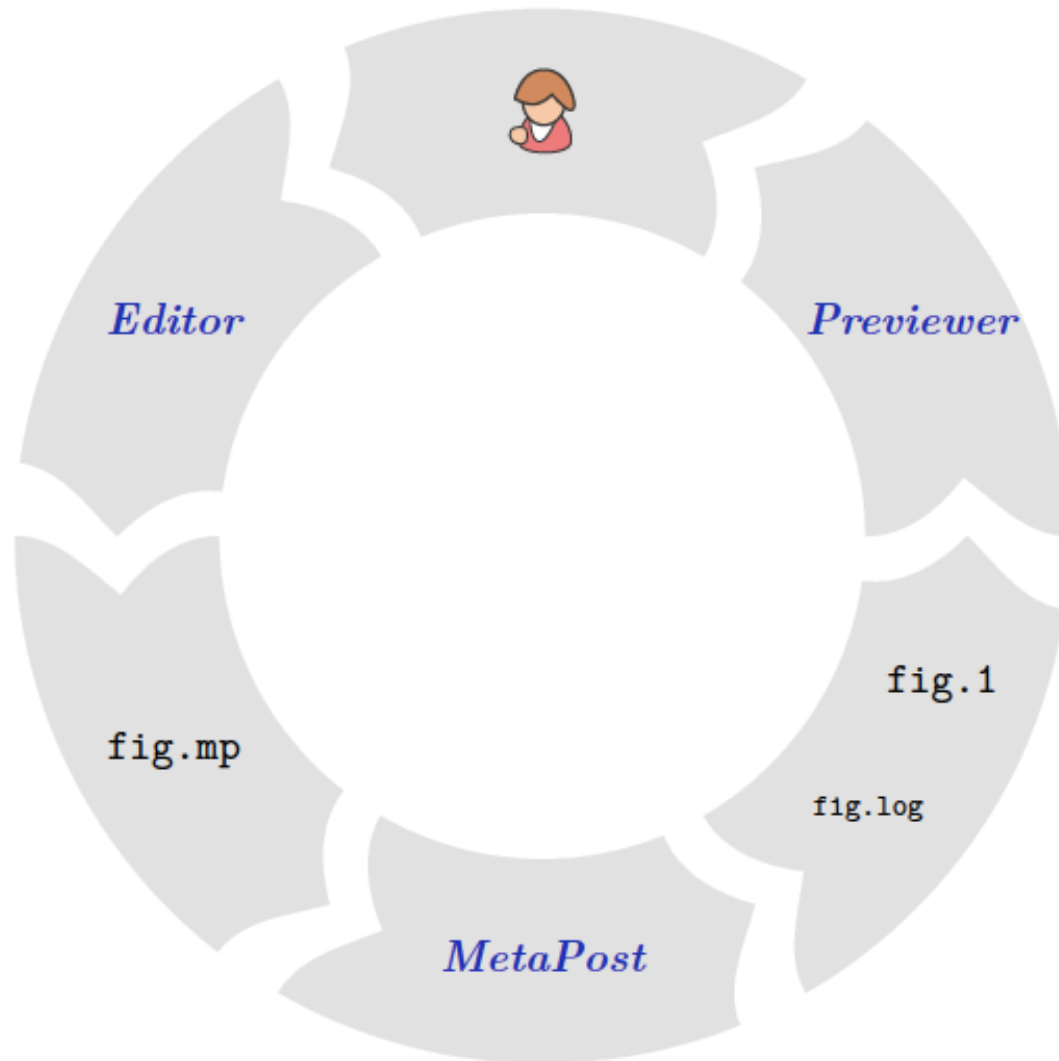
TeX, Metafont, MetaPost



Donald E. Knuth (高德纳),
Professor Emeritus of
[The Art of Computer Programming](#) at
[Stanford University](#)



The basic MetaPost workflow



To start with...

- <http://www.tlhiv.org/mppreview/>
- [http://www.tug.org/docs/metapost/
mpman.pdf](http://www.tug.org/docs/metapost/mpman.pdf)

Dot, Line, and Curve

Dot: `drawdot (0,0)`

Line: `draw (20,20)--(0,0)--(0,30)--(30,0)--(0,0)`

Curve: `draw (20,20)..(0,0)..(0,30)..(30,0)..(0,0)`

Unit: `in, cm, mm`

eg: `(2cm,2cm)--(0,0)--(0,3cm)--(3cm,0)--(0,0)`

Control line width: `pickup pencircle scaled 4pt`

Color: `withcolor red`

for loop

```
u=1cm;  
pickup pencircle scaled 4pt;  
for i=0 upto 2:  
    for j=0 upto 2: drawdot (i*u,j*u); endfor  
endfor
```

```
for i=0 step .1 until 1: show i; endfor
```

label

```
z0 = (0,0); z1 = (60,40);
z2 = (40,90); z3 = (10,70);
z4 = (30,50);
draw z0..z1..z2..z3..z4;
%draw z0..z1..z2..z3..z4..cycle;
%draw z0..z1..z2..z3--z4--cycle;

%labels
dotlabel.lft("z0", z0);
dotlabel.rt("z1", z1);
dotlabel.top("z2", z2);
dotlabel.lft("z3", z3);
dotlabel.lft("z4", z4);
%label.lft(btex $z^{4}$ etex, z4) withcolor red;
```

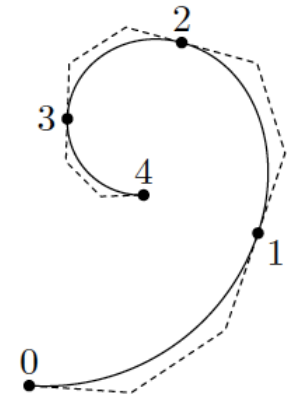
More on curves: control points, direction, tension, curl

Default: Bezier Cubic Curve with continuous slope and approximate continuous curvature

```
draw (0,0) .. (60, 40) .. (40,90) .. (10, 70) .. (30, 50) withcolor red;
```

```
draw (0,0)..controls (26.8,-1.8) and (51.4,14.6)
..(60,40)..controls (67.1,61.0) and (59.8,84.6)
..(40,90)..controls (25.4,94.0) and (10.5,84.5)
..(10,70)..controls ( 9.6,58.8) and (18.8,49.6)
..(30,50);
```

```
draw z0..z1{up}..z2{left}..z3..z4;
z0 = (0,0); z1=(10,10); z2=(30,10); z3=(40,0);
draw z0..z1..z2..z3;
draw z0..z1..tension 1.3..z2..z3 withcolor red;
draw z0..z1..tension 1.5..z2..z3 withcolor green;
```



Solve equation

Solve equations

$a+b=3;$

$2a=b+3;$

Show $a, b;$

Equal VS. Assignment

$a = 2; b = a; a := 3; c = a;$

$z<\text{number}> = (x<\text{number}>, y<\text{number}>)$

$z1=-z2=(.2in,0);$

$x3=-x6=.3in;$

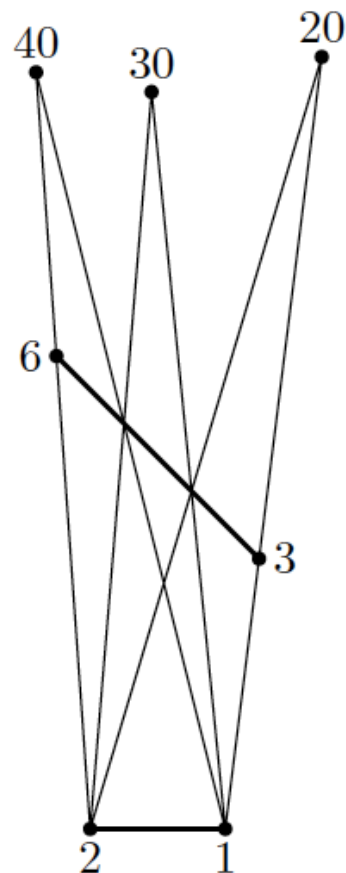
$x3+y3=x6+y6=1.1in;$

draw $z1..z2..z3..z6$

```

beginfig(13);
z1=-z2=(.2in,0);
x3=-x6=.3in;
x3+y3=x6+y6=1.1in;
z4=1/3[z3,z6];
z5=2/3[z3,z6];
z20=whatever[z1,z3]=whatever[z2,z4];
z30=whatever[z1,z4]=whatever[z2,z5];
z40=whatever[z1,z5]=whatever[z2,z6];
draw z1--z20--z2--z30--z1--z40--z2;
pickup pencircle scaled 1pt;
draw z1--z2;
draw z3--z6;
endfig;

```



Data Types

numeric, pair, path, transform, (rgb)color, cmykcolor, string, boolean, picture, and pen

Numeric: 0.6

Pair: (0, 10)

Path: (0,0) .. (0, 10)

Transform: shifted, rotated, slanted, scaled, xscaled, yscaled, zscaled

Color: Black, white, red, green, blue, or 0.4white, or (0.4, 0.4, 0.4)

Cmykcolor: (0,0,0,1)

String: "newfigure"

Boolean: True, false, and, or, not, <, <=, >, >=, =, <>

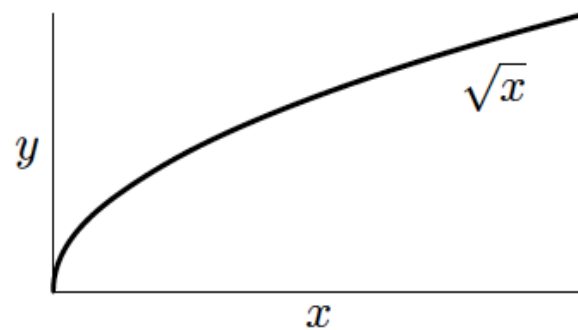
Picture: draw, drawdot..

Pen: pickup pencircle scaled 4pt;

```

beginfig(18);
numeric u;
u = 1cm;
draw (0,2u)--(0,0)--(4u,0);
pickup pencircle scaled 1pt;
draw (0,0){up}
  for i=1 upto 8: ..(i/2,sqrt(i/2))*u endfor;
label.lrt(btex  $\sqrt{x}$  etex, (3,sqrt 3)*u);
label.bot(btex  $x$  etex, (2u,0));
label.lft(btex  $y$  etex, (0,u));
endfig;

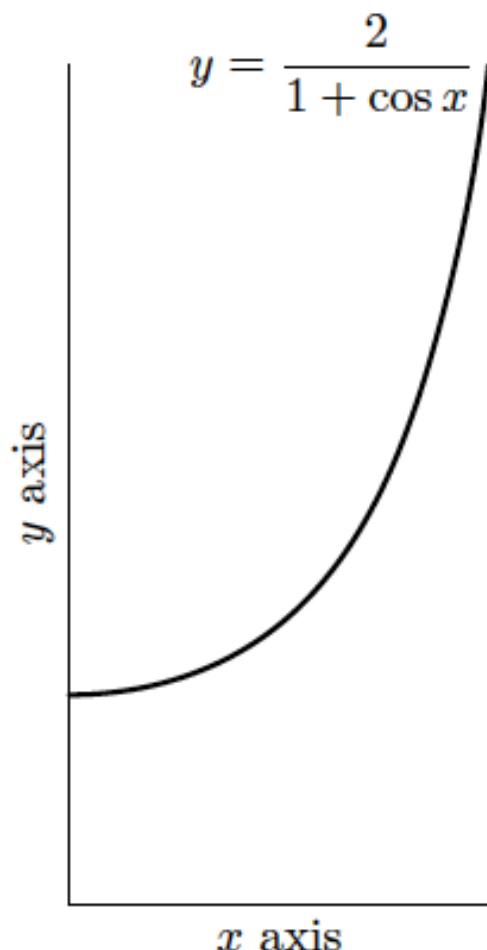
```



```

beginfig(19);
numeric ux, uy;
120ux=1.2in; 4uy=2.4in;
draw (0,4uy)--(0,0)--(120ux,0);
pickup pencircle scaled 1pt;
draw (0,uy){right}
  for ix=1 upto 8:
    ..(15ix*ux, uy*2/(1+cosd 15ix))
  endfor;
label.bot(btex  $x$  axis etex, (60ux,0));
label.lft(btex  $y$  axis etex rotated 90,
  (0,2uy));
label.lft(
  btex  $\displaystyle y=\frac{2}{1+\cos x}$  etex,
  (120ux, 4uy));
endfig;

```

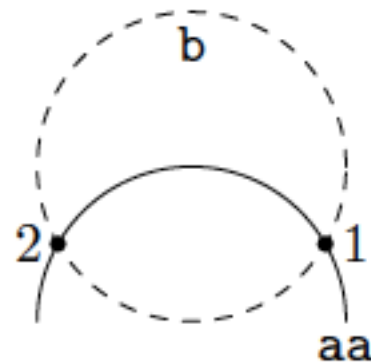


Fill and buildcycles

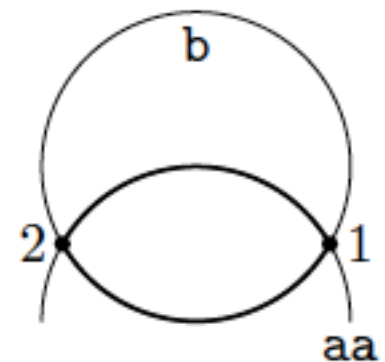
```
beginfig(21);  
path p;  
p = (-1cm,0)..(0,-1cm)..(1cm,0);  
fill p{up}..(0,0){-1,-2}..{up}cycle;  
draw p..(0,1cm)..cycle;  
endfig;
```



Buildcycle(aa,b)



(a)



(b)

```

h=2in; w=2.7in;
path p[], q[], pp;
for i=2 upto 4: ii=i**2;
p[i] = (w/ii,h){1,-ii}...(w/i,h/i)...(w,h/ii){ii,-1};
endfor

```

```

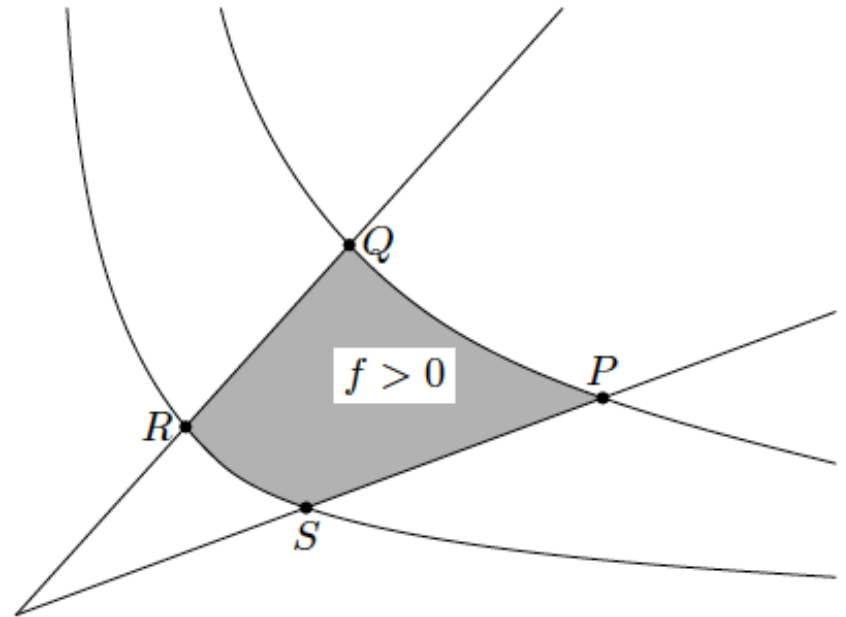
q0.5 = (0,0)--(w,0.5h);
q1.5 = (0,0)--(w/1.5,h);
pp = buildcycle(q0.5, p2, q1.5, p4);
fill pp withcolor .7white;
z0=center pp;
picture lab; lab=thelabel(btex $f>0$ etex, z0);
unfill bbox lab; draw lab;
draw q0.5; draw p2; draw q1.5; draw p4;
dotlabel.top(btex $P$ etex, p2 intersectionpoint q0.5);
dotlabel.rt(btex $Q$ etex, p2 intersectionpoint q1.5);
dotlabel.lft(btex $R$ etex, p4 intersectionpoint q1.5);
dotlabel.bot(btex $S$ etex, p4 intersectionpoint q0.5);

```

```

draw p2 withcolor red;
draw q0.5 withcolor green;
draw q1.5 withcolor blue;

```



Define macros

```
vardef a[]b(expr p) = p shifted (#@,b) enddef;
```

Then $a2b((1,2))$ means $(1,2)$ shifted $(a2,b)$ and $a3b((1,2)..(3,4))$ means $((1,2)..(3,4))$ shifted $(a3,b)$.

```
vardef z@#=(x@#,y@#) enddef;
```

```
def midpoint(expr a, b) = (.5[a,b]) enddef
```

Resources

- <http://tex.loria.fr/prod-graph/zoonekynd/metapost/metapost.html>
- <http://www.tlhiv.org/mppreview/>
- <http://www.tug.org/docs/metapost/mpman.pdf>