METAPOST

by Hooman Mesgary

SCALING

Same as PostScript

1 unit = 1/72 inch

Other units

- bp: Big Point
- pt: Printer's Point
- in: Inches
- cm: Centimeters
- mm: Milimeters

COMMANDS

drawdot

- drawdot (10,2)
- drawdot origin
- drawdot (2cm, 13mm)

draw

- draw (0,0)—(0,1)

pickup

- pickup pencircle scaled 4pt

VARIABLES

```
u = 1cm;
drawdot (2u, 3u);

z0 = (2u, 3u);
z1 = (3u, 2u);
draw z0--z1;
```

LOOP

```
u = 1mm;
for i=0 upto 4:
        drawdot (i*u,u);
endfor
```

PROGRAM

```
beginfig(1);
        draw (1cm,0)--(0,1cm);
endfig;
end
```

CURVES

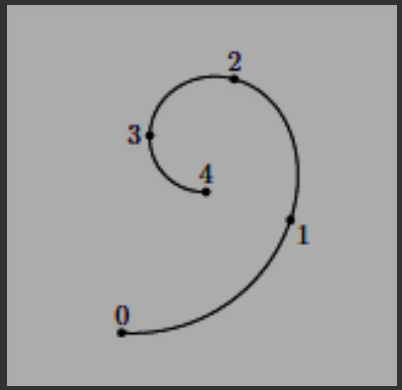
z0 = (0,0);

z1 = (60,40);

z2 = (40,90);

z3 = (10,70);

z4 = (30,50);



draw z0..z1..z2..z3..z4

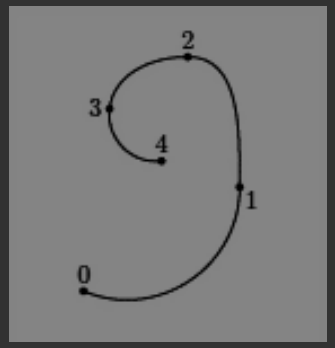
draw z0..z1..z2..z3--z4--cycle

draw z0..z1..z2..z3..z4..cycle

DIRECTION

draw z0..z1..z2..z3..z4

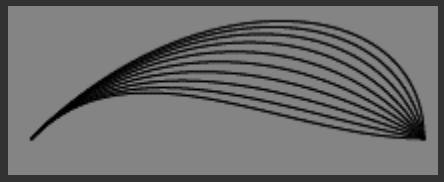
draw z0..z1{up}..z2{left}..z3..z4

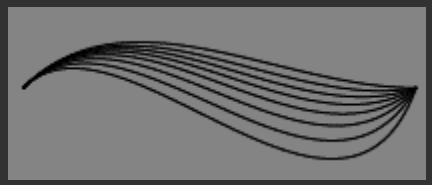


..{dir 60}(10,0){up}..

CONT.

```
beginfig(1)
        for a=0 upto 9:
                draw (0,0){dir 45}..{dir -10a}(6cm,0);
        endfor
endfig;
beginfig(2)
        for a=0 upto 9:
                draw (0,0){dir 45}..{dir 10a}(6cm,0);
        endfor
endfig;
```

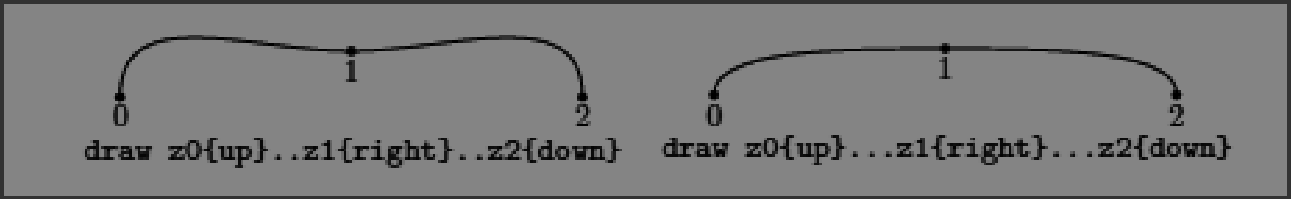




INFELCTION

draw z0{up}..z1{right}..z2{down}

draw z0{up}...z1{right}...z2{down}

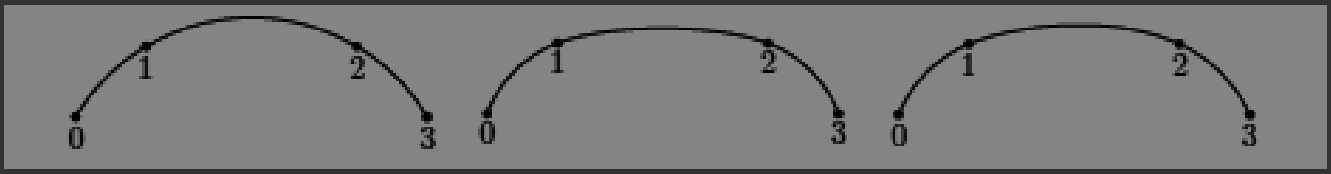


TENTION

draw z0..z1..z2..z3

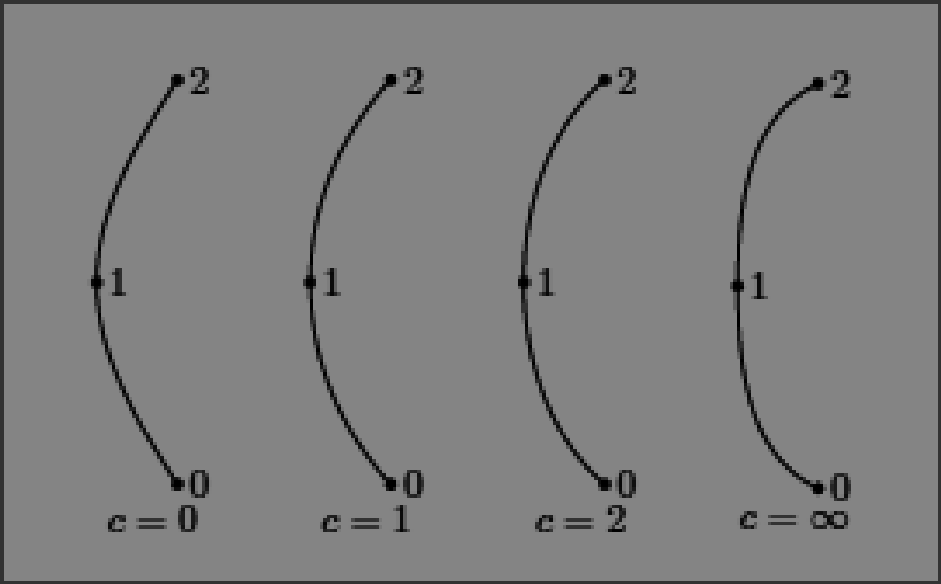
draw z0..z1..tension 1.3..z2..z3

draw z0..z1..tension 1.3 and 1..z2..z3



CURL

draw z0{curl c}..z1..{curl c}z2



EQUATION

$a+b=3;\ 2a=b+3;$

$a=2$

$b=1$

$a+b=2a-b=3$

ASSIGNMENT

a=a+1

a:=a+1

EQU. & ASSIGN.

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

x3=-x6=.3in;

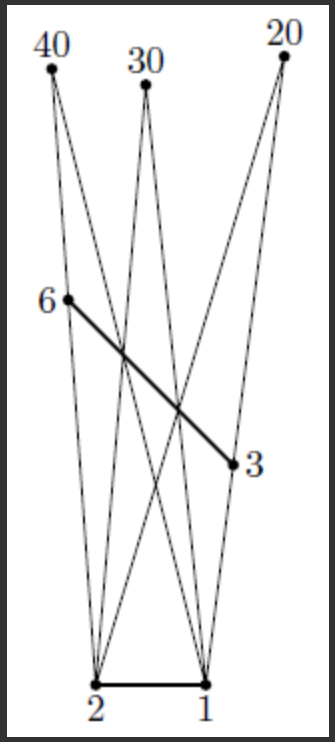
x3+y3=x6+y6=1.1in;

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

z3=(.3in,.8in); z6=(-.3in,1.4in);

SOLVING

```
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;
```



TRANSFORM

p transformed T

(*px, py*) (*tx, ty, txx, txy, tyx, tyy*)

(*tx* + *txxpx* + *txypy*, *ty* + *tyxpx* + *tyypy*)

$$
\begin{aligned}
(x,y)\ \texttt{shifted}\ (a,b) &= (x+a,\ y+b);\\
(x,y)\ \texttt{rotated}\ \theta &= (x\cos\theta - y\sin\theta,\ x\sin\theta + y\cos\theta);\\
(x,y)\ \texttt{slanted}\ a &= (x+ay,\ y);\\
(x,y)\ \texttt{scaled}\ a &= (ax,\ ay);\\
(x,y)\ \texttt{xscaled}\ a &= (ax,\ y);\\
(x,y)\ \texttt{yscaled}\ a &= (x,\ ay);\\
(x,y)\ \texttt{zscaled}\ (a,b) &= (ax-by,\ bx+ay).
\end{aligned}
$$

MORE TRANSFORM

transform T;

T = identity xscaled -1 rotated 90 shifted (1,1);

reflectedabout(p, q)

rotatedaround(p, θ)

qq = p transformed inverse T

FINDING TRANSFORMS

(0,1) transformed T' = (3,4);

(1,1) transformed T' = (7,1);

(1,0) transformed T' = (4,-3);

COLORS

black, white, red, green, blue

(0,0,0) (1,1,1)

(.4,.4,.4) = 0.4white

OPERAND

- * /
**
+ -

++
+-+

sqrt

CONT.

and
or
z1 dotprod z2 = x1*x2 + y1*y2
length “abcde”
substring(2,4) of “abcde” = “cd”
abs
round
floor
ceiling

ORDERS

-1[a,b] = -b

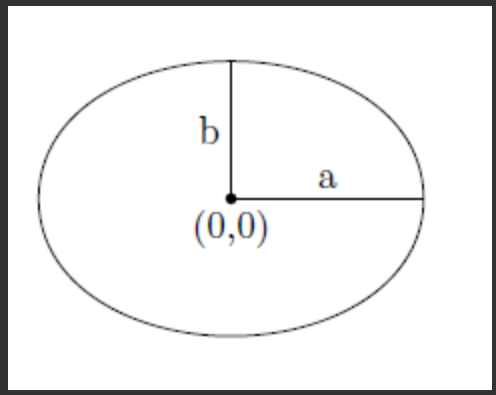
(-1)[a,b] = 2a-b

sqrt 2/3

sqrt (1+1)/3

LABELS

```
beginfig(17);
a=.7in; b=.5in;
z0=(0,0);
z1=-z3=(a,0);
z2=-z4=(0,b);
draw z1..z2..z3..z4..cycle;
draw z1--z0--z2;
label.top("a", .5[z0,z1]);
label.lft("b", .5[z0,z2]);
dotlabel.bot("(0,0)", z0);
endfig;
```



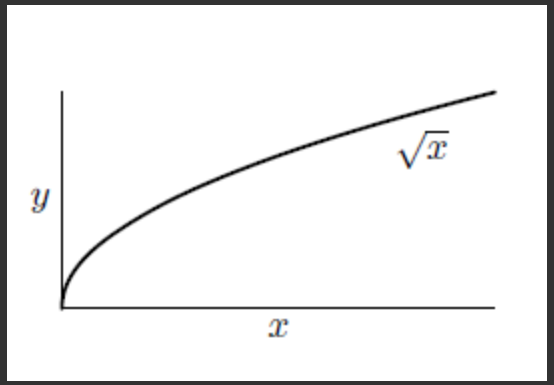
CONT.

⟨label suffix⟩ → ⟨empty⟩ | lft | rt | top | bot | ulft | urt | llft | lrt

label.lrt(btex $\sqrt x$ etex, (3,sqrt 3)*u)

USING TEX

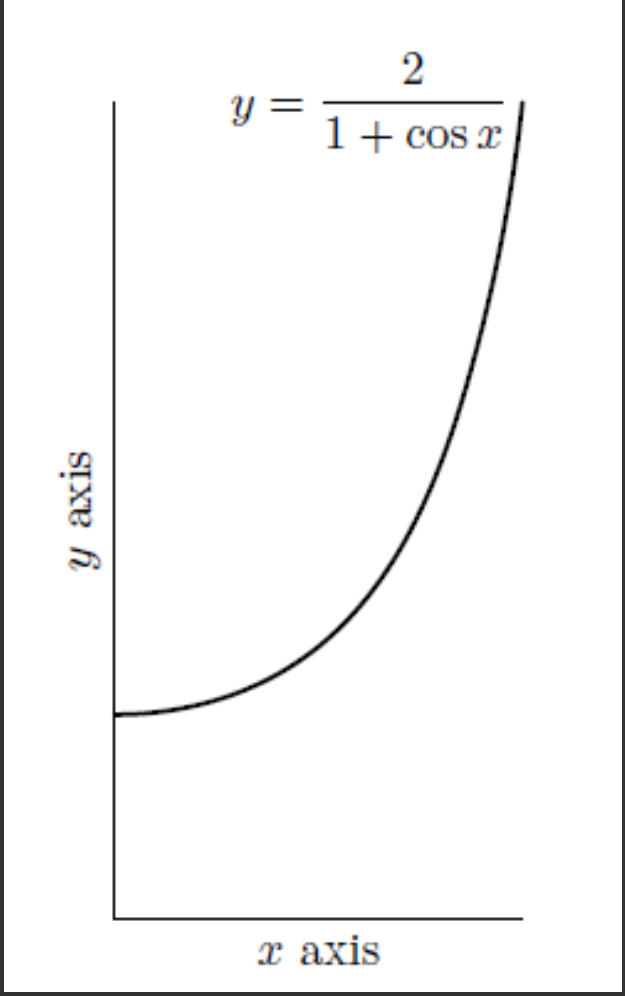
```
beginfig(18);
numeric u;
u = 1cm;
draw (0,2u)--(0,0)--(4u,0);
pickup pencircle scaled 1pt;
draw (0,0){up}
for i=1 upto 8: ..(i/2,sqrt(i/2))*u endfor;
label.lrt(btex $\sqrt x$ etex, (3,sqrt 3)*u);
label.bot(btex $x$ etex, (2u,0));
label.lft(btex $y$ etex, (0,u));
endfig;
```



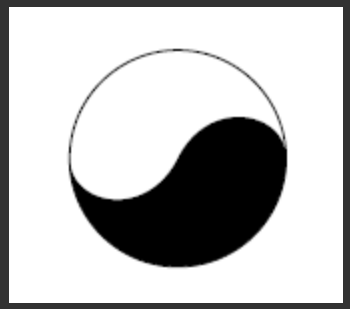
CONTINUES DRAWING

```
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 $\displaystyle
y={2\over1+\cos x}$ etex,
(120ux, 4uy));
endfig;
```

RESULT

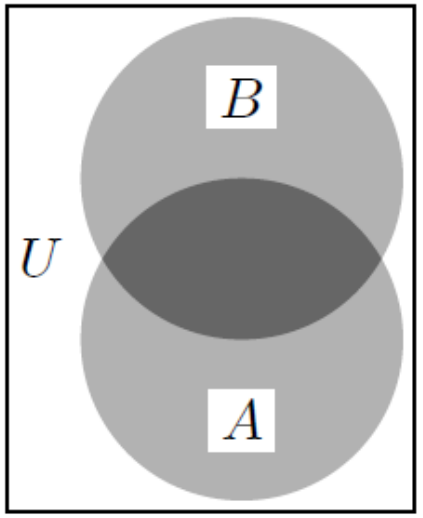


FILL



```
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;
```

```
beginfig(22);
path a, b, aa, ab;
a = fullcircle scaled 2cm;
b = a shifted (0,1cm);
aa = halfcircle scaled 2cm;
ab = buildcycle(aa, b);
picture pa, pb;
pa = thelabel(btex $A$ etex, (0,-.5cm));
pb = thelabel(btex $B$ etex, (0,1.5cm));
fill a withcolor .7white;
fill b withcolor .7white;
fill ab withcolor .4white;
unfill bbox pa;
draw pa;
unfill bbox pb;
draw pb;
label.lft(btex $U$ etex, (-1cm,.5cm));
draw bbox currentpicture;
endfig;
```



PATH WORKING

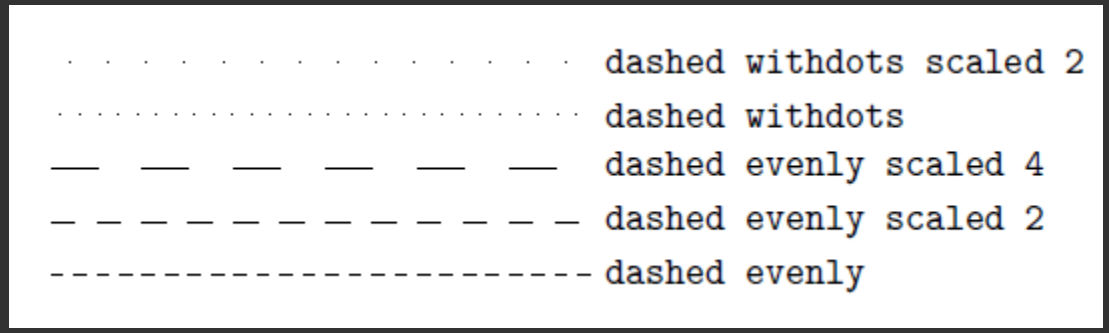
a intersectiontimes b

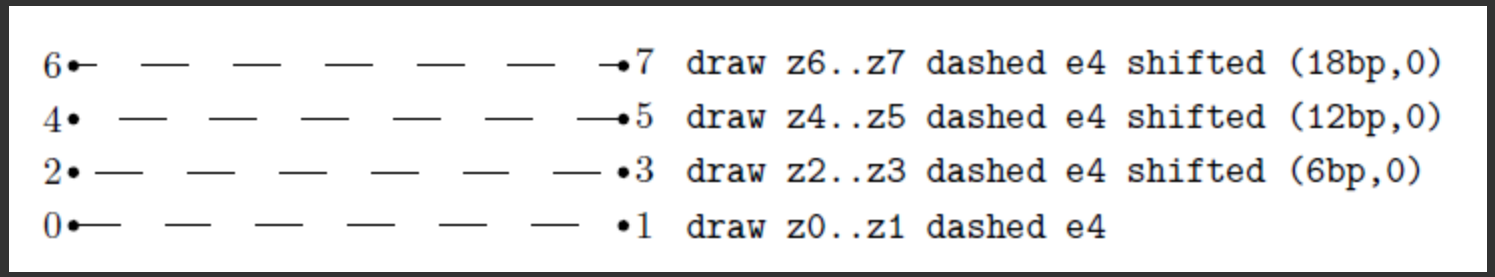
point t of p

p1 cutbefore p2

p1 cutafter p2

DASHED LINES





ARROWS

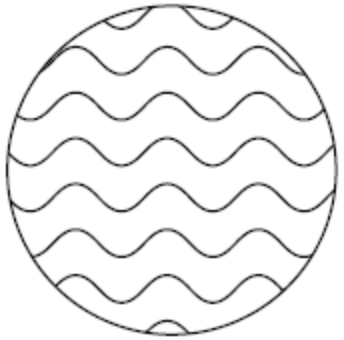
drawarrow z1..z2

drawarrow reverse (z1..z2)

drawdblarrow z1..z2

CLIPPING

```
beginfig(40);
path p[];
p1 = (0,0){curl 0}..(5pt,-3pt)..{curl 0}(10pt,0);
p2 = p1..(p1 yscaled-1 shifted(10pt,0));
p0 = p2;
for i=1 upto 3:  p0:=p0.. p2 shifted (i*20pt,0);
  endfor
for j=0 upto 8:  draw p0 shifted (0,j*10pt);
  endfor
p3 = fullcircle shifted (.5,.5) scaled 72pt;
clip currentpicture to p3;
draw p3;
endfig;
```



CONDITION

if $e1$: ... elseif $e2$: ... else: ... fi

```
def middlepoint(expr a) = if path a: (point
.5*length a of
else: .5(llcorner a + urcorner fi a) enddef;
```

LOOPS

for ⟨symbolic token⟩ = ⟨expression⟩ upto ⟨expression⟩ : ⟨loop text⟩ endfor

for ⟨symbolic token⟩ = ⟨expression⟩ downto ⟨expression⟩ : ⟨loop text⟩ endfor

for ⟨symbolic token⟩ = ⟨expression⟩ step ⟨expression⟩ until ⟨expression⟩ : ⟨loop text⟩ endfor

SAMPLE LOOP

```
for t=3.14, 2.78, (a,2a), "hello": show t; endfor

draw for p=(3,1),(6,2),(7,5),(4,6),(1,3): p-- endfor
cycle;

draw (3,1)--(6,2)--(7,5)--(4,6)--(1,3)--cycle;
```