**Bug fixes and maintenance for path resolution, binary, decimal, and double number systems. Also, more efficient integration with mplib by excluding the libraries needed only in LuaTeX.**
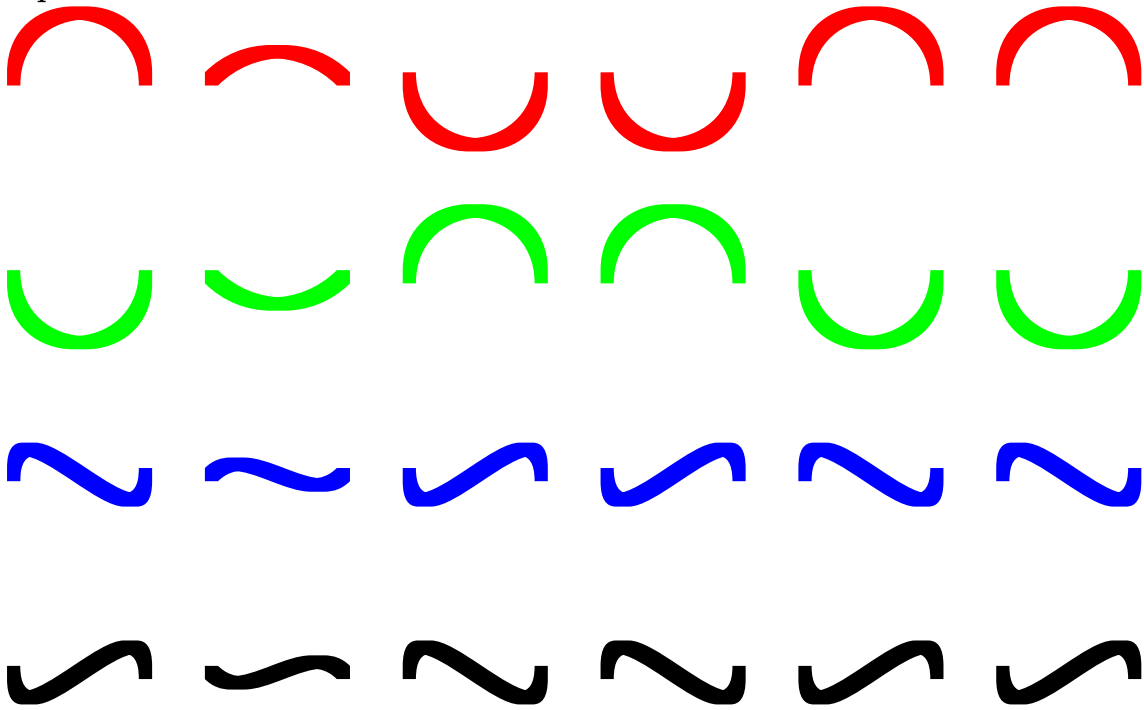
## 1 Fixed bugs when directions are ±180°

After several months of debugging, these bugs were caused by oversights — i.e. error in translations from PASCALWEB to CWEB. Probably this affect also others old bugs, as that one in http://tracker.luatex.org/view.php?id=835 A similar bug is described in http://www.tug.org/TUGboat/tb27-0/hoekwater.pdf, "5.2 Polygonal pen with 180 degree angles", but it cannot be reproduced with MetaPost 1.9991 .
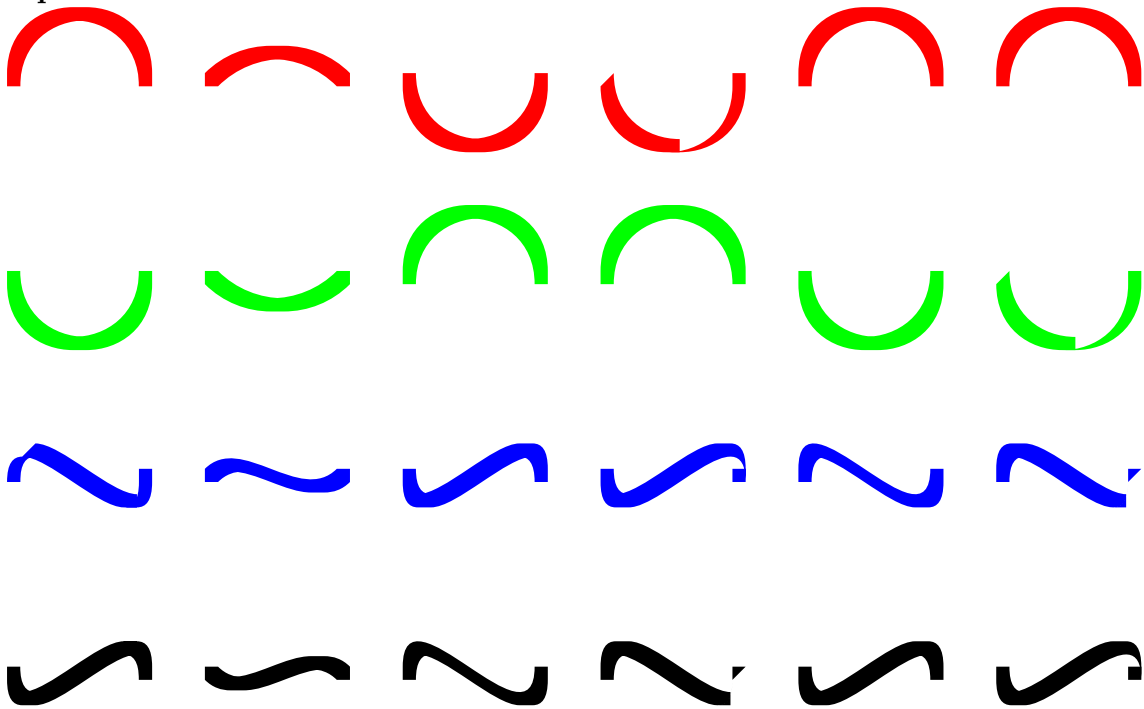
These are tests files by H. Hagen:

```
def test(expr a) =
draw image (
 draw (0,0){dir  a} .. {dir -a}(100,0) withpen pensquare scaled 10 withcolor red;
 currentpicture := currentpicture shifted (0,150);
 draw (0,0){dir -a} .. {dir  a}(100,0) withpen pensquare scaled 10 withcolor green;
 currentpicture := currentpicture shifted (0,150);
 draw (0,0){dir  a} .. {dir  a}(100,0) withpen pensquare scaled 10 withcolor blue;
 currentpicture := currentpicture shifted (0,150);
 draw (0,0){dir -a} .. {dir -a}(100,0) withpen pensquare scaled 10 withcolor black;
) ;
 currentpicture := currentpicture shifted (150,0) ;
enddef ;

beginfig(1);
test(90)      ; %
test(90-eps)  ; %
test(270)     ; %
test(270+eps) ; %
test(45)      ; %
test(90+eps)  ; %
endfig;
end.
```

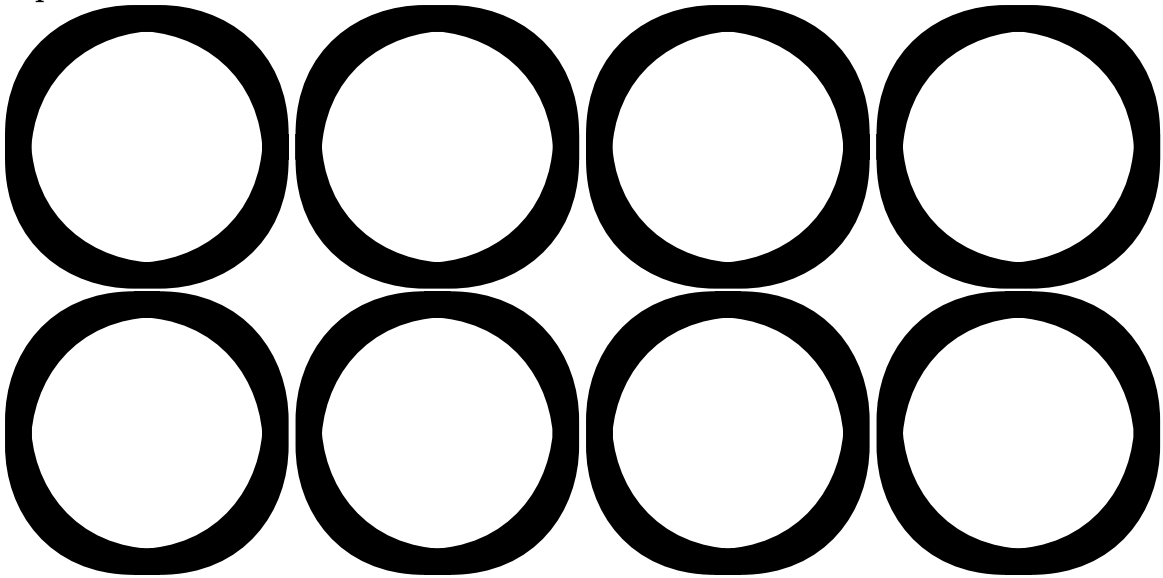**mpost 2.000:**



**mpost 1.9991:**

These are tests files by A. Kakuto:

```
beginfig(1);
path P;
pen A;
P:=(0,0){up}..(100,0){down}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
beginfig(2);
path P;
pen A;
P:=(0,0){down}..(100,0){up}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
beginfig(3);
path P;
pen A;
P:=(100,0){up}..(0,0){down}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
beginfig(4);
path P;
pen A;
P:=(100,0){down}..(0,0){up}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
beginfig(5);
```
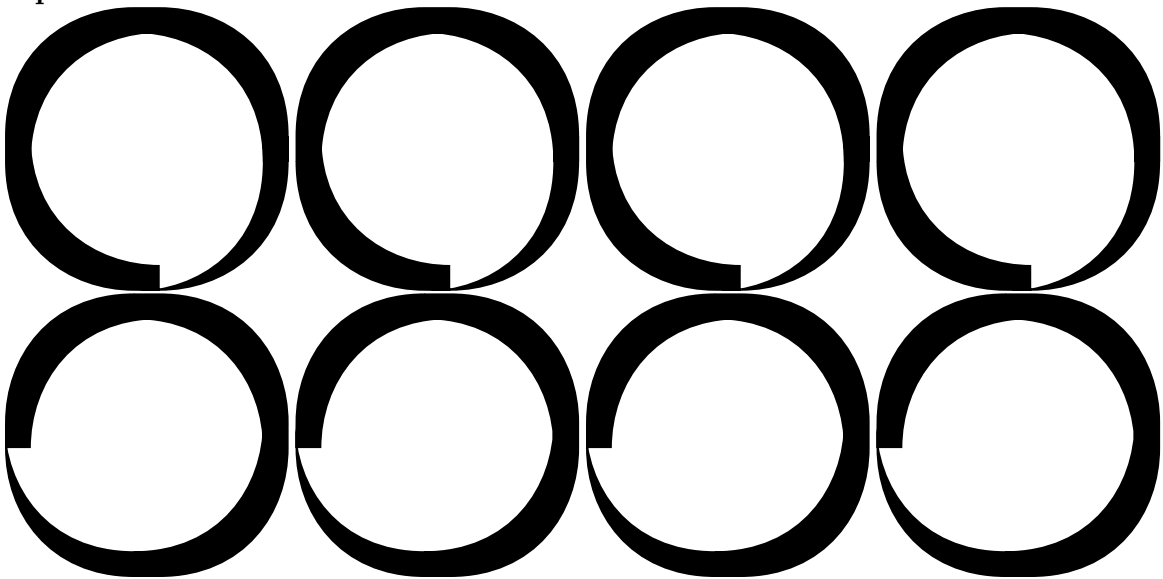
```
path P;
pen A;
P:=(50,-50){left}..(50,50){right}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
beginfig(6);
path P;
pen A;
P:=(50,-50){right}..(50,50){left}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
beginfig(7);
path P;
pen A;
P:=(50,50){left}..(50,-50){right}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
beginfig(8);
path P;
pen A;
P:=(50,50){right}..(50,-50){left}..cycle;
A:=pensquare scaled 10;
draw P withpen A;
endfig;
%
%
end.
```
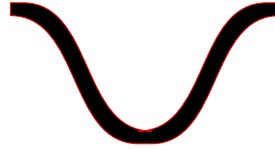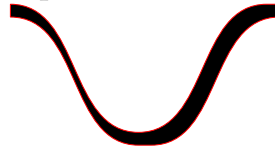
**mpost 2.000:**

**mpost 1.9991:**

Another similar issue (http://tracker.luatex.org/view.php?id=835):

```
beginfig(2);
  path p, e;
  pen sqpen;
  sqpen := pensquare scaled 10;
  p := (-100,100){right}..origin%
       ..{right}(100,100);
  e := envelope sqpen of p;
  draw p withpen sqpen;
  draw e withcolor red;
endfig;
end
```

**mpost 2.000:**

**mpost 1.9991:**

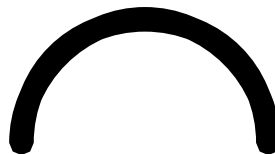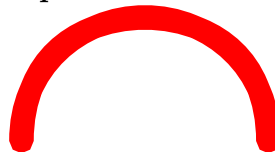This example is in the section 5.2 "Polygonal pen with 180 degree angles" of "Metapost Developments" (*EuroTeX 2005 (Pont-à-Mousson) proceedings — special non-TUGboat joint publication*), which is also now correctly rendered:

**mpost 2.000:**

```
beginfig(1);
pickup makepen(fullcircle scaled 10 pt);
draw (0,0){up}..(50,50)..{down}(100,0);
draw (0,70){up}..{down}(100,70) withcolor
red;
endfig;
end;
```

## 2 Weird MetaPost output when using PNG

The issue at http://tracker.luatex.org/view.php?id=976 was due to a wrong sequence of drawing: the matrix of transformation was applied after `mp_png_path_out`.

```
outputformat := "png";                  endfig;
outputtemplate := "%j-%c.%o";           end.

beginfig(1);
    h=100;
    w=100;
    y1=y2=y3=0;
    y4=y5=y6=h;
    x1=x4=0;
    x2=x5=w;
    x3=x6=2*w;
    pickup pencircle xscaled 0.2w %
    yscaled 0.04w rotated 45;
    draw z1..z3..z6{z2-z6}..z5..%
    {z4-z2}z4..cycle;
```
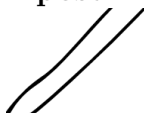
**mpost 2.000:**

**mpost 1.9991:**

## 3 Fixed segmentation fault with `transform`

The following code ended in a seg. fault (see http://tracker.luatex.org/view.php?id =993):

```
show mpversion;
pair P; P = (a,b); show P;
color R; R = (d,e,f); show R;
cmykcolor C; C = (g,h,i,j); show C;
transform T; T = (k,l,m,n,o,p); show T;
end
```

No more seg. fault with **mpost 2.000**, and we have the correct message error:

```
Preloading the plain mem file, version 1.005) (./test.mp
>> "2.000"
>> (xpart P,ypart P)
>> (redpart R,greenpart R,bluepart R)
>> (cyanpart C,magentapart C,yellowpart C,blackpart C)
! Missing `)' has been inserted.
<to be read again>
                    ,
l.5 transform T; T = (k,l,m,n,
                              o,p); show T;
?
```

## 4 Spurious `HEADER` in `tfm`

On Wed, Mar 8, 2017 Bogusław Jackowski and Piotr Strzelczyk found a bug on `tfm` files generated from MetaPost sources of Latin Modern. The bug occours from Metapost 1.211 (and higher, including the current version): in **mpost 1.9991**
`mpost "\generating:=1; \input lmro10.mp"`
gives

```
(FAMILY LMROMANSLANT10)
(FACE O 352)
(HEADER D 18 O 0)
(HEADER D 19 O 0)
(HEADER D 20 O 0)
(HEADER D 21 O 0)
(HEADER D 22 O 0)
(CODINGSCHEME QX ENCODING)
(DESIGNSIZE R 10.0)
(COMMENT DESIGNSIZE IS IN POINTS)
(COMMENT OTHER SIZES ARE MULTIPLES OF DESIGNSIZE)
(CHECKSUM O 31630546514)
```

It looks like a bug in the CWEB version. The PASCALWEB version initialize the `header_byte` array with -1, so it's easy to find the end when it's time to write it down to the header to the tfm file. CWEB uses an array of `chars` as byte stream initialized with \000, but given that it can contain \000 we cannot use \000 as end-of-array marker; we cannot use the trick of -1 of the PASCALWEB, because it's an array of chars. So we have to use a marker for `header_last` that keeps the last position, but currently it's always incremented and hence it doesn't consider the situation where the same field is written more than one time — as is in this case, the "font identifier" field is written 2 times: first

```
>> 49
>> "LMRoman10"
```

and then

```
>> 49
>> "LMRomanSlant10"
```

In **mpost 2.000** we have now

```
(FAMILY LMROMANSLANT10)
(FACE O 352)
(CODINGSCHEME QX ENCODING)
(DESIGNSIZE R 10.0)
(COMMENT DESIGNSIZE IS IN POINTS)
(COMMENT OTHER SIZES ARE MULTIPLES OF DESIGNSIZE)
```
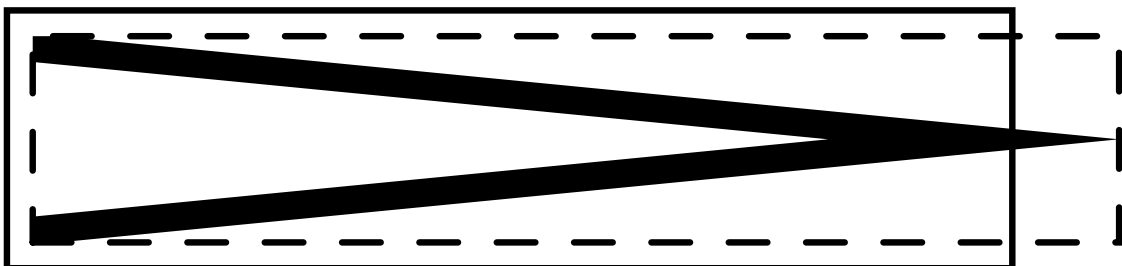
```
(CHECKSUM O 31630546514)
```

as it should be.

## 5 `boundingpath`

The miter-limitted paths can get wrong bounding boxes. A new primitive `boundingpath` returns the rectangular path of the bounding box of a path $p$ drawn with a pen $P$. It converts an elliptical pen to a polygonal one, then it calculates the envelope of the path with the pen and finally returns the rectangular cycle of the bounding box. This is an experimental primitive, so it's not still documented in the manual.

```
% Dan Luecking <luecking@uark.edu> provided an example showing that
% miter-limitted paths can get wrong bounding boxes:
picture k[];
path p[];
pen P[];
P0:= pensquare scaled 2bp;
  beginfig(1)
    miterlimit := 11;
    linejoin := mitered;
    p0 := (0,0)--(72,7)--(0,14) ;
    k1 := image ( draw p0 withpen P0 );
    draw k1;
    p1:= bbox k1;
    draw p1;
    p1:= boundingpath  P0 of  p0;
    draw p1 withcolor black dashed evenly ;
  endfig;
  end.
```

**mpost 2.000:**



The black dashed evenly path is the boundingpath of the path; the thick one is the bbox.

## 6 Weave files now compiles correctly

The weave files of the sources now compiles correctly with pdfTeX. They are:

```
mpmathbinary.pdf              mpstrings.pdf
mpmathdecimal.pdf             mpxout.pdf
mpmathdouble.pdf              pngout.pdf
mpmath.pdf                    psout.pdf
mpost.pdf                     svgout.pdf
mp.pdf                        tfmin.pdf
```

# 7 Backends separations for `mplib`

The compilations of the code now splits the Postscript backend from the PNG and
SVG backends and LuaTeX is now compiled with only the PostScript backend,
saving a bit of space. This doesn't affect MetaPost as program and `mplib` when
compiled as separated library.

Luigi Scarso
luigi.scarso@gmail.com
Wed, May 24, 2017