MetaPost update

Luigi Scarso

This is a short update on the status of solving a MetaPost bug on the envelope of a polygonal pen.

1. Current situation

The following code shows the errors that MetaPost makes in tracing a single path with a pen square:

```
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.
```

The result shows that several tracks are wrongs, and hints that the errors could be related to approximations around multiplies of 90°:

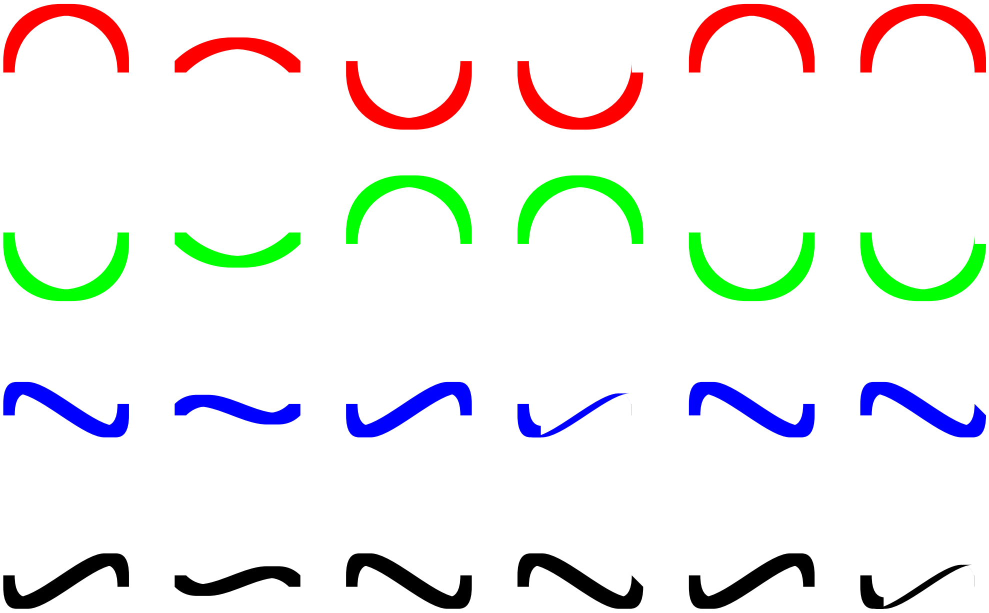

Figure 1: Test code.

Following this direction it was found that simplifications below are safe:

- `ab_vs_cd` (it checks $ab \;>=<\; cd$) is now straight in `decimal`, `double` and `binary`;
- in `decimal` and `double` mode the function `decimal_sin_cos` checks if the angle is $\pm 90°$,$\pm 180°$ and $\pm 270°$ (the `binary` mode seems to be safe) and uses the specific functions.

Quite lot of time was then spent trying to figure out how to make approximations that fixed the wrong cases using only the `eps` constant as *small number*, to be independent from the number systems. The idea was to use approximations only in these cases, hoping that there was a single bug behind each wrong track. While the first results was positives, additional tests shown that this not true.

2. Another bug

The next code shows a bug unrelated to the approximations:

```
prologues:=2;

pen T;
numeric a,b,i,dy,s,S,F;
string buildpen;
pair w[];
a:=45; b:=a;
i:=0; E:=-85; dy:=30;
s:=0.001; S:=10; F:=20;
pair V[];
path p[];
buildpen := "";
L:=3 ;
rotation:=30;
for i=0 upto (L-1):
  V[i] := 0.25*(cosd(i*360/L+rotation),
          sind(i*360/L+rotation)) scaled S;
  buildpen := buildpen & "V[" & decimal(i) & "] --";
endfor;
buildpen := buildpen & " cycle";
p[1000] := scantokens(buildpen);
T:= makepen (p[1000]);

beginfig(1);
p0 := (0,0){dir a} .. {dir b}(E,dy);
p3 := subpath(0,83.182518006/100) of p0;
draw image ( draw  p3 withpen T
   withcolor (0.6,0.6,0.6) ;) ;
if true:
  picture k;
  pair E[];
  draw p1000 withpen pencircle scaled 0.01;
  draw p1000 withpen pencircle scaled 0.01
    shifted (point 1 of p3);
  for i=0 upto (L-1):
    draw image( draw p3 withpen pencircle
      scaled s withcolor black ; ) shifted V[i];
  endfor;
  % Axes at t=0
  draw (0,0)--(0,10) withpen pencircle scaled s;
  draw (0,0)--(0,-10) withpen pencircle scaled s;
  draw (0,0)--(10,0) withpen pencircle scaled s;
  draw (0,0)--(-10,0) withpen pencircle scaled s;
  draw p3 withpen pencircle scaled s withcolor black;
  % Axes at t=1
  draw image (
    draw (0,0)--(0,10) withpen pencircle scaled s;
    draw (0,0)--(0,-10) withpen pencircle scaled s;
    draw (0,0)--(10,0) withpen pencircle scaled s;
    draw (0,0)--(-10,0) withpen pencircle scaled s;
  ) shifted (point 1 of p3);
fi;
currentpicture := currentpicture scaled F rotated 0;
endfig;
end.
```

The test is build around a triangular pen, and the path was accurately selected to exhibit the behavior shown in fig. 2. This case has three points where the derivative is vertical or horizontal, but in the end the problem was simply an oversight of the code: in the section *Find the first t where $d(t)$ crosses d_{k-1} or set t:=fraction_one+1;* two values were not negated, and the algorithm then jumped to a wrong offset of the pen (another similar oversight was fixed in the section *Make ss negative if and only if the total change in direction is more than* $180°$). This affect also the track in fig. 4, which was particularly hard to fix because it looks correct until one doesn't check the path for each offset of the pen: the solution in the end was found debugging in parallel two versions of MetaPost, the current 1.9991 release and an old 1.211 release that was correct and has the source code which is similar enough to compare the differences (of course the 1.211 release fails in the other cases as the 1.9991). The fig. 1 shows the test case without using the `eps` approximations and with the oversights fixed; in fig. 6 the current effect of approximations are shown (they emit a warning in the log). They are temporary patches, and they should disappear when a proper fix will be found.

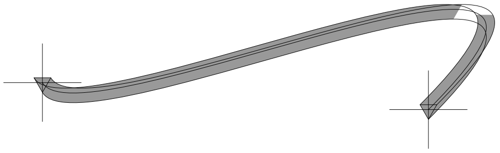

Figure 2: A bug not related to $90°$.

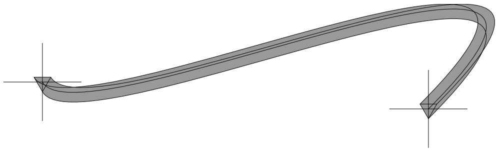

Figure 3: Correct envelope for fig. 2.

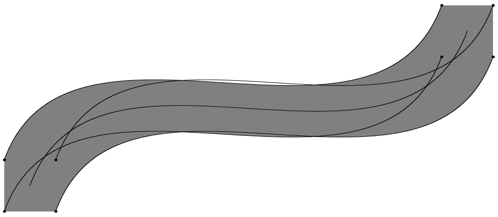

Figure 4: Different track, same bug as in fig. 2.

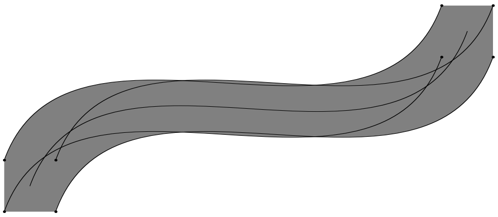

Figure 5: Correct envelope for fig. 4.

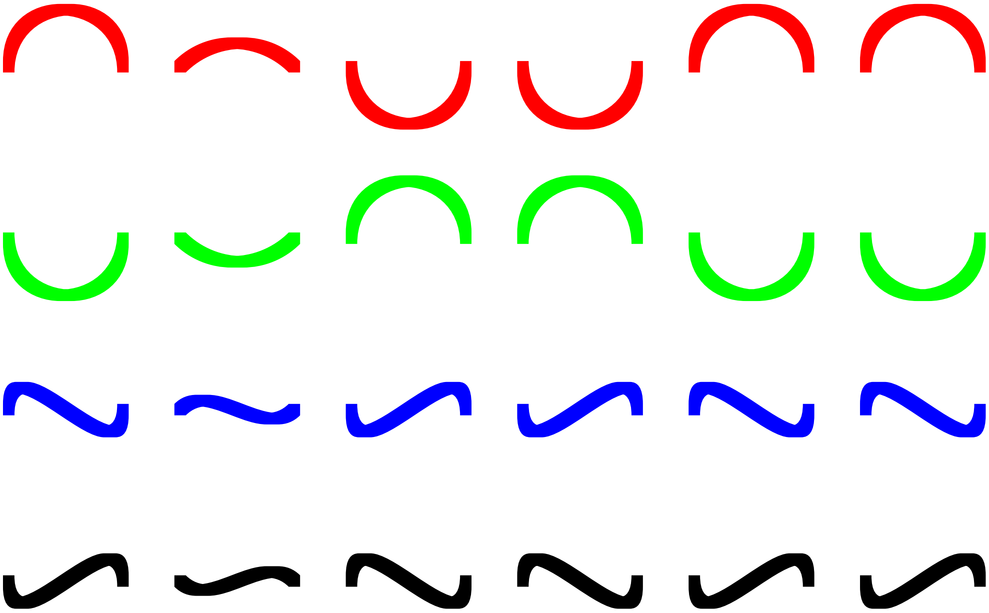

Figure 6: Test code with **eps** approximations.

3. Preliminary conclusion

At this point it's not possible to exclude that there are others oversights in the code, but at least the parallel debugging offers a way to a quick check. The test case of fig. 1 shows that there are still bugs from the old 1.211 release so investigation has to go on to isolate the next case.