MetaPost Developments – Spring 2005

Taco Hoekwater / MetaPost Team
metapost@tug.org

Abstract

The METAPOST system (by John Hobby) implements a picture-drawing language very much like that of METAFONT except that it outputs Encapsulated PostScript files instead of run-length-encoded bitmaps. METAPOST is a powerful language for producing figures for documents to be printed on PostScript printers, either directly or embedded in TeX documents. It includes facilities for directly integrating TeX text and mathematics with the graphics.

Introduction

The METAPOST[1] system by John D. Hobby implements a picture-drawing language very much like that of METAFONT, except that it outputs Encapsulated PostScript instead of bitmapped images. These graphics can be printed directly, or in embedded form from within TeX documents. METAPOST includes facilities for directly integrating TeX text and mathematics with the graphics.

In the summer of 2004, the version number of the METAPOST executable was still well below the 1.0 mark (0.641 was the current release at that time), but not much had happened in recent years. For some years, John Hobby simply could not find the time to solve the known bugs, let alone handle feature requests.

At that time, a group of people made a proposal to Dr. Hobby for the creation of a development team that would take care of the development of METAPOST from then on. He agreed, on the condition that he will only allow tested code to be inserted into the METAPOST distribution. Among the currently active group are the following people: Karl Berry, Giuseppe Bilotta, Hans Hagen, Taco Hoekwater, and Bogusław Jackowski. This list is not fixed, people can be added or removed when needed or desired.

Contact information

A home page for METAPOST has been created on the TUG server `http://www.tug.org/metapost`, and TUG also provides a mailing list for discussions and questions (`metapost@tug.org`). Details on subscription to the mailing list can be found on the home page. MetaPost development is currently hosted at sarovar.org; visit `http://www.sarovar.org/projects/metapost` for the current development team members, sources, and much else.

Please report bugs and request enhancements either on the `metapost@tug.org` list, or through Sarovar. (Please do not send reports directly to John Hobby any more.)

The current release

METAPOST is Public Domain software. Recently the manuals (mpman and mpgraph) have been released under a BSD-ish license. Dylan Thurston at Debian converted the sources to LaTeX, and from now on they will become a standard part of the distribution.

We are pleased to announce MetaPost 0.9 For those of you who have noted the large gap between 0.641 and 0.9, and the fact that version numbers disagree with the slides that were presented at EuroTeX:

We have deliberately chosen to step over some minor version numbers, such that the first major release by the current team (this release is planned for the autumn of 2005) can become version 1.0 without causing additional confusion.

Bug fixes

- Documentation improvements: all known errata and typos have been removed, better explanations of e.g. dash patterns and dotlabel have been provided, and a number of omissions has been rectified.
- The Bounding Box was not computed correctly when a filldraw command with a noticeable pen size was used at the edge of the picture.

[1] This article will be published in three different publications more or less simultaneously: In the proceedings of EuroTeX2005 and BachoTeX2005, as well as in issue 32 of the Maps. Parts of the contents of the starting page have already appeared in Maps 31, and some of the decisions documented by this report have not taken place until *after* EuroTeX, so this article is a bit of a cheat when it comes to chronology.

We apologise for that, but it makes more sense to write a single 'spring report' than to create 3 slightly different articles.

- Paths starting with degenerate constructions like in this example:

  ```
  draw (0,0)--(0,0)--(0,0)--...;
  ```

  could cause an overflow error of MetaPosts internal memory.
- The PostScript output could accidentally contain 8-bit characters within PostScript strings in previous versions because the `is_printable` test was shared between terminal printing and PostScript printing.
- A bug has been found in the assignment of serial numbers to independent variables in metafont 2.71828. This bug affected MP as well, and the same patch has been applied. Diagnose and patch were supplied by Thorsten Dahlheimer.
- The return value of the `turningnumber` primitive was sometimes wrong in unexpected ways (`turningnumber` is supposed to report whether a cyclic path turns clockwise or anticlockwise). The new implementation is still sometimes wrong when there are strange path segments involved, but in a much more predictable way: the new code always draws straight lines between the actual points, and calculates the turningnumber based on that path instead of the actual path. The effect is that cusps and loops within segments are now completely ignored. A more thorough fix of turningnumber is planned for the next release.
- There was an 'off by one' error in dvitomp, in the interpretation of virtual fonts.
- The `mpto` command (which is part of `makempx`) uses a new and improved TEX macro for the generation of labels, making it more robust with respect to strange user-supplied code within the actual label. The old version would sometimes loose track of the label's size information.
- A few macro bugs have been solved: a missing colon in `boxes.mp` that has been added, a missing save in `mfplain.mp` has been added, and the `generisize` macro in `boxes.mp` has been fixed so that it now accepts `[[` as a valid variable name.

mpversion Immediately after the presentation at EuroTEX, it became clear that both Hans Hagen and Donald Knuth have a considerable amount of pictures that more or less count on bugs in 0.641. For Knuth, this is the 'incorrect label size' that he himself reported as a bug to JDH. For Hans, it is the miscalculation of the picture's bounding box when using filldraw.

After some thinking, we decided it was best to add the already planned `mpversion` primitive right away, instead of waiting for a feature release. That way, this bug fix release can be identified from within the language. The result of `mpversion` is of type `<string>`, and those version strings are of the form `<major digit>.<minor digit>`. Possible trailing digits indicate beta releases (when the first extra digit is a nine), bug fix releases (when the first extra digit is a zero), or releases for development purposes only. For example, `"0.89"`, `"0.891"` and `"0.892"` were used to designate beta's for the current release.

For instance:

```
if known mpversion:
  message "mp = " \& mpversion;
  if mpversion > "1.0":
    message "time has flown by"
  fi
fi;
```

prints '`mp = 0.9`' on the terminal. (Incidentally, `>` does a simple ASCII comparison of strings; that works here, because of our particular version numbering—until and unless MetaPost reaches version 10!)

The version number is also included in the `Creator` comment in the PostScript output.

What is new

- It has been mentioned already that the LATEX sources of the `mpman`, `mpintro`, and `mpgraph` manuals have become part of the distribution package.
- There is a new internal string variable called `mpversion`, that reports the current MetaPost version, as explained above.
- The macro file `TEX.mp` acquired two additional routines to facilitate using LATEX to typeset labels: `TEXPRE` and `TEXPOST`. Their values are remembered, and included before and after (respectively) each call to `TEX`. Otherwise, each `TEX` call is effectively typeset independently. `TEX` calls also do not interfere with uses of `verbatimtex`. An example is given below.
- MetaPost now writes a `%%HiReSBoundingBox` comment to the PostScript output file. The values of the picture's bounding box in this comment are not rounded to integer values as the ones in `%%BoundingBox` are.
- The EPS output no longer contains actual spaces within PostScript strings. For example, the output of

  ```
  label("a space")
  ```

is now

```
(a\040space) cmr10 9.96265 fshow
```

instead of `(a space)`. This makes the generated PostScript easier to tokenize for post-processors.

- The EPS output now also has a `%%BeginProlog` DSC comment as well as `%%EndProlog`
- The comments in the Web source have been changed to point out that on modern machines, acquiring the random seed has actually become a system-dependant operation: a granularity of whole seconds is no longer small enough on new machines, where it is has become possible to start two separate mpost runs within one second.
- The 'newer' command now accepts more than two arguments. All of the supplied file arguments are tested in turn.

TEX.mp extension Here is an example of how to use the two new macros in `TEX.mp`, using the LaTeX inline math command `\( \)` instead of dollar signs:

```
input TEX;
TEXPRE("%&latex" &
       char(10) &
       "\documentclass{article}" &
       "\begin{document}");
TEXPOST("\end{document}");
beginfig(100)
  last := 10;
  for i := 0 upto last:
   label(TEX("\(n_{" & decimal(i) & "}\)"),
        (5mm*i,0));
  endfor
  ...
endfig;
```

- The `%&latex` causes LaTeX to be invoked instead of TeX. (See below, also.) Web2C- and MiKTeX-based TeX implementations, at least, understand this `%&` specification; see, e.g., the Web2C documentation for details, `http://tug.org/web2c`. (Information on how to do the same with other systems would be most welcome.)
- The `char(10)` puts a newline (ASCII character code 10, decimal) in the output.
- The `\documentclass...` is the usual way to start a LaTeX document.
- The `TEXPOST("\end{document}")` is not strictly necessary, due to the behaviour of `mpto`, but it is safer to include it.

Bugs remaining

A few bugs have not been solved yet. We promise that those will be addressed before the following (1.0) release.

web2c-specific problems These are usually caused by re-loading the mem file under different memory size settings than it was dumped with:

- strings reloading problems
- specials missing from the first output file

The solution is probably to save all of the configurable values inside the MEM file (like TeX does already for the format files).

Polygonal pen with 180 degree angles When a path segment turns exactly 180 degrees, MetaPost cannot decide which side of a polygonal pen to use on the 'return trip'.

```
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;
```

This happens because MetaPost cannot differentiate between an angle of 180 degrees and an angle of −180 degrees. The black line is what *should* be drawn, but the grey one is the one that current MetaPost actually outputs.

linecaps for polygonal pens MetaPost does no attempt to handle linecap for polygonal pens. This can produce unexpected results when the pen has a 'funny' shape, as can be seen in this picture:

```
beginfig(1);
linecap := butt;
draw (0,0)--(90,0) withpen
  pencircle scaled 10;
draw (0,20)--(90,20) withpen
```


Polska Grupa Użytkowników Systemu TeX, 2005 (http://www.GUST.org.pl)

```
 pensquare scaled 10 withcolor red;
endfig;
end
```

On the left side of the grey line, MetaPost switches from the top left corner of the square pen to the bottom right corner of the pen. This behaviour can be rationalised after some thinking, but at first sight, the produced diagonal line is a very strange phenomenon indeed.

graph.mp bugs There are a number of bugs in the `graph.mp` package. None of these have been tackled yet because in the current team nobody has used the package extensively. Most of these problems require familiarity with the macros and their usage:

- arrow heads are sometimes mis-rotated
- axis labels centre strangely
- data file format is too strict
- exponents cannot have a capital E, as in 1E4.
- curves from data are always polygonal
- axes can only be in the left of the graph, not through the origin.
- frames can only be avoided if an internal variable is set to false.

Future plans

The Birds of a Feather (special interest) gathering at EuroTEX had a slow start before dinner, followed by a much better attended continuation after dinner. This session was about all desired new features, both identification and prioritisation. We started with the list from the slides (and that list, in turn, was created by compiling the feature requests in the Sarovar database).

For the next release After a few hours of lively discussion, The top of the todo list became the following:

- Create an independent packaging system.

  This item was not really on anybody's wishlist, but it simply needs to be done. The current distribution depends on Web2C (it is, in fact, a stripped down snapshot of TEXLive) and we are far from happy with that. Even when we ignore the `troff` support, we feel that we should make the build process (how to compile MetaPost from source) as simple as possible.

  Taco intends to write a replacement for the current 'tangle + convert' construction. That replacement will output C code directly, and will maintain as much of the symbolic names as possible. In doing so, it will make source-level debugging easier.

- Remove `emergency_line_length` from the Web source.

  The web sources define a variable named `emergency_line_length`, with a compile-time value of 255, that is used solely to limit the maximum lines size of the PostScript output under unexpected conditions.

  The benefits of this check are marginal and it interferes with web2c's dynamic arrays considerably.

- Fix the still present bugs.

  As explained above.

- Allow access to the internally computed envelope of a path drawn with a polygonal pen.

```
<path>:= envelope <acyclic path> withpen <pen>;
<path>:= inner envelope <cyclic path> withpe...
<path>:= outer envelope <cyclic path> withpe...
```

  It has meanwhile been noted by several people that these actual names will not work, because MetaPost already assigns a meaning to `inner` and `outer`. The final implementation will therefore have to use different keywords. Those names are still under consideration.

  The implementation of this extension would be easy. Because MetaPost already does the needed work when writing the PostScript output file, it is just a matter of making this information available to the macro language.

In all likelihood, this is the limit of what we can put into the next release, version 1.0, to be released in the autumn of 2005.

Possibly for the next release

- Allow basic EPS inclusion.

  The idea is to allow the same kind of EPS inclusion one can find in the `dvips` driver. This EPS file could contain PostScript drawing commands or a bitmap expressed using PostScript's `image` operator.

- Implement something that is like TEX's `\special`.

  This is so that user-supplied stuff can end up in the middle of the PostScript output, not only at the very top. Both Jacko and Hans need this to do special effects in post-processing.

- Improvements to string handling.

  This was put on the board for Hans, who hopes that it is possible to improve the primitives that deal with strings.

- Implement edge structures.

  Edge structures is Metafont's data model to express images.

  This item is a huge undertaking, because it implies the recreation of all of Metafont's edge structure operations like "good.x", "withweight" and "cullit", and some of those in turn imply real unfill and overlap removal routines.

More wishes A number of items were kept for the future, but tagged as 'more thinking needed'. These are things for which a partial solution would be easy to do, but for a complete solution it is not even clear how it should be implemented

- Support for alternative colour models.

  Cmyk and grayscale would be easy but not very useful. Named colours and especially transparency would be very useful to have, but implementing that would be quite hard.

- Macro language enhancements

  The first item on this list is the desire to have namespace support. Everybody knows what is meant by this, but a precise specification is lacking.

  A second one of these is that symbolic token existence tests are missing from the language at the moment; `if known` is not always good enough. For instance, you may want to know if a variable name has been declared, regardless of whether or not it has a known value at the moment.

  Yet another is that `let` behaves a bit strange compared to the `\let` command in TEX.

- Support for 3D (or more dimensions).

  Perhaps this support can be added gradually, but that needs to be investigated in the Web source of MetaPost first.

Even more wishes Some ideas were tagged as '(much) later':

- virtual font support.

  Any code on this might become superseded within a year because of upcoming improvements to pdfTEX, so it makes no sense to do anything in this area yet.

- Proper EPSF creation.

  Embedded and possibly subsetted fonts should be included in the output.

- PDF output.

  This has to be very much later, because it needs lots of extra code.

- Bitmapped output format

  Some form of netpbm probably, to debug the edge structures. There is no sense in adding it before the edge structures are implement, because then it would need ghostscript or a similar PostScript rasterizer.

Remaining wishes

The rest of the list was agreed on as being nice, but remained unprioritized (by mistake or lack of time):

- A way to check if a path is completely within another path
- Non-continuous paths (as in PostScript)
- XML as an output format (nicer to parse for postprocessors)
- Fixed number arithmetic using 64 bits instead of the current 32 bits (also known as 'megapost').
- Unicode/UTF-8 support for easier string handling

Scratched ideas

Some ideas were scratched for the moment, either because it is strictly speaking unneeded (a macro solution is doable), or because there has not been a well-argumented request for that feature.

Among those are some shortcuts for existing macros and a switch to output PostScript level 2 or 3.

Acknowledgements

We would like to thank everybody who has helped us creating this new release of MetaPost, especially to John D. Hobby for allowing us to do so.

Our next report will be available in the autumn of 2005.