The MATHDOTS package[*]

Dan Luecking

2006/03/16

Abstract

This package provides vertical dots and diagonal dots in math, slanting in either direction. It should work equally well in LaTeX or plain TeX. It improves on the default definitions of plain TeX and LaTeX. Similar improvements are provided for the triple and quadruple dot accents of AMSTeX/LaTeX.

1 Introduction

Running LaTeX or TeX on `mathdots.ins` generates `mathdots.sty`, a LaTeX package, and `mathdots.tex` a plain TeX input file. If one puts `\input mathdots.sty` in a plain TeX file, it will cause `mathdots.tex` to be input. This is for backward compatibility.

Unlike the commands `\vdots` and `\ddots` provided by LaTeX, the versions defined by MATHDOTS change size with LaTeX size changing commands, as well as in {sub[sub] | super[super]}scripts. After MATHDOTS is loaded, `\Large$\ddots$` will be larger and `\small$\ddots$` will be smaller than `$\ddots$`. Moreover, `$2^{2^\ddots}$` will also produce a smaller symbol. This last example is very unlikely to be needed, but the version with the opposite slant is very common.

The default LaTeX (and plain TeX) definition changes the size of the dots, and the horizontal dimensions, but not the vertical dimensions. Using the default version in 'scripts produces symbols identical to those produced elsewhere. Finally, the default takes the dots from the current text font, whereas the MATHDOTS package takes them from the current math symbol font, so they should match other dots in math mode. This is only really a concern in `17pt` sizes or larger where `cmr17` is scaled to get text dots, but `cmsy12` is scaled to get math dots.

`\iddots` is similar to `\ddots`, except it produces the other diagonal. This command is not provided in basic TeX or LaTeX.

`\vdots` produces vertical dots (as in standard TeX or LaTeX), but again, the MATHDOTS version will change size with size changing commands.

The old behavior is stored in `\fixedddots` and `\fixedvdots`. If you wish `\iddots` to behave similarly then use `\fixediddots`.

---

[*]This file has version number v0.8, last revised 2006/03/16.

The `\fixed...` versions try to maintain the (awful) default behavior, which is to let the horizontal dimensions change size, as well as the dots themselves, but hard code unchanging vertical dimensions.

On the chance that some other package changes the behavior before MATHDOTS is loaded, use `\originaldddots`, `\originalvdots`, and `\originaliddots` to obtain the versions in force before the MATHDOTS is loaded.

If the LATEX and the AMSMATH package are detected, the commands `\dddot` and `\ddddot` are modified to get the same size changing behavior in 'scripts. They also correct a positioning bug in the AMSMATH code. The correction requires the command `\text`, and so is only applied if AMSMATH has been loaded *before* MATHDOTS. The old amsmath commands are saved in `\originaldddot` and `\originalddddot`. Similar comments apply to plain TEX with regard to `amstex.tex`.

In LATEX, all these commands have been made robust as of version 0.6.

2   The implementation

We define the version and date for plainTEX. The file `mathdots.sty` immediately switches to `mathdots.tex` if LATEX is not being used.

```
 1 ⟨tex⟩\def\mathdotsfiledate{2006/02/07}%
 2 ⟨tex⟩\def\mathdotsfileversion{0.7}%
 3 ⟨sty⟩\ifx\documentclass\UndEfInEd\input mathdots \expandafter\endinput\fi
```

The following are commands for plain TEX that prevent double loading, announce the package name, and change/restore the category of `@`.

```
 4 ⟨*tex⟩
 5 \expandafter
 6 \ifx\csname MathDots\mathdotsfileversion\endcsname\relax
 7 \else
 8   \immediate\write16{Mathdots already loaded.}\expandafter\endinput
 9 \fi
10 \immediate\write16{Package mathdots, \mathdotsfiledate\space version
11   \mathdotsfileversion.}%
12 \expandafter\edef\csname MathDots\mathdotsfileversion\endcsname{%
13     \catcode`\noexpand\@=\the\catcode`@}%
14 \catcode`\@=11
15 \def\@nameuse#1{\csname#1\endcsname}%
16 ⟨/tex⟩
```

Now various shorthands. For example '`\MD@us\MD@t`' becomes '`\textstyle`' and '`\MD@uf\MD@ss`' produces '`\scriptscriptfont 0`'.

```
17 \def\MD@us#1{\@nameuse{#1style}}%
18 \def\MD@uf#1{\@nameuse{#1font}0}%
19 \def\MD@t{text}\def\MD@s{script}\def\MD@ss{scriptscript}%
20 \newdimen\MD@unit \MD@unit\p@
```

The following are the main utility macros to implement changes of style. Since most of our symbols are built with boxes, they have to exit math mode and need `\mathchoice` so we know the current style (i.e., size) when we re-enter math mode.

We detect differences in sizes (for example, after `\large`) by examining the appropriate `\fontdimen` of family 0. Instead of the hard coded `1pt`, we use `0.1em` from this family.

We pass along the current style via `\everymath`.

In `\MD@palette`, `#1` is a command which we feed the current style name. For example, `\MD@palette\MD@ddots` will be our definition of `\ddots`. Then `\MD@ddots` will see the current style as its first argument.

```
21 \def\MD@changestyle#1{\relax\MD@unit0.1\fontdimen6\MD@uf{#1}%
22   \everymath\expandafter{\the\everymath\MD@us{#1}}}%
23 \def\MD@dot{$\m@th\ldotp$}%
24 \def\MD@palette#1{\mathchoice{#1\MD@t}{#1\MD@t}{#1\MD@s}{#1\MD@ss}}%
```

In the following commands, `#1` is the current style (supplied via `\MD@palette`). Except for the change in the unit used and the dot used, the first two definitions are essentially the same as plain's `\ddots` and `\vdots`, and the third is the same as the first with the oxes reversed.

```
25 \def\MD@ddots#1{{\MD@changestyle{#1}%
26   \mkern1mu\raise7\MD@unit\vbox{\kern7\MD@unit\hbox{\MD@dot}}%
27   \mkern2mu\raise4\MD@unit\hbox{\MD@dot}%
28   \mkern2mu\raise \MD@unit\hbox{\MD@dot}\mkern1mu}}%
29 \def\MD@vdots#1{\vbox{\MD@changestyle{#1}%
30   \baselineskip4\MD@unit\lineskiplimit\z@
31   \kern6\MD@unit\hbox{\MD@dot}\hbox{\MD@dot}\hbox{\MD@dot}}}%
32 \def\MD@iddots#1{{\MD@changestyle{#1}%
33   \mkern1mu\raise \MD@unit\hbox{\MD@dot}%
34   \mkern2mu\raise4\MD@unit\hbox{\MD@dot}%
35   \mkern2mu\raise7\MD@unit\vbox{\kern7\MD@unit\hbox{\MD@dot}}}}%
```

`\fixedddots`
`\fixedvdots`
`fixediddots`
`\originalvdots`
`\originalddots`
`\originaliddots`

We include current (2005/12/01) LaTeX/plainTeX definitions so a user can choose them. LaTeX and plainTeX don't have the other diagonal, so we provide a `\fixediddots` which is just `\fixedddots` with the boxes reversed.

We save the versions at load time so user can use them in case he loads MATH-DOTS for its other features. Chances are the original `\iddots` is undefined.

```
36 ⟨sty⟩\DeclareRobustCommand\fixedvdots{%
37 ⟨tex⟩\def\fixedvdots{%
38   \vbox{\baselineskip4\p@ \lineskiplimit\z@
39   \kern6\p@\hbox{.}\hbox{.}\hbox{.}}}%
40 ⟨sty⟩\DeclareRobustCommand\fixedddots{%
41 ⟨tex⟩\def\fixedddots{%
42   \mathinner{\mkern1mu
43   \raise7\p@\vbox{\kern7\p@\hbox{.}}\mkern2mu
44   \raise4\p@\hbox{.}\mkern2mu
45   \raise\p@\hbox{.}\mkern1mu}}%
46 ⟨sty⟩\DeclareRobustCommand\fixediddots{%
47 ⟨tex⟩\def\fixediddots{%
48   \mathinner{\mkern1mu
49   \raise\p@\hbox{.}\mkern2mu
50   \raise4\p@\hbox{.}\mkern2mu
51   \raise7\p@\vbox{\kern7\p@\hbox{.}}\mkern1mu}}%
```

```
52 \let\originalddots=\ddots
53 \let\originalvdots=\vdots
54 \let\originaliddots=\iddots
```

\ddots
\vdots
\iddots
Here are the actual (re)definitions of these three commands. The plainTEX version emits messages similar to that produced by LATEX's \DeclareRobustCommand.

```
55 ⟨sty⟩\DeclareRobustCommand\ddots{%
56 ⟨tex⟩\def\ddots{%
57     \mathinner{\MD@palette\MD@ddots}}%
58 ⟨sty⟩\DeclareRobustCommand\iddots{%
59 ⟨tex⟩\def\iddots{%
60     \mathinner{\MD@palette\MD@iddots}}%
61 ⟨sty⟩\DeclareRobustCommand\vdots{%
62 ⟨tex⟩\def\vdots{%
63     \mathinner{\MD@palette\MD@vdots}}%
64 ⟨tex⟩\wlog{Mathdots Info: Redefining \string\ddots.}%
65 ⟨tex⟩\wlog{Mathdots Info: Redefining \string\vdots.}%
```

Now we try to get AMS \dddot and \ddddot accents to behave.

\dddot
\ddddot
We add a \kern\z@ to prevent TEX from vertically centering #1. I don't know if the \text command is necessary, but it doesn't seem to hurt. I included it in the original version because that one used text dots and needed the text to change size. Now I use a math symbol (the dot accent) and mathpalette to implement the change. Still, I don't really understand how \ex@ works in the AMS packages, so I leave it in. For possible tuning, I have used two macros that represent the math spacing placed before and after the sequence of dots (\MD@prekern and \MD@postkern). Also, the actual dot is in the definition of \MD@D where another spacing command follows the dot (\MD@dotkern). In fact, by redefining \MD@D and possibly the spacing, one can define other double accent commands.

\originaldddot
\originaldddddot
We also save the original AMSMATH versions.

```
66 \newbox\MD@dotsbox
67 \newmuskip\MD@prekern
68 \newmuskip\MD@postkern
69 \newmuskip\MD@dotkern
70 \MD@prekern 0mu
71 \MD@postkern -1mu
72 \MD@dotkern -1.3mu
73 \def\MD@dotsaccent#1{\mathpalette{\MD@@dotsaccent{#1}}}%
74 \def\MD@@dotsaccent#1#2#3{%
75   {\setbox\MD@dotsbox\hbox{$#2\mkern\MD@prekern#1\mkern\MD@postkern$}%
76     \mathop{#3\kern\z@}\limits^{%
77     \text{\vbox to-1.4\ex@{\kern-1.8\ex@\copy\MD@dotsbox\vss}}}}}%
78 }%
79 \let\originaldddot \dddot
80 \let\originaldddddot\ddddot
81 ⟨*sty⟩
82 \@ifpackageloaded{amsmath}{%
83   \def\MD@D{\mathchar"5F\mkern\MD@dotkern}%
84   \DeclareRobustCommand\dddot {\MD@dotsaccent{\MD@D\MD@D\MD@D}}%
```

4

```
85    \DeclareRobustCommand\ddddot{\MD@dotsaccent{\MD@D\MD@D\MD@D\MD@D}}%
86  }{}%
87  ⟨/sty⟩
88  ⟨∗tex⟩
89  \def\amstexfmtname{AmS-TeX}%
90  \ifx\amstexfmtname\fmtname
91    \def\MD@D{\mathchar"5F\mkern\MD@dotkern}%
92    \wlog{Mathdots Info: Redefining \string\dddot.}%
93    \wlog{Mathdots Info: Redefining \string\ddddot.}%
94    \def\dddot {\MD@dotsaccent{\MD@D\MD@D\MD@D}}%
95    \def\ddddot{\MD@dotsaccent{\MD@D\MD@D\MD@D\MD@D}}%
96  \fi
97  \csname MathDots\mathdotsfileversion\endcsname
98  ⟨/tex⟩
99  ⟨/sty | tex⟩
```

Index

Numbers refer to the page where the corresponding entry is described.