

New PX font package

Michael Sharpe

March 29, 2013

1 Introduction

This package is meant to be a replacement for Young Ryu’s `pxfonts`—a complete text and math package with roman text font provided by a Palatino clone, sans serif based on a Helvetica clone, typewriter faces, plus math symbol fonts whose math italic letters are from a Palatino Italic clone. As with the related `txfonts` (though not as severe) the math metrics in `pxfonts` are overly tight.

The goal of this new project is use his glyphs along with a few additions and with completely reworked metrics which are generally looser, but not as loose as Computer Modern math. The following small examples (double normal size) provide some idea of the extent of the changes.

$r^k + \sum \int_0^\infty f(t)e^{\alpha t} dt$	$r^k + \sum \int_0^\infty f(t)e^{\alpha t} dt$
Math rendered with <code>pxfonts</code>	Math rendered with <code>newpxmath</code>

This new package differs from `pxfonts` in the following ways:

- the new package is split into separate text and math packages that do not need to be used in conjunction;
- both text and math packages offer options not present in the original package, described below;
- wide accent glyphs have been corrected (they should have zero depth) so that they no longer collide with the underlying glyph;
- the summation and product symbols in `pxfonts` seemed overly heavy at display size, and have been replaced by others of more suitable weights;
- for those who do not like the integral in `pxfonts`, an emboldened version of the Computer Modern integral is made available, matching the weight of the `pxfonts` symbols;
- an upright partial derivative symbol has been added, named `\uppartial`—∂;
- there is now an option to get braces more pleasing to older eyes;
- macros have been added to bring the calls to Greek symbols more into conformity with `PSNFSS` and `Mathtime Pro 2`;
- an upright Greek `\upvarkappa`, \varkappa , has been added as well as a matching italic version \varkappa ;

- problems using AMS macro packages before `pxfonts` are settled;
- `\coloneq` and `\eqcolon` now point to the correct glyphs;
- The problem with the `ogonek` accent and tabular environments (bad definition of `\k`) is fixed;
- The default encoding for `newpxtext` is now T1, but support is offered also for OT1 and LY1. As some add-on packages are available only in T1, that seems the best current choice.
- The font collection used for rendering text is based on TeXGyrePagella with additions to complete the superior figures, which are set by default to render footnote markers. (It is also possible to customize footnote markers by redefining `\thefootnote` after loading `newpxtext.sty`.) Small caps are available in all weights and styles.

2 Text mode options and macros

The text mode environment invoked by

```
\usepackage{newpxtext}
```

has two options: you may write

```
\usepackage[scaled=.95]{newpxtext}
```

to load the roman and typewriter text fonts at 95% of normal size, and the sans serif (Helvetica clone) at scale $0.95 * 0.93$. This is not of much utility if the package is used with the math package `newpxmath` to which it is already matched, but may be with other math packages. The option `osf` instructs the text fonts to use old-style figures 1234567890 rather than the default lining figures 1234567890.

The package contains a macro `\useosf` which is shorthand for

```
\renewcommand*{\rmdefault}{zplj}
```

which, when issued after loading the math package, resets the text mode to use oldstyle figures rather than lining figures.

3 Math mode options

The package invoked by

```
\usepackage{newpxmath}
```

loads the math part of the `pxfonts` (with revised metrics and additional glyphs) and should be loaded *after* the text font and its encoding have been specified, as it uses the text font settings to define how operators, numbers, math accents, `\mathrm`, `\mathbf` etc. are rendered. You should also load a Typewriter font so as not to generate mysterious error messages about `metafont` trying to generate `ectt10`. The package offers a number of options.

- – `varg` causes the math italic letter *g* to be replaced by *g*;

- `cmintegrals` instructs `newpxmath` to load a thicker version of the Computer Modern integral in place of the `newpxmath` default—the `pxfonts` integral (identical to the integral in the Wolfram fonts), which is not to everyone’s taste—a consequence is that none of the special forms of `pxfonts` integrals are available;
- The option `cmbraces` instructs `newpxmath` to ignore the brace collections from `pxfonts`, substituting a collection based on thickened versions of the Computer Modern braces, which I find much easier to distinguish from other delimiters. This works quite well in regular weight but looks a bit clunky in bold.

- the combination

```
\usepackage{amsmath}% loads amstext, amsbsy, amsopn but not amssymb
\usepackage{newpxmath}
```

causes no error, unlike the same combination with `pxfonts`, but does nothing significant. (Recall that `amsmath` is loaded automatically if you use an AMS document class such as `amstex` or `amsbook`, as is `amstex`.) The integrals are as defined in `pxfonts`. On the other hand,

```
\usepackage{amsmath}
\usepackage[cmintegrals]{newpxmath}
```

allows you to use the forms `\iint`, `\iiint`, `\iiiiiint` and `\idotsint` defined in `amsmath`, but using the pumped-up Computer Modern integral loaded by `newpxmath`.

- `uprightGreek` and `slantedGreek` determine the form of Greek alphabet loaded—the default is `uprightGreek`, which loads upright uppercase and slanted lowercase Greek symbols, as is customary in Anglo-American mathematical typesetting. With the option `slantedGreek`, which you might want to use if you cared about ISO standards, all Greek symbols are slanted. No matter which is set, `\Gammaup` (or `\upGamma`) gives you upright `\Gamma`, etc.
- The `newpxmath` package contains three different Blackboard Bold alphabets, where original `pxfonts` contained one. The default, triggered by `\mathbb{}`, takes its glyphs from the font which replaces `msbm` and has the same overall appearance of a hollowed-out text font, which I find neither bold nor blackboard-like. The second option, taken from `pxfonts`, is triggered by `\varmathbb{}`, is more geometric and, in my opinion, preferable but not optimal. The option `varbb` makes `\mathbb{}` synonymous with `\varmathbb{}`. The third option is the double-struck glyphs from the STIX collection. See the expanded discussion below.
- `nosymbolsc` causes the package to not load the `symbolsc` fonts, saving a math family. (This font contains mostly exotic symbols, along with some very useful, commonly used symbols like `\coloneq` \coloneqq , `\eqcolon` \eqcolon , `\notin` \notin , `\notni` \notni , `\neq` \neq , `\subset` \subset and `\supset` \supset , but these have been moved (virtually) to `lettersA` so they may continue to be used even if you use the option `nosymbolsc`.)
- `amssymbols` (the default) and `noamssymbols` determine whether the `pxfonts` versions of the AMS symbols (`msam`, `msbm`) are loaded—if so, they override previous settings in `amsmath`. If you use the option `noamssymbols`, then `\mathbb{}` is set to mean the same as `\varmathbb{}`. (One advantage of `noamssymbols` is that you save two of your precious math families for other purposes, such as setting a couple of external math alphabets by means of the `mathalfa` package.)

EXAMPLE:

```
\usepackage{newpxtext} % T1, lining figures so math uses lf
\usepackage{textcomp} % required for special glyphs
\usepackage{amsmath}
\usepackage[varg,cmintegrals,cmbraces]{newpxmath}
\usepackage[scr=rsfso]{mathalfa}% \mathscr is fancier than \mathcal
\usepackage{bm} % load after all math to give access to bold math
\useosf %old-style figures in text, not in math
\linespread{1.05}% Give Palatino more leading (space between lines)
```

Caution: In encodings T1 and LY1, the macro `\mathrm` does not produce what you might expect for Greek letters, due to the difference between those encodings and OT1. But what were you expecting with `\mathrm{\Gamma}` anyway? In T1 and LY1, you'll get a math accent.

4 The T_EX math font problem

Math font packages in L^AT_EX are susceptible to the “Too many math alphabets” error, due to exceeding the limit of just sixteen math font families, or mathgroups, as they are called in L^AT_EX. Put in oversimplified terms that do however correctly represent how this all works in newpxmath, the following math fonts are always loaded and permanently (if you didn't prevent loading of some features) occupying slots immediately following `\begin{document}`, and others that may be called for in typesetting a mathematical expression can add to the list as the document grows.

ALWAYS LOADED:

```
0: operators
1: letters
2: symbols
3: largesymbols
4: AMSa
5: AMSb
6: lettersA
7: symbolsC
8: largesymbolsA
9: boldoperators
10: boldletters
11: boldsymbols
12: boldlargesymbols
```

NOTES:

- The operators font is essentially the Roman text font, used for names of operators and as the target for `\mathrm`—its bold version is used as the target for `\mathbf`;
- if you typeset an expression that, say, calls for a single bold glyph from symbolsC, that costs you an entire new slot, leaving only two remaining;
- same with, eg, `\mathit`;
- same with an external Fraktur, Blackboard Bold or Script glyphs;

- if you run out of math alphabets, look first to dropping AMSa and AMSb, as well as symbolsC, which can save you three slots;
- if space is tight, do not call bold versions of the fonts listed above where the bold version is not already loaded, to avoid loading a new mathgroup;
- if you absolutely need a letter (not a math symbol) from some math font that would normally cost you another mathgroup, you might consider using as if it were text, with something like

`\mbox{\usefont{U}{zplmia}{b}{n} X}`

which allows you to use letter X from boldlettersA but without any math features;

- there is a macro `\ShowMathFonts` in `newpxmath` which may be called at any point in your document, which will provide you a list of the mathgroups currently in use. This can be helpful in figuring out where problems occur. The output lines take the form

`(<fam number>: <internal font id> = <tfm name> [newpx name])`

5 Bold math fonts

Every math font in the `pxfonts` package, and in the `newpx` package, is accompanied by a bold version. Some usage examples are given below. Generally, one may use either `\boldmath` to change an entire formula to bold, or `\boldsymbol` to change one symbol, but the spacing generally works better after loading the `bm` package and using the `\bm` macro.

The text glyphs `\dotlessi` (`\i`) and `\dotlessj` (`\j`) are rarely needed in actual text—in many fonts, roman `\dotlessi` is very similar to the numeral 1. They are however sometimes needed to build special math glyphs. The following table shows how to generate the mathematical forms of `\dotlessi` and `\dotlessj`. I illustrate with only `\dotlessi`—`\dotlessj` is entirely analogous.

Type	Weight	Command	Result
Math Italic	Regular	<code>\$_\hat{\imath}\$</code>	\hat{i}
Math Italic	Bold	<code>\$_\bm{\hat{\imath}}\$</code>	$\hat{\mathbf{i}}$
Roman	Regular	<code>\$_\hat{\textrm{i}}\$</code>	\hat{i}
Roman	Bold	<code>\$_\bm{\hat{\textbf{i}}}\$</code>	$\hat{\mathbf{i}}$

In math, bold roman characters are often used to indicate vector quantities, and for this one uses constructions like

- `$_\mathbf{x}$` produces \mathbf{x} ;
- `$_\bm{\mathrm{x}}$` produces \mathbf{x} (same as previous but may offer improved spacing);
- `$_\bm{x}$` produces \mathbf{x} (not roman);
- `$_\bm{\hat{\mathbf{x}}}$` produces $\hat{\mathbf{x}}$ (with a bold accent);
- `$_\bm{\dot{\mathbf{x}}}$` produces $\dot{\mathbf{x}}$ (bold but not roman);
- the macro definition

`\newcommand{\xyvec}[2]{\ensuremath{\#1\bm{\textbf{i}}\#2\bm{\textbf{j}}}}`

allows you to write `\xyvec{x}{+y}` to output $x\mathbf{i} + y\mathbf{j}$;

- `\bm{\Gamma}` produces Γ (bold Gamma);
- `\bm{\hat{\Gamma}}` produces $\hat{\Gamma}$ (bold Gamma with bold accent).

(The last two assume that you have effectively set the option `uprightGreek`—the default.)

6 Blackboard bold

As mentioned briefly above, the package now has a new blackboard bold alphabet built-in, and some new macros to call the non-default versions. To summarize, the problems are:

- the default, called by `\mathbb` is quite ugly and indistinct, but does cooperate with the bolding macros `\bm` and `\boldsymbol`;
- the original variant form called with the macro `\varmathbb` is better, but the macro conflicts with `\bm`. (In fact, the bold version is identical to the regular version, but it is still not right that it conflicts with them.) The problem stems from the part of the definition of `\varmathbb` which allows you to insert an argument with more than one character, like `\varmathbb{ABC}`. The solution is to make a new macro that takes a single argument, like `\vmathbb{A}`, which cooperates with `\bm`.
- the third, new, alphabet is borrowed from the STIX fonts—it is sharp and quite clear, geometric in design. The only slight problem is that it lacks a full bold version, with only the letters C, D, H, N, P, Q, R, Z, d, e, i and j available. The macro `\vmathbb` provided to access this alphabet allows only a single character for its argument. The new option `vvarbb` effectively makes `\mathbb` mean the same as `\vmathbb`. (The reason for including these in the `newpx` package rather than calling them from the `mathalfa` package is that `newpx` leaves very little space for new math symbol fonts and math alphabets, and this way requires no additional such resources.

One interesting feature of the new alphabet is that it contains blackboard bold numbers, of which 0 and 1 will likely be the most useful, perhaps as operator names. I find `\vmathbb{1}` useful in specifying an indicator, AKA characteristic function. Here are some examples:

	Regular	Bold	Remarks
Default	<code>\mathbb{R}</code> \mathbb{R}	<code>\bm{\mathbb{R}}</code> \mathbb{R}	
Variant 1	<code>\varmathbb{R}</code> \mathbb{R}		Bold is same as regular
...or	<code>\vmathbb{R}</code> \mathbb{R}		Single char. argument
Variant 2	<code>\vmathbb{R}</code> \mathbb{R}	<code>\boldmath \vmathbb{R}</code> \mathbb{R}	Make a macro for this!