

New PX font package

Michael Sharpe

March 11, 2017

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- the summation and product symbols in `pxfonts` seemed overly heavy at display size, and have been replaced by others of more suitable weights;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 a number of additions, denoted by TeXGyrePagellaX. The superior figures in this addition 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, and are offered in two sizes. The default small caps supplied by TeXGyrePagella are really PETITE CAPS, having xheights approximately the same x-heights (sometimes smaller) as lowercase letters. Option `largesc` increases the size of small caps by about 8%, approximating the size of Adobe Palatino Small Caps. IMO, this is a better match in terms of weight and size. For a side by side comparison: PETITE CAPS, SMALL CAPS.
- Sans serif is by default taken from TeXGyreHeros, and by default at 94% of the scale factor (set by `scaled`, default value 1). The option `helvratio=.98` will change that to 98%.
- New math accents such as `\widearc` have been introduced in tandem with the `newtx` package, where they are documented.

2 Text mode options and macros

The text mode environment invoked by

```
\usepackage{newpxtext}
```

has several 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 $0.95 * 0.94$. 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 options

```
\usepackage[scaled=.95,helvratio=.96]{newpxtext}
```

load roman and typewriter text fonts at 95% of normal size, and the sans serif (Helvetica clone) at scale $0.95 * 0.96$.

The option `osf` instructs the text fonts to use old-style figures 1234567890 rather than the default lining figures 1234567890. As of version 1.23, `newpxtext` loads initially with lining figures so the math package uses lining figures in math mode. The option `osf` switches to old-style figures in text at the very end of the preamble, forcing the use of oldstyle figures in text, but not math. In

previous versions, it was necessary to run `\useosf` after loading `math`. This is no longer required, but does no harm. See the discussion in section 4 for further details.

As described above, option `largesc` increases the size of small caps by about 8%.

Option `defaultsup` (same effect as `defaultsup=true`) forces the package to use the L^AT_EX default footnote markers (or, at least, those in force when the package is loaded) instead of preferred by the package—Palatino (clone) superior figures instead of spindly ordinary Palatino lining figures reduced to about 70%. (Footnote markers in minipages use the default lowercase alphabetic characters, unless otherwise specified by redefining `\thempfootnote`.) For better control over position and size of footnote markers, use the `superiors` package after loading `newpctext`.

The `theoremfont` option changes the default font used for the plain theorem style of `amsthm`, keeping italic text but substituting upright figures and punctuation. For example, with this option, you get theorem statements like this:

Theorem 2.1. *This is Theorem Italic: text numbers are upright—12345; punctuation is in many cases upright (also, parens, braces {} and brackets []). What about question marks and exclamations? Also upright! [These fit better with math mode punctuation and figures, like: for all $x \in [0, 1]$, let $f(x) := \exp(\alpha x)$].*

Compare this to traditional plain theorem style of the same text:

Theorem 2.1. *This is Theorem Italic: text numbers are upright—12345; punctuation is in many cases upright (also, parens, braces {} and brackets []). What about question marks and exclamations? Also upright! [These fit better with math mode punctuation and figures, like: for all $x \in [0, 1]$, let $f(x) := \exp(\alpha x)$].*

If you are using another theorem package (e.g., `ntheorem`, `theorem`) you will have to add your own descriptors as specified in the its documentation and set the body font to `\slshape`.

Superior letters and figures may be called with either `\sustyle ...` or `\textsu{...}`, so you can emulate 18th century orthography such as Jos W^m Smith, or print French style with , e.g., 1^{ière}, M^{me} Dubois or M^{lle} D’Orleans.

The next two sections describe options to `newpctext` of more specialized nature.

3 Spacing issues

This new version of `newpctext` has spacing that is a little different, in its default state, from that of the old `newpctext`. In small part this is due to the finer kerning of TeXGyre Pagella, but mostly because the three parameters that govern inter-word spacing are not the same.

	<code>pxfonts</code>	Pagella
<code>fontdimen2</code> (interword space)	.25em	.25em
<code>fontdimen3</code> (interword stretch)	.125em	.2em
<code>fontdimen4</code> (interword shrink)	.08333em	.1em

That is, Pagella has the same normal spacing as `pxfonts` but its spacing is more flexible in terms of both stretch and shrink. More frequently than not, a paragraph built with Pagella will occupy more space than the same built with `pxfonts`. For this reason, the package offers some ways to

change the spacing parameters. This may be important if you are trying to imitate the pagination of a document built using `pxfonts`.

Option `tighter` sets the three `fontdimen` values to those of `pxfonts`, except with a little more shrink. This should make it unlikely that text will occupy more space than it would have using `pxfonts`.

Option `looser` sets the three `fontdimen` values to `{.3em,.2em,.1em}` respectively.

If you want full control, the options `spacing`, `stretch`, `shrink` allow you to modify one or more of the above `fontdimens`. For example,

```
\usepackage[stretch=.15em,shrink=.095em]{newpxtext}
```

4 Usage with `babel`

You should normally load `babel` before loading `newpxtext` in order for `babel` to function as expected. A similar macro `\useproportional` makes proportional figures the default outside math mode. For example:

```
\usepackage[greek.polutonico,english]{babel}
% the next line makes text figures proportional, oldstyle, while math uses lining figures
\usepackage[theoremfont,largesc,tighter,p,osf]{newpxtext}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage[varqu,varl]{inconsolata}
\usepackage{amsmath,amsthm}
\usepackage[bigdelims,vvarbb]{newpxmath}
% option vvarbb gives you stix blackboard bold
\linespread{1.05}
```

5 Usage with `LuaATEX` and `XeATEX`

As far as I can tell, `newpxmath` works with both, but requires a very specific loading order and choice of options. Briefly, the math options must all be loaded prior to loading and using `fontspec`. Be aware that some text packages (e.g., `cabin`) may contain a line like

```
\RequirePackage{fontspec}
```

which would prevent (“option clash” error) a subsequent

```
\usepackage[no-math]{fontspec}
```

unless suppressed by an appropriate option. E.g.,

```
\usepackage[type1]{cabin}
```

prevents the problem with the `cabin` package.

While the math font options must be specified before `fontspec`, be aware of a potential trap. Using `\usepackage{newpxtext}` before `\usepackage{newpxmath}` results in `newpxtext`'s font-loading options being run after all other packages in the preamble, so instead of `\usepackage{newpxtext}`, use the following: EXAMPLE:

```
%load LaTeX text components before math math
\usepackage[T1]{fontenc}
\renewcommand{\rmdefault}{zpltlf}% Roman font for use in math mode
\usepackage[scaled=.85]{beramono}% used only by \mathtt
\usepackage[type1]{cabin}% used only by \mathsf
\usepackage{amsthm}% load before newtxmath
\usepackage[bigdelims,vvarbb]{newpxmath}
\usepackage[scr=rsfs]{mathalfa}
\usepackage{bm}% load after all math to give access to bold math
%Now load the otf text fonts using fontspec---won't affect math
\usepackage[no-math]{fontspec} % process with XeLaTeX or LuaLaTeX
\setmainfont{TeXGyrePagellaX} % this reads in TeXGyrePagellaX.fontspec
```

6 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.

- `upint` (new as of version 1.3.) selects upright integrals—the default shape is slanted. Each shape/size of integral takes one of twelve form, illustrated below in the case of display size slanted integrals.

$\int \oint \iint \iiint \iiiiiint \int\!\!\!\int \int\!\!\!\!\int \int\!\!\!\!\!\int \int\!\!\!\!\!\!\int \int\!\!\!\!\!\!\!\int \int\!\!\!\!\!\!\!\!\int \int\!\!\!\!\!\!\!\!\!\int \int\!\!\!\!\!\!\!\!\!\!\int \int\!\!\!\!\!\!\!\!\!\!\!\int$

named respectively

```
\int \oint \iint \iiint \iiiiiint \oiint
\oiiiiint \varointclockwise \ointctrclockwise \fint \sumint \sqint
```

The three sizes of the upright integrals look like:

Glyph	Command
\int	<code>\smallint[up]</code>
\int	<code>\int[up]</code>
\int	<code>\displaystyle{\int[up]}</code>

Note that the suffix `up` is not required unless the document’s integral style is slanted. You may find the `\smallint` is useful for inline math mode when it is important not to change the line spacing.

- `smallerops` (new as of version 1.3.) causes big operators other than integrals to render up to 20% less tall, so that displayed formulas may occupy less vertical space. For example, in the following display, the first operator is the usual `\sum`, the second is what you would get with `smallerops`, the third is `\textstyle\sum` and the fourth is `\smallsum`, the latter being used mainly with inline math.

$$\sum \sum \sum \Sigma$$

Similarly, there are `\smallprod` and `\smallcoprod` which, along with `\smallsum`, are of class `mathop`, unlike their Greek letter equivalents.

- (New as of version 1.3.) Two macros allow you to change `fontdimen` values in math mode: `\setSYdimens` and `\setEXdimens`, which allow you to change the `fontdimen` parameters for the symbol and extension fonts respectively. They may be used only in your preamble. Their arguments can be any valid T_EX commands to change `fontdimen` values. For example:

```
\def\setSYdimens{\fontdimen16\font=2pt\fontdimen17\font=1.15\fontdimen17\font }
```

Don’t use these unless you know what you’re doing.

- `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; **As of version 1.3, this option does nothing, as the new default is slanted integrals.**
- 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.

NEW The new option `bigdelims` offers delimiters which are a bit larger than the standard delimiters and the normal and big sizes, with more distinction between the two than in the standard package. With `bigdelims`, the option `cmbraces` is ignored.

- 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. The package `newpxmath` loads the package `amsmath` if it was not previously loaded. Options to `amsmath` such as `leqno`, `intlimits` may be passed to `amsmath` via options to the documentclass.

- `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 Γ , etc, and `\Deltait`, `\zetaait` give you italic (i.e., slanted) versions of those letters, and `\mathnormal{\Omega}` etc will always produce the slanted version of uppercase Greek letters. (The macro `\mathnormal` means essentially “use the version of the symbol in letters”—i.e., the math italic form. This did not always work as expected in versions prior to 1.27.)

- 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 `\nsupset` \notsupset , 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 one of your precious math families for other purposes, such as setting a couple of external math alphabets by means of the `mathalfa` package.)

EXAMPLE:

```
\documentclass[leqno]{article}
\usepackage[osf,theoremfont]{newpxtext} % T1, lining figures in math, osf in text
\usepackage{textcomp} % required for special glyphs
\usepackage[varg,bigdelims]{newpxmath}
\usepackage[scr=rsfso]{mathalfa}% \mathscr is fancier than \mathcal
\usepackage{bm} % load after all math to give access to bold math
%\useosf %no longer needed unless using babel
\linespread{1.05}% Give Palatino more leading (space between lines)
```

7 The \TeX math font problem

Math font packages in \LaTeX 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 \LaTeX . 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: AMSm (replaces former AMSa and AMSb)
5: lettersA
6: symbolsC
7: largesymbolsTXA
8: boldoperators
9: boldletters
10: boldsymbols
11: 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, e.g., `\mathit`;
- same with an external Fraktur, Blackboard Bold or Script glyphs;
- if you run out of math alphabets, look first to dropping `AMSm` and `symbolsC`, which can save you at least two 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])`

8 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 `\i` and `\j` are rarely needed in actual text—in many fonts, roman `\i` 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 `\i` and `\j`. I illustrate with only `\i`—`\j` 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{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.)

9 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 `\vvmathbb` provided to access this alphabet allows only a single character for its argument. The new option `vvarbb` effectively makes `\mathbb` mean the same as `\vvmathbb`. (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 `\mathbb{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>\vvmathbb{R}</code> \mathbb{R}	<code>\boldmath \vvmathbb{R}</code> \mathbb{R}	Make a macro for this!

For further information on the changes made to `newpxmath` 1.3, see the appendix to the documentation for `newtxmath` 1.5.