Progress in the \TeX\ Gyre Math Font Project

Bogusław Jackowski, Piotr Strzelczyk, Piotr Pianowski

Bachotek, Poland, 1–5 V 2013
Progress in the TeX Gyre Math Font Project
or assembling a hard puzzle

Bogusław Jackowski, Piotr Strzelczyk, Piotr Pianowski
Bachotek, Poland, 1–5 V 2013
OTF Math font components

Let’s start, traditionally, from recalling that according to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain several scripts (alphabets) and groups of symbols:
sans serif, calligraphic, double struck (aka blackboard bold), fraktur, typewriter (monospace), Greek symbols, Hebrew symbols, geometrical symbols, maths symbols etc.
OTF Math font components

Let’s start, traditionally, from recalling that according to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain several scripts (alphabets) and groups of symbols:

- sans serif
- calligraphic
- double struck (aka blackboard bold)
- fraktur
- typewriter (monospace)
- Greek symbols
- Hebrew symbols
- geometrical symbols
- maths symbols etc.

\[\text{ABCD/ABCDEFGHIJKLMNOPQRSTUVWXYZ}\]
\[\text{αβγδεζηθ} \left\{ \left( \int_\Phi \to \clubsuit \right) \right\} \]
\[\text{αβγδεζηθ} \left\{ \left( \int_\Phi \to \clubsuit \right) \right\} \]
\[\text{αβγδεζηθ} \left\{ \left( \int_\Phi \to \clubsuit \right) \right\} \]
\[\text{αβγδεζηθ} \left\{ \left( \int_\Phi \to \clubsuit \right) \right\} \]
OTF Math font components

Let’s start, traditionally, from recalling that according to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain several scripts (alphabets) and groups of symbols:

- sans serif
- calligraphic
- double struck (aka blackboard bold)
- fraktur
- typewriter (monospace)
- Greek symbols
- Hebrew symbols
- geometrical symbols
- maths symbols etc.

\[ \text{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \alpha \beta \gamma \delta \varepsilon \{ [(\int \rightarrow \clubsuit)] \} \]
\[ \text{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \alpha \beta \gamma \delta \varepsilon \{ [(\int \rightarrow \diamondsuit)] \} \]
\[ \text{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \alpha \beta \gamma \delta \varepsilon \{ [(\int \rightarrow \heartsuit)] \} \]
\[ \text{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \alpha \beta \gamma \delta \varepsilon \{ [(\int \rightarrow \spadesuit)] \} \]
OTF Math font components

From the point of view of a math font maker, glyphs can be divided into several groups depending on the amount of work needed to prepare such a group (this “classification” only partially depends on the number of glyphs of a given kind); moreover, the font parameters, controlling the shapes of certain glyphs and their behavior in math formulas, are to be somehow adjusted.
OTF Math font components

From the point of view of a math font maker, glyphs can be divided into several groups depending on the amount of work needed to prepare such a group (this “classification” only partially depends on the number of glyphs of a given kind); moreover, the font parameters, controlling the shapes of certain glyphs and their behavior in math formulas, are to be somehow adjusted

1 Alphabets excerpted from the relevant (basic) text font family, presumably nearly ready
OTF Math font components

From the point of view of a math font maker, glyphs can be divided into several groups depending on the amount of work needed to prepare such a group (this “classification” only partially depends on the number of glyphs of a given kind); moreover, the font parameters, controlling the shapes of certain glyphs and their behavior in math formulas, are to be somehow adjusted.

1. Alphabets excerpted from the relevant (basic) text font family, presumably nearly ready

2. Alphabets borrowed from other text fonts or relevant font variants (beware of copyrights!)
OTF Math font components

From the point of view of a math font maker, glyphs can be divided into several groups depending on the amount of work needed to prepare such a group (this “classification” only partially depends on the number of glyphs of a given kind); moreover, the font parameters, controlling the shapes of certain glyphs and their behavior in math formulas, are to be somehow adjusted

1. Alphabets excerpted from the relevant (basic) text font family, presumably nearly ready
2. Alphabets borrowed from other text fonts or relevant font variants (beware of copyrights!)
3. Programmed symbols
OTF Math font components

From the point of view of a math font maker, glyphs can be divided into several groups depending on the amount of work needed to prepare such a group (this “classification” only partially depends on the number of glyphs of a given kind); moreover, the font parameters, controlling the shapes of certain glyphs and their behavior in math formulas, are to be somehow adjusted.

1. Alphabets excerpted from the relevant (basic) text font family, presumably nearly ready
2. Alphabets borrowed from other text fonts or relevant font variants (beaware of copyrights!)
3. Programmed symbols
4. “Hybrid” symbols, i.e., symbols manually tuned and programmed
OTF Math font components

From the point of view of a math font maker, glyphs can be divided into several groups depending on the amount of work needed to prepare such a group (this “classification” only partially depends on the number of glyphs of a given kind); moreover, the font parameters, controlling the shapes of certain glyphs and their behavior in math formulas, are to be somehow adjusted

1. Alphabets excerpted from the relevant (basic) text font family, presumably nearly ready
2. Alphabets borrowed from other text fonts or relevant font variants (beware of copyrights!)
3. Programmed symbols
4. “Hybrid” symbols, i.e., symbols manually tuned and programmed
5. Manually prepared symbols
From the point of view of a math font maker, glyphs can be divided into several groups depending on the amount of work needed to prepare such a group (this “classification” only partially depends on the number of glyphs of a given kind); moreover, the font parameters, controlling the shapes of certain glyphs and their behavior in math formulas, are to be somehow adjusted.

1. Alphabets excerpted from the relevant (basic) text font family, presumably nearly ready.
2. Alphabets borrowed from other text fonts or relevant font variants (beware of copyrights!)
3. Programmed symbols
4. “Hybrid” symbols, i.e., symbols manually tuned and programmed
5. Manually prepared symbols
6. Adjusting parameters – may have an impact on any of the above mentioned groups, actually this process never ends as there is no unique set of optimal parameters.
Basic alphabets

Excerpting glyphs from the relevant (basic) text font family is a relatively easy task, as the sources were prepared by us and, thus, we roughly know what can be expected, provided a particularly nasty bug does not show up.
Basic alphabets

Excerpting glyphs from the relevant (basic) text font family is a relatively easy task, as the sources were prepared by us and, thus, we roughly know what can be expected, provided a particularly nasty bug does not show up.

AĄBCĆaąbcć/ABCabcABCabc
AĄBCĆaąbcć/ABCAbcaBcabc
AĄBCĆaąbcć/ABCAbcaBcabc
AĄBCĆaąbcć/ABCAbcaBcabc
AĄBCĆaąbcć/ABCAbcaBcabc
AĄBCĆaąbcć/ABCAbcaBcabc
Borrowed alphabets

Borrowing alphabets from other text fonts or relevant font variants is also fairly simple, although surprises lurks here and there, e.g., “unorthodox” glyph names; needless to say, the problem of copyrights is not negligible – usually we have to ask the authors for granting a permission.
Borrowed alphabets

Borrowing alphabets from other text fonts or relevant font variants is also fairly simple, although surprises lurks here and there, e.g., “unorthodox” glyph names; needless to say, the problem of copyrights is not negligible – usually we have to ask the authors for granting a permission
Borrowed alphabets

Borrowing alphabets from other text fonts or relevant font variants is also fairly simple, although surprises lurks here and there, e.g., “unorthodox” glyph names; needless to say, the problem of copyrights is not negligible – usually we have to ask the authors for granting a permission.
Programmed symbols

The programming of symbols is the most pleasant part of the job: the lion share of the work is done once for ever.
Programmed symbols

The programming of symbols is the most pleasant part of the job: the lion share of the work is done once for ever
Programmed symbols

The programming of symbols is the most pleasant part of the job: the lion share of the work is done once for ever.
“Hybrid” symbols

As a good example of a hybrid symbol may serve an integral symbol: its top, at least in the TeX Gyre fonts, should resemble (somehow) the head of the italic form of the letter long s which, in turn, resembles the top of the letter italic f.

The long s for denoting an integral operation (infinite summation) was introduced by Gottfried Wilhelm Leibniz in 1675 in an unpublished paper *Analyseos tetragonisticæ pars secunda* (Second part of analytical quadrature).
“Hybrid” symbols

As a good example of a hybrid symbol may serve an integral symbol: its top, at least in the \TeX\ Gyre fonts, should resemble (somehow) the head of the italic form of the letter long s which, in turn, resembles the top of the letter italic f

The long s for denoting an integral operation (infinite summation) was introduced by Gottfried Wilhelm Leibniz in 1675 in an unpublished paper *Analyseos tetragonisticæ pars secunda* (Second part of analytical quadrature)
“Hybrid” symbols

As a good example of a hybrid symbol may serve an integral symbol: its top, at least in the \TeX{} Gyre fonts, should resemble (somehow) the head of the italic form of the letter long s which, in turn, resembles the top of the letter italic f.

The long s for denoting an integral operation (infinite summation) was introduced by Gottfried Wilhelm Leibniz in 1675 in an unpublished paper Analyseos tetragonisticæ pars secunda (Second part of analytical quadrature).
"Hybrid" symbols

The top of the integral symbol for each font was prepared manually in an interactive font editor and included into a program that calculates the chain of variants.
Manually prepared symbols

This is undoubtedly the most tiresome part of the work, given the unpredictable number of various corrections having non infrequently an effect also on these glyphs (not speaking about the insufficient typographic skills of the authors of the TEX Gyre Math fonts)
Manually prepared symbols

This is undoubtedly the most tiresome part of the work, given the unpredictable number of various corrections having non infrequently an effect also on these glyphs (not speaking about the insufficient typographic skills of the authors of the \TeX\ Gyre Math fonts)
Never ending story: tuning font parameters

We start the work on each font with setting a few dozen of parameters – tentatively; then, as the number of prepared glyphs grows, the parameters are tuned, modified and, from time to time, new ones are being introduced – this process is resumed anew nearly each time we prepare a new set of glyphs.
Never ending story: tuning font parameters

Some of them, like the math rule thickness, have the impact on the shape of the glyphs, some, like the position of the math axis, have the impact on the positioning of glyphs; as an example of a parameter set controlling the shape of a large group of glyphs, the arrow parameters may serve – note that parameters need not to be only numerical, actually, drawing procedures can also be considered “generalized parameters”
Never ending story: tuning font parameters

\% arrow parameters:
arrow_length = math_wd + arrow_head_length;
arrow_length_both = arrow_length + 2/2 arrow_head_length;
arrow_length_long = math_long_wd;
arrow_length_both_long = arrow_length_long + 2/2 arrow_head_length;
arrow_diag_coef = 1; \% diagonal to vertical arrows length ratio (could be 1/ sqrt(2))
arrow_gap = arrow_gap' = math_gap; \% gap between arrows in a group
arrow_head_size = 360;
arrow_head_width = 1/2 arrow_head_size;
arrow_head_length = 1/2 arrow_head_size;
arrow_head_angle = 0; \% the angle governing the cavity of the edges
arrow_head_stem = math_stem;
arrow_head_stem = math_stem'';
arrowdblhead_size = 440; \ldots
arrowtplhead_size = arrowdblhead_size + math_stem + arrowdbl_gap; \ldots
arrow_acc_length = even_round 6/8 x_width; \ldots
\vartedef gen\_arrow\_head@#(expr ori, di, wd, ht, St, st, an) =\ldots\vartendef;
\vartedef gen\_arrow\_head\_black@#(expr ori, di, wd, ht, an) =\ldots\vartendef;
\vartedef gen\_arrow\_tail@#(expr ori, di, wd, ht, St, st, an) =\ldots\vartendef;
\vartedef gen\_arrow\_bar@#(expr ori, di, wd, ht, St, st, an) =\ldots\vartendef;
\vartedef gen\_arrows@#(text paths)(text tailslist)(text headslist) =\ldots\vartendef;
Never ending story: tuning font parameters

The number of arrow parameters cannot be significantly reduced because of the variety of different arrow shapes
Never ending story: tuning font parameters

An OTF math font must contain a table of math-oriented parameters (precisely, 57); some of them have obvious meaning, some of them – vague (very many thanks to Urlik Vieth for the elucidation of most tough cases)
Never ending story: tuning font parameters

An OTF math font must contain a table of math-oriented parameters (precisely, 57); some of them have obvious meaning, some of them – vague (very many thanks to Urlik Vieth for the elucidation of most tough cases)

```latex
output_otf_par("FractionNumeratorShiftUp",
               math_axis + 0[0, \text{depth}_{st}] + \text{math_gap}' + \frac{1}{2} \text{math_stem});
output_otf_par("FractionDenominatorShiftDown",
               \text{math_axis} + 0[\text{lc_height}_{st}, \text{ascender}_{st}] + \text{math_gap}' + \frac{1}{2} \text{math_stem});
output_otf_par("FractionNumeratorGapMin", \text{math_gap}');
output_otf_par("FractionDenominatorGapMin", \text{math_gap}');
output_otf_par("FractionNumeratorDisplayStyleShiftUp",
               \text{math_axis} - \text{depth} + \text{math_gap} + \frac{1}{2} \text{math_stem});
output_otf_par("FractionDenominatorDisplayStyleShiftDown",
               \text{math_axis} + \text{ascender} + \text{math_gap} + \frac{1}{2} \text{math_stem});
output_otf_par("FractionNumeratorDisplayStyleGapMin", \text{math_gap});
output_otf_par("FractionDenominatorDisplayStyleGapMin", \text{math_gap});
```
Never ending story: tuning font parameters

An OTF math font must contain a table of math-oriented parameters (precisely, 57); some of them have obvious meaning, some of them – vague (very many thanks to Urlik Vieth for the elucidation of most tough cases)

\[
\frac{1}{\frac{g}{b} \cdot \frac{d}{j}} = \frac{b}{g} \cdot \frac{j}{d} = \frac{1}{\frac{g}{b} \cdot \frac{d}{j}} \quad \text{LuaLaTeX}
\]

\[
\frac{1}{\frac{g}{b} \cdot \frac{d}{j}} = \frac{b}{g} \cdot \frac{j}{d} = \frac{1}{\frac{g}{b} \cdot \frac{d}{j}} \quad \text{MS Word}
\]

\[
\frac{1}{\frac{g}{b} \cdot \frac{d}{j}} = \frac{b}{g} \cdot \frac{j}{d} = \frac{1}{\frac{g}{b} \cdot \frac{d}{j}} \quad \text{XeLaTeX}
\]
Never ending story: tuning font parameters

An OTF math font must contain a table of math-oriented parameters (precisely, 57); some of them have obvious meaning, some of them – vague (very many thanks to Urlik Vieth for the elucidation of most tough cases)

\[
\frac{1}{\frac{g}{b} \cdot \frac{d}{j}} = \frac{b}{g} \cdot \frac{j}{d}
\]

\[
\frac{1}{\frac{g}{b} \cdot \frac{d}{j}} = \frac{b}{g} \cdot \frac{j}{d}
\]  

LuaLaTeX

\[
\frac{1}{\frac{g}{b} \cdot \frac{d}{j}} = \frac{b}{g} \cdot \frac{j}{d}
\]  

MS Word

\[
\frac{1}{\frac{g}{b} \cdot \frac{d}{j}} = \frac{b}{g} \cdot \frac{j}{d}
\]  

XeLaTeX
Another never ending story: reiterating...

So far, having finished and *extensively tested* a next font, we always found good reasons for the revision of the remaining fonts – because of spotted bugs (as a rule, a given bug inhabits all TG fonts), because of visual improvements, because of code improvement, because of some Important Personages insisting on the addition of weird and useless glyphs etc.; thus, the task can be compared to assembling a few puzzles, with tiles that permanently change their shape, at the same time...
The future
The future

- We’ve handed out the current version (0.973) of the \TeX{} Gyre Bonum to the organizers of the conference. We consider it “the eleventh hour version”; after a month or two of a “grace period”, we are going to release the improved version officially.
The future

- We’ve handed out the current version (0.973) of the \( \text{T}_{\text{EX}} \) Gyre Bonum to the organizers of the conference. We consider it “the eleventh hour version”; after a month or two of a “grace period”, we are going to release the improved version officially.

- By the end of this year, we plan to release one more TG font, namely, \( \text{T}_{\text{EX}} \) Gyre Schola Math – the last one from the \( \text{T}_{\text{EX}} \) Gyre Math collection.
The OpenType math fonts project is supported by \TeX{} Users Groups, in particular, by the Czechoslovak \TeX{} Users Group CSTUG, the German-speaking \TeX{} Users Group DANTE e.V., the Polish \TeX{} Users Group GUST, the Dutch-speaking \TeX{} Users Group NTG, TUG India, UK-TUG, and – last but not least – TUG.
Thank you for your attention & let’s meet in Bachotek again.
B. Jackowski, P. Strzelczyk, P. Pianowski
Progress in the \TeX\ Gyre Math Font Project
Bachotek, Poland, 1–5 V 2013