Another incarnation of Lucida: Towards Lucida OpenType

Ulrik Vieth and Mojca Miklavec

Abstract

TEX has been in existence for more than 30 years. During this time, TEX engines and device drivers have gone through multiple generations of font technology: METAFONT, PostScript Type 1, and OpenType. While choices for text fonts have greatly increased, there have always been few choices for math fonts and even fewer for complete font families.

The Lucida family of typefaces by Bigelow & Holmes is a notable exception, as it provides not only a math font, but also a complete font family, consisting of a large repertoire of serif, sans-serif, monospace, and fancy variants. Unfortunately, the current distribution of Lucida fonts dates back to 1993 and is limited by the use of PostScript Type 1 fonts, which support only 8-bit character sets.

In this article, we report the current status of an ongoing joint project of Bigelow & Holmes and TUG to develop a new distribution of Lucida OpenType with better Unicode language and math support.

1 Historical perspective of font technology

TEX has been in existence for more than 30 years. During this time, TEX engines and device drivers have evolved through multiple generations of font technology (METAFONT, Type 1, OpenType) and multiple generations of output formats (DVI, PostScript, PDF).

The era of METAFONT fonts When TEX was first developed in the late 1970s and early 1980s, there were no established standards of font technology which could be used, so METAFONT was developed as a companion to TEX, and all related file formats for font metrics (TFM) and bitmap fonts (PK) were invented as well.

As it turned out, METAFONT never caught on with font designers, so there were very few choices for fonts available for use with TEX in this era, both for text and math typesetting. Besides the earliest instances of METAFONT fonts, Computer Modern and AMS Euler, there were only a few more, most of them variants of the above, such as Concrete and CM Bright.

The era of PostScript Type 1 fonts When PostScript printers came into use in the early 1990s, TEX entered another era of font technology, as scalable Type 1 fonts became the preferred format. It became possible to use the commercial offerings of Type 1 fonts from many font vendors, which could now be set up for use with TEX using `fontinst` or `afm2tfm`.

This development continued when PDFTEX was developed and on-screen viewing of PDF files came into common use in the late 1990s.

While the use of METAFONT-generated bitmap fonts packaged into Type 3 fonts was still acceptable for printing, such bitmap fonts turned out to be inadequate for screen rendering in PDF viewers. As a result, the use of METAFONT fonts became unpopular, and efforts were undertaken to provide Type 1 replacements for METAFONT fonts.

In this era, choices of text fonts were increased significantly, but choices of math fonts remained limited. Besides CM and AMS Euler, converted from METAFONT, there were only the commercial offerings of MathTime and Lucida New Math at first.

It was only much later in this era that more choices of math fonts eventually became available, such as `txfonts`, `pxfonts`, `mathpazo`, `fourier`, and `mathdesign`, providing companion math fonts for use with popular text typefaces, such as Times, Palatino, Utopia, Charter, and Garamond.

Nevertheless, as we are nearing the end of this era, these choices of math fonts are still very few compared to the vast number of available text fonts, and it took more than a decade to get there.

The era of OpenType fonts In recent years, TEX has entered yet another era of font technology, as OpenType fonts are now becoming the preferred font format to support the needs of Unicode.

Many font vendors have switched their commercial offerings from Type 1 to OpenType format, and Type 1 fonts are becoming obsolete due to their limitations to 8-bit character sets.

With the development of XETEX and LuaTEX in recent years, new TEX engines have become available which support Unicode input and OpenType output directly, without the need for a complicated setup of TFM font metrics or font map files.

At the same time, support for virtual fonts is being phased out in packages such as ConTEXt MkIV, making it difficult to continue to use Type 1 fonts with virtual fonts in those packages.

As we are entering this era of font technology, there is once again a need to develop replacements for existing fonts, this time providing OpenType replacements for Type 1 fonts.

With the development of the Latin Modern and TEX Gyre fonts in recent years, replacements for the Computer Modern text fonts and several common PostScript fonts already exist, but the corresponding math fonts are still under development.

At the time of writing, choices of full-featured OpenType math fonts remain limited to Cambria Math (developed by Microsoft), Asana Math (derived from `pxfonts`) and XITS Math (derived from STIX fonts). Additional choices of OpenType math fonts are still unfinished, such as Neo Euler (derived from AMS Euler) and Latin Modern Math (derived from Computer Modern Math).

With the addition of Lucida Math as another choice of OpenType math fonts under development, we are about to reach the same level of font support in the OpenType era that was available in the early years of the PostScript era in the mid-1990s.

Nevertheless, the general trend continues also in the OpenType era in that there are few choices for math fonts and even fewer for complete font families.

2 History of Lucida font distributions

The Lucida family of fonts was developed by the Bigelow & Holmes foundry of Charles Bigelow and Kris Holmes in the mid-1980s [1]. At this time, Chuck Bigelow was on the faculty at Stanford, so he was well aware of the development of TEX and the Computer Modern fonts by Don Knuth. A primary goal of Lucida was to create a typeface design which would digitize well, even at relatively low resolutions.

Another goal of Lucida was to provide a complete font family of matching designs for serif, sans-serif, and monospace fonts, which were later augmented by a number of fancy variants [2].

Yet another goal of Lucida was to provide an extensive character set, including Latin, Greek, symbols, and even dingbats, so that the fonts could be used for math typesetting as well. (The dingbats fonts were later distributed independently [3].)

The original versions of Lucida fonts from the mid-1980s are still being sold by some font vendors under the names of Lucida Serif and Lucida Math, but these versions were never really supported with a setup for use with TEX.

The current versions of Lucida fonts were extended and revised for use with TEX in the early 1990s in cooperation with Y&Y Inc., in particular its principal Berthold Horn for TEX-specific adjustments. Y&Y sold the fonts under the names of Lucida Bright and Lucida New Math for many years, until the company was dissolved. The same font packages are now being supported and sold directly by TUG [4] and by PCTEX Inc. [5].

Finally, other versions of Lucida fonts that exist are widely distributed as system fonts with operating systems or software development kits. These include the Lucida Console and Lucida Sans Unicode fonts on MS Windows [6, 7], the Lucida Grande fonts on Apple Mac OS X [8], and a set of Lucida fonts distributed with Sun's Java Development Kits (JDK).

All of these fonts are in TrueType format and provide some level of Unicode coverage, some of them even including support for non-Latin scripts such as Greek, Cyrillic, Arabic, Hebrew, etc.

Unfortunately, most of these Lucida Unicode system fonts provide only single font instances or incomplete font families, so they are not really well suited for sophisticated typesetting.

As a result, users who want to use Lucida for typesetting with TEX are essentially stuck with the Type 1 distribution of the early 1990s, providing only a limited 8-bit character set and requiring the use of virtual fonts to support accented languages, which is no longer up to the requirements of modern TEX engines geared toward Unicode typesetting with OpenType font technology.

3 Scope of the Lucida Type 1 distribution

The current distribution of Lucida Type 1 fonts for use with TEX was originally developed in the early 1990s by Bigelow & Holmes and Y&Y Inc. It is now being supported and sold directly by TUG. The TUG distribution consists of two font packages: Lucida Basic and Lucida Complete [9, 10].

The basic distribution provides three complete font families: Lucida Bright, Lucida Sans Typewriter, and Lucida New Math (Fig. 1).

The complete distribution adds the following font families: Lucida Sans, Lucida (Serif) Typewriter, Lucida Fax, Lucida Casual, as well as several fancy variants: Lucida Blackletter, Lucida Calligraphy, and Lucida Handwriting (Fig. 2).

Given the limitations of Type 1 technology, the fonts are based on an 8-bit character set.

The recommended setup suggested by Y&Y Inc. was to use the so-called `TeXnANSI` encoding (LY1), which combines parts of the 7-bit old TEX encoding (OT1) in the lower half with the Windows ANSI 1252 encoding in the upper half [11, 12].

An alternative setup suggested by the (LA)TEX community was to use the so-called `TeXBase1` (`8r`) encoding as a base font encoding for virtual fonts, implementing the 8-bit Cork text and text companion encodings (T1 and TS1).

Since the Cork encoding extends considerably beyond the scope of Windows ANSI 1252 (Latin 1), some of the accented letters could not be provided by glyphs from the fonts, but had to be substituted by constructions in the virtual fonts.

The quality of these constructed glyphs varies considerably, but it rarely matches the quality of

- Lucida Bright
  - LucidaBright + SmallCaps
  - *LucidaBright-Italic*
  - **LucidaBright-Demi** + **SmallCaps**
  - ***LucidaBright-DemiItalic***
- Lucida Sans Typewriter
  - LucidaSansTypewriter
  - *LucidaSansTypewriter-Oblique*
  - **LucidaSansTypewriter-Bold**
  - ***LucidaSansTypewriter-BoldOblique***
- Lucida New Math
  - LucidaNewMath Roman
  - *LucidaNewMath Italic*
  - *LucidaNewMath AltItalic*
  - LucidaNewMath Symbol ($\mathcal{ABC}$)
  - LucidaNewMath Arrows ($\mathbb{ABC}$)
  - **LucidaNewMath Demi**
  - ***LucidaNewMath DemiItalic***
  - ***LucidaNewMath AltDemiItalic***
  - **LucidaNewMath SymbolDemi** ($\boldsymbol{\mathcal{ABC}}$)
  - **LucidaNewMath ArrowsDemi** ($\mathbb{ABC}$)
  - LucidaNewMath Extension

$$\odot\bigodot\oplus\bigoplus\otimes\bigotimes\Sigma\sum\Pi\prod\int\iint\oint\iint$$

Figure 1: Scope of the Lucida basic distribution.

- Lucida Sans
  - LucidaSans
  - *LucidaSans-Italic*
  - **LucidaSans-Demi**
  - ***LucidaSans-DemiItalic***
  - **LucidaSans-Bold**
  - ***LucidaSans-BoldItalic***
- Lucida (Serif) Typewriter
  - LucidaTypewriter
  - *LucidaTypewriter-Oblique*
  - **LucidaTypewriter-Bold**
  - ***LucidaTypewriter-BoldOblique***
- Lucida Fax
  - LucidaFax
  - *LucidaFax-Italic*
  - **LucidaFax-Demi**
  - ***LucidaFax-DemiItalic***
- Lucida Casual
  - LucidaCasual
  - *LucidaCasual-Italic*
- Lucida fancy variants
  - LucidaBlackletter ($\mathfrak{ABC}$)
  - *LucidaCalligraphy-Italic* ($\mathcal{ABC}$)
  - *LucidaHandwriting-Italic*

Figure 2: Scope of the Lucida complete distribution.

designed glyphs from the base fonts, so users of certain languages (such as Slovenian) were never really satisfied with those virtual fonts.

In the era of PostScript fonts used by traditional (pdf)TEX engines limited to 8-bit character sets, this was a common occurrence, which simply had to be accepted for lack of better alternatives.

In the recent era of OpenType fonts used by modern TEX engines with Unicode character sets, such deficiencies are no longer acceptable.

4 Problems of the Lucida Type 1 fonts

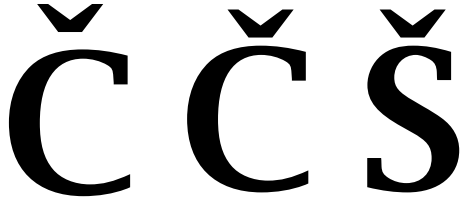
The current distribution of Lucida Type 1 fonts from TUG suffers from several problems and limitations, making it hard to set up and use the fonts with traditional TEX engines, and maybe even impossible to use them with new TEX engines.

The first problem arises from the limitations of Type 1 font technology in itself, and the associated mess of 8-bit font encodings.

When users got the original Y&Y distribution of Lucida fonts, they were confronted with making a choice how to set up the fonts. The distribution shipped with multiple sets of TFM files (using identical names for different versions) and multiple sets of font map files, that could be set up as alternatives, supporting only one choice of base font encoding, either `TeXnANSI` or `TeXBase1`.

When users installed the (LA)TEX support files for virtual fonts on top of that [13, 14], they were confronted with yet another set of TFM and VF files and another font map file (using rather cryptic, but unique font names), providing support for multiple choices of virtual font encodings (OT1, T1, TS1, or LY1) on top of multiple choices of base font encodings (`TeXBase1` or `TeXnANSI`).

Modern font distributions such as Latin Modern and TEX Gyre have solved this problem in a better way by using clearly identifiable and less cryptic font names (such as `texnansi-lmr10`, `ec-lmr10`, etc.) and providing several sets of TFM files for various encodings that can be installed in parallel, without requiring users to make a choice of one preferred encoding or using virtual fonts.



(a) (b) (c)

Figure 3: Comparison of the placement of accents: (a) constructed letter c-caron, using virtual fonts, (b) designed letter c-caron, (c) designed letter s-caron.

Moreover, these modern font distributions also support a much wider range of 8-bit font encodings, such as QX for Polish, CS for Czech and Slovak, L7X for Lithuanian, or even T5 for Vietnamese, besides EC for Cork and `TeXnANSI`.

The second problem arises from the use of virtual fonts to supply missing accented glyphs, and the associated problems of design quality.

An example of such a problem is illustrated in Fig. 3, comparing a constructed letter c-caron from a virtual font (as it used to be) with a designed letter c-caron from an OpenType font (as it should be).

In the constructed letter, the placement of the accent is done automatically, based on the glyph metrics (bounding box), so the accent is placed too high and it is centered on the geometric center of the glyph rather than the visual center.

If you compare the constructed letter c-caron with the designed letter s-caron (which happens to be available in the `TeXnANSI` encoding, despite not being part of ISO Latin 1), the difference in quality becomes very obvious. It's unquestionably desirable to have a more comprehensive set of properly designed accented letters for better language support.

Finally, a third problem consists in the requirement for virtual font support in TEX engines and device drivers to be able to provide substitutions for missing glyphs in the first place.

While the mainstream TEX distributions (such as TEX Live) have supported virtual fonts in programs such as `dvips` or `pdftex` for many years, support for virtual fonts was never universal, and it was notably absent in the device drivers of the commercial TEX distribution of Y&Y Inc.

Finally, in recent developments we are facing a situation that support for traditional virtual fonts (based on VF and TFM files) is being phased out in modern macro packages such as ConTEXt MkIV. While the LuaTEX engine still supports traditional virtual fonts, the font loader in ConTEXt MkIV now uses a completely different mechanism.

Karl Berry (TUG)	project coordination
Chuck Bigelow (B&H)	glyph design, coordination
Khaled Hosny	glyph assembly
Mojca Miklavec	testing of text fonts
Ulrik Vieth	testing of math fonts
Hans Hagen	technical advisory
Taco Hoekwater	technical advisory

Table 1: Team members of the Lucida OpenType project team and responsibilities.

5 Inception of the Lucida OpenType project

The idea of a project to create a Lucida OpenType font distribution was first conceived at last year's ConTEXt meeting in September 2010.

When a user asked how to set up Lucida for use with ConTEXt MkIV, Hans Hagen's answer was simply: "Don't use Lucida. It doesn't work!"

After a brief discussions, it was eventually concluded that something needed to be done about it, or else Lucida would soon become an obsolete and unsupported font family.

It was then suggested to hire Khaled Hosny as a developer to repackage and extend the existing Lucida Type 1 fonts into OpenType fonts and to seek support from TUG to fund and coordinate the development.

By October 2010, just a few weeks after the conference, Karl Berry had entered discussions between TUG and Bigelow & Holmes about the project, and by November 2010, the necessary legal agreement had been drafted and a project team was assembled, consisting of the team members listed in Table 1.

The agreed scope of the project was to develop OpenType versions of the Lucida basic distribution at first, which includes Lucida Bright, Lucida Math, and Lucida Sans Typewriter. Other family members, such as Lucida Sans or others may be added in a second phase of the project.

The goal for Lucida text font families was to develop OpenType fonts with good Unicode support for Latin languages, so these fonts will feature a significant number of accented Latin letters, but hardly any non-Latin scripts. In most cases, no new glyph designs will be required, just the assembly and placement of combining accents.

The goal for Lucida Math was to develop an OpenType math font with good Unicode support of math symbols and math alphabets. Besides the assembly of existing symbols from Lucida Type 1 fonts, a number of additional symbols may need to be designed, while most of the math alphabets can be taken from existing Lucida fonts.

LucidaBright	692	(955)
LucidaBright-Italic	395	
LucidaBright-Demi	395	(527)
LucidaBright-DemiItalic	395	
LucidaSansTypewriter	358	
LucidaSansTypewriter-Oblique	358	
LucidaSansTypewriter-Bold	358	
LucidaSansTypewriter-BoldOblique	358	

Table 2: Number of glyphs per font for the Lucida OpenType text fonts. (Numbers in brackets are the totals including small caps and oldstyle figures.)

6 Progress of the Lucida OpenType project

By November 2010, the project team was ready to start working, and by December 2010, the project was already well under way.

Bigelow & Holmes had supplied the designs of a number of additional glyphs for several Unicode blocks (mostly additional math symbols), and the first preliminary versions of OpenType fonts had been assembled for testing.

By January 2011, testing of the text fonts had started, while work on assembling combining accents for accented letters continued.

For the text fonts, testing mostly focused on checking the placement of combining accents and tracking the number of languages covered or the number of glyphs missing for each language.

Some statistics for the number of glyphs per font (as of May 2011) are given in Table 2 for each of the Lucida text fonts. Unsurprisingly, the regular version of Lucida Bright is the most complete one, followed by other Lucida Bright fonts.

Besides the basic ASCII and ISO Latin 1 blocks, which were already available in 8-bit Type 1 fonts, all of the Lucida Bright fonts include the complete Latin Extended-A block (U+0100 to U+017F), while only the regular Lucida Bright also includes some parts of Latin Extended-B (U+0180 to U+024F) and Latin Extended Additional (U+1E00 to U+1EFF).

Besides the more extensive glyph coverage, the regular version of Lucida Bright is also the most advanced with regard to feature support for combining marks, providing some support for multiple marks, as well as marks above and below.

For the Lucida Sans Typewriter fonts, the glyph coverage is somewhat smaller than for the Lucida Bright fonts. Most of the Latin Extended-A block is also available, but a few gaps remain, awaiting new designs from Bigelow & Holmes. Apart from that, the typewriter fonts also have fewer ligatures, but those are unlikely to be used anyway.

Compared to the old Lucida Type 1 fonts, which typically had 252 glyphs, the number of 395 glyphs in the new Lucida OpenType fonts already presents a significant advantage, especially for users of Latin languages beyond Latin 1.

Compared to other versions of Lucida fonts with Unicode support, such as Lucida Sans Unicode (1779 glyphs) or even Lucida Grande (2826 glyphs), however, the scope of the new Lucida OpenType fonts is still pretty small, as it only includes support for Latin, but not for other scripts, such as Greek, Cyrillic, Arabic, Hebrew, etc.

By March 2011, testing of the math fonts had also started, while ongoing work on extending and improving the text fonts continued.

For the math fonts, testing mostly focused on typesetting a variety of sample math documents to check for missing symbols or alphabets.

In total the new Lucida Math OpenType font includes 2148 glyphs, of which 948 glyphs are from math alphabets (U+1D400 to U+1D7FF).

An overview of the available math alphabets in Lucida Math is shown in Fig. 4. As it turned out, most of the math alphabets in Unicode could be supplied from existing Lucida Type 1 fonts (including some fancy variants such as Lucida Calligraphy and Blackletter). Only a few alphabets remain missing, such as lowercase bold Script, upper- and lowercase bold Fraktur, and lowercase Blackboard Bold letters. In addition, some individual symbols are missing in just a few alphabets.

As a unique feature, Lucida Math provides two alternate versions of the math italic alphabet, but only one version can be assigned to Unicode slots, so the other one has to be relegated to slots in the private use area and accessed via font substitutions, if the `+ss01` feature is selected.

As for the coverage of math symbols, all of the existing symbols from Lucida Type 1 math fonts have been integrated into Lucida Math OpenType. In addition, Bigelow & Holmes have supplied new designs for some additional Unicode blocks of math symbols.

While there are still a few gaps left to be filled, most of the gaps are in lesser used alphabets, so they will not affect most documents. In our tests, we have successfully typeset a number of sample math and physics documents without encountering any missing symbols.

A very small sample of math typesetting with Lucida Math is shown in Figs. 5–6 (inspired by [15]).

We are confident that Lucida Math is about as good as other existing OpenType math fonts, such

LucidaNewMath-Roman	`\mathup`	ABCXYZ abcxyz	ABΓΞΨΩ αβγξψω	0123
LucidaNewMath-AltItalic	`\mathit(-)`	*ABCXYZ abcxyz*	*ABΓΞΨΩ αβγξψω*	
LucidaNewMath-Demi	`\mathbfup`	**ABCXYZ abcxyz**	**ABΓΞΨΩ αβγξψω**	**0123**
LucidaNewMath-AltDemiItalic	`\mathbfit(-)`	***ABCXYZ abcxyz***	***ABΓΞΨΩ αβγξψω***	
LucidaNewMath-Roman	`\mathup`	ABCXYZ abcxyz	ABΓΞΨΩ αβγξψω	0123
LucidaNewMath-Italic	`\mathit(+)`	*ABCXYZ abcxyz*	*ABΓΞΨΩ αβγξψω*	
LucidaNewMath-Demi	`\mathbfup`	**ABCXYZ abcxyz**	**ABΓΞΨΩ αβγξψω**	**0123**
LucidaNewMath-DemiItalic	`\mathbfit(+)`	***ABCXYZ abcxyz***	***ABΓΞΨΩ αβγξψω***	
LucidaSans	`\mathsfup`	ABCXYZ abcxyz	(not assigned)	0123
LucidaSans-Italic	`\mathsfit`	*ABCXYZ abcxyz*	(not assigned)	
LucidaSans-Demi	`\mathbfsfup`	**ABCXYZ abcxyz**	**ABΓΞΨΩ αβγξψω**	**0123**
LucidaSans-DemiItalic	`\mathbfsfit`	***ABCXYZ abcxyz***	***ABΓΞΨΩ αβγξψω***	
LucidaNewMathSymbol	`\mathcal`	$\mathcal{ABCXYZ}$		
LucidaNewMathSymbol-Demi	`\mathbfcal`	$\boldsymbol{\mathcal{ABCXYZ}}$		
LucidaCalligraphy	`\mathscr`	$\mathscr{ABCXYZ}$ *abcxyz*		
—	`\mathbfscr`	$\boldsymbol{\mathscr{ABCXYZ}}$ (missing)		
LucidaBlackletter	`\mathfrak`	$\mathfrak{ABCXYZ}$ $\mathfrak{abcxyz}$		
—	`\mathbffrak`	(missing) (missing)		
LucidaNewMathArrows	`\mathbb`	$\mathbb{ABCXYZ}$ (missing)		

Figure 4: Overview of math alphabets in Lucida Math OpenType and where they were taken from. Note that switching between italic and alternate italic requires leaving math mode and reloading the font with different OpenType feature settings: `(+)` = fonts loaded with option `+ss01`, `(-)` = fonts loaded with option `-ss01`.

as Cambria Math or XITS Math. While Cambria Math is often used for comparison, as it was the very first OpenType math font, it also has some gaps in the math alphabets, and it may depend on the usage which ones are relevant.

7 Status of the Lucida OpenType project

As of April 2011, shortly before the presentation of the project at the EuroBachoTEX 2011 conference, a set of preliminary versions of Lucida OpenType fonts has been completed. However, the project now faces an uncertain future.

For one reason, Khaled Hosny, our main developer, will be unavailable for some time due to being drafted for military service in Egypt.

For another reason, Bigelow & Holmes did not have enough time during the academic year to supply designs for missing glyphs, so even a number of trivial issues affecting only a few glyphs have remained unfinished so far.

As for the current status, the Lucida OpenType text and math fonts clearly represent a work in progress, but not yet a finished product.

For the text fonts, it would be desirable to reach a consistent level of glyph coverage in all fonts, including all of Latin Extended-A, and possibly Latin Extended-B or Latin Extended Additional.

Of course, supporting a certain number of Latin Unicode blocks directly implies supporting a certain number of languages with Latin scripts.

In the case of Latin Extended-A and -B, this primarily implies support for European languages. In the case of Latin Extended Additional, this might even imply support for Vietnamese, although it is questionable if this will ever happen.

Besides a consistent level of glyph coverage, it would also be desirable to reach a consistent level of feature support for combining marks, including marks above, marks below, and multiple marks.

So far, only the regular version of Lucida Bright comes near this level (with some remaining gaps), while the other fonts include only Latin Extended-A (also with some remaining gaps).

For the math font, the existing coverage of math symbols and alphabets is already quite good, but it would also be desirable to close the remaining gaps in the alphabets, requiring some new designs for bold Script and bold Blackletter fonts.

Finally, once the basic set of Lucida OpenType fonts (Lucida Bright, Lucida Math, and Lucida Sans Typewriter) have been completed, there are other members of the Lucida complete set which remain to be done in a second phase, such as Lucida Sans and possibly some of the fancy variants.

Most likely, it will not be worth the effort to create a full set of accented letters for each of the

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\operatorname{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

Figure 5: Sample math document typeset with Lucida Bright and Lucida Math OpenType using the default set of math italic (OpenType feature `-ss01`).

Theorem 1 (Residue Theorem). Let f be analytic in the region G except for the isolated singularities $a_1, a_2, \ldots, a_m$. If γ is a closed rectifiable curve in G which does not pass through any of the points a_k and if $\gamma \approx 0$ in G then

$$\frac{1}{2\pi i}\int_\gamma f = \sum_{k=1}^{m} n(\gamma; a_k)\operatorname{Res}(f; a_k).$$

Theorem 2 (Maximum Modulus). *Let G be a bounded open set in $\mathbb{C}$ and suppose that f is a continuous function on G^- which is analytic in G. Then*

$$\max\{|f(z)| : z \in G^-\} = \max\{|f(z)| : z \in \partial G\}.$$

Figure 6: Sample math document typeset with Lucida Bright and Lucida Math OpenType using the alternate set of math italic (OpenType feature `+ss01`).

fancy variants, but it would certainly be useful to do so for the major variants such as Lucida Sans, and to provide a basic conversion of Type 1 fonts to OpenType for some of the other variants.

As for availability, the Lucida fonts will remain non-free commercial fonts with all rights held by Bigelow & Holmes, and licenses being sold by TUG. The members of the project team will be rewarded with a free license for the fonts, but will not get any proceeds from the sales.

8 Post-conference updates

The bulk of this article was written in May 2011 and represents the status as of EuroBachoTEX 2011.

As of June 2011 the project has been regaining momentum, as Khaled Hosny is now temporarily back to work on the project during his spare time.

As one of the first steps, the glyph coverage of the Lucida Sans Typewriter fonts has been extended to the same level as the Lucida Bright fonts, now featuring 395 glyphs representing the complete Latin Extended-A Unicode block.

As another step, a very basic conversion of the Lucida Sans fonts from Type 1 to OpenType format has been done, so that a complete set of serif, sans-serif, and monospace fonts is now available.

While the glyph coverage of the converted fonts is limited to the same 250 glyphs, having the fonts available in OpenType format should make it easier to start extending these fonts as well.

Further steps are under discussion and could be directed either towards converting more fonts or towards extending the glyph and feature coverage of existing fonts (or a bit of both).

Finally, a preliminary version of a bold math font has also been assembled, which might be used in an all-bold context such as headings or theorems. For a start, only the available glyphs from demibold Lucida Math Type 1 fonts have been assembled, but ideally, such a bold math font should eventually cover a complete set of bold symbols and alphabets, including heavy versions of bold alphabets.

In any case, work on Lucida OpenType is now continuing and has been showing great progress in just a few days, so we are confident that something useful will eventually come out of this project.

References

[1] Charles Bigelow: Notes on Lucida designs, November 2005. `http://tug.org/store/lucida/designnotes.html`

[2] Wikipedia article: Lucida
http://en.wikipedia.org/wiki/Lucida

[3] Wikipedia article: Wingdings
http://en.wikipedia.org/wiki/Wingdings

[4] TEX Users Group: Lucida and TEX
http://tug.org/store/lucida/

[5] PCTEX Inc.: Lucida Fonts
http://www.pctex.com/Lucida_Fonts.html

[6] Charles Bigelow and Kris Holmes: The design of a Unicode font, *Electronic Publishing*, 6(3), 289–305, September 1993.
http://cajun.cs.nott.ac.uk/wiley/journals/epobetan/pdf/volume6/issue3/bigelow.pdf

[7] Wikipedia article: Lucida Sans Unicode
http://en.wikipedia.org/wiki/Lucida_Sans_Unicode

[8] Wikipedia article: Lucida Grande
http://en.wikipedia.org/wiki/Lucida_Grande

[9] TEX Users Group: Lucida basic font set
http://tug.org/store/lucida/basic.html

[10] TEX Users Group: Lucida complete font set
http://tug.org/store/lucida/complete.html

[11] Y&Y Inc.: TEX'n ANSI (LY1) font encoding
http://tug.org/yandy/ly1.htm

[12] Wikipedia article: Windows 1252 encoding
http://en.wikipedia.org/wiki/Windows-1252

[13] CTAN: Lucida Font metrics
http://ctan.org/pkg/lucida/

[14] Sebastian Rahtz and David Carlisle: The `lucidabr` package, November 2005.
http://ctan.org/pkg/psnfssx-luc/

[15] Stephen G. Hartke: A survey of Free Math Fonts for TEX and LATEX, May 2006.
http://ctan.org/tex-archive/info/Free_Math_Font_Survey/en/survey.pdf

◇ Ulrik Vieth
Stuttgart, Germany
ulrik dot vieth (at) arcor dot de

◇ Mojca Miklavec
Sežana, Slovenia
mojca dot miklavec dot lists (at) gmail dot com