Representing Mathematical Knowledge in the Digital Library of Mathematical Functions

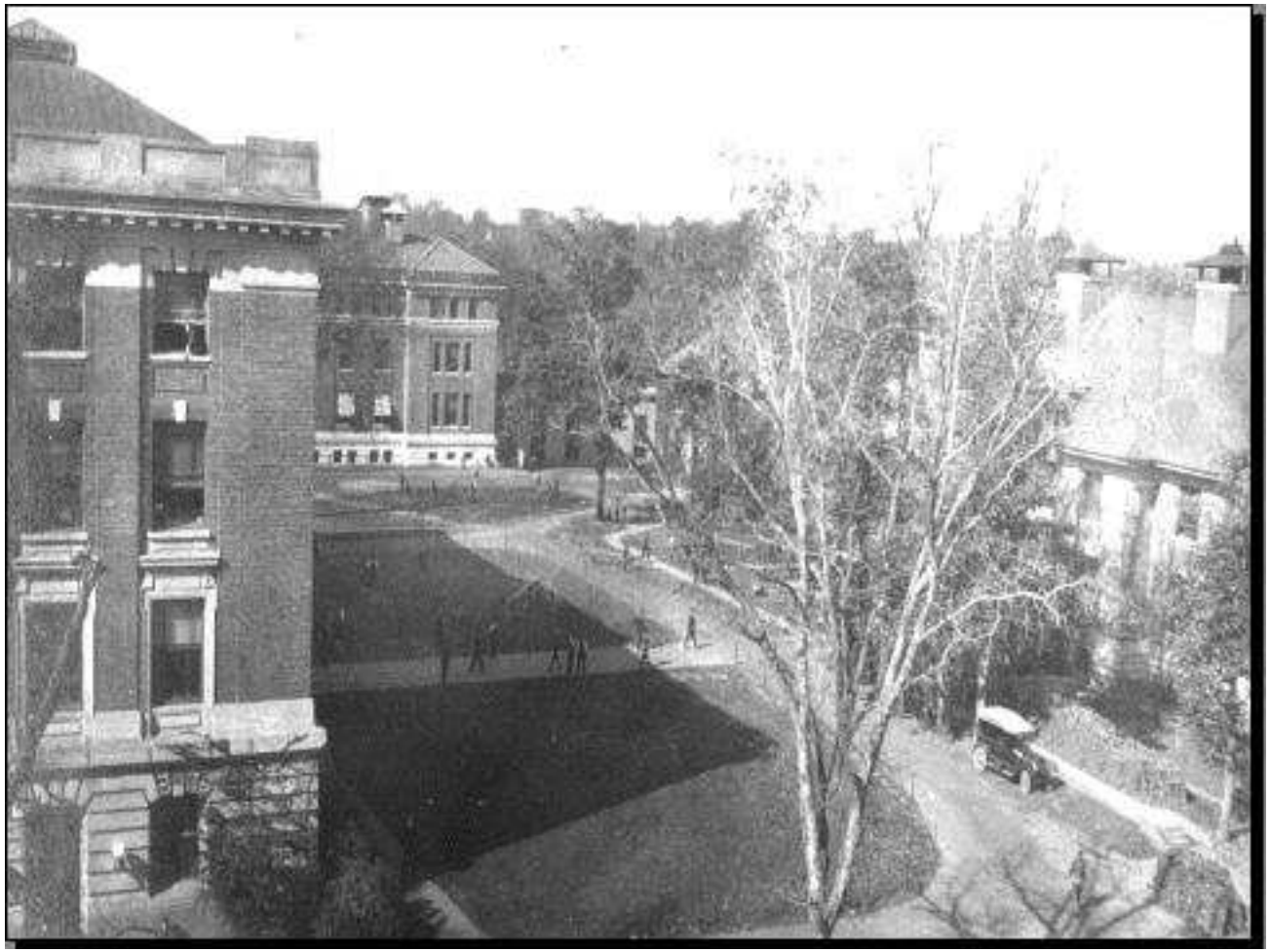
Bruce R. Miller
NIST

Digital Library of Mathematical Functions

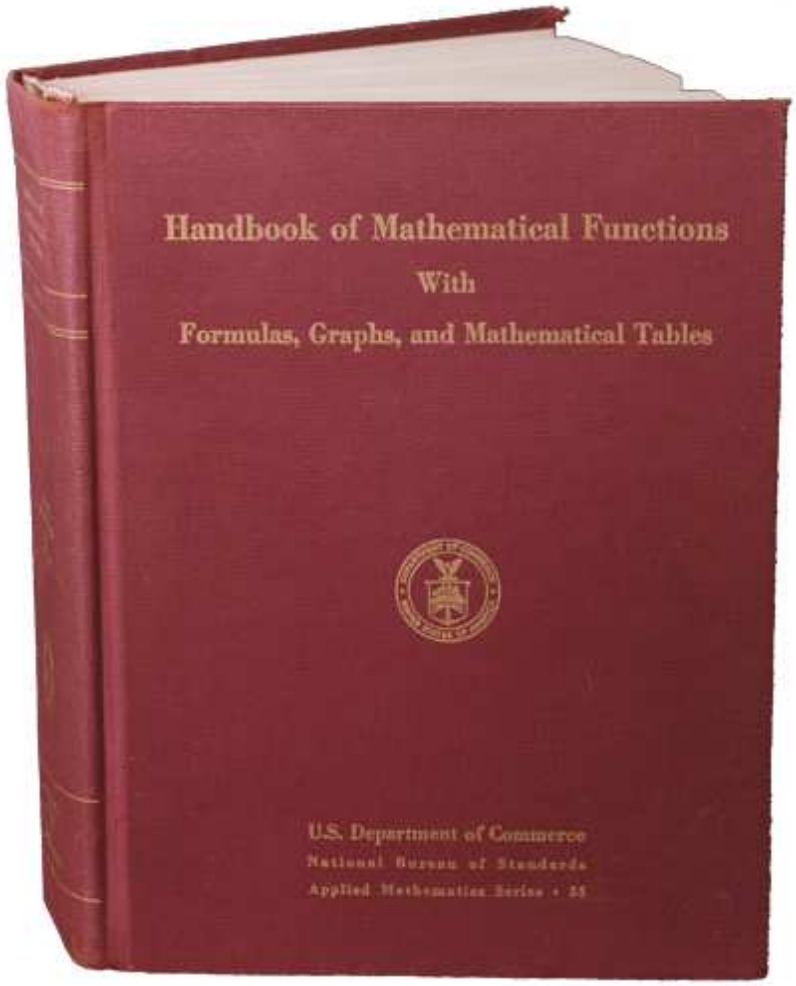
Once upon a Time

when Computers sat *at* desks

before NIST was *"formerly known as…"*



NBS published 'AMS55'[1]



1. Applied Mathematics Series No. 55

Meanwhile...

- Computers became digital,
- New functions were discovered,
- New properties of old functions were found,
- Numerical tables became *really* boring...

Meanwhile...

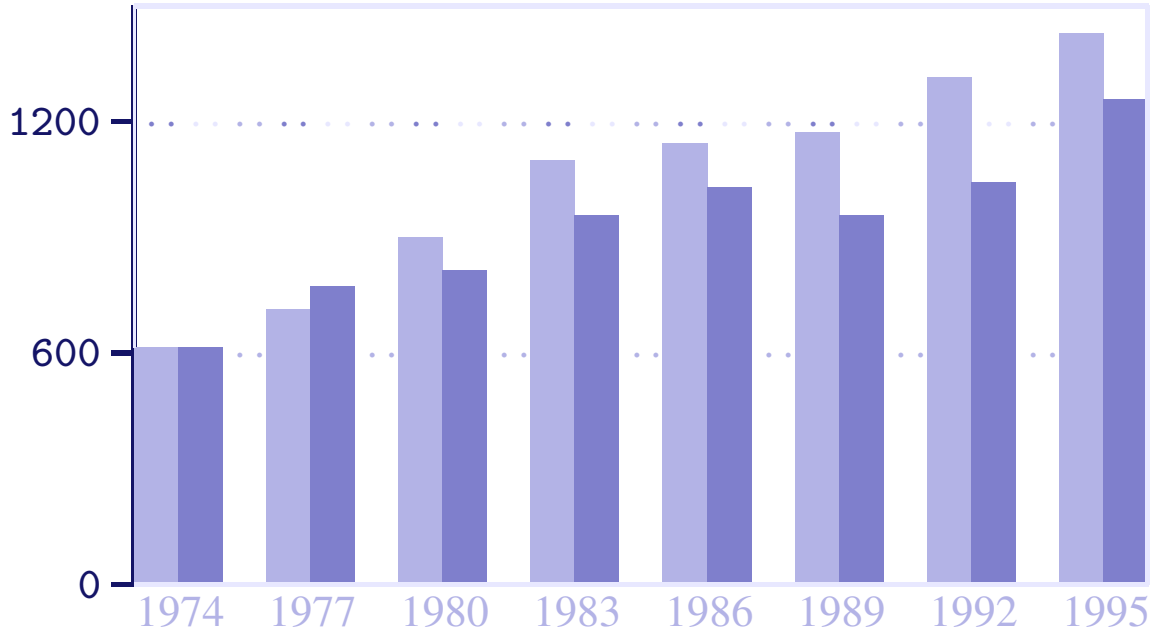
- Computers became digital,
- New functions were discovered,
- New properties of old functions were found,
- Numerical tables became *really* boring...
- And, of course,
  - The Internet
  - Hypertext
  - Computer Algebra Systems
  - MathML
  - The "*Changing Face of Mathematical Software*"!

Digital Library of Mathematical Functions

Yet...

for all the number crunching,
special functions are still *Special*.

Citations of AMS55 relative to All Scientific.



AMS55 is apparently used more than ever.

Fast Forward

- It's time for an rewrite
- . . . and many opportunities.

DLMF Project

- Started looking at feasibility in 1997.
- NSF funding for authorship in 1999.
- 4 editors, $\approx$ 12 associate editors, $\approx$ 40 authors.
- Goals:
  - New mathematical content updating AMS55,
  - in form of Digital Library,
  - and in print form,
  - by 2005.

Digital Library of Mathematical Functions

Obtaining the content: LaTeX

For our project: LaTeX

- The norm in our community and others.
- Portable and stable.
- Programmable and extensible.
- High quality typesetting.

For other projects

- Computer Algebra systems?
- (future) Word processors?

Target: XML, MathML

- Hypertext
  - $\Longrightarrow$ Connections
  - $\Longrightarrow$ Interrelations
  - $\Longrightarrow$ Annotation
- Content Oriented
  - $\Longrightarrow$ Flexible presentation
  - $\Longrightarrow$ Accessibility
  - $\Longrightarrow$ Searchability
  - $\Longrightarrow$ Reusability

Digital Library of Mathematical Functions

DLMF on the Web

For Example . . .

LaTeX as Source . . . But

- Needs more structure.
- Needs more data (often hidden).
- Quirky computational model.
- Ambiguous math markup.

DLMF Approach

Modestly Content-oriented LaTeX.

- Stay close to LaTeX standard.
- Adaptable style (print, web);
  - multi-column
  - variable width.
- Discourage presentation markup.
- Encourage Content, but keep typeable.

LaTeXML Goals

- LaTeX ⇒ XML Transformer
  - General purpose.
  - LaTeX-like DTD (or other?)
  - Math to MathML, OpenMath
- Closely mimic TeX behaviour (& Quirks).
- Lossless.
- Extensible, Adaptable.
- Encourage higher-level markup, declarations.
- . . . and finish DLMF project!

Digital Library of Mathematical Functions

Making Connections

- Traditional LaTeX: `\ref`, `\cite`, `\index`.
- Leverage our mathematics markup.
- Additional markup:
  - Annotations `\note`.
  - Special metadata: Original handbook reference.
  - Additional declarations.

Digital Library of Mathematical Functions

Using Connections

- Postprocessing XML documents.
- Disassemble XML into ‘database’.
- Note all connections.

Not really that hard.

Math: LaTeXML Data Flow

TeX source $\xrightarrow{\text{LaTeXML}}$ XML

- Let LaTeXML deal with TeX quirks.
- Acts as structure-preserving Lexer.

XML $\xrightarrow{\text{LaTeXMLpost}}$ XML'

- Use grammar-based parser.
- Use author/document-specific declarations.
- *Optionally*: math images, table transformations,...

Math: The Easy Stuff

```
a = b+c
```

LaTeXML produces the tokens

```
<XMTok>a</XMTok>
<XMTok>=</XMTok>
<XMTok>b</XMTok>
<XMTok>+</XMTok>
<XMTok>c</XMTok>
```

Math: The Easy Stuff *continued*

```
a = b+c
```

LaTeXMLpost parses this into

```
<XMApp><XMTok>=</XMTok>
  <XMTok>a</XMTok>
  <XMApp><XMTok>+</XMTok>
    <XMTok>b</XMTok>
    <XMTok>c</XMTok>
  </XMApp>
</XMApp>
```

Math: The Easy Stuff *continued*

```
a = b+c
```

Conversion to MathML yields

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>a</mi>
    <mo>=</mo>
    <mrow>
      <mi>b</mi>
      <mo>+</mo>
      <mi>c</mi>
    </mrow>
  </mrow>
</math>
```

Math: Higher Level Markup

Reduce ambiguities by introducing higher-level markup:

$$\texttt{\textbackslash deriv[n]\{f\}\{x\}} \Rightarrow \frac{d^n f}{dx^n}$$

LaTeX code:

omitted

LaTeXML declaration:

```
DefConstructor('\deriv[]{}{}',
  "<XMApp><XMTok name='deriv'/>"
  ."  <XMArg>#2</XMArg><XMArg>#3</XMArg>" ...
```

Math: Higher Level Markup *continued*

$$\texttt{\textbackslash deriv[n]\{f\}\{x\}} \Rightarrow \frac{d^n f}{dx^n}$$

LaTeXML constructs the tree:

```
<XMApp><XMTok name='deriv'/>
  <XMArg><XMTok>f</XMTok></XMArg>
  <XMArg><XMTok>x</XMTok></XMArg>
  <XMArg><XMTok>n</XMTok></XMArg>
</XMApp>
```

Parser can treat args individually,

. . . avoiding much guesswork.

Math: Special Functions

With appropriate TEX macrology:

$$\texttt{\textbackslash HyperpFq\{p\}\{q\}} \Rightarrow {}_pF_q$$

Introduce notion of *evaluating a function at*:

$$\texttt{\textbackslash HyperpFq\{p\}\{q\}@\{a\}\{b\}\{z\}} \Rightarrow {}_pF_q\left(a;b;z\right)$$

or (alternative notation)

$$\texttt{\textbackslash HyperpFq\{p\}\{q\}@@\{a\}\{b\}\{z\}} \Rightarrow {}_pF_q\left(\begin{matrix}a\\b\end{matrix};z\right)$$

Palatable notation? Easier to type than

```
\sideset{_{p}}{_{q}}{\mathop{F}}\left({a \atop b};z\right)
```

Math: Special Functions *continued*

With the end result:

```
<XMApp>
  <XMTok name='HyperpFq'>F</XMTok>
  <XMTok>p</XMTok>
  <XMTok>q</XMTok>
  <XMTok>a</XMTok>
  <XMTok>b</XMTok>
  <XMTok>z</XMTok>
</XMApp>
```

and we know *which* '*F*' is intended.

Math: Issues

- Role of text and spacing in math.
- Overloading of *symbols* (scoping?)
  - f is a function here, but a variable there.
- Palatable LaTeX extensions for math.
- For *really* meaningful math (eg. OpenMath)
  - need type analysis
  - need more info from authors
- Open ended...