BIJLAGE R

PROOF

Math into BLUes [1]

Kees van der Laan

Hunzeweg 57, 9893PB
Garnwerd, The Netherlands
+31 5941 1525

March, 1991

Abstract

TEXing mathscripts is not simply typing. Math has to be translated into TEX commands.
First the motivation for this work is given. Next traditional math page make-up is summarized along with the macroscopic math TEX commands. After answering 'Why TEXing mathscripts is difficult?' an anthology of TEXfalls and their antidotes is discussed. At the end suggestions are given in order to lessen the difficulties.

Prelude

My assistance was called for in TEXing a mathscript. Part of the mathscript was typed, contained TEX commands, but ... did not pass the TEX formatter. Inspection revealed it never could have. It occurred to me that at least three typists had been involved, demonstrating the use of LATEX, $\mathcal{AMS}$-TEX and macros from other sources, all mixed up. Furthermore, the TEXscript showed various TEXfalls. I like to define these as: correct TEXing not yielding the required or customary lay-out. Also the pseudo-guru involvement was felt, which I like to define as too complicated use of TEX, inhibiting intelligibilty of the TEXscript, with a wink to TB373

> 'Always remember, however, that there's usually a simpler and better way to do something than the first way that pops into your head.'

Not only did I look over the shoulder of a typist, I also inspected a math book, TEXed by a mathematician, Temme(1990). The book looks good. Examples are taken from it in order to show other ways of TEXing. A matter of taste?

In the sequel attention is paid to
- Traditional math page make-up.
- What makes TEXing mathscripts difficult?
- An anthology of TEXfalls[2] with antidotes.
- What ought to be done to lessen the difficulties.

For you and me

Most, if not all, (math) TEXfalls have been envisioned by the grand wizard himself and references to those or related issues are indicated by TB (The TEXbook) followed by page or exercise number.
Mathscript denotes a mathematics manuscript. TEXscript denotes a TEX formatted compuscript, especially the one my assistance was asked for. TEXnigma is a computer system with TEX installed. TEXnowledge means knowledge of TEX. TEXist is a TEX typist. BLUe is DEK's unwary B.L. User. The math book denotes Temme(1990).

1 Math page make-up

Swanson(1986) is a good source for traditional math mark-up. In publications math is either part of the running text or displayed. In displays 'indentation' on all sides is on, and formulae are sometimes aligned, for example at the '='-symbol.

TEX requires math within text to be surrounded by '$'-s, `$<math>$`. Displayed math is tagged by '$$'-s, `$$<displayed math>$$`. For the general multi-line display, plain TEX provides the macro `\displaylines`, TB194, 362, and for aligned formulae the macro `\eqalign`, TB190, 362. By default displays are centered. That is all for TEXing math, from an outer level point of view. Problems?

[1] To be presented at Dedham91, TUG conference 1991, Paris91, EuroTEX conference 1991 —© 1991, TEX Users Group

[2] The TEXfalls treated are not specific to plain, $\mathcal{AMS}$-TEX, nor LATEX. They illustrate basic pitfalls in TEXing math. Sources are: the inspected TEXscript, the math book and some I stumbled upon myself. LATEX is rather superficial with respect to math. Formula classes are not even mentioned! Dead wrong, but understandable from the viewpoint of descriptive mark-up.

Example (*Pascal triangle)*

$$\displaylines{1\cr 1\quad1\cr 1\quad2\quad1\cr 1\quad3\quad3\quad1\cr \cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot}$$

is obtained via

```
$$\displaylines{1\cr
  1\quad1\cr
  1\quad2\quad1\cr
  1\quad3\quad3\quad1\cr
  \hbox to 7em{$\cdot$\hss
    $\cdot$\hss$\cdot$\hss
        $\cdot$\hss$\cdot$}
}$$
```

The example demonstrates two levels of formatting math: the inner level where the triangle has to be defined unambigiously —here in detail TEX commands, and the outer level where this triangle is positioned within the context —here $$-signs meaning displayed, and subject to the style of the publication series.

By phone one would say: Pascal's triangle, you know, 1, with below it 1 and 1, and there below 1, 2, and 1, and there below 1, 3, 3, 1, etc. All centered. For formatting more precise information is needed than for talking math by phone, in order to eliminate ambiguity. A computer-based formatting system is not yet that intelligent.

Right or left formulae numbers can be provided after the tag `\eqno`, respectively `\leqno`, TB187, 362. Individual lines in a multi-line display can be numbered. Therefore the macro `\eqalignno`, respectively `\leqalignno`, is provided, TB192, 362.

In summary, all plain TEX's math page make-up macros are demonstrated in the following templates.

$$\sin 2x = 2\sin x\, \cos x \qquad \text{(TB186)}$$

$$F(z) = a_0 + \frac{a_1}{z} + \frac{a_2}{z^2} + \cdots + \frac{a_{n-1}}{z^{n-1}} + R_n(z), \qquad n = 0, 1, 2, \dots,$$

$$F(z) \sim \sum_{n=0}^{\infty} a_n z^{-n}, \quad z \to \infty \qquad \text{(TB ex19.16)}$$

$$\begin{aligned} \cos 2x &= 2\cos^2 x - 1 \\ &= \cos^2 x - \sin^2 x \end{aligned} \qquad \text{(TB193)}$$

$$\begin{aligned} \cosh 2x &= 2\cosh^2 x - 1 && \text{(TB192)} \\ &= \cosh^2 x + \sinh^2 x \end{aligned}$$

obtained via

```
$$\sin2x=2\sin x\, \cos x
     \eqno({\rm TB186})$$
$$\displaylines{F(z)=
a_0+{a_1\over z}+{a_2\over z^2}+\cdots
   +{a_{n-1}\over z^{n-1}}+R_n(z),\cr
         \hfill n=0,1,2,\dots\,,\cr
\hfill F(z)\sim\sum_{n=0}^\infty a_nz^{-n},
       \quad z\to\infty\qquad\qquad\hfill
       \llap{(TB ex19.16)}\cr}$$
$$\eqalign{\cos2x&=2\cos^2x-1\cr
                 &=\cos^2x-\sin^2x\cr}
  \eqno({\rm TB193})$$
$$\eqalignno{
\cosh2x&=2\cosh^2x-1&({\rm TB192})\cr
       &=\cosh^2x+\sinh^2x\cr}$$
```

Remark. It was difficult to get the example with labeled `\eqalign` right in two-column format. It would left justify because of insufficient space left by the big label. Removing the glue '`\,`', before the `\vcenter` in the body of `\eqalign` forced TEX to center the formula, see TB189.

One can also use the general `\halign` macro. For example from TB ex22.9 we have

$10w\,+$	$3x$	$+\,3y$	$+\,18z$	$=1,$		(9)
$6w\,-$	$17x$		$-\;\;5z$	$=2,$		(10)

obtained via

```
$$\openup1\jot\tabskip=0pt plus1fil
\halign to\displaywidth{\tabskip=0pt
$\hfil#$&$\hfil{}#{}$&
$\hfil#$&$\hfil{}#{}$&
$\hfil#$&$\hfil{}#{}$&
$\hfil#$&${}#\hfil$\tabskip=0pt plus1fil&
\llap{#}\tabskip=0pt\cr
10w&+& 3x&+&3y&+&18z&=1,&( 9)\cr
 6w&-&17x& &  &-& 5z&=2,&(10)\cr}$$
```

I consider `\cases`, `\(p)matrix`, and `\overbrace`, respectively `\underbrace`, parts of formulae. For example: x is called an eigenvector with eigenvalue λ of the matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}$$

if $Ax = \lambda x$, TB177. Obtained via

```
$$A=\pmatrix{
    a_{11}&a_{12}&\ldots&a_{1n}\cr
    a_{21}&a_{22}&\ldots&a_{2n}\cr
    \vdots&\vdots&\ddots&\vdots\cr
    a_{n1}&a_{n2}&\ldots&a_{nn}\cr}$$
```

2 Am I blue?

This section should have been filled with impressive and dazzling examples like those that go along TEX-product ads. Because of space restrictions they have been omitted.

3 What is wrong Doc?

Mathscripts differ from TEXscripts. The source

```
$$x=1+\left({y^2\over k+1}
      \right)^{\!\!1/3}.$$
```

looks different from[3]

$$x = 1 + \left(\frac{y^2}{k+1}\right)^{\!\!1/3}.$$

Because of this discrepancy, the problem is how to get a correct TEXscript, starting from just a mathscript. The mathscript has to be translated into a correct TEXscript. This is difficult because of the complexity of math typesetting, and the inherited complexity of TEX, if not of the bewildering variety of TEX-based products.[4]

First, one has to find the appropriate format command, from nearly a thousand.[5] In the TB the following is devoted to math formatting: chapters 16(11p), 17(21p), 18(23p), 19(14p), 22(242, ex22.9/11), 24(up to 281, 15p), 26(5p); appendices A (answers to excercises), B(6p, macros), F(13p), G(7p). Add to these the required general TEXnowledge of how to use TEX for non-complex documents, of how to use TEX for general page make-up, of how to format tabular material (matrices, commutative diagrams), of how to handle output routines, of how to use non-default fonts, and nobody would consider TEX to be trivial.[6]

Second, contents and context dependent extras have to be added, now and then, as demonstrated in this paper.

Third, once the TEX language is mastered, the difficulty remains to locate and correct errors. Misconceptions as well as typos.[7] So add to the above TB chapter 27, just for completeness.

Fourth, the (La)TEX bugs and LATEX's inconsistency. Once you have coped with all mentioned above, you are nevertheless thrown back by true bugs. I was trapped by LATEX's quote environment when I tried for the opening quote to hang out. It did not work, even not after inserting `\null`.

Spivak(1986), has dealt with TEXing math in his delightful book, but helas it is not a proper extension. My attitude is to look for what is needed and to extend plain in a compatible way, keeping overhead as small as possible. Plain provides enough TEXfalls already.

4 The bad news

I like to start with mentioning the nasty small white space on a new line after a heading. This creature can be killed by providing a comment symbol —`%`—direct after the heading command. Just a warming-up for the unwary.[8]

The sequel in this section started as a list of pitfalls. It grew out into a discussion with antidotes. If readability for BLUe is reduced below par, I pitfailed.

4.1 Too many.

The too many pitfall is a serious one. It occurs when many incompatible products are used, which are partly, or not at all, understood.

In the typing project, TEXed chapters showed different approaches. $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX was used in one, LATEX in another etc. This demonstrated involvement of several typists and the lack of a common approach. The document did not run either, showing that TEXing is one thing and getting it correct —if not alone for those braces—is quite another. This holds especially for typists not familiar with programming. Apart from the above, TEXing was done inconsistently. $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX was used for some math symbols not available in plain, like $\gtrless$. Commands like `\frac`, and `\overset` were used along with their plain functional equivalents. Obviously one typist was $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX oriented, while others were not.

In conclusion. The TEXscript was far from correct, suffered from leaning upon too many tools, and for the rest was full of horrible TEXfalls. The math book didn't suffer from this TEXfall, just plain, and an extra symbol or two.

I was trapped when preparing this paper. This paper uses LTUGproc.sty, and therefore LATEX. In LATEX `\eqalign` etc. are not available, so I defined them. But, ... I did not think of redefining `\centering`, which has another meaning within LATEX than within plain TEX. As a consequence `\eqalignno` went wrong without saying so. It just did!

Another TEXfall was that `\eqalign` did not center in two-columns format when `\eqno` was used as well! I had to inactive the first glue item '`\,`' of `\eqalign`, see for explanation TB189. Weird.

I started with LTUGproc.sty because my contribution for Cork90 was done by it, and I thought of needing the picture environment. For the NTG MAPS PROOF version, as well as for the GUTenberg cahiers reprint, I will stay with LATEX. The final version will be in plain, of course.

Another failure was when using the quote environment with the first quote hanging out. That did not work, and so I fell back upon `\midinsert`,

[3] Note that the kind of parentheses and the kind of division notation have to be specified as well.

[4] In the sequel we restrict ourselves to plain TEX, and assume that no fancy, friendly WYSIWYG user-interface is available.

[5] Cheswick(1990) has provided a KWIC with all the TEX and LATEX commands. This is handy when in doubt whether a command is already in use.

[6] Beeton(1990) states that it was the intend of the $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX-project to 'simplify input of complex mathematical expressions.'

[7] The TEXist task has been silently increased by the parsing and correcting of the TEXscript in order to provide proofs.

[8] This is overlooked in the Dutch course book on LATEX, and also in the Dutch 'brief' style, where the adressee label on the sequel page headings is preceeded by that white space.

\narrower, etc. TB340, but that was not allowed either! Then I redefined \midinsert into \bgroup\narrower\smallskip, and \endinsert into \smallskip\egroup. I don't like that kludging around, and finally comprise.

Remark. I welcome the approaches taken by TUGboat, see Whitney and Beeton(1989), and AMS, see AMS(1990), where on top of a common TEX style similar LATEX and TEX procedural markup 'user interfaces' have been built. This provides freedom for authors to submit either LATEX or plain (TEX) scripts, while the publisher can easily integrate these sources into one publication.
However, for every single mathematician it does pay to TEX as simpleas feasible. Understanding the basics, with a few macros added, will do, especially for those who otherwise have to relay on Word-whatever. This is demonstrated by the math book, and by the grand wizard himself.

4.2 Lessened SoC.

Lessened separations of concerns. *The* pitfall for author-publishing systems is insufficient awareness of accomplishments in other fields. Not only has the author to worry about the contents, the organization, the power of the examples, the use and spelling of the language, consistency, etc. He has also to worry about math in print conventions, the computer system, and typing skills.

4.3 First gains.

The typographic mark-up pitfall reflects the temptuous direct formatting of how elements should *look* like, instead of tagging the elements with the purpose of *identifying* them. A matter of abstraction and separation of concerns. One can think of the various headings: chapter, definition, theorem and the like, where the formatting can be postponed and provided separately in style or format files. This pitfall can also be classified as the portability pitfall: submitting an article to another journal needs adaptation of the copy when descriptive mark-up is not used. I encountered in the TEXscript

```
{\bf Summation of infinite
     series of complex functions.}
{\bf Theoretical background.}
\vskip1truecm
{\it1.2.1\underbar{Ten standard
                    definitions}}.
\vskip1truecm
\underbar{Definition 1.}
```

As can be seen a lot of typographical detail had been supplied. Agreed, generally it is available in the mathscript, because authors are used to denote bold and underlining. It is easy-going just to type ahead, I presume. To use formats provided by the publisher, does pay, because then it is the concern of the publisher to get the results right.

BLUe will find difficulties in defining a theorem environment where \proclaim is to be used. Proclaim is an outer command. In order to overcome the difficulties just take the source of \proclaim and remove outer.

To the same category of pitfalls belongs typing commands for extra white space along with each display, especially when the mathscript is full of crowded formulae. Instead of repeatedly typing \vkip-s, use can be made of the \everydisplay command, along with assigning new values to \abovedisplayskip, \belowdisplayskip, and their short variants.
Within a display more white space can be obtained between the lines. One does not have to modify the code because \openup increments. Just say for example \openup1\jot, and interspacing is increased by the given amount. $\mathcal{AMS}$-TEX's \spreadlines is a disguise of this assignment.

4.4 Emptyness.

The spacing pitfall is a difficult myriad.[9] Once the automatic spacing is overruled by explicit spacing commands the inconsistency pitfall opens up. A nice list of rules for spacing between symbols in math is given in Swanson(1986, chapter 3).

In math mode spaces in the input are ignored. Before and after each formula space is inserted of size \mathsurround, defaulted in plain to 0pt, TB162, 353. Within a formula the spacing is context dependent, and determined by the class of the math character, see TB170. Some symbols, for example of class binary, get extra spacing *around* them. Punctuation symbols take spacing *after* the symbol. The math character classes are given on TB154. For each class the precise spacing values, related to the context, are given in the table on TB170.

$$(\lambda)_2\,{}_2F_1, \quad \int\!f(x)\,dx, \quad \Gamma_{\!2}+\Delta^{\!2} \quad \sum^\infty_{n=-\infty}\!\!\cos nt$$

is TEXed as, TB168,

```
$$(\lambda)_2\,{}_2F_1,            \quad
       \int\!f(x)\,dx,             \quad
 \Gamma_{\!2}+\Delta^{\!2}         \quad
\sum^\infty_{n=-\infty}\!\!\cos nt$$
```

In the math book I found $n!n^2$, TEXed via $n!n^2$, instead of $n!\,n^2$, with input $n!\,n^2$.
Negative kerning after integral signs was not used either, especially with double integrals. The integral signs are spread too much and too loose from their integrands.
Another aspect of spacing is ${}_1\phi_0(a;-;q,z,)$. The empty symbol could have been used, ␣, via {\tt\char'040}.
And what about placeholders? For example $\bigl(f,K_n(\cdot,y)\bigr)$, via $\bigl(f,K_n(\cdot,y)\bigr)$? Introduce space around the placeholder via

[9] Once in a while I think of TEX as dealing essentially with flexible spaces.

`\,\cdot\,.`
Interesting are expressions in exponents or indices. The math book contained

$$e^{-z\sinh t+\nu t}$$

which does not look nice because of suppression of space around the operator. Introduce explicit thinspaces, before and after the binary operator, or use parenthesis around the argument of the function.
In the TEXscript I encountered among others

```
\wit
$$|t| \quad < \quad | \quad x \quad -
  \quad (x + 1)^{\frac12}
  \quad (x - 1)^{\frac12} \quad |,$$
\wit
```

Spacing between formulae was not understood and done inconsistently. At many, many places unnecessary extra white space was ordered for; hundreds of '`\,`.' `\quad`, respectively `\qquad` were abused.
On the use of `\(q)quad`, I can best quote TB166.

> 'The traditional hot-metal technology for printing has led to some ingrained standards for situations like this, based on what printers fall a 'quad' of space. Since these standards seem to work well in practice, TEX makes it easy for you to continue the tradition: When you type '`\quad`' in plain TEX format, you get a printer's quad of space in the horizontal direction. Similarly, '`\qquad`' gives you a double quad (twice as much); this is the normal spacing for the F_n example below.'

A little further on the page the reader's attention is drawn to the different approach needed in alternating math and text in a paragraph. Just alternate, as simple as that.

```
$F_n=F_{n-1}+F_{n-2}$, for $n\ge2$.
```

Consistency can be enhanced by defining document elements, and subsequently use the element via the name. For example the real part of z can be obtained in math mode via `\Re z`, once we have defined

```
\def\Re#1{{\rm Re}\,#1}
```

In the math book this was implemented via `{{\cal R}\,#1}`, which is handy especially when real parts of quantities are used in formulae. In the TEXscript I also encountered the following subtle examples, which after correction read

$$C_\nu^\lambda(-z) = \cos\pi\nu\, C_\nu^\lambda(z) - \sin\pi\nu\, D_\nu^\lambda(z),$$

where after the arguments of the trigonometric functions '`\,`' (extra space) had to be inserted. In the math book similar situations were circumvented via parentheses, $\cos(\pi\nu)C$, via `$\cos(\pi\nu)C$`, no extra space has to be inserted before the opening parenthesis and after the closing parenthesis, TB170.

4.5 Class unawareness.

Below several examples are provided which demonstrate the unawareness of mathematical characters belonging to one of eight classes, TB154.

4.5.1 Innocent braces.

The pitfall is that braces are not harmless but yield a formula of class 0 within math mode! Confer
`$a+b$, $a{+}b$ and a+b`
with results $a+b$, $a{+}b$ and a+b. The first + is of class binary and takes spacing according to the table on TB170, and in the second the + is reduced to class zero, and takes only `\mathsurround` spacing, defaulted in plain to 0pt.
Similarly, TB171, shows

```
$|-x|$, $\left|-x\right|$,
        and $\lfloor-x\rfloor$
```

with results $|-x|$, $\left|-x\right|$, and $\lfloor-x\rfloor$.[10] In the math book I found $\gamma^\star(a,x)$, as well as $\gamma\star(a,x)$, do you see what went wrong?
TEXperts swing around with braces, especially in alignments where empty formulae are to be used now and then. Whenever innocence shows up don't believe it![11]
So, for TEXing math, BLUe must understand the various atom classes, TB158 etc.

4.5.2 Whoops.

What about this

```
\def\Inn{{\raisebox{1pt}
        {{\hbox{{$\in$}}}}}}
```

The concept of a binary operator was not accounted for, yielding wrong spacing. TEX could not know that the raised `\in` had to be considered as an operator. It had been reduced to class ordinary, taking `\mathsurround` spacing.
In the math book I found

$$2\pi i\,{\rm Res}_{s=e^{i\pi}}f(s) = -2\pi i\, e^{i\pi z}.$$

TEXed via

```
2\pi i\, {\rm Res}_{s=e^{i\pi}}f(s)
    =-2\pi i\, e^{i\pi z}.
```

I would prefer for the `Res` operator (in display and in agreement with Swanson(1986))

$$\mathop{\rm Res}_{s=e^{i\pi}} f(s) = -e^{i\pi z}$$

via

```
 \mathop{{\rm Res}}_{s=e^{i\pi}}
       f(s)=-e^{i\pi z}
```

An example where spacing has to suppressed is $<name>$, via `${<}name{>}$`. The relational operators are not used as such, and coerced into class zero, by the braces. The latter example is taken from the BNF-notation of programming languages, denoting meta-linguistic variables.

[10] Why $|-x|$, and not just $|x|$? Furthermore, norm fences don't belong to the openings respectively closings class.
[11] Another occurrence of *harmful* braces is given by `\cnt={1}` and the like, yielding an error message.

4.5.3 Just a comma.

The number $3{,}14$ innocently formatted as `$3,14$`, would yield $3,14$.
The correct formatting is `$3{,}14$`, TB134.

The comma belongs to the punctuation class of math symbols and the surrounding braces —making a subformula—reduce it to class ordinary, which doesn't take extra spacing.

Remark. As part of text, the number could have been obtained via `3,14`, no $'s around it. Context dependency remains difficult.

4.5.4 With dots.

A dot is in use for a (binary) multiplication operator and as a punctuation mark. Three dots in a row don't yield the ellipsis result. The formatting of the ellipsis is context dependent: at the axis of the formula, at the baseline, vertical, or diagonal.
Binary operator vs. punctuation mark.
A multiplication in mathematics can be denoted by: $a \times b$, $a \cdot b$, and also implicitly by a thinspace $a\,b$, which has to be marked explicitly.
Typists, and those used to the old typewriter, err by using '`x`' for '`\times`,' and by using the punctuation dot '`.`' instead of '`\cdot`,' the binary multiplicator operator positioned centrally, and for the last notation just a space for '`\,`,' gobbled away by TEX in good mood.[12]

The general issue is that the handwritten symbols must be recognized from the context, as a punctuation symbol, as an operator or . . . as significant space!
Colons: is there a difficulty?
A colon as a punctuation symbol can be obtained via the '`\colon`' command, and as a relation symbol, via '`:`', TB134. Examples are

$$f\colon A\to B,\quad \{x:x>5\}$$

obtained via

```
$$f\colon A\to B,\quad \{x:x>5\}$$
```

The math book used ':' throughout.

4.6 CMR.

Text in displays and standard function names take traditionally roman fonts, Swanson(1986, Table IV).
$\mathrm{sinh}\, x$ was TEXed by `$\hbox{sin }hx$`, demonstrating bad handwriting, and wrong TEXing. The TEXist was not familiar with the hyperbolic function names and therefore could not compensate for the bad handwriting, and also apparently lacked awareness of how to TEX them properly. I also encountered

```
\cos\,\alpha
h^\lambda_\nu (z) \, : = _2F_1( ... )
\hbox{ for Re }\, z \, > \, 0
```

Horrible!

[12] To this category of misuses I also reckon 1 vs. l, 0 vs. o.
[13] TEX does adapt just (or \{.

4.7 Kameleons.

To the kameleon pitfall I reckon the situations where TEX can't distil from the context the right sizing. TEX provides facilities for automatically formatting the right size, given the context. TEX provides for example the right-sized openings and closings for a matrix, when these are specified by `\left...`, respectively `\right...`[13]
A TEXfall occurs when the context does not prompt for any need of another size, while BLUe expects TEX to do everything right.

Example (*Context dependent sizes)* Inspired on Spivak(1985, p55)

```
$$||\alpha(\sqrt a+\sqrt b)||
    \leq|\alpha|.
  (||\sqrt a+\sqrt b||).$$
```

with the result

$$||\alpha(\sqrt a+\sqrt b)|| \leq|\alpha|.(||\sqrt a+\sqrt b||).$$

Better TEXing is

```
$$\bigl\|\,\alpha(\sqrt{\mathstrut a}+
       \sqrt{\mathstrut b}\,)\,\bigr\|
  \leq|\alpha|\,
  \bigl\|\sqrt{\mathstrut a}+
   \sqrt{\mathstrut b}\,\bigr\|$$
```

with result

$$\bigl\|\,\alpha(\sqrt{\mathstrut a}+\sqrt{\mathstrut b}\,)\,\bigr\| \leq|\alpha|\, \bigl\|\sqrt{\mathstrut a}+\sqrt{\mathstrut b}\,\bigr\|$$

In this example norm fences are taken larger and all sqrt's must be told to have arguments of `\mathstrut` size. Ascender and descender invariance! Moreover the multiplication dot can better be replaced by a thinspace.

Another use of the vertical bar occurs in set notation, for example, TB ex18.22,

$$\bigl\{\,x^3\bigm|h(x)\in\{-1,0,+1\}\,\bigr\}.$$

obtained via

```
$$\bigl\{\,x^3\bigm|h(x)\in%
       \{-1,0,+1\}\,\bigr\}.$$
```

This not only demonstrates to use the correct size of the outer braces and the vertical bar, but also to be aware of the binary operator function of the vertical bar, defaulted with the appropriate spacing. Set notations in the math book had not been marked up via the use of `\mid|`, nor by its variants. For nested parentheses the big etc. representations were not used. The old technique with square brackets for the outer parentheses was used. For example, $[\ln(z+1)]^m$.
Note that it looks better to introduce some spacing along with the outer braces. I expect these kinds of issues to be handled inconsistently in a document of non-trivial size. If an author wants these kinds of results he has to indicate that in the mathscript.

4.8 Triads.

'Three dots in a row,' or generalized ellipsis, is heavily used in mathematics notation. For example

$$x_1 + \cdots + x_n, \quad x_1 x_2 \ldots x_n$$

is obtained by using the `\cdots` and `\ldots` commands, respectively.

The general issue is not to use '...', but to use the `\cdots`, respectively `\ldots` command.

Remark. When the ellipsis is followed by a punctuation dot a small extra space '`\,`', has to be ordered for: `1+x+x^2+\cdots\,.` An ellipsis is often used in a fixed context, for example: for $i = 1, 2, \ldots, n$. This can be obtained via

```
for $i=1$,~2, $\ldots\,$,~$n$.
```

Such sentences are candidate for abbreviation into `\for in`, by use of

```
\def\for#1#2{
   for $#1=1$,~2, $\ldots\,$,~$#2$},
```

supporting consistency. It also reduces the number of keystrokes. Note that `\dots` is not substituted for `\ldots\,`, because `\,` is needed. The pitfall of mixing up `\dots` and `\ldots` use, is also circumvented by the use of the `\for` abbreviation.
The math book was inconsistent in using `\cdots`, respectively `\ldots`.

In order to facilitate the look up of the shorthands, Wichura(1990) has provided some macros yielding a table consisting of a math-writing-column and a corresponding TEX-input column. A fancy tool, suited for typists, I presume. This is not enough in order to solve the typist's problems. It might help, though. Education is needed and discipline has to be adhered to. What about a discipline of TEXing?

4.8.1 Real life.

Other 'dots' are also in use: vertical, in matrices, TB177, and diagonal, TB177, ex18.45. As an example of diagonal dots consider, related to TB142,

$$1 + \mathop{\Phi}_{k=1}^{n} \frac{a_k}{b_k} \stackrel{\rm def}{=} 1 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \ddots + \cfrac{a_{n-1}}{b_{n-1} + \cfrac{a_n}{b_n}}}}$$

with (space saving) variant notations

$$\begin{aligned} &= 1 + \frac{a_1 \rfloor}{\lfloor b_1} + \frac{a_2 \rfloor}{\lfloor b_2} + \cdots + \frac{a_n \rfloor}{\lfloor b_n} \\ &= 1 + \frac{a_1}{b_1 +} \frac{a_2}{b_2 +} \cdots \frac{a_n}{b_n} \end{aligned}$$

The above is obtained via

```
\def\cf{\mathop{\grkop \Phi}}
$$\eqalignno{
1+\cf_{k=1}^n{a_k\over b_k}
&{}\buildrel{\rm def}\over=
 1+{a_1\over\displaystyle b_1+
   {\strut a_2\over\strut
    \vrule height3ex width0pt\relax
    \displaystyle b_2 +
     \lower2.0ex\hbox{$\ddots\,
      \lower1.25ex\hbox{$+
      {\displaystyle a_{n-1}\over
        \displaystyle b_{n-1}+
                 {\strut a_n\over
           \displaystyle b_n}}$}
                        $}
    }
   }\cr
\noalign{\noindent with variant notations}
&{}\buildrel{\rm\phantom{def}}\over=
1+{a_1\,
\smash{\vrule depth1ex}\vrule height2ex
\over\strut\vrule\,b_1}
+{a_2\,
\smash{\vrule depth1ex}\vrule height2ex
\over\strut\vrule\,b_2}
+ \cdots
+{a_n\,
\smash{\vrule depth1ex}\vrule height2ex
\over\strut\vrule\,b_n}\cr
%
&{}\buildrel{\rm\phantom{def}}\over=
1+
{a_1\over\textstyle\strut
    \vrule height2.5ex width0pt
    b_1\,+\,}
{a_2\over\textstyle\strut
    \vrule height2.5ex width0pt
    b_2\,+\,}
    \cdots
{a_n\over\textstyle\strut
    \vrule height2.5ex width0pt
    b_n}
\cr}%end\eqalignno
$$
```

The general issue is awareness of styles —display, text, script or scriptscript style—which is default where, and how to override the default, TB140 etc. Awareness of the difference between `\strut` and `\mathstrut` is needed as well. In the math book similar diagonal dots were used for denoting an infinite continued fraction. Swanson(1986) just provides an ellipsis. Her variant notations differ a little. Those given originate from Henrici. His Φ symbol is the absolute space saver for regular continued fractions.

Remark. The auxiliary symbol `\cf`, the Φ, must be made robust, such that it can be used with other styles as well, yielding suitable size. TEX provides for this aim `\mathchoice`. For example `\cf` should have been defined as

```
\def\cf{\mathop{\mathchoice{%
\grkop\Phi}{%Magnified
\Phi}{\Phi}{\Phi}
}}
```

Note the precise alignment at the $=$ symbols, and how the alignment is interrupted.

4.9 Accents differ.

Accents are treated differently in math mode than in horizontal mode. DEK decided to provide different math mode accent commands, TB135. It is unclear to me, why the commands have not been overloaded.

4.9.1 Embellishments.

We have barred letters,

$$\bar z,\ \overline z,\ \bar P,\ \overline P,\ \bar h,\ \hbar,\ \overline{AB},$$

obtained via

```
$$\bar z,\ \overline z,\
\bar P,\ \overline P,\
\bar h,\ \hbar,\
\overline{AB},$$
```

Easy to forget is that `\bar` provides a bar of fixed length, and looks strange when used over capitals. It is wrong to use it with subformulae in general, which holds for all accents. Note also the use of `\vex a`, for $\vec a$, `\vec A`, for $\vec A$, accented with arrow, and `\overrightarrow A`, for $\overrightarrow A$, or `\overrightarrow{AB}`, for $\overrightarrow{AB}$, TB136, 359. Once again a source of confusion and inconsistency. For vectors Swanson(1986) advises to use boldface. The math book `\bar`-ed capitals.

Normalized functions are often denoted by dotted letters, and because a dot is a tiny blot of ink, more pronounced dots are wanted. I encountered the use of bulleted letters, which looks awful. Something like a bold dot is needed. Bold dotted P, '$\boldsymbol{\dot P}$', and simple dotted P, '$\dot P$', are obtained via `\bdot P`, respectively `\dot P`, with the use of

```
\def\bdot#1{{\bf\dot{{\mit#1}}}}
```

Note the extra pair of braces, another TEXfall.
The math book even used triple-dotted letters, to denote the Schwarz derivative, $\overset{\cdots}{x}$. The above was done via

```
${\buildrel\textstyle{\cdot}
{\cdot}{\cdot}\over\textstyle x}$.
```

4.9.2 Prime-ry.

Characters can be primed in math mode, denoting derivatives. For instance y' via `y^\prime` or via the more natural shorthand `y'`. The `'` character is active and overloaded. The TEXfall is that symbols which take limits can't be simply primed in display. Confer

$$\sum\nolimits' \quad {\sum}' \quad \mathop{\sum'}^n_{k=0} \quad \mathop{{\sum^n_{k=0}}'} \quad \mathop{\sum\mathstrut'}^n_{k=0} \quad \mathop{{\sum}'}^n_{k=0} \quad \mathop{{\sum}'}_{k=0}^n$$

obtained via

```
$$\sum'                          \quad
{\sum}'                          \quad
\mathop{\sum'}^n_{k=0}           \quad
\mathop{{\sum^n_{k=0}}'}         \quad
\mathop{\sum\mathstrut'}^n_{k=0}\quad
\mathop{{\sum}'}^n_{k=0}         \quad
\mathop{{\sum}\acclap'}_{k=0}^n  $$
```

Primed summation symbols are used in Chebyshev expansions

$$\sum_{k=0}^{n}{}' a_kT_k(x) \stackrel{\rm def}{=} .5a_0 + a_1\,x + a_2\,T_2(x) + \cdots \\ + a_n\,T_n(x).$$

See TB ex18.44. The above is obtained via the variant solution[14]

```
\def\acclap#1{
   \raise\hgtsig\hbox to0pt{$#1$\hss}}
\newdimen\hgtsig
\setbox0=\hbox{$\displaystyle{\sum}$}
\hgtsig=\ht0\relax
\advance\hgtsig by -1.75ex
$$\displaylines{\quad
  \setbox0=\hbox{$\displaystyle
   \mathop{{\sum}\acclap'}_{k=0}^n$}
   \dp0=0pt \box0       %Neglect dp size
   a_kT_k(x)\mathrel{\mathop=^{\rm def}}
  {\textstyle.5}a_0+a_1\,x+a_2\,T_2(x)+
                      \cdots\hfill\cr
   \hfill{}+a_n\,T_n(x).\quad}$$
```

The double primed summation symbol can be obtained, similarly. As mentioned by DEK, the problem is to center the lower limit with respect to the summation symbol proper.[15] Note the uncommon use of the first half, which looks awful under the default `\displaystyle`. Note also the smash details for letting the line distance process ignore the depth of the indexed summation symbol.

4.10 $\frac{1}{2}$'s never have the right size.

BLUe invariantly goes wrong when TEXing 'halves.' The following

$$D^\lambda_0(z) = 4a_\lambda\, z\, {}_2F_1(\textstyle\lambda+\frac12, \frac12; \frac32; z)$$

is TEXed via

```
$$D^\lambda_0(z)=4a_\lambda\, z\,
{}_2F_1(\textstyle\lamda+{1\over2},
        {1\over2};{3\over2};z)$$
```

and (to be avoided, Swanson(1986))

$$D^\lambda_0 = -\sin \tfrac{\pi\nu}{2}\, C^{\frac{\lambda}{2}}_{\frac{\nu}{2}}$$

via

[14] Well, smashed a bit.
[15] Scripting the primed operator would violate this.

```
$$D^\lambda_0=-{\textstyle
\sin{\pi\nu\over2}\,
C^{\scriptscriptstyle\lambda\over
              \scriptscriptstyle2}
 _{\kern-1pt{\scriptscriptstyle\nu\over
                   \scriptscriptstyle2}}
   } $$
```

In the math book I encountered the above notation, and incidentially $F(1/2,1/2;3/2;z^2)$. I also stumbled upon $\int^{\frac12\pi}$ with later on the more usual $\int^{\pi/2}$. The latter is also recommended by Swanson(1986).

The general point is to kern and coerce the right style. Another example of where coercing the right style is needed, occurs when the summation symbol takes double lower limits. Explicit mentioning of `\scriptstyle` in both operands of the `\atop` command is needed, TB145.

Remark. DEK(1985), mentions the use of a typographer's '${}^1/_2$,' especially in recipes, which works better than a mathematician's '$\frac12$'.

4.11 Various OOOOOO's.

Mathscript O's are overloaded: '$\emptyset$,' the empty set, $f\circ g\colon x\mapsto f\bigl(g(x)\bigr)$, composition, and the order symbols $o(h^2)$, and $O(h^2)$. Obtained via

```
$\emptyset$,
$f\circ g\colon
  x\mapsto f\bigl(g(x)\bigr)$,
$o(h^2)$,
$O(h^2)$.
```

We also have trigonometric and temperature degrees 30°, respectively $^\circ\mathrm{K}$, TB180. Another challenge is a notation for the zero vector, TB ex18.6.

4.12 Backslash penances.

Because of the special function of the backslash, people are in trouble when the symbol itself is wanted. In horizontal mode the backslash as such can be obtained by selecting the symbol from the `tt`-font, TB429, position '134 (decimal 92), via `{\tt \char'134}`. In math the backslash is used for the setminus (binary) operator and for denoting cosets, the latter takes no space.
Compare

$$A\setminus A=\emptyset \quad \text{and the cosets of } G \text{ by } H\colon G\backslash H.$$

TEXed by use of `\setminus`, respectively `\backslash`, TB436. Unnecessary to say that the mathscript contained several setminus operations, while in the TEXscript the `\backslash` was used throughout.

4.13 Over and over.

In TB ex17.3 BLUe is imprinted to treat a fraction as a subformula, id est, to use braces around <*formula*-`\over`-*formula*>. A good habit to adhere to throughout. I was trapped when changing `\left(` and `\right)` into `\bigl(`, respectively `\bigr)`. The former pair creates a subformula while the latter don't. Ahhhhhh, robustness!
Swanson(1986) advises to consider the use of slashes when saving space can be obtained with clearity of exposition preserved.
In `\buildrel`, TB437, `\over` is overloaded.

4.14 Too difficult.

Hypergeometric functions take sometimes 'matrices' as arguments. As stated in TB178, the use of `\(p)matrix` in the text of a paragraph yields too big results. $M_n(z)={}_{n+1}F_n\bigl({k+a_0,\atop}{k+a_1,\dots,k+a_n\atop k+c_1,\dots,k+c_n};z\bigr)$ is obtained via

```
$M_n(z)={}_{n+1}F_n
  \bigl({k+a_0,\atop\phantom{kc_1}}
       {k+a_1,\dots,k+a_n\atop
            k+c_1,\dots,k+c_n}
;z\bigr)$
```

Note the automatic centering 'on the axis' of the last argument. A fuzzy issue is what to do with empty arguments, especially when several `\atop`-s are used in a row. The general approach is to use `\mathstruts`-s. For two `\atop`-s the use of `\phantom` will yield aligned results, as demonstrated in the given example.
The late Luke used instead of ';' the '|' symbol, for example

$${}_pF_q\Bigl(\,{\alpha_p\atop\rho_q}\mathpunct{\bigm|}\,z\Bigr)={\Gamma(\rho_q)\over\Gamma(\alpha_p)}\,G^{1,p}_{p,q+1}\Bigl(-z\,\mathpunct{\bigm|}\,{1-\alpha_p\atop 0,1-\rho_q}\Bigr)$$

is obtained via

```
$${}_pF_q
\Bigl(\,{\alpha_p\atop\rho_q}
  \mathpunct{\bigm|}\,z
\Bigr)=
{\Gamma(\rho_q)\over
   \Gamma(\alpha_p)}\,
G^{1,p}_{p,q+1}
\Bigl(-z\,\mathpunct{\bigm|}\,
  {1-\alpha_p\atop0,1-\rho_q}
\Bigr)$$
```

Here the vertical bar is coerced into a punctuation symbol, with some extra spacing added. Note also the lack of spacing in the subscripts and superscripts, TB170.

The TEXscript contained

```
\begin{dispeqs}
M_n(z):={}_{n+1}F_n
\left(\aligned
k+a_0,&k+a_1,\dots,k+a_n;z\\
      &k+c_1,\dots,k+c_n
      \endaligned\right)
\tag1.2.55
\end{dispeqs}
```

not reflecting the centering of '`; z`.' Fortunately, I never needed to talk about hypergeometric functions by phone.

4.15 As simple as math can be.

Formula numbers are perhaps the most simple math elements of a math paper. Just numbers. This way of formula numbering suffers from the modification pitfall. The disadvantage of explicit numbering comes to light when copy is changed, involving modification of formula numbers. Authors circumvent retyping of the numbers by introducing suffixes a,b,c, ..., when numbered formulae have to be inserted, and sacrifice strict sequencing when numbered formulae have to be omitted. Nowadays retyping can be circumvented by automatic numbering, jeopardizing the formula-number tie which some authors have developed. The better alternative is not thinking in number-formula ties, but in name-formula ties. How about that?

4.15.1 Automatisms.

Plain does not provide for automatic numbering of formulae. One can easily define a new counter and write a macro with the function to advance the counter globally and provide the number, preceded eventually by chapter and section number with appropriate punctuation (Remind the mode you are in however, TB ex19.6, 19.7). In the example below we assume that chapter and section counters, `\cct`, repectively `\sct`, exist next to the formula counter `\fct`, and are handled appropriately when entering a chapter or section. For example, the numbering in

$$|x| = \begin{cases} x, & \text{if } x \ge 0 \\ -x, & \text{if } x < 0 \end{cases} \tag{2}$$

is TEXed by

```
\fct=1 \def\frmcnt{\global\advance\fct1
    (\the\fct)}
$$|x|=\cases{
      x,& if $x\ge0$\cr
     -x,& if $x < 0$\cr}
     \eqno\frmcnt$$
```

Note that the 'second column' is in horizontal (text) mode[16], and that numbering of the formula is at the axis of the formula. The above approach can be used in `\eqalignno` as well. After the second `&`, insert again `\frmcnt`. For authors who cherish their habit to add a, b etc. to the number there is no problem. They can provide the counter with add-ons, for example the label (2a) is simply obtained via

```
$$|x|=\cases{
       x,& if $x\ge0$\cr
      -x,& if $x < 0$\cr}
      \eqno(\the\fct{\rm a})$$
```

Pittman(1988)[17] and Nearing(1989) have provided macros for extending plain with automatic formula numbering (and symbolic referencing as well). Both did fail with respect to compatible *extension* of plain, although their macros can be easily adapted. I would unite Nearing's `\eqnum` and `\eqalignnum` into for example the `\frmcnt` command, with the functionality as demonstrated above. `\frmcnt`, yielding the counter value, with the current lay-out embellishments, can be used as well after the `\(l)eqno` tag as after the second & of `\(l)eqalignno`. And why not along with `\displaylines`? Yes, you are right, that is compatibility!

What about the typist? Usually, the author provides the numbers as an integral part of the mathscript. I consider it easy to replace the number by the systematic call of `\frmcnt` at the place of the formula number. As simple as that, and can be done by one keystroke.[18]
The awareness that numbers should be typed after `\eqno`, or after the second & in `\eqalignno`, has to be applied anyhow.

4.16 You name it.

By symbolic referencing we mean linking a number to a name and to refer to the number via the name. This differs from automatic numbering because of the multitude of names, while automatic numbering use *one* counter (name). In so-called forward referencing the use of the name precedes the assignment of the number to the name. Linking a name to a number can be achieved via macro assignment. References made to the formula after this assignment can be done just via the call of the appropriate macro. That is the principle. The lay-out of the numbers has to be addressed somewhere, favorably in the format.

Example (*Link* $\backslash\langle name\rangle$ *to number)*

$$a^2 + b^2 = c^2 \tag{5}$$

is obtained, with `\forpyth` linked to (5), via

```
$$a^2+b^2=c^2\eqno\gdef\forpyth{(5)}
                              (5)$$
```

Subsequent references to this equation can be done via the 'name' `\forpyth`, with the number to be delivered in the format style of the context.[19]

An inconvenience is forward referencing. A suggestion how to handle this, is to print in the margin at the place

[16] While for number fields math mode is defaulted, I consider it an error-prone exception to have horizontal mode defaulted in the second column of cases. Not serious, though, it can be easily adapted.

[17] I hate trickery like `\ifnum0<0\csname...`, which is in Pittman's code. How long does it take to find the TEXfall in `\ifnum\cnt=0\else ...\fi`? The point is that the next symbol after a number will be expanded, in order to find out whether the number ended, TB208. So `\else` will be swallowed, adding the part after `\else` to the token list when the condition is true and omitting it when the condition is false. Innocent spaces!?!

[18] Assuming the editor is TEX intelligent, see Williams and Hall(1990).

[19] To reduce confusion, and support mnemotechnique, the discipline is prompted that each *for*mula reference name begins with `for`.

where the forward referencing was done a reminder that a forward reference has been made. One could print for example the <*name*>=??? (with ??? the placeholder for the forward reference number). Sooner or later the numbers have to be filled in.

Example (*Forward referencing)* At this place we like to refer forward to the Pythagoras equation, (???), via `\forref{\forpyth}{???}`.

“forpyth =(???)

$$a^2+b^2=c^2 \tag{1.2.3}$$

Above we displayed Pythagoras equation (3). Obtained via

```
\def\forref#1#2{
%#1 label name starting with \,
%#2 is number or ??? if unknown
%LaTeX's \marginpar is used.
  (#2)
\ifproof\marginpar{
 \vtop{\hbox{\string#1}
       \hbox{=(#2)}}}\fi
}%end forref marking}
$$a^2+b^2=c^2
\eqno%equation 3, chapter 1, section 2
     \xdef\forpyth{(3)}(1.2.3)$$
```

It is assumed that the author knows which name is linked to which equation, so that he can easily find the number and fills it in. Of course when the number is known, it can be typed in directly from the mathscript, but use `\forref` en type the number instead of the ???. One never can tell.

4.17 Both ways.

Automatic counting and symbolic referencing can be combined. Assume that the running formula counter is called `\fct`, and that the running chapter and section counters are called `\cct` and `\sct`, respectively. The above example extends into

$$a^2+b^2=c^2 \tag{1.2.3}$$

obtained via

```
$$a^2+b^2=c^2
%hidden values: \fct, \cct, \sct
\eqno\global\advance\fct1
     \xdef\forpyth{(\the\fct)}
   (\the\cct.\the\sct.\the\fct)$$
```

In order to reduce the number of keystrokes the above functionality can be covered by a `\labf`[20] command, appropriately defined. For example

```
\def\labf#1{
%links formula number to label
%#1 label: \<name>
%\fct is advanced and
%via xdef assigned to \<name>
  \global\advance\fct1
  \xdef#1{(\the\fct)}
(\the\cct.\the\sct.\the\fct)}
```

with the use

```
$$a^2+b^2=c^2
%hidden values: \fct, \cct, \sct
\eqno\labf\forpyth$$
```

References to the formula go via the name `\forpyth`, and the lay-out defined via the format style of the publication is obtained.[21] I chose to surround the number with parentheses. Fully automated symbolic reference systems are part of LaTeX and $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX. Generally, use is made of an auxiliary file, which stores the numbers, assigned during the previous run. With forward referencing two runs are needed. Moreover, in those system one can also ask for the page number where the reference was made. Cross-referencing between document parts processed separately and independently, is also supported!

4.18 We want more.

Sometimes more alignment positions than the one position provided by `\eqalign`, (or `\(l)eqalignno`), are wanted. See for example TB242, where it is suggested to use `\eqalign` repeatedly within one display, and ex22.9 (equation 8 and 9 in this paper) for a general approach via `\halign`. Repeated use of `\eqalign` suffers from non-guaranteed equal line distances over the various `\eqalign`. In the math book I encountered

$$\begin{aligned}
\cos(z\sin\theta) &= J_0(z) + 2\sum_{n=1}^{\infty} J_{2n}(z)\cos 2n\theta\\
\sin(z\sin\theta) &= \qquad\quad + 2\sum_{n=1}^{\infty} J_{2n+1}(z)\sin(2n+1)\theta
\end{aligned}$$

which was not appropriately aligned, possibly because of not using the empty formula. Appropriate input

```
$$\eqalign{
\cos(z\sin\theta)={}
&J_0(z)&{}+2\sum_{n=1}^\infty
        J_{2n}(z)\cos2n\theta\cr
\sin(z\sin\theta)={}
&       &{}+2\sum_{n=1}^\infty
  J_{2n+1}(z)\sin(2n+1)\theta\cr
}$$
```

With only one alignment position the same result could have been obtained with `\phantom`, and again the empty formula, {}.

[20] I chose for `\labf` but a general `\lab` can be made which inspects an environment parameter in order to decide which counter has to be advanced.

[21] Note that in the display the chapter and section number are also printed, while a reference in the sequel text yields the number according to the appropriate format.

4.18.1 Adaptation of `\eqalign`.

Eqalign can easily be adapted with respect to a variable number of alignment points. The number of alignment points does not have to be provided explicitly. No parameter is needed. The idea is to make use of the repetition facility for template elements in `\halign`; just double the appropriate &.

```
\catcode`\@=11%@ letter
\def\eqalign#1{%\,  <---
      %Glue omitted in 2-columns env.
%Variable number of alignment points,
%determined dynamically, and
%automatically.
 \vcenter{\openup1\jot\m@th
 \ialign{\strut\hfil
   $\displaystyle{##}$&&  %doubled
   $\displaystyle{{}##}$\hfil\crcr
   #1\crcr}}\,
}% end flexible \eqalign
\catcode`\@=12%@ other
```

With two alignment positions we can nicely format

$$\begin{aligned}(z^2-1)^\lambda &= (z-1)^\lambda(z+1)^\lambda, &&\quad \mathrm{Re}\,z>0,\ z\notin[0,1]\\ &= e^{\mp 2i\pi\lambda}(z-1)^\lambda(z+1)^\lambda, \\ & &&\quad \mathrm{Re}\,z<0,\ \mathrm{Im}\,z\gtrless 0\end{aligned}$$

via[22]

```
$$\eqalign{(z^2-1)^\lambda
&=(z-1)^\lambda(z+1)^\lambda,
    &\quad\Re z>0,\ z\notin[0,1]\cr
&=e^{\mp2i\pi\lambda}
  (z-1)^\lambda(z+1)^\lambda,
  \hidewidth\cr
&   &\quad\Re z<0,\ \Im z\gtrless0\cr
}$$
```

Note the use of `\hidewidth`. `\gtrless` will be discussed later. In the math book no alignment was done on occasion for function pairs, such as

$P_\nu^{-\mu}(z) = \ldots$

and

$Q_\nu^{-\mu}(z) = \ldots.$

Spivak(1986), provides a general `\alignat#1` macro, where the parameter provides the required number of alignment positions.

4.18.2 Adaptation of `\(l)eqalignno`.

Pragmatically, one can use the general `\halign` command. One can also copy `\eqalignno`, extend the template and give the modified version an appropriate name. This is simple and serving its purpose. The more so because having more than one alignment position is the exception rather than the rule.

For a general poly-macro the number of alignment positions must be provided, because the last element of the alignment template is reserved for the formula number, and differs from the in-between ones. The automatic repetition mechanism can't be used. From the given value the template elements for the in-between alignments can be repeated, dynamically. The dynamical approach can be achieved via the use of `\aftergroup`. For the idea see the paradigm of defining a control sequence for `\n` asteriks, TB374.
Another possibility is selection of the appropriate code via a case-like construction.

4.19 Loooooonnnnngggggg.

With long formulae we have two situations. First when in-line formulae extend the line, and second when in displays formulae are too long.

For the first case TEX breaks at customary points like: *after* binary operations and relations, TB195.

For the second case TEX does not automatically break displayed formulae. The reason is that displayed math is too complex to automate line breaking, under the restriction to convey optimally the meaning of the formulae. The author knows best where to split a formula, and has to prompt TEX and ipso facto the typist.
Authors should be aware of the typographic tradition to break in displays *before* binary operations and relations. For a summary of the rules see Swanson(1986, 3.3.4).

TEXing splitted formulae (in display) can be done via

1. Just use (two) consecutive displays.
   This produces too much vertical white space and both equations are centered, which does not look good.
2. To use `\(l)eqalign(no)`.
   The alignment position must be chosen. A good alignment choice is the = symbol, and to precede the second part with `&\qquad{}`, the extras. You won't find the latter in the mathscript. Schematically,
   ```
   \eqalign{
     <LHS>&=<RHS first part>\cr
          &\qquad{}RHS sec part>\cr}
   ```
   In the math book no 'extras' were inserted, just aligning at = and on the next line(s) with the $+/-$.
3. To use the free-format `\displaylines` command, see TB194.

4.19.1 Via consecutive displays.

The interdisplay width could be adapted. Because of simpler, self-contained and already available alternatives this is not further elaborated.

[22] The above could have been obtained by using the phantom mechanism with alignment position chosen at `\Re`.

4.19.2 Via eqalign.

First, the typist has to be aware of the scope restrictions. For example `\left(` and `\right)`, can't be split, can't be used in isolation. They should be given within one scope, TB196. In eqalign(no) terms: they can't be separated by '&.' The non-context dependent variants (i.e. of fixed size), like `\Bigl(` respectively `\Bigr)`, can be split. Second, the line distance may vary unnoticed, because of the context.

Example (*Non-constant line distances)* The example does not deal with splitting a one-line formula into two parts, but is taken from the math book, modified to dramatize the effect.[23]

$$\begin{aligned} E_0(x) &= 1, \\ E_1(x) &= x - \tfrac{1}{2}, \\ \Bigl(E_2(x) &= x^2 - x\Bigr)^2, \\ E_3(x) &= x^3 - \tfrac{3}{2}x^2 + \tfrac{1}{4}. \end{aligned}$$

is obtained via[24]

```
$$\eqalign{
E_0(x)&=1,\cr
E_1(x)&=x-\textstyle{1\over2},\cr%
\Bigl(E_2(x)&=x^2-x\Bigr)^2,  \cr%
E_3(x)&=x^3-\textstyle{3\over2}x^2
      +{1\over4}.\cr}$$
```

Constant line distance can be obtained by the use of `\smash`, and editing at the proof phase.

Example (*Constant line distance)*

$$\begin{aligned} E_0(x) &= 1, \\ E_1(x) &= x - \tfrac{1}{2}, \\ \Bigl(E_2(x) &= x^2 - x,\Bigr)^2, \\ E_3(x) &= x^3 - \tfrac{3}{2}x^2 + \tfrac{1}{4}, \end{aligned}$$

is obtained via

```
\def\sfr#1#2{{
    \textstyle{#1\over\smash#2}}}
$$\eqalign{E_0(x)&=1,\cr
   E_1(x)&=x-\sfr12,\cr
\smash{\Bigl(}E_2(x)&=x^2-x,
                \smash{\Bigr)^2},\cr
   E_3(x)&=x^3-\sfr32x^2+\sfr14.\cr
}$$
```

The use of `\sfr` had no effect!

4.19.3 Either way.

TB196 gives an example for flushing both ways. Better suited for 2-column format is

$$\sin z = z - \frac{z^3}{3!} + \frac{z^5}{5!} - \frac{z^7}{7!} + \frac{z^9}{9!} - \frac{z^{11}}{11!} + \cdots \qquad |z| < \infty.$$

obtained via

```
$$\displaylines{%
\sin z=z-{z^3\over\smash{3!}}
       +{z^5\over\smash{5!}}
       -{z^7\over\smash{7!}}
       +{z^9\over\smash{9!}}
       -{z^{11}\over\smash{11!}}
    +\cdots    \hfill\cr
\hfill{} |z|<\infty.    \cr
}$$
```

Note the use of the empty formula after `\hfill`, in order to make TEX recognize an eventual '$+$' symbol as *binary* operator, and provide suitable spacing.

4.19.4 Centered and right.

TB ex19.17, gives a set of formulae with one formula splitted into two parts, and the second part appropriately placed relative to the first part. The solution has made use of `\eqalign`. In order to disturb as little as possible the appearance of the aligned set of equations one could wish to flush right the splitted parts. This is in agreement with Swanson(1986, 3.3.5). In 2.5.2 she advises to slash stacked fractions in display because of space economy. The example was adapted from the math book. From the context it is clear why this representation had been chosen in the math book.

Example (*Splitted parts flushed right)*

$$\begin{aligned} u_k &= \frac{1}{(k+1)(2k+1)(4k+1)} \\ &= \frac{\frac{1}{3}}{k+1} - \frac{1}{k+\frac{1}{2}} + \frac{\frac{2}{3}}{k+\frac{1}{4}} \\ &= \frac{1}{3}(\frac{1}{k+1} - \frac{1}{k}) - (\frac{1}{k+\frac{1}{2}} - \frac{1}{k}) \\ &\qquad + \frac{2}{3}(\frac{1}{k+\frac{1}{4}} - \frac{1}{k}) \qquad (3) \\ \sum_{k=1}^{\infty} u_k &= -\tfrac{1}{3}\psi(2) + \psi(1\tfrac{1}{2}) - \tfrac{2}{3}\psi(1\tfrac{1}{4}) \end{aligned}$$

is obtained by

```
$$\eqalignno{u_k
&={1\over(k+1)(2k+1)(4k+1)}\cr
&={{1\over3}\over k+1}
  -{1\over k+{1\over2}}
  +{{2\over3}\over k+{1\over4}}\cr
&={1\over3}({1\over k+1}-{1\over k})
  -({1\over k+{1\over2}}-{1\over k})\cr
&&\displaystyle{}+{2\over3}({1\over k
  +{1\over4}}-{1\over k})  \quad
  (3)\cr%second half + form number
\sum_{k=1}^\infty u_k
&=\textstyle-{1\over3}\psi(2)
  +\psi(1{1\over2})
  -{2\over3}\psi(1{1\over4})\cr
}$$
```

[23] In the math book there was no squaring, the unequal line distances were visible without it. I could not reproduce that.
[24] Look also at the prime-ry section.

4.20 Interrupts.

Authors want to connect displayed, and aligned, formulae by texts. TEXnically this means the opposite: the alignment has to be interrupted. For example, TB193,

$$x = y + z$$

and

$$x^2 = y^2 + z^2$$

is obtained by

```
$$\eqalignno{x&=y+z\cr
    \noalign{\noindent{\rm and}}
    x^2&=y^2+z^2\cr}$$
```

Note the difference when `\eqalign` is used, TB ex19.14, and don't forget `\noindent`.[25]
I was trapped when I footnoted the text 'and'. The mark appeared on its own.

4.21 On your own.

Yes, you can be in complete control and escape the automatisms. From the math book I got the impression that `\eqalign` was used throughout, and all formulae were centered.

> 'For other displays, plain TEX provides `\displaylines`, which lets you display any number of formulae in any way you want, without any alignment.'

Incidental left justification can be obtained easily, TB194, ex19.16. The TEXfall is to adhere to `\displaylines` throughout a TEXscript, exercising typographic mark-up, at the price of cumbersome adaptation, and unnecessary exposure to the danger of being inconsistent.

Example (*Left now and then*)

$$P_\nu(\cos\theta) \qquad \text{(MD)}$$

is obtained via

```
$$\displaylines{\indent P_\nu(\cos\theta)
     \hfill\llap{(MD)}\cr}$$
```

Note however that in the MAPS of NTG `\parindent`=0.

4.22 Generality.

From TB375 we have

' The goal is to set TEX up so that the respective constructions \$\$$\alpha$\$\$, \$\$$\alpha$`\eqno`$\beta$\$\$, and \$\$$\alpha$`\leqno`$\beta$\$\$ will cause a macro `\generaldisplay` (in display) to be invoked, with `\eq` to be defined α; furthermore the test `\ifeqno` should be true when an equation number β is present, and `\ifleqno` should be true in the case of `\leqno`. When β is present, it should be stored in `\eqn`. Here α and β are arbitrary balanced token lists that don't contain either `\eqno` or `\leqno`.'

4.22.1 Lefties forever.

The pitfall for using left justification throughout the publication is that with short formulae the resulting pages look 'too white' at the right half.

Needed are the macros provided at TB376 augmented with a suitable `\generaldisplay` macro for left alignment, indented with parindent, and formulae numbers flushed right. *No* adaptation of the TEXscript![26]

$$f(x) = \begin{cases} 0, & \text{if} \quad x<0 \\ 1, & \text{if} \quad x\geq 0 \end{cases} \qquad \text{(Stepfunction)}$$

is obtained via, TB376

```
$$f(x)=\cases{0,&if\quad $x<0$\cr
             1,&if\quad $x\geq0$\cr}
  \eqno(\rm Step function)$$
%with declarations
\newif\ifeqno \newif\ifleqno
\everydisplay{\displaysetup}
\def\displaysetup#1$${
    \displaytest#1\eqno\eqno\displaytest}
\def\displaytest%
   #1\eqno#2\eqno#3\displaytest{%
   \if!#3!
   \ldisplaytest#1\leqno\leqno\ldisplaytest
   \else\eqnotrue\leqnofalse
        \def\eqn{#2}\def\eq{#1}%
   \fi
   \generaldisplay$$}
\def\ldisplaytest%
   #1\leqno#2\leqno#3\ldisplaytest{%
   \def\eq{#1}
   \if!#3!\eqnofalse
   \else\eqnotrue\leqnotrue\def\eqn{#2}\fi}
\def\generaldisplay{%
  \netdpw\hsize
  \advance\netdpw-\parindent%Compenstae
  \leftline{\indent$\displaystyle\eq$
  %No hfill, to allow eqn no in eqalign
  %to be right adjusted
  \ifeqno\hfill\llap{$\eqn$}\fi}
  %Termination display
}%end generaldisplay
```

Note the use of the parameter separators, called sentinels in traditional programming.
Partial solution.
For the case the TEXscript does not contain `\eqno` tags a simpler solution is provided in TB ex19.4. The solution given does not allow for `\eqalignno` to be used either.

[25] Again a sourse of confusion.
[26] Eqalignno can't be used, however, yielding an error message. Eqalignno can be adapted by: use `\vcenter`, give the first `\tabskip` the value zero, and provide the halign with the size `\displaywidth`−`\parindent`.

4.23 Graphs.

The TEXfall is that TEX is weak with respect to graphics. However, via the insert mechanism space can be left open to paste in figures and the like, made separately and independently by other tools. Electronic paste-up at the Postscript level is possible. To let text flow around a figure (open space) can be done. See Cork90 proceedings for details.
One easily dives into picture like environments, and then the too many pitfall may open up. In TB ex18.46 commutative diagrams are dealt with. Simple diagrams can be built from there. As example we take the diagram showing the calculation of the autocorrelation, either via $a_f = f \otimes f$, or by means of Fourier transform, followed by multiplication, and the inverse transform. The following is derived from TB ex18.46, 358, where $\mathcal{F}$ denotes Fourier transform and $\mathcal{F}^-$ the inverse Fourier transform.

$$\begin{matrix} f & \xrightarrow{\otimes} & a_f \\ \Big\downarrow\mathcal{F} & & \Big\uparrow\mathcal{F}^- \\ F(f) & \xrightarrow{\times} & \bigl(\mathcal{F}(f)\bigr)^2 \end{matrix}$$

obtained via

```
\def\llongrightarrow{\relbar\joinrel
        \relbar\joinrel\rightarrow}
$$\def\normalbaselines{
           \baselineskip20pt
           \lineskip3pt
           \lineskiplimit3pt}
\def\mapright#1{\smash{\mathop{
   \llongrightarrow}\limits^{#1}}}
\def\mapdown#1{\Big\downarrow
       \rlap{$\vcenter{\hbox{$#1$}}$}}
\def\mapup#1{\Big\uparrow
       \rlap{$\vcenter{\hbox{$#1$}}$}}
\matrix{f&\mapright\otimes&a_f\cr
      \mapdown{{\cal F}}&&\mapup{%
           {\cal F}\strut^{-}}\cr
     F(f)&\mapright\times&\bigl(
           {\cal F}(f)\bigr)^2\cr}
$$
```

Borceux(1990) has pointed out that diagonal as well as unequal length connectors are needed. The latter because of the unequal size of the 'knots'. The interested reader is referred to Borceux(1990). For curved connectors the Bezier technique might be used. $\mathcal{AMS}$-TEX is rich with respect to commutative diagrams, among others.

4.24 All in the family.

A fonts TEXfall is related to using a `\hbox{...}`. To quote from TB163

> 'But such uses of `\hbox` have two disadvantages: (1) The contents of the box will be typeset in the same size, whether or not the box occurs as a subscript; for example, '`$x_{\hbox{max}}$`' yields $x_{\rm max}$. (2) The font that's used inside `\hbox` will be the "current font," so it might not be roman.'

In the TEXscript I found `\def\Re{\hbox{Re}}`, adhering to disadvantage (2). See also TB ex19.1 for the unexpected italic result.

Plain provides basically (computer codern) roman, bold, slanted, text italic, typewriter type, math symbols and math extensions. Most fonts are available in 10pt, 7pt and 5pt. Especially the math extension is of interest, because of the composition possibilities. For example the 'n-sized' open parethesis is composed of the entries $'60 + ('102)^n + '100$. Because of mnemotechnique and because of the composition process, the symbols have been made available via names. The ordering is prompted by the functionality into, TB434 etc.
1. Lower case Greek letters,
2. Upper case greek letters,
3. Calligraphic capitals,
etc.
For example $f \mapsto g$ is obtained via `$f\mapsto g$`. BLUe does not have to worry about font tables, not even for symbols of varying size. Some understanding does not harm however. Curious are the names for $\wedge$, and $\vee$. No `\and`, respectively `\or`. This is understandable because `\or` is already in use in the case construct. The latter can be part of an expression, and therefore overloading is not possible for `\or`.

Now and then other symbols than those provided in the font tables of Appendix F are wanted. Occasionally these can be constructed, like the symbols denoting the sets of
natural numbers, IN,
integers, ZZ,
rational numbers, Q,
reel numbers, IR,
and complex numbers, C.
These are obtained by kerning — 'poor man's blackboard bold.' My version is

```
\def\N{{\rm I\kern-.5ex N}}
\def\Z{{\rm Z\kern-.9ex Z}}
\def\Q{{\rm\kern.2ex\vrule
    height1.3ex depth-.1ex
    width.4pt\kern-.7ex Q}}
\def\R{{\rm I\kern-.5ex R}}
\def\C{{\rm\kern.3ex\vrule
    height1.3ex depth-.1ex
    width.4pt\kern-.7ex C}}
```

Generally they are already available somewhere, see Quin(1990), and especially AMS(1990). The math book has used it is own brand of poor man's blackboard bold. Swanson(1986, 2.4.8e) advises just to use boldface.
I also needed $\gtrless$.

```
\def\gtrless{\mathrel{\vcenter{
   \hbox{$\buildrel\textstyle{>}
          \over{>}$}}}}
```

Note the use of `\textstyle`, otherwise the cramped textstyle is obtained.
In TEXHaX90.20, Duchier(1990) has published a macro for the external tensor product, $\boxtimes$.

A simplified version reads

```
\def\boxit#1{\vbox{\hrule\hbox{% Necessary!
       \vrule#1\vrule}\hrule}}
\def\boxtimes{\mathbin{\boxit{$\times$}}}
```

Fortunately, these kinds of symbols are now available in AMSfonts, for free.

In general one needs to know what is locally available and how it looks like. For the latter there exist a program called 'testfont,' which prints the font table with the symbols in it. Once the fonts have been selected they must be made known to TEX. Next one can integrate the fonts into families, such that automatically in displaystyle, textstyle, scriptstyle and scriptscriptstyle the right size appears. More perfect is integration into size-switching macros analogous to `\tenpoint`, `\ninepoint`, respectively `\eightpoint`, for the TEXbook itself, see Appendix E. The size-switching macros for the TB contain the families: `\itfam`, `\slfam`, `\ttfam`, and `\bffam`. How to do this is given on TB414.

Beyond the scope of this paper is construction of symbols via METAfont, as well as the system managers work to get fonts from elsewhere and install these.

5 The good news

In order to go forward classical items like: Better user interfaces, Education, Have it done, and Support, are needed.
Because of TEX, and e-mail, I could work in the spirit of Swanson(1986)

> 'Perhaps some day a typesetting language will become standardized to the point where papers can be submitted to the [publisher] from computer to computer via telephone lines. Galley proofs will not be necessary, but referees and/or copy editors could send suggested changes to the author, and he could insert these into the manuscript, again via telephone.'

5.1 Better ....

In contrast with public domain (plain) TEX the better user interfaces do cost. Appealing names are in use: The writer's workbench, Publishing Environment, and the new vogue DTP. Generally, they support inputting the (math)script and providing laser printer output. The user does not have to know that TEX is used as formatter, even worse, TEX is occasionally lacking. When TEX is used the TEX file can be accessed, generally.
The user interfaces I have seen, are weak when corrections have to be made. I also noticed limited context sensitivity. As example of the latter I asked a vendor to use a matrix as an integrand. The integral sign did not grow with the size of the matrix. I don't expect those systems to allow for explicit formatting commands. With nested parentheses, fences and the like, different sizes have to be ordered for explicitly, now and then. For example

$$\|a(x+y)\| \leq |a|\,\big\|\,x+y\,\big\| \leq |a|\,\big(\|x\|+\|y\|\big),$$

is a typist pitfall as well as a pitfall for automated SGML-based user interfaces, I presume.

5.2 A world of learning.

TEX can be learned from the TEXbook with a TEXnigma at hand. It is easier and more economical to be guided by an experienced teacher. TUG traditionally, and recently many LUGs, organize a variety of TEX, LATEX, and Metafont related courses. Although discussion has started about what the various courses should provide and how they should be related; no agreement has emerged of yet, see Childs(1989a,b), and van der Laan(1989). I agree with Martin(1990) that a class consisting of TEXnical typists needs another approach than a class of scientists. Everybody needs TEX etc. intelligent editors. I have heard of LATEX-taylored EMACS and of enhanced EDT, Williams and Hall(1990). Education is paramount. Why not release THE video tapes at cost? Imagine, PD TEX on your PC, the teaching on video, all that at home!
It might be clear that I still gaze at the quotations on TB159, ... somewhat in unbelieve.

> 'The learning time is short. A few minutes gives the general flavor, and typing a page or two of a paper generally uncovers most of the misconceptions. —Kernighan and Cherry, A system for Typesetting Mathematics (1975)'
>
> 'Within a few hours (a few days at most) a typist with no math or typesetting experience can be taught to input even the most complex equations. —Peter J. Boehm, Software and hardware considerations for a technical typesetting system (1976)'

5.3 Service.

A professional typist is better suited for typing mathscripts than an author, despite not being a mathematician and not understanding the contents. AMS provides TEX typing services, see AMS(1990). I conjecture that more such services are needed, at reasonable price.

The demand is not (yet) large, I guess, because authors consider it a challenge to TEX their documents themselves, at the expense of ample trial-and-error. Besides, the author understands what is going on, likes to remain in complete control, especially when the proofing is cumbersome because of the typist suffering from insufficient TEXnowledge.

On the other hand, most documents enjoy a local readership and the obtained (form) quality in print —via naive TEXing or via another non-optimal tool—is considered sufficient in relation to the contents, the document preparation know-how of the typist, the readership, and the life-time of the document.

When publishing an article or book is in sight, a publishing house might take over and provide professional typing service, if needed, apart from other quality warranting issues.

5.4 Lean upon.

Sooner or later typesetting challenges will be encountered. Of course one can puzzle and find out eventually oneself, but it is generally more economical to consult a guru, despite the salaries. TUGboat, and electronical digests, among others, pay attention to queries from their readers. Listservers (and the digests) opened the possibilty to query a community instead of one person. Besides, one can always have the problem 'turnkey'-solved by hiring a programmer-consultant.

Jam session

TEXing a mathscript, lacking format commands, is too difficult for a non-TEX-trained typist. Moreover the typing task is silently augmented because proofs are more difficult to provide. It is unclear in what way AMS TEX typing services fills up the gap. It is not true that once one can talk math by phone, one can TEX math. One must not only write e.g. ':', but also specify whether it is used as punctuation symbol or as an operator. A mathematician must be told to specify these kinds of things in his mathscript, providing guidance for the TEXist.
On the other hand authors must not persuade TEXists into the typography mark-up TEXfall by supplying underlining, bold etc. wishes.
For TEXing math, both the author and the TEXist need to be aware of the possibilities of TEX, and the consequences for the mathscript.

TEXophil advisors are considered harmful, demonstrating 'a little knowledge is dangerous.' The more so because only the best is good enough.

In the hands of mathematicians, TEX etc. is challenging and enslaving. From the math book I understand that using TEX costs already so much energy, that it is difficult to adhere to consistency. Early TEXscripts suffer from various TEXfalls. Nevertheless, the result in print is considered good enough, because of lack of better small-scale alternatives.

TEX is a wonderful, but unusual tool. It challenges. I admire the design. I never read a manual so many times. I love TEX! Grace to the indefinite lifetime, investing in learning TEX is worthwhile.
But, ... TEX is non-robust and error-prone. Beware! Because of the complexity, the freedom, and flexibility, augmented with unawarenes of typographic tradition, one can easily err —and, whether one likes it or not—err, and err again, DEK(1989).[27] And if ever

The quality of the results depends on what you,
yourself,
make out of it.

The difficulty with rule-books is that they lag behind new TEXnology.

References

[1] AMS A manual for authors of mathematical papers (1973): Booklet, 24p.

[2] AMS TEX Publishing Services (1990): Leaflet.

[3] $\mathcal{AMS}$-TEX (1990, 2.0): User's Guide.

[4] $\mathcal{AMS}$-TEX (1990, 2.0): Installation Guide.

[5] AMSFonts (1990, 2.0): User's Guide.

[6] AMSFonts (1990, 2.0): Installation Guide.

[7] Beeton, B.N. (1990): TUGboat production: TEX, LATEX and paste-up. Appendix S. Minutes 5^{th} NTG meeting. (Talk presented at SGML-TEX conference Groningen, 1990.)

[8] Borceux, F. (1990): De la construction de diagrammes. Cahiers GUTenberg, 5, 41–48.

[9] Cameron, P.J. (1990): Adventures of a new TEXnician. TEXline 12, 2–3.

[10] Cheswick, B. (1990): A permuted index for TEX and LATEX. CST145, AT&T Bell Laboratories.

[11] Childs, B. (1989): Teaching TEX. TUGboat 10, 2, 156–163.

[12] Childs, B. (1989): Answers to TEX tests. TUGboat 10, 3, 319–323.

[13] Duchier, D. (1990): TEX, external tensor product. TEXHaX90.20.

[14] Haeringen, H. van (1990): Priv. comm.

[15] Knuth, D.E. (1984): The TEXbook. Addison-Wesley. (TEX, frozen in version π, 3.1415... is treated in Vol A of Computers and Typesetting from the *ninth* printing onwards, and in TEXbook from *seventeenth* printing onwards. (And if you can't remember the numbers, look for the `\language` command in the index.)

[16] Knuth, D.E. (1985): Recipes and fractions. TUGboat 6, 1, 36–38.

[17] Knuth, D.E. (1989): The errors of TEX. Softw-Prac&Exp. 19, 7, 607–685.

[27] I wonder, if error-free TEXing can ever be attained. Once in a BLUe moon?

[18] Knuth, D.E, T. Larrabee, P.M. Roberts (1989): Mathematical writing. MAA Notes 14. Stanford University (STAN-CS-88-1193). (With contributions by P. Halmos, Leslie Lamport, Mary-Claire van Leunen, Nils Nilsson, Rosalie Stemer, Jeff Ullman, and Herbert S. Wilf.)

[19] Lamport, L. (1986): LaTeX, a document preparation system. User's guide & reference manual. Addison-Wesley.

[20] Martin, C.R. (1990): TeX for TeXnical typists. TUGboat, 11, 3, 425–428.

[21] Laan, C.G. van der (1989): Teaching TeX: critics and LaTeX proposal. Appendix V. Minutes 4^{th} NTG meeting.

[22] Nearing, J. (1989): Extended equation numbering in plain TeX. TUGboat, 10, 1, 82–88.

[23] Pittman, J.E. (1988): Equation numbering in plain TeX. TUGboat, 9, 3, 287–288.

[24] Quin, L.R.E. (1990): Summary of Metafont fonts available. TeXMaG, 4, 6.

[25] Spivak, M.D. (1986): The Joy of TeX. A gourmet guide to typesetting with the $\mathcal{AMS}$-TeX macro package. Addison-Wesley.

[26] Swanson, E. (1986): Mathematics into type. AMS. (Copy editing and proofreading of mathematics for editorial assistants and authors.)

[27] Temme. N.M. (1990): Speciale functies in de mathematische fysica. Epsilon 15. ISBN 90-5041-019-7. (Dutch)

[28] Whitney, R.F, B.N. Beeton (1989): TUGboat author's guide. TUGboat, 10, 3, 378–385.

[29] Wichura, M.J. (1990): Showing-off math macros. TUGboat 11, 1, 57–61.

[30] Williams, L, L. Hall (1990): Increased efficiency using advanced EDT editing features. TUGboat, 11, 3, 421–424.

Appendix

Contents Math into BLUes