Cahiers GUTenberg

MATH INTO BLUES : SING YOUR SONG

Kees VAN DER LAAN

Cahiers GUTenberg, nº 10-11 (1991), p. 147-170.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1991___10-11_147_0>

Math into BLUes*: sing your song†

Kees VAN DER LAAN

Hunzeweg 57, 9893PB, Garnwerd, The Netherlands
+31 5941 1525; *cgl@rug.nl*

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 TEX falls and their antidotes is discussed. At the end suggestions are given in order to lessen the difficulties.

Résumé. *Taper du TEXte mathématique n'est pas aussi simple que saisir du TEXte standard. Les formules doivent être traduites en commandes TEX. C'est l'idée de base de notre travail. Nous présentons divers systèmes de mise en pages de textes mathématiques, ainsi que les commandes TEX. Après avoir répondu à la question "pourquoi est-ce si difficile rentrer du TEXte mathématique ?", nous parlerons des lacunes de TEX et leurs paliatifs. Nous conclurons par des suggestions pour applanir divers problèmes.*

1. The bad news

1.1. 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, however. 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

*The fisrt publication of this paper appears in *TUGboat* vol. 12(3), Proceedings of the the TUG 1991 annual meeting. It is publish here with authorization of *TUGboat*'s editor.

†Note of the editor: with the author's agreement, only the second part of the full paper appears here. The fisrt part, called *Mourning*, occured fisrt in *Nederlandstalige TEX Gebruikersgroep*, Verlag 6e bijeenkomst, 91.1, pp. 57-74, 1991, and in *TUGboat* vol. 12(3), 1991.

sequencing when numbered formulae have to be omitted. This was done in the TEXscript. Nowadays retyping can be circumvented by automatic numbering, jeopardizing the formula-number tie which some authors have developed. The better alternative is not to think in number-formula ties, but in name-formula ties. How about that?

1.1.1. 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, *the TEXbook* ex. 19.6, 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

$$|x| = \begin{cases} x, & \text{if } x \geq 0 \\ -x, & \text{if } x < 0 \end{cases} \tag{1}$$

is TEXed by

```
\fct=0 \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[1], 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 (1a) is simply obtained via

```
$$|x|=\cases{
       x,& if $x\ge0$\cr
```

[1] 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.

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

Pittman (1988)[2] 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 `\(1)eqno` tag as after the second & of `\(1)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.[3]
The awareness that numbers should be typed after `\eqno`, or after the second & in `\eqalignno`, has to be applied anyhow.[4]

1.2. 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. Automatic numbering uses *one* counter. 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 \(name) to number)*

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

[2] 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 (*the TEXbook* p 208). 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!?!

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

[4] With automatic numbering the number is not in the TEXscript. When symbolic referencing is used the names are part of the TEXscript.

is obtained, with \forpyth linked to (2), via

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

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.[5]

An inconvenience is forward referencing. A suggestion how to handle this, is to print in the margin, at the place 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). The printing can be guarded by an \ifproof switch. 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, (1.2.3), via \forref{\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 and type the number instead of the ???. One never can tell.

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

1.3. 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

$$a^2 + b^2 = c^2 \tag{1.2.2}$$

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`[6] 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

```
$$a^2+b^2=c^2\eqno\labf\forpyth$$
```

References to the formula go via the name `\forpyth`, and the lay-out defined via the format style of the publication is obtained.[7] I chose to surround the number with parentheses. Fully automated symbolic reference systems are

[6] 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.

[7] 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.

part of LaTeX and $\mathcal{AMS}$-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!

1.4. We want more.

Sometimes more alignment positions than the one position provided by `\eqalign`, (or `\(l)eqalignno`), are wanted. See for example *the TeXbook* p 242, where it is suggested to use `\eqalign` repeatedly within one display, and ex. 22.9 for a general approach via `\halign`.[8] In the math book I encountered

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

$$\sin(z\sin\theta) = \qquad + 2\sum_{n=1}^{\infty} J_{2n+1}(z)\sin(2n+1)\theta$$

which was not appropriately aligned, possibly because of not using the empty formula. Appropriate input for a 2-alignment `\eqalign` reads

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

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

1.4.1. 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

[8] Repeated use of `\eqalign` suffers from non-guaranteed equal line distances over the various `\eqaligns`.

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{%\,  <---Inactivated!
   %Glue to be inactivated in 2-column 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

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

via[9]

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

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

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.

1.4.2. 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 (*the TEXbook* p 374). Another possibility is selection of the appropriate code via a case-like construction.

1.5. Looooooonnnnnggggggg.

For breaking 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 (*the TEXbook* p 195)[10].

For the second case TEX does not automatically break displayed formulae.[11] The reason is that displayed math is too complex to automate

[10] Not in subformulae.

[11] As a consequence switching for a math TEXscript from 1-column format into 2-column, or vice versa, is not that simple as descriptive mark-up people would like us to make believe.

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 might 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 '+' or '−'.

3. To use the free-format `\displaylines` command (see *the TEXbook* p 194).

1.5.1. Via consecutive displays.

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

1.5.2. 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 (*the TEXbook* p 196). In eqalign(no) terms: they can't be separated by '&.' The non-context dependent variants, `\Bigl(`, `\Bigr)` etc. of fixed size, 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.[12]

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

is obtained via[13]

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

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

Example (*Constant line distance)*

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

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,
```

[12] In the math book there was no squaring, the unequal line distances were clearly visible without it. I could not reproduce that.

[13] Look also at the prime-ry section.

```
                \smash{\Bigr)^2},\cr
   E_3(x)&=x^3-\sfr32x^2+\sfr14.\cr
}$$
```

The use of `\sfr` had no effect!

1.5.3. Either way.

The TEXbook p 196 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

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

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

1.5.4. Centered and right.

The TEXbook ex 19.17, 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 —without slashed fractions—had been chosen in the math book.

Example (*Splitted parts flushed right*)

$$
\begin{aligned}
u_k &= \frac{1}{(k+1)(2k+1)(4k+1)} \\
&= \frac{\frac{1}{3}}{k+1} - \frac{1}{k+\frac{1}{2}} + \frac{\frac{2}{3}}{k+\frac{1}{4}} \\
&= \frac{1}{3}\left(\frac{1}{k+1} - \frac{1}{k}\right) - \left(\frac{1}{k+\frac{1}{2}} - \frac{1}{k}\right) \\
&\qquad + \frac{2}{3}\left(\frac{1}{k+\frac{1}{4}} - \frac{1}{k}\right) \qquad (3)
\end{aligned}
$$

$$
\sum_{k=1}^{\infty} u_k = -\tfrac{1}{3}\psi(2) + \psi(1\tfrac{1}{2}) - \tfrac{2}{3}\psi(1\tfrac{1}{4})
$$

is obtained by

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

1.6. Interrupts.

Authors want to connect displayed, and aligned, formulae by texts. TEXnically this means the opposite: the alignment has to be interrupted. For example (*the TEXbook* p 193),

$$x = y + z$$

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 (*the TEXbook* ex 19.14), and don't forget `\noindent`.[14]
I was trapped when I footnoted the text 'and'. The mark appeared on its own.

1.7. 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, (*the TEXbook* p 194, ex. 19.16). 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

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

[14] Again a sourse of confusion.

1.8. Generality.

From *the TEXbook* p 375 we have

' The goal is to set TEX up so that the respective constructions $$α$$, $$α\eqnoβ$$, and $$α\leqnoβ$$ 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.'

1.8.1. 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, in 1-column format.

Needed are the macros provided at *the TEXbook* p 376 augmented with a suitable \generaldisplay macro for left alignment, indented with parindent, and formulae numbers flushed right. *No* adaptation of the TEXscript![15]

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

is obtained via (*the TEXbook* p 376)

```
$$f(x)=\cases{0,&if\quad $x<0$\cr
              1,&if\quad $x\geq0$\cr}
  \eqno(\rm Step function)$$
%with declarations
\newif\ifeqno \newif\ifleqno
\everydisplay{\displaysetup}
\def\displaysetup#1$${
    \displaytest#1\eqno\eqno\displaytest}
\def\displaytest%
```

[15] 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. Schwarz (1987) in section 3.3 proposes a more compact solution with \(l)eqno redefined. The 'lefties forever' pitfall is nicely demonstrated in the lay-out of the test examples given by Schwarz.

```
   #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 *the TEXbook* ex. 19.4. The solution given does not allow for `\eqalignno` —as is—to be used either.

1.9. 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.

Below the graphs given in the 'Am I blue'-section are discussed.

Commutative diagrams.
In *the TEXbook* p ex18.46 commutative diagrams are dealt with. Simple diagrams can be built from there. The diagram illustrates 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.
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'. As can be seen from the example, unequal length connectors can be made easily. The interested reader is referred to Borceux (1990), for a more user-friendly 'commutative diagram' package. For curved connectors the Bezier technique might be used.[16] $\mathcal{LA}_{\mathcal{M}}\mathcal{S}$-TEX, Spivak (1990), is rich with respect to commutative diagrams, among others.

Matrix icons with diagonal lines.
Upon Hendrickson (1985)'s `\diagline` some icon macros have been built. The line thickness of the diagonal lines differ from the h/v-rules, however, and no (nearly) vertical lines can be made.[17]

Rhombus scheme.
Difference schemes can be made analogously with the boxed diaglines.

1.10. All in the family.

A fonts TEXfall is related to using a `\hbox{...}`. To quote from *the TEXbook* p 163:

> '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_{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 *the TEXbook* ex. 19.1 for the unexpected italic result.

Plain provides basically (computer modern) 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

[16] Available along with LATEX, as `bezier.sty`.

[17] In the 'lines' font of LATEX some diagonal line elements are available, restricted to a handful of orientations.

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 *the TEXbook* p 434 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.

Now and then other symbols than those provided in the font tables of App F are wanted. Occasionally these can be constructed, like the symbols denoting the sets of

natural numbers	$\mathbb{N}$,
integers	$\mathbb{Z}$,
rational numbers	$\mathbb{Q}$,
reel numbers	$\mathbb{R}$,
and complex numbers	$\mathbb{C}$.

These are obtained by kerning —'poor man's blackboard bold.' My version[18] reads

```
\def\N{{\rm I\kern-.5ex N}}
\def\Z{{\rm Z\kern-.9ex Z}}
\def\Q{{\rm\kern.2ex\vrule
    height1.5ex depth-.1ex
    width.4pt\kern-.7ex Q}}
\def\R{{\rm I\kern-.5ex R}}
\def\C{{\rm\kern.3ex\vrule
    height1.5}iex depth-.05ex
    width.4pt\kern-.7ex C}}
```

[18] An instance of the well-known 20%–80% law: 20% of the energy for 80% of the results. See also Clark (1987).

Generally they are already available somewhere, see Quin (1990), and especially AMS (1990). The math book has used its 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{
   \vbox{\offinterlineskip
         \hbox{${>}$}\kern-.2ex
         \hbox{${<}$}}       }}}
```

Once again I TEXfalled. In first instance I used `\buildrel` which yielded the two stacked symbols too far apart, and the cramped textstyle for the upper symbol.

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{% <--!!!
       \vrule#1\vrule}\hrule}}
\def\boxtimes{\mathbin{\boxit{$\times$}}}
```

Fortunately, these kinds of symbols are now available in AMSfonts.[19]

In general one needs to know what is (locally) available and how it looks like. For the latter there exists 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`, and `\eightpoint`, for the TEXbook itself, see App E. The size-switching macros for the *the TEXbook* contain the families: `\itfam`, `\slfam`, `\ttfam`, and `\bffam`. How to do this is given on *the TEXbook* p 414. Another promising aspect of fonts is coined by Hoenig (1990). He uses Metafont for making his illustrations and accesses them via a font.

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.

[19] At handling costs.

2. 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.'

2.1. 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 an 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. For example

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

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

2.2. 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-tailored EMACS and of enhanced EDT, Williams and Hall (1990). Education is paramount. Why not release THE video tapes at costs? 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 *the TEXbook* p 159, ... 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)'

2.3. 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 content, 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.

2.4. 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 possibility 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 the 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 to fall 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, of mark-up varieties, and the consequences for the mathscript. Awareness of the seven classes for math characters is paramount, also when LATEXing math.

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 ♡ TEX! Grace to the indefinite lifetime,[20] 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 unawareness of typographic tradition, one can easily err —and, whether one likes it or not— err, and err again, DEK(1989).[21] And if ever

> The quality of the results depends on what you, yourself, make out of it.
> But, ... because of the power of TEX it is tempting to choose for too complicated representations, inhibiting readability.

Rule-books lag behind new TEXnology, ... alas.

Play it again, Sam

Temme communicated that an essential difficulty for math TEXscripts, is switching from 1 into 2 column format or vice versa.
Other members of the band were Amy Hendrickson, providing her macros, and Philip Taylor, for verifying that some drivers in the Netherlands yield wrong results, and that the TEXing was right afterall. Gerard van Nes blew his horn as well. Thanks folks!

References

[1] *AMS A manual for authors of mathematical papers*, booklet, 24p., 1973.

[2] *AMS TEX Publishing Services*, leaflet, 1990.

[3] $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX *User's Guide*, v. 2.0, 1990.

[4] $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX *Installation Guide*, v. 2.0, 1990.

[5] *AMSFonts User's Guide*, v. 2.0, 1990.

[6] *AMSFonts Installation Guide*, v. 2.0, 1990.

[7] B.N. BEETON *TUGboat production: TEX, LATEX and paste-up*, appendix S. Minutes, 5th NTG meeting. (talk presented at SGML-TEX conference Groningen), 1990.

[20] Plain TEX has been frozen into version π.

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

[8] F. BORCEUX, " De la construction de diagrammes", in *Cahiers GUTenberg*, no. 5, pp. 41–48, 1990.

[9] P.J. CAMERON, "Adventures of a new TEXnician", *TEXline* 12, pp. 2–3, 1990.

[10] B. CHESWICK, *A permuted index for TEX and* LATEX, CST145, AT&T Bell Laboratories, 1990.

[11] B. CHILDS, "Teaching TEX", in *TUGboat* 10(2), pp. 156–163, 1989.

[12] B. CHILDS, "Answers to TEX tests", in *TUGboat* 10(3), pp. 319–323, 1989.

[13] M. CLARK,"More symbols for TEX", in *TEXline* 5, pp. 7–8, 1987.

[14] D. DUCHIER, "TEX, external tensor product", in *TEXHaX* 90(20), 1990.

[15] H. VAN HAERINGEN, priv. comm., 1990.

[16] A. HENDRICKSON, "Some diagonal line hacks", in *TUGboat*, 6(2), pp. 83–86, 1985.

[17] A. HOENIG,"Labelling figures in TEX documents", in *TUGboat*, 12(1), pp. 125–128, 1990.

[18] D.E. KNUTH, *The TEXbook*, Addison-Wesley, 1984 (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).

[19] D.E. KNUTH, "Recipes and fractions", in *TUGboat*, 6(1), pp. 36–38, 1985.

[20] D.E. KNUTH, *The errors of TEX*, Softw-Prac&Exp. 19, 7, pp. 607–685, 1989.

[21] D.E. KNUTH, T. LARRABEE and P.M. ROBERTS, *Mathematical writing*, MAA Notes 14, Stanford University (STAN-CS-88-1193) (With contributions by P. HALMOS, L. LAMPORT, M.C. VAN LEUNEN, N. NILSSON, R. STEMER, J. ULLMAN and H.S. WILF).

[22] C.G. VAN DER LAAN, *Teaching TEX: critics and* LATEX *proposal*, Appendix V. Minutes 4^{th} NTG meeting, 1989.

[23] L. LAMPORT, LATEX, *a document preparation system. User's guide & reference manual*, Addison-Wesley, 1986.

[24] Ph. LOUARN, "Manipulons une matrice", in *Cahiers GUTenberg*, no 8, pp. 59–61, 1991 (his step-by-step explanation is very clear; the example does not contain braces connecting cells, or sub parts).

[25] C.R. MARTIN, "TEX for TEXnical typists", in *TUGboat*, 11(3), pp. 425–428, 1990.

[26] F. MITTELBACH, "E-TEX Guidelines for future TEX", in *TUGboat*, 11(3), pp. 337–345, 1990.

[27] J. NEARING, "Extended equation numbering in plain TEX", in *TUGboat*, 10(1), pp. 82–88, 1989.

[28] J.E. PITTMAN, "Equation numbering in plain TEX", in *TUGboat*, 9(3), pp. 287–288, 1988.

[29] L.R.E. QUIN, "Summary of Metafont fonts available", in *TEXMaG*, 4(6), 1990.

[30] H.R. SCHWARZ H. RUTISHAUSER and E. STIEFEL, *Numerik symmetrischer Matrizen*, Teubner, 1972 (German).

[31] N. SCHWARZ and J. KRIEGER, *Introduction to TEX*, Addison-Wesley, 1990.

[32] M.D. SPIVAK, *The Joy of TEX. A gourmet guide to typesetting with the AMS-TEX macro package*, Addison-Wesley, 1986.

[33] M.D. SPIVAK, *LAMS-TEX*, 1990.

[34] E. SWANSON "Mathematics into type", AMS, 1986 (copy editing and proofreading of mathematics for editorial assistants and authors).

[35] N.M. TEMME, *Speciale functies in de mathematische fysica*, Epsilon 15, ISBN 90-5041-019-7, 1990 (dutch).

[36] R.F. WHITNEY and B.N. BEETON, "TUGboat author's guide", in *TUGboat*, 10(3), pp. 378–385, 1989.

[37] M.J. WICHURA, "Showing-off math macros", in *TUGboat*, 11(1), pp. 57–61, 1990.

[38] J.H. WILKINSON, *The Algebraic Eigenvalue problem*, Clarendon Press, Oxford, 1965.

[39] L. WILLIAMS and L. HALL, "Increased efficiency using advanced EDT editing features", in *TUGboat*, 11(3), pp. 421–424, 1990.