Breaking equations

Michael Downes
American Mathematical Society
PO Box 6248
Providence, RI 02940
USA
mjd@ams.org

Introduction

Some of the inconvenient aspects of writing displayed equations in TEX are of such long standing that they are scarcely noticed any more except by beginning users. For example, if an equation must be broken into more than one line, `\left ... \right` constructs cannot span lines. This is a report on a new LATEX package called `breqn` that substantially eliminates many of the most significant problems (described at length in the next section). Its main goal is to support automatic linebreaking of displayed equations, to the extent possible within the current limitations of TEX and LATEX. Such linebreaking cannot be done, however, without substantial changes under the hood in the way math formulas are processed. Some of the changes are radical enough that it would be more natural to do them in LATEX3 than in LATEX2e — e.g., for LATEX3 there is a standing proposal to have nearly all nonalphanumeric characters be active by default; having `^` and `_` active this way would have eased some implementation problems. Using the package in LATEX2e is possible, with some extra care.

Current shortcomings in LATEX equation handling

Hindrances for authors The following difficulties affect authors using the standard LATEX `equation` and `eqnarray` environments. Some of them are ameliorated by the use of the `amsmath` package. (The first four also apply for plain TEX; and the main reason the next three don't apply as well is that plain TEX replaces them with a more substantial shortcoming: no automatic numbering at all.)

1. Line breaks must be inserted by hand.
2. Breaks are sensitive to changes in fonts or column width; and altering them is onerous.
3. A break within `\left-\right` delimiters requires extra work, especially if there is any difficulty getting the sizes to match.
4. Use of `\halign` freezes available shrink. Thus, for example, suppose that a given formula fits within the column when done with an `equation` environment; the exact same formula may fail to fit when done with an `eqnarray` environment, because `eqnarray` uses `\halign` internally.
5. Punctuation at the end of an equation logically belongs with the surrounding text but it must be entered with the body of the equation in order to print in the right place. This discord is especially noticeable when promoting formulas from inline math to display.
6. A numbered equation that takes several lines in an `eqnarray` requires awkward use of `\nonumber` to keep from getting a number on each line.
7. Numbers may overlap the equation body without warning (in `eqnarray` and similar structures).
8. There is no easy way to specify a variant equation number for an individual equation.
9. The space around equal signs in `eqnarray` is noticeably larger than the normal spacing for such symbols. This looks bad when adjacent equations are done one with `equation` and one with `eqnarray`.
10. There is no easy way to center an equation number across multiple lines of a broken equation. Some users manage to infer that `array` is the natural approach for this, but a plain `array` has various spacing faults for this purpose, and uses text style instead of display style for the contents.
11. There is no easy way to add a frame around the body of an equation (with or without including the equation number). You can just about do it with a one-line equation if there's no number and if you know about `\displaystyle`. But with multiline equations it's rather more difficult (use of `array` is again indicated, but it brings all the deficiencies cited in the preceding item).

The bosonic part of the action takes the form

$$I = I_{00} + I_{01} + I_{10} + \cdots \tag{14}$$

where

$$\begin{aligned} I_{00} &= \frac{(2\pi)^3}{\alpha'^2} \int d^6x \sqrt{-G} e^{-\Phi} \left[R_G + G^{MN} \partial_M \Phi \partial_N \Phi \right. \\ &\quad \left. - \frac{1}{12} G^{MQ} G^{NR} G^{PS} H_{MNP} H_{QRS} \right] \end{aligned} \tag{15}$$

where $M, N = 0, ..., 5$ are spacetime indices.

```
\begin{eqnarray}\nonumber
I_{00}&=& \frac{(2\pi)^3}{\alpha^{\prime 2}}
  \int d^6x \sqrt{-G}e^{-\Phi} \left[R_G+G^{MN}
  \partial_M\Phi\partial_N\Phi\right.\\
&& \left.-\frac{1}{12}G^{MQ}G^{NR}G^{PS}H_{MNP}H_{QRS}\right]
\end{eqnarray}
```

Figure 1: Typical equation problems in ordinary LaTeX: (a) different spacing around the equals signs in (14) and (15) because one uses `equation` and the other uses `eqnarray`; (b) equation (15) is a single equation but because it covers two lines `\nonumber` must be used on the first line; (c) and then the number is not vertically centered on the entire equation; (d) the sizes of `\left [` in the first line and `\right ]` in the second line don't match (they could be made to match, with extra work); and (e) the minus sign at the beginning of the second line is getting (wrong) unary spacing. This example is from (Duff, Minasian, and Witten, 1996), with only a couple of minor adaptations.

Issues of typeset quality

1. Symbol spacing tends to go wrong at the start of continuation lines (cf (Kopka and Daly, 1995, §5.4, p 136)). When a line break is taken before a binary infix operator, the operator will typically get unary operator spacing, though it shouldn't. (See Figure 1.)
2. Use of `\halign` (as in `eqnarray`) keeps the display short spaces from ever being applied, even when a group of equations begins with a short equation that would get the reduced spacing if it occurred by itself.
3. No distinction is made between consecutive, separate equations and lines of a single, broken equation.
4. Standard methods for reducing the type size of an individual equation all have adverse side effects; typically, the wrong line-spacing gets used for the text preceding the equation.
5. When a multiline block of text is displayed and numbered like a formula, the base-to-base spacing above and below doesn't work quite right.

Features and misfeatures of the `amsmath` package

As compared with the standard LaTeX facilities for equations, the `amsmath` package addresses some of the problems mentioned above, but introduces a few new misfeatures of its own.

Features

1. The `split` and `multline` structures match up better with the logical structure of individual equations and equation groups.
2. The multiple-equation environments `align`, `gather`, etc., use the correct spacing for equal signs.
3. The `\tag` command makes it easy to get variant equation numbers.
4. Overlap of the equation number on the equation body is mostly prevented.
5. There is more control over page breaking.
6. Environments `aligned`, `cases`, etc., can be used as building blocks in building up more complicated displays.

Misfeatures

1. For technical reasons, abbreviations like `\bal`, `\eal` for `\begin{align}`, `\end{align}` don't work.
2. There are inconsistencies between the `multline` and `split` environments; for example, the equation number for `multline` does not get centered the way it does for `split`.
3. The `equation` environment is implemented as a subcase of the `gather` environment, which means that it inherits the `\halign` deficiencies mentioned above: horizontal shrink isn't used; the short skip possibility is disregarded; and

it also is rendered unabbreviable, as described above. (Although work-arounds exist, they aren't particularly well known and deviate from canonical LaTeX syntax.)

TeXnical difficulties Looking at the above lists of deficiencies, one may well wonder why they have not been better addressed before now, more than ten years after LaTeX (and AMS-TeX) were first developed (1983–1985). One of the contributing reasons, however, is surely the intransigence of the TeXnical difficulties involved.

- TeX lacks low-level support for typical display-breaking conventions; for example, break penalties are provided only on the right side of math-bin and mathrel symbols.
- Math/text defaults for `$$` and `\eqno` are backwards. If TeX's display structure had been envisioned as a purely typographical device and started out in text mode rather than in math mode, a number of difficulties would never arise. The same can be said for `\eqno`. Thus providing a simple way such as `$$` to start a math display would have been better relegated to the macro level, not hardwired into the primitive display mechanism.
- `\left`–`\right` subformulas are wrapped in an unbreakable box.
- Displayed equation macros have been mainly written towards the *typographical* structure embodied in TeX's `$$` mechanisms, instead of towards the actual logical structure of the material (distinguishing single equations from equation groups, intra-equation punctuation from external punctuation and so on).

Features of the `breqn` package

- Overlong equations can be broken automatically to the prevailing column width following standard conventions. There will always be some equations that need special line-breaking attention from the author, but for those that don't, the process is highly automated, including standard indention conventions, avoiding overlap with the equation number, and so on.
- Line breaks can be specified in a natural way even within `\left` ... `\right` delimiters. Preferred but nonmandatory breakpoints can be specified within equations by `\linebreak` with an optional argument, as usual.
- Separate equations in a group of equations are written as separate environments instead of being bounded merely by `\\` commands. This simple change dispels, as a side effect, the problem of wrong math symbol spacing at the beginning of continuation lines.
- Horizontal shrink is made use of whenever feasible (most other equation macros are unable to get at it when it occurs between `\left` ... `\right` delimiters or in any sort of multiline structure). (However, shrinkable space inside fractions, square roots, overlined quantities, etc., is not unfrozen by this package. That is a less tractable problem.)
- The `\abovedisplayshortskip` is used when applicable (other equation macros fail to apply it in equations of more than one line).
- Displayed 'equations' that contain mixed math and text, or even text only, are handled naturally by means of a `dtext` environment that starts out in text mode instead of math mode.
- The punctuation at the end of a displayed equation can be handled in a natural way that makes it easier to promote or demote formulas from/to inline math, and to apply special effects such as omitting the punctuation, as favored by some of the more progressive book designers.
- Equation numbering is handled in a natural way, with all the flexibility of the `amsmath` package (features like `\tag` and `subequations` are provided under different guises) and with no need for a special `\nonumber` command.
- Unlike the `amsmath` equation environments, the `breqn` environments can be called through user-defined abbreviations such as `\beq` ... `\eeq`.
- It is easy to set local options for a single equation environment, e.g., changing the type size or adding a frame.
- It is possible to specify different vertical space values for the space between lines of a long, broken equation and the space between separate equations in a group of equations.
- Page breaking before, within, and after displayed math formulas is subject to more sophisticated control than it is with other extant equation macros.
- A rather noteworthy 'feature': as it pushes the envelope of what is possible within the context of LaTeX2e, the `breqn` package will tend to break other packages when used in combination with them, or to fail itself, when there are any areas of internal overlap; successful use may in some cases depend on package loading order.

Environments and commands All of the following environments take an optional argument for applying local effects such as changing the typesize or adding a frame to an individual equation.

`dmath` Like `equation` but supports line breaking and variant numbers.

`dmath*` Unnumbered; like `displaymath` but supports line breaking

`dtext` Like `equation` but starts out in text mode

`dtext*` Unnumbered variant of `dtext`

`dgroup` Like the `align` environment of `amsmath`, but with each constituent equation wrapped in a `dmath`, `dmath*`, `dtext`, or `dtext*` environment instead of being separated by `\\`. The equations are numbered with a group number. When the constituent environments are the numbered forms (`dmath` or `dtext`) they automatically switch to 'subequations'-style numbering, i.e., something like (3a), (3b), (3c), ..., depending on the current form of non-grouped equation numbers. See also `dgroup*`.

`dgroup*` Unnumbered variant of `dgroup`. If the constituent environments are the numbered forms, they get normal individual equation numbers, i.e., something like (3), (4), (5), ... Numbered and unnumbered forms can be mixed in the natural way, as needed.

`darray` Similar to `eqnarray` but with an argument like `array` for giving column specs. Automatic line breaking is not done here.

`darray*` Unnumbered variant of `darray`, rather like `array` except in using `\displaystyle` for all column entries.

Restrictions on the use of the `breqn` package

math symbol setup In order for automatic line breaking to work, the operation of all the math symbols of class 2, 3, 4, and 5 must be altered (relations, binary operators, opening delimiters, closing delimiters). This is done by an auxiliary package `flexisym`. If you use anything other than the standard LaTeX set of math symbols from the fonts `cmex`, `cmsy`, `cmmi`, you will probably need to do some configuration of the load process for the `flexisym` package.

subscripts and superscripts Because of the changes to math symbols of class 2–5, writing certain combinations such as `^+` or `_\pm` or `^\geq` without braces will lead to error messages; in general, except for letters and digits, any single math symbol must be enclosed in braces when sub or superscripted: `^{+}`, `_{\pm}`, `^{\geq}`. Actually, there is no visible sanction in the LaTeX book for omitting such braces: it uniformly shows braces in all sub and superscript examples — perhaps because the problem described here already exists in standard LaTeX to a lesser extent, as you may know if you ever tried `^\neq` or `^\cong` (in the case of `\neq` the symbol simply prints incorrectly, *without* giving an error message).

Grinchuk (Grinchuk, 1996) encountered the same sort of technical complications regarding braces around superscripted math symbols in his efforts to support Russian-style formula breaks as I did, and thinks (as I do) that turning the `^` and `_` characters into active characters might be the best user-friendly solution.

Breaking long equations

Let's begin by reviewing some first principles. Some facets of the typesetting task for displayed equations are so 'self-evident' that they rarely receive any explicit attention, but only by paying explicit attention to everything can we be sure of getting a strong grasp of the whole picture, and of the relative significance of various constraints.

Displayed mathematical expressions A *displayed mathematical expression*, commonly referred to as a *displayed formula* or *displayed equation*, is a sentence fragment whose nucleus is some sort of math formula — not necessarily containing an actual equals sign — and that stands by itself in the text column with extra space above and below. The purpose of treating the expression this way might be

- to emphasize it
- to facilitate reference to it
- to suggest its structure by the way the lines are broken and indented, or
- simply to avoid breaking it, if it contains no acceptable break points and leaving it in-line cannot be done without adversely affecting the inter-word spacing of the text.

On the subject of breaking equations, there is a statement in *The TeXbook* (Knuth, 1986, Chapter 19, p 195) suggesting (with all the weight of Knuth's mathematical typesetting knowledge behind it) that attempting to do it automatically would be, er, well, foolish:

> It's quite an art to decide how to break long displayed formulas into several lines; TeX never attempts to break them, because no set of rules is really adequate. The author of a mathematical manuscript is generally the best judge of what to do, since

> break positions depend on subtle factors of mathematical exposition. For example, it is often desirable to emphasize some of the symmetry or other structure that underlies a formula, and such things require a solid understanding of exactly what is going on in that formula.

My motivation, however, for seeking to provide automatic linebreaking in the `breqn` package is the observation that among the 10% or so of equations that need to be broken, two-thirds have entirely conventional breaks, and many of the remaining can be handled nicely by having the author indicate preferred line breaks instead of forcing line breaks. And indeed, the continuation after the above quote is less pessimistic:

> Nevertheless, it is possible to state a few rules of thumb about how to deal with long formulas in displays

Fitting a displayed equation into the available text width In the following examples the gray blocks demarcate the nominal column width within which the displayed expression must fit.

Example 1. A substantial majority of equations fit comfortably within the available width, even in relatively complex mathematical material.

$$a = b + c$$

Example 2. As an equation grows longer it begins to approach the point where not all the material will fit on a single line.

$$a = b + c + d + e + f$$

Example 3. If the equation is just slightly wider than will fit, the preferred strategy is to squeeze it a little by reducing the space around binary operator symbols such as $+$, $-$, or $\otimes$.

$$a = b+c+d+e+f+g$$

Compare the spaces around the plus signs.

You might think that reducing the space around relation symbols would be a good idea as well; however, the default math space settings in LaTeX do not have any shrinkability in the space for relation symbols.

Example 4. Shrinkable spaces inside a subscript, superscript, fraction, root, or other special construct are frozen and cannot shrink. There is a simple reason for this: TeX cannot print any sort of object that breaks out of the basic linear symbol stream into two dimensions, except by putting it into a vbox or by raising or lowering a box from the current baseline. In either case, the material winds up inside a box, and boxes have a fixed width determined at the time of construction. A fraction consists of a numerator box stacked atop a denominator box. Thus spaces within the numerator and denominator don't shrink, even when that might be useful to help an expression fit on a single line. The following equation contains no more material, horizontally speaking, than the previous one; it is the lack of shrinking that makes it overrun the right margin.

$$a = \frac{b + c + d + e + f + g}{2}$$

Example 5. In subscripts and superscripts mathrel and mathbin spaces are entirely *omitted*! So 'to shrink or not to shrink' isn't even a question.

$$a = x_{b+c+d+e+f+g} - 2$$

When shrinking isn't enough

Example 6. Suppose that an equation still exceeds the available width after all available shrink capacity has been used up. Then the obvious way to deal with the over-wide condition is to break off the tail end of the equation and move it down to the next line. Commonly the break is chosen to fall just before a binary operator such as $+$ or $-$.

$$\begin{aligned} a &= b + c + d + e \\ &\quad + f + g \end{aligned}$$

In some publishing traditions, the binary operator symbol is repeated before and after the break (Grinchuk, 1996):

$$\begin{aligned} a &= b + c + d + e + \\ &\quad + f + g \end{aligned}$$

The chief feature of the `breqn` package is that long equations will break automatically at the conventional places and the continuation lines will be indented by the conventional amounts (configurable). The general idea is to set the equation as a special sort of TeX paragraph, very much like a list item (with ragged-right, rather than justified, text). Compare the outlines of

1. A typical list item, with the text set ragged right instead of justified.

and

$$\begin{aligned} \sigma(2^{34} - 1, 2^{35}, 1) = -3 &+ (2^{34} - 1)/2^{35} + \cdots \\ &+ 7/2^{35}(2^{34} - 1) \cdots \end{aligned}$$

Example 7. Breaking an equation anywhere between a pair of delimiters is usually less desirable, by an order of magnitude, than breaking elsewhere. Bad:

$$a = b + (c + d) + (e + f) + g$$

Better:

$$a = b + (c + d) + (e + f) + g$$

But TEX does not provide any native way to test whether a given point falls between delimiters or not. So between-delimiter breaks can be more highly discouraged only if the delimiter symbols are themselves programmed to support such a test. This is in fact what the `breqn` package does.

Example 8. By giving suitable LATEX definitions to delimiter symbols such as `(`, `)`, `\langle`, `\rangle`, etc., it is possible to suppress line breaks within paired delimiters. With vert bars, however, there is a bit of a problem. The standard method of entering vert bar symbols in a document is just to use the ASCII vert bar character from the keyboard. When the verts are paired, mathematicians have little trouble telling which ones are openers and which ones are closers; but computer software such as LATEX doesn't have the discriminatory powers of a human mathematician. It is difficult, for example, for LATEX to discern that a line break such as the following is undesirable:

```
a = b +|c +d| +|e +f| +g
```

$$a = b + |c + d| + |e + f| + g$$

Example 9. Without any explicit indication of directionality, we might envision programming some sort of heuristic choice mechanism into the TEX definition of the vert bar, such as 'the first vert that comes along is an opener; the next one is a closer, and alternate thereafter'. Unfortunately, there are no heuristics for this particular problem that are actually good enough in practice.

If we have directional vert bar symbol commands `\lvert`, `\rvert`, `\lVert`, `\rVert`, we could rewrite the earlier example as follows, allowing LATEX to easily tell where line breaks should be discouraged more strongly.

```
a = b +\lvert c +d\rvert
    +\lvert e +f\rvert +g
```

$$a = b + |c + d| + |e + f| + g$$

Even better is for paired use of the vert bars to be encapsulated in a macro with some meaningful name chosen by the author:

```
\newcommand\abs[1]{\lvert#1\rvert}
```

Both the `amsmath` package and the `breqn` package (via `flexisym`) provide `\lvert` etc.

Example 10. But then consider the following example. Which break looks better: This one?

$$a = b + (c + d + e + f + g)$$

Or this one?

$$a = b + (c + d + e + f + g)$$

Example 11. For completeness I should have pointed out an intermediate form with the second line indented to the usual position:

$$a = b + (c + d + e + f + g)$$

This illustrates the point that when a line break is taken between delimiters, we generally prefer to have the delimiters clear their contents: all material within the scope of an opening delimiter should be indented at least as much as the delimiter itself, and closing delimiters should fall further to the right than the rightmost point of the material they encompass.

Example 12. Some displayed mathematical expressions don't include any relation symbol. Then the usual practice is to indent subsequent lines by a fixed amount, say one or two quads:

$$a + b + c + e + f + g$$

Example 13. An alternative indention strategy that may be used for two-line equations, especially when the first line contains no natural alignment point, is to start the first line at (near) the left margin and end the last line at (near) the right margin. In the `amsmath` package this functionality is provided by the `multline` environment (Downes, 1995). This layout may seem more well-balanced than the other if the second line is much shorter than the first.

$$a + b + c + d + e + f + g$$

Example 14. Even if a relation symbol is present, the distribution of material between the left-hand side and the right-hand side might make a break in the former more reasonable than a break in the latter. Compare

$$(a + b + c + d + e + f + g) = \alpha$$

and

$$(a+b+c+d \\ +e+f+g)=\alpha$$

Example 15. A displayed formula may have the form 'A and B':

```
Y_n=Y_k\quad\text{and}\quad Z_n=Z_k
```

$$Y_n = Y_k \quad \text{and} \quad Z_n = Z_k$$

This is a subcase of a more general list pattern that may take various other forms:

$$Y_n = Y_k, \qquad Z_n = Z_k$$
$$X_n = X_k, \quad Y_n = Y_k, \quad \text{and} \quad Z_n = Z_k$$
$$X_n = X_k, \qquad Y_n = Y_k, \qquad Z_n = Z_k$$

Example 16. It is not uncommon for a displayed equation to have an attached condition or qualifier (Knuth, 1986, Chapter 19, p 185):

$$Z_n = X_n \bmod q \qquad \text{for all } n \geq 0.$$

In most cases, if there is not enough room for everything to fit on one line, breaking between the assertion and the condition is better than breaking elsewhere. Compare

$$Z_n = X_{1n} + \cdots + X_{kn} \bmod q \\ \text{for all } n \geq 0.$$

and

$$Z_n = X_{1n} + \cdots \\ + X_{kn} \bmod q \quad \text{for all } n \geq 0.$$

The `breqn` package provides a `\condition` command for such material:

```
\begin{dmath*}
Z_n=X_{1n} +\cdots+X_{kn} \bmod q
\condition[]{for all $n\geq 0$}
\end{dmath*}.
```

By default a comma is added by the `\condition` command, but that can be overridden by an optional argument (in this example, empty).

Example 17. In a similar but rarer construction, the extra material on a line is some sort of annotation rather than a condition — i.e., it is not essential to the truth of the assertion.

$$Z_n = X_n \bmod q \qquad \text{(cf. [KF79])}$$

Algorithm for breaking equations The algorithm used by the `breqn` package for finding the optimal arrangement of an equation is necessarily rather complex. A portion of it is shown in Figure 2.

Some examples

Now consider Equation Example A. It doesn't quite fit in *The TEXbook*'s column width. Knuth suggests

```
$$\eqalign{\sigma(2^{34}-1,2^{35},1)
  &=-3+(2^{34}-1)/2^{35}
    +2^{35}\!/(2^{34}-1)\cr
  &\qquad+7/2^{35}(2^{34}-1)
    -\sigma(2^{35},2^{34}-1,1).\cr}$$
```

with the comment 'The idea is to treat a long one-line formula as a two-line formula, using `\qquad` on the second line so that the second part of the formula appears well to the right of the = sign on the first line.'

With the `amsmath` package the treatment would be nearly identical, but using the `split` environment instead of `\eqalign`:

```
\begin{equation*}\begin{split}
\sigma ...
    ... ,1).
\end{split}\end{equation*}
```

With the `breqn` package it's all automatic: the contents of the equation are written exactly the same as they would be if no line break were needed — no ampersands, no qquads, no break-here commands:

```
\begin{dmath*}
\sigma(2^{34}-1,2^{35},1)
  =-3+(2^{34}-1)/2^{35}+2^{35}\!/(2^{34}-1)
   +7/2^{35}(2^{34}-1)
   -\sigma(2^{35},2^{34}-1,1)
\end{dmath*}.
```

The first relation symbol is taken to indicate the primary alignment point, and if the total width is greater than column width, extra lines will be aligned and indented according to the class of symbol that immediately follows the break.

In a similar example (Knuth, 1986, Exercise 19.17), the point of the exercise is adding the equation number. In plain TEX this requires switching from `\eqalign` to `\eqalignno` and taking a bit of extra care with the horizontal spacing (the `\quad` in the last line).

```
$$\eqalignno{x_nu_1+\cdots+x_{n+t-1}u_t
   &=x_nu_1+(ax_n+c)u_2+\cdots\cr
   &\qquad+\bigl(a^{t-1}x_n+c(a^{t-2}
    +\cdots+1)\bigr)u_t\cr
   &=(u_1+au_2+\cdots+a^{t-1}u_t)x_n
    +h(u_1,\ldots,u_t). \quad&(47)\cr}$$
```

With the `breqn` package, the equation number is automatically placed at the location decreed by the documentclass (left, right, top, middle, bottom) and the `dmath` environment contains only the body of the

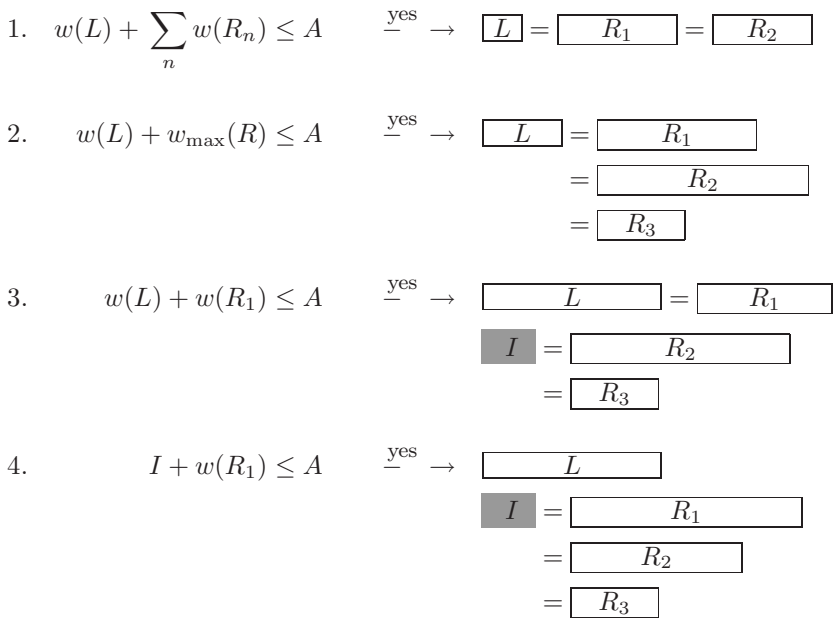

Figure 2: Equation-breaking algorithm: w width, L left-hand side, R right-hand side, A available width, I indent

---

Equation Example A. This equation (Knuth, 1986, Chapter 19, p 195) remains just a few characters too wide even after shrinking the spaces around the + and − symbols.

$\sigma(2^{34}-1, 2^{35}, 1) = -3+(2^{34}-1)/2^{35}+2^{35}/(2^{34}-1)+7/2^{35}(2^{34}-1)-\sigma(2^{35}, 2^{34}-1, 1).$

Knuth discussed how to break the equation using `\eqalign`, with the following result:

$$\sigma(2^{34}-1, 2^{35}, 1) = -3 + (2^{34}-1)/2^{35} + 2^{35}/(2^{34}-1) + 7/2^{35}(2^{34}-1) - \sigma(2^{35}, 2^{34}-1, 1).$$

Equation Example B. A change in the representation of the fractions would allow the entire equation of Equation Example A to fit on a single line:

$$\sigma(2^{34}-1, 2^{35}, 1) = -3 + \frac{2^{34}-1}{2^{35}} + \frac{2^{35}}{2^{34}-1} + \frac{7}{2^{35}(2^{34}-1)} - \sigma(2^{35}, 2^{34}-1, 1)$$

Most mathematicians would probably find this alternative more readable, as well.

---

equation. This has the potential to save authors a lot of work if they ever have to switch a book from one book design to another one that has different equation numbering conventions.

```
\begin{dmath}
x_nu_1+\dotsb+x_{n+t-1}u_t
  =x_nu_1+(ax_n+c)u_2+\dotsb
   +\bigl(a^{t-1}x_n
   +c(a^{t-2}+\dotsb+1)\bigr)u_t
  =(u_1+au_2+\dotsb+a^{t-1}u_t)x_n
   +h(u_1,\dotsc,u_t)
\end{dmath}.
```

An interesting feature of the next example (Knuth, 1986, Exercise 19.9) is that it is the second relation symbol, not the first one, to which the remaining relation symbols are aligned.

$$\begin{aligned} T(n) \le T(2^{\lceil \lg n \rceil}) &\le c(3^{\lceil \lg n \rceil} - 2^{\lceil \lg n \rceil}) \\ &< 3c \cdot 3^{\lg n} \\ &= 3c\, n^{\lg 3}. \end{aligned}$$

```
$$\eqalign{T(n)\le T(2^{\lceil\lg n\rceil})
    &\le c(3^{\lceil\lg n\rceil}
      -2^{\lceil\lg n\rceil})\cr
    &<3c\cdot3^{\lg n}\cr
    &=3c\,n^{\lg3}.\cr}$$
```

With the amsmath package that would be written

```
\begin{equation*}
\begin{split}
T(n)\le T(2^{\lceil\lg n\rceil})
  &\le c(3^{\lceil\lg n\rceil}
    -2^{\lceil\lg n\rceil})\\
  &<3c\cdot3^{\lg n}\\
  &=3c\,n^{\lg3}.
\end{split}
\end{equation*}
```

Using the dmath* environment you would use a \> command (note — not ampersand) to mark the relation symbol as the preferred alignment point:

```
\begin{dmath*}
T(n)\le T(2^{\lceil\lg n\rceil})
  \>\le c(3^{\lceil\lg n\rceil}
    -2^{\lceil\lg n\rceil})
  <3c\cdot3^{\lg n}
  =3c\,n^{\lg3}
\end{dmath*}.
```

Groups of equations

For an unaligned group of equations (see Equation Example C), Knuth recommended \displaylines:

```
$$\displaylines{%
  \hfill x\equiv x;\hfill\llap{(1)}\cr
  \hfill\hbox{if}\quad x\equiv y\quad
   \hbox{then}\quad y\equiv x;\hfill
   \llap{(2)}\cr
  \hfill\hbox{if}\quad x\equiv y\quad
   \hbox{and}\quad y\equiv z\quad
   \hbox{then}\quad x\equiv z.\hfill
   \llap{(3)}\cr}$$
```

With the amsmath package, the appropriate environment is gather:

```
\begin{gather}
  x\equiv x;\\
  \text{if}\quad x\equiv y\quad
    \text{then}\quad y\equiv x;\\
  \text{if}\quad x\equiv y\quad
    \text{and}\quad y\equiv z\quad
    \text{then}\quad x\equiv z.
\end{gather}
```

With the breqn package, it would be written as

```
\begin{dgroup*}[aligned={F}]
\begin{dmath}
  x\equiv x
\end{dmath};
\begin{dmath}
  \text{if}\quad x\equiv y\quad
    \text{then}\quad y\equiv x
\end{dmath};
\begin{dmath}
  \text{if}\quad x\equiv y\quad
    \text{and}\quad y\equiv z\quad
    \text{then}\quad x\equiv z
\end{dmath}.
\end{dgroup*}
```

A simple example of aligned equations (Knuth, 1986, Chapter 19, p 190):

```
$$\eqalign{
  X_1+\cdots+X_p&=m,\cr
  Y_1+\cdots+Y_q&=n.\cr
}$$
```

In LaTeX that would be written

```
\begin{eqnarray*}
  X_1+\cdots+X_p&=&m,\\
  Y_1+\cdots+Y_q&=&n.
\end{eqnarray*}
```

With the breqn package it would be written

```
\begin{dgroup*}
\begin{dmath*}
  X_1+\dotsb+X_p = m
\end{dmath*},
\begin{dmath*}
  Y_1+\dotsb+Y_q = n
\end{dmath*}.
\end{dgroup*}
```

Here is an equation group with a single common number (Knuth, 1986, Exercise 19.10):

```
$$\eqalign{%
  P(x)&=a_0+a_1x+a_2x^2+\cdots+a_nx^n,\cr
  P(-x)&=a_0-a_1x+a_2x^2-\cdots
        +(-1)^na_nx^n.\cr}\eqno(30)$$
```

In LaTeX this would need to be done with the array environment.

```
\begin{equation}
\begin{array}{rcl}
  P(x)&=&a_0+a_1x+a_2x^2+\cdots+a_nx^n,\\
  P(-x)&=&a_0-a_1x+a_2x^2-\cdots
          +(-1)^na_nx^n.
\end{array}
\end{equation}
```

But without further adjustments, spacing around the equals signs would be egregiously large, as in the eqnarray environment, and the interline spacing would be too small, and so on (see Equation Example D). Using the breqn package:

```
\begin{dgroup}
\begin{dmath*}
  P(x)=a_0+a_1x+a_2x^2+\dotsb+a_nx^n
\end{dmath*},
\begin{dmath*}
```

Equation Example C. This is from (Knuth, 1986, Chapter 19, p 194).

$$x \equiv x; \tag{1}$$

$$\text{if} \quad x \equiv y \quad \text{then} \quad y \equiv x; \tag{2}$$

$$\text{if} \quad x \equiv y \quad \text{and} \quad y \equiv z \quad \text{then} \quad x \equiv z. \tag{3}$$

Equation Example D. Aligned equations done with the LaTeX `array` environment in order to get the equation number centered:

$$\begin{array}{rcl} P(x) & = & a_0 + a_1x + a_2x^2 + \cdots + a_nx^n, \\ P(-x) & = & a_0 - a_1x + a_2x^2 - \cdots + (-1)^n a_nx^n. \end{array} \tag{30}$$

Notice the too-large spacing around the equal signs and the too-small interline spacing. These can be corrected, but only by rather cumbersome extra work.

```
 P(-x)=a_0-a_1x+a_2x^2-\dotsb+(-1)^na_nx^n
\end{dmath*}.
\end{dgroup}
```

Perhaps some of the lines are numbered and others are not (*The TEXbook* Chapter 19 p 192):

$$\begin{aligned} (x+y)(x-y) &= x^2 - xy + yx - y^2 \\ &= x^2 - y^2; \end{aligned} \tag{31}$$

$$(x+y)^2 = x^2 + 2xy + y^2. \tag{32}$$

```
$$\eqalignno{(x+y)(x-y)&=x^2-xy+yx-y^2\cr
    &=x^2-y^2;&(4)\cr
  (x+y)^2&=x^2+2xy+y^2.&(5)\cr}$$
```

In LaTeX this would typically be accomplished in with the `eqnarray` environment and `\nonumber`. With the `amsmath` package it would be done with a combination of `split` and `align`:

```
\begin{align}
\begin{split}
  (x+y)(x-y)&=x^2-xy+yx-y^2\\
    &=x^2-y^2;
\end{split}
\\% align break
  (x+y)^2&=x^2+2xy+y^2.
\end{align}
```

With the `breqn` package the logical distinction between a long, split equation and equations that are entirely separate is clearer. Among other things, this makes it possible for documentclasses to specify that the space between distinct equations should be larger than the space between the lines of a single multiline equation, and/or should stretch more if necessary to fill up a page.

```
\begin{dgroup}
\begin{dmath}
(x+y)(x-y) =x^2-xy+yx-y^2
     =x^2-y^2
\end{dmath};
\begin{dmath}
  (x+y)^2 =x^2+2xy+y^2
\end{dmath}.
\end{dgroup}
```

From *The TEXbook* Chapter 19 p193. You can also insert a line of text between two equations, without losing the alignment. For example, consider the two displays

$$x = y + z$$

and

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

which Knuth demonstrates as

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

The `amsmath` package provides an `\intertext` command for this situation:

```
\begin{align*}
  x&=y+z\\
\intertext{and}
  x^2&=y^2+z^2.
\end{align*}
```

The `dmath` environment reimplements 'intertext' as an environment. Among other things this means that the text can contain a bit of verbatim, should that ever be necessary.

```
\begin{dgroup}
\begin{dmath*}
  x =y+z
\end{dmath*}
\begin{intertext}
and
\end{intertext}
\begin{dmath*}
  x^2 =y^2+z^2
\end{dmath*}.
\end{dgroup}
```

Examples from (Fomin, Gelfand, and Postnikov, 1997). Note: the authors had added space by hand before all the end-of-display punctuation to get the amount of space they desired. With `breqn` conventions this is handled automatically and the amount of space is easy to change.

$$\partial_w(fg) = w(f)\partial_w g + \sum(\lambda_i - \lambda_j)\partial_{wt_{ij}} g \ ,$$

```
\begin{dmath*}
\partial_w(fg)=w(f)\partial_w g
  +\sum(\lambda_i-\lambda_j)\partial_{wt_{ij}}g
\end{dmath*},
```

$$\partial_w(fg) = w(f)\partial_w g + \sum(\lambda_i - \lambda_j)\partial_{wt_{ij}} g \ , \quad (2.5)$$

```
\begin{dmath}
\partial_w(fg)=w(f)\partial_w g
  +\sum(\lambda_i-\lambda_j)\partial_{wt_{ij}}g
\end{dmath},
```

$$\partial_i\partial_j = \partial_j\partial_i \quad \text{for } |i-j|>1 \ ,$$
$$\partial_i\partial_{i+1}\partial_i = \partial_{i+1}\partial_i\partial_{i+1} \ ,$$
$$\partial_i^2 = 0 \ .$$

```
\begin{dgroup*}[aligned={F}]
\begin{dmath*}
\partial_i \partial_j = \partial_j \partial_i
  \condition[]{for $\abs{i-j}>1$}
\end{dmath*},
\begin{dmath*}
\partial_i\partial_{i+1}\partial_i
  =\partial_{i+1}\partial_i\partial_{i+1}
\end{dmath*},
\begin{dmath*}
\partial_i^2 = 0
\end{dmath*}.
\end{dgroup*}
```

$$\partial_i\partial_j = \partial_j\partial_i \quad \text{for } |i-j|>1 \ , \quad (2.3)$$
$$\partial_i\partial_{i+1}\partial_i = \partial_{i+1}\partial_i\partial_{i+1} \ , \quad (2.4)$$
$$\partial_i^2 = 0 \ . \quad (2.5)$$

```
\begin{dgroup*}
\begin{dmath}
\partial_i \partial_j = \partial_j \partial_i
  \condition[]{for $\abs{i-j}>1$}
\end{dmath},
\begin{dmath}
\partial_i\partial_{i+1}\partial_i
  =\partial_{i+1}\partial_i\partial_{i+1}
\end{dmath},
\begin{dmath}
\partial_i^2 = 0
\end{dmath}.
\end{dgroup*}
```

Examples from (Fomin, Gelfand, and Postnikov, 1997) (continued).

$$\begin{aligned} \partial_i\partial_j &= \partial_j\partial_i \quad \text{for } |i-j|>1\ , \\ \partial_i\partial_{i+1}\partial_i &= \partial_{i+1}\partial_i\partial_{i+1}\ , \\ \partial_i^2 &= 0\ . \end{aligned} \tag{2.3}$$

```
\begin{dgroup}
\begin{dmath*}
\partial_i \partial_j = \partial_j \partial_i
  \condition[]{for $\abs{i-j}>1$}
\end{dmath*},
\begin{dmath*}
\partial_i\partial_{i+1}\partial_i
  =\partial_{i+1}\partial_i\partial_{i+1}
\end{dmath*},
\begin{dmath*}
\partial_i^2 = 0
\end{dmath*}.
\end{dgroup}
```

$$\begin{aligned} e_j^k(e_{i+1}^k - e_{i+1}^{k+1}) &= -e_i^k(e_{j+1}^{k+1} - e_{j+1}^k) \\ &\cdots \\ &= \sum_{l\geq 1}(e_{i+1-l}^{k+1}e_{j+l}^k - e_{j+1}^{k+1}e_{i+1-l}^k)\ . \end{aligned} \quad \square$$

```
\begin{dmath*}[qed={T}]
e_j^k(e_{i+1}^k-e_{i+1}^{k+1})
  = -e_i^k(e_{j+1}^{k+1}-e_{j+1}^k)
  ...
  = \sum_{l\geq 1}
    (e_{i+1-l}^{k+1} e_{j+l}^k
     - e_{j+1}^{k+1} e_{i+1-l}^k)
\end{dmath*}.
```

$$\psi : f \mapsto F\ , \quad f = F(\mathcal{X})(1)\ . \tag{5.4}$$

```
\begin{dgroup}[inline={T}]
\begin{dmath*}
\psi : f\mapsto F
\end{dmath*},
\begin{dmath*}
f=F(\mathcal{X})(1)
\end{dmath*}.
\end{dgroup}
```

The reduced minimal Gröbner basis for I_3^q consists of

$$H_1^3 = x_1 + x_2 + x_3\ ,$$
$$H_2^2 = x_1^2 + x_1x_2 + x_2^2 - q_1 - q_2\ ,$$

and

$$H_3^1 = x_1^3 - 2x_1q_1 - x_2q_1\ .$$

```
\begin{dgroup*}
\begin{dmath*}
H_1^3 = x_1+x_2+x_3
\end{dmath*},
\begin{dmath*}
H_2^2 = x_1^2+x_1 x_2+x_2^2-q_1-q_2
\end{dmath*},
\begin{intertext}
and
\end{intertext}
\begin{dmath*}
H_3^1 = x_1^3  - 2 x_1 q_1 - x_2 q_1
\end{dmath*}.
\end{dgroup*}
```

Additional features under consideration

- With the package option `refnumbers`, all equations remain unnumbered except those that are actually referred to (this idea coming from Enrico Bertolazzi's `easyeqn` package).
- In an equation group, a vertical bracket reaching from the equation number indicates the lines to which the number applies.
- More explicit support for the Russian typesetting conventions for equation breaks, as described by Grinchuk (Grinchuk, 1996). The `flexisym` provides a natural entry point.

Concluding remarks

I think it may be clear from the above discussion that the subject of typesetting equations is far too broad and complex to be covered in adequate detail in the space available here. I have perforce limited myself to sketching out the goals of the `breqn` package, the LaTeX2e user syntax, and some of the more significant constraints on the problems it attempts to solve. Further details are available in the documentation that accompanies the package (which, unlike this article, will continue to evolve as details change, as they surely will given that the package is barely into alpha stage as I write).

References

Borde, Arvind. *Mathematical TEX by Example*. Academic Press, New York, 1995.

Downes, Michael J. *AmS-LaTeX Version 1.2 User's Guide*. American Mathematical Society, Providence, R.I., 1995.

Duff, M. J., R. Minasian, and E. Witten. "Evidence for heterotic/heterotic duality". *xxx.lanl.gov e-Print archive* (hep-th/9601036), 1996.

Fomin, Sergey, S. Gelfand, and A. Postnikov. "Quantum Schubert polynomials". *J. Amer. Math. Soc.* **10**(3), 565–596, 1997.

Grinchuk, Mikhail Ivanovich. "TEX and Russian traditions of typesetting". *TUGboat* **17**, 385–388, 1996.

Knuth, Donald E. *The TEXbook*. Addison Wesley, Reading, Mass., 1986.

Kopka, Helmut and P. W. Daly. *A guide to LaTeX 2ε*. Addison-Wesley, Wokingham, England, 1995.

Spontaneous symmetry breaking is a spontaneous process of symmetry breaking, by which a physical system in a symmetric state ends up in an asymmetric state. In particular, it can describe systems where the equations of motion or the Lagrangian obey symmetries, but the lowest-energy vacuum solutions do not exhibit that same symmetry. When the system goes to one of those vacuum solutions, the symmetry is broken for perturbations around that vacuum even though the entire Lagrangian retains that symmetry. TeX starts squeezing the first equation to fit within the width constraints. For the second equation though, since it fits it will keep the width of the box constant. Oops! the two equations do not align anymore, but have small differences until about 185pt when the first equation is stretched to what TeX thinks is a good equation display. Next we view the logical issues in the breqn algorithm by forcing a break at a 160pt width and typesetting the following boxes to have some visual clues: It is obvious that the original good position