CHAPTER 9: MULTILINE MATH DISPLAYS

NOAH C. RUDERMAN

ABSTRACT. We will review

- Gathering formulas
- Splitting long formulas
- General rules
- Aligned columns
- Aligned subsidary math environments
- Adjusted columns
- Commutative diagrams

Multiline math formulas are displayed in *columns*. The columns are either *adjusted*, that is, centered, or set flush left or right, or *aligned*, that is, an alignment point is designated for each column and for each line. A *subsidary math environment* can only be used inside another math environment. An example would be the `cases` environment.

The `gather` environment groups multiple one-line formulas, each centered on separate lines.

$$x_1x_2 + x_2^2x_2^2 + x_3$$

(second equation)
$$x_1x_3 + x_1^2x_3^2 + x_2$$

(1)
$$x_1x_2x_3$$

You must separate lines with `\\` but `\\` should not be typed on the last line. Also, each line is numbered unless it has a `\tag` or `\notag` on the line before the separator `\\`. Also, the `gather*` environment is the same as `gather` except that the equations are not numbered, although you may still tag them with the `\tag` command.

The `multline` environment splits very long formulas into segments on different lines. The first line is set flush left, the last line is set flush right, and lines inbetween are centered.

(2)
$$\begin{aligned}(x_1x_2x_3x_4x_5x_6)^2 \\ + (y_1y_2y_3y_4y_5 + y_1y_3y_4y_5y_6 + y_1y_2y_4y_5y_6 + y_1y_2y_3y_5y_6)^2 \\ + (u_1u_2u_3u_4 + u_1u_3u_4u_5)^2\end{aligned}$$

Lines are separated with \\ and \\ should not be typed on the last line. The formula is numbered as a whole unless the `\tag` or `\notag` command is used. No blank lines allowed within the environment.

Note: `{}+` was not typed on the beginning of each line because the `multline` environment knows that a long formula is being broken and typesets + as a binary operation.

The `multline*` environment does not label the formula unless we use the command `\tag`.

Lastly, we can make any line of the `multline` environment flush left or flush right with the `\shoveleft` or `\shoveright` command, respectively. The argument of these commands is the entire line being shifted.

Separate parts of a formula such as each line of the `multline` environment are called subformulas. Also the different segments of the `align` environment are subformulas.

In general, in a line of an aligned formula, the first part is everything between the beginning of the line and the first & symbol. There can be a number of parts delimited by two consecutive & symbols (&&). Finally, the past part is from the last & symbol to the end of the line or the line separator \\. These parts are also called *subformulas*.

$$\begin{array}{lllll} x & +y & +z & +a & =b \\ c & +d & +e & +f & =g \end{array}$$

Subformula rules

- Each subformula must be a formula that LaTeX can type independently
- If a subformula starts with a binary operation, include `{}` so that it is recognized as a binary operation
- Do the same as the above if a formula if a subformula ends with a binary operation

Three general rules for typesetting math: Break a long formula before a binary relation or binary operation. If you break a formula with a binary operation, add `{}` so that LaTeX recognizes this as a binary operation. If you break a formula within a bracket, indent the formula to the right of the opening bracket.

I can change the numbering of equations in some clever ways

$$x_1x_2 + x_1^2x_2^2 + x_3, \tag{3}$$

$$x_1x_3 + x_1^2x_3^2, \tag{3a}$$

$$x_1x_2x_3; \tag{3b}$$

Another method for this includes including the `gather` environment in the `subequations` environment.

$$x_1x_2 + x_1^2x_2^2 + x_3, \tag{4a}$$

$$x_1x_3 + x_1^2x_3^2, \tag{4b}$$

$$x_1x_2x_3; \tag{4c}$$

The number of aligned columns in the `align` environment is restricted only by the width of the page. So here's how the &'s work for an aligned column: For single &'s, they mark alignment points and column separators. So the first & says that what's on the left and right sides of it (before the next &) will be a single column, and multiple rows will be aligned at that point. The second & states that two columns will be separated (what is on the left is one column, what is on the right (before the next column separator) is another column).

$$\begin{aligned} x + y + z \qquad & a + b + \int \sin x\,dx \qquad & c + d - \sum x^2 \\ e + f \qquad & g + h \qquad & i + j \end{aligned}$$

If we choose to use && in the `align` environment, then whatever is sandwiched between two adjacent &&'s or on the edges consist of their own rows

$$\begin{aligned} x + y + z \qquad & a + b + \int \sin x\,dx \qquad & c + d - \sum x^2 \\ e + f \qquad & g + h \qquad & i + j \end{aligned}$$

Using the && separator seems to make each column flush right while the & separator designates exactly where the columns will be aligned.

We can create gaps if we like

$$\begin{aligned} a_1 \qquad & \qquad & c_1 \\ & \qquad & c_2 \\ a_3 \qquad & b_3 \qquad & \end{aligned}$$

The `falign` environment is similar to the `align` environment except that the `falign` environment moves the leftmost column as far left and the rightmost column as far right as space allows, making more room for the formula.

In some instances you may want to align a formula at a position where the separate subformulas are not valid formulas independent of

each other. In this case you will have to use other tools, such as the `\phantom` command, to align formulas.

$$x_1 + y_1 + \left(\sum_{i<5} \binom{5}{i} + a^2\right)^2 \left(\sum_{i<5} \binom{5}{i} + \alpha^2\right)^2$$

Apparently the `alignat` environment is extremely important. The `alignat` environment leaves the intercolumn spacing to the user. There is a required argument for this environment and it is the number of columns.

$$\begin{aligned} x &= x \wedge (y \vee z) && \text{(by distributivity)} \\ &= (x \wedge y) \vee (x \wedge z) && \text{(by condition (M))} \\ &= y \vee z \end{aligned} \tag{5}$$

In the above the `\quad` command is necessary because that is what designates the space between the columns. Remember that displayed math environments ignore spaces but will respect `\quad` commands and the like. So now I see how the user controls the spacing. Also, the argument for the `alignat` environment only needs to meet the minimum number of columns, as in the program will run the same if we specify more columns than we actually use.

We can use the `\intertext` command to insert one or more lines of text in the middle of an aligned environment.

$$dr = \sqrt[2]{dx^2 + dy^2}$$

We can write this in a different way

$$= dx \sqrt[2]{1 + \left(\frac{dy}{dx}\right)^2}$$

Notice how the equal signs are aligned. The `\intertext` command must follow a line separator command `\\`.

A *subsidary math environment* is a math environment that can only be used inside another math environment. This is similar to creating a large math symbol. The `align`, `alignat`, and `gather` environments have subsidary versions which are called `aligned`, `alignedat`,

and `gathered`.

$$\begin{aligned} x &= 3 + \mathbf{p} + \alpha \\ y &= 4 + \mathbf{q} \\ z &= 4 + \mathbf{r} \\ u &= 5 + \mathbf{s} \end{aligned} \quad \text{using} \quad \begin{aligned} \mathbf{p} &= 5 + a + \alpha \\ \mathbf{q} &= 12 \\ \mathbf{r} &= 13 \\ \mathbf{s} &= 11 + d \end{aligned}$$

We can also used the `aligned` subsidary math environment so that a formula number is centered between two lines

$$\begin{aligned} r &= \sqrt[2]{dx^2 + dy^2} \\ &= dx \sqrt[2]{1 + \left(\frac{dy}{dx}\right)^2} \end{aligned} \tag{6}$$

The numbering is centered because LaTeX treats this all as a large math symbol. Symbols, as a rule, are vetrically centrally aligned. The subsidary math environments take `c`, `t`, or `b` as optional arguments to force vertically centered, top, or bottom alignment, respectively.

$$\begin{aligned} x &= 3 + \mathbf{p} + \alpha \\ y &= 4 + \mathbf{q} \\ z &= 4 + \mathbf{r} \\ u &= 5 + \mathbf{s} \end{aligned} \quad \text{using} \quad \begin{aligned} \mathbf{p} &= 5 + a + \alpha \\ \mathbf{q} &= 12 \\ \mathbf{r} &= 13 \\ \mathbf{s} &= 11 + d \end{aligned}$$

So there's a subsidary math environment called `split`. It works like the `multline` environment but it allows the contents to be aligned with an aligned environment that it is in. I see little use for this so I'm not writing it up but know it exists.

In an *adjusted* multiline math environment, the columns are adjusted so that they are displayed centered, flush left, or flush right, instead of aligned. Since you have no control line by line over the alignment of the columns, & has only one role to play, it is the column separator.

We use the `matrix` subsidary math environment to typeset matrices.

$$\begin{pmatrix} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 1340 \end{pmatrix} = \begin{pmatrix} 1 & 100 & 115 & 27 \\ 201 & 0 & 1 & 1340 \end{pmatrix}$$

You are not allowed to use `matrix` on its own outside a math environment. The `matrix` subsidary environment allows you 10 columns to work with. If you need more, you need to use chapter 15 stuff. Use the `\hdotsfor` command to fill columns with dots. The required argument is the number of rows that there will be dots for. There is also

an optional argument that multiplies the spacing between the dots.

$$\begin{pmatrix} a+b+c & \ldots\ldots\ldots\ldots \\ a+b & u+v \quad z \quad 1340 \end{pmatrix}$$

The `\hdotsfor` command must be used directly after a & or at the beginning of a line. For a single element there are also the `\dots`, `\vdots`, and `\ddots` commands which give us horizontal, vertical, and diagonal dots, respectively.

$$\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix}$$

There are delimited variants of the `matrix` subsidary math environment. They are `pmatrix`, `bmatrix`, `vmatrix`, `Vmatrix`, and `Bmatrix`.

When inserting a matrix in an inline math formula, it may be too large. Use the `smallmatrix` subsidary environment for this.
Compare $\begin{pmatrix} a+b+c & uv \\ a+b & c+d \end{pmatrix}$ with $\left(\begin{smallmatrix} a+b+c & uv \\ a+b & c+d \end{smallmatrix}\right)$.
There are no delimited variants of the `smallmatrix` environment. The `\hdotsfor` command does not work in a small matrix.

The `array` subsidary math environment is very similar to the `matrix` subsidary math environment except that the `array` environment is more versatile. For example, columns can be adjusted in the `array` environment. There is a required argument, which is the alignment of each column. The options are `c`,`l`, or `r` for centered, flush left, or flush right, respectively.

$$\left(\begin{array}{cccc} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 134 \end{array}\right)$$

An `array` subsidary math environment can take any argument that a `tabular` environment can take.

	a	b	c
1	1	1	1
2	1	-1	-1
2	2	1	0

The `cases` environment is also a subsidary math environment.

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases} \tag{7}$$

We can also write this with the `array` environment

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases} \tag{8}$$

or with the `alignedat` subsidary math environment

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases} \tag{9}$$

Include the `amscd` package to use the `CD` subsidary math environment for typesetting commutative diagrams. A cummutative diagram is a matrix made of of two kinds of rows: rows with horizontal arrows and rows with vertical arrows.

$$\begin{array}{ccccccccc} A & \xrightarrow{\log} & B & \xrightarrow[\text{bottom}]{} & C & = & D & \longleftarrow & E & \longleftarrow & F \\ \text{one-one}\downarrow & & & & \uparrow\text{onto} & & \| & & & & \\ X & = & Y & \longrightarrow & Z & \longrightarrow & U & & & & \\ \beta\uparrow & & \gamma\uparrow & & \downarrow & & \downarrow & & & & \\ D & \xrightarrow{\alpha} & E & \longrightarrow & H & & I & & & & \end{array}$$

Note: I am skipping the rest of this section because I don't care about commutative diagrams