AUlogo3-crop.pdf

Introduction to AMSMATH Equation

Guangjie Huang

Electrical and Computer Engineering
Auburn University

September 3, 2018

Contents

Contents

1. **Introduction to AMS equations**
   - **Basic introduction**
   - **New features**
   - **Options**
2. Displayed equations
   - Single Equation
   - Equation group
   - Other issues about equations
3. Miscellaneous mathematical features
4. Further study
5. Resource

Introduction

The *amsmath* package is a LaTeX package that provides miscellaneous enhancements for improving the information structure and printed output of documents that contain mathematical formulas.

New features of amsmath equation

- A convenient way to define new 'operator name' commands analogous to **sin** and **lim**, including proper side spacing and automatic selection of the correct font style and size (even when used in sub- or superscripts).
- Multiple substitutes for the **eqnarray** environment to make various kinds of equation arrangements easier to write.
- Equation numbers automatically adjust up or down to aviod overprinting on the equation contents (unlike **eqnarray**).
- Spacing around equals signs matches the normal spacing in the **equation** environment (unlike **eqnarray**).

New features of amsmath equation II

- A way to produce multiline subscripts as are often used with summation or product symbols.
- An easy way to substitute a variant equation number for a given equation instead of the automatically supplied number.
- An easy way to produce subordinate equation numbers of the form $(1.3a)(1.3b)(1.3c)$ for selected groups for equations.

default Options for the *amsmath* package

centertags For a split equation, place equation numbers vertically centered on the total height of the equation.

tbtags 'Top-or-bottom tags': For a split equation, place equation numbers level with the last (resp. first) line, if numbers are on the right.

sumlimits Place the subscripts and super scripts of summation symbols above and below, in displayed equations. This option also affects other symbols of the same type – $\prod \coprod \otimes \oplus$, and so forth – but excluding integrals (see below).

nosumlimits Always place the subscripts nd superscripts of summation-type symbols to the side, even in displayed equations.

intlimits Like *sumlimits*, but for integral symbols

nointlimits (default) Opposite of *intlimits*

namelimits (default) Like *sumlimits*, but for certain 'operator names' such as det, inf, lim, max, min, that traditionally have subscripts placed underneath when they occur in a displayed equation.

nonamelimits Opposite of *namelimits*

To use one of these package options, put the option name in the optional argument of the **usepackage** command — e.g., $\backslash usepackage[intlimits]\{amsmath\}$.

Other available options for *amsmath*

The *amsmath* package also recognize the following options which are normally selected (implicitly or explicitly) through **documentclass** command.

leqno Place equation numbers on the left

reqno Place equation numbers on the right

fleqn Position equations at a fixed indent from the left margin rather than centered in the text column.

Contents

AUlogo3-crop.pdf

equation	equation*	align	align*
gather	gather*	flalign	flalign*
multitaper	multitaper*	alignat	alignat*
split			

Single equations

```
\begin{equation}
    a = b + c
\end{equation}
```

$$a = b + c \tag{1}$$

```
\begin{equation*}
    a = b + c
\end{equation*}
```

$$a = b + c$$

Split equations without alignment

The *multiline* environment is a variant of the *equation* environment used for equations that don't fit on a single line, which has only a single equation number. The first line is at the left margin and the last line at the right margin. It is possible to force one of the middle lines to the left or right with commands **shoveleft**, **shoveright**. These commands take the entire line as an argument, up to but not including the final \\.

Split equations without alignment

```
\begin{multline}
    a=b+c\\
    =d+e\\
    =m+n
\end{multline}
```

$$
\begin{aligned}
a = b + c \\
= d + e \\
= m + n
\end{aligned}
\qquad (2)
$$

```
\begin{multline}
a=b+c\\
\shoveright{=d+e}\\
=m+n
\end{multline}
```

$$
\begin{aligned}
a = b + c \\
= d + e \\
= m + n
\end{aligned}
\qquad (3)
$$

Split equations with alignment

Unlike *multiline*, the *split* environment is for *single* equations that don't fit on one line. The *split* environment provides no numbering, because it is intended to be used *only inside some other displayed equation structure*, usually an *equation, align, or gather* environment, which provides the numbering. For example:

Split equations with alignment example

```
\begin{equation} \label{abc}
    \begin{split}
    a& =b+c-d\\
    & \quad +e-f\\
    & =g+h\\
    & =i
    \end{split}
\end{equation}
```

$$
\begin{aligned}
a &= b + c - d \\
&\quad + e - f \\
&= g + h \\
&= i
\end{aligned}
\tag{4}
$$

AUlogo3-crop.pdf

Equations groups without alignment: *gather*

The *gather* environment is used for a group of consecutive equations when there is no alignment desired among them; each one is centered separately within the text width. Equations inside *gather* are separated by a **bslash** command. Any equation in a gather may consist of a *begin*$\{split\}$ ... *end*$\{split\}$
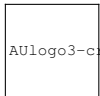
Split equations with alignment example

```
\begin{gather}
a_1=b_1+c_1\\
\begin{split}
  a& =b\\
  & =c\\
  & =d
\end{split}\\
a_2=b_2+c_2-d_2+e_2
\end{gather}
```

$$a_1 = b_1 + c_1 \qquad (5)$$
$$a = b$$
$$= c \qquad (6)$$
$$= d$$
$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (7)$$

Equations groups with mutual alignment

The *align* environment is used for two or more equations when vertical alignment is desired; usually binary relations such as equal signs are aligned.

$$x = y \qquad X = Y \qquad a = b + c \qquad (8)$$
$$x = y \qquad X = Y \qquad a = b \qquad (9)$$
$$x + x = y + y \qquad X + X = Y + Y \qquad ab = cb \qquad (10)$$

```
\begin{align}
x&=y & X&=Y & a&=b+c\\
x&=y & X&=Y & a&=b\\
x+x&=y+y & X+X&=Y+Y & ab&=cb
\end{align}
```

align with line-by-line annotations

Line-by-line annotations on an equation can be done by judicious application of \text inside an *align* environment.

$$
\begin{aligned}
x &= y_1 - y_2 + y_3 - y_5 + y_8 - \dots && \text{by (3.21)} \quad (11)\\
&= y \circ y^* && \text{by Theorem 1.} \quad (12)\\
&= y(0)y && \text{by Axiom 1.} \quad (13)
\end{aligned}
$$

```
\begin{align}
x& = y_1-y_2+y_3-y_5+y_8-\dots
&& \text{by (3.21)}\\
& = y\circ y^* && \text{by Theorem 1}\\
& = y(0) y && \text {by Axiom 1.}
```

alignat

A variant environment *alignat* allows the horizontal space between equations to be explicitly specified. This environment takes one argument, the number of "equation columns": count the maximum number of $\&s$ in any row, add 1 and divide by 2.

$$
\begin{alignedat}{2}
x &= y_1 - y_2 + y_3 - y_5 + y_8 - \dots & \quad & \text{by (3.21)} \quad (14)\\
&= y \circ y^* & & \text{by Theorem 1} \quad (15)\\
&= y(0)y & & \text{by Axiom 1.} \quad (16)
\end{alignedat}
$$

```
\begin{alignat}{2}
x& = y_1-y_2+y_3-y_5+y_8-\dots
&\quad& \text{by (3.21)}\\
& = y\circ y^* && \text{by Theorem 1}\\
& = y(0) y && \text {by Axiom 1.}
\end{alignat}
```

Alignment building blocks

Like *equation*, the multi-equation environments *gather, align*, and *alignat* are designed to produce a structure whose width is the full line width. This means, for example, that one cannot readily add parentheses around the entire structure. But variants *gathered, aligned*, and *alignedat* are provided whose total width is the actual width of the contents; thus they can be used as a component in a containing expression. Like the *array* environment, these *-ed* variants also take an optional *[t]* or *[b]* argument to specify vertical positioning.

Alignment building blocks example

$$\left. \begin{aligned} B &= -\partial \times E, \\ E &= \partial \times B - 4\pi j, \end{aligned} \right\} \qquad \text{Maxwells equations}$$

```
\begin{equation*}
\left.\begin{aligned}
B&=-\partial\times E,\\
E&=\partial\times B - 4\pi j,
\end{aligned}
\right\}
\qquad \text{Maxwells equations}
\end{equation*}
```

"Case" constructions

"Case" constructions like the following are common in mathematics:

$$P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r!\,(-1)^{(r-j)/2} & \text{if } r-j \text{ is even.} \end{cases}$$

```
\begin{equation}
P_{r-j}=\begin{cases}
0& \text{if $r-j$ is odd},\\
r!\,(-1)^{(r-j)/2}& \text{if $r-j$ is eve
\end{cases}
\end{equation}
```

Adjusting tag placement

There is a $\backslash$*raisetag* command provided to adjust the vertical position of the current equation number, if it has been shifted away from its normal position. For example, $\backslash$*raisetag* $\{6pt\}$ is to move a particular number up by six points. It is best to use it when the document is nearly finalized.

Vertical spacing and page breaks

$\backslash\backslash[< dimension >]$ command is used to get extra vertical space between lines in all the *amsmath* displayed equation environments.

To get an individual page break inside a particular displayed equation, $\backslash displaybreak$ command is provided. It is is best placed immediately before the $\backslash\backslash$ where it is to take effect. It takes an optional argument between 0 and 4 denoting the desirability of the page break. 0 means "it is permissible to break here" without encouraging a break. with no optional argument is the same as 4, which forces a break. This command does not work in *split, aligned, gathered, and alignedat*.

Interruptinig a display

The command $\backslash$*intertext* is used for a short interjection of one or two lines of text in the middle of a multiple-line display structure. It may only appear right after a $\backslash\backslash$ or $\backslash\backslash*$ command.

Interruptinig a display example

$$A_1 = N_0(\lambda; \Omega) - \phi(\lambda; \Omega), \tag{17}$$

$$A_2 = \phi(\lambda; \Omega) - \phi(\lambda; \Omega), \tag{18}$$

and

$$A_3 = \mathcal{N}(\lambda; \omega). \tag{19}$$

```
\small{
\begin{align}
A_1&=N_0(\lambda;\Omega)-\phi(\lambda;\Omega)
A_2&=\phi(\lambda;\Omega)-\phi(\lambda;\Omega
\intertext{and}
A_3&=\mathcal{N}(\lambda;\omega).
\end{align}
}
```

Numbering hierarchy

Redefine the command $\backslash$*theequation* can number equations within sections. $\backslash$*setcounter* can reset the equation counter to be zero at the beginning of a new section or chapter. The redefine command is:

```
\renewcommand{\theequation}{\thesection.\arabic{equation}}
```

A more convenient way is command $\backslash numberwithiin$, which tie the equation number to section numbering, with automatic reset of the equation counter.

```
\numberwithin{equation}{section}
```

\eqref

If $\backslash ref\{abc\}$ produces 4, then $\backslash eqref$ produces (4).

Subordinate numbering sequences

The *subequations* environment can number equations in a particular group with a subordinate numbering.

```
\begin{subequations}
...
\end{subequations}
```

causes all numbered equations within that part of the document to be numbered (4.9a) (4.9b) (4.9c) . . . , if the preceding numbered equation was (4.8).

Contents

Matrices

The matrics environment in *amsmath* is more convenient than *array*. These commands include **pmatrix, bmatrix, Bmatrix, vmatrix** that have (respectively) (), [], $\{\}$,$||$, and $||||$ delimiters built in.

matrices example

```
\bigl( \begin{smallmatrix}
a&b\\ c&d
\end{smallmatrix} \bigr)
```

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix} \right)$$

matrices example \hdotsfor

The command \hdotsfor[multiplier]{column number} produces a row of dots in a matrix spanning the given number of columns. The number in square brackets will be used as a multiplier.

```
\begin{pmatrix} D_1t&-a_{12}t_2&\dots&-a_{1n}t_n\\
-a_{21}t_1&D_2t&\dots&-a_{2n}t_n\\
\hdotsfor[2]{4}\\
-a_{n1}t_1&-a_{n2}t_2&\dots&D_nt\end{pmatrix}
```

$$\begin{pmatrix} D_1t & -a_{12}t_2 & \ldots & -a_{1n}t_n\\ -a_{21}t_1 & D_2t & \ldots & -a_{2n}t_n\\ \hdotsfor[2]{4}\\ -a_{n1}t_1 & -a_{n2}t_2 & \ldots & D_nt\end{pmatrix}$$

Dots

- **dotsc** for "dots with commas"
- **dotsb** for "dots with binary operators \ relations"
- **dotsm** for "multiplication dots"
- **dotsi** for "dots with integrals"
- **dotso** for "other dots"

Use the above commands instead of **ldots** and **cdots**, the document can adapt to different conventions on the fly.

Accents in math

amsmath improves the accents in math. $\hat{\hat{A}}$ (\hat{\hat{A}}). **dddot** and **ddddot** are available to produce triple and quadruple dot accents in addition to the **dot** and **ddot** accents.

The package *amsxtra* provides commands **sphat** or **sphat** to get a superscripted hat or tilde character.

Roots

```
$\sqrt[\beta]{k}$
```

$$\sqrt[\beta]{k}$$

In the **amsmath** package **leftroot** and **uproot** can adjust the position of the root.

```
$\sqrt[\leftroot{-2}
\uproot{2}\beta]{k}$
```

$$\sqrt[\beta]{k}$$

The root is moved to the right

Boxed formulas

The command **boxed** puts a box around its argument.

```
\boxed{\eta \leq C(\delta(\eta)
+\Lambda_M(0,\delta))}
```

$$\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0,\delta))}$$

Over, under and extensile arrows

amsmath provides additional over, under and extensile arrow commands.

```
\overleftarrow \underleftarrow
\overrightarrow \underrightarrow
\overleftrightarrow \underleftrightarrow
```

xleftarrowand **xleftarrow****xrightarrow** produce arrows that extend automatically to accommodate unusually wide subscripts or superscripts.

```
$A \xleftarrow{n+\mu-1}\quad B \xrightarrow[T]{n\pm i-1}C$
```

$$A \xleftarrow{n+\mu-1} \quad B \xrightarrow[T]{n\pm i-1} C$$

Affixing symbols to other symbols

LATEX provides **stackrel**, *amsmath* provides more convenient commands **overset**, **underset**, and **sideset**

```
$\overset{*}{X}$
$\underset{*}{X}$
$\sideset{_*^*}{_*^*}\prod$
```

$\overset{*}{X}$ $\underset{*}{X}$ $\sideset{_*^*}{_*^*}\prod$

Multiline subscripts and superscripts

The **substack** command can be used to produce a multiline subscript or superscript.

```
\sum_{\substack{
0\le i\le m\\
0<j<n}}
P(i,j)
```

$$\sum_{\substack{0\le i\le m\\0<j<n}} P(i,j)$$

Multiline subscripts and superscripts II

The **subarray** environment which allows you to specify that each line should be left-aligned instead of centered

```
\sum_{\begin{subarray}{l}
i\in\Lambda\\ 0<j<n
\end{subarray}}
P(i,j)
```

$$\sum_{\substack{i\in\Lambda\\0<j<n}} P(i,j)$$

both subscripts and superscripts

The package **stackrel** allows you to get subscripts and superscripts simultaneously.

```
\begin{equation}
L_{SC}
\stackrel[H_{0}]{H_{
\tau
\end{equation})
```

$$L_{SC} \underset{H_0}{\overset{H_1}{\gtrless}} \tau \qquad (20)$$

The \frac, \dfrac, \tfrac commands

\frac : basic; \dfrac: displayed style; \tfrac: text style

```
\begin{equation}
\frac{1}{k}\log_2 c(f)\;
\tfrac{1}{k}\log_2 c(f)\;
\sqrt{\frac{1}{k}\log_2 c(f)}\;
\sqrt{\dfrac{1}{k}\log_2 c(f)}
\end{equation}
```

$$\frac{1}{k}\log_2 c(f) \; \tfrac{1}{k}\log_2 c(f) \; \sqrt{\frac{1}{k}\log_2 c(f)} \; \sqrt{\dfrac{1}{k}\log_2 c(f)} \tag{21}$$

The \binom, \dbinom,\tbinom commands

\binom : basic; \dbinom: displayed style; \tbinom: text style

```
2^k-\binom{k}{1}2^{k-1}
+\binom{k}{2}2^{k-2}
```

$$2^k - \binom{k}{1}2^{k-1} + \binom{k}{2}2^{k-2}$$

The \genfrac command

The capabilities of \frac, \binom, and their variants are subsumed by a generalized fraction command \genfrac with six arguments.

```
\genfrac{left-delim}{right-delim}
{thickness}{mathstyle}
{numerator}{denominator}
```

Continued fractions

```
\begin{equation*}
\cfrac{1}{\sqrt{2}+
\cfrac{1}{\sqrt{2}+
\cfrac{1}{\sqrt{2}+\dotsb
}}}
\end{equation*}
```

$$\cfrac{1}{\sqrt{2} + \cfrac{1}{\sqrt{2} + \cfrac{1}{\sqrt{2} + \cdots}}}$$

Smash options

The command \smash adjusts the subformulas position with respect to adjacent symbols.

```
$\sqrt{x}
+ \sqrt{\smash[b]
{y}} + \sqrt{z}$

$\sqrt{x}
+ \sqrt{{y}} +
\sqrt{z}$
```

$$\sqrt{x} + \sqrt{y} + \sqrt{z}$$
$$\sqrt{x} + \sqrt{y} + \sqrt{z}$$

Contents

Further study

- Delimiters
- Operator names
- Using math fonts
- Math spacing commands

Contents

Resource

- http://www.ams.org/publications/authors/tex/amslatex
- The Not So Short Introduction to LATEX