Hướng dẫn sử dụng gói `amsmath` (phiên bản 2.0)

Tác giả: Hội Toán học Mỹ (AMS)

13/12/1999 (sửa đổi 25/02/2002)

Biên dịch: Ky Anh ⟨kyanh@o2.pl⟩

Bản dịch mới nhất 15/10/2005

http://VietTUG.org

Mục lục

—1—

Thuật ngữ

Dưới đây là một số thuật ngữ dùng trong tài liệu này.

dimension : độ dài trong LaTeX, ví dụ: `6pt`, `-2pc`, `5mm`,...

font : kiểu chữ

hyphen : tách một chữ (ở cuối dòng) với nhiều ký tự thành các phần nhỏ ngăn cách bởi dấu gạch ngang (dấu `hyphen`). Việc tách này giúp cho một chữ quá dài không tràn ra khỏi dòng.

number : *bla bla bla...*

preamble : phần nằm trước `\begin{document}` của tập tin nguồn LaTeX.

tag : chỉ số phương trình

robust : *bla bla bla...*

text : chuỗi các mã nguồn LaTeX, ví dụ: `"xem Tiên đề~\ref{ax:1}"`

typeset : sắp chữ nhờ LaTeX

wrap : tự động chia một dòng quá dài thành nhiều dòng nhỏ, để chúng bố trí vừa trên một chiều rộng cố định cho trước

canh cột : bố trí các phần của phương trình theo cột (chiều đứng)

dấu ngoặc : (phân cách); thuật ngữ tiếng Anh là `delimiter`; là các dấu (,),[,],{,},|,...

chỉ số phương trình : nhãn dùng để phân biệt các phương trình

phương trình : biểu thức toán học bất kỳ được biểu diễn nhờ LaTeX

v-khoảng cách : khoảng cách theo chiều đứng

—2—

Giới thiệu

Gói `amsmath` dành cho LaTeX cung cấp nhiều tiện ích để `typeset` các tài liệu Toán học phức tạp. Gói này có trong hầu hết các bản phân phối mới của LaTeX. Để lấy các thông tin cập nhật về gói này, bạn xem ở

http://www.ams.org/tex/amsmath.html

ftp://ftp.ams.org/pub/tex/

Tài liệu này mô tả các tính năng và của gói `amsmath` và thảo luận về các hướng sử dụng chúng. Tài liệu cũng đề cập sơ lược về các gói

`amsbsy` `amstext`
`amscd` `amsxtra`
`amsopn`

Các gói này đều liên quan đến việc `typseset` biểu thức toán học. Thông tin về các ký hiệu và `font` mở rộng, xem ở [1] và

http://www.ams.org/tex/amsfonts.html

Tài liệu về gói `amsthm`, các lớp[1] AMS (`amsart`, `amsbook`, etc.) có thể tìm thấy trong [3], [2] và

http://www.ams.org/tex/author-info.html

Nếu bạn đã làm việc lâu dài với LaTeX và phải `typeset` rất nhiều các biểu thức toán học, thì với gói `amsmath`, bạn có thể tìm thấy giải pháp cho những vấn đề hay gặp nhất:

- Dễ dàng định nghĩa toán tử, hàm toán học mới (tương tự như `\sin`, `\cos`); các toán tử mới sẽ tự động canh chỉnh kích thước, kiểu `font` và khoảng cách tương quan với các phần tử khác trong biểu thức.
- Nhiều biến thể của môi trường `eqnarray` để sắp xếp nhiều loại (hệ) phương trình khác nhau.
- Các số chỉ phương trình tự động chuyển dịch lên, xuống để tránh tình trạng tràn trang (khắc phục nhược điểm của `eqnarray`).
- Khoảng cách xung quanh dấu bằng (=) giống hệt khoảng cách bình thường trong môi trường `equation` (không giống như `eqnarray`).
- Có thể tạo chỉ số dưới, chỉ số trên với nhiều dòng (thường gặp khi làm việc với các ký hiệu tổng, tích)
- Dễ dàng tạo các biến thể cho việc đánh số một phương trình cho trước (nếu bạn không thích kiểu đánh số mặc định).
- Dễ dàng đánh số các phương trình con dạng (1.3a) (1.3b) (1.3c) từ một nhóm các phương trình. Việc đánh số này là *tự động*.

Gói `amsmath` được phân phối cùng với một số gói bổ trợ

[1] `documentclass`

`amsmath` Gói chính; cung cấp rất nhiều tiện ích để biểu diễn phương trình và các biểu thức toán học từ đơn giản đến phức tạp.

`amstext` Cung cấp lệnh `\text` để sắp xếp các đoạn văn bên trong biểu thức toán học.

`amsopn` Cung cấp lệnh `\DeclareMathOperator` để định nghĩa các toán tử mới, như `\sin`, `\lim`.

`amsbsy` Gói này vẫn tồn tại để bảo đảm tính tương thích; tuy nhiên, bạn nên dùng gói `bm` để thay thế cho `amsbsy`.

`amscd` Cung cấp môi trường `CD` để biểu diễn các biểu đồ giao hoán đơn giản (với gói này, bạn không thể vẽ các mũi tên chéo).

`amsxtra` Gói bổ sung, nhằm bảo đảm tương thích với tài liệu dùng phiên bản 1.1 của `amsmath`. Cung cấp: `\fracwithdelims`, `\accentedsymbol`,...

Gói `amsmath` đã bao gộp các gói `amstext`, `amsopn`, and `amsbsy`; nghĩa là khi nạp gói `amsmath`, ba gói kia sẽ tự động nạp theo. Còn để dùng các gói `amscd`, `amsxtra`, bạn phải nạp riêng chúng.

—3—

Các tùy chọn của gói `amsmath`

Để dùng tùy chọn của gói, bạn để tên của tùy chọn vào trong phần tham số bổ sung của lệnh nạp gói `\usepakage`. Các tùy chọn cách nhau bằng dấu phẩy. Ví dụ:

```
\usepackage[intlimits]{amsmath}
\usepackage[sumlimits,intlimits]{amsmath}
```

Gói `amsmath` có các tùy chọn sau đây:

`centertags` (mặc định) Đánh số phương trình bằng cách đặt chỉ số canh giữa theo chiều cao của phương trình.

`tbtags` 'Top-or-bottom tags': Đặt chỉ số của phương trình ở phía bên phải, dòng cuối cùng; hoặc ở phía bên trái, dòng đầu tiên.

`sumlimits` (mặc định) Đặt các chỉ số trên và dưới của các ký hiệu tổng ($\sum$) ở trên và ở dưới (trong công thức riêng dòng). Tùy chọn này cũng ảnh hưởng đến các ký hiệu cùng loại—$\prod$, $\coprod$, $\bigotimes$, $\bigoplus$,...—(nhưng ký hiệu tích phân thì không; xem dưới đây)

`nosumlimits` Luôn đặt chỉ số trên và chỉ số dưới của các ký hiệu dạng tổng ($\sum$, $\prod$,...) ở bên cạnh, ngay cả trong công thức riêng dòng. Ví dụ $\sum_0^1$.

`intlimits` Tương tự như `sumlimits`, nhưng cho ký hiệu tích phân.

`nointlimits` (mặc định) Ngược với `intlimits`.

`namelimits` (mặc định) Tương tự `sumlimits`, nhưng cho một số toán tử như det, inf, lim, max, min; các toán tử này theo truyền thống thường có chỉ số đặt bên dưới toán tử (chế độ công thức riêng dòng).

`nonamelimits` Ngược với `namelimits`.

`leqno` Đặt chỉ số phương trình bên trái.

`reqno` Đặt chỉ số phương trình bên phải.

`fleqn` Biểu diễn phương trình với lề trái cố định; theo mặc định, các phương trình được canh giữa (do đó, lề trái của chúng *thay đổi*).

Đối với ba tùy chọn cuối cùng (`leqno`, `reqno`, `fleqn`), bạn có thể để chúng vào phần tham số bổ sung của `\documentclass`; gói `amsmath` nhận biết điều này và do đó bạn không cần lặp lại khi nạp gói bằng `\usepackage{amsmath}`:

```
\documentclass[reqno]{report}
\usepackage{amsmath}% có tác dụng như \usepackage[reqno]{amsmath}
```

—4—

Biểu diễn phương trình

4.1 Giới thiệu

Gói `amsmath` cung cấp thêm các môi trường biểu diễn phương trình sau đây, bên cạnh các môi trường chuẩn của LaTeX:

```
equation    equation*    align       align*
gather      gather*      flalign     flalign*
multline    multline*    alignat     alignat*
split
```

(Mặc dù môi trường chuẩn `eqnarray` vẫn dùng được, nhưng tốt hơn hết nên dùng môi trường `align` hoặc tổ hợp `equation`+`split`.)

Ngoại trừ `split`, mỗi môi trường đều có hai dạng: *có sao (*)* và *không sao*; các môi trường không sao sẽ sử dụng bộ đếm `equation` của LaTeX để đánh số các phương trình một cách tự động (do đó, ta gọi chúng là *môi trường có đánh số*). Bạn có thể bỏ qua việc đánh số cho bất kỳ dòng phương trình nào bằng cách đặt lệnh `\notag` trước khi dùng `\\`; cũng có thể thay đổi kiểu đánh số cho dòng phương trình cụ thể, bằng cách dùng `\tag{`⟨*label*⟩`}`, ở đây ⟨*label*⟩ là `text` bất kỳ, chẳng hạn `$*$` hoặc `ii`. Theo mặc định, ⟨*label*⟩

của \tag sẽ được đặt trong cặp dấu ngoặc đơn, ví dụ (3.32); nếu không muốn điều này xảy ra, bạn có thể dùng \tag*. *Để ý rằng,* \tag và \tag* có thể dùng với mọi môi trường đã liệt kê ở trên, chứ không phải với chỉ các môi trường có đánh số (không sao). Ví dụ về việc dùng \tag có thể tìm thấy trong testmath.tex và subeqn.tex được phân phối cùng với tài liệu này.

Môi trường split là một dạng đặc biệt, chỉ có thể được dùng bên trong các môi trường khác. Tuy nhiên, nó lại không thể dùng bên trong multline.

Với các môi trường có chức năng canh cột (split, align,...), các ký hiệu quan hệ (dấu $=$, $>$, $\leq$, ...) phải được đặt sau dấu &. Đây là điểm khác biệt so với eqnarray.

4.2 Phương trình đơn

Môi trường equation dùng biểu diễn phương trình đơn và tự động đánh số cho phương trình đó. Môi trường equation* có tác dụng tương tự, nhưng không đánh số.[2]

Hai môi trường này chỉ biểu diễn phương trình trên đúng một dòng; bạn không thể dùng \\ bên trong hai môi trường đó. Nếu biểu thức quá dài, sẽ xảy tràn trang. Hãy xem mục tiếp theo.

4.3 Chia nhỏ phương trình nhưng không canh cột

Môi trường multline là biến thể của equation, cho phép biểu diễn các phương trình quá dài, không thể bố trí vừa khít trên một dòng. Trong môi trường này, bạn dùng \\ để tách các dòng. Dòng đầu tiên sẽ canh ở lề trái, dòng cuối cùng được canh ở lề phải; ngay trước dòng đầu tiên và ngay sau dòng cuối cùng là khoảng trắng indent được cung cấp bởi biến độ dài \multlinegap. Các dòng còn lại sẽ được canh giữa trang, trừ khi bạn dùng tùy chọn fleqn.

Giống như equation, môi trường multline chỉ đánh số tất cả các dòng của nó bởi đúng một chỉ số (do đó, bạn không thể dùng \notag cho riêng dòng nào trong môi trường). Chỉ số được đánh sẽ được ở dòng cuối cùng (tùy chọn reqno) hoặc dòng đầu tiên (tùy chọn leqno); kiểu đánh số canh giữ như split không được hỗ trợ.

Có thể làm cho các dòng giữa của môi trường dịch chuyển qua trái hoặc qua phải bằng cách lệnh \shoveleft hay \shoveright. Các lệnh này sẽ

[2] LaTeX chuẩn không có môi trường equation*, mà chỉ có dạng tương đương là displaymath.

Bảng 4.1. So sánh các môi trường biểu diễn phương trình (đường thẳng đứng dùng để chỉ lề trái, phải của tờ giấy tưởng tượng)

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

$$a = b$$

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

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

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

$$\begin{aligned} a &= b + c - d \\ &\quad + e - f \\ &= g + h \\ &= i \end{aligned} \tag{2}$$

```
\begin{multline}
a+b+c+d+e+f\\
+i+j+k+l+m+n
\end{multline}
```

$$\begin{aligned} a + b + c + d + e + f \\ + i + j + k + l + m + n \end{aligned} \tag{3}$$

```
\begin{gather}
a_1=b_1+c_1\\
a_2=b_2+c_2-d_2+e_2
\end{gather}
```

$$a_1 = b_1 + c_1 \tag{4}$$
$$a_2 = b_2 + c_2 - d_2 + e_2 \tag{5}$$

```
\begin{align}
a_1& =b_1+c_1\\
a_2& =b_2+c_2-d_2+e_2
\end{align}
```

$$a_1 = b_1 + c_1 \tag{6}$$
$$a_2 = b_2 + c_2 - d_2 + e_2 \tag{7}$$

```
\begin{align}
a_{11}& =b_{11}&
  a_{12}& =b_{12}\\
a_{21}& =b_{21}&
  a_{22}& =b_{22}+c_{22}
\end{align}
```

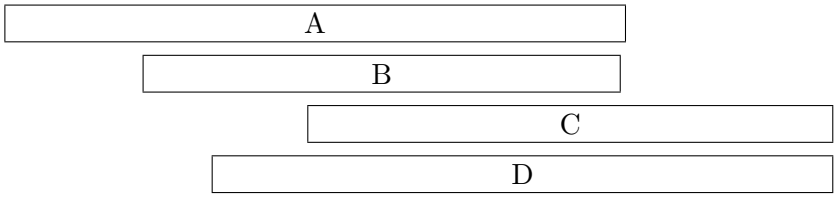
$$a_{11} = b_{11} \qquad a_{12} = b_{12} \tag{8}$$
$$a_{21} = b_{21} \qquad a_{22} = b_{22} + c_{22} \tag{9}$$

```
\begin{flalign*}
a_{11}& =b_{11}&
  a_{12}& =b_{12}\\
a_{21}& =b_{21}&
  a_{22}& =b_{22}+c_{22}
\end{flalign*}
```

$$a_{11} = b_{11} \qquad\qquad a_{12} = b_{12}$$
$$a_{21} = b_{21} \qquad\qquad a_{22} = b_{22} + c_{22}$$

nhận cả dòng cần dịch chuyển làm tham số (nhưng trừ ra \\ ở cuối dòng)

(4.10)




```
\begin{multline}
\framebox[.65\columnwidth]{A}\\
\framebox[.5\columnwidth]{B}\\
%
\shoveright{\framebox[.55\columnwidth]{C}}\\
%
\framebox[.65\columnwidth]{D}
\end{multline}
```

Giái trị của biến độ dài `\multlinegap` có thể thay đổi nhờ các lệnh của LaTeX là `\setlength` và `\addtolength`.

4.4 Chia nhỏ phương trình và canh cột

Giống như `multline`, môi trường `split` dùng biểu diễn *các phương trình đơn* quá dài, không bố trí vừa trên riêng một dòng và do đó phải chia chúng thành nhiều dòng. Nhưng không như `multline`, môi trường `split` cho phép canh cột các dòng nhờ sử dụng `&` để đánh dấu cột. Không như các môi trường phương trình khác của gói `amsmath`, `split` không đánh số phương trình, bởi *nó chỉ có thể dùng bên trong khác môi trường khác*, thường là `equation`, `align`, hay `gather` (các môi trường vừa nhắc đến thực hiện đánh số). Ví dụ:

$$\begin{aligned} H_c = \frac{1}{2n}\sum_{l=0}^{n}(-1)^l(n-l)^{p-2} & \sum_{l_1+\cdots+l_p=l}\prod_{i=1}^{p}\binom{n_i}{l_i} \\ & \cdot[(n-l)-(n_i-l_i)]^{n_i-l_i}\cdot\Big[(n-l)^2-\sum_{j=1}^{p}(n_i-l_i)^2\Big]. \end{aligned} \tag{4.11}$$

```
\begin{equation}\label{e:barwq}
\begin{split}
H_c &=\frac{1}{2n} \sum^n_{l=0}(-1)^{l}(n-{l})^{p-2}
            \sum_{l _1+\dots+ l _p=l}\prod^p_{i=1}\binom{n_i}{l _i}
```

```
\\  &\quad\cdot[(n-l )-(n_i-l _i)]^{n_i-l _i}
        \cdot\Bigl[(n-l )^2-\sum^p_{j=1}(n_i-l _i)^2\Bigr].
\end{split}
\end{equation}
```

Phần phương trình biểu diễn nhờ `split` nên được xem là một phần riêng biệt; lệnh `\label` đặt bên trong môi trường sẽ không có tác dụng.

4.5 Nhóm phương trình không canh cột

Môi trường `gather` thường dùng biểu diễn nhóm các phương trình mà không quan tâm đến việc canh cột giữa chúng; mỗi phương trình trong nhóm sẽ nằm ở dòng riêng (xem Bảng 4.1). Các phương trình bên trong môi trường `gather` được phân biệt nhờ `\\`. Bất kỳ phương trình nào bên trong `gather` cũng có thể là ở dạng `\begin{split}` ... `\end{split}`

```
\begin{gather}
  phương trình thứ nhất\\
  \begin{split}
    phương trình & thứ hai\\
           & với hai dòng
  \end{split}
  \\
  phương trình thứ ba
\end{gather}
```

4.6 Nhóm phương trình có canh cột

Môi trường `align` biểu diễn nhóm các phương trình và cho phép canh cột giữa các phương trình. Xem Bảng 4.1.

Để có vài cột phương trình đi cùng nhau, hãy *dùng thêm* các dấu `&`

$$x = y \qquad X = Y \qquad a = b + c \tag{4.12}$$

$$x' = y' \qquad X' = Y' \qquad a' = b \tag{4.13}$$

$$x + x' = y + y' \qquad X + X' = Y + Y' \qquad a'b = c'b \tag{4.14}$$

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

Có thể thêm chú thích cho phương trình (nằm cùng dòng với phương trình) bằng cách dùng lệnh `\text` bên trong môi trường `align`:

$$
\begin{aligned}
x &= y_1 - y_2 + y_3 - y_5 + y_8 - \dots && \text{theo (4.21)} && (4.15)\\
&= y' \circ y^* && \text{theo (5.1)} && (4.16)\\
&= y(0)y' && \text{theo Tiên đề 1.} && (4.17)
\end{aligned}
$$

```
\begin{align}
x& = y_1-y_2+y_3-y_5+y_8-\dots
                    && \text{theo \eqref{eq:C}}\\
 & = y'\circ y^*     && \text{theo \eqref{eq:D}}\\
 & = y(0) y'         && \text {theo Tiên đề 1.}
\end{align}
```

Biến thể `alignat` của môi trường `align` cho phép xác định khoảng cách (theo chiều ngang), nhờ đó sẽ tiết kiệm không gian hơn. Môi trường này nhận một tham số, là *"số cột phương trình"*. Để có số này, bạn đếm số dấu `&` trong các phương trình, chọn ra số lớn nhất, cộng số đó với 1 rồi chia kết quả cho 2. So sánh ví dụ dưới đây với ví dụ ở ngay trên.

$$
\begin{aligned}
x &= y_1 - y_2 + y_3 - y_5 + y_8 - \dots && \text{theo (4.21)} && (4.18)\\
&= y' \circ y^* && \text{theo (5.1)} && (4.19)\\
&= y(0)y' && \text{theo Tiên đề 1.} && (4.20)
\end{aligned}
$$

```
\begin{alignat}{2}
x& = y_1-y_2+y_3-y_5+y_8-\dots
                   &\quad& \text{theo \eqref{eq:C}}\\
 & = y'\circ y^*    && \text{theo \eqref{eq:D}}\\
 & = y(0) y'        && \text {theo Tiên đề 1.}
\end{alignat}
```

4.7 Canh cột các khối phương trình

Giống như `equation`, các môi trường phương trình nhiều dòng `gather`, `align` và `alignat` bố trí phương trình *trên cả chiều rộng của dòng.* Điều này dẫn tới, chẳng hạn, ta không thể thêm các dấu ngoặc để bao quanh phương trình.

Các môi trường `gathered`, `aligned` và `alignedat` khắc phục nhược điểm trên; chúng biểu diễn phương trình *trên một chiều rộng đúng bằng chiều rộng của nội dung phương trình.* Nhờ đó, chúng có thể xem như một phần của

biểu thức, ví dụ:

$$\left.\begin{aligned} B' &= -\partial \times E, \\ E' &= \partial \times B - 4\pi j, \end{aligned}\right\} \qquad \text{Phương trình Maxwell}$$

```
\begin{equation*}
\left.% dùng \left. để bỏ đi dấu ngoặc bên trái
\begin{aligned}
  B'&=-\partial\times E,\\
  E'&=\partial\times B - 4\pi j,
\end{aligned}
\right\}
\qquad \text{Phương trình Maxwell}
\end{equation*}
```

Giống như `array`, các biến thể `-ed` nhận tham số tùy chọn (bổ sung) `[t]` hoặc `[b]` để chỉ vị trí biểu diễn ('t' cho top-ở trên; 'b' cho bottom-ở dưới)

Biến thể `cases` dùng để *chia các trường hợp*:

$$P_{r-j} = \begin{cases} 0 & \text{nếu } r-j \text{ là số lẻ,} \\ r!\,(-1)^{(r-j)/2} & \text{nếu } r-j \text{ là số chẵn.} \end{cases} \tag{4.21}$$

```
P_{r-j}=\begin{cases}
    0&  \text{nếu $r-j$ là số lẻ},\\
    r!\,(-1)^{(r-j)/2}&  \text{nếu $r-j$ là số chẵn}.
  \end{cases}
```

Để ý đến việc dùng lệnh `\text` (xem Mục §7) và biểu thức toán `$...$` bên trong `\text`.

Cần nhớ: các biến thể `cases` và `-ed` chỉ được dùng bên trong các môi trường phương trình khác—tương tự như `split`.

4.8 Thay đổi vị trí chỉ số phương trình

Thay đổi vị trí đặt chỉ số phương trình là vấn đề khá phức tạp trong biểu diễn phương trình nhiều dòng. Các môi trường của gói `amsmath` luôn cố gắng để không đặt chỉ số phương trình *đè* lên nội dung phương trình, và nếu cần thiết thì đặt chỉ số ở trên hoặc dưới một chút, thậm chí, ở riêng một dòng. Nhưng việc tính chính xác thuộc tính của phương trình là điều khó khăn; do đó, đôi khi việc dịch chuyển chỉ số phương trình do gói `amsmath` thực hiện sẽ không mang lại kết quả đẹp mắt. Trong trường hợp đó, bạn

phải *làm bằng tay* nhờ lệnh `\raisetag`. Để dịch chuyển chỉ số lên phía trên một quãng 6*pt*, dùng `\raisetag{6pt}`; còn để dịch xuống, dùng chẳng hạn `\raisetag{-6pt}`.

Việc điều chỉnh ví trí chỉ số bằng `\raisetag` là công việc tỉ mỉ, cũng như việc ngắt dòng, ngắt trang bằng `\linebreak`, `\pagebreak`. Bạn *chỉ nên* thực hiện điều chỉnh khi tài liệu của bạn gần như hoàn tất, và cần làm lại mỗi khi bạn thay đổi nội dung tài liệu.

4.9 V-khoảng cách và ngắt trang trong biểu diễn nhiều dòng

Bạn có thể dùng `\\[`⟨*dimension*⟩`]` để thêm các v-khoảng cách giữa các dòng trong mọi môi trường biểu diễn phương trình của gói `amsmath`; điều này cũng tương tự trong LaTeX.

Khi dùng gói `amsmath`, việc ngắt trang giữa các dòng của phương trình không (tự động) xảy ra; bởi việc ngắt trang như vậy sẽ làm cho phương trình trở nên khó theo dõi đối với độc giả. Để ngắt trang bên trong phương trình, bạn phải làm bằng tay nhờ lệnh `\displaybreak`. Nơi đặt `\displaybreak` tốt nhất là ngay trước `\\` ở dòng cần ngắt trang. Giống như lệnh `\pagebreak` của LaTeX, lệnh `\displaybreak` nhận tham số bổ sung (tùy chọn) là một trong các số 0, 1, 2, 3, 4; tham số này cho biết mức độ ngắt trang. Dùng `\displaybreak[0]` để hàm ý "không được ngắt trang ở đây",...; trong khi `\displaybreak` (không có tham số bổ sung), như `\displaybreak[4]`, hàm ý "ngắt trang ở đây".

Việc ngắt trang dùng `\displaybreak` như trên chỉ có thể làm đối với phương trình cụ thể. Nếu muốn cung cấp cho gói `amsmath` "giấy phép ngắt trang" một cách toàn cục (cho mọi phương trình nhiều dòng), bạn dùng `\allowdisplaybreaks[1]` trong phần `preamble` của tài liệu. Tham số bổ sung của lệnh này nhận giá trị từ 1 đến 4: `[1]` có nghĩa là cho phép ngắt trang, nhưng tránh việc đó nếu có thể được; các giá trị 2,3,4 càng tăng khả năng ngắt trang. Khi dùng `\allowdisplaybreaks`, thì lệnh `\\*` dùng để cấm xảy ra ngắt trang sau một dòng cụ thể.

Các môi trường `split`, `aligned`, `gathered` và `alignedat` có khả năng `wrap` nội dung phương trình trong các hộp không cho phép ngắt (`unbreakable box`); hệ quả là trong trường hợp đó, cả hai lệnh `\displaybreak` và `\allowdisplaybreaks` đều không có tác dụng.

4.10 Xen liên từ vào giữa các phương trình

Lệnh `\intertext` dùng để xen liên từ[3] vào giữa các dòng của biểu diễn nhiều dòng (xem thêm về `\text` ở Mục §7). Nhờ đó, để bảm đảm việc canh cột các phương trình, bạn đỡ phải tốn công kết thúc một biểu diễn rồi sau đó bắt đầu lại. Lệnh `\intertext` chỉ có thể đặt ngay sau `\\` hoặc `\\*`. Hãy chú ý đến sự xuất hiện của liên từ "và" trong ví dụ sau.

$$A_1 = N_0(\lambda;\Omega') - \phi(\lambda;\Omega'), \tag{4.22}$$

$$A_2 = \phi(\lambda;\Omega') - \phi(\lambda;\Omega), \tag{4.23}$$

và

$$A_3 = \mathcal{N}(\lambda;\omega). \tag{4.24}$$

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

4.11 Đánh số phương trình

4.11.1 Hệ thống chỉ số phương trình

Với LaTeX, nếu bạn muốn đánh số phương trình theo mục—nghĩa là chỉ số phương trình có dạng (1.1), (1.2), ..., (2.1), (2.2), ..., trong chương 1, 2, v.v...—bạn có thể định nghĩa lại lệnh `\theequation` như lời khuyên trong sổ tay (manual) LaTeX [7, §6.3, §C.8.4]:

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

Cách làm như trên cho kết quả như ý, nhưng có điều bất tiện là chỉ số phương trình sẽ không tự động đặt về 0 khi chuyển[4] từ mục này qua mục khác, và bạn phải làm điều đó bằng tay nhờ lệnh `\setcounter`. Để cuộc sống dễ dàng hơn, gói `amsmath` cung cấp lệnh `\numberwithin`; nhờ lệnh này, bạn có thể đánh số phương trình theo mục, với chỉ số phương trình *tự động đặt về 0* khi sang mục mới.

[3] hay cái khác, tùy bạn!

[4] ở mục 1, các phương trình được đánh số (1.1), (1.2), ...; còn qua mục 2, chỉ số phải bắt đầu từ (2.1) rồi đến (2.2), ...; nghĩa là ta phải đặt bộ đếm về 0 khi vừa qua mục 2.

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

Để ý rằng, lệnh `\numberwithin` có tác dụng đối với mọi bộ đếm, chứ không riêng gì bộ đếm `equation`.

4.11.2 Tham khảo chéo đến chỉ số phương trình

Việc tham khảo chéo đến phương trình được thực hiện dễ dàng hơn với lệnh `\eqref`. Lệnh này tự động đặt chỉ số phương trình vào cặp dấu ngoặc đơn. Ví dụ: dùng `\ref{abc}` ta thu được 3.2, trong khi `\eqref{abc}` sẽ cho (3.2).

4.11.3 Đánh số các phương trình con của nhóm phương trình

Gói `amsmath` cung cấp môi trường `subequations` cho phép bạn đánh số các phương trình con của một nhóm các phương trình *một cách tự động*. Ví dụ

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

sẽ khiến các chỉ số của các phương trình bên trong môi trường `subequations` sẽ (tự động) có dạng (4.9a) (4.9b) (4.9c) . . . , nếu phương trình đi trước nhóm phương trình này có chỉ số (4.8). Chú ý là, nếu bạn đặt nhãn tham chiếu (`label`) ngay sau `\begin{subequations}`, việc tham chiếu bằng `\ref` đến nhãn đó sẽ cho ra, chẳng hạn, 4.9, chứ không phải là 4.9a. Vui lòng xem tài liệu `subeqn.pdf` và mã nguồn `subeqn.tex` để có các ví dụ về việc dùng môi trường này.

Các bộ đếm dùng bởi môi trường `subequations` là `parentequation` và `equation`; các lệnh thay đổi bộ đếm `\addtocounter`, `\setcounter`, `\value`, . . . đều có tác dụng bình thường với những bộ đếm đó. Chẳng hạn, các phương trình con được đánh số nhờ chữ cái `alphabet` (a,b,c,. . .); để thay đổi điều này, sử dụng phương pháp chuẩn của LaTeX để thay đổi kiểu đánh số (xem trong [7, §6.3, §C.8.4]). Trong ví dụ dưới đây, các phương trình con sẽ được đánh chỉ số La Mã:

```
\begin{subequations}
\renewcommand{\theequation}{\theparentequation \roman{equation}}
...
```

—5—

Linh tinh, nhưng quan trọng

5.1 Ma trận

Gói `amsmath` cung cấp một số môi trường để biểu diễn ma trận tiện lợi hơn[5] môi trường `array` của LaTeX. Các môi trường `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` và `Vmatrix` tự động có (tương tứng) cặp dấu ngoặc ( ), [ ], { }, | |, và ‖ ‖ bao xung quanh. Ngoài ra, môi trường `matrix` cho một ma trận không có kèm dấu ngoặc nào.

Các môi trường ma trận của `amsmath` sử dụng thuật toán điều chỉnh khoảng cách và canh cột tinh tế, tiết kiệm hơn cách làm hoang phí[6] của môi trường `array`. Hơn nữa, không giống như `array`, bạn không cần chỉ ra *số cột* khi dùng các môi trường `-matrix`; theo mặc định, số cột tối đa của các ma trận là 10—con số này có thể thay đổi được[7]. (Nếu bạn muốn canh phải, trái các cột hoặc muốn tinh chỉnh ma trận theo ý bạn, không còn cách nào khác hơn là bạn phải quay lại dùng môi trường `array`.)

Các ma trận cỡ nhỏ đặt xen vào biểu thức chung dòng được cho bởi môi trường `smallmatrix`. Ví dụ, ma trận $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ được bố trí vừa vẹn trên dòng này, tốt hẳn hơn so với khi bạn thay thế `smallmatrix` bởi `matrix`:

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

Chú ý rằng các dấu ngoặc phải được chỉ ra khi dùng `smallmatrix`; không có các biến thể `p`,`b`,`B`,`v`,`V` của `smallmatrix`.

Có một lệnh hết sức tiện lợi, là `\hdotsfor{`⟨*number*⟩`}`. Lệnh này sẽ sinh ra dãy các dấu chấm lửng trong matrix. Tham số của lệnh là số cột cần bỏ qua. Quan sát ví dụ dưới đây

$$\begin{matrix} a & b & c & d \\ e & \cdots & \cdots & \cdots \end{matrix}$$

```
\begin{matrix} a&b&c&d\\
e&\hdotsfor{3} \end{matrix}
```

[5] `beyond`

[6] `prodigal`

[7] *Cụ thể hơn:* số tối đa các cột của ma trận được xác định bởi bộ đếm `MaxMatrixCols` (mặc định là 10); để thay đổi số này, bạn dùng các lệnh của LaTeX là `\setcounter` hoặc `\addtocounter`.

Khoảng cách giữa các dấu chấm cho bởi `\hdotsfor` có thể thay thế nhờ tham số bổ sung của lệnh; ví dụ: `\hdotsfor[1.5]{3}`. Con số trong ngoặc vuông (`[1.5]`) được dùng như một thừa số nhân; giá trị mặc định là 1.0.

(5.1)
$$\begin{pmatrix} D_1t & -a_{12}t_2 & \dots & -a_{1n}t_n \\ -a_{21}t_1 & D_2t & \dots & -a_{2n}t_n \\ \dots & \dots & \dots & \dots \\ -a_{n1}t_1 & -a_{n2}t_2 & \dots & D_nt \end{pmatrix}$$

```
\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}\\% tăng gấp đôi khoảng cách giữa các dấu chấm
%
-a_{n1}t_1&-a_{n2}t_2&\dots&D_nt\end{pmatrix}
```

5.2 Điều chỉnh khoảng cách

Gói `amsmath` cung cấp tập hợp các lệnh điều chỉnh khoảng cách trong chế độ toán. Phiên bản rút gọn (lệnh ngắn) hoặc đầy đủ (lệnh dài) của các lệnh này đều có tính `robust` và có thể dùng bên ngoài chế độ toán.

Lệnh ngắn	Lệnh dài	Ví dụ	Lệnh ngắn	Lệnh dài	Ví dụ
	(không khoảng cách)	$\Rightarrow\Leftarrow$		(không khoảng cách)	$\Rightarrow\Leftarrow$
`\,`	`\thinspace`	$\Rightarrow\!\Leftarrow$	`\!`	`\negthinspace`	$\Rightarrow\!\!\Leftarrow$
`\:`	`\medspace`	$\Rightarrow\!\!\Leftarrow$		`\negmedspace`	$\Rightarrow\!\!\!\Leftarrow$
`\;`	`\thickspace`	$\Rightarrow\;\Leftarrow$		`\negthickspace`	$\Rightarrow\!\!\!\!\Leftarrow$
	`\quad`	$\Rightarrow\quad\Leftarrow$			
	`\qquad`	$\Rightarrow\qquad\Leftarrow$			

Tổng quát hơn, để điều chỉnh khoảng cách trong chế độ toán, bạn dùng lệnh `\mspace` và *đơn vị toán*. *Đơn vị toán*, ký hiệu là `mu` (`math unit`), là đơn vị độ dài của LaTeX trong chế độ toán, bằng 1/18 em[8]. Ví dụ, để có khoảng `\quad` âm, bạn có thể dùng `\mspace{-18.0mu}`.

5.3 Các dấu chấm

Vị trí của dấu ba chấm (phía trên dòng hay ở ngay chân dòng) trong các ngữ cảnh khác nhau thường không nhất quán, và có thể là vấn đề sở thích cá nhân. Bằng cách sử dụng các lệnh

[8] Thế `em` là cái chi?

- `\dotsc` cho "dấu ba chấm đi với dấu phẩy"
- `\dotsb` cho "dấu ba chấm đi với toán tử nhị phân/quan hệ"
- `\dotsm` cho "dấu ba chấm đi trong biểu thức nhân"
- `\dotsi` cho "dấu ba chấm đi trong biểu thức tích phân"
- `\dotso` cho "dấu ba chấm cho mọi trường hợp còn lại"

thay vì dùng `\ldots` và `\cdots`, bạn có thể làm cho mã nguồn TEX của tài liệu trở nên trong sáng hơn, dễ chỉnh sửa hơn khi cần thiết (chẳng hạn theo yêu cầu của nhà xuất bản). Định nghĩa của các lệnh ở trên là theo quy ước của Hội Toán học Mỹ (AMS).

```
Ta có chuỗi $A_1, A_2, \dotsc$,
tổng vô hạn $A_1 +A_2 +\dotsb $,
tích vô hạn $A_1 A_2 \dotsm $,
và tích phân không xác định
\[\int_{A_1}\int_{A_2}\dotsi\]
```

Ta có chuỗi $A_1, A_2, \dotsc$, tổng vô hạn $A_1 + A_2 + \dotsb$, tích vô hạn $A_1 A_2 \dotsm$, và tích phân không xác định

$$\int_{A_1}\int_{A_2}\dotsi$$

5.4 Gạch ngang không vỡ

Lệnh `\nobreakdash` giúp ta hạn chế sự ngắt dòng sau một dấu `hyphen` hoặc sau dấu gạch ngang. Ví dụ, nếu bạn viết 'các trang 1–9' bằng cách dùng `các trang 1\nobreakdash--9` thì việc ngắt dòng sẽ không bao giờ xảy ra giữa dấu gạch ngang và số 9. Bạn cũng có thể dùng `\nobreakdash` để hạn chế việc tách chữ (`hyphenation`) ngoài ý muốn, ví dụ trong tổ hợp `$p$-adic`. Để tiện lợi, bạn có thể định nghĩa các lệnh tắt như sau:

```
\newcommand{\p}{$p$\nobreakdash}% cho "\p-adic"
\newcommand{\Ndash}{\nobreakdash--}% cho "các trang 1\Ndash 9"
%    Cho không gian "\n chiều" ("n-chiều"):
\newcommand{\n}[1]{$n$\nobreakdash-\hspace{0pt}}
```

Ví dụ cuối cùng ở trên cho biết cách làm thế nào để ngăn cản ngắt dòng sau dấu `hyphen` nhưng cho phép tách chữ bình thường ở từ tiếp theo dấu `hyphen` đó. (Chỉ cần thêm một độ rộng 0 sau dấu `hyphen`.)

5.5 Dấu nhấn trong toán học

Thông thường, LATEX khó có thể biểu diễn cho đẹp cùng lúc hai dấu nhấn trên một ký hiệu. Với gói `amsmath`, việc này khá hơn rất nhiều; ví dụ $\hat{\hat{A}}$ (cho bởi `\hat{\hat{A}}`).

Các lệnh `\dddot` và `\ddddot` biểu diễn hai và ba dấu chấm như là dấu nhấn, ví dụ $\dddot{C}$. Đây là các lệnh do `amsmath` cung cấp; còn `\dot` và `\ddot` là các lệnh của LaTeX.

Để có được chỉ số trên là dấu mũ hoặc dấu ngã, dùng gói `amsxtra` và các lệnh của gói này là `\sphat`, `\sptilde`. Ví dụ, $A^{\wedge}$ cho bởi `A\sphat` (để ý ở đây ta không dùng dấu `^`).

Để đặt ký hiệu bất kỳ vào vị trí của dấu nhấn, hoặc để có các dấu nhấn bên dưới, vui lòng xem tài liệu hướng dẫn gói `accents`(viết bởi Javier Bezos).

5.6 Căn số

Cách biểu diễn căn số của LaTeX nhiều khi không thật đẹp mắt, như trong ví dụ $\sqrt[\beta]{k}$ (cho bởi `\sqrt[\beta]{k}`). Gói `amsmath` cung cấp các lệnh `\leftroot` (dịch qua trái) và `\uproot` (dịch lên trên) cho phép bạn điều chỉnh vị trí của bậc căn số[9]. Chẳng hạn

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

sẽ cho $\sqrt[\beta]{k}$. Nếu bạn dùng `dimension` âm làm tham số của `\leftroot` (như trong ví dụ trên), thì bậc căn số sẽ dịch chuyển qua phải.

5.7 Đóng khung biểu thức

Biểu thức `\boxed` sẽ đóng khung chữ nhật quanh nội dung là tham số của nó; lệnh này tương tự như lệnh `\fbox`, nhưng nội dung của lệnh ở chế độ toán.

$$\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))} \tag{5.2}$$

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

5.8 Mũi tên bên trên, bên dưới

LaTeX chuẩn chỉ cung cấp các mũi tên bên-trên-qua-trái hoặc bên-trên-qua-phải nhờ các lệnh `\overrightarrow`, `\overleftarrow`. Dưới đây là các dạng mũi tên khác do gói `amsmath` cung cấp.

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

Ví dụ, `\overleftarrow{XYZ}` sẽ cho bạn $\overleftarrow{XYZ}$.

[9] bậc của căn số, ví dụ β trong $\sqrt[\beta]{k}$.

5.9 Mũi tên có chỉ số.

Các lệnh `\xleftarrow` và `\xrightarrow` cho ta các mũi tên *tự động điều chỉnh độ dài cho bằng với nội dung của chỉ số trên hoặc dưới*. Tham số bổ sung của hai lệnh này là chỉ số dưới, còn tham số bắt buộc là chỉ số trên. Cả chỉ số trên và chỉ số dưới của hai lệnh này đều có thể rỗng.

$$A \xleftarrow{n+\mu-1} B \xrightarrow[T]{n\pm i-1} C \tag{5.3}$$

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

5.10 Gắn các ký hiệu với nhau

LaTeX cung cấp lệnh `\stackrel` để đặt chỉ số trên cho các quan hệ (toán tử) nhị phân. Với gói `amsmath`, các lệnh `\overset` và `\underset` còn làm được việc tổng quát hơn: đặt một ký hiệu bên trên hoặc bên dưới một ký hiệu khác, bất kể ký hiệu đó có phải là toán tử nhị phân hay không. Ví dụ, với `\overset{*}{X}` sẽ đặt dấu sao $*$ bên trên chữ X: $\overset{*}{X}$; lệnh `\underset` cũng tương tự, nhưng nó đặt chỉ số bên dưới ký hiệu.

Xem thêm cách dùng `\sideset` ở Mục §8.2.

5.11 Phân số và cấu trúc liên quan

5.11.1 Các lệnh `\frac`, `\dfrac`, và `\tfrac`

Lệnh `\frac`, là lệnh cơ bản của LaTeX, nhận hai tham số—tử số và mẫu số—và biểu diễn phân số theo kiểu thông thường trong toán học. Để ý rằng, các phân số ở chế độ toán chung dòng thường nhỏ; để khiến chúng to hơn, bạn dùng `\displaystyle\frac...`. Để tiện lợi, gói `amsmath` cung cấp thêm các lệnh: `\dfrac`, `\tfrac` lần lượt là dạng viết tắt của `{\displaystyle\frac ... }` và `{\textstyle\frac ... }`. Ví dụ

$$\frac{1}{k}\log_2 c(f) \quad \tfrac{1}{k}\log_2 c(f) \quad \sqrt{\frac{1}{k}\log_2 c(f)} \quad \sqrt{\dfrac{1}{k}\log_2 c(f)} \tag{5.4}$$

```
\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}
```

5.11.2 Ký hiệu Tổ hợp: `\binom`, `\dbinom`, `\tbinom`

Để biểu diễn tổ hợp dạng $\binom{n}{k}$, bạn dùng các lệnh `\binom`, `\dbinom` và `\tbinom`. Các tiếp vĩ ngữ `d` và `t` được hiểu tương tự như ở các lệnh về

phân số.

$$2^k - \binom{k}{1}2^{k-1} + \binom{k}{2}2^{k-2} \tag{5.5}$$

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

5.11.3 Tạo phân số tổng quát với \genfrac

Khả năng của các lệnh \frac, \binom và các biến thể có thể có được nhờ lệnh \genfrac. Đây là lệnh tạo phân số tổng quát với sáu tham số. Hai tham số cuối cùng của \genfrac lần lượt là *tử số* và *mẫu số*; hai tham số bổ sung đầu tiên là các dấu ngoặc trái và phải (như trong kết quả của \binom); tham số thứ ba là độ dày của đường kẻ ngang giữa tử số và mẫu số (lệnh \binom dùng độ dày 0, nghĩa là *không thấy được*); tham số thứ tư xác định chế độ toán trong đó phân số được biểu diễn: tham số này nhận giá trị 0,1,2,3 tương ứng với các chế độ \displaystyle, \textstyle, \scriptstyle, \scriptscriptstyle.

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

Ví dụ, các lệnh \frac, \tfrac và \binom có thể được định nghĩa như sau:

```
\newcommand{\frac}[2]{\genfrac{}{}{}{}{#1}{#2}}
\newcommand{\tfrac}[2]{\genfrac{}{}{}{1}{#1}{#2}}
\newcommand{\binom}[2]{\genfrac{(}{)}{0pt}{}{#1}{#2}}
```

Các lệnh nguyên thủy \over, \overwithdelims, \atop, \atopwithdelims, \above, \abovewithdelims khi dùng với gói amsmath sẽ sinh ra cảnh báo lỗi. Vui lòng xem thêm tài liệu technote.tex có trong texmf/source/latex/amsmath/

5.12 Phân số liên tục

Phân số liên tục

$$\cfrac{1}{\sqrt{2}+\cfrac{1}{\sqrt{2}+\cfrac{1}{\sqrt{2}+\cdots}}} \tag{5.6}$$

có thể thu được nhờ

```
\cfrac{1}{\sqrt{2}+
 \cfrac{1}{\sqrt{2}+
  \cfrac{1}{\sqrt{2}+\dotsb
}}}
```

Việc dùng `\cfrac` cho kết quả dễ nhìn hơn so với khi bạn tạo phân số liên tục bằng `\frac`. Bạn có thể đặt vị trí của tử số ở bên trái hoặc bên phải tùy ý, bằng cách dùng `\cfrac[l]` hoặc `\cfrac[r]` thay vì `\cfrac`.

5.13 Lệnh `\smash` đặt độ cao/sâu về 0

Lệnh `\smash` khiến cho các biểu thức con có chiều cao (`heigth`) và chiều sâu (`depth`) bằng 0; việc dùng biểu thức con như vậy rất có hữu ích, chẳng hạn khi điều chỉnh vị trí của biểu thức con theo các ký hiệu. Gói `amsmath` cung cấp lệnh `\smash` với tham số tùy chọn là `t` và `b`, bởi đôi khi ta chỉ cần làm cho chiều cao (ứng với `t`) hoặc chiều sâu (ứng với `b`) trở về 0.

Ví dụ, chiều cao của các dấu căn bậc hai trong một biểu thức không bằng nhau vì chiều cao đó tùy thuộc vào nội dung bên dưới dấu căn. (Xem ở độ phóng đại lớn,) bạn sẽ thấy dấu căn trong $\sqrt{y}$ trong biểu thức $\sqrt{x}+\sqrt{y}+\sqrt{z}$ đi xuống phía dưới chân dòng hơn so với hai dấu căn còn lại. Ta sẽ dùng `\smash[b]` để đặt độ sâu của chữ y về 0, nhờ đó, dấu căn trong $\sqrt{y}$ sẽ giống hệt dấu căn trong $\sqrt{x}$:

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

Bây giờ, bạn có $\sqrt{x} + \sqrt{y} + \sqrt{z}$. Bạn có thấy sự khác biệt ở biểu thức này với biểu thức trước kia không?

5.14 Dấu ngoặc

5.14.1 Kích cỡ dấu ngoặc

Việc dùng dấu ngoặc tự động điều chỉnh kích cỡ với `\left` và `\right` có hai hạn chế: *thứ nhất,* các lệnh này máy móc điều chỉnh cho dấu ngoặc có chiều cao vừa bằng với chiều cao của nội dung cao nhất, và *thứ hai,* tập hợp các kích cỡ do chúng tạo ra không biến thiên đều đặn, mà có những bước nhảy quá lớn[10].

Trong thực tế, thường có hai hoặc ba trường hợp ta cần điều chỉnh kích cỡ của dấu ngoặc nhờ vào các lệnh 'big' của gói `amsmath`.

[10] *Nghĩa là:* hai dấu ngoặc kế tiếp nhau trong cùng biểu thức tạo bởi `\left` hoặc `\right` có thể có chiều cao chênh nhau đến 3pt hoặc nhiều hơn, và như thế sẽ quá xấu!

Cỡ ngoặc	Cỡ text	\left \right	\bigl \bigr	\Bigl \Bigr	\biggl \biggr	\Biggl \Biggr
Kết quả	$(b)(\frac{c}{d})$	$(b)\left(\frac{c}{d}\right)$	$(b)(\frac{c}{d})$	$\Bigl(b\Bigr)\Bigl(\frac{c}{d}\Bigr)$	$\biggl(b\biggr)\biggl(\frac{c}{d}\biggr)$	$\Biggl(b\Biggr)\Biggl(\frac{c}{d}\Biggr)$

Trường hợp đầu tiên, là khi xảy ra sự chồng chất (`cumulative`) các toán tử với chỉ số trên và dưới. Với `\left` và `\right`, các dấu ngoặc tạo ra thường lớn hơn nhiều so với ý muốn; việc dùng `Big` or `bigg` thay thế cho `\left` và `\right` cho kết quả khả quan hơn hẳn:

$$\left[\sum_i a_i \left|\sum_j x_{ij}\right|^p\right]^{1/p} \quad \text{so với} \quad \biggl[\sum_i a_i \Bigl|\sum_j x_{ij}\Bigr|^p\biggr]^{1/p}$$

```
\biggl[\sum_i a_i\Bigl\lvert\sum_j x_{ij}\Bigr\rvert^p\biggr]^{1/p}
```

Trường hợp thứ hai, là khi các dấu ngoặc lồng nhau liên tiếp. Trong trường hợp này, lệnh `\left` và `\right` cho ra các dấu ngoặc cùng một kích cỡ[11], trong dấu ngoặc càng ở bên ngoài càng cần có kích thước lớn hơn.

$$\left((a_1b_1)-(a_2b_2)\right)\left((a_2b_1)+(a_1b_2)\right) \quad \text{versus} \quad \Big((a_1b_1)-(a_2b_2)\Big)\Big((a_2b_1)+(a_1b_2)\Big)$$

```
\left((a_1 b_1) - (a_2 b_2)\right)
\left((a_2 b_1) + (a_1 b_2)\right)
\quad\text{so với}\quad
\bigl((a_1 b_1) - (a_2 b_2)\bigr)
\bigl((a_2 b_1) + (a_1 b_2)\bigr)
```

Trường hợp thứ ba, là khi biểu diễn các biểu thức quá khổ trong một dòng, chẳng hạn biểu thức $\left|\frac{b'}{d'}\right|$ trong dòng này. Các dấu ngoặc tạo bởi `\left` và `\right`, như bạn thấy, sẽ làm cho dòng bị dãn quá nhiều theo chiều đứng. Khi đó, việc dùng `\bigl` và `\bigr` chỉ cho ra dấu ngoặc vừa đủ cao và nhờ đó dòng không bị v-dãn nhiều quá, như trong $\bigl|\frac{b'}{d'}\bigr|$.

Các lệnh `\big`, `\bigg`, `\Big` và `\Bigg` mà LaTeX cung cấp không thay đổi kích cỡ dấu ngoặc một cách chính xác trong các kích cỡ `font` khác nhau của LaTeX. Gói `amsmath` khắc phục nhược điểm đó.

[11] because that is adequate to cover the encompassed material.

5.14.2 Ký hiệu |

Gói `amsmath` cung cấp các lệnh `\lvert`, `\rvert`, `\lVert`, `\rVert` (so sánh với `\langle`, `\rangle`) nhằm giải quyết vấn đề trùng hợp ý nghĩa của ký tự `|`. Ký tự này được dùng trong tài liệu LaTeX để chỉ nhiều đối tượng toán học khác nhau: chỉ quan hệ 'ước số' trong lý thuyết số, ví dụ như trong biểu thứ $p|q$; chỉ giá trị tuyệt đối $|z|$; hay là viết tắt của 'sao cho' khi biểu diễn tập hợp; hoặc hàm ý 'lấy giá trị tại' trong ký hiệu $f_\zeta(t)\big|_{t=0}$. Tính đa nghĩa của ký hiệu | không phải là điều gì xấu; cái không hay là ở chỗ: không phải tất cả cách dùng của | đều có cùng cách thể hiện xét về mặt in ấn (`typographical`), và sự phân biệt phức tạp của độc giả có kiến thức không thể nào thể hiện được trong `typeset`. Vì vậy, trong bất kỳ tài liệu nào, bạn cũng nên cho tương ứng ký tự `|` với một khái niệm toán học cụ thể; và cũng nên làm điều tương tự với ký tự `\|`. Điều này không bao gồm cách dùng `|` và `\|` như các dấu ngoặc, bởi vì cách dùng dấu ngoặc trái hoặc phải khác hẳn với các cách dùng của | như kể trên.

Ví dụ, bạn nên định nghĩa

```
\providecommand{\abs}[1]{\lvert#1\rvert}
\providecommand{\norm}[1]{\lVert#1\rVert}
```

và suốt trong tài liệu của mình, bạn dùng `\abs{z}` để có $|z|$ và dùng `\norm{v}` để có $\|v\|$.

—6—

Tên toán tử

6.1 Định nghĩa toán tử mới

Các toán tử toán học như log, sin, lim theo truyền thống được `typeset` với kiểu chữ `roman` để có thể dễ dàng phân biệt chúng với các tên biến toán học (các tên biến được `typeset` với `font` nghiêng). Các toán tử thường gặp nhất là `\log`, `\sin`, `\lim`,... đã được định nghĩa; còn các toán tử mới có thể định nghĩa dễ dàng nhờ lệnh `\DeclareMathOperator`. Lệnh này chỉ có thể đặt trong phần `preamble` của tài liệu. Chẳng hạn, để định nghĩa hàm số `\tg` (là cách viết hàm số `tangement` theo kiểu Việt Nam), có thể dùng

```
\DeclareMathOperator{\tg}{tg}
```

nhờ đó, việc dùng `\tg` sẽ cho ra tg với `font` chính xác đồng thời tự động thêm cách khoảng cách xung quanh toán tử tg khi cần thiết, nhờ đó, ta thu được $A(\operatorname{tg} x)B$ thay vì $A(\mathrm{tg}x)B$.

Nếu toán tử mới cần các chỉ số trên và chỉ số dưới, như trong các toán tử lim, sup, or max, thì bạn dụng dạng * của lệnh `\DeclareMathOperator`:

```
\DeclareMathOperator*{\Lim}{Lim}
```

Xem thêm ở Mục 8.3 về việc đặt chỉ số dưới.

Các toán tử được định nghĩa trước là

`\arccos`	$\arccos$	`\deg`	$\deg$	`\lg`	$\lg$	`\projlim`	$\projlim$
`\arcsin`	$\arcsin$	`\det`	$\det$	`\lim`	$\lim$	`\sec`	$\sec$
`\arctan`	$\arctan$	`\dim`	$\dim$	`\liminf`	$\liminf$	`\sin`	$\sin$
`\arg`	$\arg$	`\exp`	$\exp$	`\limsup`	$\limsup$	`\sinh`	$\sinh$
`\cos`	$\cos$	`\gcd`	$\gcd$	`\ln`	$\ln$	`\sup`	$\sup$
`\cosh`	$\cosh$	`\hom`	$\hom$	`\log`	$\log$	`\tan`	$\tan$
`\cot`	$\cot$	`\inf`	$\inf$	`\max`	$\max$	`\tanh`	$\tanh$
`\coth`	$\coth$	`\injlim`	$\injlim$	`\min`	$\min$		
`\csc`	$\csc$	`\ker`	$\ker$	`\Pr`	$\Pr$		

`\varlimsup`	$\varlimsup$	`\varinjlim`	$\varinjlim$
`\varliminf`	$\varliminf$	`\varprojlim`	$\varprojlim$

Ngoài ra, còn có lệnh `\operatorname`: ví dụ

```
\operatorname{abc}
```

trong biểu thức toán học tương đương với việc khai báo nhờ `\DeclareMathOperator` rồi dùng `\abc`. Lệnh `\operatorname` chỉ nên dùng nếu toán tử thi thoảng gặp trong tài liệu; còn nếu toán tử xuất hiện nhiều lần, bạn nên khai báo toán tử mới nhờ `\DeclareMathOperator`.

Để ý dùng `\operatorname*` trong trường hợp toán tử cần chỉ số.

6.2 Ký hiệu Đồng dư

Các lệnh `\mod`, `\bmod`, `\pmod`, `\pod` biểu diễn các ký hiệu đồng dư. Lệnh `\bmod` và `\pmod` đã có trong LaTeX, nhưng trong gói `amsmath`, chúng được tinh chỉnh để cho ra khoảng cách tốt hơn trong các biểu thức chung dọng. Các lệnh `\mod` và `\pod` là biến thể của `\pmod`; lệnh `\mod` không tạo ra cặp dấu ngoặc đơn, trong khi lệnh `\pod` lại bỏ "mod" và các thứ khác vào cặp dấu ngoặc.

(6.1) $\gcd(n, m \bmod n)$; $x \equiv y \pmod{b}$; $x \equiv y \mod{c}$; $x \equiv y \pmod{d}$

```
\gcd(n,m\bmod n);\quad x\equiv y\pmod b;
\quad x\equiv y\mod c;\quad x\equiv y\pod d
```

—7—

Lệnh \text chèn text vào biểu thức

Mục đích chính của lệnh `\text` là để chèn văn bản, `text` vào giữa các biểu thức toán học. Lệnh có tác dụng tương tự như lệnh `\mbox` của LaTeX, nhưng dùng `\text` hay hơn ở nhiều điểm. Nếu bạn muốn văn bản thêm vào ở cỡ chữ `sub-script`, bạn có thể dùng `..._{\text{từ hoặc câu}}`, trong khi với `\mbox`, bạn phải dùng `..._{\mbox{\scriptsize từ hoặc câu}}`. Điểm hay khác, là tên của lệnh `\text` phản ánh được đầy đủ tác dụng của nó hơn lệnh `\mbox`.

(7.1) $$f_{[x_{i-1},x_i]} \text{ là đơn điệu,} \quad i = 1,\dots,c+1$$

```
f_{[x_{i-1},x_i]} \text{ là đơn điệu,}
\quad i = 1,\dots,c+1
```

—8—

Tích phân và Tổng

8.1 Chỉ số trên/dưới nhiều dòng

Nhờ lệnh `\substack`, bạn biểu diễn các chỉ số trên, dưới với nhiều dòng. Ví dụ

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

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

Tổng quát hơn, môi trường `subarray` còn cho phép bạn canh trái các dòng của chỉ số trên/dưới, thay vì canh giữa như mặc định. Hãy để ý đến tham số `{l}` của `subaray` trong ví dụ dưới đây.

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

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

8.2 Lệnh \sideset

Lệnh \sideset dùng cho mục đích đặc biệt, là đặt các ký hiệu vào vị trí của chỉ số trên và chỉ số dưới của các toán tử lớn như $\sum$ hay $\prod$. *Chú ý rằng, lệnh này không có mục đích áp dụng vào các ký hiệu không thuộc lớp ký hiệu tổng.* Ví dụ dễ thấy nhất là khi bạn muốn có một dấu phảy để chỉ biến thể của ký hiệu tổng:

$$\sum\nolimits' E_n \tag{8.1}$$

Để được như trên, bạn có thể dùng \sum\nolimits' E_n; ở đây, \nolimits phải được dùng, bởi nếu không, dấu phảy sẽ được đặt ở bên trên chứ không phải là góc trên-bên-phải của ký hiệu tổng. Nếu bạn làm như vậy, thì nảy ra vấn đề nan giải là, bây giờ khó có thể đặt thứ gì vào bên trên và bên dưới của $\sum$ nữa (bởi bạn đã dùng \nolimits). Cách giải quyết vấn đề này nhờ lệnh \sideset được minh họa trong ví dụ sau:

```
\sideset{}{'}
  \sum_{n<k,\;\text{$n$ odd}} nE_n
```

$$\sum_{n<k,\ n\text{ odd}}' nE_n$$

Phải dùng cặp dấu ngoặc rỗng ({}) như trên, là bởi lệnh \sideset có khả năng đặt các ký hiệu vào bốn góc của toán tử lớn. Hãy học hỏi ví dụ rất thú vị sau đây:

```
\sideset{_*^*}{_*^*}\prod_{n=1}^{\infty}
```

$$\sideset{_*^*}{_*^*}\prod_{n=1}^{\infty}$$

8.3 Vị trí của chỉ số dưới và 'limit'

Vị trí mặc định của chỉ số dưới của ký hiệu phụ thuộc vào tính chất ký hiệu đó. Với các ký hiệu dạng tổng ($\sum$, $\prod$), vị trí đó là 'displaylimits': nghĩ là, khi ký hiệu dạng tổng xuất hiện trong biểu thức riêng dòng, chỉ số trên và chỉ số dưới được bố trí bên trên hoặc bên dưới ký hiệu; còn nếu ký hiệu đó xuất hiện trong chế độ chung dòng, thì các chỉ số trên, dưới được đặt ở bên cạnh ký hiệu, nhằm tránh việc dãn dòng qua mức cần thiết. Đối với các ký hiệu dạng tích phân, vị trí mặc định của các chỉ số luôn là ở bên cạnh ký hiệu, ngay cả trong các công thức riêng dòng. (Nhưng xem về tùy chọn intlimits trong Mục 3.)

Các toán tử như sin, lim đều có vị trí chỉ số thuộc loại 'displaylimits' hay 'limits' tùy theo cách chúng được định nghĩa, và điều đã được chọn theo thói quen phổ biến trong giới toán học.

Các lệnh `\limits` và `\nolimits` dùng để bật qua lại giữa hai chế độ đặt vị trí chỉ số:

$$\sum\nolimits_X, \qquad \iint\limits_A, \qquad \varliminf\nolimits_{n\to\infty}$$

```
\sum\nolimits_X,\qquad \iint\limits_{A},
\qquad\varliminf\nolimits_{n\to \infty}
```

Để định nghĩa một lệnh mới mà chỉ số dưới của nó có thuộc tính 'displaylimits', bạn đặt lệnh `\displaylimits` và cuối định nghĩa của lệnh. Trong dãy gồm các lệnh `\limits`, `\nolimits`, `\displaylimits` tiếp nhau, thì lệnh cuối cùng có tác dụng ưu tiên.

8.4 Dấu tích phân bội

Các lệnh `\iiint`, `\iiint` và `\iiiint` cho ta các dấu tích phân bội với khoảng cách giữa các dấu tích phân được tinh chỉnh tốt hơn. Lệnh `\idotsint` cho ta dấu tích phân bội với số lớp tùy ý.

$$\iint_A f(x,y)\,dx\,dy \qquad \iiint_A f(x,y,z)\,dx\,dy\,dz \tag{8.2}$$

$$\iiiint_A f(w,x,y,z)\,dw\,dx\,dy\,dz \qquad \int_A\cdots\int f(x_1,\ldots,x_k) \tag{8.3}$$

—9—

Biểu đồ giao hoán

Một số lệnh tạo biểu đồ giao hoán như trong $\mathcal{AMS}$-TEX được gộp thành gói riêng biệt, là `amscd`. Với các biểu đồ phức tạp, bạn nên dùng gói chuyên dụng hơn là `kuvio` hoặc `xypic`; nhưng với các biểu đồ đơn giản, không có các mũi tên chéo, thì dùng gói `amscd` sẽ nhanh chóng cho kết quả. Dưới đây là ví dụ.

$$\begin{array}{ccc} S^{\mathcal{W}_\Lambda}\otimes T & \xrightarrow{\ j\ } & T \\ \big\downarrow & & \big\downarrow{\scriptstyle \operatorname{End} P} \\ (S\otimes T)/I & = & (Z\otimes T)/J \end{array}$$

```
\begin{CD}
S^{{\mathcal{W}}_\Lambda}\otimes T   @>j>>   T\\
```

```
@VVV                                    @VV{\End P}V\\
(S\otimes T)/I                  @=      (Z\otimes T)/J
\end{CD}
```

Trong môi trường `CD`, các lệnh `@>>>`, `@<<<`, `@VVV` và `@AAA` lần lượt cho các mũi tên qua phải, trái, xuống, lên. Với các mũi tên ngang, mọi thứ đặt giữa hai ký hiệu `>` hoặc `<` đầu tiên sẽ được xem là chỉ số trên, các đối tượng đặt giữa ký hiệu thứ hai và thứ ba được xem là chỉ số dưới. Tương tự, đối tượng đặt giữa ký hiệu `A` hay `V` thứ nhất và thứ hai sẽ là chỉ số bên trái, còn đối tượng đặt giữa ký hiệu `A` hay `V` thứ nhì và thứ ba sẽ là chỉ số bên phải. Các lệnh `@=` và `@|` sẽ cho các đường-ngang-kép và đường-đứng-kép. Một "mũi tên rỗng" được cho bởi lệnh `@.`.

—10—

Sử dụng 'font' toán

10.1 Giới thiệu

Mọi vấn đề về `font` có thể tìm thấy trong hướng dẫn (`fntguide.tex`) hoặc *The LaTeX Companion* [4].

Các lệnh cơ bản (của LaTeX) điều chỉnh `font` trong chế độ toán bao gồm: `\mathbf`, `\mathrm`, `\mathcal`, `\mathsf`, `\mathtt`, `\mathit`. Ngoài ra, gói `amsfonts` cung cấp các lệnh đặc biệt: `\mathbb` cho kiểu 'blackboard bold' ($\mathbb{R}$), `\mathfrak` cho kiểu 'Fraktur' ($\mathfrak{R}$).

10.2 Lời khuyên

Nếu bạn phải thường xuyên thay đổi kiểu `font` trong chế độ toán, bạn có thể nghĩ đến các định nghĩa vắn tắt, chẳng hạn, dùng `\mb` thay vì `\mathbf`. Tất nhiên, không gì ngăn cản bạn làm điều đó, vì bạn có trong tay lệnh-tạo-định-nghĩa `\newcommand`. Nhưng với LaTeX, việc đưa ra các dạng định nghĩa vắn tắt như vậy cần nên tránh, cần thay thế bởi giải pháp hay hơn: *định nghĩa một lệnh dựa trên* thuộc tính của đối tượng toán học*, hơn là dựa trên tên của* `font` *dùng để phân biệt các đối tượng.* Ví dụ, nếu bạn muốn dùng `font` đậm chỉ các véctơ, bạn nên định nghĩa một lệnh tạo véctơ với `font` đậm, hơn là định nghĩa một lệnh tạo `font` đậm để tạo véctơ; nhờ

```
\newcommand{\vect}[1]{\mathbf{#1}}
```

bạn có thể viết `\vect{a} + \vect{b}` để có $\mathbf{a} + \mathbf{b}$; bạn không nên dùng, chẳng hạn `\mb{a}+\mb{b}` với lệnh `\mb` là dạng viết tắt của `\mathbf`. Sau

một vài tháng chẳng hạn, bạn thấy cần phải dùng `font` đậm cho mục đích khác, chứ không phải để chỉ các véctơ, và muốn các véctơ bây giờ có thêm các mũi tên, thì bạn có thể dễ dàng thực hiện được ý đồ của mình nếu bạn dùng định nghĩa `\vect` để tạo véctơ; còn nếu ngược lại, bạn phải thay thế toàn bộ các lệnh `\mb` trong tài liệu của bạn—điều này nguy hiểm ở chỗ, việc thay thế có thể ảnh hưởng đến các đối tượng không là véctơ nhưng được biểu diễn bằng `\mb`.

Sẽ có ích nếu bạn tạo các lệnh mới để lấy ra các ký tự đặc biệt trong một `font` cụ thể nào đó, ví dụ

```
\DeclareSymbolFont{AMSb}{U}{msb}{m}{n}% or use amsfonts package
\DeclareMathSymbol{\C}{\mathalpha}{AMSb}{"43}
\DeclareMathSymbol{\R}{\mathalpha}{AMSb}{"52}
```

Các dòng ở trên định nghĩa các lệnh `\C`, `\R` để lấy ra ký hiệu 'blackboard' từ bộ 'AMSb' là `font` các ký hiệu. Nếu bạn thường xuyên làm việc với các trường số thực, phức, bạn có thể có một cách tiện lợi, là định nghĩa lệnh `\field` và sau đó dùng `\field{C}`, `\field{R}`,... để có các trường số mong muốn. Nhưng để *tăng tối đa tính uyển chuyển và khả năng điều khiển*, bạn hãy định nghĩa lệnh `\field` rồi sau đó định nghĩa các lệnh `\C`, `\R` dựa trên `\field` như sau:

```
\usepackage{amsfonts}% to get the \mathbb alphabet
\newcommand{\field}[1]{\mathbb{#1}}
\newcommand{\C}{\field{C}}
\newcommand{\R}{\field{R}}
```

10.3 Các ký hiệu in đậm

Lệnh `\mathbf` thường được dùng để có được phiên bản đậm của các chữ cái trong chế độ toán, nhưng với hầu hết các ký hiệu toán khác (không phải là chữ cái), lệnh này không có tác dụng, hoặc có tác dụng nhưng kết quả của nó không có liên hệ chặt chẽ với `font` đang được dùng. Ví dụ, viết

```
\Delta \mathbf{\Delta}\mathbf{+}\delta \mathbf{\delta}
```

sẽ cho bạn $\Delta\mathbf{\Delta}+\delta\mathbf{\delta}$; để ý rằng, trong kết quả thu được, các dấu cộng và dấu δ không bị ảnh hưởng bởi `\mathbf`.

Gói `amsmath` vì vậy cung cấp thêm hai lệnh mới, là `\boldsymbol` và `\pmb`; các lệnh này có tác dụng làm đậm mọi loại ký hiệu. Lệnh `\boldsymbol` có thể được dùng cho các ký hiệu toán không bị ảnh hưởng bởi lệnh `\mathbf`

nếu và chỉ nếu `font` toán đang dùng có phiên bản **đậm** tương ứng của ký hiệu đó. Lệnh `\pmb` được dùng, như là giải pháp cuối cùng, cho trường hợp lệnh `\boldsymbol` bất lực; "pmb" là viết tắt của "poor man's bold"; nguyên lý làm việc của lệnh `\pmb` là `typeset` nhiều bản sao của ký hiệu và sắp xếp các bản sao đó sát bên nhau (giống như khi bạn dùng bút kẻ nhiều đường liên tiếp nhau trên giấy để có một đường đậm nét). Rõ ràng, chất lượng kết quả của lệnh `\pmb` là không tốt lắm, nhất là độ nét của ký hiệu. Trong khi bộ `font` chuẩn của LaTeX được dùng để `typeset` các biểu thức toán, là bộ `font` CM (Computer Modern), thì cần đến lệnh `\pmb` chỉ là các ký hiệu toán tử lớn, như `\sum`, các dấu ngoặc quá cỡ, hay là các ký hiệu toán học bổ sung bởi gói `amssymb` (xem [1]).

Các biểu thức sau cho thấy một số kết quả khả dĩ:

```
A_\infty + \pi A_0
\sim \mathbf{A}_{\boldsymbol{\infty}} \boldsymbol{+}
  \boldsymbol{\pi} \mathbf{A}_{\boldsymbol{0}}
\sim\pmb{A}_{\pmb{\infty}} \pmb{+}\pmb{\pi} \pmb{A}_{\pmb{0}}
```

$$A_\infty + \pi A_0 \sim \mathbf{A}_{\boldsymbol{\infty}} \boldsymbol{+} \mathbf{A}_{\boldsymbol{0}} \sim \boldsymbol{A}_{\boldsymbol{\infty}} \boldsymbol{+} \boldsymbol{\pi} \boldsymbol{A}_{\boldsymbol{0}}$$

Nếu bạn muốn dùng lệnh `\boldsymbol` độc lập mà không tải gói `amsmath`, thì hãy dùng gói `bm`; đây là gói thuộc chuẩn LaTeX, không thuộc bộ phân phối của các gói AMS. Gói này tích hợp trong bộ LaTeX phiên bản năm 1997 hoặc cao hơn.

10.4 Các chữ cái Hy Lạp in nghiêng

`\varGamma`	$\varGamma$	`\varSigma`	$\varSigma$
`\varDelta`	$\varDelta$	`\varUpsilon`	$\varUpsilon$
`\varTheta`	$\varTheta$	`\varPhi`	$\varPhi$
`\varLambda`	$\varLambda$	`\varPsi`	$\varPsi$
`\varXi`	$\varXi$	`\varOmega`	$\varOmega$
`\varPi`	$\varPi$		

—11—

Lỗi thường gặp khi dùng gói amsmath

11.1 General remarks

This is a supplement to Chapter 8 of the LaTeX manual [7] (first edition: Chapter 6). For the reader's convenience, the set of error messages discussed here overlaps somewhat with the set in that chapter, but please be aware that we don't provide exhaustive coverage here. The error messages are arranged in alphabetical order, disregarding unimportant text such as `! LaTeX Error:` at the beginning, and nonalphabetical characters such as \. Where examples are given, we show also the help messages that appear on screen when you respond to an error message prompt by entering `h`.

There is also a section discussing some output errors, i.e., instances where the printed document has something wrong but there was no LaTeX error during typesetting.

11.2 Error messages

`\begin{split} won't work here.`
Example:

```
! Package amsmath Error: \begin{split} won't work here.
 ...

l.8 \begin{split}

? h
\Did you forget a preceding \begin{equation}?
If not, perhaps the 'aligned' environment is what you want.
?
```

Explanation: The `split` environment does not construct a stand-alone displayed equation; it needs to be used within some other environment such as `equation` or `gather`.

`Extra & on this line`
Example:

```
! Package amsmath Error: Extra & on this line.

See the amsmath package documentation for explanation.
Type  H <return>  for immediate help.
 ...
```

```
l.9 \end{alignat}

? h
\An extra & here is so disastrous that you should probably exit
 and fix things up.
?
```

Explanation: In an `alignat` structure the number of alignment points per line is dictated by the numeric argument given after `\begin{alignat}`. If you use more alignment points in a line it is assumed that you accidentally left out a newline command `\\` and the above error is issued.

Improper argument for math accent

Example:

```
! Package amsmath Error: Improper argument for math accent:
(amsmath)                Extra braces must be added to
(amsmath)                prevent wrong output.

See the amsmath package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.415 \tilde k_{\lambda_j} = P_{\tilde \mathcal
                                                {M}}
?
```

Explanation: Non-simple arguments for any LaTeX command should be enclosed in braces. In this example extra braces are needed as follows:

```
... P_{\tilde{\mathcal{M}}}
```

Font OMX/cmex/m/n/7=cmex7 not loadable ...

Example:

```
! Font OMX/cmex/m/n/7=cmex7 not loadable: Metric (TFM) file not found.
<to be read again>
                   relax
l.8 $a
      b+b^2$
? h
I wasn't able to read the size data for this font,
so I will ignore the font specification.
[Wizards can fix TFM files using TFtoPL/PLtoTF.]
```

```
You might try inserting a different font spec;
e.g., type `I\font<same font id>=<substitute font name>'.
?
```

Explanation: Certain extra sizes of some Computer Modern fonts that were formerly available mainly through the AMSFonts distribution are considered part of standard LaTeX (as of June 1994): `cmex7`–`9`, `cmmib5`–`9`, and `cmbsy5`–`9`. If these extra sizes are missing on your system, you should try first to get them from the source where you obtained LaTeX. If that fails, you could try getting the fonts from CTAN (e.g., in the form of Metafont source files, directory `/tex-archive/fonts/latex/mf`, or in PostScript Type 1 format, directory `/tex-archive/fonts/cm/ps-type1/bakoma`).

If the font name begins with `cmex`, there is a special option `cmex10` for the `amsmath` package that provides a temporary workaround. I.e., change the `\usepackage` to

```
\usepackage[cmex10]{amsmath}
```

This will force the use of the 10-point size of the `cmex` font in all cases. Depending on the contents of your document this may be adequate.

`Math formula deleted: Insufficient extension fonts`

Example:

```
! Math formula deleted: Insufficient extension fonts.
l.8 $ab+b^2$

?
```

Explanation: This usually follows a previous error `Font ... not loadable`; see the discussion of that error (above) for solutions.

`Missing number, treated as zero`

Example:

```
! Missing number, treated as zero.
<to be read again>
                   a
l.100 \end{alignat}

? h
A number should have been here; I inserted `0'.
(If you can't figure out why I needed to see a number,
look up `weird error' in the index to The TeXbook.)

?
```

Explanation: There are many possibilities that can lead to this error. However, one possibility that is relevant for the `amsmath` package is that you forgot to give the number argument of an `alignat` environment, as in:

```
\begin{alignat}
 a&  =b&    c& =d\\
a'& =b'&   c'& =d'
\end{alignat}
```

where the first line should read instead

```
\begin{alignat}{2}
```

Another possibility is that you have a left bracket character `[` following a linebreak command `\\` in a multiline construction such as `array`, `tabular`, or `eqnarray`. This will be interpreted by LaTeX as the beginning of an 'additional vertical space' request [7, §C.1.6], even if it occurs on the following line and is intended to be part of the contents. For example

```
\begin{array}
a+b\\
[f,g]\\
m+n
\end{array}
```

To prevent the error message in such a case, you can add braces as discussed in the LaTeX manual [7, §C.1.1]:

```
\begin{array}
a+b\\
{[f,g]}\\
m+n
\end{array}
```

`Missing \right. inserted`

Example:

```
! Missing \right. inserted.
<inserted text>
                \right .
l.10 \end{multline}

? h
I've inserted something that you may have forgotten.
(See the <inserted text> above.)
With luck, this will get me unwedged. But if you
really didn't forget anything, try typing '2' now; then
```

```
my insertion and my current dilemma will both disappear.
```

Explanation: This error typically arises when you try to insert a linebreak inside a `\left`-`\right` pair of delimiters in a `multline` or `split` environment:

```
\begin{multline}
AAA\left(BBB\\
  CCC\right)
\end{multline}
```

There are two possible solutions: (1) instead of using `\left` and `\right`, use 'big' delimiters of fixed size (`\bigl \bigr \biggl \biggr` ...; see §5.14.1); or (2) use null delimiters to break up the `\left`-`\right` pair into parts for each line:

```
AAA\left(BBB\right.\\
  \left.CCC\right)
```

The latter solution may result in mismatched delimiter sizes; ensuring that they match requires using `\vphantom` in the line that has the smaller delimiter (or possibly `\smash` in the line that has the larger delimiter). In the argument of `\vphantom` put a copy of the tallest element that occurs in the other line, e.g.,

```
xxx \left(\int_t yyy\right.\\
  \left.\vphantom{\int_t} zzz ... \right)
```

```
Paragraph ended before \xxx was complete
```

Example:

```
Runaway argument?

! Paragraph ended before \multline was complete.
<to be read again>
                   \par
l.100

? h
I suspect you've forgotten a `}', causing me to apply this
control sequence to too much text. How can we recover?
My plan is to forget the whole thing and hope for the best.
?
```

Explanation: This might be produced by a misspelling in the `\end{multline}` command, e.g.,

```
\begin{multline}
...
\end{multiline}
```

or by using abbreviations for certain environments, such as `\bal` and `\eal` for `\begin{align}` and `\end{align}`:

```
\bal
...
\eal
```

For technical reasons that kind of abbreviation does not work with the more complex displayed equation environments of the `amsmath` package (`gather`, `align`, `split`, etc.; cf. `technote.tex`).

Runaway argument?

See the discussion for the error message `Paragraph ended before \xxx was complete`.

Unknown option ‘xxx’ for package ‘yyy’

Example:

```
! LaTeX Error: Unknown option ‘intlim’ for package ‘amsmath’.
...
? h
The option ‘intlim’ was not declared in package ‘amsmath’, perhaps you
misspelled its name. Try typing  <return>  to proceed.
?
```

Explanation: This means that you misspelled the option name, or the package simply does not have an option that you expected it to have. Consult the documentation for the given package.

Old form ‘\pmatrix’ should be \begin{pmatrix}.

Example:

```
! Package amsmath Error: Old form ‘\pmatrix’ should be
                         \begin{pmatrix}.

See the amsmath package documentation for explanation.
Type  H <return>  for immediate help.
 ...

\pmatrix ->\left (\matrix@check \pmatrix
                                        \env@matrix
l.16 \pmatrix
```

```
             {a&b\cr c&d\cr}
? h
‘\pmatrix{...}’ is old Plain-TeX syntax whose use is
ill-advised in LaTeX.
?
```

Explanation: When the `amsmath` package is used, the old forms of `\pmatrix`, `\matrix`, and `\cases` cannot be used any longer because of naming conflicts. Their syntax did not conform with standard LaTeX syntax in any case.

`Erroneous nesting of equation structures`
Example:

```
! Package amsmath Error: Erroneous nesting of equation structures;
(amsmath)                trying to recover with ‘aligned’.

See the amsmath package documentation for explanation.
Type  H <return>  for immediate help.
 ...

l.260 \end{alignat*}
                     \end{equation*}
```

Explanation: The structures `align`, `alignat`, etc., are designed for top-level use and for the most part cannot be nested inside some other displayed equation structure. The chief exception is that `align` and most of its variants can be used inside the `gather` environment.

11.3 Warning messages

`Foreign command \over [or \atop or \above]`
Example:

```
Package amsmath Warning: Foreign command \over; \frac or \genfrac
(amsmath)                should be used instead.
```

Explanation: The primitive generalized fraction commands of TeX—`\over`, `\atop`, `\above`—are deprecated when the `amsmath` package is used because their syntax is foreign to LaTeX and `amsmath` provides native LaTeX equivalents. See `technote.tex` for further information.

`Cannot use ‘split’ here`
Example:

```
Package amsmath Warning: Cannot use ‘split’ here;
(amsmath)                trying to recover with ‘aligned’
```

Explanation: The `split` environment is designed to serve as the entire body of an equation, or an entire line of an `align` or `gather` environment. There cannot be any printed material before or after it within the same enclosing structure:

```
\begin{equation}
\left\{ % <-- Not allowed
\begin{split}
...
\end{split}
\right. % <-- Not allowed
\end{equation}
```

11.4 Wrong output

11.4.1 Section numbers 0.1, 5.1, 8.1 instead of 1, 2, 3

This most likely means that you have the arguments for `\numberwithin` in reverse order:

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

That means 'print the section number as *equation number.section number* and reset to 1 every time an equation occurs' when what you probably wanted was the inverse

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

11.4.2 The `\numberwithin` command had no effect on equation numbers

Are you looking at the first section in your document? Check the section numbers elsewhere to see if the problem is the one described in §11.4.1.

Tài liệu tham khảo

[1] *AMSFonts version* 2.2*—user's guide*, Amer. Math. Soc., Providence, RI, 1994; distributed with the AMSFonts package.

[2] *Instructions for preparation of papers and monographs—*$\mathcal{AMS}$*-LaTeX*, Amer. Math. Soc., Providence, RI, 1996, 1999.

[3] *Using the* `amsthm` *Package*, Amer. Math. Soc., Providence, RI, 1999.

[4] Michel Goossens, Frank Mittelbach, and Alexander Samarin, *The LaTeX companion*, Addison-Wesley, Reading, MA, 1994. [*Note: The 1994 edition is not a reliable guide for the* `amsmath` *package unless you refer to the errata for Chapter 8—file* `compan.err`*, distributed with LaTeX.*]

[5] G. Grätzer, *Math into LaTeX: An Introduction to LaTeX and AMS-LaTeX* http://www.ams.org/cgi-bin/bookstore/bookpromo?fn=91&arg1=bookvideo&itmc=MLTEX, Birkhäuser, Boston, 1995.

[6] Donald E. Knuth, *The TeXbook*, Addison-Wesley, Reading, MA, 1984.

[7] Leslie Lamport, *LaTeX: A document preparation system*, 2nd revised ed., Addison-Wesley, Reading, MA, 1994.

[8] Frank Mittelbach and Rainer Schöpf, *The new font family selection—user interface to standard LaTeX*, *TUGboat* **11**, no. 2 (June 1990), pp. 297–305.

[9] Michael Spivak, *The joy of TeX*, 2nd revised ed., Amer. Math. Soc., Providence, RI, 1990.