

amsmath パッケージユーザガイド (Version 2.1)

アメリカ数学会, L^AT_EX3 Project

1999-12-13 (改訂 2002-02-25, 2016-11-14, 2018-04-05)

この原語（英語）ファイルは L^AT_EX Project が管理しています.
英語版のバグリポートは, 次にお送りください (category は `amslatex`) です.
<http://latex-project.org/bugs.html>.

Contents

1	イントロダクション	1
2	amsmath パッケージのオプション	2
3	ディスプレイ方程式	3
3.1	イントロダクション	3
3.2	一つだけの数式	4
3.3	式を位置揃えしないで分割	4
3.4	式を位置揃えて分割	6
3.5	位置揃えなしの数式の集まり	6
3.6	互いに揃える数式の集まり	7
3.7	構成要素の位置揃え	8
3.8	タグの位置を揃える	9
3.9	複数行のディスプレイ数式での縦方向の空白と改ページ	9
3.10	ディスプレイ数式への割り込み	10
3.11	数式番号	11
4	そのほかの数学に現れる構造	12
4.1	行列	12
4.2	数式の空白を指定するコマンド	13
4.3	ドット	13
4.4	分割されないダッシュ	14
4.5	数式でのアクセント	14
4.6	根号	14
4.7	枠で囲まれた数式	15
4.8	上につく矢印, 下につく矢印	15
4.9	長さが伸びる矢印	15
4.10	記号を別の記号に加える	15
4.11	分数, それに関連する構成	15
4.12	連分数	16
4.13	スマッシュオプション	17
4.14	デリミタ	17
5	オペレータの名前	18
5.1	新しいオペレータの定義	19
5.2	\mod とそのなかま	20
6	\text コマンド	20

7	積分と総和記号	20
7.1	複数の行で作られる下付き上付き添え字	21
7.2	<code>\sideset</code> コマンド	21
7.3	添え字と上限下限の配置	22
7.4	多重積分の記号	22
8	可換図式	22
9	数学フォントを使う	23
9.1	イントロダクション	23
9.2	数学フォントの勧められる使い方	23
9.3	太字の数学記号	24
9.4	イタリックのギリシア文字	25
10	エラーメッセージと出力の問題	25
10.1	一般的な注意	25
10.2	エラーメッセージ	25
10.3	警告メッセージ	31
10.4	正しくない出力	32
11	さらなる情報	32
11.1	他のパッケージとの互換性	32
11.2	既存のドキュメントを利用する	33
11.3	技術的注意	33
11.4	助けが必要な時	33
11.5	関心のある方へ	33
	参考文献	35
	Index	36

—1—

イントロダクション

`amsmath` は \LaTeX のパッケージの一つで、数式を含むドキュメントの情報構造と印刷結果を改善するためのさまざまな機能拡張を提供します。読者の方で \LaTeX に慣れていないのなら、[3] を読んでください。あなたが \LaTeX 最新のバージョンを持っているのなら、ここで説明する `amsmath` は、標準として備わっています。新しいバージョンの `amsmath` が公開されたら <http://mirror.ctan.org/macros/latex/required/amsmath.zip> から入手できます。

今お読みのドキュメントは `amsmath` パッケージの機能と、その使い方を解説します。その他に次のパッケージについても扱います：

```
amsbsy  amsopn  amsxtra
amscd   amstext
```

これはらすべて数式を含んだ文書の作成に関係します。数学記号と数学フォントについては、[8] および <https://www.ams.org/tex/amsfonts.html> で扱われています。`amsthm` パッケージあるいは AMS ドキュメントクラス (`amsart`, `amsbook` など) については [9] あるいは [7]、および <https://www.ams.org/tex/author-info.html> をご覧ください。

これまでに \LaTeX を使っている方で、たくさん数式を書いている方ならば、こうしたいと思ったことが `amsmath` の機能の一覧にあるのがわかるでしょう：

- `\sin` および `\lim` のように、作用素の両側には適切な空白（スペース、アキ）を確保して、そして適切なフォントスタイル大きさが（上下付き文字または上付き文字で使われる場合でも）自動的に選ばれる「作用素（演算子）名」コマンドを定義する便利な方法があります。
- `eqnarray` 環境のかわりになる、さまざまな種類の方程式を簡単に配置するための機能があります。
- 数式番号は、式と重複しないように自動的に式の上あるいは下へ調整されます（`eqnarray` では、そうではありません）。
- 等号の前後の空白は、`equation` 環境の通常の空白と一致します（`eqnarray` では、そうではありません）。
- 総和または総乗記号の範囲を示す数式を複数行でも可能にします。
- 式番号を自動的に振るわかりに目的に合わせて簡単に変更できます。
- 選択された方程式の集まりに、(1.3a) (1.3b) (1.3c) のような副番号をつけることが簡単にできます。

`amsmath` パッケージはいくつかの小さな補助パッケージとともに配布されています：

`amsmath` これが基本となるパッケージで、数式やその他の数学的な構造を表現する様々な機能を与えます。

`amstext` ディスプレイ数式内に短いテキストをタイプセットする `\text` コマンドを与えます。

`amsopn` `\sin` や `\lim` のような「作用素名」を定義するコマンド `\DeclareMathOperator` を与えます。

`ambsy` 以前のバージョンとの互換性のためのものです。したがって、これからは \LaTeX と一緒に配布される `bm` を使ってください。

`amscd` 単純な可換図式のための CD 環境を提供しています。(斜め矢印はサポートしていません)。

`amxtra` `\fracwithdelims` と `\accentedsymbol` のようなバージョン 1.1 で作成されたドキュメントの互換性のための、雑多なコマンドを集めたものです。

`amsmath` パッケージは、`amstext`、`amsopn`、そして `ambsy` が組み込まれています。ただし、`amscd` と `amxtra` の機能は、これらのパッケージを個別に呼び出さなければなりません。

独立のパッケージ `mathtools`[10] は、`amsmath` のいくつかの機能を拡張します；`mathtools` は `amsmath` を自動的にロードするので、個別に `amsmath` をロードする必要はありません。いくつかの `mathtools` の機能については、このあと必要な時に説明します。

—2—

`amsmath` パッケージのオプション

`amsmath` パッケージには、次のオプションがあります：

`centertags` (デフォルト) 方程式が `split` 環境の場合、式番号を、それらの方程式を合わせた高さの中央に配置します。

`tbtags` ‘Top-or-bottom tags’: 方程式が `split` 環境の場合、式番号を右に置く時は最後の式の行に、式番号を左に置くときは最初の式の行に配置します。

`sumlimits` (デフォルト) ディスプレイ数式 (別行立て数式) の場合、総和記号の範囲 (limits) を示す上付き下付きの数式は、記号の上下に配置します。このオプションは、 \prod , \coprod , \otimes , \oplus などの他の記号にも影響しますが、積分記号には当てはまりません (下記参照)。

`nosumlimits` ディスプレイ数式でも、総和記号の (和の範囲を示す) 下付き文字と上付き数式を記号の右側に配置します。

`intlimits` `sumlimits` 記号と同じような処理を積分記号でも行ないます。

`nointlimits` (デフォルト) `intlimits` とは反対の処理をします。

`namelimits` (デフォルト) `sumlimits` と同じように、`det`, `inf`, `lim`, `max`, `min` のような特定の「作用素名」に対しては、ディスプレイ数式では下付き文字は伝統的に記号の真下に配置されます。

`nonamelimits` `namelimits` とは反対の処理をします。

`alignedleftspaceyes`

`alignedleftspaceno`

`alignedleftspaceyesifneg`

これらのパッケージオプションのいずれかを使用するには、`\usepackage` コマンドのオプション引数を、たとえば `\usepackage[intlimits]{amsmath}` のようにして、オプション名を入れます。AMS ドキュメントクラスと、`amsmath` がロー

ドされている他のクラスでは、必要なオプションは`\documentclass`の際に`\documentclass[intlimits,tbtags,reqno]{amsart}`のように特定します。

`amsmath` パッケージは、`\documentclass` コマンドで（暗黙的または明示的に）通常は選択されている、次に示すオプションを認識しています。したがって`\usepackage{amsmath}`ステートメントでオプションを繰り返して指定する必要はありません。

`leqno` 数式番号を左に置きます。

`reqno` 数式番号を右に置きます。

`fleqn` 数式の配置を文章幅の左右中央ではなくて、左から一定の字下げの後で行います。

`aligned` と `gather` 環境の左に空白を制御するための3つのオプションが追加されました。2017年以前には、これらの構造の左側には空白が追加されていましたが、右側には追加されていませんでした。これは実装したさいの偶発的な機能であり、通常、環境の先頭に`\!`を付けることで修正されました。

新しいデフォルトの振る舞いは、ほとんどの場合、環境に細い空白が追加されないようにすることを目的としています。既存の文章に対しては、`\!\begin{aligned}`とすることで、前と同じように結果になります。

`alignedleftspaceyes` つねに`\,`を `aligned` と `gathered` の左に入れる。

`alignedleftspaceno` `\,`を `aligned` と `gathered` の左に入れない。

`alignedleftspaceyesifneg` 環境が負の空白で調整されているときに限り`\,`を入れる。（新しい振る舞い。）

—3—

ディスプレイ方程式

3.1 イントロダクション

`amsmath` パッケージには、基本の`LATEX`にはないディスプレイ方程式（別行立て数式）の構造が、いくつも追加されています。それらは次のものです：

<code>equation</code>	<code>equation*</code>	<code>align</code>	<code>align*</code>
<code>gather</code>	<code>gather*</code>	<code>alignat</code>	<code>alignat*</code>
<code>multline</code>	<code>multline*</code>	<code>flalign</code>	<code>flalign*</code>
<code>split</code>			

（もちろん標準の `eqnarray` 環境も残っていますが、それを使わずに `align` と `equation+split` を使ほうが良い結果をえます。）`eqnarray` では、等号の両側の空白が数式にふさわしくない幅になっている上に、他の環境との整合性がありません。この環境では、長い数式は式番号がふさわしくない位置に置かれたり、式と重なることがあります。この環境は、その上、定理環境で提供されている`\qed`あるいは`\qedhere`をサポートしていません。

`split`を除いて、これらの環境は星印ありと星印なしがあります。星印なしは`LATEX`の`equation`カウンタを使って自動的に数式番号を割り当てます。数式番号や特定の行は、その行の終わりに`\notag`を置くことで表示させないようにで

きます。さらに`\tag{label>}`を使えば、式番号を上書きできます。ここで`label>`は式番号“number”に使われる`$$`や`ii`などの任意のテキストです。タグは参照形式`\tag{\ref{label>}modifier>}`を使えば、それぞれで参照できます。ここで`modifier>`はオプションです。`hyperref`を使うときは、`\ref*`を使います；星印ありの`\ref`を使うと、元の表示にリンクされているネストされたリンクを含む変更されたタグへの参照を防ぎます。

There is also a `\tag*` コマンドは、丸括弧に囲われない、そのままのテキストの表示に使うことができます。`\tag` と `\tag*` は、`amsmath` のすべての整列構造の中で、番号なしのバージョンで使うことができます。`\tag` の使い方の例は、`amsmath` パッケージとともに提供されているサンプルファイルの `testmath.tex` と `subeqn.tex` に与えられています。

`split` 環境は特別な下部形式で、他のものの中でのみで使用されます。ただし、`multline` の中では使用できません。`split` は一つだけの整列 (&) コラムだけをサポートしています；さらに必要な場合は `aligned` あるいは `alignedat` を使うべきです。`split` 構造の幅は、1 行の長さいっぱい（全幅）です。

整列を行う構造 (`split`, `align` と、その他の類似物) では、`eqnarray` とは異なり、揃えたい記号の前に `&` を置きます。後ではありません。関係記号の後に `&` を置くと、通常の空白取りに干渉します；つまり前に置かなければなりません。

すべての複数行の環境で、行は `\\` で分けられます。`\\` を最後の行で使ってはいいません。そうしてしまうと、意図しない垂直方向の空白が生じます。

すべての数学環境（インラインでもディスプレイでも）空白の行（`\par` と同値）は許されていないので、エラーとなります。

3.2 一つだけの数式

`equation` 環境は、自動的に数式番号が生成される単一の方程式に対するものです。`equation*` 環境は、数式番号を表示しないこと以外は同じです。¹`[...]` は、`equation*` と同値です。

3.3 式を位置揃えしないで分割

`multline` 環境は、1 行に収まらない方程式に使用される `equation` 環境のバリエーションです。`multline` の最初の行は、両側のインデントが `\multlinegap` で与えられることを除いて、数式の始まりは左マージンから、数式の終わりの行は右マージンになります。その間の追加の行は、表示幅内で独立して中央揃えされます（`fleqn` オプションが有効な場合を除く）。

`equation` と同様に、`multline` には 1 つの式番号を持ちます。（個々の行には `\notag` と書く必要がありません。）式番号は最後の行（`reqno` オプション）または最初の行（`leqno` オプション）に置かれます。`split` の場合のような上限中央への配置は `multline` ではサポートされていません。

`\shoveleft`, `\shoveright` のコマンドを使えば、中央の行の 1 つを左または右に動かすことができます。これらのコマンドは行全体を引数としてとります。

¹ `LATEX` は基本的には `equation*` 環境を提供していませんが、`displaymath` という名前の機能が同等の環境を与えています。

Table 3.1. ディスプレイ方程式環境の比較（縦線は通常のマージンを示す）

<code>\begin{equation*}</code> <code>a=b</code> <code>\end{equation*}</code>		$a = b$	
<code>\begin{equation}</code> <code>a=b</code> <code>\end{equation}</code>		(1) $a = b$	
<code>\begin{equation}\label{xx}</code> <code>\begin{split}</code> <code>a&=b+c-d\\</code> <code>&\quad +e-f\\</code> <code>&=g+h\\</code> <code>&=i</code> <code>\end{split}</code> <code>\end{equation}</code>		(2) $\begin{aligned} a &= b + c - d \\ &\quad + e - f \\ &= g + h \\ &= i \end{aligned}$	
<code>\begin{multline}</code> <code>a+b+c+d+e+f\\</code> <code>+i+j+k+l+m+n</code> <code>\end{multline}</code>		(3) $\begin{aligned} a + b + c + d + e + f \\ + i + j + k + l + m + n \end{aligned}$	
<code>\begin{gather}</code> <code>a_1=b_1+c_1\\</code> <code>a_2=b_2+c_2-d_2+e_2</code> <code>\end{gather}</code>		(4) $a_1 = b_1 + c_1$ (5) $a_2 = b_2 + c_2 - d_2 + e_2$	
<code>\begin{align}</code> <code>a_1&=b_1+c_1\\</code> <code>a_2&=b_2+c_2-d_2+e_2</code> <code>\end{align}</code>		(6) $a_1 = b_1 + c_1$ (7) $a_2 = b_2 + c_2 - d_2 + e_2$	
<code>\begin{align}</code> <code>a_{11}&=b_{11}&</code> <code>a_{12}&=b_{12}\\</code> <code>a_{21}&=b_{21}&</code> <code>a_{22}&=b_{22}+c_{22}</code> <code>\end{align}</code>		(8) $a_{11} = b_{11}$ $a_{12} = b_{12}$ (9) $a_{21} = b_{21}$ $a_{22} = b_{22} + c_{22}$	
<code>\begin{flalign*}</code> <code>a_{11} + b_{11}&= c_{11}&</code> <code>a_{12}&=b_{12}\\</code> <code>b_{21}&= c_{21}&</code> <code>a_{22}&=b_{22}+c_{22}</code> <code>\end{flalign*}</code>		$\begin{array}{ll} a_{11} + b_{11} = c_{11} & a_{12} = b_{12} \\ b_{21} = c_{21} & a_{22} = b_{22} + c_{22} \end{array}$	

しかし改行の\\は含みません．たとえば

$$(3.10) \quad \begin{array}{c} \boxed{A} \\ \boxed{B} \\ \boxed{C} \\ \boxed{D} \end{array}$$

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

`\multlinegap` の値は \LaTeX のコマンド `\setlength` あるいは `\addtolength` で変更することができます．

3.4 式を位置揃えて分割

`multline` と同様に，`split` 環境は，1 行に収まらない長すぎる一つの数式ために複数の行に分割する必要があるときに使います．しかし `split` 環境は `multline` とは異なり，`&` を使用して整列箇所に印を付けて，数式を分割する位置を合わせます．他の `amsmath` 方程式の構造とは異なり，`split` 環境は他のディスプレイ方程式構造の内部でのみ使用されるため，数式番号はつきません．通常，`equation`，`align`，または `gather` 環境では，番号付けが行われます．たとえば：

$$(3.11) \quad H_c = \frac{1}{2n} \sum_{l=0}^n (-1)^l (n-l)^{p-2} \sum_{l_1+\dots+l_p=l} \prod_{i=1}^p \binom{n_i}{l_i} \cdot [(n-l) - (n_i - l_i)]^{n_i - l_i} \cdot \left[(n-l)^2 - \sum_{j=1}^p (n_i - l_i)^2 \right].$$

```
\begin{equation}\label{e:barwq}\begin{split}
H_c&=\frac{1}{2n} \sum_{l=0}^n (-1)^l (n-l)^{p-2} \\
&\sum_{l_1+\dots+l_p=l} \prod_{i=1}^p \binom{n_i}{l_i} \\
&\quad \cdot [(n-l) - (n_i - l_i)]^{n_i - l_i} \cdot \left[ (n-l)^2 - \sum_{j=1}^p (n_i - l_i)^2 \right].
\end{split}\end{equation}
```

`split` 構造は，`\label` のような目に見えないものを生成するコマンドを除いて，全体を構成する必要があります．

3.5 位置揃えなしの数式の集まり

`gather` 環境は，一連の連立方程式の中で位置揃えが望ましくない場合に使用されます．それぞれがテキスト幅内で別々に中央揃えされます（表 3.1 参照）．`gather` 内の方程式は，`\\` コマンドで区切られています．`gather` の中のどの方程式も，`\begin{split} ... \end{split}` 構造などで構成されます．たとえば：

```

\begin{gather}
  \text{first equation}\\
  \begin{split}
    \text{second \& equation}\\
    \text{\& on two lines}
  \end{split}
  \\
  \text{third equation}
\end{gather}

```

3.6 互いに揃える数式の集まり

`align` 環境は、縦方向に揃えたい 2 つ以上の式に使用されます。通常は等号などの 2 項関係で揃えます (表 3.1 参照)。

複数の方程式列を並べて使用するには、必要な個数のアンパサンドを使用して列を区切ります：

$$\begin{array}{llll}
 (3.12) & x = y & X = Y & a = b + c \\
 (3.13) & x' = y' & X' = Y' & a' = b \\
 (3.14) & x + x' = y + y' & X + X' = Y + Y' & a'b = c'b
 \end{array}$$

```

\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}

```

`align` 環境の中の数式に行単位で注釈をつけるときには、その都度 `\text` を使います：

$$\begin{array}{ll}
 (3.15) & x = y_1 - y_2 + y_3 - y_5 + y_8 - \dots & \text{by (3.24)} \\
 (3.16) & = y' \circ y^* & \text{by (4.1)} \\
 (3.17) & = y(0)y' & \text{by Axiom 1.}
 \end{array}$$

```

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

```

この変種環境 `alignat` は、方程式間の水平の空白を明示的に指定できます。この環境は一つの引数を取りますが、これは“数式の列数” (整列された右左の一对の個数；ペアの個数) の数を指定します。列の中の `&` の最大の個数を数えて、それに 1 を足して 2 で割ります。

$$\begin{aligned}
 (3.18) \quad & x = y_1 - y_2 + y_3 - y_5 + y_8 - \dots \quad \text{by (3.24)} \\
 (3.19) \quad & = y' \circ y^* \quad \text{by (4.1)} \\
 (3.20) \quad & = y(0)y' \quad \text{by Axiom 1.}
 \end{aligned}$$

```

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

```

環境 `flalign` (“整列する最大の長さ”) は、数式の間空白を最大の幅まで引き延ばし、数式番号の間に可能ならば十分な空白を入れます。

$$\begin{aligned}
 (3.21) \quad & x = y & X = Y \\
 (3.22) \quad & x' = y' & X' = Y' \\
 (3.23) \quad & x + x' = y + y' & X + X' = Y + Y'
 \end{aligned}$$

```

\begin{flalign}
x&=y && X&=Y\backslash\backslash \\
x'&=y' && X'&=Y'\backslash\backslash \\
x+x'&=y+y' && X+X'&=Y+Y'
\end{flalign}

```

$$\begin{aligned}
 & x = y & X = Y \\
 & x' = y' & X' = Y' \\
 & x + x' = y + y' & X + X' = Y + Y'
 \end{aligned}$$

```

\begin{flalign*}
x&=y && X&=Y\backslash\backslash \\
x'&=y' && X'&=Y'\backslash\backslash \\
x+x'&=y+y' && X+X'&=Y+Y'
\end{flalign*}

```

3.7 構成要素の位置揃え

`equation` と同じ様に、複数の方程式環境 `gather`, `align`, `alignat` は、ページ幅全体にわたった構造を生成するように設計されています。これは、たとえば、構造全体にかっこを簡単に追加できないことを意味します。しかし、変種である `gathered`, `aligned`, および `alignedat` は、実際に生成した構造の実際の幅を与えます。したがって、含まれる数式を一つの要素として使用することができます。つまり、

$$\left. \begin{aligned} B' &= -\partial \times E, \\ E' &= \partial \times B - 4\pi j, \end{aligned} \right\} \quad \text{マクスウェルの方程式}$$

```

\begin{equation*}
\left.\begin{aligned}
B' &= -\partial \times E, \\
E' &= \partial \times B - 4\pi j,
\end{aligned}\right\}
\\[10pt]
\text{Maxwell's equations}
\end{equation*}

```

array 環境と同様に、これらの`-ed`変種は、オプション引数 `[t]`、`[b]` あるいはデフォルトの `[c]` を使用して、縦方向の位置を指定します。互換性を最大限にするために、オプションの前に空白あるいは改行を入れないようにします。また?? ページの`-ed`環境における改ページの説明を見てください。

“場合わけ”は、数学ではよく現れます：

$$(3.24) \quad P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r!(-1)^{(r-j)/2} & \text{if } r-j \text{ is even.} \end{cases}$$

`amsmath` パッケージには `cases` 環境があり、場合分けを簡単に書くことができます：

```

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}

```

`\text{cf. §6}` が使われていて、数式がその中にあることに注意してください。`cases` は `\textstyle` に置かれています。`\displaystyle` が必要なら、明示します；そのために `mathtools` には `dcases` 環境が用意されています。

`-ed` と `cases` 環境は、数式環境、テキストや `$...$` の間、あるいは任意のディスプレイ環境の中になければなりません。

3.8 タグの位置を揃える

複数行のディスプレイ数式では、式番号をどこに置くかということは、かなり複雑な問題になります。`amsmath` パッケージの環境では、式の内容に式番号を重複して印刷しないようにしています。必要であれば、式番号は上か下の別の行に移動してください。方程式の長さを正確に計算することが困難な場合は、式番号が正しく表示されないことがあります。`\raisetag` コマンドは、現在の方程式の番号を通常的位置から垂直方向にずらすために用意されています。6 ポイント上に移動するには、`\raisetag{6pt}` と書いてください。(ディスプレイの終わりで、ディスプレイに続くテキストを持ち上げます。) この種の改行や改ページの調整は、ドキュメントが完成したと言えるまでは行わないようにしてください。そうでなければ、ドキュメントの内容を変えるたびに微調整をやり直すことになります。

3.9 複数行のディスプレイ数式での縦方向の空白と改ページ

`\[dimension>]` コマンドを使えば、通常の \LaTeX のように、`amsmath` のディスプレイ数式の行間に余分な垂直方向のスペースを入れることができます。`\[` と次

の [の間にスペースを入れません ; `amsmath` で定義されているディスプレイ環境だけがスペースを角カッコ (ブラケット) で囲まれて部分はドキュメントの一部であると解釈されます。

`amsmath` パッケージが使用されている場合、方程式の行間で改ページは通常は許可されません。ディスプレイ数式の中のある行で改ページすることは著者の責任であると考えます。特定の表示式の中で改ページを行うためには、`\displaybreak` コマンドが提供されています。`\displaybreak` は、`\` の直前に配置するのが最も効果的です。L^AT_EX の `\pagebreak` のように、`\displaybreak` は、改ページの望ましさを示す 0 から 4 の任意の引数を取ります。`\displaybreak[0]` は、ここでの改ページを勧めないが “必要なら改ページを認める” こととされます。オプション引数なしの `\displaybreak` は `\displaybreak[4]` 場合と同じ意味で、改ページを強制します。

改ページを、どの場所でも、複数行の式の途中で行ってもよいのなら、ドキュメントのプレアンブルに `\allowdisplaybreaks[1]` を置くことを勧めます。オプションの引数 1-4 は、細かい制御のために使用できます : [1] は改ページを許可するが、できるだけ避けたいことを意味します ; 2,3,4 の値は許容度を増加させることを意味します。`\allowdisplaybreaks` で改ページを有効にしておいても、いつものように、`\`* コマンドはある行の後で改ページを禁止させることができます。

注意 : `labelbreaknote` ある種の方程式環境は、内容を分割できないボックスで囲むので、結果的に `\displaybreak` や `\allowdisplaybreaks` の影響を受けません。それらには `split`, `aligned`, `gathered`, `alignedat` などがあります。

3.10 ディスプレ数式への割り込み

コマンド `\intertext` は、複数行のディスプレイ数式の途中で、1 行または 2 行のテキストを挿入するために使用されます (§6 の中の `\text` コマンドも参照)。その顕著な特徴は一揃えが保存されていることです。一度ディスプレイ数式を終えて、テキストを挟み新しいディスプレイ数式を使用した場合は、こうなりませ。`\intertext` は、`\` や `\`* コマンドの直後にだけに現れます。この例では、単語 “and” の位置を確認してください。

$$(3.25) \quad A_1 = N_0(\lambda; \Omega') - \phi(\lambda; \Omega'),$$

$$(3.26) \quad A_2 = \phi(\lambda; \Omega') - \phi(\lambda; \Omega),$$

and

$$(3.27) \quad A_3 = \mathcal{N}(\lambda; \omega).$$

```
\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}
```

`mathtools` パッケージは、短いテキストを挿入するためのコマンド `\shortintertext` を与えています ; これは `\intertext` よりも少し狭い垂直のスペースを用います。これが便利な場合は、数式番号が右に表示される時です。

3.11 数式番号

3.11.1 数式番号の階層構造

L^AT_EX では、数式番号を 1 節なら (1.1), (1.2), ..., 2 節なら (2.1), (2.2), ..., のように節ごとに開始したいなら、L^AT_EX のマニュアル [3, §6.3, §C.8.4] にあるように `\theequation` を再定義して使います:

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

これで良いのですが、数式番号のカウントは、`\setcounter` で自分で指定しない限り新しい節や章の開始がゼロにリセットされません。これをもう少し簡単に行うために `amsmath` パッケージには `\numberwithin` コマンドがあります。数式番号を節の番号に結びつけて、自動的にカウンタをリセットしたいのなら、

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

と書きます。その名前が意味するように、`\numberwithin` コマンドは、`equation` だけでなくどのカウンタにも働きます。

3.11.2 相互参照と数式番号

相互参照の作成は簡単にできます。`\eqref` コマンドをつかいます。これは、数式番号を自動的に生成して、その周りを丸カッコで囲みます。つまり `\ref{abc}` は番号 3.2 だけを生成しますが、`\eqref{abc}` は丸カッコつきの (3.2) を生成します。数式番号 `\eqref` の丸括弧は、文脈のスタイルに関係なく立体で表示されます。

3.11.3 数式番号の副番号の割り当て

`amsmath` パッケージには、数式のまとまりに対して副番号を生成するラッパー環境の `subequations` という機能もあります。`\align` など作られた数式のグループに副番号をつけるのに便利です。たとえば

```
\begin{subequations}
```

```
...
```

```
\end{subequations}
```

とすれば、この前の数式の番号が (4.8) であれば、このグループの数式番号は (4.9a) (4.9b) (4.9c) ..., となります。`\label` コマンドは `\begin{subequations}` の直後に置きます。そうすれば `\ref` が指すのは 4.9 であり 4.9a ではありません。副番号環境が使うカウンタは `parentequation` と `equation` および `\addtocounter`, `\setcounter`, `\value`, などであり、通常のカウントに使われます。副番号に小文字のアルファベット以外を使いたい場合は、通常の L^AT_EX の番号付の規則を使います [3, §6.3, §C.8.4]。たとえば、`\theequation` を再定義すれば、ローマ数字を使うことができます。

```
\begin{subequations}
```

```
\renewcommand{\theequation}{\theparentequation \roman{equation}}
```

```
...
```

3.11.4 番号付のスタイル

デフォルトの数式番号は `\normalfont` に設定されています。つまり節見出しがボールドでも、節の番号はボールドにならないことを意味しています；ボールドにしたければ、`\eqref{...}` ではなくて (`\ref{...}`) を使います。

同じような状況は、番号づけられたディスプレイの時に数式番号のサイズも小さくなることあげられます。ドキュメント全体でデフォルトのサイズを強制的に指示するためには、プレアンプルに次のように定義します：

```
\makeatletter
\renewcommand{\maketag@@@}[1]{\hbox{\m@th\normalsize\normalfont#1}}%
\makeatother
```

(この変更は将来のバージョンの `amsmath` に取り入れられます。)

—4—

そのほかの数学に現れる構造

4.1 行列

`amsmath` パッケージは、`LaTeX` の基本的な `array` 環境より高機能の、行列環境を提供します。`pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix`, `Vmatrix` はそれぞれ $()$, $[]$, $\{ \}$, $||$ および $|||$ デリミタ (丸カッコ, 縦棒, 各カッコなど) が組み込まれています。命名の一貫性のために、デリミタの無い (要素が整列されただけの) `matrix` もあります。これは、`array` 環境があるにも関わらず無駄に作ったわけではありません。行列環境はすべて、`array` 環境の間隔よりも、無駄のないように水平の空白を使用します。また、`array` 環境とは異なり、任意の行列環境に対して列指定を行う必要はありません；デフォルトでは、10 列まで設定できます。² (列の左あるいは右を揃えたいときや、ほかの整列基準を使いたい場合は、`array` あるいは `mathtools` パッケージで提供されているこれらの環境で左あるいは右の整列を指示するオプション引数を持つ*で行なってください。)

本文の中に小さい行列を入れるために `smallmatrix` 環境があります。(つまり $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$) とすれば、普通サイズの行列とは異なり 1 行に収まります。デリミタは与えなければなりません。(`mathtools` パッケージには `smallmatrix` 向けの `p`, `b`, `B`, `v`, `V` があります。上で説明したような*変種もあります。) 上の例は次のようにして作られました。

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

`\hdotsfor{number>}` は、行列の中に必要な列にまたがるドットだけの列を作ります。たとえば

$a \quad b \quad c \quad d$	<code>\begin{matrix} a&b&c&d\\</code>
$e \quad \dots\dots$	<code>e&\hdotsfor{3} \end{matrix}</code>

ドットとドットの間は、各カッコ `[]` オプションを使って、たとえば `\hdotsfor[1.5]{3}` のように指定できます。各カッコの中の数値は、ある数字を掛けるのか、1 の何

²正確には：行列の列の最大値はカウンタ `MaxMatrixCols` (通常は 10) できめられています。この値は `LaTeX` の `\setcounter` あるいは `\addtocounter` コマンドで増やすことができます。

倍かという形でしていします（つまり、通常の値は 1.0）。

$$(4.1) \quad \begin{pmatrix} D_1 t & -a_{12} t_2 & \dots & -a_{1n} t_n \\ -a_{21} t_1 & D_2 t & \dots & -a_{2n} t_n \\ \dots & \dots & \dots & \dots \\ -a_{n1} t_1 & -a_{n2} t_2 & \dots & D_n t \end{pmatrix},$$

```
\begin{pmatrix} D_1 t & -a_{12} t_2 & \dots & -a_{1n} t_n \\ -a_{21} t_1 & D_2 t & \dots & -a_{2n} t_n \\ \dots & \dots & \dots & \dots \\ -a_{n1} t_1 & -a_{n2} t_2 & \dots & D_n t \end{pmatrix}
```

4.2 数式の空白を指定するコマンド

amsmath パッケージは、次に示すように数式のための空白コマンドを少し拡張しています。省略なし（文字通り）のコマンドもコマンドも省略コマンドもロバストです。つまり数式の外でも使用できます。

省略形	省略なし	例	省略形	省略なし	例
	no space	$\Rightarrow\Leftarrow$		no space	$\Rightarrow\Leftarrow$
<code>\,</code>	<code>\thinspace</code>	$\Rightarrow\Leftarrow$	<code>\!</code>	<code>\negthinspace</code>	$\Rightarrow\Leftarrow$
<code>\:</code>	<code>\medspace</code>	$\Rightarrow\Leftarrow$		<code>\negmedspace</code>	$\Rightarrow\Leftarrow$
<code>\;</code>	<code>\thickspace</code>	$\Rightarrow\Leftarrow$		<code>\negthickspace</code>	$\Rightarrow\Leftarrow$
	<code>\quad</code>	$\Rightarrow\Leftarrow$			
	<code>\qquad</code>	$\Rightarrow\Leftarrow$			

数式の中の空白を調整するためのコマンドは `\mspace` と ‘数学単位 (math units)’ です。1 数学単位は `mu` と呼ばれ、値が 1/18 em です。つまり負の `\quad` が必要なら `\mspace{-18.0mu}` と書けばよいのです。

4.3 ドット

省略を示すドットの列（右上がり、一列など）は文脈によって異なり、共通した決まりはありません。好みの問題となることがあります。意味を持たせたコマンドが `\ldots` と `\cdots` 以外に用意されています。

- `\dotsc` は “コンマ付きのドット”
- `\dotsb` は “二項作用素や二項関係が続くドット”
- `\dotsm` は “複数のドット”
- `\dotsi` は “積分が続くドット”
- `\dotso` は “その他”（上のどれにも当てはまらない場合）

`\ldots` と `\cdots` 以外は、あなた自身の目的に応じて使い分けてください。ただし（たとえば）論文の投稿では発行元の規則があればそれに従います。デフォルトの処理は、次の例で示すアメリカ数学会 (American Mathematical Society) の規則に従っています：

Then we have the series A_1, A_2, \dots , the regional sum $A_1 + A_2 + \dots$, the orthogonal product $A_1 A_2 \dots$, and the infinite integral $\int_{A_1} \int_{A_2} \dots$.

多くの場合、ひとつながりの \dots が使われています。amsmath は、多くの場合状況に応じ適切な形を出力します；好ましくない結果を得たら、出力を調べて修正してください。

4.4 分割されないダッシュ

`\nbreakdash` コマンドは、ハイフンやダッシュの後に起こり得る改行（行の分割）を抑制します。たとえば、`pages 1\nbreakdash--9` と書けば、このダッシュと 9 との間で改行は起こりません。`\nbreakdash` を使っても、 p -adic の間などの望まないハイフネーションを防ぐことができます。たびたび使うなら、簡易に入力する方法を勧めます。つまり

```
\newcommand{\p}{\p\nbreakdash}% for "p-adic"
\newcommand{\Ndash}{\nbreakdash--}% for "pages 1\Ndash 9"
% For "n dimensional" ("n-dimensional"):
\newcommand{\n}[1]{\n\nbreakdash-\hspace{0pt}}
```

とします。最後の例が示している通り、ハイフンの後の改行を抑制されましたが、そのあとの単語には通常のハイフネーションは行っています。（ゼロ幅の空白をハイフンの後に入れています。）

4.5 数式でのアクセント

通常 L^AT_EX では、数式アクセントに 2 つ目を加えた時、2 つ目の配置は、よくありません。amsmath パッケージでは、2 つ目のアクセントをつけたときの配置が改善されています： $\hat{\hat{A}}$ (`\hat{\hat{A}}`).

コマンド `\dddot` と `\ddddot` は、L^AT_EX で用意されている `\dot` と `\ddot` に続いて 3 重と 4 重ドットを作ります。

上付きのハットとチルダを得るためには、amsxtra パッケージを読み込み、`\sphat` あるいは `\sptilde` を使います。使い方は `A\sphat` です。（上付きを指示する \wedge を用いないことに注意してください。）

数学アクセントとして任意の記号を配置したい場合、あるいは文字の下につけたい場合は、Javier Bezos が作成した `accents` を試してください。（amsmath は `pkgaccents` の前にロードしなければなりません。）

4.6 根号

通常の L^AT_EX では根号の指数の位置が、あまり良くないことがあります： $\sqrt[k]{\beta}$ (`\sqrt[\beta]{k}`). amsmath パッケージには `\leftroot` と `\uproot` があり、根号の位置を調整できます：

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

は、ベータを上へ上げて、右に移動させています： $\sqrt[k]{\beta}$ `\leftroot` に負の引数を与えれば β は右に寄ります。移動単位は、この調整に便利な大きさになっています。

4.7 枠で囲まれた数式

コマンド`\boxed`は引数で与えられた式を箱に入れます。`\fbox`と同じような働きですが, これは中身が数式モードのときだけに使います:

$$(4.2) \quad \boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))}$$

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

4.8 上につく矢印, 下につく矢印

基本の \LaTeX には`\overrightarrow`と`\overleftarrow`コマンドがあります。`amsmath`パッケージには, さらにいろいろな矢印が用意されています:

```
\overleftarrow      \underleftarrow
\overrightarrow    \underrightarrow
\overleftrightharrow \underleftrightharrow
```

4.9 長さが伸びる矢印

`\xleftarrow`と`\xrightarrow`は, とめて長い下付き添え字あるいは上付きに添え字にも対応した自動的に伸びる矢印を生成します。これらのコマンドは一つのオプション引数(添え字)と1つの必須引数(上付き文字, 空の場合もあります)を取ります。

$$(4.3) \quad A \xleftarrow{n+\mu-1} B \xrightarrow[n \pm i-1]{T} C$$

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

4.10 記号を別の記号に加える

\LaTeX は, 二項関係の上に上付き文字を置くための`\stackrel`を提供します。`amsmath`パッケージには, もっと一般的なコマンド`\overset`と`\underset`があります。これらのコマンドは, 記号を別の記号の上あるいは下に置くことができます。`\overset{*}{X}`とすれば添字サイズ(superscript-size)の`*`が`X`の上に置かれま:`\underset`は, これと同じような機能で, 記号の下に置きます。`\S7.2`の`\sideset`についての説明も参照してください。

4.11 分数, それに関連する構成

4.11.1 `\frac`, `\dfrac`, および`\tfrac`コマンド

`\frac`コマンドは, \LaTeX の基本コマンドですが, 引数を2つ取り, つまり分母と分子, 通常の分数の形を作ります。`amsmath`パッケージは, さらに便利な追加機能として`\dfrac`と`\tfrac`があります。これらは`\displaystyle\frac ...`およびの省略形です。`\textstyle\frac ...`。

$$(4.4) \quad \frac{1}{k} \log_2 c(f) \quad \frac{1}{k} \log_2 c(f) \quad \sqrt{\frac{1}{k} \log_2 c(f)} \quad \sqrt{\frac{1}{k} \log_2 c(f)}$$

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

4.11.2 \binom, \dbinom および \tbinom コマンド

$\binom{n}{k}$ のような二項関係のために amsmath は \binom, \dbinom および \tbinom を持っています:

$$(4.5) \quad 2^k - \binom{k}{1}2^{k-1} + \binom{k}{2}2^{k-2}$$

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

4.11.3 \genfrac コマンド

\frac, \binom そして、これらの変種との互換性のために、6つの引数をとる分数生成コマンド \genfrac があります。最後の二つは \frac の分子と分母に対応しています; 最初の二つはオプションのデリミタです (\binom で見た通りです)。三番目が線の太さ (\binom は、線の太さをの値を 0 にして、見えなくしているのです) を与えます; そして、四番目の引数は、上に乗る数式のスタイルです; 整数の値の 0-3 にたいして、それぞれ \displaystyle, \textstyle, \scriptstyle, および \scriptscriptstyle になります。三番目の値が与えられていないと、線の太さはデフォルトの標準の太さになります。

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

次に、\frac, \tfrac および \binom の定義を示します。

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

執筆されているドキュメントで、特別な記号として頻繁に \genfrac を使うのならば、好み (あるいは出版社の要請) に応じた、\frac と \binom にならった意味のある名前にした省略形を定義するのが便利でしょう。

プリミティブで一般化された分数コマンド \over, \overwithdelims, \atop, \atopwithdelims, \above, \abovewithdelims の使用は警告メッセージを出します。この理由については technote.tex に説明されています。

4.12 連分数

次の連分数

$$(4.6) \quad \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}}$$

は,

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

として生成できます. この見た目は`\frac`を使うよりも綺麗です. 分子の左あるいは右の配置は`\cfrac`ではなくて, `\cfrac[l]` あるいは `\cfrac[r]` を用いています.

4.13 スマッシュオペション

コマンド`\smash`は有効な高さがあり, 深さがゼロの部分式をタイプセットする時に, 部分式の記号と隣の記号の位置の調整をするために便利です. `amsmath` パッケージの`\smash`はオプション引数, `[t]` と `[b]` を持っています. 自然な深さや高さを保ちながら, ある部分の上あるいは下を広げる (“smash”) ことができると便利だからです. たとえば, 構成要素の高さや深さが異なるので, 位置やサイズを揃えたいときに`\smash`は, それらを自然に見せます. $\sqrt{x} + \sqrt{y} + \sqrt{z}$ と $\sqrt{x} + \sqrt{y} + \sqrt{z}$ を比べてください. あとは `\sqrt{x} + \sqrt{\smash[b]{y}} + \sqrt{z}` として作りました.

4.14 デリミタ

4.14.1 デリミタのサイズ

`\left` と `\right` を使った自動的なデリミタのサイズの生成には, 二つだけ制限があります: 一つは, デリミタで囲まれる最大の大きさに対応するものを機械的に作ってしまうことと, サイズの変更の範囲が連続的でなく, 突然大きすぎるものになることがあることです. これは, 指定されたデリミタのサイズに対して極端に大きい数式の部分があると, 普通のテキストモードで次のサイズである 3pt 程度の, 次に大きいサイズを選んでしまうからです. 二, 三の状況でデリミタのサイズが一般的に調整できる名前に ‘big’ を含むコマンドのセットがあります.

デリミタ サイズ	テキスト サイズ	<code>\left</code> <code>\right</code>	<code>\bigl</code> <code>\bigr</code>	<code>\Bigl</code> <code>\Bigr</code>	<code>\biggl</code> <code>\biggr</code>	<code>\Biggl</code> <code>\Biggr</code>
Result	$(b)(\frac{c}{d})$	$(b)(\frac{c}{d})$	$(b)(\frac{c}{d})$	$(b)(\frac{c}{d})$	$(b)(\frac{c}{d})$	$(b)(\frac{c}{d})$

まず最初に考えられる状況は, 総和作用素の上下に範囲を示す添え字をおくことでしょう. `\left` と `\right` を使うと, デリミタは必要以上に大きくなります. さらに `Big` あるいは `bigg` を使うと良い結果を得ます:

$$\left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p} \quad \text{見比べる} \quad \left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p}$$

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

次に考えられる状況は、複数のデリミタを`\left`と`\right`を使った場合、(内部の数式には十分なので) 同じ大きさになることです。しかし、デリミタのサイズを識別できるように変えたいでしょう。

$$((a_1b_1) - (a_2b_2))((a_2b_1) + (a_1b_2)) \quad \text{見比べる} \quad ((a_1b_1) - (a_2b_2))((a_2b_1) + (a_1b_2))$$

```
\left((a_1 b_1) - (a_2 b_2)\right)
\left((a_2 b_1) + (a_1 b_2)\right)
%\quad\text{versus}\quad
\quad\text{見比べる}\quad
\bigl((a_1 b_1) - (a_2 b_2)\bigr)
\bigl((a_2 b_1) + (a_1 b_2)\bigr)
```

三番目の状況はテキストの中にサイズが大きい要素が含まれる場合です。たとえば、 $\left|\frac{b'}{a'}\right|$ ではデリミタは`\left`と`\right`とで作られますが、行間を広げてしまいます。このような場合は、`\bigl`と`\bigr`を使うと、通常より大きいデリミタが使われますが、普通のテキストの行間に適切な高さです： $\left|\frac{b'}{a'}\right|$

通常、 \LaTeX の`\big`、`\bigg`、`\Big`、および`\Bigg` デリミタは \LaTeX のフォントの大きさにたいして適切に変換しません。amsmath パッケージを使えば、うまくゆきます。

4.14.2 垂直棒の記号

amsmath パッケージはコマンド`\lvert`、`\rvert`、`\lVert`、`\rVert` (`\angle`、`\rangle` 比較して) によって垂直棒|の上書き問題に対処しています。この文字は \LaTeX ドキュメントでは、さまざまな数学関係式で使われています：数論で割り切れる記号は $p|q$ 、あるいは絶対値を表す $|z|$ 、あるいは集合の記号での‘such that’条件、あるいは‘その点での値’を示す記号 $f_\zeta(t)|_{t=0}$ です。これらを一つの記号で表すことは、それほど悪いことではありませんが、正しい文字としての処理が行えませんし、数学の素養がある読者が識別しにくくしコンピュータを使った数学ドキュメントの再利用に向いていません。したがって意味に応じて、それぞれの|を使い分け、数学的に正しい記号を選ぶべきです。このことは、二重垂直棒`\lVert`にも当てはまります。すなわちデリミタには|を使うのをやめ`\lVert`を使います。なぜなら左右のデリミタは、隣の記号とは関係しない振る舞いが必要です；縦棒を使うペアの記号が必要なら、ドキュメントのプリアンブルに適切なコマンドを定義するの実際的です：次のようにすれば

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

絶対値`\abs{z}`は $|z|$ となり、ノルム`\norm{v}`は $\|v\|$ となります。mathools には、`\abs` のようなマクロを定義するためのコマンド`\DeclarePairedDelimiter`があります。これはサイズを適切に変えます。

—5—

オペレータの名前

5.1 新しいオペレータの定義

数学関数 \log , \sin , そして \lim などは普通は立体で描かれ、一つのイタリック文字で表される変数と見た目で区別されます。よく使われる関数 $\backslash\log$, $\backslash\sin$, $\backslash\lim$ などは定義済みですが、数学論文では、様々な新しい関数が使われます。そのため `amsmath` パッケージでは新しい‘オペレータ名’を定義する一般的な方法を用意しています。新しい数学関数 $\backslash\mathrm{xxx}$ を $\backslash\sin$ にのように定義するなら

```
\DeclareMathOperator{\xxx}{xxx}
```

と書きます。こうすれば $\backslash\mathrm{xxx}$ は xxx を適切なフォントが自動的に選ばれて両側に適切な空白どりで生成します。したがって、 $A\mathrm{xxx}B$ とせずに $A\mathrm{xxx}B$ とすればよいわけです。`\DeclareMathOperator` (名前を示す文字) の二番目の引数は、擬似テキストモードです：ハイフン文字-はマイナスではなくてテキストのハイフンとして、そして星印*は中央ではなく、やや上に配置されます。(比べてみましょう。 $a-b*c$ と $a-b*c$) しかし、名前を示す文字は数学モードが使われるので、上付きや下付きを使うことができます。新しいオペレータが上付き下付きを \lim , \sup , あるいは \max のように ‘limit’ 型の配置をしたいものであれば、`\DeclareMathOperator` コマンドの*形式を使います：

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

下付き添え字の配置については 7.3 節も参照してください。

次のオペレータの名前は定義済みです：

<code>\arccos</code>	\arccos	<code>\deg</code>	\deg	<code>\lg</code>	\lg	<code>\projlim</code>	\projlim
<code>\arcsin</code>	\arcsin	<code>\det</code>	\det	<code>\lim</code>	\lim	<code>\sec</code>	\sec
<code>\arctan</code>	\arctan	<code>\dim</code>	\dim	<code>\liminf</code>	\liminf	<code>\sin</code>	\sin
<code>\arg</code>	\arg	<code>\exp</code>	\exp	<code>\limsup</code>	\limsup	<code>\sinh</code>	\sinh
<code>\cos</code>	\cos	<code>\gcd</code>	\gcd	<code>\ln</code>	\ln	<code>\sup</code>	\sup
<code>\cosh</code>	\cosh	<code>\hom</code>	\hom	<code>\log</code>	\log	<code>\tan</code>	\tan
<code>\cot</code>	\cot	<code>\inf</code>	\inf	<code>\max</code>	\max	<code>\tanh</code>	\tanh
<code>\coth</code>	\coth	<code>\injlim</code>	\injlim	<code>\min</code>	\min		
<code>\csc</code>	\csc	<code>\ker</code>	\ker	<code>\Pr</code>	\Pr		
		<code>\varinjlim</code>	\varinjlim	<code>\varliminf</code>	\varliminf		
		<code>\varprojlim</code>	\varprojlim	<code>\varlimsup</code>	\varlimsup		

次のように使う `\operatorname` コマンド

```
\operatorname{abc}
```

は方程式で使いますが、これは `\DeclareMathOperator` で定義された $\backslash\mathrm{abc}$ と同値です。かなり複雑な気泡や他の目的によっては、こちらの方が便利です。(変種 `\operatorname*` は limit 型の添え字を持ちます。)

5.2 `\mod` とそのなかま

コマンド `\mod`, `\bmod`, `\pmod`, `\pod` は, “mod” 記号における特別な空白を持たせるために用意されました. `\bmod` と `\pmod` は \LaTeX にもありますが, `amsmath` パッケージは `\pmod` の空白どりをディスプレイ数式では小さい値にします. `\mod` と `\pod` は `\pmod` の変種で, これを好む著者のためにあります; `\mod` は丸括弧無しですが, `\pod` は “mod” を残して丸括弧を残します.

$$(5.1) \quad \gcd(n, m \bmod n); \quad x \equiv y \pmod{b}; \quad x \equiv y \bmod{c}; \quad x \equiv y \pod{d}$$

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

—6—

`\text` コマンド

コマンド `\text` の主な利用は, 単語あるいは成句 (フレイズ) の表示させるためです. \LaTeX コマンドの `\mbox` に効果はよく似ていますが, さらに良い機能があります. 単語や成句を下付き文字にしたい場合, `..._{\textrm{word or phrase}}` というようにタイプできます. これは `\mbox` を使う: `..._{\mbox{\rmfamily\scriptsize word or phrase}}` より簡単です. 標準の `\textrm` コマンドは, `amsmath` の `\text` の定義に使われますが, 強制的に `\rmfamily` が使われます.

$$(6.1) \quad f_{[x_{i-1}, x_i]} \text{ is monotonic, } i = 1, \dots, c+1$$

```
f_{[x_{i-1},x_i]} \text{ is monotonic,}
\quad i = 1,\dots,c+1
```

`\text` に使われるフォントは, 囲み環境と同じです; つまり定理では, `\text` はイタリックになります.

数式が `\text` の文字列に含まれている場合, 明示的に数式であると指定 (つまり `$...$`) しなければなりません.

$$\partial_s f(x) = \frac{\partial}{\partial x_0} f(x) \quad \text{for } x = x_0 + I x_1.$$

```
\partial_s f(x) = \frac{\partial}{\partial x_0} f(x)\quad
\text{for } x= x_0 + I x_1$.
```

関数の名前は `\text` で入力してはなりません. その代わりに `\mathrm` あるいは `\DeclareMathOperator` を適切に使います. これらは表現が変更されないのど, どこにおいても変わりません (定理環境でイタリックにならない). そしてオペレータの宣言で使われている場合, 適切な空白どりが自動的に使われます.

—7—

積分と総和記号

7.1 複数の行で作られる下付き上付き添え字

\substack コマンドは複数の行で作られる下付き上付き添え字に使われます:たとえば

$$\begin{array}{c} \sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j) \end{array}$$

やや一般的なのは subarray 環境で, それぞれの行を中央揃えでなく, 次のように左揃えにします:

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

7.2 \sideset コマンド

特殊な目的のために\sideset という名前のコマンドがあります: 上付き下付きの文字を \sum や \prod などの大きなオペレータ記号の四隅に配置します. 注意: このコマンドは, これらのような和を表すような記号以外で使われることを想定していません. 総和記号にプライムを配置したことが, 典型的な例です. 総和記号ではインを示す必要がなければ, 単に\nolimits を使えば良いでしょう: つまりディスプレイ数式で\sum\nolimits' E_n は

$$(7.1) \quad \sum' E_n$$

しかし, 総和記号にはプライムだけでなく, 下付きの位置にではなくて何かを起したことがあります, \sideset を使わないと難しいことです. \sideset を使えば, 簡単に

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

と書くことができます. 空白の中カッコのペアは, \sideset が大きなオペレータの四隅に記号を配置できることを見せています; 総乗記号の四隅に星印を配置したいのなら

$$\prod_{*}^{*} *$$

とします.

7.3 添え字と上限下限の配置

デフォルトの下付き添え字の位置は、基本となる記号によって変わります。総和などの記号ではいわゆる ‘displaylimits’ 位置です：総和などの記号をディスプレイ数式で表示するとき、上付き下付きは ‘limits’ 位置、つまり記号の上と下になります。しかし、インライン（文中）数式の場合は、右上と右下に配置されます。テキストの上下に余分な空白が出ないようにするためです。積分記号のデフォルトは、文中数式であっても、上付き下付きは横（右上、右下）に置きます。（`\intlimits` と関連するオプションについては 2 節を参照してください。）

`\sin` や `\lim` などのオペレータの名前は、それらの定義に従って ‘displaylimits’ あるいは ‘limits’ 位置が決まっています。標準的なオペレータの名前は通常の数学の慣習に従っています。

コマンド `\cnlimits` と `\nolimits` は基本となる記号の通常の振る舞いを上書きできます：

$$\sum_X, \quad \iint_A, \quad \lim_{n \rightarrow \infty}$$

`\sum` のような `displaylimits` の振る舞いするようにコマンドを定義するには末尾に `\cndisplaylimits` をおきます。`\limits`, `\nolimits`, あるいは `\displaylimits` が、続いて現れるときは、最後のものが優先されます。

7.4 多重積分の記号

`\iint`, `\iiint`, および `\iiiiiint` は多重積分の記号を、積分記号の間の空白を適切にして作ります。`\idotsint` は同じ考えを発展させたもので、積分記号の間にドットを表示させます。

$$(7.2) \quad \iint_A f(x, y) \, dx \, dy \quad \iiint_A f(x, y, z) \, dx \, dy \, dz$$

$$(7.3) \quad \iiiiiint_A f(w, x, y, z) \, dw \, dx \, dy \, dz \quad \int \cdots \int_A f(x_1, \dots, x_k)$$

—8—

可換図式

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ にあるいくつかの可換図式は別のパッケージ `amscd` にあります。複雑な可換図式が必要な著者には `TikZ`（とくに、`tikz-cd`）あるいは `XY-pic` が必要です。しかし対角線のない簡単な図式であれば、`amscd` コマンドが便利でしょう。次に例を示します。

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

```

\begin{CD}
S^{\backslash\mathrm{cal}\{W\}_\backslash\Lambda}\backslash\otimes T @>j>> T\backslash \\
@VVV @VVV{\backslash\mathrm{End}\ P}\backslash V\backslash \\
(S\backslash\otimes T)/I @= (Z\backslash\otimes T)/J \\
\end{CD}

```

CD 環境では、コマンド@>>>, @<<<, @VVV, および@AAA は、それぞれ右、左、下向き、上むきの矢印を与えます。水平の矢印ならば、最初と二番目の要素>あるいは<記号の間の要素は上付きで表示され、二番目と三番目の間の要素は上付きで表示されます。同じように、垂直の場合の最初と二番目、あるいは二番目と三番目の要素AやVは、“sidescripts”の左か右に表示されます。コマンド@=と@|は水平と垂直の二重線を与えます。“null arrow”コマンド@. は目に見える矢印で埋める代わりに必要な配列に使うことができます。

—9—

数学フォントを使う

9.1 イントロダクション

L^AT_EX でのフォントの使い方の包括的な議論は L^AT_EX font guide (fntguide.tex) あるいは *The L^AT_EX Companion* を参照してください。[4] L^AT_EX で数学フォントを指定するコマンドは\mathbf, \mathrm, \mathcal, \mathsf, \mathtt, \mathit です。これに加えて、数学アルファベットのコマンドは、黒板太字の\mathbb, Fraktur フォント\mathfrak, Euler 筆記体の\mathscr が amsfonts パッケージと euscript パッケージにあります（これらは個別に配布されています）。

9.2 数学フォントの勧められる使い方

あなたのドキュメントでいろいろ数学フォントを使い分ける必要があるなら、それらのフォントを短い名前にして、たとえば\mathbf の代わりに\mb として、使いたいでしょ。もちろん適切な\newcommand を使って、略語を提供することを妨げはしません。しかし、L^AT_EX は短い名前があり、実際には不便になり、もっと良い方法があります：コマンド名を使用するフォントをもとに作るのではなく、数学オブジェクトの名前をもとにしたコマンド名を定義しましょう。たとえば、ベクトルを示すために太字を使用するのなら、‘math-bold’ コマンドではなくて ‘vector’ コマンドを

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

のように定義するほうが後々有効です：これを使えば\vect{a} + \vect{b} とすれば **a + b** が得られます。しばらくして、太字のフォントを別の目的で使うことになって、ベクトルには矢印をつけることになったとしましょう。そのときは、\vect の定義を変えるだけですみます：そうできないときはドキュメントに使った\mathbf を変更しなければならなくなり、太字がベクトルなのかそうでないのか、いちいち確認しなければなりません。

異なる文字や特定のフォントに対して適切にコマンド名を作るのは便利です：

```

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

```

```
\DeclareMathSymbol{\R}{\mathalpha}{AMSb}{"52}
```

これらの定義によってコマンド`\C`と`\R`は、‘AMSb’ 数学記号フォントから黒板太字を生成します。ドキュメントで複素数あるいは実数を示す記号を度々使うのならば、たとえば`\field` コマンドを定義して`\field{C}`、`\field{R}`と書く方が、もっと便利だと思うでしょう。しかし、制限がをなくして自由に行うためには`\field`のようなコマンドを定義して、それを使って`\C`と`\R`を定義する方が良いでしょう：

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

9.3 太字の数学記号

`\mathbf` コマンドは、数式で太字のラテン文字を得るために、ひろく使われていますが、他の数学記号では太字にならないか、使っているフォントによっては信頼できる結果が得られません。たとえば

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

とすると、 $\Delta \Delta + \delta \delta$ となります；`\mathbf` コマンドはプラス記号あるいは小文字のデルタ記号に働いていません。`amsmath` パッケージでは、これに対処するために2つのコマンド`\boldsymbol`と`\pmb`を提供しており、その他の数学記号に働きます。`\boldsymbol`は`\mathbf`では太字にならない数学記号に使うことができます。ただし使っているフォントに太字が定義されている場合に限りです。`\pmb`は、最後の手段といえるもので、太字のフォントが定義されていなくても太字にします；“`pmb`”とは、“poor man’s bold”ということで、文字を少しずらして重ねることで太字を作ります。この出力結果の品質は、とくに細かい形状をもつ数学記号の場合は、劣ります。 $\mathrm{I\!T\!E\!X}$ の標準の数学フォント（Computer Modern）で`\pmb`が必要となるときは、大きな作用素、たとえば`\sum`、デリミタ記号、あるいは`amssymb` パッケージ [8] で提供されている数学記号などです。

次に示した数式は、いくつかの例を示したものです：

```
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}}
```

$$A_{\infty} + \pi A_0 \sim \mathbf{A}_{\infty} + \pi \mathbf{A}_0 \sim \mathbf{A}_{\infty} + \pi \mathbf{A}_0$$

If you want to use only the `\boldsymbol` command without loading the whole `amsmath` package, the `bm` package is recommended (this is a standard $\mathrm{I\!T\!E\!X}$ package, not an AMS package; you probably have it already if you have a 1997 or newer version of $\mathrm{I\!T\!E\!X}$). `amsmath` パッケージをロードせずに`\boldsymbol` コマンドだけを使いたいのなら、`bm` を勧めます（これは AMS のパッケージではなくて標準の $\mathrm{I\!T\!E\!X}$ パッケージです；つまり 1997 年以降に配布されている $\mathrm{I\!T\!E\!X}$ には備わっています）。

9.4 イタリックのギリシア文字

ギリシア文字の大文字のイタリックは、次のコマンドで生成できます：

<code>\varGamma</code>	Γ	<code>\varSigma</code>	Σ
<code>\varDelta</code>	Δ	<code>\varUpsilon</code>	Υ
<code>\varTheta</code>	Θ	<code>\varPhi</code>	Φ
<code>\varLambda</code>	Λ	<code>\varPsi</code>	Ψ
<code>\varXi</code>	Ξ	<code>\varOmega</code>	Ω
<code>\varPi</code>	Π		

—10—

エラーメッセージと出力の問題

10.1 一般的な注意

ここで解説する内容は \LaTeX マニュアル第8章（初犯の場合は第6章）[3]の補足です。*Companion* [4]の付録Bは \LaTeX のエラーがすべて記されています。`amsmath`に関連することも記されています。読者のために、ここでの説明は、上記の参考文献の内容と重複していますが、徹底的に解説していません。エラーメッセージはアルファベット順に示しました。冒頭に現れる`! LaTeX Error:`やアルファベットではないなど重要でないテキストは示していません。例を示す時には、画面に現れるヘルプメッセージ、ユーザが応答するさいのプロンプト`h`も記しました。

重要なエラーメッセージは、一番目の行に示されます。ノンストップモードで実行すると、エラーは累積し、最初のエラーが原因で、その後に回避できないエラーが発生する可能性があります。このような状況では、エラーの数が100個になった時に終了するため、報告されたエラーは完全ではない可能性があります。本当に重要なエラーとそうでないものの区別ができません。

最後の節では、出力時のエラーについて解説します。つまり、 \LaTeX の処理中のエラーでなくて、印刷結果が何かおかしいという事象です。

10.2 エラーメッセージ

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

例：

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

```
1.8 \begin{split}
```

```
? h
```

```
\Did you forget a preceding \begin{equation}?
```

```
If not, perhaps the 'aligned' environment is what you want.
```

```
?
```

説明：`split`環境は、これだけではディスプレイ方程式を構成しません。`equation`や`gather`のような他の環境で使う必要があります。

■ Erroneous nesting of equation structures

例：

```
! 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.

...

```
1.260 \end{alignat*}
          \end{equation*}
```

説明：数式構造 align, alignat などはトップレベルで利用するために設計されているので、他のディスプレイ数式の構造に入れることはできません。重要な例外としては align とこの変種だけは envgather の中に入れることができます。

■ Extra & on this line

例：

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

See the amsmath package documentation for explanation.

Type H <return> for immediate help.

...

```
1.9 \end{alignat}
```

? h

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

?

説明：alignat 構造の中では、1 行あたりの整列される個数は、\begin{alignat} の後に与えられる数値引数によって決まります。行中により多くのアラインメントポイントを使用すると、間違って改行コマンド\\を省略したとみなされ、上記のエラーが発行されます。

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

例：

```
! Font OMX/cmex/m/n/7=cmex7 not loadable: Metric (TFM) file not found.
<to be read again>
          relax
```

```
1.8 $a
      b+b^2$
```

? 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 `\font<same font id>=<substitute font name>'.`
?

説明：Computer Modern フォントの大きなサイズのいくつかは、標準 L^AT_EX の一部として（1994 年 6 月までは）AMSF_{onts} で配布されていました：cmex7-9, cmmib5-9, および cmb_{sy}5-9. これらの大きなサイズのフォントが手元のシステムに含まれていないなら、まず最初に L^AT_EX を入手したところに有無を調べます。そこになれば、CTAN（たとえば Metafont ソースファイルはディレクトリ `/tex-archive/fonts/latex/mf`, あるは PostScript Type 1 フォーマットはディレクトリ `/tex-archive/fonts/cm/ps-type1/bakoma`）から手に入ります。

フォントの名前が cmex ではじまるときは、amsmath のための特別なオプション cmex10 があります。このときは、`\usepackage` を次のようにします。

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

これが行なっていることは、cmex を使うところでは、すべて 10 ポイントの大きさのフォントを使うということです。作成したドキュメントによっては、これで十分な場合があります。

■ Improper argument for math accent

例：

```
! 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.

...

```
1.415 \tilde k_{\lambda_j} = P_{\tilde \mathcal{M}}
```

?

説明：任意の L^AT_EX コマンドでは単純ではない引数は中括弧で囲む必要があります。この例では、次のように中カッコの追加が必要です：

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

■ Math formula deleted: Insufficient extension fonts

例：

```
! Math formula deleted: Insufficient extension fonts.
1.8 $ab+b^2$
```

?

説明：これはたいていの場合、その前で示した Font ... not loadable に原因があります：上で示した解決策をみてください。

■ Missing number, treated as zero

例：

```
! Missing number, treated as zero.
<to be read again>
```

```
          a
1.100 \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.)
```

```
?
```

説明：このエラーの原因は、いろいろ考えられます。しかし、適切な `amsmath` パッケージでの可能性は、`alignat` 環境で、正しい値を引数に指定していないことです：

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

この例で、最初の行は次のようにすべきです。

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

もう一つの可能性は、`array`, `tabular`, または `eqnarray` のような複数行の構文で改行コマンド`\\`の後に左括弧`[`があることです。これは、`[`が次の行に記述されていて内容の一部としたい場合でも、`LaTeX` によって‘追加の垂直の空白’が要求された開始点として解釈されます [3, §C.1.6]。たとえば

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

このような場合のエラーメッセージにたいしてできることは、`LaTeX` マニュアル [3, §C.1.1] で示されている通り中かっこで囲むことです：

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

あるいは`\relax`によって中かっこを前に置きます。

■ Missing `\right.` inserted

例：

```
! Missing \right. inserted.
<inserted text>
      \right .
```


1.10 `\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.

説明：このエラーは典型的には、複数行の環境で改行あるいは&をデリミタのペア`\left-\right`の中に置いた時に現れます。splitの場合も同様です。

```
\begin{multline}
```

```
AAA\left(BBB\
```

```
CCC\right)
```

```
\end{multline}
```

解決策は2つあります：(1) `\left` と `\right` を使わずに、サイズが決まっている大きなデリミタ (`\bigl \bigr \biggl \biggr ...`; §4.14.1 をみてください) を使います；あるいは (2) `\left-\right` ペアの中にヌルデリミタを入れて、各行（あるいはセル）を分けます：

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

```
\left.CCC\right)
```

最後の解決策は、デリミタのサイズが合わないことがあります；小さすぎるデリミタが選ばれたら `\vphantom` を使って必要な高さを与えます（あるいは、大きなデリミタが使われている部分に `\smash` を入れます）。`\vphantom` の引数には、他の部分での最大の高さを要素を入れます。つまり、

```
xxx \left(\int_t yyy\right.\
```

```
\left.\vphantom{\int_t} zzz ... \right)
```

❏ Missing } inserted

例：

! Missing } inserted.

<inserted text>

\right .

```
1.10 \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.

説明：このエラーが発生する多くの場合は、複数行のディスプレイ数式環境の中で\$を使った結果です。そのような\$記号を取り除きます（ただし`\text{...}`の中にあるものはそのままです）。

■ Old form `\pmatrix` should be `\begin{pmatrix}`.

例：

```
! 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.

?

説明：amsmath パッケージが使われていると、`\pmatrix`、`\matrix`、および `\cases` の古い形は、名前が衝突するために使えません。これらのシンタックスは LaTeX のシンタックスと両立しません。

■ Paragraph ended before `\xxx` was complete

例：

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.

?

説明：これは空白行が `\begin` と `\end` の間にある場合でしょう。あるいは他の可能性としては、`\end{multline}` コマンドのスペルを間違えた場合です。つまり

```
\begin{multline}
```

```
...
```

```
\end{multiline}
```

あるいは、環境名の `\begin{align}` と `\end{align}` に `\bal` と `\eal` のように省略したコマンドを使っているのでしょう：

```
\bal
```

```
...
```

```
\eal
```

技術的に理由によって、このような省略名は `amsmath` が提供する複雑なディスプレイ数式環境 (`gather`, `align`, `split` など) では働きません: 詳しくは `technote.tex` を参照してください).

■ Runaway argument?

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

■ Unknown option ``xxx'` for package ``yyy'`

例:

```
! 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.
?
```

説明: これは、オプション名のスペルを間違えたか、単にパッケージにはあるはずだと思ったオプションがないことを意味しています。使用したいパッケージのドキュメントを調べます。

10.3 警告メッセージ

■ Cannot use ``split'` here

例:

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

説明: `split` 環境は、方程式全体にたいして、または `align` または `gather` 環境の行全体にたいして働くように設計されています。同じ構造内の前または後に同じような構造を置くことはできません:

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

■ Foreign command `\over` [or `\atop` or `\above`]

例:

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

説明: $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ のプリミティブな分数生成コマンド `\over`, `\atop`, `\above` は `amsmath` では使用できません。これらのシンタックスは $\mathrm{IAT}_{\mathrm{E}}\mathrm{X}$ のものでない上に、`amsmath` パッケージは $\mathrm{IAT}_{\mathrm{E}}\mathrm{X}$ にとって自然で同値なコマンドを提供しています。さらに詳しいことは `technote.tex` をみてください。

10.4 正しくない出力

❗ 説番号が正しい 1, 2, 3 でなくて 0.1, 5.1, 8.1 のようになってしまう

この問題は `\numberwithin` の引数が逆順に並んでいるからでしょう：

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

つまり ‘節の番号を *equation number.section number* として出力し、式が出現するたびに 1 にリセット’ されているようなので、次のように数式番号の後ろに節番号入れます。

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

❗ `\numberwithin` コマンドは式番号に対してはなんの効果ない

あなたの文章の、最初の節をよく見ましたか？ 節の番号を確認して、前の項目で説明した問題かどうかを調べましょう。

❗ Bracketed material disappears at the beginning of `aligned` or `gathered`

多くの複数行の数式環境で `amsmath` は \LaTeX の規約は機能せず、`[bracketed]` という表記は、これが環境の最初に現れるとき、あるいは `\` のすぐ後に現れるときにはオプションとして解釈され、空白が挿入されます。これは `aligned` と `gathered` の冒頭で失敗しますが、バグです。開きの中かっこの前に `\relax` を挿入することで、目的の結果を得ます。

—11—

さらなる情報

11.1 他のパッケージとの互換性

`amsmath` に備わっていない機能は、ほかのパッケージで補われています。

- `bm` は、太字の数学記号が必要な時には勧められます。
- `mathtools` は `amsmath` 環境の機能を拡張を与えているだけでなく、互換性のある拡張、`rcases` (`cases` と似ていますが、中かっこが右に表示されます)、`multlined` (`multline` と互換性のある副環境)、`\shortintertext` (`\intertext` に比べて小さい空白)、を与えています。`mathtools` を指定すると `amsmath` は自動的にロードされるので、個別にロードする必要はありません。
- `unicode-math` は、Unicode の（ほとんどの？）数学記号を与えます。

これらのパッケージは `amsmath` を使った時に問題があることは知られています。

- `lineno` はディスプレイ環境の数式番号を省きます。
- `breqn` は他のパッケージのたくさんのコマンドを再定義しています。`amsmath` や他の数式関連パッケージの後でロードするのが良いです。
`breqn` は全体として安定しておらず、たくさんのバグが [tex.stackexchange\[13\]](https://tex.stackexchange.com/questions/13) などで報告されています。
- `wasysym`, `mathabx` および多分他のフォントパッケージは、`amsmath` と同じ名前で多重積分を定義しています。

11.2 既存のドキュメントを利用する

11.2.1 plain L^AT_EX からの再利用

ドキュメントのプリアンブルに`\usepackage{amsmath}`を追加しておけば、L^AT_EX ドキュメントの仕上がりはほとんどの点で同じです。ただし、デフォルトでは、`amsmath` パッケージは、`eqnarray`, `align`, `gather` などの複数行のディスプレイ方程式の構造内での改ページを抑制します。`amsmath` を使用することにしたなら、`eqnarray` の中での改ページ許可するには、ドキュメントのプリアンブルに次の行を追加する必要があります：

```
\allowdisplaybreaks[1]
```

関係記号（イコールなど）の両側に通常の空白を入れるためには `eqnarray` ではなくて、できるだけ `align`, `multline`, あるいは `equation/split` を使います。

`amsmath` パッケージを使用した場合の大きな違いは多くの微調整が出来ることです、たとえば、以前のような`\newcommand{\Hom}{\mbox{Hom}}` としなくても

```
\DeclareMathOperator{\Hom}{Hom}
```

とすれば良いです。

11.2.2 AMS-L^AT_EX 1.1 からの再利用

`diffs-m.txt` を参照してください。

11.3 技術的注意

`technote.tex` には、あまり一般的ではないのですが、いくつかの技術的な事柄で注意すべきことが載っています。

11.4 助けが必要な時

`amsmath` は、L^AT_EX チームが管理しています。[11] バグを見つけた方は <https://www.latex-project.org/bugs/> に従って連絡してください。`amsmath` と関連した AMS の他のドキュメント作成クラス (`amsart`, `amsbook`, および `amsproc`) とは異なるので混乱しないでください；クラスのバグは直接 AMS の `tech-support@ams.org` 宛に連絡してください。

オンラインフォーラム `tex.stackexchange` [13] は、よくある疑問に対しての答えを見つけるための良い情報源です。質問を投稿する前に、自分の疑問がすでに投稿されているかを確認してください；見つからなければ、その時があなたが質問をする番です。

ディスカッショングループ `comp.text.tex` [12] があります。ここは L^AT_EX と T_EX について、なかなか良い情報源です。このグループは、`tex.stackexchange` 以前の Unix のニュースグループが始まりですので、歴史的な内容を調べるのに便利です。

11.5 関心のある方へ

AMSF_{onts} あるいは他の T_EX 関連ソフトウェアを www.ams.org にある AMS のウェブサーバーから入手したければ電子メールで `tech-support@ams.org` へてに問い合わせてください。このソフトウェアは CTAN にも提供されており、T_EX Live として配布されているものの中にあります。

T_EX Users Group は非営利団体で、会員向けの雑誌 (*TUGboat*) を発行し、ミーティングの企画を行い、T_EX と T_EX 関連のソフトウェアについての情報を収集して管理しています。

T_EX Users Group
PO Box 2311
Portland, OR 97208-2311
USA
Phone: +1-503-223-9994
Email: office@tug.org

T_EX Users Group のメンバーになることで T_EX 関連の自由なソフトウェアをサポートできます。色々な国や地域に T_EX ユーザグループがあるでしょう；それらについても T_EX Users Group から得ることができます。

参考文献

出版物

- [1] George Grätzer, *More Math into L^AT_EX*, fifth ed., Springer, New York, 2016.
- [2] Donald E. Knuth, *The T_EXbook*, Addison-Wesley, Reading, MA, 1984.
- [3] Leslie Lamport, *L^AT_EX: A document preparation system*, 2nd revised ed., Addison-Wesley, Reading, MA, 1994.
- [4] Frank Mittelbach, Michel Goossens, et al., *The L^AT_EX companion*, second ed., Addison-Wesley, Reading, MA, 2004. 本書は電子書籍でも入手できます。英語版とドイツ語版があります；詳しくは <https://www.latex-project.org/help/books/> をご覧ください。ここには、各種の書籍の一覧が掲載されています。
- [5] Frank Mittelbach and Rainer Schöpf, *The new font family selection—user interface to standard L^AT_EX*, *TUGboat* **11**, no. 2 (June 1990), pp. 297–305.
- [6] Michael Spivak, *The joy of T_EX*, 2nd revised ed., Amer. Math. Soc., Providence, RI, 1990.

パッケージ付属のドキュメンテーション

- [7] *AMS author handbook*, separate versions for journal articles, monographs and proceedings articles, Amer. Math. Soc., Providence, RI, 2017; <https://www.ams.org/publications/authors/tex/author-handbook>.
- [8] *AMSF_{onts} version 2.2d—user’s guide*, Amer. Math. Soc., Providence, RI, 2002; distributed with the AMSF_{onts} package <http://mirror.ctan.org/tex-archive/fonts/amsfonts/doc/amsfndoc.pdf>.
- [9] *Using the amsthm Package*, version 2.20.3, Amer. Math. Soc., Providence, RI, 2017; <http://mirror.ctan.org/tex-archive/macros/latex/required/amscs/doc/amsthdoc.pdf>.
- [10] Morten Høgholm, Lars Madsen, *The mathtools package*, 2018; <http://mirror.ctan.org/tex-archive/macros/latex/contrib/mathtools/mathtools.pdf>.

オンラインにある情報

- [11] The L^AT_EX Project, <https://www.latex-project.org>.
- [12] Online discussion group `comp.text.tex`, <https://groups.google.com/forum/#!forum/comp.text.tex>.
- [13] Online question and answer forum `tex.stackexchange`, <https://tex.stackexchange.com>.

Index

&, 29
\\!, 3, 13
\\,, 3, 13
\\:, 13
\\;, 13
\\[... \\], 4
\\, 4, 6, 9, 10, 26, 28, 32
*, 10
\\|, 18
|, 18

\\above, 16, 31
\\abovewithdelims, 16
\\accentedsymbol, 2
accents package, 14
\\addtocounter, 11, 12
\\addtolength, 6
align environment, 3, 4, 6–8, 26,
31, 33
\\align, 11
alignat environment, 7, 8, 26, 28
aligned environment, 3, 4, 8, 10,
32
alignedat environment, 4, 8, 10
alignedleftspaceno option, 2, 3
alignedleftspaceyes option, 2, 3
alignedleftspaceyesifneg
option, 2, 3
\\allowdisplaybreaks, 10
amsart class, 1, 33
amsbook class, 1, 33
amsbsy package, 1, 2
amscd package, 1, 2, 22
AMSTeX collection, 27
amstext package, 1, 2
amsthm package, 1, 35
amsxtra package, 1, 2, 14
\\arccos, 19
\\arcsin, 19
\\arctan, 19
\\arg, 19
array environment, 9, 12, 28
arrows
 extensible, 15
 in commutative diagrams, 22
\\atop, 16, 31
\\atopwithdelims, 16

b (bottom) option, 9, 17
BaKoMa fonts, 27
\\big, \\Big, \\bigg, ... delimiters,
17, 18
\\biggl, 29
\\biggr, 29
\\bigl, 29
\\bigr, 29
\\binom, 16
binomials, 16
bm package, 2, 24, 32
Bmatrix environment, 12
bmatrix environment, 12
\\bmod, 20
\\boldsymbol, 24
\\boxed, 15
breqn package, 32

\\C, 24
c (center) option, 9
cases environment, 9, 32
\\cases, 30
CD environment, 2, 23
\\cdots, 13
centertags option, 2
\\cfrac, 17

- cmbsy5, 27
- cmex, 27
- cmex10, 27
- cmex7, 27
- cmmib5, 27
- commutative diagrams, 22
- comp.text.tex, 33
- continued fractions, 16
- \cos, 19
- \cosh, 19
- \cot, 19
- \coth, 19
- \csc, 19
- CTAN (Comprehensive T_EX
Archive Network), 33
- \dbinom, 16
- dcases environment, 9
- \ddddot, 14
- \dddot, 14
- \ddot, 14
- \DeclareMathOperator, 1, 19, 20
- \DeclarePairedDelimiter, 18
- \deg, 19
- delimiters, 12, 16–18, 24, 29
 - fixed size, 17–18, 29
 - mismatched sizes, 29
 - null, 29
- \det, 19
- \dfrac, 15
- diffs-m.txt, 33
- \dim, 19
- \displaybreak, 10
- displayed equations, 3
 - centering, 3
- \displaylimits, 22
- displaymath environment, 4
- \displaystyle, 9, 15, 16
- \documentclass, 3
- \dot, 14
- dots, *see* ellipsis dots
- \dots, 14
- \dotso, 13
- \dotso, 13
- ed environments, 8–9, 32
- ed 環境, 8–9
- ellipsis dots
 - in matrices, 12
- eqnarray environment, 1, 3, 4, 28, 33
- \eqref, 11
- equation environment, 1, 3, 4, 6, 8, 25, 33
- equation * environment, 4
- equation numbers
 - cross-references, 11
 - subordinate numbering, 11
 - vertical placement, 2
- equation* environment, 4
- equation numbers
 - hierarchy, 11
 - left or right placement, 3, 10
 - overriding, 4
 - vertical placement, 2
- equations, *see* displayed equations
- euscript package, 23
- \exp, 19
- \fbox, 15
- flalign environment, 8
- fleqn option, 3, 4
- fntguide.tex, 23
- \frac, 15–17
- fractions, 15–17
- \fracwithdelims, 2
- function names, *see* operator names
- gather environment, 3, 6, 8, 25, 31, 33
- gathered environment, 3, 8, 10, 32
- \gcd, 19
- \genfrac, 16
- \hat, 14
- \hdotsfor, 12
- \hom, 19
- horizontal spacing, 3, 13

- around operator names, 19, 20
- hyperref package, 4
- `\idotsint`, 22
- `\iiiint`, 22
- `\iiint`, 22
- `\iint`, 22
- `\inf`, 19
- `\injl`, 19
- integrals
 - multiple, 22, 32
 - placement of limits, 2
- `\intertext`, 10, 32
- `intlimits` option, 2, 22
- `\ker`, 19
- `\label`, 6, 11
- `\langle`, 18
- `\ldots`, 13
- `\left`, 17, 18, 29
- `\leftroot`, 14
- `leqno` option, 3, 4
- `\lg`, 19
- `\lim`, 1, 19
- `\liminf`, 19
- limits, *see* subscripts and superscripts
- `\limits`, 22
- `\limsup`, 19
- line break, 9, 14, 28, 29
- lineno package, 32
- `\ln`, 19
- `\log`, 19
- `\lVert`, 18
- `\lvert`, 18
- math fonts, 23
- math symbols, *see* math fonts
- mathabx package, 32
- `\mathbb`, 23, 24
- `\mathbf`, 23, 24
- `\mathcal`, 23
- `\mathfrak`, 23
- `\mathit`, 23
- mathools package, 18
- `\mathrm`, 20, 23
- `\mathscr`, 23
- `\mathsf`, 23
- mathtools package, 2, 9, 10, 12, 32, 35
- `\mathtt`, 23
- matrices, 12
 - ellipsis dots, 12
- matrix environment, 12
 - * variants, 12
- `\matrix`, 30
- `\max`, 19
- `\mbox`, 20
- `\medspace`, 13
- Metafont source files, 27
- `\min`, 19
- `\mod`, 20
- `\mspace`, 13
- multiline environment, 4, 6, 32, 33
- multlined environment, 32
- `\multlinegap`, 4, 6
- `namelimits` option, 2
- `\negmedspace`, 13
- `\negthickspace`, 13
- `\negthinspace`, 13
- `\newcommand`, 23
- `\nobreakdash`, 14
- `nointlimits` option, 2
- `\nolimits`, 21, 22
- `nonamelimits` option, 2
- `\normalfont`, 11
- `nosumlimits` option, 2
- `\notag`, 3, 4
- `\numberwithin`, 11, 32
- online Q & A forum, 33
- operator names, 19
- `\operatorname`, 19
- `\operatorname*`, 19
- options
 - adjust positioning, 14, 17
 - amsmath package options, 2–3
 - behavior of particular options, 4, 10, 12, 15, 17, 22, 27
 - extra vertical space after `\`, 9

- positioning of `-ed` environments, 9, 32
 - space before `[`, 9, 32
 - unknown, 31
- `\over`, 16, 31
- `\overleftarrow`, 15
- `\overleftrightharrow`, 15
- `\overrightarrow`, 15
- `\overset`, 15
- `\overwithdelims`, 16
- `pagebreak`, 10
- `\pagebreak`, 10
- `\par`, 4
- `pmatrix` environment, 12
- `\pmatrix`, 30
- `\pmb`, 24
- `\pmod`, 20
- `\pod`, 20
- PostScript fonts, 27
- `\Pr`, 19
- `\prolim`, 19
- `\qed`, 3
- `\qedhere`, 3
- `\qquad`, 13
- `\quad`, 13
- `\R`, 24
- `\raisetag`, 9
- `\rangle`, 18
- `rcases` environment, 32
- `\ref`, 4, 11
- `\ref*`, 4
- `\relax`, 28, 32
- `reqno` option, 3, 4
- `\right`, 17, 18, 29
- `\rVert`, 18
- `\rvert`, 18
- `\scriptscriptstyle`, 16
- `\scriptstyle`, 16
- `\sec`, 19
- `\setcounter`, 11, 12
- `\setlength`, 6
- `\shortintertext`, 10, 32
- `\shoveleft`, 4
- `\shoveright`, 4
- `\sideset`, 15, 21
- `\sin`, 1, 19
- `\sinh`, 19
- `smallmatrix` environment, 12
 - * variants, 12
- `\smash`, 17, 29
- `\sphat`, 14
- `split` environment, 2–4, 6, 10, 25, 29, 31, 33
- `\sptilde`, 14
- `\stackrel`, 15
- `subarray` environment, 21
- `subeqn.tex`, 4
- `subequations` environment, 11
- subscripts and superscripts, 15
 - multiline, 21
 - on sums, 21
 - placement, 2
- `\substack`, 21
- `\sum`, 22, 24
- `sumlimits` option, 2
- `\sup`, 19
- superscripts, *see* subscripts and superscripts
- `t` (top) option, 9, 17
- `tabular` environment, 28
- `\tag`, 4
- `\tag*`, 4
- `\tan`, 19
- `\tanh`, 19
- `\tbinom`, 16
- `tbtags` option, 2
- `technote.tex`, 16, 31, 33
- `testmath.tex`, 4
- TeX Live, 33
- TeX Users Group, 34
- `\text`, 1, 7, 9, 10, 20
- text fragments inside math, 10, 20
- `\textrm`, 20
- `\textstyle`, 9, 15, 16
- `\tfrac`, 15, 16
- `\theequation`, 11
- `\thickspace`, 13

\backslash thinspace, 13
 TikZ package, 22
 TikZ パッケージ, 22
 tikz-cd package, 22
 TUGboat, 34

 \backslash underleftarrow, 15
 \backslash underleftrightharrow, 15
 \backslash underrightharrow, 15
 \backslash underset, 15
 unicode-math package, 32
 \backslash uproot, 14
 \backslash usepackage, 2, 3, 27

 \backslash value, 11
 \backslash varDelta, 25
 \backslash varGamma, 25
 \backslash varinjlim, 19
 \backslash varLambda, 25
 \backslash varliminf, 19
 \backslash varlimsup, 19
 \backslash varOmega, 25
 \backslash varPhi, 25
 \backslash varPi, 25
 \backslash varprojlim, 19
 \backslash varPsi, 25
 \backslash varSigma, 25
 \backslash varTheta, 25
 \backslash varUpsilon, 25
 \backslash varXi, 25
 Vmatrix environment, 12
 vmatrix environment, 12
 \backslash vphantom, 29

 wasysym package, 32
 www.ams.org, 33

 \backslash xleftarrow, 15
 \backslash xrightarrow, 15
 XY-pic package, 22
 XY-pic パッケージ, 22

 オプション
 [の前の空白, 9
 amsmath パッケージのオプション, 2–3
 位置の調整, 14, 17

特別なオプションの振る舞い,
 4, 10, 12, 15, 22, 27
 ディスプレイ方程式, 3
 中央揃え, 3
 デリミタ, 12, 16, 18, 24
 固定サイズ, 17–18
 ドット, *see* 省略のドット

 下付きと上付き, 15
 配置, 2
 二項関係, 16
 作用素の名前, 19

 分数, 15
 小さい行列環境
 * variants, 12

 改ページ, 10
 改行, 9, 14
 数学フォント, 23
 数学記号, *see* 数学フォント
 数式, *see* ディスプレイ方程式
 数式番号
 垂直の配置, 2
 左あるいは右への配置, 3
 重なり, 4
 階層, 11
 数式の中の短いテキスト, 20
 数式内の短いテキスト, 10
 数式番号
 副番号, 11
 垂直の配置, 2
 相互参照, 11
 数式番号
 左あるいは右に配置, 10
 水平の空白
 作用素の両側の, 19, 20
 水平の配置, 3

 省略のドット
 行列の中で, 12
 矢印
 可換図式で使われる, 22
 長い, 15
 積分
 多重, 22

範囲指定の配置, 2
範囲, *see* 下付きと上付き

行列, 12
省略のドット, 12

行列環境
* 変種, 12

連分数, 16, 17
関数の名前, *see* 作用素の名前