LyX の数式詳細説明書

LyX プロジェクトチーム[1]

第 2.0.x 版

2011 年 5 月 15 日

[1] コメントや誤りの修正などがございましたら、LyX 文書化メーリングリスト lyx-docs@lists.lyx.org までお知らせください。

目次

1 はじめに

この文書は、LyX の数式機能の説明書であると同時に、なによりも数式記号および数式要素に使用される LaTeX コマンドのコレクションでもあります。説明は、コマンドの使用を念頭に置いています。したがって、ユーザーの手引きの数式の節をすでにお読みになっていることを前提としています。

この説明書で説明されている、ほとんどの数式記号と、数式要素の多くは、挿入▷数式メニューか数式ツールバーからアクセスすることが可能です。しかし、たくさんの数式を書かなくてはならない人はみな、数式ツールバーを使うよりもコマンドを使った方がずっと速いことに気付くことになるのです。したがって、この説明書はコマンドに焦点を当てますが、対応するツールバーボタンが利用可能なときには、それにも言及することにします。

とくに断らなければ、コマンドは数式内からのみ利用可能です。この文書で説明されているすべてのコマンドを利用できるようにするためには、文書設定（文書▷設定▷数式オプションメニュー）で AMS math パッケージを使うオプションを有効にしなくてはなりません[1]。

説明を明瞭にするために、この文書はすべての $\mathcal{AMS}$-math コマンド[2]を列挙はしません。

2 一般的な説明

本文に埋め込まれた行内数式を作成するには、短絡キー Ctrl+M, Alt+C M, Alt+M M のうちのいずれか、あるいはツールバーボタン  を使用してください。
大きく別の段落として表示される別行建て数式を作成するには、Ctrl+Shift+M, Alt+M D のうちいずれかの短絡キーを使用して下さい。

別行建て様式の数式を行内数式に変更するには、カーソルを数式内に合わせて Ctrl+M, Alt+C M, Alt+M M のいずれかの短絡キーか、編集▷数式▷数式の表記を変更メニューを使用して下さい。同じ方法が、行内数式を別行建て数式に変更するのにも使用できます。

行内数式の一部を別行建て数式の大きさで表示するには、**\displaystyle** を数式に入力して下さい。すると、青いボックスが新規に現れて、希望する数式の箇所を挿入することができます。

表の中では、行内数式のみの使用が許されています。

数式ツールバーは、表示▷ツールバーメニューで表示することができます。そのメニューで「数式」をクリックすると、ツールバーが下部に永続的に表示されます。この状態は、ツールバーメニューの中ではチェック印で表されます。この状態から、ツールバーメニューの「数式」をもう一度クリックすると、数式ツールバーは、カーソルが数式内部にあるときのみ表示されるようになります。この状態は、メニュー項目が「数式」から「数式（自動）」に変わることで表されます。

TeX モードは、ツールバーボタン  を押すか、挿入▷TeX コード（短絡キー Ctrl+L）メニューを使うことで、起動できます。

[1] AMS math パッケージを自動的に使うオプションは、LyX でサポートされている数式要素が見つかったときのみ、$\mathcal{AMS}$-math パッケージを使用します。

[2] すべての $\mathcal{AMS}$-math コマンドの一覧は、amsguide.psファイルに収録されています。このファイルは、すべての LaTeX 標準頒布版に含まれています。

LaTeX プリアンブルを変更するには、文書▷設定▷LaTeX プリアンブルメニューを使用してください。

行列や場合分け、多行数式を続けて編集するには、編集▷数式メニューと編集▷行と列メニューを使うか、表ツールバーを使用することができます。メニューから行や列を交換するように指定されたときには、カーソルのある列や行は、それぞれ右側の列や下の行と交換されます。カーソルが最後の列や行にあるときには、左の列や上の行と交換されることになります。

数式内で文章を書く[3]には、**数式テキスト**が使用されます。このモードには、短絡キー Ctrl+M を使うか、**\text** コマンドを挿入することで入ることができます。テキストは、LyX 中では黒字で表示されるので、青字で表示される他の数式部分とは区別することができます。出力においては、数式テキストは、他の数式部分とは違って、アップライト体に組まれます。

コマンドの構成

数式要素に使われるほとんどの LaTeX コマンドは、以下のような構成になっています。

\ コマンド名 [非必須引数]{ 必須引数 }

コマンドは、つねにバックスラッシュ「\」で始まります。非必須の引数を省略するときには、随伴する括弧も省略しなくてはなりません。必須引数の前後の括弧は、この文書中では、TeX 括弧と呼ぶことにします。数式中でコマンド名に左括弧を付けると、LyX は自動的に TeX 括弧を生成します。数式中ではそれ以外に、**\{** コマンドを使えば、つねに TeX 括弧を生成することができます。LyX 中で、青字で表示される通常の括弧とは違って、TeX 括弧は赤字で表示されます。TeX モード中では、TeX 括弧を得るのに、とくにコマンドは必要としません。また、TeX 括弧は出力中では表示されません。

記号のコマンドのように引数のないコマンドを TeX モードに入力するときには、コマンドの終わりを表すために、コマンドの後に空白が**かならず**入力されなくてはなりません。この空白は出力中には現れません。空白を出力中に表示したいときには、空白の後に、通常テキストモードの保護された空白が来なくてはなりません。

保護された空白は、Ctrl+Space で入力できます。

[3]多行数式では、**\intertext** コマンドが使用されます。18.7を参照のこと。

文法の説明

- 記号[4]␣は、空白文字を入力することを表します。
- → のような矢印は、キーボードから対応する矢印キーを押すことを表します。

使用できる単位

表 1: 使用できる単位

単位	名称／摘要
mm	ミリメートル
cm	センチメートル
in	インチ
pt	ポイント ($72.27\,\mathrm{pt} = 1\,\mathrm{in}$)
pc	パイカ ($1\,\mathrm{pc} = 12\,\mathrm{pt}$)
sp	スケールポイント ($65536\,\mathrm{sp} = 1\,\mathrm{pt}$)
bp	ビッグポイント ($72\,\mathrm{bp} = 1\,\mathrm{in}$)
dd	ディドー ($72\,\mathrm{dd} \approx 37.6\,\mathrm{mm}$)
cc	シセロ ($1\,\mathrm{cc} = 12\,\mathrm{dd}$)
ex	現在のフォントの文字「x」の高さ
em	現在のフォントの文字「M」の幅
mu	数式単位（$1\,\mathrm{mu} = {}^{1}/_{18}\,\mathrm{em}$）

[4] この可視化された空白文字は、**\textvisiblespace** コマンドを TEX モード中に挿入することで作ることができます。

3　基礎的な函数

3.1　指数および添字

添字は、アンダースコア「_」を打鍵するか、数式ツールバーボタン $\square_{\square}$ を使って入力することができ、指数は、キャレット「^」を打鍵するか、数式ツールバーボタン $\square^{\square}$ を使って入力することができます。

コマンド	結果
B_V	B_V
B^V	B^V
B^␣A	B^A

キャレットは、言語によってはアクセント記号として使用されているので、そのような場合には、母音字の後でキャレットを押すと、指数にならずにアクセントをつけることになってしまいます[5]。この場合に指数を作るには、上記の最後の例のように、キャレットの後に Space を押してください。

3.2　分数

分数は、コマンド **\frac** か数式ツールバーボタン $\frac{\mathbf{a}}{\mathbf{b}}$ で作ることができます。フォント寸法は、分数が行内数式にあるか別行建て数式にあるかに応じて、自動的に調整されます。数式ツールバーボタン $\frac{\square}{\square}$ を使えば、分数の種類を選ぶことができます。

コマンド **\dfrac** を使えば、つねに別行建て数式の大きさを持つ分数を作成することができます。また、コマンド **\tfrac** では、つねに行内数式の大きさで分数が表示されます。以下はこれらの例です。

これは、コマンド **\frac** を使用して作った分数 $\frac{1}{2}$ を含む行です。

これは、コマンド **\dfrac** を使用して作った分数 $\dfrac{1}{2}$ を含む行です。

コマンド	出力
\frac␣A↓B	$\frac{A}{B}$
\dfrac␣A↓B	$\dfrac{A}{B}$
\dfrac␣e^␣\frac␣1↓2↓↓3	$\dfrac{e^{\frac{1}{2}}}{3}$

[5]使用しているキーボード設定によっては、同様のことが母音以外の文字でも起こることがあります。

入れ子の分数を作るには、コマンド **\cfrac** が使えます。以下がその例です。

\frac を使用して作成

$$\frac{A}{B+\frac{C+\frac{E}{F}}{D}}$$

\cfrac を使用して作成

$$\cfrac{A}{B+\cfrac{C+\cfrac{E}{F}}{D}}$$

上記の例で使用したコマンドは、

\cfrac A↓B+\cfrac C+\cfrac E↓F↓D

です。

\cfrac は、他の分数中に入れ子になっている場合も含め、分数をつねに別行建て数式の大きさに設定します。

分子の揃え方は、指定することができます。**\cfracleft** コマンドは左揃えにし、**\cfracright** は右揃えにします。**\cfrac** は中央揃えです。以下の各分数は、それぞれの揃え位置を示しています。

$\cfracleft{A}{B+C}$, $\cfrac{A}{B+C}$, $\cfracright{A}{B+C}$

（註）\cfracleft と \cfracright は、生粋の LaTeX コマンドではなく、実体は、コマンド \cfrac[揃え位置]{ 分子 }{ 分母 } です。したがって、これらを TeX モードで使うことはできません。

ときに、以下のように **\cfrac** と **\frac** を組み合わせて使うと便利です。

$$\cfrac{A}{B+\frac{C+\frac{E}{F}}{D}}$$

斜めの分数線を持つ行内分数を作るには、コマンド **\nicefrac**（例：5/31）を使うか、コマンド **\unitfrac**（例：5/31）を使います。さらに、2 1/3 のような帯分数を作るコマンド **\unitfracthree** もあります。

（註）実は、**\unitfracthree** は生粋の LaTeX コマンドではなく、実体は **\unitfrac[自然数]{ 分子 }{ 分母 }** というコマンドなので、TeX コードでは使用できません。

分数線を変更できるような独自の分数の定義のしかたは、第 23.4節に説明があります。

3.3 根号

平方根は、**\sqrt** か数式ツールバーボタン$\sqrt{\square}$で作成することができ、他のすべての根号は、コマンド **\root** か数式ツールバーボタン$\sqrt[\square]{\square}$で作成することができます。

コマンド	出力
\sqrt␣A-B	$\sqrt{A-B}$
\root␣3↓A-B	$\sqrt[3]{A-B}$

平方根は、根号指数フィールドを空白のままにしておけば、**\root** でも作成することができます。

$\sqrt[\beta]{B}$ の例のように、指数のとる値によっては、根号への距離が近すぎることがあります。
この場合には、β が根号に触れてしまいます。これを避けるためには、以下のようなコマンド書式で、コマンド **\leftroot** と **\uproot** を使います。

\leftroot{ 距離 } および **\uproot{ 距離 }**

ここで「距離」は、指数を左あるいは上に動かす、Big Point（単位 bp；72 bp = 1 インチ）での数値です。これらのコマンドは、指数に書き込みます。このようにして、コマンド
\root\leftroot{-1→\uproot{2→\beta␣→B
は、正しく組版された数式 $\sqrt[\beta]{B}$ を生成します。

3.4 二項係数

二項係数は、コマンド **\binom** か数式ツールバーボタン  の下位メニューを使って挿入することができます。分数（**\frac**）と同様に、**\binom** の他に、コマンド **\dbinom** および **\tbinom** があります。二項係数のまわりの括弧に、他の括弧を使うには、コマンド **\brace** と **\brack** があります。

コマンド	出力
\binom␣A↓B	$\binom{A}{B}$
\dbinom␣A↓B	$\dbinom{A}{B}$
\tbinom␣A↓B	$\tbinom{A}{B}$
\brack␣A↓B	$\brack{A}{B}$
\brace␣A↓B	$\brace{A}{B}$

3.5 場合分け

コマンド	出力
\cases␣A→B>0	$\begin{cases} A & B>0 \end{cases}$
\cases␣Ctrl+Return	$\begin{cases} A & \text{for } x>0 \\ B & \text{for } x=0 \end{cases}$

\cases を挿入するか数式ツールバーボタン  を使用した後では、短絡キー Ctrl+Return か表ツールバーボタン  を使えば、新しい行を作ることができます。

コマンド **\cases** は、挿入▷数式▷Cases 環境メニューで挿入することもできます。

3.6 否定

\not を挿入することで、すべての文字を取り消し形で表示できます。文字はスラッシュを上書きされた形になります。

コマンド	出力
\not=	$\neq$
\not \le	$\not\leq$
\not \parallel	$\nparallel$

最後の例が示すように、すべての否定形がきれいに出力されるわけではありません。このことから、否定形に専用のコマンドを持つものもあります（第 13.1節および第 14節を参照）。

3.7 埋め草

たとえば同位体[6]を表示しようとすると、次のような問題が起こります。

上付き文字と下付き文字を使用して作った指数： $^{19}_{9}\mathrm{F}$

正しい指数： $^{19}_{9}\mathrm{F}$

短い方の指数は、既定で、長い方の指数の一文字目の下ないし上に配置されてしまいます。これを避けるには、一文字ないし複数の空の文字を生成するコマンド **\phantom** や数式ツールバーボタン[7]  があります。**\phantom** を挿入すると、二つの赤い矢印が重なった青枠が表示されます。矢印は、箱の中身の幅と高さの両方が、埋め草（指定した文字と同じ大きさの余白を確保するために使われる空打ち文字）として適用されることを示しています。したがって、**\phantom** の作る文字は、箱の中身の文字の大きさを持つ埋め草となります。

コマンド	出力
^19␣␣\phantom␣1→9␣F	$^{19}_{9}\mathrm{F}$
^235␣␣\phantom␣23→9␣F	$^{235}_{9}\mathrm{F}$
\Lambda^␣\phantom␣ii→t␣␣MMt	Λ^{t}_{MMt}

さらに、**\vphantom**（ツールバーボタン  ）および **\hphantom**（ツールバーボタン  ）というコマンドもあります。**\vphantom** は、枠内部の文字の最大高のみの空白を作り、幅は考慮しません。**\hphantom** は、枠の内容の幅のみの空白を作ります。このことから、これらの枠は一本の赤矢印のみで表示されます。

たとえば、**\vphantom␣a\int** は、積分記号[8]が最大高の文字なので、積分記号の高さを持つ空白を作ります。実際の適用例については、第 18.1.4節を参照してください。

[6] 同位体と化学記号の組版に関しては、第 20節に記述があります。

[7] ツールバーボタン  の下位メニューに入っています。

[8] **\int** コマンドは、積分記号を生成します。第 10.1節を参照してください。

埋め草は、メニュー挿入▷整形▷埋め草を使えば、以下のように本文中でも使用することができます。

これは本文です。
　　本文です。

3.8　横線

コマンド	出力
\overline␣A+B	$\overline{A+B}$
\underline␣A+B	$\underline{A+B}$
\overline␣\underline␣A+B	$\overline{\underline{A+B}}$

上記最後の例では、先に **\overline** が来ようが **\underline** が来ようが、関係ありません。

二重下線を引くには、**\underline** を二回使います。

文字の上下 6 本の線まで引くことができます。

自製の線は、以下の書式を持つ **\rule** コマンドで作成することができます。

\rule[垂直オフセット幅]{ 長さ }{ 厚み }

オプションの**「垂直オフセット幅」**は、行を上方に（値が負であれば下方に）移動させます。値としては、第 1表に掲げてある単位を用いることができます。以下に、
\rule[-2ex]{3cm}{2pt} および **\rule{2cm}{1pt}**
というコマンドを用いて作成したふたつの例を例示します。

この行には、 ——— 二本の線があります。

\rule は、メニュー挿入▷整形▷水平線を使っても、本文に挿入することができます。

これは一行の――――――――文章です。

3.9　省略符号

省略符号には、いくつかの種類が使用できます[9]。列挙のためには、ベースラインの点々（**\ldots**）を使用しますが、演算子の場合は、演算子と同じ高さの点々（**\cdots**）が必要です。**\dots** コマンドを使うと、LaTeX は次に来る文字がどのような種類の文字であるかによって、自動的にどの種類を使うかを選択します。

[9]数式ツールバー中の▪▪▪ボタンで表示されている下位メニューです。

コマンド	出力
A_1␣,\dots␣,A_n	$A_1, \dots, A_n$
A_1␣+\dots␣+A_n	$A_1 + \dots + A_n$
A_1␣,\ldots␣,A_n	$A_1, \ldots, A_n$
A_1␣+\cdots␣+A_n	$A_1 + \cdots + A_n$
\vdots	$\vdots$
\ddots	$\ddots$
\iddots	$\iddots$
いろいろな点々を使った3×3行列	$\begin{matrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nm} \end{matrix}$

挿入 ▹ 省略符号メニューで挿入される省略符号は **\ldots** です。

\iddots を使うには、文書設定の数式オプションにある mathdots パッケージを（自動的に）使うオプションのうちいずれかを有効にしなくてはなりません。
mathdots パッケージを使うオプションを使用すると、文書中のフォント様式や寸法が既定値でないときのあらゆるドットの表示が改善されます。

とくに行列には、複数列にわたることのできる省略符号があります。これは、以下の書式を持つ **\hdotsfor** コマンドで作ることができます。

\hdotsfor[距離]{ 列数 }

ここで「列数」は、何列に広げるかを指定します。「距離」は、点々のあいだの距離を示す因子です。

以下の行列では、2行目の1つ目の枠に **\hdotsfor[2]{4}** を挿入して、**\dots** コマンドの2倍の点間距離を持つ省略符号を挿入しています。

$$\begin{pmatrix} A & B & C & D \\ \hdotsfor[2]{4} \\ q & w & e & r \end{pmatrix}$$

省略符号を広げる対象となる行列フィールドは空白にしておく必要があることに注意して下さい。さもないと LaTeX エラーが発生します。

さらに、**\dotfill** コマンドを使えば、行の残りを点々で埋めることもできます。このコマンドの働きは、**\hfill** と同様のものです。第8.2節をご参照下さい。

たとえば、**A\dotfill␣B** コマンドは、

A . . . . B

のようになります。点々を使う **\dotfill** の直線版として、**\hrulefill**

A ——————————————————————————————————————— B

があります。これらのコマンドを本文で使用するには、これらのコマンドは TeX モードで挿入される必要があります。

4　行列

行列は、数式ツールバーボタンの  か挿入▷数式▷行列メニューで挿入することができます。すると、行列の行数・列数・配置方法・装飾を尋ねられます。ここで垂直配置は、行内数式内の行列でのみ意味を持ちます。

最初の行列は「上」配置 $\begin{matrix} A & D & G & J \\ B & E & H & K \\ C & F & I & L \end{matrix}$ で、二番目は「中央」配置 $\begin{matrix} A & D & G & J \\ B & E & H & K \\ C & F & I & L \end{matrix}$ 、三番目は「下」配置 $\begin{matrix} A & D & G & J \\ B & E & H & K \\ C & F & I & L \end{matrix}$ です。

水平配置は、各列がどのように配置されるべきかを指定します。これは、各列に対応した文字を一つずつ入力することによって設定します。l は左寄せ、c は中央揃え、r は右寄せを意味します。たとえば、第 1 列が左寄せで第 2 列と第 3 列が中央揃え、第 4 列が右揃えの 4×4 行列を作成するには、水平配置のところに **lccr** と入力します。通常、行列では各列は中央揃えですから、各列の既定値は **c** です。

水平行列の例です。

lll：$\begin{matrix} 10000 & D & G & \\ B & 10000 & H & \\ C & F & 10000 & \end{matrix}$, **ccc**：$\begin{matrix} 10000 & D & G & \\ B & 10000 & H & \\ C & F & 10000 & \end{matrix}$, **rrr**：$\begin{matrix} 10000 & D & G \\ B & 10000 & H \\ C & F & 10000 \end{matrix}$

つづいて行や列を追加したり削除したりするには、数式ツールバーボタンの  や  などや編集▷行と列メニューを使用することができます。また、行は Ctrl+Return で作成することもできます。

装飾は、行列の前後に選択した様式の括弧を加えます。他にも括弧は、**\left** コマンドや **\right** コマンドで作成することもできます（短絡キー Alt+M 括弧）。第 5.1.2節を参照してください。あるいは、以下のコマンドを使うこともできます。

コマンド	出力
\bmatrix␣2×2 行列	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
\Bmatrix␣2×2 行列	$\begin{Bmatrix} 0 & -i \\ i & 0 \end{Bmatrix}$
\pmatrix␣2×2 行列	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$

コマンド	出力
\vmatrix␣2×2 行列	$\begin{vmatrix} 0 & -i \\ i & 0 \end{vmatrix}$
\Vmatrix␣2×2 行列	$\begin{Vmatrix} 0 & -i \\ i & 0 \end{Vmatrix}$
\matrix␣2×2 行列	$\begin{matrix} 0 & -i \\ i & 0 \end{matrix}$

たとえば **\vmatrix** などを挿入すると、青枠が二つの垂直線のあいだに現れるので、そこに行列を挿入することができます。

じつは多行数式はすべて行列なので、行列の各列の間隔を変更するには、第 18.1.2節に説明されている距離 **\arraycolsep** をここでも使用することができます。

行間隔を変更するには、**\arraystretch** コマンドを使用します。以下のようにして使用します。

\renewcommand{\arraystretch}{ 伸長因子 }

\renewcommand コマンドは、伸長因子を定義済みの **\arraystretch** コマンドに割り当てます。たとえば行間隔を 2 倍にするには、因子として 2 を指定して下さい。すると、以降の行列すべてにこれが使用されるようになります。元の間隔に戻すには、**\arraystretch** に因子 1 を割り当てて下さい。

本文行中に行列を入れるには、**\smallmatrix** コマンドを使います。これを挿入すると、二つの点線に囲まれた青枠が現れます。この枠のなかに行列を入れることができます。

これは、本文行中の行列 $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$ です。

5 括弧と区分記号

5.1 垂直括弧と区分記号

コマンド	出力
(	(
{	{
[	[
\langle	⟨
\lceil	⌈
\lfloor	⌊
/	/
|	|

コマンド	出力
)	)
}	}
]	]
\rangle	⟩
\rceil	⌉
\rfloor	⌋
\\	\
\\|	‖

（注意）TEX モードでは、\\ コマンドはその場所に改行を入れてしまうので、バックスラッシュを入力するには **\textbackslash** を使わなくてはなりません。

上に列挙した文字すべてについて、以下の二小節で説明されているコマンドを使って、大きさを調整することができます。これらのコマンドを使用するにあたっては、**\langle** や **\rangle** コマンドを使用せずに < や > の文字を直接使用することができます。

5.1.1 手動の括弧丈

括弧の丈は、LATEX コマンドの **\big**、**\Big**、**\bigg** および **\Bigg** を使って、手動で指定することができます。**\big** が最小の大きさであり、**\Bigg** が最大の括弧丈になります。

これらのコマンドは、括弧の階層を強調するのに使われます。

すべての括弧が同じ大きさ： $((A+B)(A-B))^C$

こちらの方が良い： $\Big((A+B)(A-B)\Big)^C$

二つ目の数式では、**\Big((A+B)(A-B)\Big)^␣C** というコマンドが使われています。

以下は、すべての括弧丈の羅列です。

\Bigg(\exp\bigg<\Big[\big{\ln(3x)\big}^2␣\sin(x)\Big]^␣A␣\bigg>\Bigg)^0,5

$$\Bigg(\exp\bigg<\Big[\big\{\ln(3x)\big\}^2␣\sin(x)\Big]^␣A␣\bigg>\Bigg)^{0,5}$$

\big 型コマンドの他に、括弧と中身のあいだにもう少し空白を加える **\bigm** という派生型と、空白を追加しない **\bigl**-**\bigr** 派生型があります。**\bigl** コマンドの最後の l は、左括弧であることを示し、右括弧の場合には、l の代わりに r を用います。左括弧と右括弧は、それぞれ括弧の開始と終了に用いられます。

以下の表は、これらの派生型の比較です。

コマンド	出力
\Bigm(\bigm(\ln(3x)\bigm)^2␣\Bigm)	$\Big(\big(\ln(3x)\big)^2\Big)$
\Big(\big(\ln(3x)\big)^2␣\Big)	$\Big(\big(\ln(3x)\big)^2\Big)$
\Bigl(\bigl(\ln(3x)\bigr)^2␣\Bigr)	$\Big(\big(\ln(3x)\big)^2\Big)$
\bigl)\ln(3x)\bigr(	$\big)\ln(3x)\big($

5.1.2 自動の括弧丈

可変の丈を持つ括弧は、**\left** コマンドおよび **\right** コマンド、あるいは数式ツールバーボタンの[]で挿入することができます。**\left** および **\right** の直後には、必要とする括弧を挿入しなくてはなりません。すると、括弧丈は出力時に自動的に計算されます。

通常の括弧：**\ln(\frac␣A↓C␣)** というコマンドは

$$\ln(\frac{A}{C})$$

を生成します。

複数行の括弧：**\ln\left(\frac␣A↓C␣\right)** というコマンドは

$$\ln\left(\frac{A}{C}\right)$$

を生成します。

\left や **\right** の代わりに、短絡キー Alt+M 括弧を使うこともできます。これを使うと、LyX 中で即座に実際の括弧丈を確認することができるという利点と、対応する右括弧も生成されると

いう利点があります。
すると、先ほどの例を作るコマンドは **\ln Alt+M (\frac␣A↓C** となります。

左括弧あるいは右括弧を省略するには、ドットを挿入します。たとえば、**\left.\frac␣A↓B␣\right}** というコマンドは

$$\left.\frac{A}{B}\right\}$$

を生成します。**\left** コマンドおよび **\right** コマンドは、文書が再度読み込まれたときには、LyX によって正しい丈の括弧に変換され、省略された括弧は、点線として表示されます。

著名な LaTeX 頒布版は、すべて LaTeX の拡張である eTeX を使用しているので、これらの頒布版では、すべての括弧および極限に対して **\middle** コマンドも使用することができます[10]。このコマンドでは、物理ベクトル

$$\left\langle \phi \;\middle|\; J = \frac{3}{2}\,,\, M_J \right\rangle$$

で必要とされるように、次に続く文字の高さは、囲まれる括弧の高さに調節されます。物理ベクトルに関しては、第 23.3節に説明されているように特殊な LaTeX パッケージがあります。

5.2 水平括弧

コマンド	出力
\overbrace␣A+B␣^␣3	$\overbrace{A+B}^{3}$
\underbrace␣A+B␣␣5	$\underbrace{A+B}_{5}$
\overbrace␣\underbrace␣A+B␣w␣␣␣7␣␣^␣C	$\overbrace{\underbrace{A+B_w}_{7}}^{C}$

最後の例では、**\overbrace** が先に挿入されようが **\underbrace** が先に挿入されようが代わりはありません。

括弧をお互いに重ねる必要がある場合には、第 18節に説明されているように、次のような多行数式を使わなくてはなりません。

$$A \;=\; \underbrace{\underbrace{gggg + bbqq}_{r} + dddd}_{s}$$

一行目には、数式が一つめの括弧とともに挿入されています。ここで、空白コマンド[11]**\:**を最初の d の前に挿入しておくことが重要です。さもないと、q の後ろで終わる括弧のせいで、直後の「+」の周りに正しく空白が入ることが妨げられてしまう[12]ためです。二行目には、二つめの括弧

[10] **（訳註）** pLaTeX では、標準では **\middle** コマンドは使えません。よって、以下の例では「**\middle**|」の代わりに「**\biggm**|」を用いています。

[11] 空白コマンドは第 8.1章に説明があります。

[12] これは、括弧が文字として取り扱われないためです。第 10.3章参照。

が挿入されています。b の直前から始まるようにするために、まず **\hphantom{gggg+\:}** というコマンド[13]が挿入されています。この数式中の「+」も空白で囲まれるようにするために、この空白コマンドが必要になっています。二つめの括弧は、**\hphantom{bbqq+dddd}** コマンドの下に置きます。

以下の例のように、括弧が反対側に重なる場合には、もっと複雑になります。

$$A \quad = \quad \underbrace{gggg + \overbrace{bbqq}^{s} + dddd}_{r}$$

最初の数式行は、括弧が上に来ていること以外は、先の例の第二行と同じです。二行目には、二つめの括弧と一緒に数式が入っています。一行目の括弧と数式のあいだに余白が入ることを防ぐために、行間を減らさなくてはならないのですが、これは LyX のバグ[14]のせいで簡単にはできません。この問題を回避するためには、数式直前に TeX モードで **setlength{\jot}{-6pt}** というコマンドを入れて、大域的な数式行間 **\jot** を-6 pt に変更しなくてはなりません。**\jot** は、数式直後に同様のコマンドを使って標準値の 3 pt に戻します。数式中の行間について、詳しくは第18.1.1章に説明があります。

6 矢印

矢印は、数式ツールバーボタンの←か、以下の各小節に列挙してあるコマンドで挿入することができます。

[13] **\hphantom** に関する詳細は、第 3.7章を参照してください。

[14] LyX-bug #1505

6.1 水平矢印

コマンド	出力
\gets	$\gets$
\Leftarrow	$\Leftarrow$
\longleftarrow	$\longleftarrow$
\Longleftarrow	$\Longleftarrow$
\leftharpoonup	$\leftharpoonup$
\leftharpoondown	$\leftharpoondown$
\hookleftarrow	$\hookleftarrow$

コマンド	出力
\to	$\to$
\Rightarrow	$\Rightarrow$
\longrightarrow	$\longrightarrow$
\Longrightarrow	$\Longrightarrow$
\rightharpoonup	$\rightharpoonup$
\rightharpoondown	$\rightharpoondown$
\hookrightarrow	$\hookrightarrow$

コマンド	出力
\leftrightarrow	$\leftrightarrow$
\Leftrightarrow	$\Leftrightarrow$
\longleftrightarrow	$\longleftrightarrow$
\Longleftrightarrow	$\Longleftrightarrow$
\rightleftharpoons	$\rightleftharpoons$

コマンド	出力
\mapsto	$\mapsto$
\longmapsto	$\longmapsto$
\leadsto	$\leadsto$
\dasharrow	$\dashrightarrow$

たとえばベクトル記号の矢印のようにアクセントとして使用される矢印は、第 7節に一覧があります。

さらに、ラベル付き矢印として、**\xleftarrow** と **\xrightarrow** があります。これらのコマンドを数式に挿入すると、二つの青枠のついた矢印が現れるので、そこにラベルを入れることができます。矢印の長さは、ラベルの幅に応じて調整されます。

コマンド	出力
F(a)\xleftarrow␣x=a↓x>0→F(x)	$F(a) \xleftarrow[x>0]{x=a} F(x)$
F(x)\xrightarrow␣x=a↓x>0→F(a)	$F(x) \xrightarrow[x>0]{x=a} F(a)$

6.2 垂直矢印および対角矢印

コマンド	出力
\uparrow	$\uparrow$
\Uparrow	$\Uparrow$
\updownarrow	$\updownarrow$
\Updownarrow	$\Updownarrow$
\Downarrow	$\Downarrow$
\downarrow	$\downarrow$

コマンド	出力
\nearrow	$\nearrow$
\searrow	$\searrow$
\swarrow	$\swarrow$
\nwarrow	$\nwarrow$

垂直矢印は、第 5.1.1節および第 5.1.2節に述べられているコマンドを使うと、区分記号として使用することもできます。

7　アクセント

アクセントは、数式ツールバーボタンの  か、以下の各小節に列挙してあるコマンドで入力することができます。

7.1　一文字に付けるアクセント[15]

コマンド	出力
\dot␣A	$\dot{A}$
\ddot␣A	$\ddot{A}$
\dddot␣A	$\dddot{A}$
\ddddot␣A	$\ddddot{A}$
\vec␣A	$\vec{A}$
\bar␣A	$\bar{A}$
\mathring␣A	$\mathring{A}$

コマンド	出力
\tilde␣A	$\tilde{A}$
\hat␣A	$\hat{A}$
\check␣A	$\check{A}$
\acute␣A	$\acute{A}$
\grave␣A	$\grave{A}$
\breve␣A	$\breve{A}$

é のようなアクセントは、数式に直接入れることができます。LyX は、それを対応するアクセントコマンドに変換します。ウムラウトに関しては、母音の前に引用符を挿入する方法の方がよいでしょう。ウムラウトのある数式部分がドイツ語に指定してあれば、LaTeX は、引用符と母音をまとめて一つの文字として取り扱います。**\ddot** と違い、この方法では、以下の例に示すように「本物の」ウムラウトが作られます。

コマンド	出力
“i	$\ddot{\imath}$
\ddot␣i	$\ddot{i}$

\ddot に比べて良いもう一つの利点は、上記のアクセントコマンドが**数式中テキストでは使用できない**のに対し、ウムラウトは直接数式中テキストに変換されることです。（アクセントコマンドによる）アクセント付き文字を数式中テキストに変換すると、アクセントの下にある文字しか変換されません。これは、たとえばイタリック体やボールド体への変換など、他のすべての変換に関しても言えることです。

ウムラウトと他のアクセント付き文字は、数式中テキストに直接入れることができます。

7.2　演算子に付けるアクセント

\overset コマンドや **\underset** コマンドを使うと、それぞれ演算子の上や下に、文字をアクセントとして付けることができます。また、**\sideset** コマンドを使うと、文字を演算子の前や後ろに付けることができます。コマンド書式は、

[15] 本文中のアクセントについては、第 16.2節を参照。

\sideset{ 前置文字 }{ 後置文字 }

\sideset は、かならずアクセントを付ける演算子の前に置かなくてはなりません。複数の文字や他の演算子、記号もアクセントとして使用することができます。たとえば、**\sideset** を使って演算子の後だけに文字を配置したい場合には、最初の括弧の中には何も書かないようにしますが、括弧を省略することはできません。

たとえば、**\sideset{→\{'→\sum_k=1␣^n** というコマンドを入力すると、

$$\sum_{k=1}^{n}{}'$$

のようになります。

また、**\overset␣\maltese␣↑a** というコマンドならば、

$$\overset{a}{\maltese}$$

のようになります。最後の例からわかるように、**\overset** や **\underset** では、記号や文字にアクセントをつけることもできます。一方、**\sideset** では、このようなことはできません。

7.3 複数の文字に付けるアクセント

コマンド	出力
\overleftarrow␣A=B	$\overleftarrow{A=B}$
\underleftarrow␣A=B	$\underleftarrow{A=B}$
\overleftrightarrow␣A=B	$\overleftrightarrow{A=B}$
\underleftrightarrow␣A=B	$\underleftrightarrow{A=B}$

コマンド	出力
\overrightarrow␣A=B	$\overrightarrow{A=B}$
\underrightarrow␣A=B	$\underrightarrow{A=B}$
\widetilde␣A=B	$\widetilde{A=B}$
\widehat␣A=B	$\widehat{A=B}$

これらのコマンドでは、好きなだけ多くの文字にアクセントを付けることができます。しかし、**\widetilde** および **\widehat** のアクセントは、以下の例のように、出力では 3 文字分の長さにしかなりません。

$$A+\widetilde{B=C}-D$$

前小節で述べた **\overset** コマンドと **\underset** コマンドを使っても、複数の文字にアクセントを付けることができます。**\underset␣A=B↓*****というコマンドは、

$$\underset{***}{A=B}$$

のようになります。

8 空白

8.1 定義済みの空白

数式に水平方向の空白を挿入することが、必要になることがあります。これは、保護された空白（短絡キー Ctrl+Space）を挿入することで実現できます。「␣」が現れるので、Space を何回か押す

ことによって、8 種の異なる長さの空白のうち一つを選択することができます。空白は、数式ツールバーボタンの ␣ を押すか、特定のコマンドを入力することで、挿入することができます。挿入したコマンド如何に関わらず、直後に Space を押すことによって、長さを変更することができます。

コマンド	\,	\:	\;	\quad	\qquad	\!
保護された空白を挿入したのち、Space を叩く回数	0	1	2	3	4	5
出力	$A\,B$	$A\:B$	$A\;B$	$A\quad B$	$A\qquad B$	$A!B$

一番右の長さは、一見、空白を生まないように見えます。実はこれは負の長さなので、他の長さと異なり、LyX 中では赤で表示されます。以下のように、他にもう二つ、負の長さの空白があります。

コマンド	\negmedspace	\negthickspace
保護された空白を挿入したのち、Space を叩く回数	6	7
出力	$A\negmedspace B$	$A\negthickspace B$

負の空白を使うと、文字が重なってしまうことがあります。これを利用して、合字処理を強制することができます。これは、たとえば以下のように、和演算子に使えます。

コマンド	出力
\sum\sum_f_kl	$\sum\sum f_{kl}$
\sum\negmedspace\sum_f_kl	$\sum\negmedspace\sum f_{kl}$

イコール記号などの関係子は、つねに空白を前後に伴うようになっていますが、これを抑制するには、イコール記号を TeX 括弧で囲みます。以下の例にこれを示します。

$$\text{通常の数式} \qquad A = B$$
$$\text{空白なしの数式} \qquad A{=}B$$

二行目の数式を作るコマンドは、**A\{=→B** です。

物理単位には、値と単位のあいだに通常の空白ではなく、最小の空白を入れる必要があるために、それに適した空白が必要です。本文中の単位には、挿入▷整形▷小空白メニュー（短絡キー Ctrl+Shift+Space）で、最小の空白を挿入することができます。

違いを示す例を以下に掲げます。

24 kW·h　値と単位のあいだに通常の空白を入れた例
24 kW·h　値と単位のあいだに最小の空白を入れた例

8.2 可変長の空白[16]

指定した長さの空白が、**\hspace** コマンドで入力することができます。すると、ながい「␣」が現れます。長さは、「␣」を左クリックすることによって指定することができます。長さは負の値でも構いません。数式が使用できる空白をすべて使い尽くすだけの空白を挿入するには、**\hfill** コマンドを使用します。

コマンド（\hspace 長さ）	出力
A=B\hspace␣→A\not=C (3 cm)	$A = B \qquad A \neq C$
A\hspace␣→A\not=A (-1 mm)	$AA \neq A$
A=A\hfill␣B=B	$A = A \qquad B = B$

上記の最後の例では、使用できる空白は、表の列中もっとも長い要素によって規定されます。行内数式では、空白は、**\hfill** が挿入された行の長さに依存します。つまり、その行が全幅を使用している場合、空白はまったく作られません。また **\hfill** は、別行建て数式中では、**行頭下げ**数式スタイルが使われているときのみ、意味を持ちます（数式スタイルは第 17節で説明されています）。

\hfill の他にも、空白を模様で埋める **\dotfill** や **\hrulefill** といったコマンドがあります。用例については第 3.9節をご参照下さい。

本文中では、可変長空白は、挿入▷整形▷水平方向の空白メニューで挿入することができます。

（例）

この行には、　　2 cm の空白が入っています。

この行には、　　最大の空白が入っています。

8.3 行内数式周りの空白

行内数式前後の空白は、長さ **\mathsurround** を使って調節することができます。長さの値は、以下の書式を持つ **\setlength** コマンドを使って設定することができます。

\setlength{ 長さ名 }{ 値 }

\mathsurround を 5 mm の値に設定するには、以下のコマンド

\setlength{\mathsurround}{5mm}

を TEX モードで挿入します。すると、5 mm の空白がすべての行内数式の前後に設定されることになります。

この行には、周囲に 5 mm の余白を設定した行内数式　$A = B$　があります。

既定値に戻すには、**\mathsurround** を 0 pt の値に戻して下さい。

9 ボックスと枠

本文中のボックスについては、取扱説明書埋め込みオブジェクト篇の Boxes の章に述べられています。

[16] 数式中の垂直方向の空白については、第 18.1.1節をご覧下さい。

9.1 縁付きボックス

\fbox コマンドや **\boxed** コマンドを使えば、数式やその一部を枠の中に入れることができます。

どちらかのコマンドを数式に挿入すると、枠の中に青枠が現れ、数式の断片を入れることができます。**\fbox** の場合には、そのままではボックスの中身が数式テキストとして取り扱われてしまうので、Ctrl+M を使って、このボックスの中にもう一度数式を作らなくてはなりません。**\boxed** を使った場合には、新しい数式が自動的に枠内に作られます。

\fbox コマンドは、数式がつねに本文の大きさに設定されてしまうので、別行建て数式に枠を付けるのには適していません。逆に **\boxed** は、数式がつねに別行建て数式の大きさに設定されてしまうので、行内数式に枠をつけるのには適していません。

\fbox の拡張として、枠幅と配置も指定することができる **\framebox** コマンドがあります。**\framebox** は、以下の書式を持ちます。

\framebox[枠幅][位置]{ ボックスの内容 }

「位置」は、l か r の値をとります。l は、ボックス中で数式を左寄せ、r は右寄せにします。位置を指定しない時には、数式は中央揃えになります。
「枠幅」を指定しない時には、位置を指定することができません。この場合には、**\fbox** と同様、枠幅がボックスの内容に応じて調節されるのです。

\framebox コマンドを挿入すると、三つの青枠を含むボックスが現れます。最初の二つの枠は括弧で囲まれており、二つとも非必須の変数であることを意味します。三つ目の枠は、**\fbox** 同様、数式の断片を入れるためのものです。

コマンド	出力
\fbox␣Ctrl+M \int␣A=B	$\boxed{\int A = B}$
\boxed␣\int␣A=B	$\boxed{\int A = B}$
A+\fbox␣B	$A + \boxed{\mathrm{B}}$
\framebox␣20mm→→Ctrl+M \frac␣A↓B	$\boxed{\frac{A}{B}}$

枠の厚みも調節可能です。そのためには、以下のコマンドを数式の前に TEX モードで挿入しなくてはなりません。

\fboxrule “厚み” \fboxsep “距離”

“距離” は、枠とボックス内の一文字目との間の距離を示します。これを使った例として、以下の枠付き数式をご覧下さい。

$$\boxed{A + B = C}$$

この数式の直前には、

\fboxrule 2mm \fboxsep 3mm

というコマンドが、TEX モードで挿入されています。ここで与えられた値は、以後のすべてのボックスに適用されます。

標準の枠寸法に戻すには、

\fboxrule 0.4pt \fboxsep 3pt

というコマンドを、次の数式が始まる前に TEX モードで挿入しておきます。

9.2 縁なしボックス

縁のないボックスを作るには、**\mbox**・**\makebox**・**\raisebox** の三つのコマンドがあります。

\raisebox を使うと、ボックスを上付きにしたり下付きにしたりすることができます。しかし、通常の上付き文字・下付き文字とは違い、ボックス内の文字寸法はそのまま保たれます。**\raisebox** は、以下の書式で用いられます。

\raisebox{ 高さ }{ ボックスの内容 }

\fbox と同様、ボックスに数式を入れる際には、明示的に数式として入れる必要があります。**(註)** 下の最後の **\raisebox** のところで、Ctrl+M を一回でなく二回押すことによって、もう一段数式をいれています。これは、LYX が **\raisebox** を直接サポートしていないためです。

コマンド	出力
H\raisebox{2mm→\{al→ lo	$H^{\mathrm{al}}lo$
H\raisebox{-2mm→\{al→lo	$H_{\mathrm{al}}lo$
A=\raisebox{-2mm→\{Ctrl+M Ctrl+M \sqrt␣B	$A=\sqrt{B}$

縁がないことを除けば、**\mbox** コマンドは **\fbox** と同じであり、**\makebox** は **\framebox** と同じです。

9.3 色付きボックス

本節で説明されているコマンドをすべて使えるようにするためには、LATEX プリアンブルに

\usepackage{color}

という行[17]を書き加えて、LATEX パッケージの **color**[18]を読み込む必要があります。

ボックスに色を付けるには、**\colorbox** コマンドを以下の書式で使用します。

\colorbox{ 色 }{ ボックスの内容 }

[17]定義済みの色を使って、文書中のどこかで文章に色を付けてある場合、LYX は、自動的に LATEX パッケージ **color** を読み込みます。したがって、本パッケージが二度読み込まれる可能性があるわけですが、そうなったとしても問題は生じません。

[18]LATEX パッケージ **color** は、すべての標準的な LATEX 頒布版に含まれています。

ボックスの内容には、別のボックスが含まれても構いませんし、**\colorbox** 自体も、別のボックスに入っていても構いません（以下の二番目と三番目の例を参照してください）。ボックスに数式を含める場合には、**\raisebox** と同様、明示的に数式を作らなくてはなりません[19]。

選択できる定義済みの色としては、

black（黒）・**blue**（青）・**cyan**（シアン）・**green**（緑）・**magenta**（マゼンタ）・**red**（赤）・**white**（白）・**yellow**（黄）

があります。

コマンド	出力
\colorbox{yellow→\{A=B	A=B
\colorbox{green→\{\fbox␣A=B	A=B
\fbox␣\colorbox{green→\{Ctrl+M Ctrl+M \int␣C=D	$\int C = D$

\colorbox は、ボックスに色をつけるだけで、ボックス内の文字には色付けをしません。すべての文字に色付けするには、数式全体を選択し、文字様式ダイアログで欲しい色を選択します。このダイアログは、ツールバーボタン  か編集▷文字様式▷任意設定メニューで開くことができます。すると、数式番号も数式と同じ色になります。数式番号が数式の文字とは別の色になるようにするには、数式内部で色を変えなくてはなりません。

たとえば、

$$\int A = B \tag{1}$$

$$\int A = B \tag{2}$$

数式 (1) は、全体が赤色です。
数式 (2) は、数式番号を緑色にするために、まず全体を緑色にします。その後、数式内の文字を赤色にします。

ボックスの縁だけ別の色にするには、**\fcolorbox** コマンドを以下の書式で使用します。

\fcolorbox{ 縁の色 }{ 色 }{ ボックスの内容 }

つまり、**\fcolorbox** は **\colorbox** コマンドの拡張です。**\framebox** と同様に、縁の厚みは **\fboxrule** と **\fboxsep** で設定します。たとえば、

A=B

のようにします。

[19] これは、**\fcolorbox** コマンドにも当てはまります。

上記の数式は、以下のコマンドで作成されています。
\fcolorbox{cyan→\{magenta→\{A=B.

定義済みの色以外の色を使いたい場合には、まずその色を定義しなくてはなりません。

たとえば、「**darkgreen**」という色を定義するには、LaTeX プリアンブルに

\definecolor{darkgreen}{cmyk}{0.5, 0, 1, 0.5}

という行を書き加えます。

cmyk とは、**cyan**（シアン）・**magenta**（マゼンタ）・**yellow**（黄）・**black**（黒）の各色を表す色空間です。コンマで区切られた四つの数字は、この色空間における各色の出力強度です。強度は、0 - 1 の範囲をとることができます。定義には、**cmyk** の他に、**rgb** という色空間を使うこともできます。**rgb** とは、**red**（赤）・**green**（緑）・**blue**（青）の各色を意味し、この場合には、各色に対応した三つの出力強度を指定します。さらに、灰色の出力強度のみをとる **gray** という色空間もあります。

例として、文字が **yellow** に色付けされ、新しく定義した **darkgreen** という色を持つ縁付きボックスを挙げておきます。

$$\int A\,\mathrm{d}x = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)} \tag{3}$$

\textcolor コマンドを使うと、以下のように、自前で定義した色をテキスト中でも使用することができます。

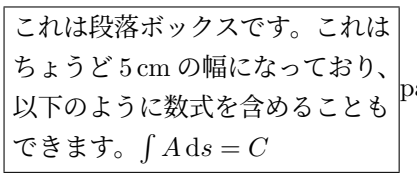
この文は「darkgreen」です。

\textcolor は、**\textcolor{ 色 }{ 色付けをする文 }** という書式で使用することができます。

9.4 段落ボックス

いくつかの行や段落を含む、いわゆる段落ボックス（parbox）は、挿入▷ボックスメニューかツールバーボタン  で作成することができます。

以下の例は、行中の縁付き parbox を示したものです。

この行は、 

parbox の入った行です。

このようなボックスは、灰色のボックス挿入枠を右クリックすることによって作ることができます。すると、ボックスの特性を表示したダイアログが現れます。上の例では、**装飾**：簡素な長方形の箱型、**内部ボックス**：parbox コマンド、**幅**：5 cm、**垂直ボックス配置**：中央、に設定されています。

LaTeX では、parbox は、以下の書式を持つ **\parbox** コマンドによって作られます。

\parbox[位置]{ 幅 }{ ボックスの内容 }

「位置」は、*b* と *t* の値をとることができます。下揃えを意味する *b*（bottom）は、ボックスを、周囲の本文中の最後の行と合わせることを意味します。上揃えを意味する *t*（top）は、これを最初の行に合わせます。位置を指定しない時には、ボックスは事実上中央揃えになります。用例については、取扱説明書埋込オブジェクト篇の *Boxes* の節をご参照下さい。

数式番号を含めて、数式を完全に縁で囲むためには、数式を parbox 内に収めなくてはなりません。こうするには、数式前に TEX モードで **parbox{\linewidth-2\fboxsep-2\fboxrule}{** というコマンドを挿入します。ここで **\linewidth** は、使用中の文書に設定されている行幅です。縁は、parbox の外側にあるので、縁余白と縁幅の 2 倍を行幅から差し引かなくてはなりません。バグ[20]のせいで LYX はこれを自動的に行いませんので、TEX モードを使用する必要があります。引数中で掛け算や引き算を行うためには、LATEX パッケージの **calc**[21]を、LATEX プリアンブル中で

\usepackage{calc}

のように読み込んでおく必要があります。数式の後では、TEX モードで **}}** を入力して、二つのボックスを閉じておかなくてはなりません。以下に例を挙げます。

$$\int A\,\mathrm{d}x = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)} \tag{4}$$

\fbox の引数として parbox が使われているので、この場合には、**\fbox** を使おうが **\boxed** を使おうが、差は生じません。

段落ボックスは、数式にじかにコメントを付けるのにたいへん便利です。これを行うには、**\parbox** を **\tag** コマンドといっしょに使います（**\tag** についての詳細は、第 19.4節をご参照下さい）

以下は、**\parbox** を使ってコメントを付けた数式の例です。

$$5x - 7b = 3b \tag{これは説明です。数式や多行数式本体からはっきりと離れています。}$$

LYX は、まだ数式中での **\parbox** コマンドをサポートしていないので、上のような数式は、完全に TEX モードで挿入しなくてはなりません。この数式は、以下のようなコマンド列を使って作ってあります。

まず、**\[5x-7b=3b\tag*\{\parbox{5cm}{** というコマンドを TEX モードで挿入します[22]。それから、説明を通常のテキストとして入れ、最後に **}}**\] を TEX モードで挿入します。ここで \[および \] コマンドは別行建て数式を作るためのものです。

[20] LyX-bug #4483

[21] **calc** は、標準的 LATEX 頒布版のすべてに含まれています。

[22] **行頭下げ**数式様式を使用している時には、**\tag*\{** の代わりに **\hfill** を用いることもできます（数式様式に関しては、第 17節をご参照下さい）。

\parbox を使う利点は、数式テキストモードを使用して「コメントを付けた」以下の例と比較すると、よくわかるでしょう。

$$5x-7b=3b \text{ これは説明です。数式本体から離れていません...}$$

10　演算子

10.1　大演算子

ここに挙げた積分演算子をすべて使えるようにするには、文書設定の数式オプションの面にある esint パッケージを自動的に使うオプションを有効にしなくてはなりません。

コマンド	出力
\sum	$\sum$
\int	$\int$
\oint	$\oint$
\ointctrclockwise	$\ointctrclockwise$
\ointclockwise	$\ointclockwise$
\sqint	$\sqint$
\bigcap	$\bigcap$
\bigcup	$\bigcup$

コマンド	出力
\prod	$\prod$
\coprod	$\coprod$
\bigodot	$\bigodot$
\bigotimes	$\bigotimes$
\bigoplus	$\bigoplus$
\bigwedge	$\bigwedge$
\bigvee	$\bigvee$
\bigsqcup	$\bigsqcup$
\biguplus	$\biguplus$

すべての大演算子は、数式ツールバーボタンの ∫ でも挿入することができます。

これらの演算子は、よく見ないと同じように見える二項演算子よりも大きいので、大演算子と呼ばれます。大演算子はすべて、次小節で説明する「範囲」をとることができます。

積分演算子はすべて、**\intop** や **\ointop** のように、**op** で終わる別バージョンがあります。これらの演算子は、**\int** とは範囲の表示のしかたが異なります。第 10.2節をご参照下さい。

積分の子細

積分中で用いられる文字 d は演算子なので、アップライト体で組まれなくてはなりません。これを行うには d を選択して、短絡キー Alt+C R を用います[23]。最後に、演算子の慣例に倣って、d の前に最小空白を挿入しなくてはなりません。たとえば、

正しくない例：$\int A(x)dx$
正しい　　例：$\int A(x)\,\mathrm{d}x$

多重積分に関しては、以下のコマンドがあります。

---

[23] 文字様式については、第 11.1節参照。

コマンド	出力
\iint	$\iint$
\oiint	$\oiint$
\sqiint	$\sqiint$

コマンド	出力
\iiint	$\iiint$
\iiiint	$\iiiint$
\dotsint	$\int\cdots\int$

10.2 演算子の範囲

範囲は、上付き文字と下付き文字とで作成することができます。

コマンド	出力
\prod^\infty␣→␣0→A(x)	$\prod_0^\infty A(x)$

行内数式では、範囲は演算子の右横に表示されます。別行建て数式での範囲は、積分範囲を除き、演算子の上と下に表示されます。

範囲が演算子の横に表示されるように強制するには、カーソルを当該演算子の直後において、編集▷数式▷範囲の表記を変更メニューで**行内形式**（短絡キー Alt+M L）を選択することで範囲形式を変更することができます。以下はその用例です。

既定の範囲形式は、以下のようになっています。

$$\sum_{x=0}^{\infty} \frac{1}{x^2}$$

以下は、範囲形式を**行内形式**に変更したときの表示です。

$$\sum\nolimits_{x=0}^{\infty} \frac{1}{x^2}$$

\intop や **\ointop** などのように **op** で終わるもの以外の積分記号では、範囲は、既定で演算子の横に設定されます。しかし、多重積分においては、範囲を演算子の下に置くべきときがあります。このことから、以下の例では、範囲形式を**別行建て形式**にして積分記号の下に置くようにしています。

$$\iiint\limits_{V} X \,\mathrm{d}V = U \tag{5}$$

範囲に条件を指定したい場合には、**\subarray** コマンドや **\substack** コマンドを使用します。たとえば、以下の表記

$$\sum_{\substack{0<k<1000\\ k\in\mathbb{N}}}^{n} k^{-2} \tag{6}$$

を作成するには、以下のようにしなくてはなりません。
まず、**\sum^n␣␣**というコマンドを入力します。すると、和演算子の下の青枠に移動するので、ここに **\subarray**␣コマンドを挿入します。すると、青枠が紫枠の中に入って、ここに複数の行を書き込むことができるようになります。新しい行は、改行（Ctrl+Return）を挿入することで作ることができます。ここに

0<k<1000 Ctrl+Return
と入力すると、新規行のための新しい枠が現れます。

各行の揃え方は、表ツールバーか編集▷行と列メニューで変更することができますが、右揃えにするには、行頭に **\hfill␣**を挿入しなくてはなりません。

\substack コマンドは、各行がつねに中央揃えになることを除いては、**\subarray** と同じです。

演算子の後に来る文字は、範囲の横に来るので、(6) 式のように、演算子の横の余白が大きくなりすぎることがあります。

これを避けるには、LaTeX プリアンブルに以下のようなマクロを指定する方法があります。

```
\def\clap#1{\hbox to 0pt{\hss #1\hss}}
\def\mathclap {\mathpalette \mathclapinternal}
\def\mathclapinternal #1#2{\clap{$\mathsurround =0pt #1{#2}$}}
```

これは、範囲の幅を 0 pt に設定する **\mathclap** コマンドを定義しています。このコマンドの書式は、

\mathclap{ 範囲 }

となっていて、「範囲」には複数の条件を入れることができます。

これを (6) 式に応用すると、以下のようなコマンド

\sum␣\mathclap{\substack␣0<k<1000 Ctrl+Return

を使用して下限を作成することになります。これによって、足される要素は、和演算子の直後に来ることになります。

$$\sum_{\substack{0<k<1000\\ k\in\mathbb{N}}} k^{-2}$$

一つの範囲を複数の演算子に用いる方法が、第 10.4節に述べられています。

10.3 二項演算子

二項演算子は、前後に文字がある場合、周囲に余白が入ります。

コマンド	出力
+	+
-	−
\pm	±
\mp	∓
\cdot	·
\times	×
\div	÷
*	*
\star	⋆
\circ	∘
\diamond	⋄
\bullet	•

コマンド	出力
\nabla	∇
\bigtriangledown	▽
\bigtriangleup	△
\Box	□
\cap	∩
\cup	∪
\dagger	†
\ddagger	‡
\wr	≀
\bigcirc	◯
\wedge	∧
\vee	∨

コマンド	出力
\oplus	⊕
\ominus	⊖
\otimes	⊗
\oslash	⊘
\odot	⊙
\amalg	⨿
\uplus	⊎
\setminus	\
\sqcap	⊓
\sqcup	⊔
\triangleleft	◁
\triangleright	▷

二項演算子は、すべて数式ツールバーボタンの$\pm$から挿入することもできます。

ラプラス演算子を組版するには、**\bigtriangleup** 以外に、**\Delta** や **\nablaˆ2**（∇^2）を使用することもできます。

挿入▷特殊文字メニューのメニュー区切りで入力される文字は、**\triangleright** 演算子です。

10.4 自己定義演算子

LaTeX プリアンブルで **\DeclareMathOperator** コマンドを使用すると、自製演算子を定義することができます。このコマンドの書式は

\DeclareMathOperator{ 新規コマンド }{ 表示 }

です。「表示」は、出力での演算子の表示され方を定義する文字や記号です。大演算子を定義するには、コマンドの後に「*」を置きます。自己定義の大演算子は、すべて第 10.2節で述べられた範囲を指定することができます。

たとえば、以下のような LaTeX プリアンブル行

\DeclareMathOperator*{\Lozenge}{\blacklozenge}

は、第 13.2節にある菱形記号を使った大演算子を挿入する、以下のようなコマンド **\Lozenge** を定義します。

$$\Lozenge_{n=1}^{\infty}$$

上記の数式を作るコマンドは、**\Lozengeˆ\infty→_n=1** です。

自己定義演算子を、同一文書内で複数回用いない時には、以下の書式を持つ **\mathop** コマンドおよび **\mathbin** コマンドを用いて定義を行うこともできます。

（書式）**\mathop{ 表示 }** および **\mathbin{ 表示 }**

\mathop は大演算子を定義し、**\mathbin** は二項演算子を定義します。

たとえば **\mathop** は、以下のように、複数の演算子に共通の範囲指定を行うのに用いることができます。

$$\mathop{\sum\negmedspace\sum}_{i,j=1}^{N}$$

上記の数式では
\mathop{\sum\negmedspace\sum␣**→^N**␣**i,j=1**
というコマンドを用いています。

11 書体

11.1 書体様式

数式中のラテン文字は、以下の書体様式のうちいずれかに設定することができます。

コマンド	出力	短絡キー
\mathbb␣ABC	$\mathbb{ABC}$	Alt+C C
\mathbf␣AbC	$\mathbf{AbC}$	Ctrl+B
\boldsymbol␣AbC	$\boldsymbol{AbC}$	Ctrl+Alt+B, Alt+C B
\mathcal␣ABC	$\mathcal{ABC}$	Ctrl+E
\mathfrak␣AbC	$\mathfrak{AbC}$	-
\mathscr␣AbC	$\mathscr{AC}$	-

コマンド	出力	短絡キー
\mathit␣AbC	AbC	-
\mathrm␣AbC	AbC	Alt+C R
\mathsf␣AbC	AbC	Alt+C S
\mathtt␣AbC	AbC	Ctrl+Shift+P

（注意）\mathbb 様式と \mathcal 様式は、大文字にのみ使用することができます。

既定では、**\mathnormal** 様式に設定されています。

書体様式コマンドは、以下のように数式構成要素内の文字に対しても機能します。

$$\mathfrak{A} = \frac{\mathfrak{b}}{\mathfrak{C}}$$

数式テキストに含まれる文字に対しては、数式書体様式は反映せず、**\textrm** 様式で表示されます。数式テキストの様式を文字様式ダイアログで設定することができないのは、LyX のバグです[24]。

書体様式コマンドの代わりに、編集▷数学▷文字様式ダイアログや、  を使用することもできます。

[24] LyX-bug #4629

11.2 ボールド体の数式

数式全体をボールド体にしようとすると、前節の **\mathbf** コマンドは、ギリシャ文字の小文字に対しては機能しないので、使用することができません。さらにこのコマンドは、以下の式のように、ラテン文字をつねにアップライト体に印字してしまいます。

$$\int_{\mathbf{n}}^{\mathbf{2}} \mathbf{f}(\theta) = \mathbf{\Gamma} \qquad \text{\mathbf を使用した数式}$$

この数式を正しく表示するには、以下のように、**\boldsymbol** コマンドを使用します。

$$\int_{\boldsymbol{n}}^{\boldsymbol{2}} \boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{\Gamma} \qquad \text{\boldsymbol を使用した数式}$$

また、数式を **boldmath 環境**に設定する方法もあります。この環境は、TEX モードで **\boldmath** コマンドを挿入することによって作ることができます。環境を閉じるには、**\unboldmath** コマンドを TEX モードで挿入します。

$$\int_{\boldsymbol{n}}^{\boldsymbol{2}} \boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{\Gamma} \qquad \text{boldmath 環境に置いた数式}$$

11.3 色付きの数式

数式も、通常の本文と同様、色を付けることができます。数式あるいは数式の一部を選択して、文字様式ダイアログを使用して下さい。下記は、マゼンタ色にした数式です。

$$\int A\,\mathrm{d}x = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)}$$

第 9.3節に述べられているように、自製の色を定義することもできます。自製の色は、以下の書式を持つ **\textcolor**TEX コードコマンドで適用することができます。

\textcolor{ 色 }{ 文字ないし数式 }

下記の例は、全体を濃緑にし、一部を赤にしています。

$$\int A\,\mathrm{d}x = \frac{\sqrt[5]{B}}{\ln\left(\frac{1}{3}\right)}$$

LYX のバグのため、自製色は数式全体に対してしか使用することができません[25]。

11.4 書体寸法

数式内の文字については、本文中の文字同様、以下の書体寸法設定コマンドがあります。

[25] LyX-bug #5269

\Huge、**\huge**、**\LARGE**、**\Large**、**\large**、**\normalsize**、**\small**,
\footnotesize、**\scriptsize**、および **\tiny**

これらのコマンドによって生成される実際の書体寸法は、文書の書体寸法に依存し、文書の書体寸法が **\normalsize** コマンドに設定されます。他のコマンドは、**\normalsize** を基準として拡大ないし縮小されます。しかしながら、書体寸法は一定の値を越えることができないようになっています。たとえば、文書書体寸法が 12 pt であるならば、**\Huge** コマンドは **\huge** コマンドと同じ大きさに落とされます。

ある場所以降のすべての数式と本文文字を変更するには、書体寸法コマンドを TEX モードで挿入します。元の書体寸法に戻すには、数式の後に TEX モードで **\normalsize** コマンドを挿入します。

数式内では、以下の寸法コマンドを使用して、寸法を変更することができます。

コマンド	出力
\displaystyle	$E_{\mathrm{pot}_1} = \dfrac{K}{l + \frac{m}{n_2}}$
\textstyle	$E_{\mathrm{pot}_1} = \frac{K}{l+\frac{m}{n_2}}$
\scriptstyle	$E_{\mathrm{pot}_1}=\frac{K}{l+\frac{m}{n_2}}$
\scriptscriptstyle	$E_{\mathrm{pot}_1}=\frac{K}{l+\frac{m}{n_2}}$

これらのコマンドを入力すると、青いボックスが現れるので、そこに数式のパーツを入れることができます。

フォント寸法を変更するにはもう一つの方法がありますが、これは記号と数式内テキストのみに使うことができます。これを使うには、書体寸法コマンドを数式テキスト内に挿入します。数式テキストの終わりか、別の書体寸法コマンドが現れるまでの文字すべてが、選択した寸法になります。以下に二つの例を挙げます。

$$A = \frac{B}{c} \cdot \maltese$$

$$\maltese A \maltese A_{\maltese A}$$

二つの式の前には、**\huge** コマンドが挿入されています。二つ目の数式を入力するコマンドは、
\maltese␣A Alt+M M \Large␣\maltese␣\textit␣A→→
　　Alt+M M \tiny␣\maltese␣\textit␣A
のようになります。

ある記号を別の寸法で表示することができないときには、その記号はつねに既定寸法で表示されます。

12 ギリシャ文字

すべてのギリシャ文字は、ツールバーボタンのαからでも挿入することができます。各国の組版規則では、数式内のギリシャ文字はどれもイタリック体か斜体で組版されなくてはならないことになっていますが、フランス語やロシア語などいくつかの言語では、それにもかかわらず立体で組版されることがあります。

12.1 小文字

コマンド	出力
\alpha	α
\beta	β
\gamma	γ
\delta	δ
\epsilon	ϵ
\varepsilon	ε
\zeta	ζ
\eta	η
\theta	θ
\vartheta	ϑ

コマンド	出力
\iota	ι
\kappa	κ
\varkappa	$\varkappa$
\lambda	λ
\mu	μ
\nu	ν
\xi	ξ
o	o
\pi	π
\varpi	ϖ
\rho	ρ

コマンド	出力
\varrho	ϱ
\sigma	σ
\varsigma	ς
\tau	τ
\upsilon	υ
\phi	ϕ
\varphi	φ
\chi	χ
\psi	ψ
\omega	ω

アップライト体のギリシャ文字を作成する方法は、第 23.9節に説明されています。

12.2 大文字

コマンド	出力
\Gamma	Γ
\Delta	Δ
\Theta	Θ
\Lambda	Λ
\Xi	Ξ
\Pi	Π

コマンド	出力
\Sigma	Σ
\Upsilon	Υ
\Phi	Φ
\Psi	Ψ
\Omega	Ω

大文字のギリシャ文字が立体で表示されるのは、TEX の開発途上に生じたデザイン上のバグによるものです。正しいイタリック体の大文字を得るためには、各コマンドの頭に **var** を付けてください。たとえば、**\varGamma** コマンドは、$\mathit{\Gamma}$ を生成します。もう一つの方法は、パッケージ **fixmath**[26] を LATEX プリアンブル行に

\usepackage{fixmath}

と書いて読み込む方法です。すると、文書中の大きなギリシャ文字は、すべて自動的にイタリック体として組版されます。

[26] **fixmath** は、LATEX パッケージ **was** に含まれています。

12.3 ボールド体

ギリシャ文字は、ラテン文字のようには、多様な書体様式に設定することができません。ギリシャ文字をボールド体にできるのは、**\boldsymbol** コマンドのみです。

コマンド	出力
\Upsilon\boldsymbol\Upsilon	$\Upsilon\boldsymbol{\Upsilon}$
\theta\boldsymbol\theta	$\theta\boldsymbol{\theta}$

13 記号[27]

本節に掲げてある各記号の多くは、ツールバーボタンの∇や$\digamma$でも挿入することができます。

13.1 数学記号

コマンド	出力
\neg	$\neg$
\Im	$\Im$
\Re	$\Re$
\aleph	$\aleph$
\partial	∂
\infty	∞
\wp	$\wp$
\imath	$\imath$
\jmath	$\jmath$

コマンド	出力
\forall	$\forall$
\exists	$\exists$
\nexists	$\nexists$
\emptyset	$\emptyset$
\varnothing	$\varnothing$
\dag	$\dagger$
\ddag	$\ddagger$
\complement	$\complement$
\Bbbk	$\Bbbk$

コマンド	出力
\prime	$\prime$
\backprime	$\backprime$
\mho	$\mho$
\triangle	$\triangle$
\angle	$\angle$
\measuredangle	$\measuredangle$
\sphericalangle	$\sphericalangle$
\top	$\top$
\bot	$\bot$

13.2 その他の記号

コマンド	出力	コマンド	出力	コマンド	出力
\flat	$\flat$	\hbar	$\hbar$	\diamondsuit	$\diamondsuit$
\natural	$\natural$	\hslash	$\hbar$	\Diamond	$\Diamond$
\sharp	$\sharp$	\clubsuit	$\clubsuit$	\heartsuit	$\heartsuit$
\surd	$\surd$	\spadesuit	$\spadesuit$	\P	¶
\checkmark	✓	\bigstar	★	\copyright	©
\yen	¥	\blacklozenge	⧫	\circledR	®
\pounds	£	\blacktriangle	▲	\maltese	✠
\$	$	\blacktiangledown	▼	\diagup	／
\S	§	\bullet	•	\diagdown	＼

ここにある以上の記号が、第 16.4節に挙げてあります。

寸法を変えて表示することのできる記号もあります。第 11.4節をご参照下さい。

[27] 各 LaTeX パッケージに含まれる全記号をほとんど網羅した一覧が、[4] にあります。

13.3 ユーロ通貨記号€

ユーロ通貨記号を数式で使用するには、LaTeX パッケージ **eurosym** が導入されていて、以下のような LaTeX プリアンブル行によって読み込まれていなくてはなりません。

\usepackage[gennarrow]{eurosym}

すると、ユーロ通貨記号を **\euro** コマンドで挿入することができるようになります。

数式テキストには、**eurosym** が導入されていなくても、ユーロ通貨記号を直接€キーを使って挿入することができます。**eurosym** が導入されていれば、**\euro** は TeX モードでも挿入することができます。また、正式な通貨記号を **\officialeuro** コマンド（これは TeX モードでのみ使用することができます）で挿入することができます。

以下は、各ユーロ通貨記号のまとめです。

	コマンド	出力
数式	\euro	€
数式テキスト	€	€
TeX モード	\officialeuro	€

14 関係子

関係子はすべて、ツールバーボタンの≤でも挿入することができます。

コマンド	出力
<	$<$
\le	$\leq$
\ll	$\ll$
\prec	$\prec$
\preceq	$\preceq$
\subset	$\subset$
\subseteq	$\subseteq$
\sqsubseteq	$\sqsubseteq$
\in	$\in$
\vdash	$\vdash$
\smile	$\smile$
\lhd	$\lhd$
\unlhd	$\unlhd$
\gtrless	$\gtrless$
\mid	$\mid$
\nmid	$\nmid$

コマンド	出力
=	$=$
\not=	$\neq$
\equiv	$\equiv$
\sim	$\sim$
\simeq	$\simeq$
\approx	$\approx$
\cong	$\cong$
\bowtie	$\bowtie$
\notin	$\notin$
\perp	$\perp$
\propto	$\propto$
\asymp	$\asymp$
\doteq	$\doteq$
\circeq	$\circeq$
\models	$\models$
\widehat=	$\widehat{=}$

コマンド	出力
>	$>$
\ge	$\geq$
\gg	$\gg$
\succ	$\succ$
\succeq	$\succeq$
\supset	$\supset$
\supseteq	$\supseteq$
\sqsupseteq	$\sqsupseteq$
\ni	$\ni$
\dashv	$\dashv$
\frown	$\frown$
\rhd	$\rhd$
\unrhd	$\unrhd$
\lessgtr	$\lessgtr$
\parallel	$\parallel$
\nparallel	$\nparallel$

\lhd と **\rhd** の文字は、同じように見える演算子 **\triangleleft** および **\triangleright** よりも大きくなっています。

関係子は、記号とは違って、つねに前後に余白が置かれます。

\stackrel コマンドを使うと、以下のように、ラベル付きの関係子を作ることができます。

コマンド	出力
A(r)\stackrel␣r\to\infty␣↓\approx␣␣B	$A(r) \overset{r\to\infty}{\approx} B$

15 函数

15.1 定義済み函数

一般的に、数式表現では変数はイタリック体に設定されますが、函数名はイタリック体にしません。なぜなら、sin は $s \cdot i \cdot n$ であるかのように誤解させる恐れがあるためです。そのために、定義済み函数が存在し、これらは先行する要素よりも少し離れて配置されます。定義済み函数は、函数名の前にバックスラッシュを加えたコマンドとして挿入します。

コマンド	出力	コマンド	出力
Asin(x)+B	$Asin(x) + B$	A\sin(x)+B	$A \sin(x) + B$

以下の函数が定義済みです。

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

上記は、数式ツールバーボタンの $\overset{\exp}{\tan}$ でも挿入することができます。

15.2 自己定義函数

たとえば符号函数 sgn(x) のように、定義済みでない函数を使うには、二つの方法があります。

- 以下の行を LaTeX プリアンブルに加えることによって函数を定義します。[28]

  \DeclareMathOperator{\sgn}{sgn}

  これによって、新しく定義された函数を **\sgn** コマンドで呼び出すことができるようになります。

[28] **\DeclareMathOperator** についての詳細は、第 10.4節をご参照下さい。

- 数式を普通に書き下し、函数名を選択して（上記の例では *sgn* の文字）、それを数式テキストに変更します。最後に、空白を先行する要素と函数の間に入れます。

双方とも定義済み函数と同等の出力をもたらします[29]。

コマンド	出力
A\sgn(x)+B	$A\,\mathrm{sgn}(x) + B$
A\,$\underbrace{\text{sgn}}_{\text{Alt+M M}}$(x)+B	$A\,\mathrm{sgn}(x) + B$

自己定義函数を何回か使用する場合には、一番目の方法の方が適切です。

15.3 極限

極限用には、**\lim・\liminf・\limsup** の他に、以下の函数があります。

コマンド	出力
\varliminf	$\underline{\lim}$
\varlimsup	$\overline{\lim}$
\varprojlim	$\varprojlim$
\varinjlim	$\varinjlim$

極限は、下付き文字を挿入することによって示されます。行内数式では、極限は、以下のように函数の横に置かれます。

コマンド	出力
\lim_x\to_A_x=B	$\lim_{x\to A} x = B$

別行建て数式では、極限は、以下のように通常どおり下に置かれます。

$$\lim_{x\to A} x = B$$

15.4 剰余函数

剰余函数は、特別に 4 つの派生型があります。

以下は、別行建て数式での派生型です。

コマンド	出力
a\mod_b	$a \mod b$
a\pmod_b	$a \pmod b$
a\bmod_b	$a \bmod b$
a\pod_b	$a \pod b$

行内数式では、すべての派生型で、関数名の前の余白がすこし小さく設定されます。

[29] LaTeX 上では、自己定義函数は赤で表示され、定義済み函数は黒で表示されます。

16 特殊文字

16.1 数式テキストにおける特殊文字

以下の各コマンドは、数式テキストか TEX モード中でのみ使用することができます。

コマンド	出力
\oe	œ
\OE	Œ
\ae	æ
\AE	Æ
\aa	å
\AA	Å
\i	ı

コマンド	出力
\o	ø
\O	Ø
\l	ł
\L	Ł
!‘␣	¡
?‘␣	¿
\j	ȷ

Å と Ø の各文字は、数式ツールバーボタンの⌜からも挿入することができます。

例外は、**!‘**と**?‘**の各コマンドで、これらは直接 LYX 中の本文に入れることができます。

16.2 文章中のアクセント

以下に挙げる各コマンドを使えば、すべての文字にアクセントを付けることができます。これらのコマンドは、TEX モードで入れなくてはなりません。

コマンド	出力
\“e	ë
\‘e	è
\^␣e	ê
\=e	ē
\u␣e	ĕ
\b␣e	e̲
\t␣ee	e͡e

コマンド	出力
\H␣e	e̋
\’e	é
\~e	ẽ
\.e	ė
\v␣e	ě
\d␣e	ẹ
\c␣e	ȩ

\t コマンドは異なる二つの文字にアクセントを付けることもできます。たとえば、コマンド **\t␣sz** は、s͡z となります。

‘ ・ ’ ・ ^ の各アクセントは、TEX モードを使わなくても、母音といっしょに直接キーボードから入力することもできます。チルダ[30] ~を、*a* や *n* や *o* といっしょに使うときも同様です。

\b・\c・\d・\H・\t・\u・\v の各コマンドと、キーボードから直接挿入するアクセントは、数式テキスト中でも使うことができます。他のアクセントについては、数式内向けの特別な数式コマンドがあります。第 7.1節をご参照下さい。

さらに、**\textcircled** コマンドを使えば、著作権マークのように、あらゆる数字や文字を丸で囲む—敢えて言えば、丸囲みでアクセントを付けるようなものといえるでしょう—ことができます。

[30] これは、チルダがアクセントとして定義されているキーボードのみに適用されます。

コマンド	出力
\textcircled{w}	ⓦ
\Large \textcircled{\normalsize\protect\raisebox{-1.5pt}{W}}	Ⓦ

ここではユーザーが、文字が丸のなかに収まるように調整してやらなくてはなりません。ここでは、**\Large**[31]で丸の大きさを指定しています。そして **\raisebox**[32]を使って、文字が真ん中にくるようにしています。

16.3 小数字

小数字（minuscule number）は、**\oldstylenums** コマンドで作成することができます。このコマンドは、数式中でも TEX モード中でも使うことができます。コマンド書式は、

\oldstylenums{ 数字 }

です。**\oldstylenums{0123456789}** というコマンドは、0123456789 のようになります。

16.4 他の特殊文字

以下の各文字は、数式中でコマンドを使用してのみ、挿入することができます。

コマンド	出力
\^␣	^
\-	‐
^␣\circ	°

しかしながら、角度記号「°」は、以下の行を LATEX プリアンブルに書き加えれば、直接挿入することができます[33]。

\DeclareInputtext{176}{\ifmmode^\circ\else\textdegree\fi}

17 数式様式

- 以下の二つの配置様式があります。

  中央揃え 事前に定義された標準です。

  行頭下げ これを使うには、文書▷設定メニューの文書クラスにおいて、**fleqn** オプションを指定しておかなくてはなりません。

[31]第 11.4節参照のこと。

[32]第 9.2節参照のこと。

[33]この件に関する詳細は、第 23.10節にあります。

行頭下げを用いる場合には、行頭下げの大きさを **\mathindent** の値で調整することができます。これを 15 mm にするには、LaTeX プリアンブルに以下のコマンドを入れておきます。

\setlength{\mathindent}{15mm}

\mathindent を明示的に指定しない場合には、事前に定義されている 30 pt が適用されます。

- また、以下の二つの連番様式があります。

  右 事前に定義された標準です。

  左 これを使うには、文書▷設定メニューの文書クラスにおいて、**leqno** オプションを指定しておかなくてはなりません。

fleqn と **leqno** は、いっしょに指定することができます。両方のオプションを同時に入れる場合には、コンマで区切って下さい。

これで選択した様式は、文書中のすべての別行建て数式に用いられます。もし、中央揃えと行頭下げの両様式を同一文書中で用いたい場合には、**中央揃え**様式を指定するようにします。そして、行頭下げにしたい数式は、flalign 環境に指定するようにします。第 18.2.3節をご覧下さい。

18 多行数式

18.1 概要

LyX では、多行数式は、数式中で **Ctrl+Return** を押すことで作られます。この操作によって、第 18.3節に述べられている **eqnarray 環境**が作り出されるか、あるいは文書設定で AMS math パッケージを使うオプションが選択されている場合には、第 18.2.1節に述べられている **align 環境**が作り出されることになります。

他にも、挿入▷数式メニューで作ることのできる多行数式環境があります。これらの環境は、以下の各節で説明します。

すべての多行数式において、新規行は、Ctrl+Return を押すことによって作られます。行を足したり削ったりするには、数式ツールバーボタンの  や  を使うか、編集▷行と列メニューを使うことができます。

18.1.1 行間

以下のように、多行数式において行のあいだの空白が足りないことが、ときどき起こります。

$$\begin{aligned} B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + 4x_0^2x^2 + 4x_0xD &= -4x^2B^2 + 4x_0xB^2 \\ 4x^2\left(B^2 + x_0^2\right) + 4x_0x\left(D - B^2\right) + B^2\left(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2\right) &= 0 \end{aligned}$$

LaTeX において行間を付け加えるには、新規行コマンドに非必須の引数をとらせて指定しますが、これはまだ LyX には実装されていない[34]ので、数式全体を TeX モードで入れなくてはなりませ

[34] LyX-bug #1505を参照。

ん。上記の例の行間を大きくするには、最初の行の最後に \\[**3mm**] というコマンドを入れます。すると、次のようになります。

$$\begin{aligned} B^2(B^2-2r_g^2+2x_0^2-2r_k^2)+4x_0^2x^2+4x_0xD &= \text{-}4x^2B^2+4x_0xB^2 \\ 4x^2\left(B^2+x_0^2\right)+4x_0x\left(D-B^2\right)+B^2\left(B^2-2r_g^2+2x_0^2-2r_k^2\right) &= 0 \end{aligned}$$

同一数式内のすべての行の行間を一律に指定するには、**\jot** 変数を変更します。定義は、行間 = 6 pt + \jot となっています。**\jot** の既定値は、3 pt です。上記の例のように、行間を 3 mm 追加するには、

\setlength{\jot}{3mm+3pt}

というコマンドを、数式直前に TEX モードで入れておきます。これを行うには、LATEX プリアンブルに

\usepackage{calc}

という行を入れて、LATEX パッケージ **calc**[35] を読み込んでおく必要があります。すると、

$$\begin{aligned} B^2(B^2-2r_g^2+2x_0^2-2r_k^2)+4x_0^2x^2+4x_0xD &= \text{-}4x^2B^2+4x_0xB^2 \\ 4x^2\left(B^2+x_0^2\right)+4x_0x\left(D-B^2\right)+B^2\left(B^2-2r_g^2+2x_0^2-2r_k^2\right) &= 0 \end{aligned}$$

のような結果を得ます。行間を既定値に戻すには、**\jot** をふたたび 3 pt に戻します。

18.1.2 列間

多行数式は、行列を形成します。たとえば、eqnarray 環境の数式は、3 列からなる行列です。この環境で列間を変更すれば、関係子周辺の余白を変更することができます。

列間は、**\arraycolsep** 変数を使って指定し、
列間 = 2 **\arraycolsep**
という関係があります。したがって、

\setlength{\arraycolsep}{1cm}

というコマンドを TEX モードで入れると、ここから後のすべての数式の列間が 2 cm になります。これを既定値に戻すには、**\arraycolsep** を 5 pt に戻して下さい。

以下は、2 cm の列間を持つ数式です。

$$\begin{array}{ccc} A & = & B \\ C & \neq & A \end{array}$$

行列の既定の列間 10 pt を持つ数式です。

$$\begin{array}{ccc} A & = & B \\ C & \neq & A \end{array}$$

[35] **calc** は標準的な LATEX 頒布版のすべてに付属しています。

18.1.3 長い数式

長い数式は、以下の方法を使って組版することができます。

- 左辺ないし右辺の一方が、行幅よりもかなり短いときには、以下のように、短い方を左辺に置き、右辺を二行に分けて組版します。

$$\begin{aligned} H &= W_{SB} + W_{mv} + W_D - \frac{\hbar^2}{2m_0}\Delta - \frac{\hbar^2}{2m_1}\Delta_1 - \frac{\hbar^2}{2m_2}\Delta_2 - \frac{e^2}{4\pi\varepsilon_0|\mathbf{r}-\mathbf{R}_1|} \\ &\quad - \frac{e^2}{4\pi\varepsilon_0|\mathbf{r}-\mathbf{R}_2|} + \frac{e^2}{4\pi\varepsilon_0|\mathbf{R}_1-\mathbf{R}_2|} \end{aligned} \tag{7}$$

  二行目の最初のマイナス記号は、行頭の文字になってしまうため、通常、表示の上で演算子としては取り扱われません。前後に余白が置かれることもなく、分数線からも離れて表示されません。これを避けるために、マイナス記号の後に **\hspace**.[36]コマンドを使って 3 pt 空白を入れてあります。

- 数式の両辺がともに長すぎるときには、**\lefteqn** コマンドを使います。これを最初の行の第一列に入れると、以下のように、続きの内容が他の列にかかって表示されます。

$$\begin{aligned} &4x^2\left(B^2+x_0^2\right) + 4x_0x\left(D-B^2\right) + B^2\left(B^2-2r_g^2+2x_0^2-2r_k^2\right) + D^2 \\ &\quad - B^2 - 2B\sqrt{r_g^2-x^2+2x_0x-x_0^2} + r_g^2 - x^2 + 2x_0x - x_0^2 \\ &\quad = B^2 + 2\left(r_g^2+2x_0x-x_0^2-r_k^2\right) + \frac{\left(r_g^2+2x_0x-x_0^2-r_k^2\right)^2}{B^2} \end{aligned} \tag{8}$$

  \lefteqn を入力すると、青枠から少し左にずれたところに現れる紫枠にカーソルが移るので、ここに数式を入力します。
  二行め以降の行の内容は、二列め以降の列に挿入します。挿入する列が右になるほど、字下げの量が大きくなります。

  \lefteqn を使用する際には、以下のことにご注意下さい。

  - 数式では、ページ幅全部を使うことはしません。たとえば、上記の例で、最初の行に $-B^2$ という項を置いたとすると、ページ余白の領域に出てしまいますが、これはよくありません。幅をうまく使うには、最初の行の行頭に負の空白を入れる方法もあります。
  - LyX のバグによって、最初の行にマウスでカーソルを入れることはできません[37]。カーソルを行頭に合わせて、矢印キーで移動するしかありません。

- 長い数式を組む他の方法として、第 18.5節と第 18.6節で述べられている環境を用いる方法があります。

[36] **\hspace** に関しての詳細は、第 8.2節をご覧下さい。

[37] LyX-bug #1429

18.1.4 多行にわたる括弧

多行にわたる括弧を作ろうとすると、以下のような問題が生じます。

$$\begin{aligned} A &= \sin(x)\left[\prod_{R=1}^{\infty}\frac{1}{R}+\cdots \right. \\ &\qquad \left. \cdots + B - D\right] \end{aligned}$$

可変寸法の括弧は多行にわたることができないので、閉じ括弧が初めの括弧よりも小さくなってしまっています。

二行めの括弧の大きさを正しく設定するには、最初の行の終わりを **\right.** とし、二行めの始めを **\left.**[38]とします。一行めにおいては、範囲付き積演算子がもっとも大きな記号であり、これが二行めの括弧の大きさにならなくてはならないので、**\left.** の後に、**\vphantom␣\prod^␣\infty␣↓␣R=1}** というコマンドを挿入します。

その結果が以下の数式です。

$$\begin{aligned} A &= \sin(x)\left[\prod_{R=1}^{\infty}\frac{1}{R}+\cdots \right. \\ &\qquad \left. \cdots + B - D\right] \end{aligned}$$

18.2 align 環境

align 環境は、すべての型の多行数式に使うことができ、とくに、いくつかの数式を並べて表示させるのに便利です。

align 環境には列があり、奇数列は右揃え、偶数列は左揃えに設定されます。align 環境の行にはすべて、付番することができます。

align 環境は、挿入▷数式メニューから作ることができます。編集▷数式▷数式の表記を変更メニューを使えば、既存の数式を align 環境に変更することができます。

列を追加したり削除したりするには、数式ツールバーボタンの  や  を使うか、編集▷行と列メニューを使います。

18.2.1 標準 align 環境

この align 環境は、数式中で Ctrl+Return を押すか、挿入▷数式▷AMS align 環境メニューで作ることができます。

以下は、二つの数式を横に並べた例ですが、これは 4 列からなる align 環境として作ります。

$$\begin{aligned} A &= \sin(B) & C &= D \\ C &\neq A & B &\neq D \end{aligned}$$

[38] **\left** と **\right** に関する詳細は、第 5.1.2節をご覧下さい。

ご覧になって分かるように、この環境の数式は、一列めの前と偶数列の後に \hfill[39]があるかのように配置されます。数式様式として**行頭下げ**[40]を使う場合には、第一列の前の \hfill はない形で数式が設定されます。

18.2.2 alignat 環境

alignat 環境には、事前に設定された列間が存在しません。列間は、必要ならば、第 8節に述べられている空白を使用して手動で入れます。

以下は、上記の例を alignat 環境に設定し、二つめの数式の頭に 1 cm の空白を入れたものです。

$$A = \sin(B) \qquad C = D$$
$$C \neq A \qquad B \neq D$$

列間を各列ごとに設定することができるので、この環境は、とくに三つないし四つの数式を横に並べるのに向いています。

18.2.3 flalign 環境

この環境では、つねに、最初の二列をできるだけ左寄せにし、最後の二列をできるだけ右寄せにするように設定されます。以下がその例です。

$A = 1$ $B = 2$ $C = 3$

$X =$ -1 $Y =$ -2 $Z = 4$

奇数列の flalign 環境を作成し、最後の列に空の TEX 括弧を入れておくと、数式様式として**中央揃え**が用いられているときでも、一部の数式を左寄せにすることができます。以下は、例として (5) 式を行頭下げにしたものです。

$$\iiint_V X \,\mathrm{d}V = U \tag{9}$$

ここで、最初の二列には数式が入れられており、行頭下げ数式様式と同等の字下げを行うために、30 pt の空白が第 1 列の頭に入れてあります。

18.3 eqnarray 環境

この環境を作成すると、三つの青枠が現れます。最初の枠の内容は右寄せに設定され、最後の枠の内容は左寄せに設定されます。中央の枠は、関係子のみを入れることを想定しているので、その内容は中央揃えで少し小さく設定されます。

$$\begin{array}{rcl} \dfrac{ABC}{D} & \dfrac{ABC}{D} & \dfrac{ABC}{D} \\ AB & AB & AB \\ A & = & A \end{array}$$

[39] \hfill に関する詳細は、第 8.2節をご覧下さい。

[40] **数式様式については、第** 17節をご覧下さい。

18.4 gather 環境

この環境には、中央揃えに設定された列一つしかありません。行はすべて付番することができます。

$$A = 1 \tag{10}$$

$$X = \text{-}1 \tag{11}$$

18.5 multline 環境

gather 環境同様、multline 環境には、中央揃えに設定された列一つしかありません。ただし、一行めが左揃えに設定され、最終行が右揃えに設定されるのです。他の行はすべて中央揃えになります。このことから、この環境は、長い数式に使うのに向いています。用例として、(8) 式を multline 環境に置いたものを示します。

$$\begin{aligned} 4x^2\left(B^2+x_0^2\right)+4x_0x\left(D-B^2\right)+B^2\left(B^2-2r_g^2+2x_0^2-2r_k^2\right)+D^2 \\ -B^2-2B\sqrt{r_g^2-x^2+2x_0x-x_0^2}+r_g^2-x^2+2x_0x-x_0^2 \\ =B^2+2\left(r_g^2+2x_0x-x_0^2-r_k^2\right)+\frac{\left(r_g^2+2x_0x-x_0^2-r_k^2\right)^2}{B^2} \end{aligned} \tag{12}$$

文書の付番設定が右寄せ（左寄せ）になっているときには、出力では、multline 環境の最後（最初）の行だけが付番されます[41]。

\shoveright コマンドや \shoveleft コマンドを使えば、中央揃えの行を右寄せや左寄せにすることができます。これらのコマンドは、以下のようにして使います。

\shoveright{ 行の内容 } あるいは **\shoveleft{ 行の内容 }**

\multlinegap 長は、一行めの左ページ余白からの距離を指定します。既定値は 0 pt の長さです。

以下は、上記の数式に

\setlength{\multlinegap}{2cm}

というコマンドを、TEX モードで直前に挿入した例です。

$$\begin{aligned} 4x^2\left(B^2+x_0^2\right)+4x_0x\left(D-B^2\right)+B^2\left(B^2-2r_g^2+2x_0^2-2r_k^2\right)+D^2 \\ -B^2-2B\sqrt{r_g^2-x^2+2x_0x-x_0^2}+r_g^2-x^2+2x_0x-x_0^2 \\ =B^2+2\left(r_g^2+2x_0x-x_0^2-r_k^2\right)+\frac{\left(r_g^2+2x_0x-x_0^2-r_k^2\right)^2}{B^2} \end{aligned} \tag{13}$$

二行めは、**\shoveleft** を使って左揃えにしています。

[41] 付番様式については、第 17節を参照。

18.6　数式の一部の多行化

数式の一部分のみを多行表示したい場合には、**aligned・alignedat・gathered・split** のうちのいずれかの環境を使用します。これらは、挿入▷数式メニューか、本節で解説している各コマンドを使用して挿入することができます。

最初の三つの環境は、環境名から **ed** を省いた同名の多行数式環境と同じ性格を持ちますが、環境の前後に数式を続けることが可能です。たとえば、

$$\left.\begin{aligned}\Delta x\Delta p &\geq \frac{\hbar}{2}\\ \Delta E\Delta t &\geq \frac{\hbar}{2}\end{aligned}\right\}\text{不確定性原理}$$

この数式を作るには、別行建て数式をまず作っておいて、そこに **\aligned** コマンドを挿入しています。紫枠の中に青枠が現れるので、そこに必要に応じて、列や行を加えていきます。この多行環境の外には、括弧などの他の数式要素を入れることができます。

aligned 環境は、長い数式を水平方向を揃えて表示するのにも向いています。別行建て数式内で aligned を用いるようにすると、数式番号を行末の、数式全高の中心に配置できる利点があります。以下に例として、(7) 式に aligned 環境を適用したものを示します。

$$\begin{aligned}H = W_{SB} + W_{mv} + W_D - \frac{\hbar^2}{2m_0}\Delta - \frac{\hbar^2}{2m_1}\Delta_1 - \frac{\hbar^2}{2m_2}\Delta_2 - \frac{e^2}{4\pi\varepsilon_0|\mathbf{r}-\mathbf{R}_1|}\\ - \frac{e^2}{4\pi\varepsilon_0|\mathbf{r}-\mathbf{R}_2|} + \frac{e^2}{4\pi\varepsilon_0|\mathbf{R}_1-\mathbf{R}_2|}\end{aligned} \tag{14}$$

alignedat・gathered・split の各環境を使うには、それぞれ **\alignedat・\gathered・\split** の各コマンドを挿入します。split 環境は、aligned 環境と同じ性格を持ちますが、二つの列しか作ることができません。

18.7　多行数式中のテキスト

各 align 系環境および multline・gather 環境では、独立した行に列揃えの影響を受けない形でテキストを挿入することができます。これを行うには、以下の書式を持つ **\intertext コマンドを使います。**

\intertext{ テキスト }

テキストのハイフネーションを行うことはできないので、テキストは一行に収めなくてはなりません。LyX は、現時点では **\intertext** を直接サポートしていないので、テキストは数式テキストとして書き入れます。ここで、**\intertext** は行頭になくてはならず、当該行の上に出力されます。以下は、二行めの行頭にテキストを入れた例です。

$$I = a\sqrt{2}\int_0^{2\pi}\sqrt{1+\cos(\phi)}\,\mathrm{d}\phi \tag{15}$$

integrand is symmetric to $\phi = \pi$, therefore

$$= 2a\sqrt{2}\int_0^{\pi}\sqrt{1+\cos(\phi)}\,\mathrm{d}\phi \tag{16}$$

19　数式番号

19.1　概要

付番数式は、挿入▷数式▷付番数式メニュー（短絡キー Ctrl+Alt N）で作ることができます。既存の数式に番号を振るには、編集▷数式▷数式全体を付番メニュー（短絡キー Alt+M N）を使います。LyX 中において数式番号は、数式の後に括弧に囲まれた「#」で表されます。「#」は、実際の出力では数式番号に置き換えられます。

多行数式で付番が有効になっているときには、すべての行に番号が振られます。ただし、編集▷数式▷この行を付番メニュー（短絡キー Alt+M Shift+N）を使用すれば、各行毎に付番するかどうか指定することができます。

行内数式を除いて、すべての数式は二通りの様式で番号を振ることができます。第 17節をご覧下さい。

19.2　相互参照

ラベルを付けた数式は、すべて相互参照することができます。ラベルは、挿入▷ラベルメニューか、ツールバーボタンの  で付けることができます。このとき、カーソルは別行建て数式の中になくてはなりません。すると、テキストフィールドの中に **eq:**という接頭語の入ったダイアログが現れるので、接頭語の後にラベルを挿入します。この既定の接頭辞は「equation（数式）」を意味し、こうして数式ラベルであるとの標を付けることによって、節ラベルなどから区別し、大きな文書の中でラベルを見つけるのを容易にします。ラベルを変更するには、挿入▷ラベルメニューをもういちど使って下さい。

LyX 中でラベル名は、数式の後ろに、二つの括弧に囲まれて表示されます。ラベル付きの数式はつねに付番されます。

相互参照は、挿入▷相互参照メニューかツールバーボタンの  を使って挿入します。数式相互参照は、出力では数式番号として表示されます。相互参照ダイアログで「**(<参照>)**」書式を選択した場合には、出力での相互参照は、括弧に囲まれた数式番号として表示されます。
LyX 中で相互参照を右クリックすると、参照先の数式に移動することができます。

以下は、後の各小節に現れる数式への相互参照を含む例です。

(何とかかんとか) 式と (17b) 式は、等価です。(XXI) 式とは異なり、(W) 式では、付番にラテン数字を使用しています。

\tag[42]の引数が、第 9.4節で述べたボックスを含んでいるときには、その数式を参照することはできません。

[42] **\tag** は、第 19.4節に説明があります。

19.3 細目番号

\begin{subequations} および **\end{subequations}** コマンドを使うと、数式に細目番号を付けることができます。これらのコマンドは、TEX モードで入れます。

たとえば、

$$A = C - B \tag{17}$$

$$B = C - A \tag{17a}$$

$$C = A + B \tag{17b}$$

この例を作るには、次のようにします。

1. 一つめの数式を入力します。
2. 一つめの数式の後に
   \addtocounter{equation}{-1} \begin{subequations}
   を入力します。
3. 二つめの数式を入力します。
4. 三つめの数式を入力します。
5. 三つめの数式の後に、**\end{subequations}** を入力します。

\begin コマンドと **\end** コマンドのあいだの数式はすべて、a・b・c... のように細目番号が振られます。多行数式の場合は、すべての行に細目番号が振られます。細目番号が振られた数式はすべて、**ひとつの**付番数式として扱われますが、それぞれの付番数式が **equation** カウンタを一つずつ進めてしまうので、**\addtocounter** コマンドを使ってカウンタを戻さなくてはなりません。これを怠ると、(17) 式・(17a) 式・(17b) 式は、それぞれ (17) 式・(18a) 式・(18b) 式として番号が振られてしまいます。

上記のように、コマンドを TEX モードで入れると、最初の二つの数式のあいだに空白が生じてしまいます。これを戻すために、-5 mm の垂直空白を **\begin{subequations}** コマンドの後に入れています。数式様式として**行頭下げ**[43]を用いているときには、これを-7 mm 空白にしてください。

以下は、二行めのみ付番を無効にしている多行数式の例です。

$$\begin{aligned} A &= (B-Z)^2 = (B-Z)(B-Z) && \text{(18a)} \\ &= B^2 - ZB - BZ + Z^2 \\ &= B^2 - 2BZ + Z^2 && \text{(18b)} \end{aligned}$$

[43] **数式様式に関しては、第** 17節を参照。

19.4 ユーザー定義番号

標準の付番では、数式番号の周りに括弧が表示されます。括弧をたとえば縦棒に置き換えるには、LaTeX プリアンブルに以下の行を付け加えます。

\def\tagform@#1{\maketag@@@{|#1|}}

他の記号を使いたいときには、**#1** 脇の縦棒を一つないし複数の文字で置き換えて下さい。数式番号だけで良い時は、縦棒を削除して下さい。

数式の後ろに、括弧に囲まれた連番の代わりに、何かしらの表現が欲しいときには、以下のように **\tag** コマンドを使います。

$$A + B = C \tag{何とかかんとか}$$

上記の例では、**\tag␣何とかかんとか**というコマンドを数式に打ち込んでいます。

代わりに **\tag*␣何とかかんとか**というコマンドを使うと、星印は表現の周りの括弧を抑制するので、以下のようになります。

$$A + B = C \tag*{something}$$

数式番号を、文書中の新しい部門や節ごとに振りなおしたいときには、部に関しては
\@addtoreset{equation}{part}
節に関しては
\@addtoreset{equation}{section}
というコマンドを使います。

これらのコマンドを TeX モードで使えるようにするためには、**\makeatletter** コマンドで「@」字を LaTeX 中で「有効」にしてやらなくてはなりません。一方、**\makeatother** コマンドはこれを無効にします。したがって、TeX モード中での上記コマンド列は、

\makeatletter
\@addtoreset{equation}{section}
\makeatother
のようにならなくてはなりません。

LaTeX プリアンブル中では、**\makeatletter** と **\makeatother** は、LyX が内部的に自動で挿入するので省略してかまいません。

\@addtoreset を戻すには、まず LaTeX プリアンブル中に

\usepackage{remreset}

という行を入れて、**remreset.sty**[44] ファイルを読み込んでおかなくてはなりません。その後 **\@removefromreset** コマンドを **\@addtoreset** と同じ書式で使用すると、**\@addtoreset** を戻すことができます。

ときには、数式を

(節番号. 数式番号)

[44] **remreset** は、LaTeX パッケージの **carlisle** の一部として含まれており、LaTeX 標準頒布版には含まれています。

のような形で付番し、節ごとに数式番号を「1」から始めさせなくてはならないときがあります。

このような場合のために、**\numberwithin** というコマンドがあり、

\numberwithin{ カウンタ }{ 節階層 }

という書式で用います。「カウンタ」は、どの番号を制御するかを表し、「節階層」は点の前に何の番号を振るのかを表します。

したがって、ここでは LaTeX プリアンブルか TeX コードで

\numberwithin{equation}{section}

という行を用いることにしましょう。その結果がこれです。

$$A + B = C \tag{19.19}$$

たとえば、部番号を節階層として使用して、表に付番を施すときには
\numberwithin{table}{part} を用います。

標準の付番方式に戻したいときや、この種の付番が文書クラスで定義されているときに、それを止めさせたい場合には、

\renewcommand{\theequation}{\arabic{equation}}
あるいは
\renewcommand{\thetable}{\arabic{table}}

というコマンドを TeX コードとして入れるか、LaTeX プリアンブルに入れます。**\numberwithin** は、内部的に上記で述べた **\@addtoreset** コマンドを使用しているので、これも使用後は戻しておかなくてはなりません。

19.5 ローマ数字や文字を使った付番

数式は、ローマ数字やラテン文字を使って付番することもできます。たとえば、小文字のローマ数字を使って付番するには、数式の前に TeX モードで

\renewcommand{\theequation}{\roman{equation}}

というコマンドを入れます。**\renewcommand** は、定義済みのコマンド **\theequation** をコマンド **\roman{equation}** に再定義します[45]。ここで、**equation** は数式カウンタです。コマンド **\the** をカウンタの接頭辞として使用すると、カウンタの値がアラビア数字として出力されます。数式に番号を振ると、LaTeX は、内部的に **\theequation** コマンドを数式の後ろに置くのです。**\roman{equation}** は、カウンタを小文字のローマ数字として出力します。

こうして、**\renewcommand** コマンド以降の数式はすべて、ローマ数字で付番されるようになります。大文字のローマ数字での付番に切り替えたいときは、同じコマンドの **\roman** の部分を **\Roman** に変えて挿入します。また、小文字ラテン文字を使って「付番」したいときのために、**\alph** コマンドがあり、大文字ラテン文字を使って付番したいときのためには、**\Alph** コマンドがあります。

[45] **\renewcommand** コマンドは、第 22.1節に述べられている **\newcommand** コマンドと同じ書式を持ちます。

（注意）ラテン文字を使うと、一つの文書の中で、最大 26 個の数式しか番号を振ることができません。

$$A = \text{小文字ローマ数字} \tag{xx}$$

$$B = \text{大文字ローマ数字} \tag{XXI}$$

$$C = \text{小文字ラテン文字} \tag{v}$$

$$D = \text{大文字ラテン文字} \tag{W}$$

既定の付番方式に戻すには、以下のコマンドを挿入してください。

\renewcommand{\theequation}{\arabic{equation}}

$$E = \text{アラビア数字} \tag{24}$$

以上からわかるとおり、付番様式の違いに関わらず、数式番号は連番で振られます。様式変更時に「1」から番号が始まるようにするためには、新しい数式カウンタを定義しなくてはなりません。この点に関する説明は、ファイルFormula-numbering.lyxにあります。

20　化学記号と化学式

以下は、化学関係の文章の例です。

> SO_4^{2-} イオンは、2 つの Na^+ イオンと反応して、硫酸化塩 (Na_2SO_4) を形成します。
> この化学式は以下のようになります。
>
> $$2\,Na^+ + SO_4^{2-} \longrightarrow Na_2SO_4 \tag{25}$$

この化学式は、直接数式として作成することができます。記号がイタリック体として表示されることを防ぐには、全体を選択してから短絡キー Alt+C R を押せば、アップライトフォント様式に変更することができます[46]。

化学式を組版するのにもう少し便利な方法は、LaTeX パッケージ **mhchem** が導入されているときに使用することができる **\ce** コマンドを使用することです。**\ce** を数式に入力すると、新しい青いボックスが現れ、直感的に化学式を入力することができます。

[46] フォント様式に関しては、第 11.1節を参照のこと。

コマンド	出力
\ce␣H2CO3	H_2CO_3
\ce␣SO4^2-	SO_4^{2-}
\ce␣(NH4)2S	$(NH_4)_2S$
\ce␣KCr(SO4)2.12H2O	$KCr(SO_4)_2{\cdot}12\,H_2O$
\ce␣A-B\dbond␣C\tbond␣D	A−B=C≡D
\ce␣^227↓␣90→Th+	$^{227}_{90}Th^+$
\ce␣CO2␣+␣C␣¡=¿␣2CO	$CO_2 + C \rightleftharpoons 2\,CO$
\ce␣CO2␣+␣C␣-¿[\alpha][\beta]␣2CO}	$CO_2 + C \xrightarrow[\beta]{\alpha} 2\,CO$

【註】\ce ボックスに数式を入れようとすると、LaTeX エラーが発生します。このような場合には、**\ce{$\mu\hyphen$Cl}**（μ-Cl）のように TeX コードを使わなくてはなりません。

\ce を使うと (25) 式のコマンドは
\ce␣2Na+␣+␣SO4^2-␣␣-¿␣Na2SO4
のようになります。

複数行の化学式を作るには、第 18節に述べられている方法で、多行数式をまず作ります。その後、数式の小さな青いボックスそれぞれに **\ce** コマンドを使用します。(26) 式と (27) 式は、多段化学反応式の例で、一つの式毎に番号が振られています。

$$\mathrm{TEOS} + 4\,O \longrightarrow Si(OH)_4 + 4\,C_2H_4O \tag{26}$$

$$Si(OH)_4 \longrightarrow SiO_2 + 2\,H_2O \tag{27}$$

mhchem パッケージは、**\ce** の他に、特殊ケースに使用する **\cf** コマンドを提供しています。**\cf** の詳しい情報と例示については、**mhchem**[6] の取扱説明書をご覧ください。

21 図解

LyX は、二つの型の可換図 **amscd** および **xymatrix** をサポートしており、以下でこれらの説明をします。

21.1 amscd 図解

この型の図解は、以下のように、関係を縦横の線や矢印で図示します。

$$\begin{array}{ccccc} A & \longrightarrow & B & \longrightarrow & C \\ \uparrow & & & & \downarrow \\ F & \longleftarrow & E & \longleftarrow & D \end{array}$$

これを作るには、数式に **\CD** コマンドを挿入します。二つの点線に囲まれた青枠が現れるので、ここにコマンドを入れていきます。Ctrl+Return を押すと、新しい行が作られます。水平方向の関係は奇数行に入れ、垂直方向の関係は偶数行に入れます。

関係を作るには、以下のコマンドがあります。

- **@ <<<** は左矢印、**@ >>>** は右矢印、**@=**は長い等号を生成します。
- **@AAA** は上矢印、**@VVV** は下矢印、**@|** は縦向きの等号を生成します。
- **@.** は関係が存在しない部分に置きます。

矢印はすべて、以下のようにラベル付けをすることができます。

- 文章を、第 1 と第 2 の「<」ないし「>」のあいだに入れると、この文章は矢印の上に表示されます。第 2・第 3 の「<」ないし「>」のあいだに入れると、矢印の下に表示されます。
- 縦矢印に付ける文章を、第 1・第 2 の「A」ないし「V」のあいだに入れると、この文章は矢印の左に表示されます。第 2・第 3 のもののあいだに入れると、矢印の右に表示されます。文章中に「A」や「V」の文字があるときには、これらは TEX 括弧の中に入れなくてはなりません。

以下は、上記のすべての関係を使った例です。

$$\begin{array}{ccccccc} A & \xrightarrow{j} & B & \xrightarrow[k]{} & C & = & F \\ {\scriptstyle m}\uparrow & & & & \downarrow{\scriptstyle V} & & \| \\ D & \xleftarrow[j]{} & E & \xrightarrow{k} & F & = & C \end{array}$$

これを作るコマンドは、以下のとおりです。

\CD␣A@ >j>>B@ >>k>C@=F Ctrl+Return
@AmAA@.@VV\{V→V@| Ctrl+Return
D@ <<j<E@ >k>>F@=C

21.2 xymatrix 図解

xymatrices を使うには、LaTeX パッケージの **xypic** が導入済みである必要があります。xymatrix は、数式中に **\xymatrix** コマンドを入れることで作ることができます。すると、通常の行列と同じようにして、列や行を付け加えることができます。第 4節をご参照下さい。

amscd 図解とは異なり、xymatrices は、対角矢印や曲がった矢印など多様なサポートをしています。作ることのできる可換図と装飾は、ヘルプ▷用途別説明書▷XY-pic 説明書メニューにある *XY-pic* **説明書**で詳しく網羅しています。

21.3 ファインマン・ダイアグラム

ファインマン・ダイアグラムを使うには、LaTeX パッケージ **feyn** を導入しておかなくてはなりません。すると、ファインマン・ダイアグラムは、数式中で **\Diagram** コマンドを挿入すれば生成されます。通常の行列で行うのと同じようにして新規行や新規列を加えることができます（第4節参照）。

LyX でのファインマン・ダイアグラムの作り方は、メニューヘルプ▷用途別説明書内の『**ファインマン・ダイアグラム**』にあります。

22　ユーザー定義コマンド

（注意）ユーザー定義コマンド名及びマクロ名には、ラテン文字しか使用することができません。

22.1　\newcommand コマンド

頻繁に用いるには、長すぎる LaTeX コマンドはたくさんありますが、**\newcommand** コマンドを使えば、新しい短縮コマンドを定義することが可能です。

\newcommand コマンドの書式は、

\newcommand{ 新コマンド名 }[引数の数][オプションの値]
{ コマンド定義 }

です。

（注意）新コマンド名が、使用中の文書や呼び出している LaTeX パッケージで、既に使用されていないことを確認して下さい。たとえば、**\Leftarrow** の短縮のつもりで **\le** というコマンドを定義したとすると、**\le** は既に「≤」を表すコマンドとして定義されてしまっているので、エラーメッセージが表示されます。

「引数の数」は、0 - 9 の範囲の整数であり、新コマンドがいくつの引数をとるかを指定するものです。「オプションの値」では、非必須の引数の既定値を定義できます。これを指定すると、新コマンドの**最初の**引数は、自動的に非必須の引数になります。

以下にいくつかの例を挙げます。

- **\Longrightarrow** の短縮形として **\gr** というコマンドを定義するには、LaTeX プリアンブルに以下の行を加えます。

  \newcommand{\gr}{\Longrightarrow}

- **\underline** の短縮形として **\us** というコマンドを定義するには、（下線を引くべき文字列を示す）引数を考慮に入れなくてはなりません。このためには、以下のようなプリアンブル行を入れます。

  \newcommand{\us}[1]{\underline{#1}}

  「**#**」という文字は、引数の入る場所を示し、その後ろの「**1**」は、これが第 1 引数の入る場所であることを示します。

- **\framebox** の短縮形として、たとえば **\fb** というコマンドを定義するには、

  \newcommand{\fb}[3]{\framebox#1#2{$#3$}}

  二つのドルマークは、**\framebox** が必要とする内部の数式を作り出します。第 9.1節をご参照下さい。

- ボックスの色を指定する必要がない **\fcolorbox** 用の新コマンドを作るには、以下のように、色を示す引数を非必須として定義します。

\newcommand{\cb}[3][white]{\fcolorbox{#2}{#1}{$#3$}}

\cb を使うときに色が指定されなければ、事前に定義された色である **white** が使用されます。

以下は、上で定義したコマンドの動作テストです。

コマンド	出力
A\gr␣B	$A \Longrightarrow B$
\us{ABcd}	$\underline{ABcd}$
\fb{[2cm]→\{→\{\int␣A=B	$\boxed{\int A = B}$
\cb{red→\{\int␣A=B	$\boxed{\int A = B}$
\cb[green]\{red→\{\int␣A=B	$\boxed{\int A = B}$

22.2 数式マクロ

ユーザー定義コマンドは、複雑な表現を使うときに特に便利です。たとえば、文書中で二次方程式を扱っているとすると、同じような解の形が何度も出てきます。二次方程式の一般型は、

$$0 = \lambda^2 + p\lambda + q$$

であり、その解の一般型は

$$\lambda_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$$

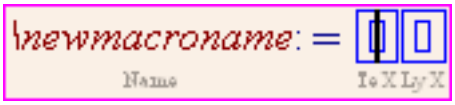
です。λ・p・q の 3 つのパラメータを指定することが必須であり、λ の指数をオプションとして与えることができるような、解の公式のコマンドを定義するには、以下のような LaTeX プリアンブル行を加えます。

\newcommand{\qG}[4][1,\,2]{#2␣{#1}=-\frac{#3}{2}\pm
\sqrt{\frac{#3^{2}}{4}-#4}}

これを使って解の公式を作るには、
\qG{\lambda→\{p→\{q というコマンドを数式に入れます。

新コマンドを定義する方法は、たとえば LaTeX 中で分数は **\frac{ 分子 }{ 分母 }** の形で入れなくてはならないことなど、使用するすべての LaTeX コマンドの書式を知っている必要があるので、直感的ではありません。さらに、定義中で中括弧を入れ忘れることはよくあり、それをやってしまうと、LyX からは新コマンドが何をやらかしているか確認しにくくなってしまいます。これらの問題を回避するために、LyX は、**\newcommand** コマンドの代わりに、数式マクロを使う方法を提供しています。

数式マクロは、挿入▷数式▷マクロメニューか、ツールバーボタンの  で作ることができます。すると、数式マクロツールバーが表示されるとともに、マクロを定義した箇所に以下のようなボックスが現れます。

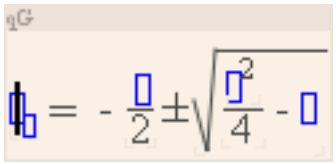


\newmacroname が既定のマクロ名として現れますが、意味のある名称に変更するべきでしょう。欲しい数式は一つ目の青枠の中に入れます。引数を置く場所は、**\#1** のように **\#引数番号**というコマンドで入力するか、マクロツールバーボタンの  を使用します。引数位置は赤で表示されます。引数は、最大で 9 つまでとることができます。非必須引数は、ツールバーボタンので作ることができます。最初の必須引数は、ツールバーボタンの  を使って、非必須引数にすることができます。二つ目の青枠には、LyX 中でのマクロの表示のしかたを定義することができます。通常は、定義したとおりに表示された方が便利なので、この枠は空白にしておきます。しかし、画面の過半を占拠してしまうようなマクロを作ってしまった場合には、たとえばこの枠に

qG: \#1 , \#2 , \#3, \#4

のように入れることができます。このようにすると、マクロ名と引数のみが LyX 上に表示され、見通しが良くなります。一方、出力での数式は、最初の枠で定義したように表示されます。
さらに、数式中のマクロ表示は、マクロの中にカーソルを置いて、表示▷数式マクロを展開（畳む）メニューを使うことで、マクロ毎に変えることができます。

マクロを使うには、数式中にマクロ名をコマンドとして入れます。上記の例では、**\qG** とします。このマクロは、LyX 中では以下のように表示されます。



以下は、上記の例に、引数 x・$\ln(x)$・B を指定したものです。

$$x_{1,2} = -\frac{\ln(x)}{2} \pm \sqrt{\frac{\ln(x)^2}{4} - B}$$

LyX は、ツール▷設定▷編集▷制御メニューで、マクロを編集するのに複数の様式を用意しています。あなたに最も合った様式を見つけるには、様式を選択してから、違いを見るために数式マクロにカーソルを合わせてみてください。

数式マクロは、文書書き出し時に、内部的に **\newcommand** コマンドに変換されます。こうして生成された **\newcommand** コマンドは、LaTeX プリアンブルには置かれませんので、マクロは、文書中、マクロ定義ボックスよりも後の数式でのみ使うことができます。

数式マクロは、**\newcommand** コマンドから直接作ることもできます。たとえば、LyX 中に通常の文章として、

\newcommand{\larrow}[2]{\xleftarrow[#2]{#1}}

というコマンドを書き入れ、この全体を選択して、短絡キー Ctrl+M を押すと、このコマンドは数式マクロに変換されます。この方法を使うに当たっては、**\newcommand** コマンドが正しく入力されていることに気をつけなくてはなりません。さもないと、間違ったマクロが作られてしまって、LaTeX エラーが発生します。

数式マクロには、まだ、マクロ定義中に再帰的に数式を入れてしまうと、正しく処理されないという問題が残っています。したがって、第 22.1節で例として作った **\fb** は、マクロとしては作ることができません。

カーソルがマクロ定義ボックスの中にあるとき、LyX 中に以下のようなマクロツールバーが表示されます。

マクロツールバーは、左から右に、以下の各ボタンがあります。


編集▷数式▷マクロ定義▷最後の引数を削除


編集▷数式▷マクロ定義▷引数を追加


編集▷数式▷マクロ定義▷最初の必須引数を非必須引数にする


編集▷数式▷マクロ定義▷最後の非必須引数を必須引数にする


編集▷数式▷マクロ定義▷非必須引数を削除


編集▷数式▷マクロ定義▷非必須引数を挿入


編集▷数式▷マクロ定義▷右に吐き出す形で最後の引数を削除


編集▷数式▷マクロ定義▷右から喰う形で引数を追加


編集▷数式▷マクロ定義▷右から喰う形で非必須引数を追加

23 さまざまな秘訣

23.1 負の数

数式中の負の数は、数の前の負符号が、差演算子記号と同じ長さに設定されてしまうために、汚く見えてしまうことがあります。負の数を通常の文章として書くと、負符号は正しく表示されます。

したがって、この問題は、負符号を数式テキストに変換することによって、解消されます。

以下は、この問題を示す例です。

通常の文章：	x = -2
数式：	$x = -2$
解決策：	x = -2

23.2 位区切りとしてのコンマ

LaTeX では、英語の慣習にしたがい、数式中のコンマを数字の位区切りに使用します。よって、数式中のコンマの後ろには、つねに空白が加わります。

これを回避するためには、コンマを選択して、数式テキストに変更して下さい（短絡キー Ctrl+M）。

文書中の数式コンマを、すべて小数点として使うには、LaTeX プリアンブルに

\usepackage{icomma}

という行を加えて、**icomma.sty**[47]ファイルを読み込みます。

23.3 物理ベクトル

LaTeX パッケージ **braket**[48]には、定義済みのベクトルが提供されており、

\usepackage{braket}

という LaTeX プリアンブル行で読み込むことができます。

以下のコマンドが定義されています。

コマンド	出力
\Bra{\psi	$\langle\psi\|$
\Ket{\psi	$\|\psi\rangle$
\Braket{\psi|\phi	$\langle\psi\,\|\,\phi\rangle$

\Braket コマンドを使うと、以下のように、すべての縦棒がそれを囲む括弧と同じ大きさに設定されます。

$$\left\langle \phi \middle| J = \frac{3}{2}\,,\, M_J \right\rangle$$

\Braket と同じ効果は、第 5.1.2節に説明されているとおり、**\middle** コマンドを用いることによっても実現できます。

[47] **icomma** は、LaTeX パッケージ **was** に含まれています。

[48] **braket** は標準的 LaTeX 頒布版のすべてに含まれています。

23.4 自己定義の分数

分数用の自製コマンドを定義するには、以下の書式を持つ **\genfrac** コマンドを使います。

\genfrac{ 左括弧 }{ 右括弧 }{ 分数線の厚み }{ 様式 }
{ 分子 }{ 分母 }

ここで「様式」は、0 - 3 の範囲の数字です。

数字	様式（大きさ）
0	別行建て様式の数式
1	行内数式
2	やや小（small）
3	最小（tiny）

「様式」を指定しないときには、**\frac** コマンドのように、大きさは周囲の環境に合わせて調節されます。

「分数線の厚み」を指定しないときには、既定値である 0.4 pt が用いられます。

たとえば、第 3.2節の **\dfrac** コマンドおよび **\tbinom** コマンドは、

\newcommand{\dfrac}[2]{\genfrac{}{}{}{0}{#1}{#2}}

あるいは

\newcommand{\tbinom}[2]{\genfrac{(}{)}{0pt}{1}{#1}{#2}}

というコマンドで定義できます。

分数線の厚みを非必須の引数として与えることのできる分数を定義するには、LaTeX プリアンブルに

\newcommand{\fracS}[3][]{\genfrac{}{}{#1}{}{#2}{#3}}

という行を入れます。

以下は、そのテストです。

コマンド　　\fracS[1mm]\{A → \{B　　\fracS[5mm]\{A → \{B

出力　　$\frac{A}{B}$　　$\frac{A}{B}$

ご覧になってわかるように、分子や分母から分数線までの距離は、分数線の厚みの約 3 倍になります。

23.5　数式の取り消し

数式あるいはその一部を取り消すには、**cancel**[49] LaTeX パッケージを、LaTeX プリアンブル行に

\usepackage[samesize]{cancel}

と書いて読み込む必要があります。

数式を取り消すには、4 つの方法があります。

コマンド	出力
\cancel{\int␣A=B	$\cancel{\int A = B}$
\bcancel{\int␣A=B	$\bcancel{\int A = B}$
\xcancel{\int␣A=B	$\xcancel{\int A = B}$
\cancelto{1→\{\int␣A=B	$\cancelto{1}{\int A = B}$

\cancelto は、以下のように、とくに数式中の分数を約分を表示するのに適しています。

$$\frac{(x_0 + bB)^2}{(1+b^2)^{\cancel{2}}} = \frac{x_0^2 + B^2 - r_g^2}{\cancel{1+b^2}}$$

23.6　節見出し中の数式

数式を節見出し中で使う際には、以下のことに留意しなくてはなりません。

文書設定ダイアログの PDF 特性で **hyperref** サポートが有効になっている場合、PDF のしおりが、目次にある節見出しすべてに関して生成されます。しおり中に数式を入れることは PDF の慣習に違反しているため、節見出しに数式が含まれている場合、数式はしおり中に誤った文字列として表示されます。

これらの問題は、挿入▷短縮タイトルメニューを使って、問題となる節見出しの最後に短縮タイトルを入れることで解決することができます。短縮タイトルは、目次が美しく整形されるように、多行にわたる節見出しに別名を付けるものです。目次中には、短縮タイトルのみが表示され、したがって PDF しおり中にも短縮タイトルのみが表示されます。

数式を目次中でも使わなくてもならないが、**hyperref** も使用しなくてはならないときには、

\texorpdfstring{ 部分 }{ 代替文字列 }

というコマンドを TeX モードで使う方法があります。

「部分」は、見出し中、PDF しおりに表示したくない部分です。これは、文字・数式・脚注のほかに相互参照をとることもできます。しおりには、この部分の代わりに、「代替文字列」が用いられます。

以下の二つは、見出しの例です。

[49] **cancel** は、標準的な LaTeX 頒布版のすべてに含まれています。

23.6.1　目次中では数式を使わない見出し $\sqrt{-1} = \mathrm{i}$

23.6.2　目次中で数式を使う見出し $\sqrt{-1} = \mathrm{i}$

一つめの見出しでは短縮タイトルが使われており、二つめの見出しでは **\texorpdfstring** が使われています。

他の節見出しと同じ書式を得るために、上の見出し全体は **boldmath** 環境に設定してあります[50]。

23.7　多段組文中の数式

多段組文中に数式を作ると、段の中に収まりきらないことも多く、ページ幅全体に広がるようにする必要があることがあります。これは、**multicol**[51] LaTeX パッケージを、

\usepackage{multicol}

という LaTeX プリアンブル行を書いて読み込むことで、実現できます。

ここで、文書▹設定メニューの本文レイアウトで、二段組文書の設定を**有効にしてはならない**ことに注意してください。

多段組文の前に

\begin{multicols}{ 段数 }

というコマンドを TeX モードで書き入れます。「段数」は、2 - 10 のあいだの数字です。多段組文の終わる数式の前には、

\end{multicols}

というコマンドを TeX モードで入れます。

このコマンドによって、数式の前にいくらかの余白が、自動的に作られます。これをなくすには、数式の前に-6 mm の垂直空白を入れて下さい。数式様式[52]として**行頭下げ**を使用している場合には、代わりに-9 mm の垂直空白を入れて下さい。

以下は、別行建て数式を含む、多段組文の例です。

Das Spektrum wird fouriertransformiert. Die Fouriertransformation wird verwendet, um die erlagerten Signale (Netzwerk, L ungsmittel) zu trennen. Nachdem wir die Phasenverschiebung bestimmen konnten, interessiert uns nun das Aussehen des Ausgangssignals. Im Experiment haben wir es mit sehr vielen Teilchen zu tun, so das man er alle Phasen integrieren muss. Sei nun S unser normiertes Ausgangssignal und P die Phasenverteilungsfunktion, so ergibt sich die Beziehung

$$S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t) \mathrm{e}^{\mathrm{i}\phi} \, \mathrm{d}\phi \tag{28}$$

[50] 第 11.2節参照。

[51] **multicol** は、標準的 LaTeX 頒布版のすべてに含まれています。

[52] **数式様式に関しては、第** 17節をご覧下さい。

wobei S_0 das Signal ohne Gradient ist und die Normierungsbedingung $\int_{-\infty}^{\infty} P(\phi,t)\,\mathrm{d}\phi = 1$ gilt. Nun d fen wir aber nicht den Relaxationsprozess auer Acht lassen. Direkt nach dem $\pi/2$-rf-Puls beginnt sich die Magnetisierung zu entfokussieren, wodurch sich das Signal zus 舩zlich abschw 臘ht. Diese Abschw 臘hung verl 舫ft exponentiell in Abh 舅gigkeit der so genannten T_2-Zeit.

23.8 変数の説明付き数式

(29) 式のように、数式内で変数の説明をするには、n 個の変数が使われている場合、左寄せの列を持つ $2\times n$ 行列を使用します[53]。説明を小さな文字にするには、行列の前に、たとえば **\footnotesize** コマンドを挿入します[54]。

数式様式に**行頭下げ**[55]を使っている場合、行列を数式とページ余白から等距離に置くために、行列の前後に **\hfill**[56]を入れます。

数式様式に**中央揃え**を使っている場合、数式を字下げするには、第 18.2.3節で述べた方法を使用します。(29) 式には 5 列があり、最初の 2 列には数式、3 列めには行列、最終列には空の TEX 括弧が入っています。

$$F_A = \rho \cdot V \cdot g \qquad \begin{array}{ll} \rho & \text{density} \\ V & \text{volume} \\ g & \text{gravitational acceleration} \end{array} \tag{29}$$

23.9 アップライト体のギリシャ小文字

ほとんどの数式書体は、イタリック体のギリシャ小文字しか提供していません。しかし、π 中間子やニュートリノのような素粒子の記号には、アップライト体のギリシャ文字が必要とされます。**upgreek.sty**[57] ファイルを

\usepackage{upgreek}

という LATEX プリアンブル行で読み込めば、これらが提供されるようになります。

この小節のすべてを出力で見るには、**upgreek** LATEX パッケージを導入する必要があります。

23.10 数式中のテキスト文字

折にふれて、テキスト文字を直接数式中に入れたいときがあるでしょう。たとえば、中黒「·」を $\nu = 5\cdot 10^5\,\mathrm{Hz}$ のように数式中で頻繁に用いようとすると、この中黒はすべてのエンコーディングでテキスト文字として定義されているために、代わりに **\cdot**[58]コマンドを挿入しなくてはならなくなることでしょう。しかし、

[53] 行列に関しては、第 4節参照。

[54] フォント寸法に関しては、第 11.4節参照。

[55] **数式様式に関しては、第** 17節参照。

[56] **\hfill** は、**行頭下げ**様式のときのみ機能します。第 8.2節をご覧下さい。

[57] **upgreek** は、**was** LATEX パッケージの一部です。

[58] 第 10.3節参照。

\Declare Inputtext{183}{\ifmmode\cdot\else\textperiodcentered\fi}

という LaTeX プリアンブル行を使えば、エンコーディングに変更を加えることができます。

文字エンコーディング（文書▷設定▷言語メニュー）は、キーボード上のキーが押されたときにどの文字が表示されるかを指定します。「·」文字に対応するキーが押されると、内部的には **\textperiodcentered** コマンドが使用されます。しかし、このコマンドは数式中では使えないので、LaTeX エラーが発生するのです。変更後のエンコーディングでは、文字が数式中に挿入されたか否かによって、正しいコマンドが自動的に選択されます。

定義ファイル中には、複数の文字のエンコーディングが保管されています。たとえば、**latin9** エンコーディングは、LaTeX がインストールされたフォルダにある **latin9.def** ファイルに定義されています。エンコーディングは、LaTeX プリアンブルで変更するべきであって、定義ファイルを変更してはなりません。さもないと、自分の作成した文書は、他のコンピューターで作業をしている他のユーザーによっては編集することができなくなってしまいます。

中黒の他にこの文書では、角度記号「°」が、数式に直接入れることができるよう、以下のような LaTeX プリアンブル行で定義されています[59]。

\DeclareInputtext{176}{\ifmmode^\circ\else\textdegree\fi}

[59]（訳註）pLaTeX では、これらの定義は必要ないので、コメントアウトして無効にしてあります。

A　組版上の助言

この節は、ISO 規範に掲げてある、もっとも重要な組版ルールの要約です[60]。

- 物理単位は、つねに（イタリック文中にあるときも）アップライト体にします[61]：30 km/h
  値と単位の間には、最小空白を入れます。第 8.1節を参照。
  この慣習は、**\unittwo** コマンドを使用すると、つねに満たされます。このコマンドを数式に入れると、二つの枠が現れます。最初の枠には値をいれ、第二の枠に単位を入れると、上記と同じような結果が得られます：30 km/h。実は、**\unittwo** は、LaTeX コマンドの実体ではなく、**\unit[値]{ 単位 }** というコマンドです。したがって、これを TeX コード中で使用することはできません。
- 百分率記号と千分率記号は、物理単位と同様に組みます：
  血中アルコール 1,2 ‰
- 角度記号は値の直後に置きます：15°。しかし、単位として用いられるときは別です：15 °C
- 4 桁以上の数は、3 桁ごとに最小空白を直前に挿入して、グループ化します：18 473 588
- 120×90×40 cm のような寸法には、積記号「×」を用います。これは、**\times** コマンドか、挿入▷特殊文字▷記号メニューから入れることができます。
- いくつかの文字を含む函数名は、混乱を防ぐためにアップライト体にします。第 15.1節を参照。
- 複数の文字を含む指数は、アップライト体にします：E_{kin}
  行列要素はイタリック体にします：$\hat{H}_{kl}$
- 微分作用素・積分作用素「d」、オイラー数「e」、虚数単位「i」は、他の変数と間違えることを避けるために、アップライト体にします。
- フーリエ変換を表す文字は、**\mathscr␣{F** コマンドか挿入▷特殊文字▷記号▷文字様記号メニューの $\mathscr{F}$ で入れることができます。
  \mathscr コマンドを使うためには、LaTeX パッケージの **mathrsfs** を、LaTeX プリアンブルで **\usepackage{mathrsfs}** として読み込む必要があります。

[60] この要約の一部は、ISO 規則を取り上げている「Duden」[8] と呼ばれるドイツの半公的辞書から採られています。

[61] 書体様式で指定します。第 11.1節を参照。

B 同義語

いくつかの文字や記号は、複数のコマンドから作ることができます。以下は、同義のコマンドの一覧です。

コマンド	同義のコマンド
\ast	*
\choose	\binom
\geq	\ge
\lbrace	{
\lbracket	[
\leftarrow	\gets
\leq	\le
\lor	\vee
\neq	\not=
\slash	/
\vert	|

コマンド	同義のコマンド
\backslash	\\
\dasharrow	\dashrightarrow
\land	\wedge
\rbrace	}
\rbracket	]
\rightarrow	\to
\lnot	\neg
\ne	\not=
\owns	\ni
\square	\Box
\Vert	\\|

参考文献

[1] Mittelbach, F. ; Goossens, M.: *The LaTeX Companion*, 2nd ed. Addison Wesley, 2004

[2] LaTeX の数式能力の説明

[3] $\mathcal{AMS}$-LaTeX の説明

[4] LaTeX パッケージで利用できる記号の全覧

[5] LaTeX **hyperref** パッケージの取扱説明書

[6] LaTeX **mhchem** パッケージの取扱説明書

[7] 第 10.2節に述べられている **\mathclap** コマンドの説明

[8] *Duden Band 1*. 22. Auflage, Duden 2001

[9] 原稿見直しのチェックリスト