

初等数学
L^AT_EX マクロ集
emath.sty ver. 1.54
使用例

tDB

2005/10/31

概要

中学・高校で数学のプリントを作る際に必要な記号，環境などを集めたマクロ集です。L^AT_EX 2_εを前提とします。

このマクロ集のマクロについてのご質問，バグ報告，修正・追加の提案等は

<http://emath.s40.xrea.com/>

の掲示板へどうぞ。

目次

1	代数		1
1.1	等しくない記号	\neq	1
1.2	近似等号	\approx	1
1.3	分数記号	$\frac{1}{2}$	1
1.4	約分記号	$\frac{\cancel{3}}{\cancel{6}}$	1
1.5	パーセント記号	3
1.6	訂正	誤り	3
1.7	整式の除法 (縦書割算)	$\begin{array}{r} x - 4 \\ x - 1 \overline{) x^2 - 5x + 2} \end{array}$	6
1.8	組み立て除法	$\begin{array}{r rrrrr} 1 & 1 & 3 & 4 & -6 & -2 \\ & & 1 & 4 & 8 & 2 \\ \hline & 1 & 4 & 8 & 2 & 0 \end{array}$	7
1.9	タスキガケ	$\begin{array}{r} 1 \quad \times \quad 3 \rightarrow 6 \\ 2 \quad \times \quad 5 \rightarrow 10 \\ \hline 11 \end{array}$	8
1.10	加減法	$\begin{array}{r} -2x + 13y = 11 \\ +) 2x - 3y = -1 \\ \hline 10y = 10 \end{array}$	9
1.11	累乗根	$\sqrt[3]{2}$	10
1.12	累乗の累乗	$(a^p)^q$	11
1.13	ベクトル記号	$\vec{a}, \overrightarrow{AB}$	12
1.13.1	鎌の形状	\overrightarrow{AB}	13
1.14	絶対値記号	$ \vec{a} , \left \frac{1+x}{1-x} \right $	14
1.14.1	¥zettaiti	14
1.14.2	¥emabs	14
1.14.3	¥vabs	15
1.14.4	絶対値記号のネスト	17
1.15	ガウス記号	$[x]$	17
1.16	行列	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	17
1.17	複素数	$x + yi$	19
1.18	共役複素数	\bar{z}	19
1.19	極形式	$\cos \theta + i \sin \theta$	20
2	幾何		21
2.1	度の記号 (小さな丸)	60°	21
2.2	角, 三角形	$\angle A, \triangle PQR$	21

2.3	平行とその否定	$\parallel, \not\parallel$	21
2.4	相似の記号	21
2.5	円弧を表す記号	\widehat{AB}	22
2.6	平行四辺形の記号	\square	22
3	解析		23
3.1	数列記号	$\{a_n\}, \sum_{k=1}^n a_k$	23
3.2	極限	$\lim_{n \rightarrow \infty} a_n$	23
3.3	階差数列	$\begin{array}{ccccccccc} 1 & 2 & 4 & 7 & 11 & \cdots \\ & \swarrow & \searrow & \swarrow & \searrow & \cdots \\ & 1 & 2 & 3 & 4 & \cdots \end{array}$	24
3.4	増減・凹凸	$\curvearrowright \curvearrowleft$	24
3.5	積分記号	$\int_a^b f(x)dx$	25
3.6	定積分	$\left[-\cos x \right]_0^\pi$	25
3.7	デルタ記号	Δ	25
4	確率		26
4.1	集合の記号	$\{x \mid 0 < x < 1\}$	26
4.1.1	$\forall ni$ の否定	$\not\exists$	26
4.2	階乗記号	$n!$	26
4.3	順列・組合せの記号	${}_nC_r \quad {}_nP_r \quad {}_nH_r \quad {}_n\Pi_r$	26
5	数式番号		28
5.1	丸付き文字	28
5.1.1	丸付き文字	$\textcircled{1} \textcircled{A} \textcircled{\mathcal{P}}$	28
5.1.2	白黒反転の丸付き文字	$\textcircled{1} \textcircled{A} \textcircled{\mathcal{P}}$	29
5.1.3	縦長の丸付き文字	$\textcircled{1} \textcircled{A}$	30
5.1.4	横長の丸付き文字	$\textcircled{1} \textcircled{A}$	31
5.1.5	のサイズ変更	31
5.2	数式番号のフォーマット	32
5.3	数式番号のリセット	33
5.4	連立方程式	35
5.5	本文中の連立方程式に数式番号	36
5.6	ダッシュ付きの数式番号①'	37
5.7	align*環境で数式番号	38
5.8	等号の縦揃え	38
5.9	$\forall item$ 直後の縦揃え数式	40
5.9.1	aligned 環境	40
5.9.2	$\forall itemtopmath$	41
5.9.2.1	左インデントの調整	41
5.9.2.2	縦方向の補正	43
5.9.2.3	$\forall itemtopmath$ の書式	43

6	図の取り込み	44
6.1	図の周りへの回り込み	44
6.1.1	mawarikomi 環境	44
6.1.2	mawarikomi*環境	44
6.1.3	mawarikomi 環境内に list 環境	45
6.1.4	回り込みの行数指定オプション	46
6.1.5	テキストと図の横間隔	47
6.1.6	図の位置の微調整	48
6.1.7	段落途中からの回り込み	49
6.1.8	¥caption の使用	51
6.1.9	書式	53
6.1.10	図の左配置	53
6.1.11	図幅指定の省略	54
6.1.12	テキスト部が短いとき	55
6.1.13	改ページとの関係 (1)	57
6.1.14	改ページとの関係 (2)	66
6.2	複数の図の周りへの回り込み	72
6.2.1	図の配置 — zuhaiti 環境	72
6.2.2	枠の設定 — ¥wakudori	73
6.2.3	図の左配置	75
6.2.4	図の位置の微調整	76
6.2.5	行の左右に枠	77
6.2.6	list との併用	78
6.2.7	¥caption の使用	80
6.2.8	テキスト部が複数の段落を持つ場合	81
6.2.9	¥unitlength の変更について	81
6.3	図番号の書式修正	82
7	enumerate 環境の機能拡張	82
7.1	小問を横に並べる。	83
7.1.1	横に並べる個数指定	83
7.1.2	ベタ並べ	90
7.1.3	yokoenumerate	91
7.2	enumerate 環境の中断・継続	92
7.3	指定した項目番号に*などを附加	93
7.4	横幅一定のローマ数字	94
7.5	固定桁数のナンバリング 001, 002, 003, ...	95
7.6	番号付けにカタカナ・ひらがなを使う。	96
7.7	enumerate 環境のオプション文字の追加	96
7.8	enumerate 環境における番号付けの初期値・刻み値変更	98
7.9	enumerate 環境の前後左右の空き調整	100
7.9.1	左インデント	102
7.9.2	縦間隔	103

7.9.3	縦間隔のグルー	104
7.9.4	<code>\enumSep</code>	104
7.9.5	<code>\enumLmargin</code>	105
7.10	<code>edaenumerate</code> 環境の前後左右の空き調整	107
7.10.1	<code>edaenumerate</code> 環境	107
7.10.2	<code>preedasep=..</code> オプション	108
7.10.3	<code>postedasep=..</code> オプション	110
7.10.4	<code>edatopsep=..</code> オプション	111
7.11	<code>\item</code> 直後の <code>edaenumerate</code> 環境	111
7.11.1	問題提示	111
7.11.2	大問の左インデント操作との競合	112
7.11.3	<code>edaenumerate</code> 環境に <code><edafirstindent=..></code> オプション	113
7.11.4	<code>emathAe</code> との併用	113
7.11.5	<code>betaenumerate</code> 環境では	115
7.11.6	解答部の <code>betaenumerate</code>	116
7.12	<code>list</code> 環境の行間	117
7.12.1	<code>itemize</code> 環境	117
7.12.2	<code>description</code> 環境	118
7.12.3	<code>enumerate</code> 環境	118
7.13	問題番号の参照	120
8	その他一般	123
8.1	数式モードの中で英大文字をローマン体にする。	123
8.1.1	<code>caprm</code> 環境	123
8.1.2	<code>[o]</code> オプション	124
8.1.3	コマンド型	124
8.1.4	<code>[1]</code> オプション	125
8.1.5	<code>\mathRM</code>	125
8.1.6	<code>\bekutoru*</code>	125
8.1.7	<code>\mitS</code> など	126
8.2	インデント	126
8.3	<code>\EMparbox</code>	130
8.3.1	<code>\parbox</code>	130
8.3.2	<code>\EMparbox</code>	130
8.3.3	<code>\EMparbox</code> のベースライン	131
8.4	<code>\emovalbox</code>	133
8.5	数式の配置 左（右）寄せと中央揃えの切り替え	133
8.6	数式番号を左に出力	135
8.7	行末にマーク	136
8.8	リーダー罫	137
8.9	支柱	138
8.10	下線	140
8.10.1	<code>\underline</code>	140

8.10.2	波下線	140
8.10.2.1	ulem.sty	140
8.10.2.2	¥namikasen	141
8.10.2.3	波二重下線	141
8.10.2.4	太波下線	141
8.10.2.5	¥namikasen と上下のアキ	142
9	メモリ不足への対応	143
9.1	pool size	143
9.2	main memory size	143
9.3	hash size	145
9.4	save size	145
9.5	number of strings	145
10	謝辞	146

1 代数

1.1 等しくない記号 \neq

等しくない記号を表すには `\ne` があります。しかし、日本では否定の斜線を逆向きにする習慣のようです。そのためのコマンド `\neqq` を定義しました。その効用は：

—— `\ne` と `\neqq` の違い ——

<pre>\$a \ne b\$ \$a \neqq b\$</pre>	$a \neq b$ $a \neqq b$
--	---------------------------

この記号は、数式の中で使われる位置によりサイズが自動的に変わります。

—— `\neqq` のサイズ調整 ——

<pre>\$i \neqq j\$ \$\displaystyle\sum_{i \neqq j} A_{ij}\$</pre>	$i \neqq j$ $\sum_{i \neqq j} A_{ij}$
---	--

1.2 近似等号 \doteq

AMSFonTS には、`\fallingdotseq` が用意されています。emath では、`\kinzi` という同義語を定義してあります。

—— `\kinzi` ——

<pre>\$(1+x)^n \kinzi 1+nx\$ \rightarrow (1+x)^n \doteq 1+nx</pre>
--

1.3 分数記号 $\frac{1}{2}$

分数を表すには、`\frac` というコマンドがあります。しかし、分数罫線が短いという憾みがあります。また、分数罫線と分母・分子との縦間隔が空き過ぎである、とのご意見もあります。そこで `\bunsuu` というコマンドを作りました。次の2つを比較してください。

—— `\dfrac` と `\bunsuu` の比較 ——

<pre>\$\$\dfrac{1}{2} \rightarrow \frac{1}{2}</pre>
<pre>\$\$\bunsuu{1}{2} \rightarrow \frac{1}{2}</pre>

1.4 約分記号 $\frac{3}{\cancel{6}}_2$

分数で、分母・分子の数値に斜線を引いて約分後の数値を記述するためのコマンド `\yakubun` です。斜線の引き方に2種類、約分した後の数値を記述する位置に3通りの方法があります。デフォルトは元の数値の上下です。

なお、この節のコマンドは、epic.sty の中で定義されている ¥drawline を用いています。したがって epic.sty, eepic.sty を読み込んでおくことが必要です。斜線の向きは [r] オプションを付けると逆向きになります。

$$\begin{array}{l} \text{--- ¥yakubun ---} \\ \$¥yakubun\{16\}\{6\}\{8\}\{3\}\$ \quad \longrightarrow \frac{16^8}{6^3} \\ \$¥yakubun[r]\{16\}\{6\}\{8\}\{3\}\$ \quad \longrightarrow \frac{16^8}{6^3} \end{array}$$

次は、約分した数値を元の数値の右肩に乗せます。

$$\begin{array}{l} \text{--- ¥yakubun<r> ---} \\ \$¥yakubun<r>\{16\}\{6\}\{8\}\{3\}\$ \quad \longrightarrow \frac{16^8}{6^3} \\ \$¥yakubun[r]<r>\{16\}\{6\}\{8\}\{3\}\$ \quad \longrightarrow \frac{16^8}{6^3} \end{array}$$

最後は、約分した数値を元の数値の左肩に乗せます。

$$\begin{array}{l} \text{--- ¥yakubun<l> ---} \\ \$¥yakubun<l>\{16\}\{6\}\{8\}\{3\}\$ \quad \longrightarrow \frac{8^8 16}{3^6} \\ \$¥yakubun[r]<l>\{16\}\{6\}\{8\}\{3\}\$ \quad \longrightarrow \frac{8^8 16}{3^6} \end{array}$$

¥yakubun の書式です。

$$\text{--- ¥yakubun の書式 ---}$$

¥yakubun [#1] (#2)<#3>#4#5#6#7

- #1: 斜線の引き方を指定するオプション引数で、
 - s : 斜線 (/) [デフォルト]
 - r : 斜線 (\)
- #2: 線の色
- #3: 約分した結果の表示位置を指定するオプション引数で、
 - c : 中央上下 [デフォルト]
 - l : 左肩
 - r : 右肩
- #4: 約分する前の分子
- #5: 約分する前の分母
- #6: 約分した後の分子
- #7: 約分した後の分母

なお、分母・分子の最大公約数を求め、自動的に既約分数にしてしまうコマンド ¥Yakubun もあります。

書式は ¥yakubun の書式とほとんど同様で、最後の 2 つの引数 (約分後の分母・分子) が不要になる点だけが異なります。

ただし、このコマンドを使用するときは、emathW.sty を読み込んでおく必要があります。

また、いくつかの分数を掛ける式での約分を表現するには、後述の `\teisei` コマンドが有効です。

1.5 パーセント記号

L^AT_EX でのパーセント記号は `\%` で出力できますが、その形状に不満がある、という方もあるようです。pxfonts のそれを `\pxpercent` として用いることもできます。当然、pxfonts がインストールしてあるという前提です。具体的な使用法は `sampleFx.tex` をご覧ください。

1.6 訂正

訂正
誤り

約分後の数値を表示せず、単に斜線だけを引くコマンドが `\teisei` です。数学では、プラスとマイナスで相殺されるものに斜線を引く、などといったことができます。

なお、この節のコマンドは、`epic.sty` の中で定義されている `\drawline` を用いています。したがって `epic.sty`, `eepic.sty` を読み込んでおく必要があります。

```

\teisei
$%left(1-%teisei{\%bunsuu12}%right)
+%left(%teisei{\%bunsuu12}-%teisei{\%bunsuu13}%right)
+%left(%teisei{\%bunsuu13}-%teisei{\%bunsuu14}%right)
+%cdots %cdots
+%left(%teisei{\%bunsuu{1}{n}}-%bunsuu{1}{n+1}%right)
= 1-%bunsuu{1}{n+1}$

```

をタイプセットすると、

+ と - で相殺されるものに斜線

$$\left(1 - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \left(\frac{1}{3} - \frac{1}{4}\right) + \cdots + \left(\frac{1}{n} - \frac{1}{n+1}\right) = 1 - \frac{1}{n+1}$$

`\teisei` コマンドは指定した文字列に 2 種類の斜線を引く他、水平線、二本の水平線などで取消し線を引くことができます。その際、必要なら線の色を指定することができます。さらに、訂正前の文字列の上方に、訂正後の文字列をおくことができます。

```

\teisei[h]
$a_n=a_1+%retuwa{k=1}
{%teisei[h](red){n}[%
{\scriptstyle n-1}]}
b_k$

```

$$\rightarrow a_n = a_1 + \sum_{k=1}^{n-1} b_k$$

`\teisei` の書式です。

¥teisei の書式

¥teisei[#1](#2)#3[#4]<#5>

#1: 線の引き方についてのオプション引数で,

s : 斜線 (/) [デフォルト]

r : 斜線 (\)

h : 横線

d : 横二本線

#2: 線の色

#3: 線を引く対象

#4: 訂正後の文字列

#5: 斜線の位置を修正するベクトル

key=val の形式

dLT=(dx,dy), dLB=, dRT=, dRB= 右辺値は pt を単位とする数値

<#5>オプションは、斜線の位置を修正するためのオプションですが、この機能は emathPh.sty を必要とします。

まずは基本的な使用法の確認です。

¥teisei

¥teisei{ab}

ab

打ち消し線を,

左下 (LB) を少し左へ (dLB={(-1,0)}),

右上 (RT) を少し右下へ (dRT={(1,-2)})

動かしてみます。ここで

L : left, R : right, T : top, B : bottom

右辺ベクトル成分は, pt を単位とする無名数です。

¥teisei

¥teisei{ab}<dLB={(-1,0)},dRT={(1,-2)}>

ab

水平の打ち消し線は, 左端を LT, 右端を RT としてあります。

¥teisei

¥teisei[h]{ab}¥¥

¥teisei[h]{ab}<dLT={(-.5,-1)},dRT={(.5,-1)}>

ab

ab

水平の二重線は

上のほうの左端が LT, 右端が RT

下のほうの左端が LB, 右端が RB

```
¥Teisei
¥teisei[d]{ab}¥¥
¥teisei[d]{ab}<dLT={(-.5,-1.5)},dRT={(.5,-1.5)},%
dLB={(-.5,-1)},dRB={(.5,-1)}>
```



¥teisei は、訂正後の文字列を、訂正前の文字列と同じサイズで上方に配置します。
それに対して ¥Teisei は、訂正後の文字列を小さ目のサイズ (scriptstyle) で、配置場所をいろいろと指定することができます。使用例を見ていただきましょう。

```
$$$bunsuu{5}{¥Teisei{6}{2}}
¥times¥bunsuu{¥Teisei{9}{3}}{4}
=¥bunsuu{15}{8}$
```

→
$$\frac{5}{8^2} \times \frac{9^3}{4} = \frac{15}{8}$$

¥Teisei はネストすることもできます。

```
$$$bunsuu{¥Teisei{6}{¥Teisei[r]{¥scriptstyle 2}{1}}}{13}
¥times¥bunsuu{5}{¥Teisei{33}{11}}
¥times¥bunsuu{1}{¥Teisei[r]{2}{1}}=¥bunsuu{5}{143}$
```

→
$$\frac{8^{2^1}}{13} \times \frac{5}{33^{11}} \times \frac{1}{2^1} = \frac{5}{143}$$

¥Teisei の書式です。

```

¥Teisei[#1](#2)#3[#4]#5
#1: 線の引き方についてのオプション引数で,
    s : 斜線 ( / ) [デフォルト]
    r : 斜線 ( \ )
    h : 横線
    d : 横二本線
#2: 線の色
#3: 線を引く対象
#4: 訂正後の文字位置
    r = 右上 (=tr=rt)
    l = 左上 (=tl=lt)
    t = 上
    b = 下
    rb=br=右下
    lb=bl=左下
#5: 訂正後の文字列 (scriptstyle)

```

注 ¥teisei コマンドなどにより描画される斜線は `tpic-specials` を用いています。special は `dvi-ware` に依存します。そこで、¥vrule で描画した線を ¥rotatebox で回転して斜線を描画する道も用意してあります。もちろんこれも `dvi-ware` に依存しますから、ケースバイケースで使い分けをする必要がありましょう。

さて、その実現法は

```
¥def¥syasen{¥rotateline}
```

を宣言します。

ただし、¥rotatebox はグラフィックスパッケージを必要としますから、`emath` では、`emathPh.sty` を読み込んでおかなければなりません。

1.7 整式の除法 (縦書割算)

$$x-1 \overline{) x^2 - 5x + 2}$$

整式の除法を縦書きで記述するためのコマンドが ¥zyohou です。

縦書割算

```

 $\$zzyohou{x^2,-5x,+2}$  被除式
{x,-1} 除式
{x,-4} 商
{x^2,-x,} 途中の計算式
-4x,+2,} を , で区切って
-4x,+4,} 並べる .
-2}$
    
```

$$\begin{array}{r}
 x \ -4 \\
 x-1 \overline{) x^2 -5x +2} \\
 \underline{x^2 \ -x} \\
 -4x +2 \\
 \underline{-4x +4} \\
 -2
 \end{array}$$

係数に分数が登場するときは、`\frac` をお使いください。

縦書割算

```

 $\$zzyohou{2x^2,-2x,-1}$ 
{2x,-1}
{x,-\frac{1}{2}}
2x^2,-x,
-x,-1,
-x,+\frac{1}{2},
-\frac{3}{2}}$
    
```

$$\begin{array}{r}
 x \ -\frac{1}{2} \\
 2x-1 \overline{) 2x^2 -2x -1} \\
 \underline{2x^2 \ -x} \\
 -x -1 \\
 \underline{-x +\frac{1}{2}} \\
 -\frac{3}{2}
 \end{array}$$

`\dffrac` や `\bunsuu` の場合は、`\noexpand` を併用することになりますが、面倒です。

なお、係数が整数の範囲内の一変数の場合は `emathW.sty` で定義されている `\izyohou` が便利です。こちらは、被除式、除式の係数だけを与えれば、商はもちろん、途中の計算式もすべて `TEX` が求めてくれますから、

```
\izyohou{1,-5,2}{1,-1}
```

と記述するだけです。詳しくは `emathW.sty` の使用例 `sampleW.tex` をご覧ください。

1.8 組み立て除法

$$\begin{array}{r|rrrr}
 1 & 1 & 3 & 4 & -6 & -2 \\
 & & 1 & 4 & 8 & 2 \\
 \hline
 & 1 & 4 & 8 & 2 & 0
 \end{array}$$

整式を1次式で割ったときの商、余りを求める計算法に‘組み立て除法’があります。

組み立て除法

```
 $\$syndiv{1,-6,11,-6}{2}$ 
```

$$\begin{array}{r|rrrr}
 2 & 1 & -6 & 11 & -6 \\
 & & 2 & -8 & 6 \\
 \hline
 & 1 & -4 & 3 & 0
 \end{array}$$

すなわち `\syndiv` コマンドの第1引数に、被除式の係数を降べき順にコンマで区切ったものを並べ、第2引数に除式のゼロ点を与えます。

第2引数にコンマ区切りで複数のものを与えれば、商をさらに割っていくことができます。

多段組み立て除法

`\syndiv{1,-6,11,-6}{2,3}`

$$\begin{array}{r|rrrr} 2 & 1 & -6 & 11 & -6 \\ & & 2 & -8 & 6 \\ \hline 3 & 1 & -4 & 3 & 0 \\ & & 3 & -3 & \\ \hline & 1 & -1 & 0 & \end{array}$$

`\syndiv` は整数係数しか扱うことができませんが, `emathB.sty` には, 分数係数を扱うことを可能とした `\Fsyndiv` が用意してあります。例えば

`\Fsyndiv{3,-1,0,4}{-2/3}`

とすることで, $f(x) = 3x^3 - x^2 + 4$ に対して $f\left(-\frac{2}{3}\right)$ を求めることができます。

1.9 タスキガケ

$$\begin{array}{r} 1 \quad \times \quad 3 \quad \longrightarrow \quad 6 \\ 2 \quad \times \quad 5 \quad \longrightarrow \quad 5 \\ \hline 11 \end{array}$$

二次三項式を因数分解するとき, タスキガケの図を用いることがあります。

タスキ

`2x^2+11x+15=(x+3)(2x+5)`

`\tasuki{1}{2}{3}{5}`

$$2x^2 + 11x + 15 = (x + 3)(2x + 5)$$

$$\begin{array}{r} 1 \quad \times \quad 3 \quad \longrightarrow \quad 6 \\ 2 \quad \times \quad 5 \quad \longrightarrow \quad 5 \\ \hline 11 \end{array}$$

横幅を少し狭くしたいときは `\stasuki` を用います。

`\stasuki`

`\tasuki{1}{2}{3}{5}`

`\stasuki{1}{2}{3}{5}`

$$\begin{array}{r} 1 \quad \times \quad 3 \quad \longrightarrow \quad 6 \\ 2 \quad \times \quad 5 \quad \longrightarrow \quad 5 \\ \hline 11 \end{array}$$

$$\begin{array}{r} 1 \quad \times \quad 3 \quad \longrightarrow \quad 6 \\ 2 \quad \times \quad 5 \quad \longrightarrow \quad 5 \\ \hline 11 \end{array}$$

文字式の場合は, タスキ計算もこちらでしておく必要があります。

文字式のタスキ

```
\begin{align*}
& 2x^2+xy-3y^2+5x+5y+2 \\
= & \{& 2x^2+(y+5)x-(3y^2-5y-2)\} \\
= & \{& 2x^2+(y+5)x-(y-2)(3y+1)\} \\
= & \{& \mathbf{(x-y+2)(2x+3y+1)}\} \\
\end{align*}
\Tasuki{1}{2}{-(y-2)}{3y+1}{-2y+4}{3y+1}{y+5}
```

$$\begin{aligned}
& 2x^2 + xy - 3y^2 + 5x + 5y + 2 \\
&= 2x^2 + (y+5)x - (3y^2 - 5y - 2) \\
&= 2x^2 + (y+5)x - (y-2)(3y+1) \\
&= (x-y+2)(2x+3y+1)
\end{aligned}$$

→

1	\times	$-(y-2)$	→	$-2y+4$
2	\times	$3y+1$		$\frac{3y+1}{y+5}$

タスキの左側だけを表示したいときは、`¥tasuki` に [1] オプションをつけます。

<code>¥tasuki</code> の [1] オプション <code>¥tasuki[1]{1}{2}{3}{5}\$</code>	→	<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">1</td> <td style="padding-right: 10px;">\times</td> <td style="padding-right: 10px;">3</td> </tr> <tr> <td>2</td> <td>\times</td> <td>5</td> </tr> </table>	1	\times	3	2	\times	5
1	\times	3						
2	\times	5						

文字式の場合も左側だけほしいときは `¥Tasuki` ではなく、`¥tasuki` の [1] オプションを使います。

<code>¥tasuki</code> の [1] オプション <code>¥tasuki[1]{1}{2}{-(y-2)}{3y+1}\$</code>	→	<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">1</td> <td style="padding-right: 10px;">\times</td> <td style="padding-right: 10px;">$-(y-2)$</td> </tr> <tr> <td>2</td> <td>\times</td> <td>$3y+1$</td> </tr> </table>	1	\times	$-(y-2)$	2	\times	$3y+1$
1	\times	$-(y-2)$						
2	\times	$3y+1$						

なお、たすきがけの図第 3 行については、次の 2 種類の変形版があります。

<code>¥tasukikata=1</code> <code>¥def¥tasukikata{1}</code> <code>¥tasuki{1}{2}{3}{5}\$</code>	→	<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">1</td> <td style="padding-right: 10px;">\times</td> <td style="padding-right: 10px;">3</td> <td style="padding-right: 10px;">→</td> <td>6</td> </tr> <tr> <td>2</td> <td>\times</td> <td>5</td> <td>→</td> <td>5</td> </tr> <tr> <td colspan="4"></td> <td style="text-align: right;">11</td> </tr> </table>	1	\times	3	→	6	2	\times	5	→	5					11
1	\times	3	→	6													
2	\times	5	→	5													
				11													

<code>¥tasukikata=2</code> <code>¥def¥tasukikata{2}</code> <code>¥tasuki{1}{2}{3}{5}\$</code>	→	<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">1</td> <td style="padding-right: 10px;">\times</td> <td style="padding-right: 10px;">3</td> <td style="padding-right: 10px;">→</td> <td>6</td> </tr> <tr> <td>2</td> <td>\times</td> <td>5</td> <td>→</td> <td>5</td> </tr> <tr> <td>2</td> <td></td> <td>15</td> <td></td> <td>11</td> </tr> </table>	1	\times	3	→	6	2	\times	5	→	5	2		15		11
1	\times	3	→	6													
2	\times	5	→	5													
2		15		11													

なお、`¥tasukikata` のデフォルト値は 0 です。

$$\begin{array}{r}
-2x + 13y = 11 \\
+) 2x - 3y = -1 \\
\hline
10y = 10
\end{array}$$

1.10 加減法

連立方程式を加減法で解くときの計算式を表示するためのマクロです。

加減法 <code>¥kagenhou{3,-6,9}+{2,6,-4}</code>	→	<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">$3x - 6y = 9$</td> </tr> <tr> <td style="padding-right: 10px;">+) $2x + 6y = -4$</td> </tr> <tr> <td style="padding-right: 10px;">$5x = 5$</td> </tr> </table>	$3x - 6y = 9$	+) $2x + 6y = -4$	$5x = 5$
$3x - 6y = 9$					
+) $2x + 6y = -4$					
$5x = 5$					

未知数の個数が増えたり、文字が x, y ではない場合については、オプション引数で未知数を表す文字を指定します。次の例は a, b, c を未知数とする 3 元の連立です。

加減法 (3元)

`\kagenhou[a,b,c]{3,-6,9,3}-{2,-6,-4,8}`

→

$$\begin{array}{r} 3a - 6b + 9c = 3 \\ -) 2a - 6b - 4c = 8 \\ \hline a \quad + 13c = -5 \end{array}$$

問題部分のみを表示させるには、`<M>` オプションを付加します。

`\kagenhou` の `<M>` オプション

`\kagenhou<M>{1,2,3}-{4,2,-1}`
`\smallskip`

$$\begin{array}{r} x + 2y = 3 \\ -) 4x + 2y = -1 \end{array}$$

縦方向の配置は、`array` 環境に `[b]` オプションを与えてあります。

`\kagenhou` の縦配置

`a\kagenhou<M>{1,2,3}-{4,2,-1}z`
`\smallskip`

$$\begin{array}{r} x + 2y = 3 \\ a -) 4x + 2y = -1 z \end{array}$$

これを変更するには、`<pos=.>` オプションを用います。右辺値は `t/b/c` のいずれかです。また、このオプションを `<M>` オプションと併用するときは、`<kotae=no>` の形式で与えます。

`\kagenhou` の `<pos=t>` オプション

`a\kagenhou<pos=t,kotae=no>{1,2,3}-{4,2,-1}z`
`\smallskip`

$$\begin{array}{r} a \quad x + 2y = 3 z \\ -) 4x + 2y = -1 \end{array}$$

1.11 累乗根

$$\sqrt[3]{2}$$

平方根、3乗根などを表記するには \LaTeX で `\sqrt` が用意されています。

累乗根

`\begin{enumerate}[(1)]`
`\item $\sqrt{2}$`
`\item $\sqrt[3]{2}$`
`\end{enumerate}`

- (1) $\sqrt{2}$
- (2) $\sqrt[3]{2}$

この記号は、中の数式のサイズにしたがって大きさが変化します。便利な反面、次の (1) では、高さが不揃いとなりますので、支柱 `\mathstrut` をいれて高さをそろえるなどの工夫が必要になります。

高さの調節 (1)

`$$\sqrt{ab}=\sqrt{a}\sqrt{b}$$$`
`$$\sqrt{\mathstrut ab}`
`=\sqrt{\mathstrut a}\sqrt{\mathstrut b}$$`

→

$$\sqrt{ab} = \sqrt{a}\sqrt{b}$$

$$\sqrt{ab} = \sqrt{a}\sqrt{b}$$

`\mathstrut` は、‘(’の高さと深さを持つ支柱ですが、深さがある分根号が下の方に伸びてしまう、という批判もあります。このときは、支柱を `\vphantom` を用いて

——— 高さの調節 (2) ———

$$\sqrt{\sqrt{\vphantom{b}}ab}$$

$$= \sqrt{\sqrt{\vphantom{b}}a}\sqrt{b}$$

→

$$\sqrt{ab} = \sqrt{a}\sqrt{b}$$

つぎに、二重根号を `\displaystyle` で組むと

——— 二重根号 ———

$$\sqrt{\sqrt{2+\sqrt{3}}}$$

→

$$\sqrt{2+\sqrt{3}}$$

外側の根号が大きくなりすぎて2つの根号の縦間隔が空きすぎる、というご意見があります。対策は外側の根号に `\textstyle` をつけることです。emath.sty では、`\textstyle\sqrt` の省略形として `\tsqrt` を用意しています。

——— `\tsqrt` ———

$$\sqrt{\tsqrt{2+\sqrt{3}}}$$

→

$$\sqrt{2+\sqrt{3}}$$

1.12 累乗の累乗

$$(a^p)^q$$

累乗の累乗 $(a^p)^q$ は、 $\{(a^p)\}^q$ と記述すると、 $(a^p)^q$ となり、2種の累乗 p と q のベースラインが異なります。高校の教科書などでは、これが揃っているものが多いようです。そのためのコマンドが `\dpower` です。次の2つを比較してください。

——— `\dpower` の効用 ———

$$\{(a^p)\}^q \quad \longrightarrow (a^p)^q$$

$$\dpower[a]{p}{q} \quad \longrightarrow (a^p)^q$$

底 a が複雑になり、カッコを含むときは、外側のカッコの種類を `\dpower` のオプション引数で指定することができます。

——— `\dpower` のオプション引数 ———

$$\dpower[\left]{\left(\bunsuu{y}{x}\right)}{p}{q} \quad \longrightarrow \left\{\left(\frac{y}{x}\right)^p\right\}^q$$

$$\dpower[\left[\left]{\left(\bunsuu{y}{x}\right)}{p}{q} \quad \longrightarrow \left[\left(\frac{y}{x}\right)^p\right]^q$$

話が前後しますが、累乗の位置を気にされる方があります。次の3つを比較してください。

累乗の位置

`2^x` `¥¥`
`¥displaystyle 2^x$` `¥¥`
`¥{2¥mathstrut}^x$`

2^x
 2^x
 2^x

1.13 ベクトル記号 \vec{a} , \overrightarrow{AB}

LaTeX で用意されている矢線付きのベクトル記号は `¥vec` です。

しかし、`¥vec{a}`、`¥vec{b}` などと並べたとき、矢線の高さが不揃いになります。その点を修正した `¥beku` を作りました。

下の例で、矢線の高さにご注目ください。

¥vec と ¥beku の比較

`¥vec{a}`、`¥vec{b}` $\longrightarrow \vec{a}, \vec{b}$
`¥beku{a}`、`¥beku{b}` $\longrightarrow \vec{a}, \vec{b}$

添字がついた場合です。

添字付きベクトル (1)

`¥beku{a_n}` $\longrightarrow \vec{a}_n$
`¥beku{a}_n` $\longrightarrow \vec{a}_n$
`¥overrightarrow{¥mathstrut a_n}` $\longrightarrow \vec{a}_n$

上の3つのどれがお好みでしょうか。

次は始点と終点を並べてその上に矢線を付ける表現です。

LaTeX では、`¥overrightarrow` を使いますが、これは数式モードの中でなければ使えません。`¥overrightarrow{AB}` とすると、 AB が斜体になります。教科書では、点を表す記号は立体となっています。その点を工夫したのが `¥bekutoru` コマンドです。

¥bekutoru

`¥overrightarrow{AB}` $\longrightarrow \vec{AB}$
`¥bekutoru{AB}` $\longrightarrow \vec{AB}$

添字がつくと面倒になります。下の二つどちらがお好みでしょうか。

添字付きベクトル (2)

`¥bekutoru{AB}_n` $\longrightarrow \vec{AB}_n$
`¥bekutoru{AB}_n` $\longrightarrow \vec{AB}_n$

矢線の高さを調整するコマンドが `¥bekutorusityuu` です。デフォルトは

`\vrule height .9zh width \z@`

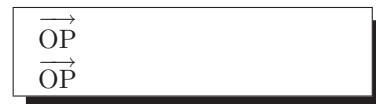
すなわち `.9zh` となっています。

———— `\bekutorusityuu` ————

```

\bekutoru{OP}\#
\bekutorusityuu{\vphantom{OP}}%
\bekutoru{OP}

```



すなわち，自然な高さでは矢印が文字にかかるのを嫌って，高くしています。

`\beku` についても，`\bekusityuu` で高さを変更できます。

———— `\bekusityuu` ————

```

\beku{a}\#
\bekusityuu{\vphantom{a}}%
\beku{a}

```



こちらのデフォルトは，`\vphantom{b}` としてあります。

`\bekutoru` にはアスタリスク付きのコマンド `\bekutoru*` があります。これは後で述べる `caprm` 環境下で用いることを想定していますので，§ 8.1.6 (p.125) で解説します。

1.13.1 鋸の形状 \overrightarrow{AB}

鋸の形状にご不満をお持ちの方が多ようです。それを変更する方法もあります。まずは，鋸を塗りつぶした三角形にする方法です。

———— 塗りつぶした鋸 ————

```

\bekutoru{AB}

\bekutorukata{fill}
\bekutoru{AB}

```



すなわち，鋸の形状を変更するには，`\bekutorukata` コマンドを用います。ただし，このコマンドは `emath.sty` ではなく，`emathPh.sty`, `emathFx.sty` で定義されています。

さらに，塗りつぶした二等辺三角形に窪みをつけたい，というご要望もあります。

———— 窪みも ————

```

\bekutoru{AB}

\bekutorukata<.25>{fill}
\bekutoru{AB}

```



`\bekutorukata` の引数は，上記の `fill` の他に 4 種類 — `cm`, `eu`, `tx`, `xy` — あります。ただし，`tx` を使用するには，`txfonts` が必要です。また，`xy` を使用するには，`Xy-pic` パッケージが必要です。さらに，`emathFx` をロードしておかねばなりません。具体的な実現方法については，`sampleFx.tex` をご覧ください。

1.14 絶対値記号

$$|\vec{a}|, \left| \frac{1+x}{1-x} \right|$$

1.14.1 `\zeta`

ベクトルの大きさを表すのにも使われる絶対値記号です。

L^AT_EX では `|\bektu{b}|` としますが、矢印が絶対値記号の上に飛び出しています。これを修正するコマンド `\zeta` を用意しました。上下を比べてみてください。

|...| と `\zeta` の比較

<code> \bektu{AB} </code>	→	$ \vec{AB} $
<code>\zeta{\bektu{AB}}</code>	→	$ \vec{AB} $

この記号は中の式の高さにより伸縮します。

絶対値記号の伸縮

<code>\zeta{\bunsuu{1}{3}\bektu{AB}}</code>	→	$\left \frac{1}{3} \vec{AB} \right $
<code>\zeta{\bunsuu{m}\bektu{OA}}{m+n}</code>	→	$\left \frac{m \vec{OA}}{m+n} \right $

このコマンドはベクトルに限らず一般の絶対値記号を表すことができます。

`|1-|x||` では、外側の絶対値記号と内側の絶対値記号が同じとなります。できれば外側の絶対値記号を大きくしたいものです。

絶対値記号の伸縮オプション

<code> 1- x </code>	→	$ 1- x $
<code>\bigl 1- x \bigr </code>	→	$ 1- x $
<code>\zeta[1.5pt]{1-\zeta{x}}</code>	→	$ 1- x $

3行目は外側の `\zeta` コマンドのオプション引数に `1.5pt` を与えることで実現しています。この数値だけ、標準の高さを上下それぞれ引き伸ばします。これは2行目のものとほぼ同じです。

なお、`\zeta` の別名として、`\EMabs` も用いることができます。

1.14.2 `\emabs`

`\zeta` は絶対値記号を付与する数式の高さ・深さを基準としていますから、上下不均等です。したがって他の括弧記号と併記するとおかしなことになります。

→ %zettaiti

```

$%zettaiti%
  %bunsuu{a_1}{1+%bunsuu{1}{n}}$%
だけをみているとよいのですが、
%[ %bunsuu{n}%left(
  %zettaiti{
    %bunsuu{a_1}{1+%bunsuu{1}{n}}
+%cdots+
  %zettaiti{
    %bunsuu{a_n}{1+%bunsuu{n}{n}}
  %right) %]
などと、他の括弧記号と並べると

```

$$\left| \frac{a_1}{1 + \frac{1}{n}} \right|$$

だけをみているとよいのですが、

$$\frac{1}{n} \left(\left| \frac{a_1}{1 + \frac{1}{n}} \right| + \dots + \left| \frac{a_n}{1 + \frac{n}{n}} \right| \right)$$

などと、他の括弧記号と並べると

上下不均等が目立ってしまいます。

ということで、上下均等な絶対値記号を表すコマンド %emabs を用意しました。

→ %emabs

```

%[ %bunsuu{n}%left(
  %emabs{
    %bunsuu{a_1}{1+%bunsuu{1}{n}}
+%cdots+
  %emabs{
    %bunsuu{a_n}{1+%bunsuu{n}{n}}
  %right) %]

```

$$\frac{1}{n} \left(\left| \frac{a_1}{1 + \frac{1}{n}} \right| + \dots + \left| \frac{a_n}{1 + \frac{n}{n}} \right| \right)$$

括弧記号とのバランスはよくなりましたが、絶対値記号の中で天が空きすぎである、というご批判があるでしょうね。これは、式表現を工夫して

→ %emabs

```

%[ %bunsuu{n}%left(
  %emabs{
    %bunsuu{a_1}{1+1/n}}
+%cdots+
  %emabs{
    %bunsuu{a_n}{1+n/n}}
  %right) %]

```

$$\frac{1}{n} \left(\left| \frac{a_1}{1 + 1/n} \right| + \dots + \left| \frac{a_n}{1 + n/n} \right| \right)$$

などとするのも一法でしょう。

1.14.3 %vabs

次に、 \vec{AB} などベクトル記号に絶対値を付与することを考えてみます。%zettaiti を用いると

ベクトルにゼータ

```

 $\zeta_{\text{ベクトル}\{OP\_1\}}$ 
だけをみているとよいのですが、
 $\left[ \zeta_{\text{ベクトル}\{OP\_1\}} + \dots + \zeta_{\text{ベクトル}\{OP\_n\}} \right]$ 
などと、他の括弧記号と並べると

```

$|\overrightarrow{OP_1}|$ だけをみているとよいのですが、
 $\frac{1}{n} (|\overrightarrow{OP_1}| + \dots + |\overrightarrow{OP_n}|)$
 などと、他の括弧記号と並べると

上下不均等が目立ってしまいます。さりとて、emabs では

ベクトルに emabs

```

 $\left[ \text{emabs}_{\text{ベクトル}\{OP\_1\}} + \dots + \text{emabs}_{\text{ベクトル}\{OP\_n\}} \right]$ 

```

$$\frac{1}{n} (|\overrightarrow{OP_1}| + \dots + |\overrightarrow{OP_n}|)$$

折衷案で

ベクトルに absbeku

```

\def\absbeku#1{\left|\vphantom{.5ex}\{#1}\right.\vphantom{\text{ベクトル}\{#1}\right}\vphantom{\text{ベクトル}\{#1}\right}}
\begin{caprm}
 $\left[ \text{absbeku}\{OP\_1\} + \dots + \text{absbeku}\{OP\_n\} \right]$ 
\end{caprm}

```

$$\frac{1}{n} (|\overrightarrow{OP_1}| + \dots + |\overrightarrow{OP_n}|)$$

いっそのこと、矢印は無視して

単純に ‘|’

```

 $\left[ |\text{ベクトル}\{OP\_1\}| + \dots + |\text{ベクトル}\{OP\_n\}| \right]$ 

```

$$\frac{1}{n} (|\overrightarrow{OP_1}| + \dots + |\overrightarrow{OP_n}|)$$

1.14.4 絶対値記号のネスト

絶対値記号の中に更に絶対値記号が入る場合、例えば次のような場合、外側の絶対値記号を大きくしたいことがあります。

—— 絶対値記号のネスト ——
 $\$[||a|-|b|| \$]$
 $\$cmd\{emabs\}$ を用いると
 $\$[\emabs\{\emabs\{a\}-\emabs\{b\}\} \$]$

$||a|-|b||$
 \emabs を用いると
 $||a|-|b||$

外側の絶対値記号を大きくするには、 \emabs に [...] オプションを附加します。数値は適宜で

—— \emabs の [...] オプション ——
 $\$[\emabs[1pt]\{\emabs\{a\}-\emabs\{b\}\} \$]$
 $\$[\emabs[2pt]\{\emabs\{a\}-\emabs\{b\}\} \$]$

$||a|-|b||$
 $||a|-|b||$

などといった調子です。

やっていることは、外側の絶対値記号の中身について、実際の高さ・深さそれぞれにオプション値を附加した支柱を立てています。

1.15 ガウス記号 $[x]$

x を超えない最大の整数を表すガウス記号を $[x]$ とすると $[x]$ となり、 x とその左右の括弧の間が詰まりすぎています。それを修正したのが $\$gauss$ です。

このコマンドは $\$ensuremath$ であるから、地の文でも数式内でも使用することができます。

—— ガウス記号 ——
 $\$gauss\{x\}$ は $\$x\$$ を超えない最大の整数を表します。例えば
 $\$[\$gauss\{\pi\}=3 \$]$

→

$[x]$ は x を超えない最大の整数を表します。
 例えば
 $[\pi] = 3$

1.16 行列 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

高校では、二次の正方行列が登場します。そこで二次の正方行列専用のコマンド $\$gyouretu$ を作りしました。

—— $\$gyouretu$ ——
 $\$gyouretu\{a\}\{b\}\{c\}\{d\} \rightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix}$
 単位行列 $\$gyouretu1001 \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

`\gyouretu` では、各列の要素は中央揃えに配置されます。これを右揃えにするには `[r]` オプションを用います。

配置オプション

`\gyouretu{12}{-3}{4}{5}`
`\gyouretu[r]{12}{-3}{4}{5}`

$$\begin{pmatrix} 12 & -3 \\ 4 & 5 \end{pmatrix}$$

$$\begin{pmatrix} 12 & -3 \\ 4 & 5 \end{pmatrix}$$

2 次の列ベクトルも良く登場します。
 正方行列と絡めて、

`\retube`

$$\text{\retube}{x}{y} \longrightarrow \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\text{\gyouretu}{abcd}\text{\retube}{xy} \longrightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

今後は三次の正方行列も登場するでしょう。

`\gyouretu`

$$\text{\gyouretu}{a}{b}{c}{d}{e}{f}{g}{h}{i} \longrightarrow \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$$\text{零行列 } \text{\gyouretu}00000000 \longrightarrow \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

また、三次の列ベクトルは `\Retube` です。

三次の列ベクトル

`\gyouretu001010100`
`\Retube xyz`
`=\Retube zyx`

$$\longrightarrow \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} z \\ y \\ x \end{pmatrix}$$

行ベクトルは `\gyoube` です。

`\gyoube`

すべての実数 x, y について
 $\text{\gyoube}{xy}\text{\gyouretu}{a}{5a-2}{2-a}{3a+2}\text{\retube}{xy} \geq 0$
 が成立するような実数 a の値の範囲を求めよ。
\syutten{2004 福島医科大学}

すべての実数 x, y について $(x \ y) \begin{pmatrix} a & 5a-2 \\ 2-a & 3a+2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \geq 0$ が成立するような
 実数 a の値の範囲を求めよ。 [2004 福島医科大学]

一般の行列を表すには、`\pgyouretu` を用意してあります。列の区切りを ‘,’、行の区切りを ‘;’ で表します。

```
\pgyouretu
\pgyouretu{
  1 , 2 , 3;
  4 , 5 , 6}
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

`amsmath.sty` には、`pmatrix` 環境があります。

```
pmatrix 環境
$
\begin{pmatrix}
  1 & 2 & 3 \\
  4 & 5 & 6
\end{pmatrix}
$
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

1.17 複素数 $x + yi$

複素数 $x + yi$ において、 y と i の間を少し空けたい、という場合のために `\Cnum` を用意しました。

```
\Cnum
$2-3i$, $x+yi$
\Cnum{2}-{3}$, \Cnum{x}+{y}$
```

$$2 - 3i, x + yi$$

$$2 - 3i, x + yi$$

1.18 共役複素数 \bar{z}

共役複素数を表す `\kyouyaku` です。

```
\kyouyaku
\kyouyaku{z\cdot w}=\kyouyaku z\cdot\kyouyaku w \rightarrow z \cdot w = \bar{z} \cdot \bar{w}
```

文字と横棒の間隔は `\mathstrut` で支柱を立てています。これを変更するオプションが `[..]` です。

```
\kyouyaku[..]
\kyouyaku{a}
\kyouyaku[\vphantom a]{a}
```

$$\bar{a}$$

$$\bar{a}$$

なお、`\kyouyaku` の別名として `\conj` も使えます。

広範囲で支柱の高さを変更するには `\conjstrut` を用いることもできます。

```

\conjstrut
\conjstrut{\vphantom{b}}
\conj b=\conj a$

```

$$\bar{b} = \bar{a}$$

1.19 極形式 $\cos \theta + i \sin \theta$

複素数の極形式を表現するコマンド `\kyokukeisiki` です。[...] オプションで絶対値を指定することもできます。

なお、`\kyokukeisiki` と同値なコマンド `\kyoukukeisiki` も使用できます。

```

\kyoukukeisiki
\kyokukeisiki{\theta}$          \to \cos \theta + i \sin \theta
\polar{60\Deg}$                \to \cos 60^\circ + i \sin 60^\circ
\kyokukeisiki[r]{\theta}$      \to r (\cos \theta + i \sin \theta)
\kyokukeisiki[r]{\{-\theta\}}$ \to r \{\cos (-\theta) + i \sin (-\theta)\}

```

共役なものあらわすには、<c>オプションをつけます。

```

\polar<c>
\polar[r]{\theta}$\polar
\polar<c>[r]{\theta}$

```

$$r (\cos \theta + i \sin \theta)$$

$$r (\cos \theta - i \sin \theta)$$

2 幾何

2.1 度の記号 (小さな丸) 60°

角度の単位, 例えば 60 度を表すときの記号です。下の 3 種類, どれがお好みでしょうか。

`\Deg` `\DEG` `\degree`

<code>\$60\Deg\$</code>	→	60°
<code>\$60\DEG\$</code>	→	60°
<code>\$60\degree\$</code>	→	60°

2.2 角, 三角形 $\angle A, \triangle PQR$

角は `\angle`, 三角形は `\triangle` です。ただし, `\triangle{ABC}` などと数式環境で使うと, ABC が斜体になってしまいます。教科書などでは, 点を表す記号は立体という習慣です。そこで

`\kaku`, `\sankaku`

<code>\kaku{ABC} < 90\Deg</code>	→	$\angle ABC < 90^\circ$
<code>\sankaku{ABC} \equiv \sankaku{DEF}</code>	→	$\triangle ABC \equiv \triangle DEF$

2.3 平行とその否定 $//, \not\parallel$

L^AT_EX には平行記号を表すコマンド `\parallel` が, また AMSFonts には, その否定を表すコマンド `\nparallel` があります。しかし, 日本で使われている記号とは違っていています。そこで `\heikou` と `\nheikou` を決めました。

次の上段と下段を比較してください。

`\heikou` と `\nheikou`

<code>\parallel m</code>	→	$l \parallel m$	<code>\nparallel m</code>	→	$l \not\parallel m$
<code>\heikou m</code>	→	$l // m$	<code>\nheikou m</code>	→	$l \not\parallel m$

2.4 相似の記号

相似の記号は L^AT_EX には用意されていません。ここでは全角の記号を使ってしまおうという, 無精な案です。

`\souzi`

<code>\sankaku{ABC} \souzi \sankaku{PQR}</code>	→	$\triangle ABC \sim \triangle PQR$
---	---	------------------------------------

この記号は, 文字サイズの変更に追従するようになっています。`\Huge` とすると

`\Huge`

<code>\Huge\sankaku{ABC}</code>	<code>\souzi \sankaku{PQR}</code>
---------------------------------	-----------------------------------

$\triangle ABC$	$\triangle PQR$
-----------------	-----------------

2.5 円弧を表す記号 \widehat{AB}

次は円弧を表すお椀を伏せたような記号です。

<code>\\$¥ko{AB}:¥ko{BC}=3:4\$</code>	$\widehat{AB} : \widehat{BC} = 3 : 4$
<code>\\$¥ko{APB}\$</code>	\widehat{APB}
<code>¥ko{APQB}</code>	\widehat{APQB}
<code>¥ko{A\$_1\$B\$_1\$}</code>	$\widehat{A_1B_1}$
<code>¥ko{P\$_n\$P\$_{n+1}\$}</code>	$\widehat{P_nP_{n+1}}$

最後の $\widehat{P_nP_{n+1}}$ では、弧の記号が P の添字につられて右に寄ってしまいます。¥ko のオプション引数で、弧の記号の位置を調整することができます。次の二つを比べてください。

<code>¥ko{P\$_n\$P\$_{n+1}\$}</code>	$\widehat{P_nP_{n+1}}$
<code>¥ko[.6em]{P\$_n\$P\$_{n+1}\$}</code>	$\widehat{P_nP_{n+1}}$

コマンド ¥ko の引数はテキストモードとしてあります。したがって添え字をつけたいときは

<code>¥ko{A\$_1\$A\$_2\$}</code>	$\widehat{A_1A_2}$
----------------------------------	--------------------

と、添え字部分を数式モードにする必要があります。数列がらみの問題では、ちょっと煩わしいので、引数が数式モードに入る ¥ko* もあります。

<code>¥begin{caprm}</code> <code>¥ko*{A_1A_2}</code> <code>¥end{caprm}</code>	$\widehat{A_1A_2}$
---	--------------------

ただし、英大文字をローマン体とするには、caprm 環境などを用いる必要があります。

2.6 平行四辺形の記号 \square

平行四辺形を表す記号です。picture 環境で表現する方法をトニイさんが FPRINT で発表されました。ここで使うことを認めていただきましたので、それをこのマクロ集に入れさせていただきました。

有り難うございます。> トニイさん。

フォントサイズを大き目にします。

<code>平行四辺形 ¥shikaku{}ABCD</code>	$\square ABCD$
-----------------------------------	----------------

それでも平行四辺形の記号がやや大き目です。小さくできれば良いのですが、 \LaTeX の `picture` 環境の制約で不可能です。そこで、`epic` というパッケージの助けを借りて小さくしたのが、私の修正案です。

`\theikousihenkei`

平行四辺形 `\theikousihenkei{}`ABCD \rightarrow 平行四辺形 \square ABCD

3 解析

3.1 数列記号 $\{a_n\}, \sum_{k=1}^n a_k$

\LaTeX では、数列 $\{a_n\}$ を書くには、`\$a_n\$\$` としますが、わずらわしいので、コマンド `\suuretu` を作りました。次のように本文中の中で使うことが多いでしょう。

`\suuretu`

等比数列 `\suuretu{a_n}` が \dots \rightarrow 等比数列 $\{a_n\}$ が \dots

数列の和記号も良く登場します。`\displaystyle\sum` のシグマ記号は少し大きいようです。少し小さ目に見してみました。

`\retuwa`

`\displaystyle\sum_{k=1}^n{k(k+1)}` \rightarrow $\sum_{k=1}^n k(k+1)$
`\retuwa{k=1}{n}k(k+1)` \rightarrow $\sum_{k=1}^n k(k+1)$

上の話は本文中のことです。別行立ての場合は大きい方がよいでしょう。この場合も小さくするには、`\tretuwa` を用います。

`\tretuwa`

```
\begin{gather*}
\retuwa{k=1}{n}k^2\retuwa
\tretuwa{k=1}{n}k^2
\end{gather*}
```

$$\sum_{k=1}^n k^2$$

$$\sum_{k=1}^n k^2$$

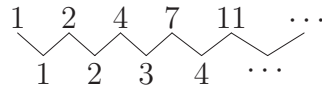
3.2 極限 $\lim_{n \rightarrow \infty} a_n$

`\displaystyle` を毎回書くのは面倒だというだけです。`\dfrac` と同じねらいでしょう。

`\dlim`

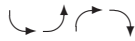
`\dlim{x \to \infty}e^{-x}=0` \rightarrow $\lim_{x \rightarrow \infty} e^{-x} = 0$

3.3 階差数列



階差数列の表現,あるいは数列の上下に,規則性を示すための矢印を配置する機能は `emathPh.sty` に用意されています。 `sampleP.tex` をご覧ください。

3.4 増減・凹凸



増減表を作るときの矢印は, \LaTeX に

`\searrow`, `\nearrow`

が用意されています。

x	-2	...	-1	...	1	...	2
y'		-	0	+	0	-	
y		\searrow	極小	\nearrow	極大	\searrow	

凹凸を表す記号は, 集合の演算子

`\cup`, `\cap`

が流用できます。

x	-2	...	0	...	2
y''		+	0	-	
y		\cup	変曲点	\cap	

`\smile`, `\frown`

の方が良いという方もいらっしゃるかもしれません。

x	-2	...	0	...	2
y''		+	0	-	
y		\smile	変曲点	\frown	

増減と凹凸をいっしょにまとめた表を作るときに必要な矢印付きの四分円は \LaTeX , `AMSFonTS` ではみかけません。 `picture` 環境で作ってみました。

```

\ne?arrow
\nevarrow → ↗
\necarrow → ↘
\sevarrow → ↙
\secarrow → ↖

```

`ne(se)` と `arrow` の間の `v`, `c` は

`v`: `convex`

`c`: `concave`

からとりました。

これらのコマンドを用いると増減表は下のようになります。

↑e?arrow									
x	-2	...	-1	...	0	...	1	...	2
y'		-	0	+			0	-	
y''		+			0	-			
y		↘	極小	↗	変曲点	↖	極大	↙	

増減表で、微分不能のところには斜線を引いたりしたいことがあります。emathT.styでは、hyou環境を定義して、それを可能としています。emathP.styの説明書 sampleP.texをご覧ください。

3.5 積分記号

$$\int_a^b f(x)dx$$

積分記号も `\dfrac` と同じ意味合いです。

↑dint	
<code>\dint{a}{b}f(x)\,dx</code>	$\int_a^b f(x) dx$

3.6 定積分

$$\left[-\cos x \right]_0^\pi$$

定積分の計算で登場する大括弧でくくった記号ですが、大括弧を大きくするため、最低でも分数式の高さ深さを加味しています。

↑teisekibun	
<code>\dint{0}{\pi}\sin x\,dx</code>	$\int_0^\pi \sin x dx$
<code>[-\cos x]_0^{\pi}</code>	$[-\cos x]_0^\pi$
<code>\dint{0}{\pi}\sin x\,dx</code>	$\int_0^\pi \sin x dx$
<code>=\teisekibun{-\cos x}{0}{\pi}</code>	$[-\cos x]_0^\pi$

$\int_0^\pi \sin x dx = [-\cos x]_0^\pi$
$\int_0^\pi \sin x dx = [-\cos x]_0^\pi$

3.7 デルタ記号



変分を表す記号デルタはイタリック体のギリシャ文字が使用されますが、これをもっと傾けたいというための記号です。

↑Deruta	
<code>\varDelta</code>	Δ
<code>\Deruta</code>	Δ

Δ
Δ

ただし、この記号は emathPh.sty で定義されています。emath.sty にもありますが、コーナーがぎざぎざになっています。

4 確率

4.1 集合の記号 $\{x \mid 0 < x < 1\}$

集合を表す記号として、`\syuugou` があります。

```
\syuugou
\syuugou{1,2,3}
\syuugou[x]{0<x<1}
```

```
{1,2,3}
{x | 0 < x < 1}
```

`\syuugou#1` は、`#1` に要素を列挙します。

`\syuugou[#1]#2` は、`#2` に条件を記述します。

4.1.1 `\ni` の否定 \ni

次の記号は、標準的な $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ に用意されています。

```
\in など
\begin{gather*}
a \in A
a \notin A
A \ni a
\end{gather*}
```

```
a \in A
a \notin A
A \ni a
```

しかし、`\ni` の否定記号がありません。(`txfonts` には用意されています。) そこで、`emath` では `\notni` を用意しました。

```
\notni
\ [ A \notni a ]
```

```
A \notni a
```

4.2 階乗記号 $n!$

階乗を表す記号は `!` ですが、前後に少し空白を附加したものが `\kaizyou` です。比較してみてください。

```
\kaizyou
$5!=120$
$5\kaizyou=120$
```

```
5! = 120
5\kaizyou = 120
```

4.3 順列・組合せの記号 ${}_nC_r$ ${}_nP_r$ ${}_nH_r$ ${}_n\Pi_r$

組合せの記号は $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ で `\binom` などが用意されていますが、列ベクトルのような記号です。

¥kumiawase, ¥zyunretu

$$\begin{aligned} \text{\$}\displaystyle\text{\$binom}\{n}\{2}\text{\$} &\longrightarrow \binom{n}{2} \\ \text{\$}\text{\$kumiawase}\{n}\{2}\text{\$} &\longrightarrow {}_n\text{C}_2 \\ \text{\$}\text{\$kumiawase}\{5}\{3}\text{\$} &\longrightarrow {}_5\text{C}_3 \\ \text{\$}\text{\$zyunretu}\{n}\{2}\text{\$} &\longrightarrow {}_n\text{P}_2 \end{aligned}$$

重複順列，重複組合せはそれぞれ ¥Zyunretu, ¥Kumiawase です。

重複順列，重複組合せ

¥Zyunretu{n}{r}¥¥
¥Kumiawase{n}{r}

${}_n\Pi_r$
 ${}_n\text{H}_r$

5 数式番号

5.1 丸付き文字

5.1.1 丸付き文字 ① ④ ⑦

式の番号などに丸付き数字がよく使われます。

L^AT_EX2e には、`\textcircled` というコマンドがあります。

`\textcircled`
`\textcircled{1}`, `\textcircled{2}` を連立させて

①, ② を連立させて

しかし、丸記号と中の数字とのバランスが悪いですね。そこで `emath.sty` では、中の数字を `\scriptstyle` にした `\maru` というコマンドを用意しました。

`\maru`
`\maru{1}`, `\maru{2}` を連立させて

①, ② を連立させて

中の数字を小さくしすぎたようですが、数字が 2 桁になると

2 桁の番号
`\maru{19}`, `\maru{20}` を連立させて

①9, ②0 を連立させて

中の数字が外枠にかかってしまいます。さて、どうしましょうか。

対策その 1 は、中の数字を `\resizebox` で横幅を `1zw` に圧縮しよう、というのが `\zwmaru` です。

`\zwmaru`
`\zwmaru{19}`, `\zwmaru{20}` を連立させて

①9, ②0 を連立させて

対策その 2 は、逆に外枠の を大きくしよう、という行きかたです。

`ifsym` フォントに大き目の が用意されています。

ただし、このフォントはスタンダードなインストールでは、インストールされていませんから、ここで使用するのには遠慮しておきます。サンプルは `emath` のホームページ、目次から

`emath` 講座

リファレンス編

`\maru`

とたどってください。

文字サイズは `\normalsize` のままで を大きくするには、 を `picture` 環境で描画した `\emPmaru` が `emathPh.sty` で定義されています。

`\emPmaru`
`\emPmaru{19}`, `\emPmaru{20}` を連立させて

①9, ②0 を連立させて

対策その 3 は、独自にデザインされたフォントを探してくる方法です。その一つに `pifont` があります。下のリストをタイプセットするには、`emathFx` を `[pzd]` オプションつきでロードしておく必要があります。

¥pzdmaru

¥pzdmaru{9}, ¥pzdmaru{10} を連立させて

⑨, ⑩ を連立させて

ただし、残念ながら 10 までしか用意されていません。もう少し大きな数値まで用意されているものに go フォントがあります。このフォントは、囲碁の棋譜を表示するために作成されたもので、252 まで用意されています。

このフォントもスタンダードなインストールでは、インストールされていませんから、上記 emath 講座をご参照ください。

最後に、utf パッケージを用いる方法です。ただし、この方法は dvi-ware & OS に依存します。やはり、ここにはサンプルを載せられませんから、上記 emath 講座をご参照願います。

注 ベースライン

¥maru で囲まれた文字は、その行のベースラインから少し上に上がっています。これを、一致させるコマンドが ¥bmaru です。次の例では、ベースラインを赤で引いています。

¥bmaru

```

¥begin{picture}(0,0)
¥put(0,0){¥color{red}¥line(1,0){50}}
¥end{picture}
¥maru{1}

¥begin{picture}(0,0)
¥put(0,0){¥color{red}¥line(1,0){50}}
¥end{picture}
¥bmaru{1}

```

¥maru に代えて、¥bmaru を使いたければ

```
¥let¥maru¥bmaru
```

を宣言します。

5.1.2 白黒反転の丸付き文字 ① A ア

白黒反転の丸付き文字は ¥kmaru コマンドです。ただし、このコマンドは color パッケージを必要とします。emathPh など emathP 系のパッケージをロードする場合は自動的に color も読み込まれます。

¥kmaru

```

¥kmaru{1}

¥kmaru{あ}

¥kmaru{¥textgt{ア}}

```

番号付けに使ってみましょう。

enumerate 環境に適用

```
¥def¥kanaban#1{¥kmaru{¥textgt{#1}}}  
¥begin{enumerate}[¥protect¥expandafter¥kanaban ア]  
  ¥item ああああ  
  ¥item いいいい  
  ¥item うううう  
¥end{enumerate}
```

㊦ ああああ

㊧ いいいい

㊨ うううう

5.1.3 縦長の丸付き文字 ① ㊦

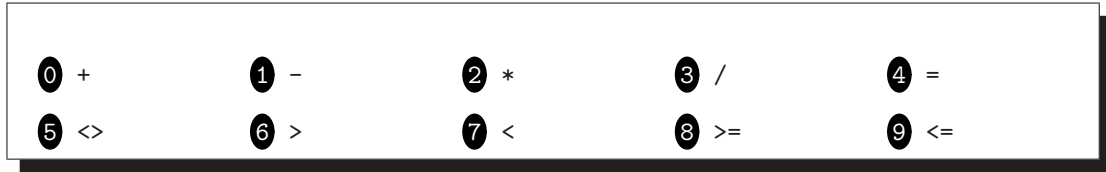
縦長の も使用できます。

```
{¥ttfamily  
¥vspace*{-.5¥baselineskip}  
  ¥begin{edaenumerate}<5>[¥protect¥expandafter¥nagamaru 1]  
  ¥setcurrentenum{-1}%  
  ¥item +  
  ¥item -  
  ¥item{*}  
  ¥item /  
  ¥item =  
  ¥item{<>  
  ¥item >  
  ¥item{<  
  ¥item >=  
  ¥item{<=  
  ¥end{edaenumerate}  
}%
```

→

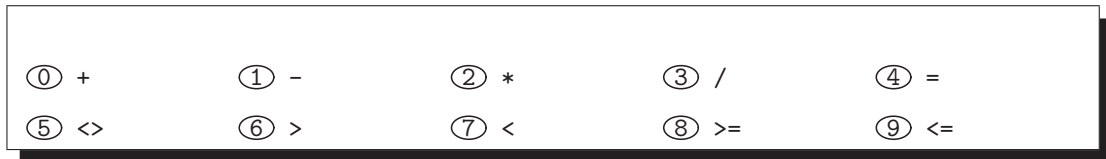
① + ② - ③ * ④ / ⑤ =
⑥ <> ⑦ > ⑧ < ⑨ >= ⑩ <=

白黒反転は ¥knagamaru です。上の例を ¥knagamaru に変更すると：

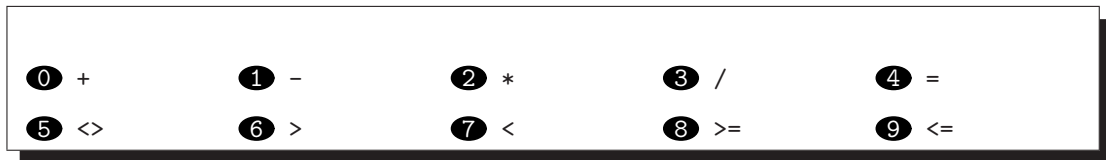


5.1.4 横長の丸付き文字 ① ㉠

横長の は, ¥nagamaru です。



その白黒反転は ¥kynagamaru です。



5.1.5 のサイズ変更

の中に桁数の多いものを入れたいときなど, のサイズを変更するには

¥nagamaruyokoHankei, ¥nagamarutateHankei

を用います。



¥ynagamaru は

¥nagamarutateHankei{14ex}

¥nagamaruyokoHankei{21ex}

としてあります。¥nagamaru については, この値が固定されています。変更したいときは,

¥ynagamaru ではなく, ¥nagamaru

を用います。

5.2 数式番号のフォーマット

複数行の式を並べて式に番号を付ける際、 \LaTeX では

(1), (2), (3), ...

のようにカッコ付きの番号が振られます。これに対して、われらの日常環境では丸付きの番号を振ることが多いようです。

この機能は、`amsmath` パッケージを前提とします。さらにその一部を書き換えていますので、パッケージの読み込み順序が問題となります。`amsmath` を読み込んだ後に `emath` を読み込んでください。

数式番号に丸囲い数字

<pre> \begin{align} y&=x \quad \text{\label{eq:1}}\text{\} y&=x^2 \quad \text{\label{eq:2}} \end{align} </pre>	→	$y = x \quad \dots\dots\dots \textcircled{1}$ $y = x^2 \quad \dots\dots\dots \textcircled{2}$
<p><code>\eqref{eq:1}</code>を <code>\eqref{eq:2}</code>に 代入して</p>		<p>①を②に代入して</p>

リーダー罫が長くて数式とダブるときは、`\preEqlabel{\$}\cdots\%cdots\%` などとして、リーダー罫の長さを短くすることができます。

`\preEqlabel`

<pre> \preEqlabel{\\$}\cdots\%cdots\% \begin{align} y&=x \quad \text{\} y&=x^2 \end{align} </pre>	→	$y = x \quad \dots\dots \textcircled{3}$ $y = x^2 \quad \dots\dots \textcircled{4}$
---	---	---

また、`fleqn` オプションを付けているときは、数式が短いと数式と式番号の間が長くなりますから、`jquote` 環境で右マージンを設定するのも一法です。

右マージン

<pre> \begin{jquote}(0zw)(4zw) \begin{align} y&=x \quad \text{\} y&=x^2 \end{align} \end{jquote} </pre>	→	$y = x \quad \dots\dots \textcircled{5}$ $y = x^2 \quad \dots\dots \textcircled{6}$
---	---	---

`jquote` 環境については、126 ページをご覧ください。
あるいは、逆に `\mathindent` で、数式を右に寄せるのもありましょう。

```

\mathindent
\mathindent=6zw
\begin{align}
y&=x \tag*{} \tag{7}
y&=x^2 \tag{8}
\end{align}

```

`\preEqlabel{}` とすれば、リーダー罫をつけないこともできます。

```

\preEqlabel{}
\begin{align}
y&=x \tag{9}
y&=x^2 \tag{10}
\end{align}

```

なお、数式番号をデフォルトの

(1),(2),(3),....

に戻したければ、

```

\renewcommand{\tagform}[1]{(#1)}%
\preEqlabel{}%

```

とします。これをまとめて処理する `\resettagform` コマンドもあります。

数式番号の形式をデフォルトに戻す。

```

\resettagform
\begin{align}
y&=x \tag{11}
y&=x^2 \tag{12}
\end{align}

```

$$y = x \tag{11}$$

$$y = x^2 \tag{12}$$

5.3 数式番号のリセット

デフォルトでは、数式番号は1つの文書で通し番号となります。これを変更するには `\setcounter` を用いるのが普通です。

数式番号の変更

```

\setcounter{equation}{10}
\begin{align}
y&=x \tag{11}
y&=x^2 \tag{12}
\end{align}

```

$$y = x \tag{11}$$

$$y = x^2 \tag{12}$$

`\setcounter` で設定した値 (上の例では 10) の次の番号から数式に付与されていきます。
 特に数式番号を 1 から振りなおすには `\setcounter{equation}{0}` とすることになりますが、よく使いますので `emath` では、`\resetcounter` というコマンドを作っています。

数式番号の初期化

```
\resetcounter{equation}
\begin{align}
y&=x \quad \#\#
y&=x^2
\end{align}
```

$$y = x \quad \dots\dots\dots ①$$

$$y = x^2 \quad \dots\dots\dots ②$$

このコマンドは、カウンタの親子関係を作るのが主目的で、数学の問題集を作るときなど、数式番号は大問ごとによりセットすることが多いでしょう。
 このようなときは

カウンタの親子関係

```
\resetcounter{equation}[enumi]
\begin{enumerate}[問題 1.~]
\item ああああああああああ
\begin{align}
y&=x \quad \#\#
y&=x^2
\end{align}
\item いろいろいろいろいろいろ
\begin{gather}
x^2+y^2=1\#\#
x+y=1
\end{gather}
\end{enumerate}
```

問題 1. ああああああああああ

$$y = x \quad \dots\dots ①$$

$$y = x^2 \quad \dots\dots ②$$

問題 2. いろいろいろいろいろいろ

$$x^2 + y^2 = 1 \quad \dots\dots ①$$

$$x + y = 1 \quad \dots\dots ②$$

冒頭の 1 行は、`enumi` カウンタが更新されるたびに、`equation` カウンタを初期化せよ、という意味です。`\resetcounter` の書式です。

カウンタ値のリセット

`\resetcounter{foo}` カウンタ `foo` の値を初期化します。

カウンタに親子関係を設定する。

`\resetcounter{foo}[F00]` `F00` の値が更新されると、`foo` は初期化されます。

5.4 連立方程式

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ で連立方程式を記述するのに、cases 環境を使う方法があります。

$$\begin{cases} a + b = 2 \\ a - b = 0 \end{cases} \dots\dots\dots ①$$

などという具合です。ただ、残念ながら align 環境の中に記述しても数式番号は、連立方程式全体に対して1つだけしか発行されません。連立方程式それぞれに番号が欲しいことがあります。

そのためには、cases.sty で定義されている numcases 環境を用います。

```
numcases 環境
\preEqlabel{\%Cdots\Cdots}
\begin{numcases}{ }
    a+b=3 \label{eq:a} %%
    a-b=1 \label{eq:b}
\end{numcases}
\%eqref{eq:a}+\%eqref{eq:b}$
から $2a=4$ ゆえに $a=2$
```

$$\begin{cases} a + b = 3 & \dots\dots ② \\ a - b = 1 & \dots\dots ③ \end{cases}$$

② + ③ から $2a = 4$ ゆえに $a = 2$

numcases 環境の書式です。

```
\begin{numcases}{括弧の左側に来る数式}
数式 1 & 説明 1 %%
数式 2 & 説明 2 %%
...
\end{numcases}
```

この機能をフルに用いた例です。

```
numcases 環境
\resettagform
\begin{numcases}{|x|=}
    x & $(x \geqq 0)$ のとき %%
    -x & $(x < 0)$ のとき
\end{numcases}
```

$$|x| = \begin{cases} x & (x \geqq 0) \text{ のとき} & (4) \\ -x & (x < 0) \text{ のとき} & (5) \end{cases}$$

(注 1) cases.sty と amsmath.sty とで衝突する部分がありますから、emath.sty よりも後で cases.sty を読み込む必要があります。

(注 2) cases.sty のバージョンによっては、エラーが発生することがあります。この節での記述は

cases.sty 2002/05/02 ver 2.5

を前提としています。

(注 3) emath には、`\renritu` というコマンドもありますが、`numcases` の方が一般的ですから、`\renritu` はお蔵入りとします。

5.5 本文中の連立方程式に数式番号

連立方程式に数式番号を振る `\renritu` は、別行立て数式となります。cases 環境と `\houteisiki` を併用して、本文中の連立方程式にも数式番号を振ることができます。

本文中の連立方程式

```
連立方程式\begin{cases}
2x+3y=-1 & \houteisiki{\label{emcases1}}\%
3x-2y=5 & \houteisiki{\label{emcases2}}
\end{cases}$を解け。
```

→ 連立方程式 $\begin{cases} 2x + 3y = -1 & \dots\dots \textcircled{1} \\ 3x - 2y = 5 & \dots\dots \textcircled{2} \end{cases}$ を解け。

これを見ると、本文中では連立記号というか左括弧が大きく感じられます。それを修正した `emcases` 環境を用意しました。

emcases 環境

```
連立方程式\begin{emcases}
2x+3y=-1 & \houteisiki{\label{emcases3}}\%
3x-2y=5 & \houteisiki{\label{emcases4}}
\end{emcases}$を解け。
```

→ 連立方程式 $\begin{cases} 2x + 3y = -1 & \dots\dots \textcircled{3} \\ 3x - 2y = 5 & \dots\dots \textcircled{4} \end{cases}$ を解け。

`emcases` 環境には、オプションが 2 種類あります。

オプションその 1 は括弧を右につけるためのもので、次のように使います。

emcases 環境

```
連立方程式 \eqref{emcases3}, \eqref{emcases4}を解くと,
\begin{emcases}[r]
x=1\% y=-1
\end{emcases}\%kotae
```

→ 連立方程式 $\textcircled{3}$, $\textcircled{4}$ を解くと, $\left. \begin{matrix} x = 1 \\ y = -1 \end{matrix} \right\} \dots\dots (\text{答})$

第 2 のオプションは `cases` 環境に内包されている `array` 環境の引数を変更します。デフォルトでは、`@{}l{}{\quad}l{}{}` となっており、方程式と数式番号の間に、`\quad` による空白が入ります。これを取り除いて

array 環境の引数指定

```
連立方程式$%begin{emcases}[1][@{}l@{}l@{}]
2x+3y=-1 & %houteisiki{%label{emcases5}}%¥¥
3x-2y=5 & %houteisiki{%label{emcases6}}
%end{emcases}$を解け。
```

→

連立方程式 $\begin{cases} 2x + 3y = -1 & \dots\dots \textcircled{5} \\ 3x - 2y = 5 & \dots\dots \textcircled{6} \end{cases}$ を解け。

もう一段、括弧記号を小さくするのが %baaiwake コマンドです。

%baaiwake

```
連立方程式
%baaiwake{2x+3y=-1 & %houteisiki{%label{emcases7}}}%¥cr
3x-2y=5 & %houteisiki{%label{emcases8}}}%¥cr}$を解け。
```

→

連立方程式 $\begin{cases} 2x + 3y = -1 & \dots\dots \textcircled{7} \\ 3x - 2y = 5 & \dots\dots \textcircled{8} \end{cases}$ を解け。

5.6 ダッシュ付きの数式番号①'

数式番号にダッシュ付きの丸付き数字を使用する方法です。

ダッシュ付き数式番号

```
{%preEqlabel{${%cdots}$}
¥renritu{%
&x+2y=3 %label{Ed-1}}¥¥
& 2x-y=1 %label{Ed-2}}
¥eqref{Ed-1}より
¥begin{equation}
x=3-2y %marudashtag{Ed-1}
¥end{equation}
¥eqref{Ed-2}に代入して
¥begin{align*}
&2(3-2y)-y=1¥¥
¥intertext{ すなわち }
&y=1
¥end{align*}
¥marudashref{Ed-1}
に代入して $x=1$
}
```

→

$\begin{cases} x + 2y = 3 & \dots \textcircled{1} \\ 2x - y = 1 & \dots \textcircled{2} \end{cases}$

①より
 $x = 3 - 2y$... ①'

②に代入して
 $2(3 - 2y) - y = 1$

すなわち
 $y = 1$

①' に代入して $x = 1$

のように、ダッシュのつかない式へのラベルを用いて

`¥marudashtag` でダッシュ付き番号をつけ、
`¥marudashref` で参照します。

本文中の数式に対してダッシュをつけるには `¥houteisiki` のオプション引数に `¥marudashtag` を記述します。

—— 本文中の数式へのダッシュ付き数式番号 ——

```
¥houteisiki{x+2y=1¥label{Ed-3}}を
¥houteisiki[¥marudashtag{Ed-3}]{x=1-2y}
と変形して
```

→ $x + 2y = 1 \dots\dots \textcircled{3}$ を $x = 1 - 2y \dots\dots \textcircled{3}'$ と変形して

5.7 align*環境で数式番号

`align*`, `gather*`環境など数式番号を付けない環境において、特定の式だけは数式番号を付与したいときのために `¥atag` コマンドを用意しました。

—— ¥atag ——

```
¥begin{gather*}
|z-3i|^2=4|z|^2¥¥
(z-3i)(¥conj z+3i)=4z¥conj z¥¥
3(z¥conj z-iz+i¥conj z-3)=0¥¥
(z+i)(¥conj z-i)=4¥¥
¥therefore |z+i|=2¥atag
¥end{gather*}
```

$$|z - 3i|^2 = 4|z|^2$$

$$(z - 3i)(\bar{z} + 3i) = 4z\bar{z}$$

$$3(z\bar{z} - iz + i\bar{z} - 3) = 0$$

$$(z + i)(\bar{z} - i) = 4$$

$$\therefore |z + i| = 2$$

\dots\dots\dots \textcircled{1}

5.8 等号の縦揃え

長い式の変形を表すとき、左端に等号を置いて式を並べます。

—— =による左揃え ——

```
¥begin{align*}
&|x|^2¥¥
&=|x|^2¥¥
&=|x|^2
¥end{align*}
```

$$|x|^2$$

$$= |x|^2$$

$$= |x|^2$$

では、一番上の行の位置に不満があります。`¥phantom` を用いて、等号分だけ右にずらします。

— `\phantom=` を用いる —

```
\begin{align*}
&\phantom{=}|x|^2\phantom{=}
&=|x|^2\phantom{=}
&=|x|^2
\end{align*}
```

$$\begin{aligned} &|x|^2 \\ &=|x|^2 \\ &=|x|^2 \end{aligned}$$

等号の幅はずれましたが、1行目と2行目の先頭がいぜん不揃いです。=の右に{}を附加します。

— `\phantom{={}}` と修正 —

```
\begin{align*}
&\phantom{={}}|x|^2\phantom{=}
&=|x|^2\phantom{=}
&=|x|^2
\end{align*}
```

$$\begin{aligned} &|x|^2 \\ &=|x|^2 \\ &=|x|^2 \end{aligned}$$

まだ少しずれています。透明の等号の属性を `\mathrel` と指定すれば

— `\mathrel{\phantom=}` と修正 —

```
\begin{align*}
&\mathrel{\phantom{=}}|x|^2\phantom{=}
&=|x|^2\phantom{=}
&=|x|^2
\end{align*}
```

$$\begin{aligned} &|x|^2 \\ &=|x|^2 \\ &=|x|^2 \end{aligned}$$

式の先端がきれいに揃います。これはよく使いますから

```
\def\dumyreq{\mathrel{\phantom{=}}}
```

と定義することにしました。

— `\dumyreq` —

```
\begin{align*}
&\dumyreq|x|^2\phantom{=}
&=|x|^2\phantom{=}
&=|x|^2
\end{align*}
```

$$\begin{aligned} &|x|^2 \\ &=|x|^2 \\ &=|x|^2 \end{aligned}$$

別法です。等号の右に&を置くのもあります。ただし、

— `&` の位置 —

```
\begin{align*}
&|x|^2\phantom{=}
&=&|x|^2\phantom{=}
&=&|x|^2
\end{align*}
```

$$\begin{aligned} &|x|^2 \\ &=&|x|^2 \\ &=&|x|^2 \end{aligned}$$

では、等号と右辺との間隔が詰まりすぎです。

```
&の位置
\begin{align*}
&|x|^2
= {}&|x|^2
= {}&|x|^2
\end{align*}
```

$$\begin{aligned} &|x|^2 \\ = &|x|^2 \\ = &|x|^2 \end{aligned}$$

とすればよいのですが、すべての行に={}&とするのは煩雑ですね。

5.9 \item 直後の縦揃え数式

5.9.1 aligned 環境

\item 直後に別行立て数式を配置すると

```
\item 直後の別行立て数式
\begin{enumerate}[(1)]
\item
\begin{align*}
y &= (a+b)^2
&= a^2 + 2ab + b^2
\end{align*}
\end{enumerate}
```

$$\begin{aligned} (1) \quad &y = (a + b)^2 \\ &= a^2 + 2ab + b^2 \end{aligned}$$

この結果について次のような不満の声を耳にします。

項目番号と数式が

- (1) 縦方向にずれている(改行されている)
- (2) 横方向も空きすぎである。

これは、不満を述べる方が心得違いをしています。

(1) に対しては、

別行立て数式は改行する

(2) に対しては、

別行立て数式の横位置は

センタリング (fleqn オプションがない場合)

左寄せ (fleqn オプション付きの場合で、

インデント量は \mathindent)

という、別行立て数式の仕様に従って配置されているまでです。

対応策のひとつは別行立てではなく、本文中数式モードで済ませる、というのがあります。(この節は、井汲 景太 さんからご教示いただきました。)

amsmath.sty で定義されている aligned 環境を配置オプション [t] を付加して用います。

aligned 環境

```
¥begin{enumerate}[(1)]
  ¥item
    ¥begin{aligned}[t]
      y&=(x+1)^2¥¥
      &=x^2+2x+1
    ¥end{aligned}$
¥end{enumerate}
```

$$(1) \quad \begin{aligned} y &= (x+1)^2 \\ &= x^2 + 2x + 1 \end{aligned}$$

ただし、この環境は別行立てではありませんから、数式番号はつきません。
数式番号など、別行立てにこだわるなら、次節で紹介する ¥itemtopmath を用いるのも一法です。

5.9.2 ¥itemtopmath

別行立て数式は、説明文があって、それに続くもの、というのが正しい使い方でしょう。

別行立て数式の前には説明文

```
¥begin{enumerate}[(1)]
  ¥item 説明文の後に別行立て数式
    ¥begin{align*}
      y&=(a+b)^2¥¥
      &=a^2+2ab+b^2
    ¥end{align*}
¥end{enumerate}
```

$$(1) \quad \begin{aligned} &\text{説明文の後に別行立て数式} \\ y &= (a+b)^2 \\ &= a^2 + 2ab + b^2 \end{aligned}$$

にもかかわらず、なんとかならないか、という要望は却下すべきものではありませんが、余りに多いので、emath パッケージでは、¥itemtopmath というコマンドを用意してあります。

¥itemtopmath

```
¥begin{enumerate}[(1)]
  ¥item ¥itemtopmath
    ¥begin{align*}
      y&=(a+b)^2¥¥
      &=a^2+2ab+b^2
    ¥end{align*}
¥end{enumerate}
```

$$(1) \quad \begin{aligned} y &= (a+b)^2 \\ &= a^2 + 2ab + b^2 \end{aligned}$$

すなわち、¥item 直後に

```
¥itemtopmath
```

というコマンドを発行して、別行立て数式を上方に引っ張り上げます。

5.9.2.1 左インデントの調整 つぎは、項目番号と数式の横方向の間隔調整です。これは、その文書に

fleqn オプション

がついているか否かで対応が異なります。

まずは、fleqn オプションがついている場合です。この場合は、数式行の左インデントを

```
¥mathindent=0pt
```

などと指定します。(その影響範囲を限定するため、グルーピングをしておくなどの対策が必要です。)

左インデント (1)

```
¥begin{enumerate}[(1)]
  ¥item ¥itemtopmath
    ¥mathindent=0pt¥relax
    ¥begin{align*}
      y&=(a+b)^2¥¥
      &=a^2+2ab+b^2
    ¥end{align*}
¥end{enumerate}
```

$$(1) \quad y = (a + b)^2 \\ = a^2 + 2ab + b^2$$

fleqn オプションをつけない文書の場合は、局所的に fleqn オプションがついた状態にします。すなわち、align 環境を fleqnon 環境に、左インデントをオプションで与えたものにくるみます：

```
¥begin{fleqnon}[Opt]
  ¥begin{align}
    .....
    .....
  ¥end{align}
¥end{fleqnon}
```

という具合です。

左インデント (2)

```
¥begin{enumerate}[(1)]
  ¥item ¥itemtopmath
    ¥begin{fleqnon}[Opt]
      ¥begin{align*}
        F(x)&=(x+b)^2¥¥
        &=x^2+2bx+b^2
      ¥end{align*}
    ¥end{fleqnon}
¥end{enumerate}
```

$$(1) \quad F(x) = (x + b)^2 \\ = x^2 + 2bx + b^2$$

5.9.2.2 縦方向の補正 さて、数式が複雑になると縦方向の補正量が `\itemtopmath` では間に合わなくなります。

数式が複雑になると

```
\medskip
\begin{enumerate}[(1)]
  \item \itemtopmath
    \mathindent=0pt\relax
    \begin{align*}
      y&=\int (ax+b)^2 dx \\
      &=\frac{1}{3a}(ax+b)^3 + C
    \end{align*}
\end{enumerate}
```

$$(1) \quad y = \int (ax + b)^2 dx \\ = \frac{1}{3a}(ax + b)^3 + C$$

この場合は、1行目の数式で一番背の高いもの — この場合は積分記号 `\int` — を `\itemtopmath` の [...] オプションに与えます。

`\itemtopmath` の [...] オプション

```
\medskip
\begin{enumerate}[(1)]
  \item \itemtopmath[\int]\relax
    \mathindent=0pt\relax
    \begin{align*}
      y&=\int (ax+b)^2 dx \\
      &=\frac{1}{3a}(ax+b)^3 + C
    \end{align*}
\end{enumerate}
```

$$(1) \quad y = \int (ax + b)^2 dx \\ = \frac{1}{3a}(ax + b)^3 + C$$

縦方向の位置を、更に細かく調整するには

`\itemtopmath` の <...> オプションを利用します。

5.9.2.3 `\itemtopmath` の書式 最後に、`\itemtopmath` の書式です。

```
\itemtopmath<#1>[#2]
  #1 : 微調整量 (単位つき) +で上に、-で下に移動
  #2 : 1行目の高さが高いとき、その部分を与える
      ( \emtokena に保存される)
```

6 図の取り込み

この節のコマンド類は、`emath.sty` ではなく、`emathMw.sty` に含まれています。使用する際は

```
\usepackage{emathMw}
```

としておかねばなりません。

6.1 図の周りへの回り込み

6.1.1 mawarikomi 環境

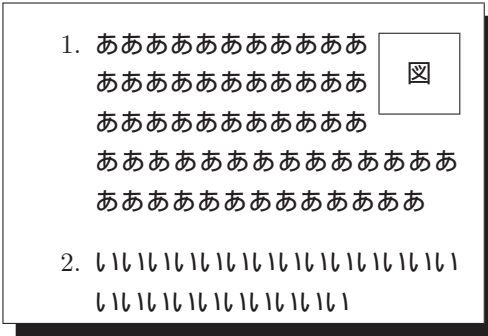
図のまわりにテキストを回り込ませる環境に、`wrapfigure` があります。しかし、この環境は `enumerate` 環境など `list` 環境下では使用できません。

そこで、`list` 環境と併用するために `mawarikomi` 環境を作ってみました。

その使用例を見ていただきましょう。

mawarikomi 環境

```
\begin{enumerate}
\item
\begin{mawarikomi}{30pt}{%
  \begin{picture}(30,30)
    \put(0,0){%
      \framebox(30,30){図}}
    \end{picture}}
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
\end{mawarikomi}
\item しいしいしいしいしい
      しいしいしいしいしい
\end{enumerate}
```



基本的な使用法は

```
\begin{mawarikomi}{図の横幅}{図の記述}
  回り込ませるテキスト部の記述
\end{mawarikomi}
```

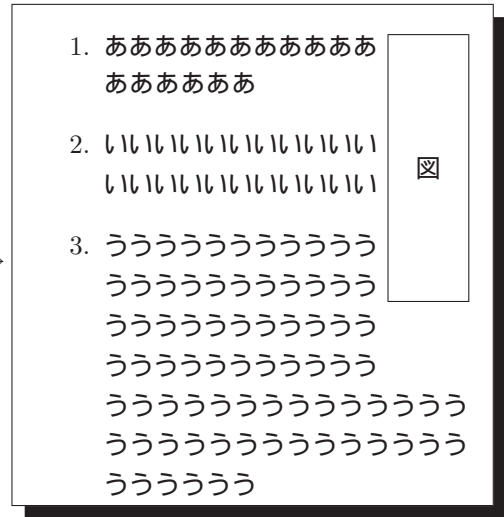
となっています。細かな調整を行うためのオプション引数については後述します。

6.1.2 mawarikomi*環境

図が大きくて複数の問題で回り込みを行わせるには、

複数の問題で回り込み

```
¥begin{enumerate}
¥item
¥begin{mawarikomi}{30pt}{%
  ¥begin{picture}(30,100)
  ¥put(0,0){%
  ¥framebox(30,100){図}}
¥end{picture}}
ああああああああ
ああああああああ
¥end{mawarikomi}
¥item
¥begin{mawarikomi*}
いいいいいいいいいい
いいいいいいいいいい
¥end{mawarikomi*}
¥item
¥begin{mawarikomi*}
うううううううううう
うううううううううう
うううううううううう
うううううううううう
うううううううううう
うううううううううう
¥end{mawarikomi*}
¥end{enumerate}
```



この場合、回り込みを行うタイミングがずれてきています。回り込みの行数は、図のサイズから計算で求めています，この場合は、enumerate 環境において、¥item 間の縦間隔が大きめにとられる関係で計算値とずれてきました。この修正法は後述します。(p.46, § 6.1.4)

6.1.3 mawarikomi 環境内に list 環境

mawarikomi 環境内に list 環境を入れることができます。

— mawarikomi 内に list —

```
¥begin{enumerate}
¥item
¥begin{mawarikomi}{30pt}{¥}
  ¥begin{picture}(30,100)
    ¥put(0,0){¥}
    ¥framebox(30,100){¥}
  ¥end{picture}
  次の問いに答えよ。
  ¥begin{enumerate}[(1)]
    ¥item アアアアアアア
      アアアアアアアア
      アアアアアアアア
      アアアアアアアア
    ¥item イイイイイイイ
      イイイイイイイイ
      イイイイイイイイ
      イイイイイイイイ
      イイイイイイイイ
      イイイイイイイイ
    ¥end{enumerate}
  ¥end{mawarikomi}
¥item
  いいいいいいいい
  いいいいいいいい
¥end{enumerate}
```




1. 次の問いに答えよ。

(1) アアアアアアアア
アアアアアアアア
アアアアアアアア
アアアアアアア

(2) イイイイイイイイ
イイイイイイイイ
イイイイイイイイ
イイイイイイイイイイイ
イイイイイイイイイイ

2. いいいいいいいい
いいいいいいいい



6.1.4 回り込みの行数指定オプション

回り込みを行う行数は、図の高さから計算で求めています。しかし、テキスト部分の状態によっては不適切な場合もできます。そこで mawarikomi 環境に [...] オプションで行数の指定ができるようにしてあります。45 ページの例では、計算では 8 行となっていますが、テキスト部分の状況から 7 行とした方が良いでしょう。

下の例では、

```
¥begin{mawarikomi}[7]{30pt}{¥}
```

としています。

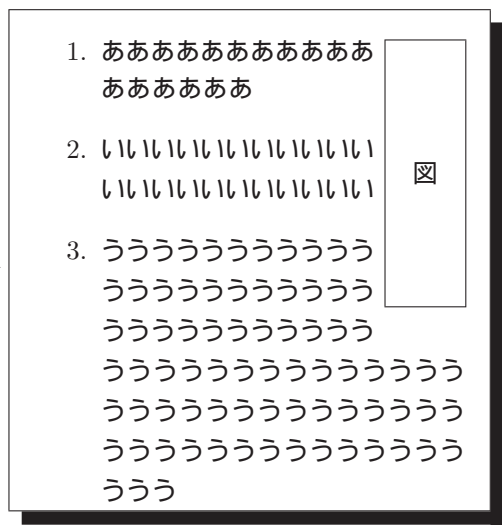
なお、この指定法は相対指定も可能で、数値に '+' '-' を付けることにより計算で得られた行数を増減させます。下の例を

```
¥begin{mawarikomi}[-1]{30pt}{¥}
```

としても同じ結果が得られます。

回り込み行数の調整

```
¥begin{enumerate}
¥item
¥begin{mawarikomi}[7]{30pt}{%
  ¥begin{picture}(30,100)
  ¥put(0,0){%
  ¥framebox(30,100){図}}
¥end{picture}}
ああああああああ
ああああああああ
¥end{mawarikomi}
¥item
¥begin{mawarikomi*}
いはいはいはいはいはいはいはい
いはいはいはいはいはいはいはい
¥end{mawarikomi*}
¥item
¥begin{mawarikomi*}
うううううううううううう
うううううううううううう
うううううううううううう
うううううううううううう
うううううううううううう
うううううううううううう
¥end{mawarikomi*}
¥end{enumerate}
```



6.1.5 テキストと図の横間隔

テキスト部分と図の部分との横間隔はデフォルトで

```
¥newlength{¥mawarikomisep}%
¥setlength{¥mawarikomisep}{2pt}%
```

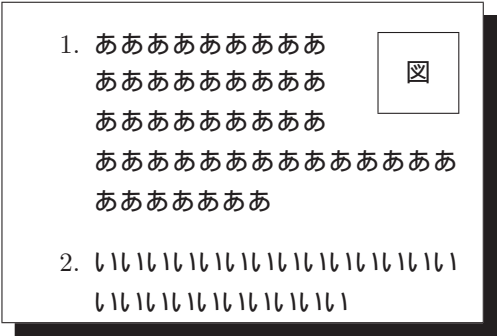
としてあり、その2倍 4pt が空くようにしてあります。
次の例では、

```
¥setlength{¥mawarikomisep}{1zw}%
```

として、その2倍 2zw の空きを作ってみました。

テキストと図の横間隔

```
¥begin{enumerate}
¥item
¥setlength{¥mawarikomisep}{%
  1zw}%
¥begin{mawarikomi}{30pt}{%
  ¥begin{picture}(30,30)
  ¥put(0,0){%
    ¥framebox(30,30){図}}
  ¥end{picture}}
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
¥end{mawarikomi}
¥item 丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
  丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
¥end{enumerate}
```

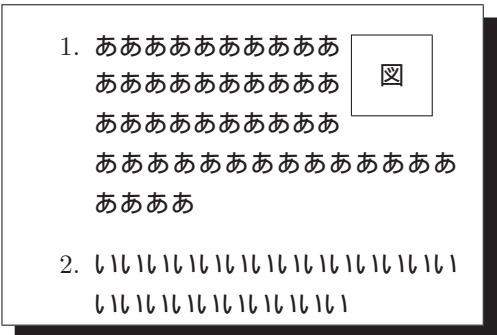


6.1.6 図の位置の微調整

図の位置を細かく調整したいことがあります。そのために mawarikomi 環境に (x,y) オプションを用意しました。一例です。

図の位置 (調整前)

```
¥begin{enumerate}
¥item
¥begin{mawarikomi}{40pt}{%
  ¥begin{picture}(30,30)
  ¥put(0,0){%
    ¥framebox(30,30){図}}
  ¥end{picture}}
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
¥end{mawarikomi}
¥item 丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
  丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
¥end{enumerate}
```



図をもう少し右上に動かしたいですね。mawarikomi 環境に (5pt,2pt) オプションを付加してみました。


図の位置 (調整後)

```

¥begin{enumerate}
¥item
¥begin{mawarikomi}%
  (5pt,2pt){40pt}{%
    ¥begin{picture}(30,30)
      ¥put(0,0){%
        ¥framebox(30,30){図}}
    ¥end{picture}}
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
¥end{mawarikomi}
¥item しいしいしいしいしい
      しいしいしいしいしい
¥end{enumerate}

```



1. ああああああああああ
 ああああああああああ 
 ああああああああああ
 ああああああああああ
 ああああ

2. しいしいしいしいしいしい
 しいしいしいしいしいしい

6.1.7 段落途中からの回り込み

<n> オプションで段落の始めから n 行は回り込みをしないように指定することができます。下の例は<1>としたものです。


<n>オプション

```

¥begin{enumerate}
¥item
¥begin{mawarikomi}<1>{30pt}{%
  ¥begin{picture}(30,30)
    ¥put(0,0){%
      ¥framebox(30,30){図}}
    ¥end{picture}}
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
  ああああああああああ
¥end{mawarikomi}
¥item しいしいしいしいしいしい
      しいしいしいしいしいしい
¥end{enumerate}

```



1. ああああああああああ
 ああああああああああ 
 ああああああああああ
 ああああああああああ
 ああああああああああ
 ああああああああああ
 ああああああああああ

2. しいしいしいしいしいしいしい
 しいしいしいしいしいしい

ただしテキスト部に数式があるなど行ピッチが変わると、図の位置の調整が必要になります。下の例では、2行目の数式を回り込みの対象外にするため、<2>オプションをつけました。テキスト部はお望み通り3行目から回り込みが始まっていますが、図の位置が不適切です。

図位置の調整が必要

```

\begin{enumerate}
\item
\begin{mawarikomi}<2>{30pt}{%
  \begin{picture}(30,30)
    \put(0,0){%
      \framebox(30,30){図}}
    \end{picture}}
ああああああああああ
ああああああああ
\mathbunsuu{1}{1+\mathbunsuu{1}{%
1+\mathbunsuu{1}{x}}}$
ああああああああああ
ああああああああああ
ああああああああああ
ああああああああああ
\end{mawarikomi}
\item ㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱ
      ㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱ
\end{enumerate}

```



1. あああああああああああ
 ああああああ $\frac{1}{1 + \frac{1}{1 + \frac{1}{x}}}$ あ

ああああああああああ
 あああああああああ
 あああああああああ

2. ㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱ
 ㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱㄱ

では、(0,-24pt) オプションで図を下に動かしてみます。


図位置の調整後

```

\begin{enumerate}
\item
\begin{mawarikomi}<2>%
(0,-24pt){30pt}{%
\begin{picture}(30,30)
\put(0,0){%
\framebox(30,30){図}}
\end{picture}}
ああああああああああ
ああああああああ
\$\bunsuu{1}{1+\bunsuu{1}{x}}\$
ああああああああああ
ああああああああああ
ああああああああああ
\end{mawarikomi}
\item 111111111111111111
111111111111111111
\end{enumerate}

```



1. あああああああああああああああ
ああああああ $\frac{1}{1 + \frac{1}{1 + \frac{1}{x}}}$ あ
ああああああああああああ 
ああああああああああああ
ああああああああああああ

2. 111111111111111111111111111111
111111111111111111111111111111

6.1.8 \caption の使用

mawarikomi 環境内では、\caption を使用できません。そこで、

- Fmawarikomi 環境：図番号を使用
- Tmawarikomi 環境：表番号を使用

なる 2 つの環境を用意してあります。

下の例では、Tmawarikomi 環境、Fmawarikomi 環境内で \caption を使用しています。

```

¥begin{enumerate}
¥item ¥begin{Tmawarikomi}(0,5pt){9zw}{%
  ¥caption{表の例}¥label{T1}¥hfil
  ¥begin{tabular}{|c|c|c|}¥hline
    aaa & bbb & ccc ¥¥¥hline
    1 & 2 & 3 ¥¥¥hline
  ¥end{tabular}}
  第 1 問 右の表 ¥ref{T1}において
  あああああああああああああああああああああああああああああ
  あああああああああああああああああああああああああああああああ
  あああああああああああああああああああああああああああああああ
  あああああああああああああああああ
¥end{Tmawarikomi}
¥item ¥begin{Fmawarikomi}{60pt}{%
  ¥unitlength1pt%
  ¥begin{picture}(60,30)%
    ¥put(0,0){¥line(2,1){60}}%
  ¥end{picture}%
  ¥caption{図の例}¥label{Fig1}}
  第 2 問 右の図 ¥ref{Fig1}において ,
  11111111111111111111111111111111111111111111111111111111111111
  11111111111111111111111111111111111111111111111111111111111111
  11111111111111111111111111111111111111111111111111111111111111
  11111111111111111111111111111111111111111111111111111111111111
  11111111111111111111111111111111111111111111111111111111111111
¥end{Fmawarikomi}
¥end{enumerate}

```

1. 第 1 問 右の表 1 において
 あああああああああああああああ
 あああああああああああああああああああああああああああああああ
 あああああああああああああああああああああああああああああああ
 あああああああああああああああああああああああああああああああ
 あああああああああああああああああ

表 1: 表の例

aaa	bbb	ccc
1	2	3

2. 第 2 問 右の図 1 において ,
 11
 11
 11
 11
 11
 11

図 1: 図の例

なお別法として、mawarikomi 環境において、

```
¥fgcaption  
¥tbcaption
```


コマンドで ¥caption の代行をさせることも可能です。ただし、これらのコマンドは emathCap.sty で定義されていますから、使用する際は

```
¥usepackage{emathCap}
```

としておく必要があります。

6.1.9 書式

mawarikomi 環境の書式です。



```
¥begin{mawarikomi}<#1>[#2](#3,#4)#5#6
```

#1 : 段落当初の回り込みをしない行数
#2 : 回り込み行数 (相対指定可)
先頭に '1' を付けたときは図を左に配置
(#3,#4) : 図の位置修正ベクトル
#3 : (>0) 右, (<0) 左
#4 : (>0) 上, (<0) 下
#3,#4 共に単位必須 (0 のみは単位不要)
#5 : 図の横幅 (単位必須)
#6 : 図, 表など

関連パラメータ
¥mawarikomisep テキストと図の間隔 (デフォルト値 2pt)
この 2 倍の空きができます。

6.1.10 図の左配置

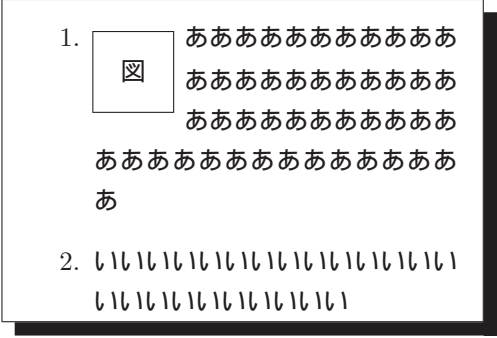
行数指定オプション [#2] で、引数の先頭に '1' (left) をつけると図などが左に配置されます。

図の左配置

```

¥begin{enumerate}
¥item
¥begin{mawarikomi}[1]{30pt}{%
  ¥begin{picture}(30,30)
  ¥put(0,0){%
  ¥framebox(30,30){図}}
¥end{picture}}
ああああああああああ
ああああああああああ
ああああああああああ
ああああああああああ
¥end{mawarikomi}
¥item 丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
      丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
¥end{enumerate}

```



6.1.11 図幅指定の省略

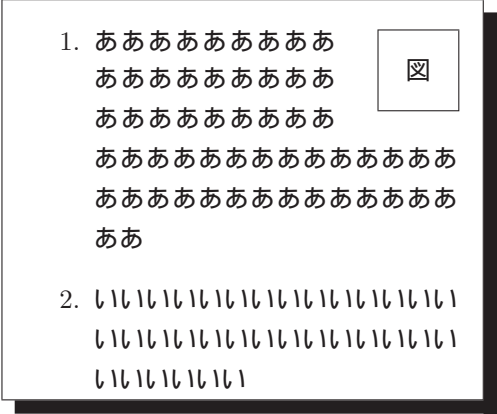
図幅を指定する引数を空にすると、 $\text{T}_{\text{E}}\text{X}$ が認識する図幅で代行されます。ただし、図の部分複数の段落を持つときは省略できません。したがって、図（表）番号を付与するときはこの機能は使えません。

図幅省略

```

¥begin{enumerate}
¥item
¥begin{mawarikomi}{}{%
  ¥begin{picture}(30,30)
  ¥put(0,0){%
  ¥framebox(30,30){図}}
¥end{picture}}
ああああああああああああああああああああ
ああああああああああああああああああああ
ああああああああああああああああああああ
¥end{mawarikomi}
¥item 丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
      丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨丨
¥end{enumerate}

```



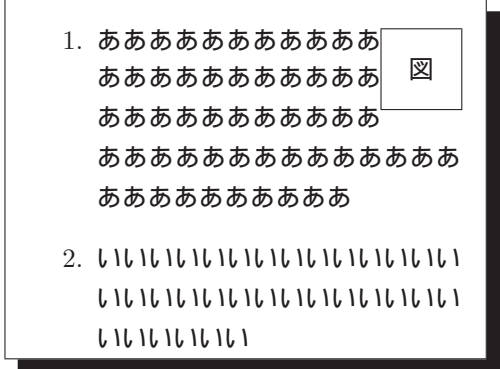
この場合、図とテキスト部の間隔は 16pt としてあります。増減したいときは、増減する数値を図幅指定の引数に、‘+’あるいは‘-’の符号付で与えます。極端な例です。

— 図幅省略時の本文と図の間隔 —

```

%begin{enumerate}
%item
%begin{mawarikomi}{-16pt}{%
  %begin{picture}(30,30)
  %put(0,0){%
  %framebox(30,30){図}}
  %end{picture}}
  あああああああああああああああああ
  あああああああああああああああああ
  あああああああああああああああああ
%end{mawarikomi}
%item しいしいしいしいしいしいしいしい
  しいしいしいしいしいしいしいしいしい
%end{enumerate}

```



文書全体でこの間隔を変更したいときは、`%mawarikomikankaku` の値を変更します。デフォルトは

```
%mawarikomikankaku=8pt
```

となっており、その 2 倍の 16pt が空きます。

文書全体でこの間隔を変更したければ、プリアンブルで

```
%mawarikomikankaku=10pt
```

などと、右辺値を変更します。(この変更で 2 倍の 20pt が空くことになります。)

6.1.12 テキスト部が短いとき

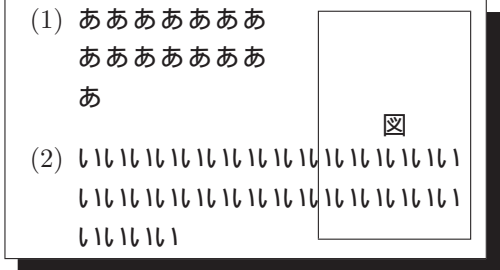
図に比して、テキスト部が短く、回り込みが始まる前にテキスト部が終わってしまうときの処理です。まずは、具合の悪い例をご覧ください。

— テキストが短いとき —

```

%begin{enumerate}[(1)]
%item %begin{mawarikomi}{-}{%
  %begin{zahyou*}[ul=5mm]%
  (0,4)(0,6)
  %framebox(4,6){図}}
  %end{zahyou*}}
  あああああああああああああああ
  %end{mawarikomi}
%item しいしいしいしいしいしいしいしい
  しいしいしいしいしいしいしいしい
  しいしいしいしいしいしいしいしい
%end{enumerate}

```



6.1.13 改ページとの関係 (1)

mawarikomi 環境の図が、ページの境界にさしかかると、おかしなことがおきる可能性があります。完全に解決することはできませんが、いくつかの例を見ていただきましょう。

まず、p.58 ~ p.59 (この部分、背景色を ivory としています) では、問題なしですが、

mawarikomi 環境より前の部分が延びてくると，図は次ページに送られ，p.61 の下部には大きな空白ができます。（背景色は lavender）

図が次ページに送られるのは仕方がないとして、テキスト部は一部を繰り上げて、空白ができな
いようにしたければ

```
\begin{mawarikomi}<0>{}{%
```

と、mawarikomi 環境に<0>オプションをつけます（背景色は skyblue）。その効用は：

6.1.14 改ページとの関係 (2)

mawarikomi 環境部がページの境目に差し掛かると、いろいろ厄介な現象が生じます。

その一事例です。

この文書の 67~69 ページを見てください (この部分は黄色の背景色がついています)。p.68 は、たった 2 行で改ページされています。p.69 に配置されている mawarikomi 環境は、p.68 の続きに配置できそうなのに、なぜか改ページされています。

その理由の説明と、この現象を回避する一手段を見ていただきましょう。

まずは、理由の推測です。

- 1) emathMw.sty は、図がページの境目にひっかかる場合、次ページに送る仕様となっています。¹
- 2) この文書では、p.68 の終りで図がページの境目にひっかかるので、図の前で改ページされます。この時点では、フロート (脚注) はないものとして判断が行われます。
- 3) そのあとで、フロートが入ることにより、p.68 の終りの

```
¥section{mawarikomi を次ページに送らなくても良いのに}  
¥repeatchar{!}{30}
```

の 2 行が次ページに送られますが、図の前の改ページは取り消されることはありません。

ということで、p.68 は、2 行だけということとなってしまったのです。

対症療法です。この文書では、p.69 に移された mawarikomi 環境は、ページ境目にきて改ページをしないようにします。そのための環境が、新設された Pmawarikomi 環境です。

p.70~p.71 では (この部分の背景色はシアンです)、この新設環境を用いています。

¹emathMw.sty を [samepage] オプション付でロードすると、次ページ送りの処理を禁止することもできます。この場合、以下の議論は無意味です。

mawarikomi を次ページに送らなくても良いのに

!!

6.2 複数の図の周りへの回り込み

横幅の異なる複数の図の周りへテキストを回り込ませるのは面倒です。現時点では、自動ではなく手動で行う方法しかありません。

- (1) 図を配置する。
- (2) 図の周りに、テキストを回り込ませる。

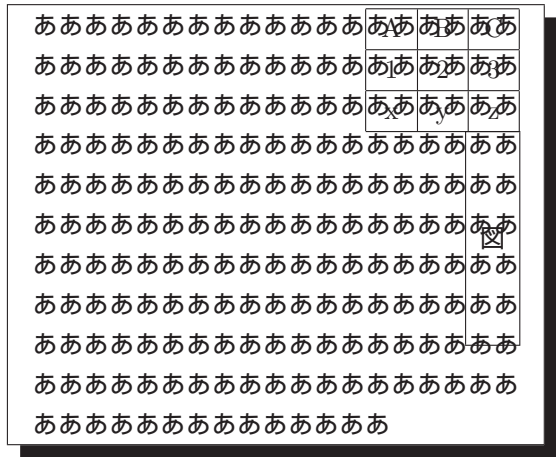
という手順で実現することにします。

6.2.1 図の配置 — zuhaiti 環境

まずは、図の配置です。

図の配置

```
¥begin{zuhaiti}
  ¥haiti{%
    ¥begin{tabular}{|c|c|c|}
      ¥hline
      A & B & C ¥¥¥hline
      1 & 2 & 3 ¥¥¥hline
      x & y & z ¥¥¥hline
    ¥end{tabular}}
  ¥haiti{%
    ¥begin{picture}(20,80)%
      ¥framebox(20,80){図}
    ¥end{picture}}
¥end{zuhaiti}%
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
```



段落の先頭で zuhaiti 環境を置き、その中で ¥haiti コマンドで図表などを配置していきます。

6.2.2 枠の設定 — `¥wakudori`

この時点では、文字が図にかぶっています。この文字を図の周りに回りこませます。

そのためには、図の横幅を知らねばなりません。この場合は、`zahyou` 環境の `¥unitlength` と x の範囲から計算で求められますが、`TEX` が認識している図の横幅がタイプセットした際にできるログファイルに記録されています：

図幅=57.8056pt

図幅=20.0pt

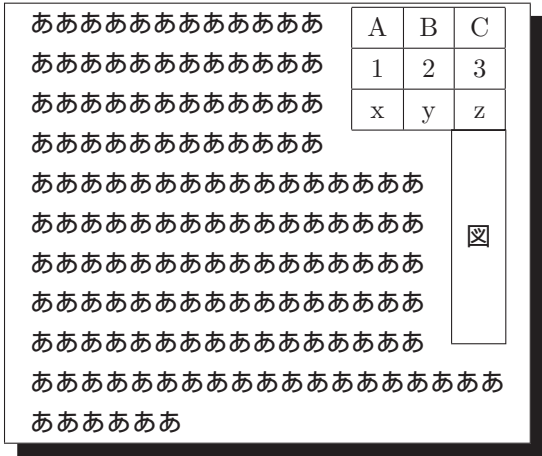
図と文字部分との間隔を 10pt として、

初めの 4 行は 67.8pt、

次の 5 行は、30pt 行の長さを短くしてみます。

枠の設定

```
¥begin{zuhaiti}
  ¥haiti{%
    ¥begin{tabular}{|c|c|c|}¥hline
      A & B & C ¥¥¥hline
      1 & 2 & 3 ¥¥¥hline
      x & y & z ¥¥¥hline
    ¥end{tabular}}
  ¥haiti{%
    ¥begin{picture}(20,80)%
      ¥framebox(20,80){図}
    ¥end{picture}}
¥end{zuhaiti}%
¥wakudori{4*67.8pt;5*30pt}
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああ
```



新設コマンド

```
¥wakudori{4*67.8pt;5*30pt}
```

で、4 行分右に 67.8pt の空を作り、引き続いて 5 行分 30pt の空を作ります。すなわち `¥wakudori` の引数には

行数*横幅

を ; 区切りで並べます。なお、このコマンドは zuhaiti 環境の外に記述します。

6.2.3 図の左配置

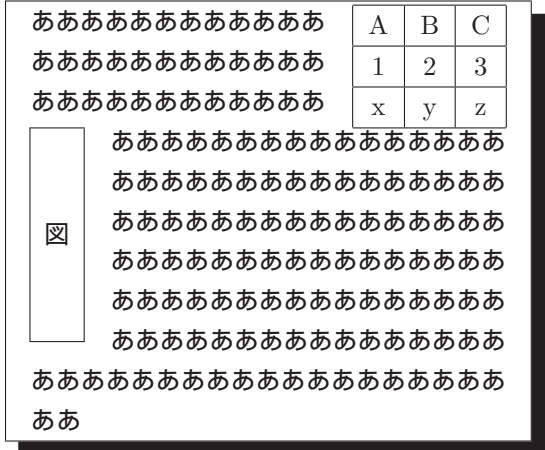
2 番目の図を左に配置してみましょう。

図の左配置

```

¥begin{zuhaiti}
  ¥haiti{%
    ¥begin{tabular}{|c|c|c|}¥hline
      A & B & C ¥¥¥hline
      1 & 2 & 3 ¥¥¥hline
      x & y & z ¥¥¥hline
    ¥end{tabular}}
  ¥haiti[1]{%
    ¥begin{picture}(20,80)%
      ¥framebox(20,80){図}
    ¥end{picture}}
¥end{zuhaiti}%
¥wakudori{3*67.8pt;6*(30pt,0pt)}
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ

```



すなわち，図を左配置するには，¥haiti コマンドに [1] オプションをつけます。

また，左配置した図を回り込ませるには，¥wakudori の引数を

行数*(左空きのサイズ, 右空きのサイズ)

の形で与えます。

一見良さそうですが，よく見ると 4 行目が第 1 の図にかぶっています。この行は 2 つの図両方にかかっているのです。

その修正法を 2 つ紹介します。

6.2.4 図の位置の微調整

修正法その 1 は、2 番目の図を 1 行分下に動かし、4 行目は第 1 の図に対して回り込ませます。

図の位置を調整

```

¥begin{zuhaiti}
  ¥haiti{%
    ¥begin{tabular}{|c|c|c|}¥hline
      A & B & C ¥¥¥hline
      1 & 2 & 3 ¥¥¥hline
      x & y & z ¥¥¥hline
    ¥end{tabular}}
  ¥haiti[1]%
    (Opt,-¥baselineskip){%
    ¥begin{picture}(20,80)%
      ¥framebox(20,80){図}
    ¥end{picture}}
¥end{zuhaiti}%
¥wakudori{4*67.8pt;6*(30pt,0pt)}
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ

```

ああああああああああああ	A	B	C
ああああああああああああ	1	2	3
ああああああああああああ	x	y	z

ああああああああああああああああ
 あああああああああああああああああ
 あああああああああああああああああ
図 あああああああああああああああ
 あああああああああああああああああ
 あああああああああああああああああ
 あああああああああああああああああ
 あああああああああああああああああ
 あああああああああああああああああ
 あああああああ

すなわち、図の位置をずらすには、¥haiti コマンドに (dx,dy) オプションをつけます。dx, dy は単位を伴った長さで、

- dx は左右の動きで
 - dx>0 のとき右へ、dx<0 のとき左へ動きます。
- dy は上下の動きで
 - dy>0 のとき上へ、dy<0 のとき下へ動きます。

ここでは、¥haiti[1](Opt,-¥baselineskip){... により図を 1 行分下に動かしています。

6.2.5 行の左右に枠

修正法その2です。2つの図にかかる行の左右双方に枠をあげてしまう、という方法です。

— 行の左右双方に枠 —

```

\begin{zuhaiti}
  \haiti{%
    \begin{tabular}{|c|c|c|}\hline
      A & B & C \\ \hline
      1 & 2 & 3 \\ \hline
      x & y & z \\ \hline
    \end{tabular}}
  \haiti[l]{%
    \begin{picture}(20,80)%
      \framebox(20,80){図}
    \end{picture}}
\end{zuhaiti}%
\wakudori{%
  3*67.8pt;
  1*(30pt,67.8pt);
  5*(30pt,0pt)}
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ

```

	A	B	C
	1	2	3
	x	y	z

ああああああああ
 あああああああああああああああ
 あああああああああああああああ
 あああああああああああああああ
 あああああああああああああああ
 あああああああああああああああ
 あああああああああああああああ
 あああああああああああああああ
 あああああああああああ

6.2.6 list との併用

enumerate など list 系の環境下で用いるときの注意です。

list と併用 – 失敗

```

%begin{enumerate}
%item
%begin{zuhaiti}
  %haiti{%
    %begin{tabular}{|c|c|c|}%hline
    A & B & C %%hline
    1 & 2 & 3 %%hline
    x & y & z %%hline
    %end{tabular}}
    %haiti[1]%
      (0pt,-%baselineskip){%
        %begin{picture}(20,80)%
          %framebox(20,80){図}
          %end{picture}}
    %end{zuhaiti}%
    %wakudori{4*67.8pt;6*(30pt,0pt)}
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ
    あああああああああああああああ

    11111111111111111111111111111111
    11111111111111111111111111111111
    11111111111111111111111111111111
    11111111111111111111111111111111

    %item
    うううううううううううううううう
    うううううううううううううううう
    うううううううううううううううう
    うううううううううううううううう

    %end{enumerate}

```

1. あああああああああ
 あああああああああ
 あああああああああ
 あああああああああ

A	B	C
1	2	3
x	y	z

図 あああああああああああああ
 あああああああああああああ
 あああああああああああああ
 あああああああああああああ
 あああああああああああああ
 あああああああああああああ
 あああああああああああああ
 あああああああ

111111111111111111
 111111111111111111
 111111111111111111
 111111111111111111
 11111111111111111111111111111111
 11111111111111111111111111111111
 11111111111111111111111111111111

2. うううううううう
 うううううううう
 うううううううう
 うううううううう
 うううううううううううう
 うううううううううううう
 うううううう

あらま。これでは困ります。%wakudori は TeX の %parshape を用いています。普通は段落が変

われは `¥parshape` は解消してしまいましたが，`list` 環境下では，段落が変わっても `¥parshape` を使い続けることになっています。

対応法は `¥wakudori` で設定した枠どりが終わった次の段落の先頭で

```
¥wakudori{}
```

と枠どりの終了を宣言し，`¥parshape` を `list` 環境のものに戻しておくことです。

list と併用 - 成功

```
¥begin{enumerate}
¥item
¥begin{zuhaiti}
¥haiti{%
¥begin{tabular}{|c|c|c|}¥hline
A & B & C ¥¥¥hline
1 & 2 & 3 ¥¥¥hline
x & y & z ¥¥¥hline
¥end{tabular}}
¥haiti[1]%
(Opt,-¥baselineskip){%
¥begin{picture}(20,80)%
¥framebox(20,80){☒}
¥end{picture}}
¥end{zuhaiti}%
¥wakudori{4*67.8pt;6*(30pt,0pt)}
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ

¥wakudori{}%
いいいいいいいいいいいいいいいい
いいいいいいいいいいいいいいいい

¥item
うううううううううううううううう
うううううううううううううううう
¥end{enumerate}
```

1. ああああああああ
 ああああああああ
 ああああああああ
 ああああああああ

A	B	C
1	2	3
x	y	z

あああああああああああああ
 ああああああああああああああ
 ああああああああああああああ
 ああああああああああああああ
 ああああああああああああああ
 ああああああああああああああ
 あああああ

いいいいいいいいいいいいいいいい
 いいいいいいいいいいいいいいいい
 いい

2. うううううううううううううううう
 うううううううううううううううう
 うう

6.2.7 %caption の使用

%caption をつけるときは, %fgcaption, %tbcaption を用いますが, %haiti の引数は %hbox に入れてサイズを測っていますので, 複数の段落を含むものは記述できません。そのため %caption を使用する際は, minipage 環境にくるんでおく必要があります。

— %caption の使用 —

```

%begin{zuhaiti}
  %haiti{%
    %begin{minipage}{57.8pt}
      %tbcaption{表の例}
      %begin{tabular}{|c|c|c|}%\hline
        A & B & C %\hline
        1 & 2 & 3 %\hline
        x & y & z %\hline
      %end{tabular}
    %end{minipage}}
  %haiti[l](Opt, -%baselineskip){%
    %begin{minipage}{50pt}
      %begin{picture}(50,80)%
        %framebox(50,80){図}
      %end{picture}
      %fgcaption{ず}
    %end{minipage}
  }
%end{zuhaiti}%
%wakudori{5*67.8pt;8*(60pt,0pt)}
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ

```

A	B	C
1	2	3
x	y	z

図

図 2: ず

6.3 図番号の書式修正

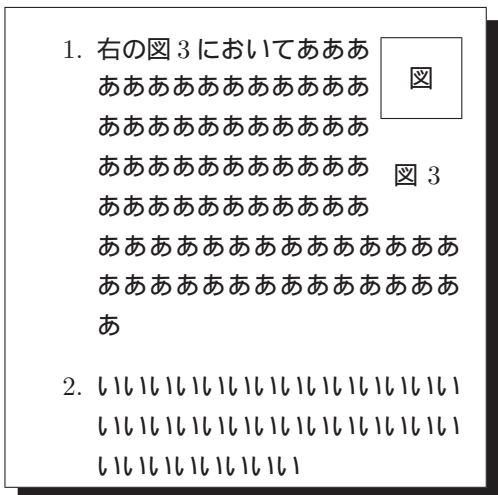
`\caption` コマンドによる図番号は

図 1: なんとかの図

という具合に番号の後ろにコロンがつきます。これが邪魔であるというご意見をよく聞きます。確かに、タイトルをつけずに使用するにはコロンは不要ですね。そこで `\EMcaption` を作りました。

使用例です。

```
\EMcaption
\begin{enumerate}
\item
\begin{Fmawarikomi}{30pt}{%
\begin{picture}(30,30)
\put(0,0){%
\framebox(30,30){図}}
\end{picture}
\EMcaption{}
\label{Fig13}}
右の図 \ref{Fig13}において
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
ああああああああああああああああ
\end{Fmawarikomi}
\item いろいろいろいろいろいろいろいろ
い
いろいろいろいろいろいろいろいろ
\end{enumerate}
```



関連して、

`\fgEMcaption`, `\tbEMcaption`

も定義されています。ただし、これらのコマンドは `emathCap.sty` で定義されていますから、使用する際は

```
\usepackage{emathCap}
```

としておく必要があります。

7 enumerate 環境の機能拡張

この節のコマンド類は、`emath.sty` ではなく、`emathE.sty` に含まれています。

7.1 小問を横に並べる。

7.1.1 横に並べる個数指定

小問を横に並べる環境 `edaenumerate` です。

ただし, `edaenumerate` 環境は `emath.sty` ではなく, `emathEy.sty` で定義されています。使用するには

```
\usepackage{emathEy}
```

を宣言しておく必要があります。

A. 一番多いのは, 横に 2 問ずつ並べるタイプでしょうか。

edaenumerate 環境

```
\begin{enumerate}
\item 次の問に答えよ .
  \begin{edaenumerate}
    \item aaa
    \item bbb
    \item ccc
    \item ddd
  \end{edaenumerate}
\end{enumerate}
```

→ **【1】** 次の問に答えよ .

(1) aaa

(2) bbb

(3) ccc

(4) ddd

B. 横に3問ずつ並べたいときは <..> オプションを使います。

```
----- edaenumerate<3> -----
\begin{enumerate}
\item 次の問に答えよ .
  \begin{edaenumerate}<3>
\item aaa
\item bbb
\item ccc
\item ddd
\item eee
\item fff
\end{edaenumerate}
\end{enumerate}
```

→

【1】 次の問に答えよ .

- | | | |
|---------|---------|---------|
| (1) aaa | (2) bbb | (3) ccc |
| (4) ddd | (5) eee | (6) fff |

なお、 $\text{\LaTeX} 2_{\epsilon}$ で enumerate に [...] オプションを併用する場合は

```
\begin{edaenumerate}<3>[(1)]
```

のように、<..> オプションを先に付けてください。

C. 小問の一部は2問ずつ、一部は3問ずつ並べたい、という例です。

変則

```
¥begin{enumerate}
¥item 次の問に答えよ .
  ¥begin{edaenumerate}
  ¥item aaaaa
  ¥item bbbbb
  ¥item ccccc
  ¥item ddddd
  ¥item<3> eee
  ¥item<3> fff
  ¥item<3> ggg
  ¥end{edaenumerate}
¥end{enumerate}
```

→ 【1】 次の問に答えよ .

(1) aaaaa

(2) bbbbb

(3) ccccc

(4) ddddd

(5) eee

(6) fff

(7) ggg

D. 小問の位置 (左インデント) を変更するには (...) オプションを用います。

edaenumerate(...)

```
¥begin{enumerate}
¥item 次の問に答えよ .
  ¥begin{edaenumerate}(2zw)
  ¥item aaa
  ¥item bbb
  ¥item ccc
  ¥item ddd
  ¥end{edaenumerate}
¥end{enumerate}
```

→ 【1】 次の問に答えよ .

(1) aaa

(2) bbb

(3) ccc

(4) ddd

E. テスト問題などでは、縦方向に空白を作りたいことがあります。<...> オプションに

```
gyoukan=2cm
```

などと行間の長さを指定します。

```
<gyoukan=...>
¥begin{enumerate}[1.~]
  ¥item 次の問いに答えよ。
  ¥begin{edaenumerate}%
    <gyoukan=2cm>[(1)]
    ¥item aaa
    ¥item bbb
    ¥item ccc
    ¥item ddd
  ¥end{edaenumerate}
¥end{enumerate}
```

```
1. 次の問いに答えよ。
(1) aaa      (2) bbb
(3) ccc      (4) ddd
```

横に並べる小問数も指定したければ、オプションを

```
<retusuu=3,gyoukan=2cm>
```

などと記述します。

均等割りなら、`\vfill` も使えますし、

```
<gyoukan=\vfill>
```

と指定するのもあります。

`\vfill` で縦方向の空白

```
\begin{enumerate}
\item 次の問に答えよ .
  \begin{edaenumerate}<gyoukan=\vfill>
\item aaa
\item bbb
\item ccc
\item ddd
  \end{edaenumerate}
\end{enumerate}
```

をタイプセットすると、下のようになります。

【1】 次の問に答えよ .

(1) aaa

(2) bbb

(3) ccc

(4) ddd

また、横に並べた小問の間に縦罫線を入れたいことがあります。解答の余白を縦 2cm として、縦罫線を入れる例です。

小問の間に縦罫線

```
¥begin{enumerate}
¥item 次の問に答えよ .
  ¥begin{edaenumerate}
  ¥item aaa ¥vspace{2cm}% 解答欄の高さ
  ¥edasikiri%                縦罫線
  ¥item bbb%
  ¥item ccc ¥vspace{2cm}%
  ¥edasikiri%
  ¥item ddd%
  ¥end{edaenumerate}
¥end{enumerate}
```

【1】 次の問に答えよ .

(1) aaa

(2) bbb

(3) ccc

(4) ddd

→

F. eda... の書式です。

【1】環境

edaenumerate<横に並べる小問の個数 (デフォルトは 2)>
(小問の左インデント)
[enumerate のオプションと同じ (LaTeX2e のみ)]

【2】コマンド

¥item<横に並べる小問の個数
(デフォルトは ¥begin{edaenumerate}の指定値)>
(小問の横幅)
[¥item のオプションと同じ]

7.1.2 ベタ並べ

edaenumerate 環境は、横に並べる個数を指定しますが、個数は指定せず、ベタに並べるのが betaenumerate 環境です。

```

betaenumerate
下の選択肢から選んで記号で答えよ。
\begin{jquote}(3zw)
\begin{betaenumerate}[A. ]
\item \bunsuu{12}
\item \bunsuu{13}
\item \bunsuu{14}
\item \bunsuu{15}
\item \bunsuu{32}
\item \bunsuu{43}
\item 4
\item 2
\item 3
\item 1
\item 5
\item \bunsuu{1}{n+1}
\item \bunsuu{n}{n+1}
\item \bunsuu{1}{rn+1}
\item \bunsuu{n}{rn+1}
\item \bunsuu{1}{r(n+1)+1}
\item \bunsuu{n}{n(n+1)+1}
\end{betaenumerate}
\end{jquote}

```

```

下の選択肢から選んで記号で答えよ。
A.  $\frac{1}{2}$  B.  $\frac{1}{3}$  C.  $\frac{1}{4}$  D.  $\frac{1}{5}$  E.  $\frac{3}{2}$  F.  $\frac{4}{3}$  G. 4 H. 2 I. 3
J. 1 K. 5 L.  $\frac{1}{n+1}$  M.  $\frac{n}{n+1}$  N.  $\frac{1}{rn+1}$  O.  $\frac{n}{rn+1}$ 
P.  $\frac{1}{r(n+1)+1}$  Q.  $\frac{n}{n(n+1)+1}$ 

```

余白量はデフォルトで

```
\def\betalabelwidth{3em}
```

としてあります。これを再定義して小さくすれば更に詰め込むことができます。

7.1.3 yokoenumerate

betaenumerate 環境は追い込み式の横並べですが、項目幅は項目ごとに異なります。長さが最大の項目幅で横並べするのが yokoenumerate 環境です。

yokoenumerate 環境

```
¥begin{enumerate}[1.~]
¥item ¥texts{betaenumerate}環境
¥begin{betaenumerate}
  ¥item aaa
  ¥item b
  ¥item ccccc
  ¥item dddddddddd
  ¥item ee
  ¥item ffff
  ¥item ggggg
  ¥item hhhhhh
¥end{betaenumerate}

¥item ¥texts{yokoenumerate}環境
¥begin{yokoenumerate}
  ¥item aaa
  ¥item b
  ¥item ccccc
  ¥item dddddddddd
  ¥item ee
  ¥item ffff
  ¥item ggggg
  ¥item hhhhhh
¥end{yokoenumerate}
¥end{enumerate}
```

1. betaenumerate 環境

(a) aaa (b) b (c) ccccc (d) dddddddddd (e) ee (f) ffff (g) ggggg
(h) hhhhhh

2. yokoenumerate 環境

(a) aaa (b) b (c) ccccc (d) dddddddddd
(e) ee (f) ffff (g) ggggg (h) hhhhhh

この環境と edaenumerate 環境との違いは、edaenumerate 環境は横に並べる項目数をこちらが指定するのに対して、yokoenumerate 環境は項目幅と版面との兼ね合いで T_EX が決めてくれるとい

う違いです。

7.2 enumerate 環境の中断・継続

enumerate 環境をいったん中断して、後刻再開したいことがあります。その際、番号を継続させるための環境

Enumerate, Enumerate*

を定義しました。例えば、次のように用います。

enumerate 環境の中断・継続

次の問いに答えよ。

```
¥begin{Enumerate}
```

```
¥item 第 1 問
```

```
¥item{¥label{val-a}} ... を満たすように、定数$a$の値を定めよ。
```

```
¥end{Enumerate}
```

以下、 a は ($\text{\%ref{val-a}$) で求めた値とする。

```
¥begin{Enumerate*}
```

```
¥item 第 3 問
```

```
¥item 第 4 問
```

```
¥end{Enumerate*}
```

→

次の問いに答えよ。

(1) 第 1 問

(2) ... を満たすように、定数 a の値を定めよ。

以下、 a は (2) で求めた値とする。

(3) 第 3 問

(4) 第 4 問

すなわち、Enumerate 環境で始めて、それを Enumerate* 環境で続けることになります。

Enumerate 環境は、実質 enumerate 環境と同じですが、後で番号を継続させる仕掛けを追加してあります。Enumerate* 環境は直前の Enumerate 環境または Enumerate* 環境の番号を引き継ぎます。

7.3 指定した項目番号に*などを附加

問題集などで、一部の問題に*印などを付けようという話です。

```
----- %item* -----  
%begin{enumerate}[1.~]  
  %item 次の問いに答えよ。  
  %begin{edaenumerate}{%  
    %protect%preitem(1)]  
    %item* aaa  
    %item bbb  
    %item ccc  
    %item* ddd  
  %end{edaenumerate}  
%end{enumerate}
```

```
1. 次の問いに答えよ。  
  
  *(1) aaa          (2) bbb  
  (3) ccc          *(4) ddd
```

すなわち、書式指定に %preitem を置き、*を付けたい項目をアスタリスク付の %item* で記述します。

*を番号の後ろにおきたければ

```
----- %item** -----  
%begin{edaenumerate}[(1){%  
  %makebox[0pt][1]%postitem}~]  
  %item** aaa  
  %item bbb  
  %item ccc  
  %item** ddd  
%end{edaenumerate}
```

```
1. 次の問いに答えよ。  
  
  (1)* aaa          (2) bbb  
  (3) ccc          (4)* ddd
```

書式指定に %postitem を加え、アスタリスクを2個つけた %item** を用います。

また、後置記号のありなしで番号部分の横幅が揃うように

```
%begin{edaenumerate}[(1){%makebox[0pt][1]%postitem}~]
```

と、細工をしています。

以外の記号を使いたいときは、%item, %item** に [..] オプションをつけます。

```
----- %item*[..] -----  
%begin{edaenumerate}{%  
  %protect%preitem(1)]  
  %item*[%dag] aaa  
  %item bbb  
  %item ccc  
  %item* ddd  
%end{edaenumerate}
```

```
1. 次の問いに答えよ。  
  
  †(1) aaa          (2) bbb  
  (3) ccc          *(4) ddd
```

後置する場合です。

— ¥item**[...] —

```
¥begin{edaenumerate}[(1){%
¥makebox[Opt][l]¥postitem~}
  ¥item**[¥dag] aaa
  ¥item bbb
  ¥item ccc
  ¥item** ddd
¥end{edaenumerate}
```

1. 次の問いに答えよ。

- (1)† aaa (2) bbb
(3) ccc (4)* ddd

前と後ろの両方に記号をつけたい，ということは起こりそうもありませんが，可能ではあります。

— 前後置 —

```
¥begin{edaenumerate}[%
¥protect¥preitem(1){%
¥makebox[Opt][l]¥postitem~}
  ¥item*[¥dag] aaa
  ¥item* bbb
  ¥item**[¥dag] ccc
  ¥item*[] [¥ddag] ddd
¥end{edaenumerate}
```

1. 次の問いに答えよ。

- * (1)† aaa *(2) bbb
(3)† ccc (4)† ddd

と，¥item*に [前置記号][後置記号] と 2 つのオプションをつけます。この場合，前置記号が ‘*’ であっても，明示する必要があります。

以上の例はすべて edaenumerate 環境で示しましたが，一般の enumerate 環境，各種の横並び enumerate 環境でも有効です。

7.4 横幅一定のローマ数字

小問などの番号にローマ数字を用いると

— 小問番号にローマ数字 —

```
¥begin{enumerate}[(i)]
  ¥item あああ
  ¥item いいい
  ¥item ううう
¥end{enumerate}
```

- (i) あああ
(ii) いいい
(iii) ううう

ローマ数字が左にせり出す感じになります。その点を配慮して

```


\emroman
\begin{enumerate}[(%
\protect\expandafter\emroman i)]
  \item あああ
  \item いいい
  \item ううう
\end{enumerate}


```

```

(i) あああ
(ii) いいい
(iii) ううう

```

大文字のローマ数字に対しては

```


\emRoman
\begin{enumerate}[(%
\protect\expandafter\emRoman I)]
  \item あああ
  \item いいい
  \item ううう
\end{enumerate}


```

```

(I) あああ
(II) いいい
(III) ううう

```

7.5 固定桁数のナンバリング 001, 002, 003, ...

左に 0 を補って固定桁数で番号付けをするためのコマンドが `\0fil` です。

```


\0fil
\begin{edaenumerate}<5>'(\0fil{2}{\getcurrentenum})'
  \item aa
  \item bb
  \item cc
  \item dd
  \item ee
  \item ff
  \item gg
  \item hh
  \item ii
  \item jj
\end{edaenumerate}


```

```

→ (01) aa    (02) bb    (03) cc    (04) dd    (05) ee
   (06) ff    (07) gg    (08) hh    (09) ii    (10) jj

```

すなわち、`\0fil` は、桁数と数値の 2 つの引数をとります。

7.6 番号付けにカタカナ・ひらがなを使う。

この節の機能は、Waver さんから使用を認めていただいたマクロを使用させていただいています。有り難うございます。> Waver さん。

enumerate 環境での番号付けは通常、

1, 2, 3, ...
a, b, c, ...

のように行われますが、これを

ア, イ, ウ, ...
イ, ロ, ハ, ...

のようにする例です。

```
{%
  \renewcommand{\labelenumi}{%
    (\% カタカナ{enumi})}
  \begin{enumerate}
    \item $y=x$           → (イ)  $y = x^2$ 
    \item $y=x^2$        (ウ)  $y = x^3$ 
    \item $y=x^3$
  \end{enumerate}
}%
```

\%カタカナ というコマンドで、

ア, イ, ウ, エ, オ, カ, ...

と、カタカナの五十音順にナンバリングされます。ここを次のように変更することもできます。

\%イロハで

イ, ロ, ハ, ニ, ホ, ヘ, ...

\%ひらがなで

あ, い, う, え, お, か, ...

\%いろはで

い, ろ, は, に, ほ, へ, ...

7.7 enumerate 環境のオプション文字の追加

enumerate パッケージを使用すると、以下のオプション文字を用いて番号付けを制御することができます。

1 : 1,2,3,4,5,6,7,.....
I : I,II,III,IV,V,VI,VII,.....
i : i,ii,iii,iv,v,vi,vii,.....
A : A,B,C,D,E,F,G,.....
a : a,b,c,d,e,f,g,.....

emathE.sty では、オプションに使用できる文字として、下の 7 個を追加してあります。

m : ①,②,③,④,⑤,⑥,⑦,.....
n : ①,②,③,④,⑤,⑥,⑦,.....
ア : ア, イ, ウ, エ, オ, カ, キ, ...
イ : イ, 口, 八, 二, ホ, ヘ, ト, ...
あ : あ, い, う, え, お, か, き, ...
い : い, ろ, は, に, ほ, へ, と, ...
ー : 一, 二, 三, 四, 五, 六, 七, ...

ただし、最後の「ー」は plect パッケージを必要とします。

その他の文字はそのまま出力されますから

```
¥begin{enumerate}[(ア)]
```

とすれば、番号付けが

```
(ア), (イ), (ウ), (エ), (オ), (カ), (キ), ...
```

となります。

このオプション文字の追加は、edaenumerate 環境に対しても有効です。その一例として n オプション — 縦長の丸付き数字 — を用いたものをごらん頂きましょう。

enumerate 環境のオプション文字 m

右の ¥texttt{BASIC} のプログラム (省略) の ¥framebox{ } に
次のいずれかを補い、プログラムを完成させよ。

```
{¥ttfamily
  ¥begin{edaenumerate}<5>[n]
    ¥setcurrentenum{-1}%
    ¥item +
    ¥item -
    ¥item $¥ast$
    ¥item /
    ¥item =
    ¥item{}<>
    ¥item >
    ¥item{}<
    ¥item >=
    ¥item{}<=
  ¥end{edaenumerate}
}
```

右の BASIC のプログラム (省略) の に次のいずれかを補い、プログラムを完成させよ。

→

① +	② -	③ *	④ /	⑤ =
⑥ <>	⑦ >	⑧ <	⑨ >=	⑩ <=

ここで `\setcurrentenum` というコマンドが登場しますが、これは、`enumerate` 環境の現在の深さの番号変数の値を変更するものです。この例では、番号が 1 からではなく、0 から始まっています。そのため、初期値を変更しておく必要があります。

```
\setcounter{enumi}{-1}
```

とするのが普通ですが、ある問題集の中にこの問題を収録したとき、問題集の番号を `enumerate` 環境で付けますと、`enumerate` 環境の深さが変わるにより

```
\setcounter{enumii}{-1}
```

と修正を必要とします。ここで用いた

```
\setcurrentenum{-1}
```

は、現在の `enumerate` 環境の深さを判断して、その番号を変更する機能を持つコマンドで、`emath.sty` の中で定義されています。

これと対をなすコマンドが `\getcurrentenum` で、現在の問題番号を取得します。

7.8 `enumerate` 環境における番号付けの初期値・刻み値変更

`enumerate` 環境における番号付けで

初期値を 1 以外の値にしたい。

刻み値を 1 以外に指定したい。

という要望にこたえるため、`enumerate` 環境の `<key=val>` オプションの `key` に

```
syokiti, kizamiti
```

を新設しました。

<kizamiti=10>

```
¥begin{reidai}
  刻み値を 10 に設定する例
  ¥begin{jquote}
    ¥begin{ttfamily}
      ¥begin{enumerate}<syokiti=90,kizamiti=10>[1]
        ¥item INPUT "分数 A/B(0<A<B) を入力 , A, B =";A, B
        ¥item IF (B MOD A) = 0 THEN GOTO ¥ref{end}
        ¥item N = INT(¥Hako)+¥Hako
        ¥item PRINT N;
        ¥item A = A*N-B
        ¥item B = B*N
        ¥item GOTO ¥Hako
        ¥item ¥label{end}PRINT ¥Hako
        ¥item END
      ¥end{enumerate}
    ¥end{ttfamily}
  ¥end{jquote}
¥end{reidai}
```

例題 1 刻み値を 10 に設定する例

```
100 INPUT "分数 A/B(0<A<B) を入力 , A, B =";A, B
110 IF (B MOD A) = 0 THEN GOTO 170
120 N = INT( ア )+ イ
130 PRINT N;
140 A = A*N-B
150 B = B*N
160 GOTO ウ
170 PRINT エ
180 END
```

次は、刻み値を負にする例です。

<kizamiti=-1>

```
¥begin{reidai}
  初期値を 27, 刻み値を$-1$に設定する例
  ¥begin{jquote}
    ¥begin{enumerate}<syokiti=27,kizamiti=-1>[1.~]
      ¥item zzz
      ¥item yyy
      ¥begin{align}
        x&=1¥¥
        y&=2¥label{yy}¥¥
        z&=3
      ¥end{align}
      ¥item xxx
      ¥item www
    ¥end{enumerate}
    ¥eqref{yy}において
  ¥end{jquote}
¥end{reidai}
```

例題 2 初期値を 27, 刻み値を -1 に設定する例

```
26. zzz
25. yyy
      x = 1 ..... ①
      y = 2 ..... ②
      z = 3 ..... ③
24. xxx
23. www
    ②において
```

7.9 enumerate 環境の前後左右の空き調整

enumerate 環境の前後左右の空きを調整するには, enumerate 環境の<...>オプション

apnenum=....

を利用します。

まずは, 標準の enumerate 環境です。

enumerate 環境

```
¥begin{enumerate}[1.~]  
  ¥item 次の問いに答えよ。  
    ¥begin{enumerate}[(1)]  
      ¥item あああああああああああ  
        あああああああああああああ  
      ¥item いいいいいいいいいいいい  
        いいいいいいいいいいいい  
    ¥end{enumerate}  
  ¥end{enumerate}
```

1. 次の問いに答えよ。

- (1) ああああああああああ
ああああああああああ
あああああ
- (2) いいいいいいいいいい
いいいいいいいいいい
いいいいいい

7.9.1 左インデント

では、左インデントの修正です。

前ページ、標準の `enumerate` 環境では、小問の左インデントは、大問のそれより全角 2 文字分深くなっています。それを同じにしてみましょう。

— `\leftmargin` の変更 —

```
\begin{enumerate}[1.~]
  \item 次の問いに答えよ。
  \begin{enumerate}%
    <apnenum={\leftmargin=0zw}>[(1)]
    \item あああああああああああ
      あああああああああああああ
    \item しいしいしいしいしいしい
      しいしいしいしいしいしいしい
    \end{enumerate}
  \end{enumerate}
```

1. 次の問いに答えよ。

- (1) あああああああああああ
ああああああああああああ
ああ
- (2) しいしいしいしいしいしい
しいしいしいしいしいしい
しい

小問番号が左に出すぎていますから、小問の 1 行目だけ右に押し込みます。`\itemindent` を変更します。

— `\itemindent` の変更 —

```
\begin{enumerate}[1.~]
  \item 次の問いに答えよ。
  \begin{enumerate}%
    <apnenum={\leftmargin=0zw%
      \itemindent=1zw%
    }>[(1)]
    \item あああああああああああ
      あああああああああああああ
    \item しいしいしいしいしいしい
      しいしいしいしいしいしいしい
    \end{enumerate}
  \end{enumerate}
```

1. 次の問いに答えよ。

- (1) あああああああああああ
ああああああああああああ
ああ
- (2) しいしいしいしいしいしい
しいしいしいしいしいしい
しい

7.9.2 縦間隔

次は縦間隔の修正です。

大問と小問の間を詰めたい，というご意見はよく耳にします。¥topsep, ¥parskip を変更します。

大問と小問の縦間隔

```
¥begin{enumerate}[1.~]
  ¥item 次の問いに答えよ。
  ¥begin{enumerate}%
    <apnenum={¥leftmargin=0zw%
      ¥itemindent=1zw%
      ¥topsep=0pt%
      ¥parskip=0pt%
    }>[(1)]
    ¥item ああああああああああああ
      ああああああああああああ
    ¥item いいいいいいいいいいいいいい
      いいいいいいいいいいいいいい
  ¥end{enumerate}
¥end{enumerate}
```

- ```
1. 次の問いに答えよ。
(1) ああああああああああああ
 ああああああああああああ
 あああ
(2) いいいいいいいいいいいいいい
 いいいいいいいいいいいいいい
 いいいい
```

小問同士の間隔も詰めたい，となると，¥itemsep を修正します。

### 小問同士の縦間隔

```
¥begin{enumerate}[1.~]
 ¥item 次の問いに答えよ。
 ¥begin{enumerate}%
 <apnenum={¥leftmargin=0zw%
 ¥itemindent=1zw%
 ¥topsep=0pt%
 ¥parskip=0pt%
 ¥itemsep=0pt%
 }>[(1)]
 ¥item ああああああああああああ
 ああああああああああああ
 ¥item いいいいいいいいいいいいいい
 いいいいいいいいいいいいいい
 ¥end{enumerate}
¥end{enumerate}
```

- ```
1. 次の問いに答えよ。
(1) ああああああああああああ
    ああああああああああああ
    あああ
(2) いいいいいいいいいいいいいい
    いいいいいいいいいいいいいい
    いいいい
```

7.9.3 縦間隔のグルー

ところで、縦間隔の初期値がどうなっているかを確認しておきます。

```
topsep = 8.0pt plus 2.0pt minus 4.0pt
parskip = 4.0pt plus 2.0pt minus 1.0pt
itemsep = 4.0pt plus 2.0pt minus 1.0pt
```

ある意味で、グルーは T_EX の命です。これらを 0pt 固定にするということは、ページ全体のバランスがとれなくなり、折角の T_EX の美しさを損なうこととなりかねません。

ということでグルーを付けておきましょう。

```
enumerat 環境
\begin{enumerate}[1.]
  \item 次の問いに答えよ。
  \begin{enumerate}%
    <apnenum={\leftmargin=0zw%
      \itemindent=1zw%
      \topsep=0pt plus 6pt%
      \parskip=0pt plus 3pt%
      \itemsep=0pt plus 3pt%
    }>[(1)]
    \item あああああああああああ
      あああああああああああ
    \item いいいいいいいいいいいい
      いいいいいいいいいいいい
  \end{enumerate}
\end{enumerate}
```

```
1. 次の問いに答えよ。
(1) あああああああああああ
    あああああああああああ
    あああ
(2) いいいいいいいいいいいい
    いいいいいいいいいいいい
    いいいい
```

7.9.4 \enumSep

<apnenum=...>オプションは、当該 enumerate 環境のみに対する局所的な変更です。文書全体に対して同様の変更をするためには、

```
\enumSep, \enumSepi, \enumSepii, ...
```

コマンドが用意されています。

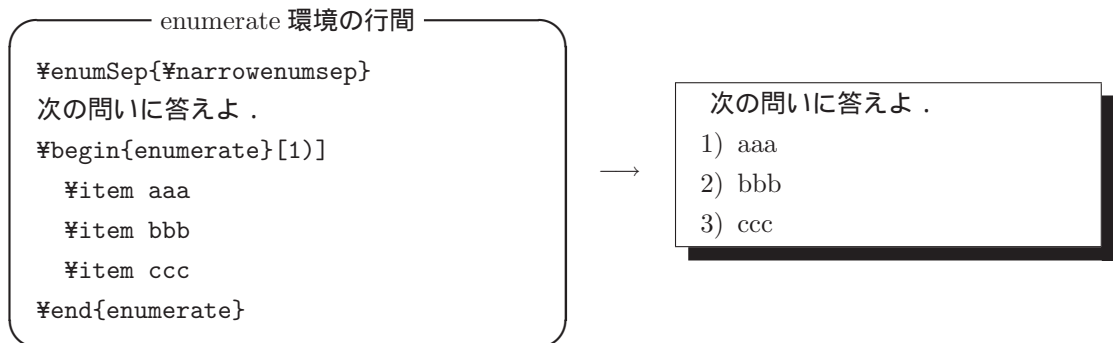
\enumSep コマンドの引数に list 環境の第二引数に記述するものを与えます。例えば、

```
\enumSep{\topsep=0pt\parskip=0pt%
  \parsep=0pt\itemsep0pt}%
```

と記述すると、行間は狭くなります。これはよく使いそうですから、

```
\def\narrowenumsep{\topsep=0pt\parskip=0pt\parsep=0pt%
  \itemsep0pt}%
```

と、代名詞 `\narrowenumsep` を定義してあります。



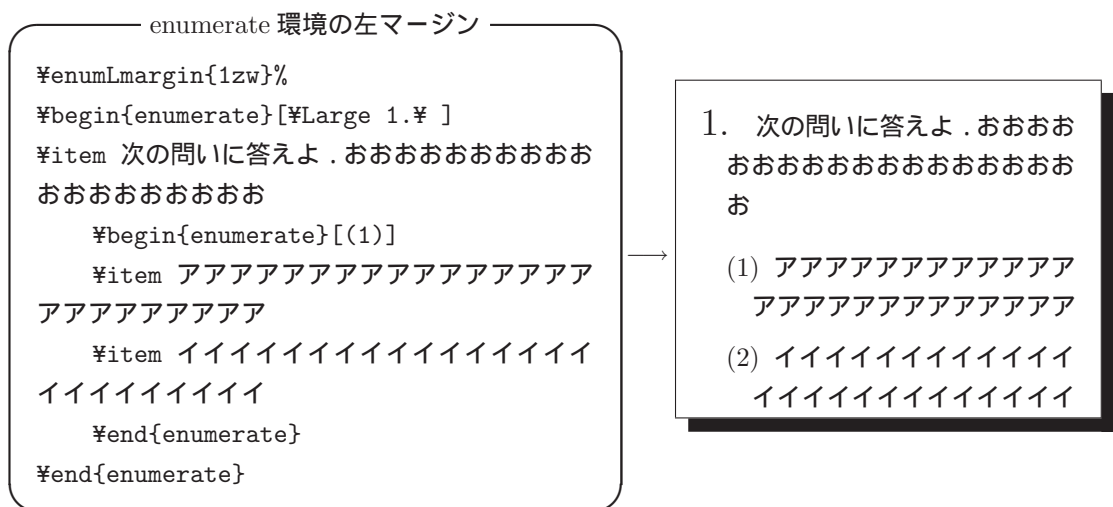
`\enumSep` は `enumerate` 環境 4 レベルすべてについて共通の設定をしますが、レベル毎に変更したいというときは、`\enumSepi`, `\enumSepii`, `\enumSepiii`, `\enumSepiv` を使用することもできます。

なお、これらのコマンドと、`<apnenum=...>` オプションを併用した場合は、当該 `enumerate` 環境に対しては `<apnenum=...>` オプションが優先されます。

7.9.5 `\enumLmargin`

文書全体で左マージンを変更するコマンドが `\enumLmargin` です。

例えば次のように用います。



前節の行間を狭くするコマンドと併用すると、

ケチケチ enumerate

```

%enumSep{%narrowenumsep}%
%enumLmargin{1zw}%
%begin{enumerate}[%Large 1.% ]
%item 次の問いに答えよ . おおおおおおおお
おおおおおおお
    %begin{enumerate}[(1)]
    %item アアアアアアアアアアアアアア
アアアアアアアア
        %item イイイイイイイイイイイイイイ
イイイイイイイイ
    %end{enumerate}
%end{enumerate}

```

1. 次の問いに答えよ . おおおお
 おおおおおおおおおおお
 お

(1) アアアアアアアアアアアア
 アアアアアアアアアアアア

(2) イイイイイイイイイイイイ
 イイイイイイイイイイイイ

%enumLmargin は enumerate 環境 4 レベルすべてについて左マージンを変更しますが , レベル毎に変更したいというときは , %enumLmargini, %enumLmarginii, %enumLmarginiii, %enumLmarginiv を使用することもできます。

なお , この目的にも %apnlist を使用することができます。
 まずはデフォルトの確認です。

enumerate 環境

```

ああああああああああああああ
ああああああああああああああ
%begin{enumerate}[1.~]
%item 次の問いに答えよ。
    おおおおおおおおおお
    %begin{enumerate}[(1)]
    %item あああああああああ
    あああああああああ
    %item しいしいしいしいしいしい
    しいしいしいしいしいしい

    うううううううううう
    うううううううううう

    %end{enumerate}
%item アアアアアアアアアアアア
    アアアアアアアアアア

%end{enumerate}
しいしいしいしいしいしいしい
しいしいしいしいしいしいしい

```

ああああああああああああああ
 あああああああああああああ

1. 次の問いに答えよ . おおおおお
 おおおおお

(1) あああああああああ
 あああああああ

(2) しいしいしいしいしいしい
 しいしいしいしいしい
 うううううううううう
 うううううううううう

2. アアアアアアアアアアアア
 アアアアアアアアアア

しいしいしいしいしいしいしい
 しいしいしいしいしいしい

左インデントを調整した一例です。

—— %apnlist ——

```

ああああああああああああああああ
ああああああああああああああああ
%apnlist{%itemindent=1zw
  %leftmargin=1zw
  %listparindent=1zw}
%begin{enumerate}[1.~]
  %item 次の問いに答えよ。
  おおおおおおおおおおおおお
%apnlist{%narrowsep
  %itemindent=1zw
  %topsep=0zw
  %leftmargin=1zw
  %listparindent=1zw}
  %begin{enumerate}[(1)]
% 以下前の例と同じ

```

```

ああああああああああああああああ
ああああああああああああああああ
1. 次の問いに答えよ。おおおおお
  おおおおおおお
  (1) あああああああああああああ
  ああああああああ
  (2) いろいろいろいろいろいろ
  いろいろいろいろ
  うううううううううううう
  うううううううう
2. アアアアアアアアアアアアアアアア
  アアアアアアアアア
  いろいろいろいろいろいろいろいろ
  いろいろいろいろいろいろいろいろ

```

7.10 edaenumerate 環境の前後左右の空き調整

edaenumerate 環境における項目周りのアキの調整です。

7.10.1 edaenumerate 環境

デフォルトの edaenumerate 環境を見ておきます。

—— edaenumerate 環境 ——

```

%begin{enumerate}[1.~]
  %item 次のおおおおおおおおおおお
  %begin{edaenumerate}[(1)]
    %item
    ああああああああああああああ
    ああああああああああああああ
  %item
  いろいろいろいろいろいろいろいろ
  いろいろいろいろいろいろいろいろ
  %end{edaenumerate}
%end{enumerate}

```

1. 次のおおおおおおおおおおお

```

(1) ああああああああああああああ (2) いろいろいろいろいろいろ
  ああああああああああああああ いろいろいろいろいろいろ

```

7.10.2 preedasep=.. オプション

上のデフォルト状態に対して、「小項目の 2 行目以降を左に突き出したい」という要望があります。そこで、<preedasep=..>オプションを新設しました。

(preedasep のデフォルト値は 1.75zw です。)

— preedasep=.. オプション —

```
¥begin{enumerate}[1.~]
  ¥item 次のoooooooooooooooooooooooooooo
    ¥begin{edaenumerate}<preedasep=0pt>[(1)]
      ¥item
        ああああああああああああああ
        ああああああああああああああ
      ¥item
        いろいろいろいろいろいろいろいろ
        いろいろいろいろいろいろいろいろ
    ¥end{edaenumerate}
  ¥end{enumerate}
```

1. 次のoooooooooooooooooooooooooooo

(1) ああああああああああああああ(あ) いろいろいろいろいろいろいろいろ
あああああああああああああ いろいろいろいろいろいろいろいろ

2行目以降のみならず, 1行目も左に突き出てしまいましたから, <apnenum=..>オプションで1行目を修正します。

— apnenum=.. オプション —

```
¥begin{enumerate}[1.~]
  ¥item 次のoooooooooooooooooooooooooooo
    ¥begin{edaenumerate}
      <preedasep=0pt, apnenum={¥itemindent=1.75zw}>[(1)]
      ¥item
        ああああああああああああああ
        ああああああああああああああ
      ¥item
        いろいろいろいろいろいろいろいろ
        いろいろいろいろいろいろいろいろ
    ¥end{edaenumerate}
  ¥end{enumerate}
```

1. 次のoooooooooooooooooooooooooooo

(1) ああああああああああああああ(あ) いろいろいろいろいろいろいろいろ
あああああああああああああ いろいろいろいろいろいろいろいろ

7.10.3 postedasep=.. オプション

小項目間のアキは、<postedasep=..>オプションで調整できます。
(postedasepのデフォルト値は.75zwです。)

postedasep=.. オプション

```
¥begin{enumerate}[1.~]
  ¥item 次のoooooooooooooooooooooooooooo
    ¥begin{edaenumerate}
      <preedasep=0pt,postedasep=2zw,
      apnum={¥itemindent=1.75zw}>
      [(1)]
    ¥item
      あああああああああああああああ
      あああああああああああああああ
    ¥item
      LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
      LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
    ¥end{edaenumerate}
  ¥end{enumerate}
```

1. 次のoooooooooooooooooooooooooooo

(1) あああああああああああああああ (2) LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
ああああああああああああ LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL

7.10.4 edatopsep=.. オプション

大問と edaenumerate 環境との縦間隔は<edatopsep=..>オプションで調整可能です。
(edatopsep のデフォルト値は 0pt です。)

```
----- edatopsep=.. オプション -----
%begin{enumerate}[1.~]
  %item 次のoooooooooooooooooooooooooooooooo
    %begin{edaenumerate}
      <preedasep=0pt,postedasep=2zw,edatopsep=-.8zh,
      apnum={%itemindent=1.75zw}>[(1)]
      %item
        ああああああああああああああああ
        ああああああああああああああああ
      %item
        いいいいいいいいいいいいいいいいいい
        いいいいいいいいいいいいいいいいい
      %item
        ううううううううううううううううう
        ううううううううううううううううう
      %item
        えええええええええええええええええ
        えええええええええええええええええ
    %end{edaenumerate}
  %end{enumerate}
```

1. 次のoooooooooooooooooooooooooooooooo	
(1) ああああああああああああああああ	(2) いいいいいいいいいいいいいいいいい
あああああああああああああ	いいいいいいいいいいいいい
(3) ううううううううううううううううう	(4) えええええええええええええええええ
ううううううううううううう	えええええええええええええ

7.11 %item 直後の edaenumerate 環境

7.11.1 問題提示

大問 %item には文章があって、そのあとに小問が続くのが普通ですが、

普通は

```
¥begin{enumerate}[¥protect¥expandafter¥fbox 1]
  ¥item 次の問いに答えよ。
    ¥begin{edaenumerate}[(1)]
      ¥item 第1問
      ¥item 第2問
    ¥end{edaenumerate}
¥end{enumerate}
```

1 次の問いに答えよ。

(1) 第1問

(2) 第2問

大問には文章はなく、いきなり小問を edaenumerate 環境で記述するときの話です。

大問直後の edaenumerate 環境

```
¥begin{enumerate}[¥protect¥expandafter¥fbox 1]
  ¥item
    ¥begin{edaenumerate}[(1)]
      ¥item 第1問
      ¥item 第2問
    ¥end{edaenumerate}
¥end{enumerate}
```

1 (1) 第1問

(2) 第2問

特に問題はないようですが、大問の左インデントをいじっていると問題が発生します。

7.11.2 大問の左インデント操作との競合

大問の左インデントを変更している場合は問題が発生します。

¥enumLmargin との併用

```
¥enumLmargini{Opt}
¥begin{enumerate}[¥protect¥expandafter¥fbox 1]
  ¥item
    ¥begin{edaenumerate}[(1)]
      ¥item 第1問
      ¥item 第2問
      ¥item 第3問
      ¥item 第4問
    ¥end{edaenumerate}
¥end{enumerate}
```

- | | |
|-----------|-----------|
| ① 第 1 問 | (2) 第 2 問 |
| (3) 第 3 問 | (4) 第 4 問 |

極端に

```
¥enumLmargini{Opt}
```

として、大問の左マージンをなくしてしまうと、大問番号と小問番号が重なります。

```
# こんな馬鹿な使い方はしないほうがよろしいかと思いますが、
# 小問番号を横に並べる
# enumerate の 2 行目以降を左に突き出す
# などという発想の先にこのような事態もおこってくるということでしょうか。
# どこまで続くぬかるみぞ、といたくなりますが .....
```

7.11.3 edaenumerate 環境に<edafirstindent=..>オプション

edaenumerate 環境に <edafirstindent=..>オプションを付加する解決法です。

<edafirstindent=..>オプション

```
¥enumLmargini{Opt}
¥begin{enumerate}[¥protect¥expandafter¥fbox 1]
  ¥item
    ¥begin{edaenumerate}<edafirstindent=2zw>[(1)]
      ¥item 第 1 問
      ¥item 第 2 問
      ¥item 第 3 問
      ¥item 第 4 問
    ¥end{edaenumerate}
¥end{enumerate}
```

- | | |
|-------------|-----------|
| ① (1) 第 1 問 | (2) 第 2 問 |
| (3) 第 3 問 | (4) 第 4 問 |

7.11.4 emathAe との併用

emathAe.sty を用いて、解答を巻末に集めるときはこの方式のほうがすっきりするようです。また、¥kaitou コマンドも使用可能となります。

```

\enumLmargin{0pt}
\openKaiFile
\begin{enumerate}[\protect\expandafter\fbbox1]
  \item 大問 1
    \kaitou{\def\Kaienumerate{edaenumerate}%
            \def\Kaienumopt{<edafirstindent=2zw,retusuu=3>}}%
    \begin{enumerate}[(1)]
      \item 小問 1 \kaitou{解答 1}
      \item 小問 2 \kaitou{解答 2}
      \item 小問 3 \kaitou{解答 3}
      \item 小問 4 \kaitou{解答 4}
      \item 小問 5 \kaitou{解答 5}
      \item 小問 6 \kaitou{解答 6}
    \end{enumerate}
  \end{enumerate}
\closeKaiFile
\hrule
【解答】

\inputKaiFile

```

1 大問 1

(1) 小問 1

(2) 小問 2

(3) 小問 3

(4) 小問 4

(5) 小問 5

(6) 小問 6

【解答】

1 (1) 解答 1

(2) 解答 2

(3) 解答 3

(4) 解答 4

(5) 解答 5

(6) 解答 6

7.11.5 betaenumerate 環境では

新設した`<edafirstindent=..>`は, `betaenumerate` 環境に対しても有効です。

— betaenumerate 環境の場合 —

```
¥enumLmargini{Opt}
¥begin{enumerate}[¥protect¥expandafter¥fbox 1]
  ¥item
    ¥begin{betaenumerate}<edafirstindent=2zw>[(1)]
      ¥item 第 1 問
      ¥item 第 2 問
      ¥item 第 3 問
      ¥item 第 4 問
      ¥item 第 5 問
      ¥item 第 6 問
      ¥item 第 7 問
      ¥item 第 8 問
    ¥end{betaenumerate}
  ¥end{enumerate}
```

1 (1) 第 1 問 (2) 第 2 問 (3) 第 3 問 (4) 第 4 問 (5) 第 5 問 (6) 第 6 問
(7) 第 7 問 (8) 第 8 問

7.11.6 解答部の betaenumerate

emathAe.sty を用いて、解答部を betaenumerate とする場合も同様です。

```
emathAe で betaenumerate

\enumLmargin{0pt}
\openKaiFile
\begin{enumerate}[\protect\expandafter\fbbox1]
  \item 大問 1
    \kaitou{\def\Kaienumerate{betaenumerate}%
            \def\Kaienumopt{<edafirstindent=2zw>}}%
    \begin{enumerate}[(1)]
      \item 小問 1 \kaitou{解答 1}
      \item 小問 2 \kaitou{かいとう 2}
      \item 小問 3 \kaitou{answer3}
      \item 小問 4 \kaitou{解答 4}
      \item 小問 5 \kaitou{解答 5}
      \item 小問 6 \kaitou{解答 6}
      \item 小問 7 \kaitou{解答 7}
      \item 小問 8 \kaitou{解答 8}
    \end{enumerate}
  \end{enumerate}
\closeKaiFile
\hrule
【解答】
\inputKaiFile
```

1 大問 1

(1) 小問 1

(2) 小問 2

(3) 小問 3

(4) 小問 4

(5) 小問 5

(6) 小問 6

(7) 小問 7

(8) 小問 8

【解答】

1 (1) 解答 1 (2) かいとう 2 (3) answer3 (4) 解答 4 (5) 解答 5

(6) 解答 6 (7) 解答 7 (8) 解答 8

7.12 list 環境の行間

enumerate 以外の list に基づく環境

description, itemize

環境について、行間の修正をするために

`\apnlist`

を用意してあります。

7.12.1 itemize 環境

itemize 環境のデフォルト状態は

itemize 環境

```
ああああああああああああああああ
ああああああああああああああああ
\begin{itemize}
  \item aaa
  \item bbb
  \item ccc
\end{itemize}
llllllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllllll
```

```
ああああああああああああああああ
ああああああああああああああああ
• aaa
• bbb
• ccc
llllllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllllll
```

項目間の行間を狭くするには `\apnlist` の引数に list 環境の第 2 引数に引き渡すものを記述します。

行間を狭く

```
ああああああああああああああああ
ああああああああああああああああ
\apnlist{\narrowsep}
\begin{itemize}
% 以下同じ
```

```
ああああああああああああああああ
ああああああああああああああああ
• aaa
• bbb
• ccc
llllllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllllll
```

`\narrowsep` は

```
\def\narrowsep{\parsep#z@#itemsep#z@#parskip#z@}
```

と `emathE.sty` で定義されています。すなわち

`\parsep`, `\itemsep`, `\parskip`

を `0pt` としています。この値を細かく調整することで好みのスタイルを実現できます。

本文との間も詰めるには、`\topsep=0pt` を附加します。

— itemize 環境 —

```

ああああああああああああああああ
ああああああああああああああああ
%apnlist{%narrowsep%topsep=0pt}
%begin{itemize}
% 以下同じ

```

```

ああああああああああああああああ
ああああああああああああああああ
• aaa
• bbb
• ccc
llllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllll

```

7.12.2 description 環境

description 環境についても同様です。

— description 環境 —

```

ああああああああああああああああ
ああああああああああああああああ
%begin{description}
  %item [AAA] aaa
  %item [BB] bbbb
  %item [C] ccccc
%end{description}
llllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllll

```

```

ああああああああああああああああ
ああああああああああああああああ
AAA aaa
BB bbbb
C ccccc
llllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllll

```

— 行間を狭く —

```

ああああああああああああああああ
ああああああああああああああああ
%apnlist{%narrowsep}
%begin{description}
% 以下同じ

```

```

ああああああああああああああああ
ああああああああああああああああ
AAA aaa
BB bbbb
C ccccc
llllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllll

```

— 本文との間隔 —

```

ああああああああああああああああ
ああああああああああああああああ
%apnlist{%narrowsep%topsep=0pt}
%begin{description}
% 以下同じ

```

```

ああああああああああああああああ
ああああああああああああああああ
AAA aaa
BB bbbb
C ccccc
llllllllllllllllllllllllllllllllll
llllllllllllllllllllllllllllllllll

```

7.12.3 enumerate 環境

enumerate 環境に対しても %apnlist は有効ですが、層ごとに指定するのはちと面倒です。

まずはデフォルトの enumerate 環境です。

— enumerate 環境 —

```
ああああああああああああああああ
ああああああああああああああああ
¥begin{enumerate}[1.~]
  ¥item 次の問いに答えよ。
    ¥begin{enumerate}[(1)]
      ¥item あああああああああああ
        あああああああああああ
    ¥item いいいいいいいいいいいい
      いいいいいいいいいいいい
    ¥end{enumerate}
  ¥item アアアアアアアアアアアアアア
    アアアアアアアアアアアア
¥end{enumerate}
いいいいいいいいいいいいいいいい
いいいいいいいいいいいいいいいい
```

```
ああああああああああああああああ
ああああああああああああああああ
1. 次の問いに答えよ。
  (1) あああああああああああああ
      あああああああああああ
  (2) いいいいいいいいいいいいいい
      いいいいいいいいいいいい
2. アアアアアアアアアアアアアア
   アアアアアアアアアアアア
   いいいいいいいいいいいいいいいい
   いいいいいいいいいいいいいいいい
```

第2層の enumerate 環境の直前に `¥apnlist{¥narrowsep¥topsep=0pt}` を置いて、小問間の空き・大問と小問の間の空きを取り去ってみます。

— ¥apnlist —

```
ああああああああああああああああ
ああああああああああああああああ
¥begin{enumerate}[1.~]
  ¥item 次の問いに答えよ。
  ¥apnlist{¥narrowsep¥topsep=0pt}
  ¥begin{enumerate}[(1)]
  % 以下前の例と同じ
```

```
ああああああああああああああああ
ああああああああああああああああ
1. 次の問いに答えよ。
  (1) あああああああああああああ
      あああああああああああ
  (2) いいいいいいいいいいいいいい
      いいいいいいいいいいいい
2. アアアアアアアアアアアアアア
   アアアアアアアアアアアア
   いいいいいいいいいいいいいいいい
   いいいいいいいいいいいいいいいい
```

なお、`¥narrowsep` は

```
¥def¥narrowsep{¥parsep=¥z@¥itemsep¥z@¥parskip¥z@}
```

と `emathE.sty` で定義されています。すなわち

```
¥parsep, ¥itemsep, ¥parskip
```

を `0pt` としています。この値を細かく調整することで好みのスタイルを実現できます。

先の例に戻り、大問間の行間も詰めれば、第1層の enumerate 環境の前に

```
¥apnlist{narrowsep}
```

を入れますと、

```
----- %apnlist -----
ああああああああああああああああ
ああああああああああああああああ
%apnlist{%narrowsep}
%begin{enumerate}[1.~]
  %item 次の問いに答えよ。
%apnlist{%narrowsep%topsep=0pt}
  %begin{enumerate}[(1)]
% 以下前の例と同じ
```

```
ああああああああああああああああ
ああああああああああああああああ
1. 次の問いに答えよ。
  (1) あああああああああああああ
      あああああああああ
  (2) いいいいいいいいいいいいいい
      いいいいいいいいいい
2. アアアアアアアアアアアアアアアア
   アアアアアアアアアア
   いいいいいいいいいいいいいいいい
   いいいいいいいいいいいいいいいい
```

本文と enumerate 環境部との縦間隔も詰めたい、となると第 1 層にも %topsep=0pt を附加します。

```
----- %apnlist -----
ああああああああああああああああ
ああああああああああああああああ
%apnlist{%narrowsep%topsep=0pt}
%begin{enumerate}[1.~]
  %item 次の問いに答えよ。
  %begin{enumerate}[(1)]
% 以下前の例と同じ
```

```
ああああああああああああああああ
ああああああああああああああああ
1. 次の問いに答えよ。
  (1) あああああああああああああ
      あああああああああ
  (2) いいいいいいいいいいいいいい
      いいいいいいいいいい
2. アアアアアアアアアアアアアアアア
   アアアアアアアアアア
   いいいいいいいいいいいいいいいい
   いいいいいいいいいいいいいいいい
```

この場合、第 1 層、第 2 層の %apnlist の引数は同一となりますから、第 2 層は省略可能です。(省略すれば、第 1 層のものが第 2 層以下すべてに有効となります。第 2 層はデフォルトのままにしたい、ということなら第 2 層の前に

```
%apnlist{%relax}
```

などと指定する必要があります。)

7.13 問題番号の参照

enumerate 環境で番号付けされた問題番号にラベルをつけて、あとでそれを参照することができますが、第 2 層以下の場合、参照の形式がそれより上の層の番号まで含まれてしまいます。具体例を挙げてみましょう。

- 第 1 層の番号付けが 1,2,3,...
- 第 2 層の番号付けが a,b,c,...

であるとしてます。

問題番号の参照 (デフォルト)

```
¥begin{enumerate}
¥item 第 1 問
¥begin{enumerate}
¥item 小問 a
¥item¥label{mon1b}小問 b
¥item ¥ref{mon1b}において
¥end{enumerate}
¥end{enumerate}
```

→

1. 第 1 問
 - a. 小問 a
 - b. 小問 b
 - c. 1b において

しかし、この場合は小問 c における参照は

1b において

ではなく、大問番号を取り去って

b において

で十分です。このように、当該番号だけの参照を実現するために

¥refcurrentenum

を用意しました。次のように用います。

¥refcurrentenum の効用

```
¥refcurrentenum
¥begin{enumerate}
¥item 第 1 問
¥begin{enumerate}
¥item 小問 a
¥item¥label{mon2b}小問 b
¥item ¥ref{mon2b}において
¥end{enumerate}
¥end{enumerate}
```

→

- 1. 第 1 問
 - a. 小問 a
 - b. 小問 b
 - c. b において

デフォルト状態に戻すために、`¥refallemum` を用意しました。

```
¥refallemum
¥begin{enumerate}
¥item 第 1 問
¥begin{enumerate}
¥item 小問 a
¥item¥label{mon3b}小問 b
¥item ¥ref{mon3b}において
¥end{enumerate}
¥end{enumerate}
```

→

- 1. 第 1 問
 - a. 小問 a
 - b. 小問 b
 - c. 1b において

8 その他一般

8.1 数式モードの中で英大文字をローマン体にする。

8.1.1 caprm 環境

数式モードの中ではアルファベットはすべて斜体となります。ところが、点を表す記号は立体（ローマン体）で表す習慣です。例えば点 $A(x_1, y_1)$ などと記述すると、 A が斜体となってしまいます。対応策は $\mathrm{A}(x_1, y_1)$, $\text{A}(x_1, y_1)$ などと A を立体にするコマンドを記述するのが普通です。別の対策として、数式モードで英大文字は立体で、小文字は斜体で表示させようという方法もあります。そのための環境 `caprm` 環境です。

```
----- caprm -----
\begin{caprm}
  二点  $A(a_1, a_2)$ ,  $B(b_1, b_2)$  を結ぶ直線
\end{caprm}
```

という記述をタイプセットすると

二点 $A(a_1, a_2)$, $B(b_1, b_2)$ を結ぶ直線

と、大文字は立体、小文字は斜体となります。しかし、大文字はいつでも立体にするというのも問題です。例えば正弦定理

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} = 2R$$

では、大文字も斜体で書くのが習慣です。これは `caprm` 環境ではなく、通常の数式モードの方が良いでしょう。

点を表す英大文字は立体で、三角形の頂角の大きさを表す英大文字は斜体で、などという区別はなかなかやっかいです。

しかし、この `caprm` 環境はなかなか便利な場合もあります。例えば、

点列 $P_1, P_2, P_3, \dots, P_n, \dots$

の表現は `caprm` 環境だと

```
----- caprm -----
$ 点列  $P_1, \sim P_2, \sim P_3, \cdots \cdots, \sim P_n, \cdots \cdots$  $
```

と、簡潔に表されますが、普通の数式モードではやっかいで、一例を挙げると

```
----- 普通の数式モード -----
 $\mathrm{P}_1, \sim \mathrm{P}_2, \cdots \cdots, \sim \text{P}_n, \cdots \cdots$ 
とか
 $P_1, P_2, \cdots \cdots, P_n, \cdots \cdots$ 
```

などと面倒になります。

なお、`caprm` 環境内で英大文字を斜体にしたいときは `\mathnormal` を用います。例えば

caprm で大文字を斜体

```
$BC=2\mathnormal{R}\sin \mathnormal{A}$
```

をタイプセットすると

```
BC = 2R sin A
```

となります。

(注) caprm 環境の中が数式モードになるわけではありません。caprm 環境の中で、数式モードに入ると、そこでは大文字は立体で、小文字は数式イタリックになるというだけです。

8.1.2 [o] オプション

caprm 環境内で、あるブロックは大文字をデフォルトの数式イタリック体に戻すには、その部分を [o] オプション付の caprm 環境で囲みます。

[o] オプション

```
\begin{caprm}
三角形 ABC において
\math[ BC=a, \sim CA=b, \sim AB=c ]
とすると,
\begin{caprm}[o]
三角形 ABC の面積  $S$  は
\math[ S=\sqrt{s(s-a)(s-b)(s-c)} ]
\end{caprm}
となる。ここで
\math[ s=\frac{BC+CA+AB}{2} ]
\end{caprm}
```

三角形 ABC において

$$BC = a, CA = b, AB = c$$

とすると、三角形 ABC の面積 S は

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

となる。ここで

$$s = \frac{BC + CA + AB}{2}$$

8.1.3 コマンド型

環境にせず、コマンドでローマン体、数式イタリック体を切り替える方法もあります。

`\caprm` ローマン体に

`\caprm[o]` 数式イタリック体に

— `\caprm` コマンド —

```

\caprm%
三角形 ABC において
\ [ BC=a, \sim CA=b, \sim AB=c ]
とすると, %
\caprm[o]\relax%
三角形 ABC の面積$$$は
\ [ S=\sqrt{s(s-a)(s-b)(s-c)} ]
となる。ここで
\caprm
\ [ s=\bunsuu{BC+CA+AB}{2} ]

```

三角形 ABC において
 $BC = a, CA = b, AB = c$
 とすると, 三角形 ABC の面積 S は

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

 となる。ここで

$$s = \frac{BC + CA + AB}{2}$$

8.1.4 [1] オプション

数式環境内で, 小文字もローマン体にしたときは, `caprm` 環境に [1] オプションをつけます。化学式などは小文字もローマン体です。

— `caprm` 環境の [1] オプション —

```

\begin{caprm}[1]
\Nd_2Fe_{14}B$ \longrightarrow Nd_2Fe_{14}B
\end{caprm}

```

8.1.5 `\mathRM`

数式環境内で大文字だけをローマン体にするとき, それが 1 個所だけであるなどというときは, 環境よりはコマンドの方が使いやすいでしょう。そのためのコマンド `\mathRM` です。例えば

— `\mathRM` —

```

\mathRM{A_nB_n}$ \longrightarrow A_nB_n

```

において, 大文字はローマン体, 小文字は数式イタリックでタイプセットされます。

8.1.6 `\bekutoru*`

始点, 終点に添え字がついた場合のベクトルは面倒です。これを処理するため, `caprm` 環境内での使用を前提とする `\bekutoru*` コマンドを作りました。`\bekutoru` との違いは,

`\bekutoru` の引数はテキストモードであるのに対し,
`\bekutoru*` の引数は数式モードである, ということです。

例えば、ベクトル $A_n A_{n+1}$ を両方のコマンドで表現してみます。

```

                                ¥bekutoru と ¥bekutoru* の違い
¥bekutoru{A$_n$A$_{n+1}$} →  $\overrightarrow{A_n A_{n+1}}$ 
¥bekutoru*{A_nA_{n+1}} →  $A_n A_{n+1}$ 

```

同様の趣旨のコマンドに ¥kaku*, ¥sankaku* があります。

8.1.7 ¥mitS など

caprm 環境のもとでは、英大文字はローマン体となります。しかし、面積を表す S は数式イタリックとするのが慣わしです。そのためには $\mathnormal S$ と記述すればよいのですが、タイピングを楽にするために ¥mitS と、略記することも可能としました。

```

                                ¥mitS など
¥begin{caprm}
  ¥sankaku{ABC}において
  ¥[ AB=4, ~BC=5, ~CA=6 ¥]
  のとき、次の問いに答えよ。
  ¥begin{enumerate}[(1)]
    ¥item ¥cos¥mitA$を求めよ。
    ¥item ¥sankaku{ABC}の面積 ¥mitS を求めよ。
  ¥end{enumerate}
¥end{caprm}

```

```

△ABC において
    AB = 4, BC = 5, CA = 6
のとき、次の問いに答えよ。
(1) cos A を求めよ。
(2) △ABC の面積 S を求めよ。

```

¥mitS などは、¥ensuremath をかぶせてありますから、

```
$¥cos¥mitA$
```

と、数式モードの中で使用してもよし、

```
面積 ¥mitS を求めよ。
```

などと、地の文中で使用することもできます。(ただし、¥mitS の後ろに半角空白を置いて、コマンドが “¥mitS を求めよ。” ではないことを明示する必要があります。)

8.2 インデント

条件部分の記述などで、字下げをしたいときがあります。

quote(quotation) 環境

を用いることもできますが、

- 左インデント量を変えたい、
- 右インデントは要らない、

といった場面で使うために、jqote(jqotation) 環境を用意しました。


```
¥begin{jquote}<key=val>(左インデント量)(右インデント量)[上下の行間隔]
```

key は

tsep=.. : 直前の段落との間隔調整 (ex. -¥topsep)

bsep=.. : 直後の段落との間隔調整

vsep=.. : tsep, bsep を同時指定

デフォルト値は

左インデント量 2zw

右インデント量 0pt

上下の行間隔 0pt

となっています。

jquotation 環境も同様です。

なお、これらの環境は jquote.sty で定義されていますから、使用する際は

```
¥usepackage{jquote}
```

としておく必要があります。

8.3 ¥EMparbox


この節では、ボックスのサイズを論じますので、¥fbox でボックスを枠線で囲んで表示します。その際、¥fboxsep=0pt としてあります。

8.3.1 ¥parbox

複数行からなるブロックをまとめて一つのボックスとするのに ¥parbox を用いることがあります。ただし、このコマンドはボックスの横幅を指定する必要があります。

— ¥parbox —

```
¥bigskip
¥fbox{%
  ¥parbox{8zw}{%
    あいうえお ¥¥
    かきくけこ ¥¥
    やゆよ}%
  }
¥bigskip
```




あいうえお
かきくけこ
やゆよ

8.3.2 ¥EMparbox

この横幅指定が煩わしいからさぼろう、というのが新設したコマンド ¥EMparbox です。

— %EMparbox —

```
%bigskip
%fbbox{%
  %EMparbox{%
    あいうえお %%
    かきくけこ %%
    やゆよ}%
}
%bigskip
```

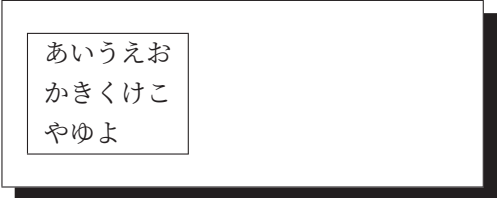


```
あいうえお
かきくけこ
やゆよ
```

もっとも、この例の場合は tabular 環境を使う方が普通でしょう。

— tabular 環境 —

```
%bigskip
%fbbox{%
  %begin{tabular}{l}
    あいうえお %%
    かきくけこ %%
    やゆよ
  %end{tabular}%
}
%bigskip
```



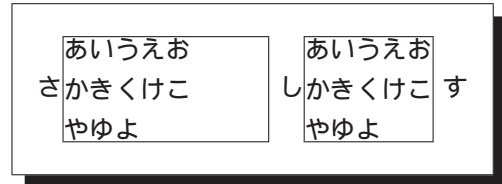
```
あいうえお
かきくけこ
やゆよ
```

8.3.3 %EMparbox のベースライン

そのベースラインは、%parbox と同じです。

ベースライン

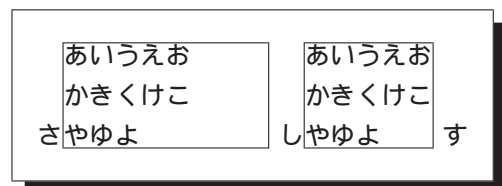
```
¥bigskip
さ
¥fbox{%
  ¥parbox{8zw}{%
    あいうえお ¥¥
    かきくけこ ¥¥
    やゆよ}%
}
し
¥fbox{%
  ¥EMparbox{%
    あいうえお ¥¥
    かきくけこ ¥¥
    やゆよ}%
}
す
¥bigskip
```



ベースラインを変更する方法も ¥parbox と同様です。

ベースラインの変更

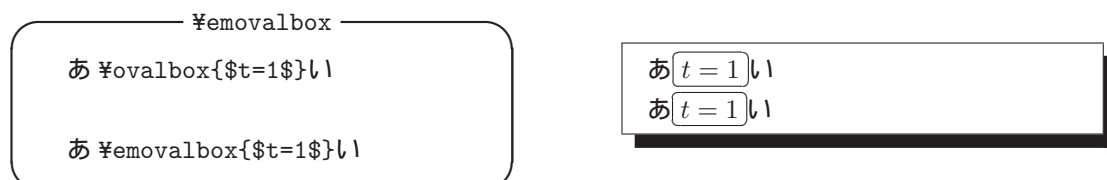
```
¥bigskip
さ
¥fbox{%
  ¥parbox[b]{8zw}{%
    あいうえお ¥¥
    かきくけこ ¥¥
    やゆよ}%
}
し
¥fbox{%
  ¥EMparbox[b]{%
    あいうえお ¥¥
    かきくけこ ¥¥
    やゆよ}%
}
す
¥bigskip
```



8.4 `\femovalbox`

`fancybox.sty` で定義されている `\fovalbox` は、`eepic.sty` を用いた場合、上下の罫線と右四分円とが繋がらない現象が発生します。

そこで、`emath.sty` に `\femovalbox` を新設しました。



コーナーの四分円の半径はデフォルトでは `\fboxsep` としてありますが、変更するには [...] オプションを用います。

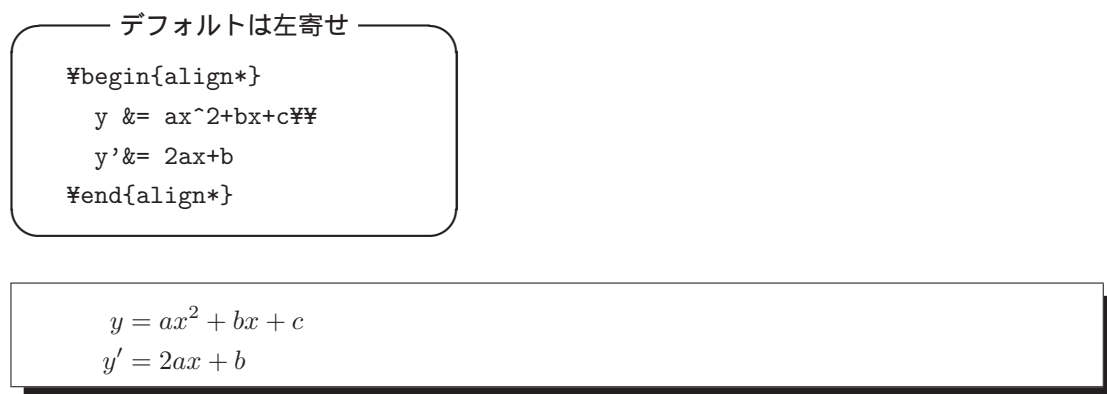


罫線の太さを変えたいなどに対しては、`emathPs.sty` で定義されている `\EMpsovalbox` をご使用ください。

8.5 数式の配置 左(右)寄せと中央揃えの切り替え

別行立て数式の配置は、`\documentclass` のオプションで `fleqn` を指定したときは左寄せ、そうでないときは中央揃えとなりますが、`fleqnon`、`fleqnoff` 環境を用いると、`fleqn` を指定していてもいなくても、2つの状態を切り替えることができます。

サンプルをごらんいただきましょう。この文書は `fleqn` オプションをつけていますから、数式は左寄せとなります。



`fleqnoff` 環境内では、数式は中央揃えとなります。

— `\fleqnoff` で中央揃え —

```
\begin{fleqnoff}
  \begin{align*}
    y &= ax^2+bx+c
    y' &= 2ax+b
  \end{align*}
\end{fleqnoff}
```

$$y = ax^2 + bx + c$$
$$y' = 2ax + b$$

なお、左寄せにしたときの左インデント量は `\mathindent` で指定することができますが、`\fleqnon` のオプション引数で指定することもできます。

—— 左インデント量の指定 ——

```
\begin{fleqnon}[4zw]\relax
  あいうえお
  \begin{align*}
    y &= ax^2+bx+c\relax
    y' &= 2ax+b
  \end{align*}
\end{fleqnon}
```

あいうえお

$$y = ax^2 + bx + c$$

$$y' = 2ax + b$$

このオプション引数のデフォルト値は `\leftmargini` です。
数式を右寄せすることはあまり行われませんが、`\tag*` を用いれば実現できます。

—— 数式の右寄せ ——

```
\begin{gather}
\tag*{\$x=-1}\relax
\tag*{\$y=1}
\end{gather}
```

$$x = -1$$

$$y = 1$$

8.6 数式番号を左に出力

`emath` では、数式番号は右端に出力します。

$$y = ax + b \quad \dots\dots\dots ①$$

$$y = px^2 + qx + r \quad \dots\dots\dots ②$$

文書全体で数式番号を左に出力するには、

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

とでもすればよいのですが、`emath` を使用する文書では、その必要が生ずることはないでしょう。
下の公式集は、次の要求にこたえたものです。

- (1) 公式にナンバリングをする。
- (2) 公式は等号の位置で桁揃えをする

(2) の桁揃えの要求がなければ、(1) のナンバリングは `enumerate` 環境で実現できるのですが、桁揃えをするとなると、数式部分は `align`, `alignat` などを使いたくなります。ということで、ナンバリングを数式番号を左に出すことで対応しようとしたのが下の結果です。

\EMleqno など

```

\begin{itemsquarebox}{公式 (その 1)}
\EMleqno%                この環境内のみ数式番号左配置
\EMsaveeqno%            数式番号退避
\resetcounter{equation}% 数式番号リセット
\preEqlabel{}%          数式番号に 3 点リーダをつけない
\def\tagform#1{#1.}%    数式番号の形式を変更
\vspace{-\baselineskip}
\begin{alignat}{3}
& (fg)' \quad \&=f'g \quad \&+fg' \quad \&
& (fgh)' \quad \&=f'gh \quad \&+fg'h+fg'h'
\end{alignat}
\EMrestoreeqno%         数式番号復元
\end{itemsquarebox}

```

公式 (その 1)

1. $(fg)' = f'g + fg'$
2. $(fgh)' = f'gh + fg'h + fgh'$

上の環境の外では、数式番号、形式などに影響がないことを確認します。

$ax + by + c = 0$ ③

$x^2 + y^2 = r^2$ ④

8.7 行末にマーク

証明終わりを表すマークを行末に置くためのコマンドが `\owari` です。

\owari

よって証明された。 `\owari`

→ よって証明された。 ■

このコマンドは別行立て数式行でも有効です。また、終わり記号を変更するには `\owari` の [...] オプションを用います。

別行立て数式内の \owari

```

\fleqnoff
\begin{equation}
y=2x^2-3x+1 \owari[\%Cdots (答)]
\end{equation}

```

→ $y = 2x^2 - 3x + 1 \quad \dots (答)$ ①

なお、`amsmath.sty` の ver.2 では上の例は

¥[y=2x^2-3x+1 ¥owari[¥Cdots (答)] ¥]
と簡潔に書くことができます。

行末に「... (答)」とすることは多いので、省略コマンド ¥kotae もあります。これはオプションで3点リーダーの個数を指定することができます。一例です。

¥kotae
\$¥therefore ~ x=2\$ ¥kotae[4] → ∴ x = 2 (答)

位置を行末から少し内側に寄せたいときは<...>オプションを用います。

¥kotae
\$¥therefore ~ x=2\$ ¥kotae[4]<2zw> → ∴ x = 2 (答)

8.8 リーダー罫

文字列と文字列の間を ... でつなげるのに、¥dotfill コマンドがあります。

¥dotfill
\$y=ax^2+bx+c\$¥dotfill 放物線

→ $y = ax^2 + bx + c$ 放物線

点々の位置を ¥baseline ではなく、少し上にあげたい、というのでコマンド ¥cdotfill を作りました。

¥cdotfill
\$y=ax^2+bx+c\$¥cdotfill 放物線

→ $y = ax^2 + bx + c$ 放物線

このコマンドは [#1] オプションでリーダー罫の長さを指定することができます。

¥cdotfill の長さ指定オプション
\$a_1\$, \$a_2\$, ¥cdotfill[3em], \$a_n\$, ¥cdotfill[3em]

→ $a_1, a_2, \dots, a_n, \dots$

左右の文字列とリーダー罫あわせた長さを指定する例です。

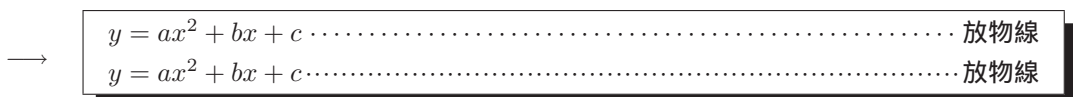
¥cdotfill の長さ指定オプション
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 ¥par¥noindent
¥makebox[16zw]{\$y=ax^2+bx+c\$¥cdotfill 放物線}

→ $y = ax^2 + bx + c$ 放物線

点と点の間隔を調整したいときは、`\cdotskip` を再定義します。
 デフォルトは `\def\cdotskip{1.5mu}` となっています。

```

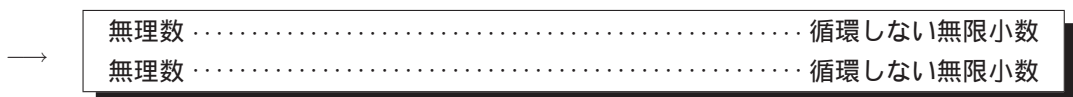
\cdotskip の変更
$y=ax^2+bx+c\cdotfill 放物線 \par\noindent
\def\cdotskip{0.5mu}%
$y=ax^2+bx+c\cdotfill 放物線
```



左右の文字列がともに全角文字のときは、リーダー罫の位置を少し上に上げたい、というのが `\Cdotfill` です。

```

\Cdotfill
無理数 \cdotfill 循環しない無限小数 \par\noindent
無理数 \Cdotfill 循環しない無限小数
```



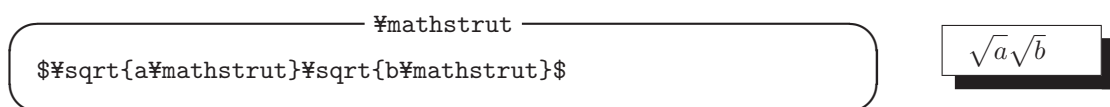
`\Cdotfill` にも、長さ指定オプションがあります。その使い方などは `\cdotfill` と同様です。

8.9 支柱

高さを揃えるには、支柱を使います。
 まずは、高さ・深さが不揃いな例をご覧ください。



a, b の高さ・深さが異なるため、根号が不揃いとなっています。このような場面では、`\mathstrut` という支柱が使われます。



`\mathstrut` は、`'(` の高さ・深さをもつ幅が 0 の支柱です。plain.tex で

```
\def\mathstrut{\vphantom{}}
```

と定義されており、 \LaTeX もその定義を採用しています。

```
\vphantom{\#1}
```

は、`\#1` と同じ高さ・深さをもつ支柱ですが、次の場合は分数 $\frac{1}{2}$ よりも少し高さ・深さを増やした支柱がほしくなります。

—— 天地寸詰まりの表 ——

```


$$\begin{array}{|c|c|} \hline x & y \\ \hline 2 & \frac{1}{2} \\ \hline \end{array}$$


```

x	y
2	$\frac{1}{2}$

emath.sty では、`\EMvphantom` なるコマンドでそれを実現しています。

—— `\EMvphantom` ——

```


$$\begin{array}{|c|c|} \hline x & y \\ \hline 2 & \EMvphantom[4pt][3pt]{\bunsuu12}\bunsuu12 \\ \hline \end{array}$$


```

x	y
2	$\frac{1}{2}$

`\EMvphantom[4pt][3pt]{\bunsuu12}` で、分数 $\frac{1}{2}$ よりも、

高さを 4pt,

深さを 3pt

増やした支柱を作っています。この支柱は利用頻度が高そうですから、別名を `\bsityuu` としています。

—— `\bsityuu` ——

```


$$\begin{array}{|c|c|} \hline x & y \\ \hline 2 & \bsityuu\bunsuu12 \\ \hline \end{array}$$


```

x	y
2	$\frac{1}{2}$

`\EMvphantom`, `\bsityuu` の書式です。

$\text{\textbackslash EMvphantom}[#1][#2]\#3$
 $\#3$ の高さに $\#1$ を附加し,
 $\#3$ の深さに $\#2$ を附加した支柱を表す。
 $\#1$ のデフォルト値は 0pt
 $\#2$ のデフォルト値は $\#1$

$\text{\textbackslash EMvphantom}^*[#1][#2]\#3$
 支柱を立てるだけではなく、 $\#3$ をタイプセットする。

$\text{\textbackslash bsityuu}[#1][#2]$
 分数 2 分の 1 の高さに $\#1$ を附加し,
 分数 2 分の 1 の深さに $\#2$ を附加した支柱を表す。
 $\#1$ のデフォルト値は 4pt
 $\#2$ のデフォルト値は 3pt

8.10 下線

8.10.1 $\text{\textbackslash underline}$

下線を引くコマンドは $\text{\textbackslash underline}$ です。

$\text{\textbackslash underline}$
 下線を引きます。 $\text{\textbackslash underline}\{\text{あいうえお}\}$
 $\text{\textbackslash underline}\{\text{ただし、このコマンドは、行を}\}$
 $\text{\textbackslash underline}\{\text{またぐことができません。}\}$

下線を引きます。 あいうえお
 ただし、このコマンドは、行をまたぐことができません。

複数行に渡る下線を引くスタイルファイルに `jumoline.sty` があります。
`jumoline.sty` については、その作者である 中島 浩さんのホームページ

<http://www.para.tutics.tut.ac.jp/~nakasima/latex/>

をご覧ください。

8.10.2 波下線

8.10.2.1 `ulem.sty` 波下線を引くには、`ulem.sty` で定義されている $\text{\textbackslash uwave}$ を用いるのが一般的な方法です。

$\text{\textbackslash uwave}$
 あいうえお
 $\text{\textbackslash uwave}\{\text{かきくけこ}\}$
 さしすせそ

あいうえお かきくけこ さしすせそ

数式に適用するときは、

数式に対する `\uwave`

`\cmd{uwave}` の引数内で数式モードにしてもだめで、
`\uwave{\$dlim{x\to\infty}\bunsuu1x=0\$}`
 数式モード内に `\cmd{uwave}` を配置しなければなりません。
`\$uwave{\$dlim{x\to\infty}\bunsuu1x=0\$}`

`\uwave` の引数内で数式モードにしてもだめで、

$$\lim_{x \rightarrow \infty} \frac{1}{x} = 0$$

数式モード内に `\uwave` を配置しなければなりません。

$$\lim_{x \rightarrow \infty} \frac{1}{x} = 0$$

8.10.2.2 `\namikasen` ところで、`dviout.exe` でみている限り綺麗なのですが、pdf にしてみると、波のつながり具合が乱れてきます。ということもあって、`emath.sty` に `\namikasen` を新設しました。こちらは、コマンドの引数を数式モードにする方式です。

`\namikasen`

あいうえお
`\namikasen{\$dlim{x\to\infty}\bunsuu1x=0\$}`
 さしすせそさしすせそさしすせそさしすせそ
 さしすせそさしすせそさしすせそさしすせそ

あいうえお $\lim_{x \rightarrow \infty} \frac{1}{x} = 0$ さしすせ
 そさしすせそさしすせそさしすせ
 そさしすせそさしすせそさしすせ
 そさしすせそ

ただし、行をまたがることはできません。

なお、`\namikasen` を用いるには、`ulem.sty` は不要です。

8.10.2.3 波二重下線 `\uwave` による波下線を二重にするコマンド `\uuwave` を `emath.sty` に新設しました。ただし、この下請けは `ulem.sty` にありますから、このスタイルファイルを読み込んでおく必要があります。

`\uuwave`

`\$uuwave{\$dlim{x\to\infty}\bunsuu1x=0\$}`
 あああああああああああああああああ
 あああああああああああああああああ

$\lim_{x \rightarrow \infty} \frac{1}{x} = 0$ ああああああああ
 ああああああああああああああああ
 ああああああああああああああ

`\namikasen` の方で、波下線を二重とするには、`\namikasen` に `[1.5]` などと、オプションを与えます。`[...]` 内の数値は、二重線の間隔（単位は pt）です。

`\namikasen[...]`

`\namikasen[1.5]{%`
`\$dlim{x\to\infty}\bunsuu1x=0\$}`
 あああああああああああああああああ
 あああああああああああああああああ

$\lim_{x \rightarrow \infty} \frac{1}{x} = 0$ ああああああああ
 ああああああああああああああああ
 ああああああああああああ

8.10.2.4 太波下線 太い波下線を引くには、コマンド `\hutonamikasen` を用います。

```

 $\namikasenUehosei$ 
 $\namikasen$ {かきくけこ}

 $\hutonamikasen$ {かきくけこ}

 $\hutonamikasen[2]$ {かきくけこ}

```



ただし、 \hutonamikasen は、内部で \scalebox を用いていますから、`graphicx.sty` を必要とします。emath では、`emathP*.sty` を用いれば、`graphicx.sty` は自動的に読み込まれます。

8.10.2.5 \namikasen と上下のアキ

\namikasenUehosei \namikasen とその下線を引く対象との間隔を調整するコマンドが \namikasenUehosei です。

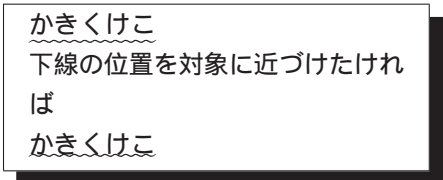
```

 $\namikasenUehosei$ 
 $\namikasen$ {かきくけこ}

下線の位置を対象に近づけたければ

 $\namikasenUehosei$ {-2pt}
 $\namikasen$ {かきくけこ}

```



\namikasenUehosei の引数に、波線縦位置を補正する数値（単位付）を与えます。正の値を与えれば下に、負の値を与えれば上に動きます。

\namikasenSitahosei 逆に下線の下、すなわち下線と下の行との間隔を調整するのが \namikasenSitahosei です。

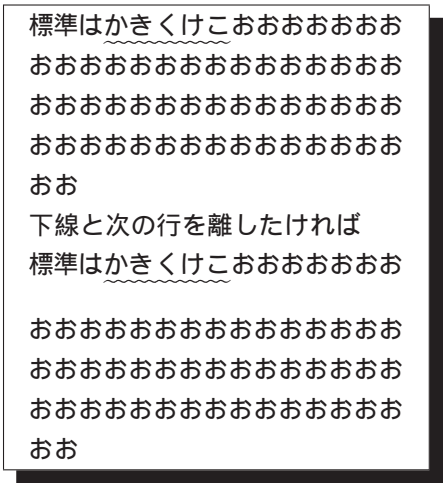
```

 $\namikasenSitahosei$ 
標準は  $\namikasen$ {かきくけこ}
おおおおおおおおおおおおおお
おおおおおおおおおおおおおお
おおおおおおおおおおおおおお
おおおおおおおおおおおおおお

下線と次の行を離れたければ

 $\namikasenSitahosei$ {10pt}
標準は  $\namikasen$ {かきくけこ}
おおおおおおおおおおおおおお
おおおおおおおおおおおおおお
おおおおおおおおおおおおおお

```



上の例、後半では \namikasenSitahosei {10pt} として、波下線と次の行との間隔をデフォルトより 10pt 増やしています。

9 メモリ不足への対応

9.1 pool size

図の描画には多くのメモリを必要とするようで、1 ページの図が多いときなど

```
! TeX capacity exceeded, sorry [pool size=67010].
```

といったエラーメッセージに遭遇することがあります。

この場合は、`texmf.cnf` というファイルを編集します。編集する前に、フロッピなどにオリジナルを保存した上でお願いします。

このファイルは \TeX をインストールしたフォルダの

```
...%texmf%web2c
```

に存在します。このファイルはテキストファイルですから、エディタで開いて

```
pool_size = 125000
```

という行を探します。(右辺値は異なるかもしれません。)

この式の右辺値を 300000 などと修正し、上書き保存します。

```
pool_size = 300000
```

なお、機種によってはこのファイルの拡張子が表示されないかもしれません(単に `texmf` と表示されます)。

9.2 main memory size

```
! TeX capacity exceeded, sorry [main memory size=263001].
```

といったエラーメッセージの処理は、かなり厄介です。

以下の記述では、 \TeX が `c:%ptex` にインストールしてあるものとします。インストールドライブ、フォルダが異なる場合は読み替えてください。

(1) 修正するファイルのバックアップの作成

修正するファイルは

```
texmf.cnf, platex.fmt
```

の2つです。現在、`c:%ptex%texmf%web2c` にあるものをフロッピ、MO、などにコピーします。

(2) `texmf.cnf` の修正

エディタで `texmf.cnf` を開きます。検索機能を利用して

```
main_memory
```

を探してください。

```
main_memory = 263000 % words of inimemory available; also applies to inimf&mp
```

という行が見つかるはずですが、ここを修正します。私は

```
main_memory = 526000 % words of inmemory available; also applies to inimf&mp
```

と修正しました。ついでに、その少し下の

```
pool_size = 125000
```

も

```
pool_size = 250000
```

と増やしました。

(3) platex.fmt の作成

DOS 窓で、キーボードから

```
c:
cd %ptex%\texmf\tex\platex\base
```

と打ち込み、

```
c:%ptex%\texmf\tex\platex\base
```

をカレントフォルダにします。

その上で、キーボードから

```
ptex --ini platex.ltx
```

と打ち込み、リターンキーを押します。いろいろなメッセージが流れますが、作業が終了すると、DOS Prompt が出ます。

(注) ここで \LaTeX のバージョンが古いと

```
!Latex source files more than 1 year old!.
1.573 ...LaTeX source files more than 1 year old!}
```

などのメッセージが表れることがあります。このときは委細構わず、リターンキーを押せば作業は続行されます。もちろん、この機会に \LaTeX のバージョンアップをされることをお勧めしますが.....

(4) platex.fmt を所定の場所に移す。(3) の作業の結果、

```
c:%ptex%\texmf\tex\platex\base
```

に二つのファイル

```
platex.fmt, platex.log
```

が作成されます。前者を

```
c:%ptex%\texmf\web2c
```

にコピーします。

(5) c:%ptex%\texmf\tex\platex\base に作成された platex.fmt, platex.log を削除します。

9.3 hash size

メモリ不足もいろいろあります。

```
! TeX capacity exceeded, sorry [hash size=10000].
```

への対処法です。

こちらは簡単で、`texmf.cnf` を書き換えるだけです。

エディタで `texmf.cnf` を開き 'hash' で検索をかけると

```
hash_extra = 0
```

という行が見つかります。この右辺値を、たとえば 2000 などとしてやれば OK です。フォーマットファイルを作り変える必要はありません。

9.4 save size

```
! TeX capacity exceeded, sorry [save size=4000].
```

への対処法です。

これも `texmf.cnf` を書き換えるだけです。

`texmf.cnf` 中の一行

```
save_size = 4000 % for saving values outside current group
```

の数値を例えば、8000 に増やします。

フォーマットファイルの作り直しは不要です。

9.5 number of strings

```
! TeX capacity exceeded, sorry [number of strings=10603].
```

への対処法です。

これも `texmf.cnf` を書き換えるだけです。

`texmf.cnf` 中の一行

```
max_strings = 15000 % max number of strings
```

の数値を例えば、20000 に増やします。

フォーマットファイルの作り直しは不要です。

10 謝辞

このマクロ集の作成については、Waver さんからいろいろと有用なご助言をいただきました。また、7.6 節のマクロはすべて Waver さんから使用を認めていただいたものです。

また、平行四辺形の記号については、トニイさんからここでの使用を認めていただきました。あわせて感謝の意を表します。

索引

- apnenum=, 100
- ¥apnlist, 117
- ¥atag, 38

- ¥baaiwake, 37
- ¥beku, 12
- ¥bekusityuu, 13
- ¥bekutoru, 12
- ¥bekutoru*, 125
- ¥bekutorukata, 13
- ¥bekutorusityuu, 13
- betaenumerate 環境, 90
- ¥bsityuu, 139
- ¥bunsuu, 1

- caprm 環境, 123
- ¥Cdotfill, 138
- ¥cdotfill, 137
- ¥Cnum, 19
- ¥conj, 19

- ¥DEG, 21
- ¥Deg, 21
- ¥degree, 21
- ¥Deruta, 25
- ¥dint, 25
- ¥dlim, 23
- ¥dpower, 11
- ¥dumyeq, 39

- edaenumerate 環境, 83
- ¥edaitem, 83
- ¥edasikiri, 88
- edatopsep=, 111
- ¥EMabs, 14
- ¥emabs, 15
- ¥EMcaption, 82
- emcases 環境, 36
- ¥EMleqno, 135
- ¥emovalbox, 133
- ¥EMparbox, 130

- ¥EMrestoreeqno, 135
- ¥emRoman, 95
- ¥emroman, 94
- ¥EMsaveeqno, 135
- ¥EMvphantom, 139
- ¥enumLmargin, 105
- ¥enumSep, 104

- fleqnoff 環境, 133
- fleqnon 環境, 133
- Fmawarikomi 環境, 51
- ¥Fsyndiv, 8

- ¥gauss, 17
- ¥getcurrentenum, 98
- ¥gyoube, 18
- gyoukan=, 86
- ¥Gyouretu, 18
- ¥gyouretu, 17

- ¥heikou, 21
- ¥heikousihenkei, 23
- ¥hutonamikasen, 141

- ¥item*, 93
- ¥item**, 93
- ¥itemtopmath, 41
- ¥izyohou, 7

- jquotion 環境, 130
- jquote 環境, 130

- ¥kagenhou, 9
- ¥kaizyou, 26
- ¥kaku, 21
- ¥kaku*, 126
- ¥kinzi, 1
- kizamiti=, 98
- ¥kmaru, 29
- ¥knagamaru, 30
- ¥ko, 22
- ¥ko*, 22

¥kotae, 137
 ¥Kumiawase, 27
 ¥kumiawase, 26
 ¥kynagamaru, 31
 ¥kyokukeisiki, 20
 ¥kyouyaku, 19

 ¥marudashref, 38
 ¥marudashtag, 38
 ¥mathindent, 32, 33
 ¥mathRM, 125
 mawarikomi 環境, 44, 53
 mawarikomi* 環境, 44
 ¥mawarikomikankaku, 55
 ¥mawarikomiowari, 56
 ¥mawarikomisep, 47
 ¥mitS, 126

 ¥nagamaru, 30
 ¥namikasen, 141
 ¥narrowenumsep, 105
 ¥narrowsep, 117
 ¥necarrow, 24
 ¥neqq, 1
 ¥nevarrow, 24
 ¥nheikou, 21
 nidan 環境, 56

 ¥Ofil, 95
 ¥owari, 136

 ¥pgyouretu, 19
 ¥polar, 20
 postedasep=, 110
 ¥postitem, 93
 preedasep=, 108
 ¥preEqlabel, 32
 ¥preitem, 93
 ¥pxpercent, 3

 ¥refallexnum, 122
 ¥refcurrentenum, 121
 ¥resetcounter, 34
 ¥resettagform, 33
 ¥Retube, 18

 ¥retube, 18
 retusuu=, 86
 ¥retuwa, 23

 ¥sankaku, 21
 ¥sankaku*, 126
 ¥secarrow, 24
 ¥setcurrentenum, 98
 ¥sevarrow, 24
 ¥shikaku, 22
 ¥souzi, 21
 ¥sqrt, 10
 ¥stasuki, 8
 ¥suuretu, 23
 ¥syndiv, 7
 syokiti=, 98
 ¥syugou, 26

 ¥tagform, 33
 ¥Tasuki, 9
 ¥tasuki, 8
 ¥tasukikata, 9
 ¥tbEMcaption, 82
 ¥Teisei, 5
 ¥teisei, 3
 ¥teisekibun, 25
 Tmawarikomi 環境, 51
 ¥tretuwa, 23
 ¥tsqrt, 11

 ¥wakudori, 74

 ¥Yakubun, 2
 ¥yakubun, 1
 ¥ynagamaru, 31
 yokoenumerate 環境, 91

 ¥zettaiti, 14
 ¥zuhaiti, 72
 ¥zyohou, 6
 ¥Zyunretu, 27
 ¥zyunretu, 26

 円弧, 22
 凹凸, 24

角 (), 21
角度の記号, 21

弧, 22

三角形, 21
3乗根, 10

シグマ, 23
斜体と立体, 123

数式のインデント, 33
数式番号, 32
数式番号の書式, 33
数式番号のリーダー罫, 32
数列記号, 23
数列の和, 23

絶対値記号, 14

増減と凹凸, 24
増減表, 24
相似, 21

度の記号, 21

二重根号, 11

平行, 21
平行四辺形, 22

約分, 1, 2, 5

立体と斜体, 123