

# Web 利用の理数教育に役立つ数式送受システムの開発

東邦大学・理学部 高遠 節夫

Setsuo Takato, Faculty of Sciences, Toho University

長野高専・一般科 濱口 直樹

Naoki Hamaguchi, National Institute of Technology (KOSEN), Nagano College

## 1 オンライン授業と KeTCindy


コロナ禍の中で、多くの教育機関ではオンライン授業（以下、オンラインという）の実施を余儀なくされることになった。今後感染拡大が収束するにつれて、オンラインの割合は減少するにせよ、対面授業（以下、対面という）とは異なる有効性が認識されて、それと併用されていくことが予想される。

理数の授業では、数式と図が重要な要素であり、これらを黒板にかいて説明を加えるのが通常の形態であるが、オンラインでの実現には苦心することが多い。すなわち、オンデマンド形式では、黒板での講義をそのまま録画することがよく行われるが、動画ファイルは編集や修正が容易ではない上に、授業のシナリオを対面以上に考えて作っておかなければ、インパクトに欠けるものになってしまう恐れがある。また、オンライン形式で、手書きボードなどのアプリとマーカーを使うにしても、図表や基礎となる数式は予め書き込んでおいた方が授業を進めるために望ましいであろう。

私たちのメンバーが開発している KeTCindy<sup>1</sup>は、動的幾何ソフト Cinderella を用いて TeX 文書に挿入する図表を次の手順に沿って作成するものである。

- (1) Cinderella を立ち上げて、必要な幾何点 (A,B,C,D) をとり、CindyScript エディタにコマンド列を書く。

- ・ Ospline は大島利雄教授提案の Bézier 曲線
- ・ 幾何点 P は 3 行目のコマンドで定める。



```
1 Ketinit();
2 Ospline("1", [A,B,C,D]);
3 Putoncurve("P", "bzol");
4 m=Derivative("bzol", "x="+P.x);
5 Lineplot("1", [P, P+[1,m]]);
6 Windisp();
7
```

- (2) Cinderella の画面に図が現れる。

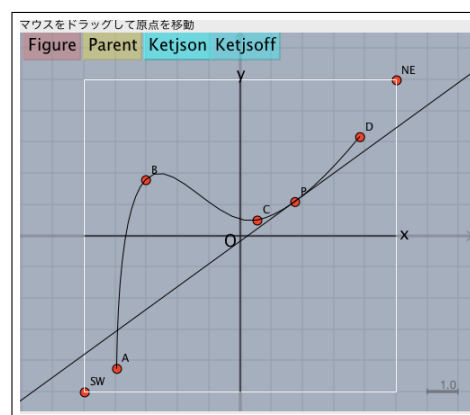
- ・ A,B,C,D は自由に動かすことができる。
- ・ P は曲線上を動かすことができる。

- (3) 左上の Figure ボタンを押すと、KeTCindy から R が実行されて、TeX の描画コードからなるファイルが生成される。

- ・ 描画コードは Tpic, pict2e, Tikz をサポート

- (4) 生成された図を TeX コマンド `\input` を用いて TeX 文書に挿入する。

- ・ `\includegraphics` ではない。

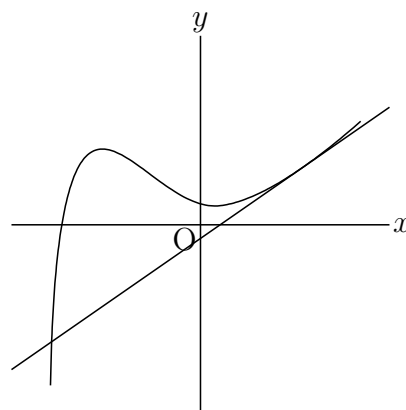


<sup>1</sup>KeTCindy のパッケージは CTAN>ketcindy>package>repository からダウンロードできる。

右の図は、`\input` によって本稿に挿入されている。ここでは、やはり私たちが開発した `ketlayer.sty` で定義されている環境 `layer` を用いて、次のように図の配置を定めている。

```
\begin{layer}{160}{40}
\putnote{94}{5}{\input{fig.tex}}
\end{layer}
```

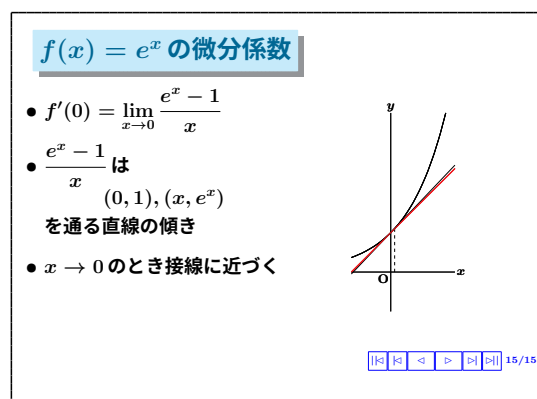
ここで、`layer` の引数は Grid を描く横幅と縦幅である（単位 mm）。また、`\putnote` は基準点の `se`（南東）に図を配置するコマンドである。



横幅を 0 とすると Grid が消去されて、図だけが表示される。なお、他の文字の位置には影響しない点が重要で、これにより図を自由に配置することができるようになる。

オンラインではスライドを用いたプレゼンテーションが多い。このとき、数学系の授業用スライドの作成には、図表や数式が容易に作成して自由に配置することが必要である。通常の論文や教材作成において、数式入力には  $\text{T}_\text{E}\text{X}$  が便利であり、スライド作成にも Beamer などの  $\text{L}_\text{A}\text{T}_\text{E}\text{X}$  パッケージが用いられることも多いが、その場合でも `layer` 環境は有用である。さらに、 $\text{K}_\text{E}\text{T}\text{Cindy}$  では、以下のような簡易記法で書かれたテキストファイルからスライド用の  $\text{T}_\text{E}\text{X}$  ファイルを生成、コンパイルしてスライド形式の PDF を作成することもできる。

```
1:new::\f(x)=e^x$の微分係数
2:%repeat=15
3:%para=slope:{0}:s{100}{10}:input
4:layer::{120}{0}
5:putnote::s{100}{10}::exptan
6:end
7:itemize
8:item::\f'(0)=
9:  \dlim_{x\to 0}\dfrac{e^x-1}{x}$
10:%[2,-]::item::\dfrac{e^x-1}{x}$は\
11:%[3,-]::\hspace{15mm}$(0,1),(x,e^x)$\
12:%[3,-]::を通る直線の傾き
13:%[15]::item::\x\to 0$のとき接線に近づく
14:end
```



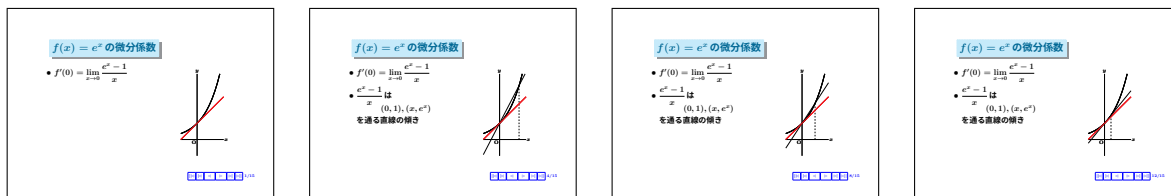
上記スクリプトのうち、4行-6行と7行-14行はそれぞれ `layer` 環境と `itemize` 環境の省略記法である。2行の `%repeat` はこのページの番号を変えずに指定のコマ数をおくことの指示で、3行のパラパラ動画（後述）のために15コマをとることとしている。また、10行の `%[2,-]` は2コマ以降にこの行を活かすことを意味しており、11行-13行も同様である。

パラパラ動画は各コマのスライドを順次表示することで実現される。この場合も、図などの要素の位置を正確に指定する `layer` 環境が重要である。具体的には、パラメータ

値  $s$  のときの状態を記述する関数  $\text{mf}(s)$  (上例では  $(0, 1)$ ,  $(0.1s, e^{0.1s})$  を通る直線を描くコマンドなどから成る) を定義し, CindyScript で

```
Setpara("slope", "mf(s)", reverse(1..15));
```

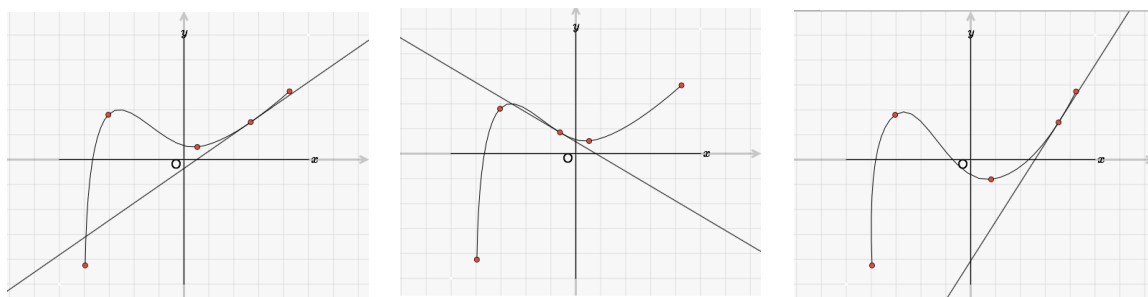
を実行してから, Cindy 画面で ParaF ボタンを押すと, サブフォルダ slope 内に各コマの図が生成される. 上記スクリプトの 3 行は, それらの図を layer 環境で表示するコマンドである.



このように, K<sub>E</sub>TCindy のスライド作成機能は使いやすく, オンラインの教材作成に適しているといえる. 特にパラパラ動画<sup>2</sup>は, オンラインにおいて有効な教育効果が期待される. ただし, PC のようにフルスクリーンなど画面を瞬時に切り替える機能が必要であり, タブレットやスマホで学生が自習するには向いていない. そこで, K<sub>E</sub>TCindy にインタラクティブな HTML 教材を作成するライブラリを追加した. これを私たちは K<sub>E</sub>TCindyJS と呼んでいる. 次節では, K<sub>E</sub>TCindyJS について詳述する.

## 2 K<sub>E</sub>TCindyJS による HTML 教材作成

ミュンヘン工科大学のグループは, 2016 年に CindyJS を公開した [3]. これは, Cinderella とほぼ互換性のある JavaScript のコマンド群であり, Cinderella の「HTML に書き出す」メニューによって, Cinderella 画面および CindyScript から成る HTML ファイルを作成することができる. ただし, このままでは K<sub>E</sub>TCindy のコマンドを用いることができないため, 教材作成には不十分である. そこで, CindyJS で書き出された HTML に CindyScript で使われている K<sub>E</sub>TCindy のコマンドだけを追加する K<sub>E</sub>TCindyJS を開発した [4]. 例えば, 1 節の最初の例で, そのまま Cindy 画面にある Ketjson または Ketjsoff のボタンを押せば, 幾何点を自由に動かすことができるインタラクティブな HTML が作成される. ここで, Ketjson は CindyJS のライブラリをオンラインで取得し, Ketjsoff は HTML ファイルと同じ階層にある ketcindyjs フォルダからオフラインで取得する. また, 数式表示には KaTeX を利用するが, そのライブラリについても同様である.



<sup>2</sup>TeX の animate.sty も利用できるが, AdobeReader に限定されるため使いやすいとはいえない.

このHTML ファイルは約 2200 行、サイズは 50KB 程度で、ketcindyjs フォルダのサイズは 1.5MB 程度と、いずれもコンパクトであり、したがって動作も軽快である。

もう 1 つの例として、Basel 問題

$$\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots = \frac{\pi^2}{6}$$

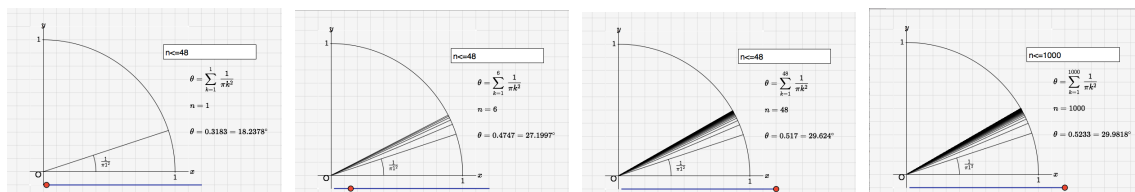
の HTML 教材を作成する。この級数について、平田典子教授の研究室のグループが

$$\frac{1}{1^2\pi} + \frac{1}{2^2\pi} + \frac{1}{3^2\pi} + \cdots = \frac{\pi}{6}$$

と変形して、角の和が  $\frac{\pi}{6}$  になることを GeoGebra で作図した教材案を提示した [2]。以下は、このアイデアに基づいて K<sub>E</sub>T CindyJS で発展させた教材の作成例であり、CindyScript の主要部分は次の通りである。

```
1:Slider("A",[0.25,-1],[12,-1]);
2:r=10; p0=[0,0]; p1=[r,0];
3:Circledata("1",[p0,r],["Rng=[0,pi/2]"]);
4:strn="n<=48";
5://strn=Textedit(51);//only ketjs
6:tmp=Strsplit(strn,""); tmp=parse(tmp_2)/48*4; n=max([round(A.x*tmp),1]);
7:strk="fr(1,pi*k^2)";
8:akc=Tocindyform(strk);
9:akt=Totexform(strk);
10:tmp="\displaystyle\theta=\sum_{k=1}^{\text{text(n)}}\{\text{+akt+}\}";
11:Expr([11,7],"e",tmp,["Size=1.5"]);
12:t=0;
13:forall(1..n,k,
14:  t=t+parse(akc);
15:  pP=r*[cos(t),sin(t)];
16:  Listplot(text(k),[p0,pP]);
17:);
18:str="\theta="+format(arctan2(pP),4)+"="+format(tmp*180/pi,4)+"^{\circ}";
19:Expr([11,3],"e",str,["Size=1.5"]);
20:tmp=Assign("Expr="+akt,["k","1"]);
21:Anglemark("1",[p1,p0,Ptend("sg1")],[8,tmp,"Size=1.5"]);
22:Expr([11,5],"e","n="+text(n),["Size=1.5"]);
23:Expr([10,0],"s","1",[0,10],"w","1",["Size=1.5"]);
```

1 行では項数を決めるスライダを作る。5 行-6 行では、K<sub>E</sub>T CindyJS の入力窓から文字列 strn を取得して項数の最大値を決めている。7 行は次節で説明する数式の簡易記法であり、8 行と 9 行でそれぞれ Cindy 書式と T<sub>E</sub>X 書式に変換した文字列を求めている。13 行-17 行で線分によって積み重ねられる角度を表示する。



左から,  $\frac{1}{\pi}, \sum_{k=1}^6 \frac{1}{k^2\pi}, \sum_{k=1}^{48} \frac{1}{k^2\pi}, \sum_{k=1}^{1000} \frac{1}{k^2\pi}$  の図であり, 右端の図では入力窓で  $n$  の値を変えている.

この HTML について, 大島利雄教授から上からと下からの評価に関するいくつかのコメントをいただいた [1]. それに基づいて, 評価の線分を加えた HTML を作成した. 下の図は

$$A_n = \pi\theta_n = \sum_{k=1}^n \frac{1}{k^2}$$

とおくとき, それぞれ

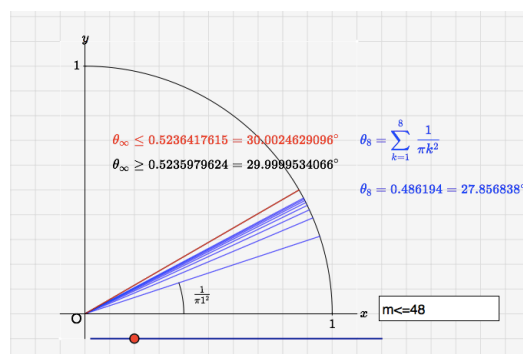
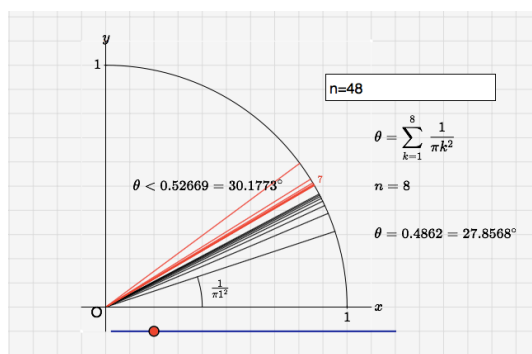
$$A_\infty < \sum_{k=1}^m \frac{1}{k^2} + \sum_{k=m+1}^{\infty} \frac{1}{k(k-1)} < \sum_{k=1}^m \frac{1}{k^2} + \frac{1}{m}$$

および

$$A_\infty < \sum_{k=1}^m \frac{1}{k^2} + \sum_{k=m+1}^{\infty} \frac{1}{(k+\frac{1}{2})(k-\frac{1}{2})} < \sum_{k=1}^m \frac{1}{k^2} + \frac{1}{m+\frac{1}{2}}$$

$$A_\infty > \sum_{k=1}^m \frac{1}{k^2} + \frac{1}{m+\frac{1}{2}} - \frac{1}{12(m^2-\frac{1}{4})(m+\frac{3}{2})}$$

の評価を表す線分を描き入れたものである.



この例でもわかるように, KeTCindyJS による HTML 教材は, いったん作成したものを教員自身または学生が実際に動かしてみることでいろいろな知見を得て, さらに作り込んでいくことができるものである. その際, Cinderella が CindyScript という使いやすいプログラム言語をもっていることが大きく, また, 教員のプログラミングの基礎的な力が教材の出来具合に大きな影響を与えている.

KeTCindy のサンプルページ (Samples of KeTCindy)

<https://s-takato.github.io/ketcindysample/>

には, KeTCindy だけでなく KeTCindyJS についても多くの例が掲載されている. 特に, Basel 問題の HTML 教材は次の URL などにある.

<https://s-takato.github.io/ketcindysample/misc/offline/basel1mainoff.html/>

<https://s-takato.github.io/ketcindysample/misc/offline/basel4mainoff.html/>

### 3 数式送受システムの開発

これまで述べたように, KeTCCindy は, 図入り配付教材 (PDF), 授業用スライド (PDF), インタラクティブ教材 (HTML) のいずれの作成においても有用な, すなわち, 教員から学生に配付または提示する教材作成のための総合的支援ツールである. さらに, オンラインでも対面でも役立つものであるといつてよい. 一方, 小問からなる課題の配付, 提出, フィードバックのルーチン (以下, 小課題ルーチンという) は, 授業内容を理解させるために必須な授業要素であるが, 対面とオンラインとでは形態が大きく異なる. 小課題ルーチンは, (1) 課題を配付, (2) 解答を提出, (3) 添削して返却, のステップが一般的であるが, 理数系で記述式の解答の場合は, 数式のやりとりが必要となる. 対面では手書きが普通で簡便であるが, オンラインで手書きを用いるには問題が多い. 解答を書いたものをスマホなどで撮るとファイルサイズ, したがって送信のデータ量が大きくなってしまう. 電子ペンでファイルに直接書き込めばデータ量は小さくなるが, その環境が使えない学生もいる. いずれにせよ, 別ファイルとして添付しなければならない. 周知のように, 数式の表示には  $\text{T}_{\text{E}}\text{X}$  書式による記法や計算ソフトのコマンドのような 1 次元表現と手書きや  $\text{T}_{\text{E}}\text{X}$  の出力などの 2 次元表現がある. 1 次元表現の方が送受のやりとりに向いているが, そのままでは数式の意味を捉えることが難しい. そこで,  $\text{T}_{\text{E}}\text{X}$  をベースとした簡易数式記述ルールを策定して, その 1 次元数式を 2 次元数式に変換するシステムの開発を今年度の授業と並行して進めることにした.

最初に定めた簡易数式記法は以下の通りである.

数式名	簡易数式	2 次元表現
分数	fr(a,b)	$\frac{a}{b}$
積	a*b	$ab$
平方根	sq(a)	$\sqrt{a}$
べき乗根	sq(n,a)	$\sqrt[n]{a}$
指数	a^x	$a^x$
対数関数 1	log(x)	$\log x$
対数関数 2	log(a,x)	$\log_a x$
三角関数 1	sin(x)	$\sin x$
三角関数 2	sin(n,x)	$\sin^n x$
円周率	pi	$\pi$

これに微積分に関する数式を順次追加して, 数式送受のための HTML (KeTCCindy Math Sender and Receiver 以下, KMSR という) を KeTCCindyJS で作成した.

数式は ASCII 文字 (半角文字) のみ, 改行は //, 分数を / で表すときは括弧に注意  $a+b/c+d = a + \frac{b}{c} + d$

$\pi$	$\frac{b}{a}$	$a^x$	$\sqrt{a}$	$\sqrt[n]{a}$	$\log_a x$	$\sin x$	$\sin^2 x$	$\alpha$	$\beta$	$\theta$	·	space	$\leq$	$\geq$	$\Leftrightarrow$	代	$<$	$>$	$\wedge$	$+$	$=$
pi	fr(b,a)	a^x	sq(a)	sq(b,a)	log(a,x)	sin(x)	sin(2,x)	Ga	Gb	Gt	dot	sp	(=)	(=)	(=)	用	(.	.)	**	--	::

$\lim_{x \rightarrow a} f(x)$	$\int f(x) dx$	$\int_a^b f(x) dx$	$[x^2]_a^b$
lim(x,a,f(x))	int(f(x),dx)	int(a,b,f(x),dx)	[x^2]_a^b





学生は、まず解答の簡易数式を入力窓に入力して、瞬時に表示される2次元数式を確認した上で、入力窓からコピー&ペーストして提出する。教員は、送られた簡易数式について、必要であれば入力窓にペーストして確認してから、添削コメントを簡易数式で作成して返送すればよい。例えば、次のやり取りをKMSRで確認しながら送受することになる。

教員 → 学生  $15x^2 - 13x + 2 = 0$  を解け。

15x^2-13x+2=0を解け

$$15x^2 - 13x + 2 = 0 \text{を解け}$$

学生 → 教員  $15x^2 - 13x + 2 = (3x - 1)(5x - 2) = 0$  より  $x = \text{fr}(1, 3), \text{fr}(2, 5)$

15x^2-13x+2=(3x-1)(5x-2)=0よりx=fr(1,3),fr(2,5)

$$15x^2 - 13x + 2 = (3x - 1)(5x - 2) = 0 \text{より } x = \frac{1}{3}, \frac{2}{5}$$

教員 → 学生  $15x^2 - 13x + 2 = (3x - 2)(5x - 1)$  が正しい。  $x = \text{fr}(2, 3), \text{fr}(1, 5)$

15x^2-13x+2=(3x-2)(5x-1)が正しい。 x=fr(2,3),fr(1,5)

$$15x^2 - 13x + 2 = (3x - 2)(5x - 1) \text{が正しい。 } x = \frac{2}{3}, \frac{1}{5}$$

このように、ASCII以外の文字が含まれていても正しく変換される。実際に授業で使ってみて、簡易数式方式についての学生たちの反応はよく、前期授業終了後のアンケートでは、「おもしろかった。新しかった」、「とても面白いシステムだと思った」、「打ち込んで数式が完成した時の達成感があった」など肯定的な回答が多くあった。しかし、スマホでの入力はやりにくいとの声が多く、以下のような問題点があった。

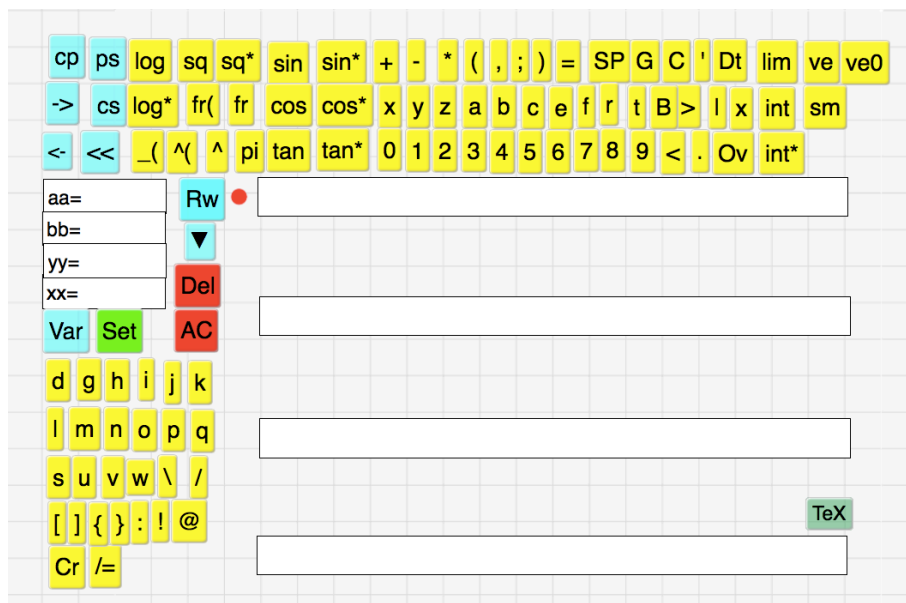
- (1) 数式入力ではASCII文字だけでよいのだが、全角文字との違いを理解していない。
- (2) キーボードアプリでは $+$ ,  $^$ などの記号の配置が機種によって違いがある。GboardはiOS, Androidのいずれでも使えるが、3種のキーボードの切り替えが必要となる。



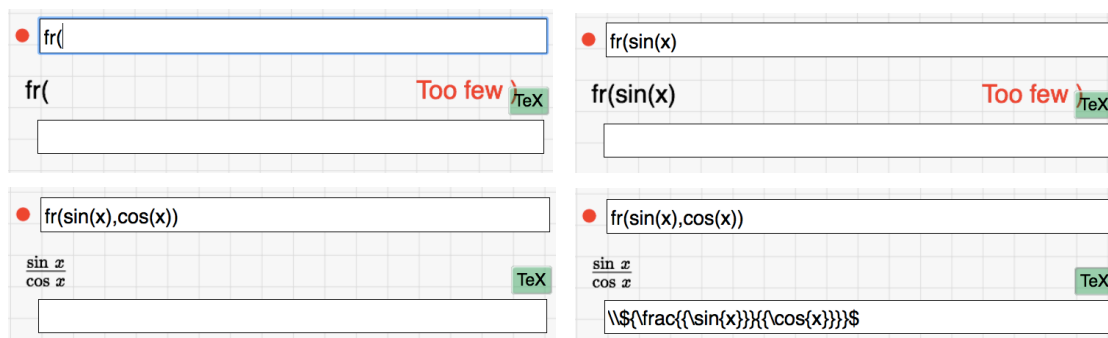
$+$ ,  $=$ ,  $^$ などの入力はどうしてもキーボードの切り替えが必要

- (3) スマホによっては、3バイト文字の数式記号のキーボードがあって、それを用いて $\sqrt{2}$ ,  $\int$ などと入力してしまう。
- (4) iOSでは、 $y'$ は $y'$ に、 $--$ は $—$ などに変換されるため、数式入力には不都合である。

そこで、KMSR の画面に数式入力に必要なボタンを配置して、ボタンを押すだけで数式が入力できるようにした。



例えば、**fr** ボタンを押すと、**fr(aa,bb)** が入力される。ここで aa, bb は引数で、左側の **Var** ボタンによって値を定め、**Set** ボタンを押すことで値が代入されて、分数が表示される。代わりに、通常のキーボードのように関数や文字ボタンを順次押して数式を作ることでもある。また、下段の **TeX** ボタンを押すと、その  $\text{\TeX}$  書式が入力窓に現れるから、 $\text{\TeX}$  文書にコピー&ペーストすることもできる。



前期授業終了時のアンケートで簡易記法の定着度を調べた。以下はその問題、正解、正解率である。

(1) $\frac{3}{8}$	fr(3,8)	94%	(2) $\frac{a+b}{c+d}$	fr(a+b,c+d)	100%
(3) $ax^2+bx+c$	ax^2+bx+c	94%	(4) $2^{10}$	2^(10)	72%
(5) $\sqrt{5}$	sq(5)	94%	(6) $\sqrt[3]{7}$	sq(3,7)	94%
(7) $\sin \pi$	sin(pi)	83%	(8) $\sin(x + \frac{\pi}{2})$	sin((x+fr(pi,2)))	72%
(9) $\sin^2 x + \cos^2 x$	sin(2,x)+cos(2,x)	67%	(10) $3 \log x$	3log(x)	83%
(11) $\log_2(x^2+1)$	log(2,(x^2+1))	72%	(12) $\int_0^1 x^2 dx$	int(0,1,x^2,x)	83%

上記データについて、多少の考察を加える。分数について、当初は / を使う学生が多



かったが、`fr`が定着したと考えられる。ただし、(8)のように簡単な分数の場合は `/` が使われることもある。べき乗(4)では、 $2^{10}$  と書いたミスで、 $\text{T}_{\text{E}}\text{X}$  でも間違えることが多い。べき乗根の定着度はよいが、底をもつ対数  $\log(a,b)$  については間違えた学生もいる。三角関数(7)や対数関数(10)では、`sinpi`, `logx` と書いてしまう学生がいた。この場合、KMSRの現バージョンでは  $\sin\pi$ ,  $\log x$  と表示される。また、(8), (11)にあるように、 $\sin(x+\pi)$ ,  $\log(x^2+1)$  と答えて、 $\sin x + \pi$ ,  $\log x^2 + 1$  と表示されるミスもあった。さらに、 $\sin^2 x$  は  $\sin(2,x)$  と書くことになっているが、 $\sin^2 x$  として、 $\sin^2 x$  と表示されるミスもしばしば起きている。これらへの対応については、次節で再考する。

## 4 今後の課題

当初に定めた数式記述ルールは、既に  $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{Cindy}$  の関数 `Totexform` に基づいている。実は、同時に `CindyScript` の書式を返す `Tocindyform` も定義されていて、例えば、`fr(a,b)`, `sq(n,x)`, `log(a,x)`, `sin(x)`, `sin(2,x)` では、それぞれ

`Totexform`

`\{\frac{a}{b}\}, \{\sqrt[n]{x}\}, \{\log_{a} x\}, \{\sin{x}\}, \{\sin^2{x}\}`

`Tocindyform`

`(a)/(b), (x)^(1/(n)), log(x)/log(a), sin(x), (sin(x))^(2)`

の文字列が出力されて、KMSRでは `Totexform` の文字列を  $\text{K}_{\text{a}}\text{T}_{\text{E}}\text{X}$  を用いて表示している。一方、`Tocindyform` の文字列は、`CindyScript` の関数 `parse` によって数値計算することができる。さらに、この数式表現は *Maxima* と近いので、その数式に変換するコマンドは容易に作成できる。これは、*Maxima* による数式計算により、解答の自動採点を可能にするものである。

しかし、KMSRに対する学生の反応を考えると修正を要する点もある。

- (1) 文字変数は1文字に限定して、積  $a * b$  は  $ab$  としてもよいようにする。
- (2) 簡単な分数の場合、 $a/b$  を許容する。具体的には数字が連続する  $120/100$ , 1文字だけの  $a/b$ , 括弧で区切られる  $(a+b)/(c)$ , 円周率を含むときは、`/` を認める。ただし、`fr` の定着度を考えると、`fr` だけでいいとも考えられる。
- (3) べき乗では、 $2^{100}$  でもよいとする。
- (4) `sin(x)`, `sin(x+y)` で、通常の  $\sin x$ ,  $\sin(x+y)$  と表示されるようにする。
- (5) `sin(2,x)` は、`CindyScript` での計算を想定しているが、数式表示としては  $\sin^2(x)$  の方がよい。

これらについては、`Totexform` と合わせて、`Tocindyform` も修正する必要がある。

現バージョンの KMSR では、間違いを最小限に抑えるために、`sin(aa)(bb)` のように、引数に値を代入する方式を採用しているが、そのため入力の手間が増していると考えられる。むしろ、`sin` で `\sin` が出力するようにした方が入力しやすく、またボタンの数を減らすこともできる。今後はその方向で開発を進めるつもりである。

キーボードについては、PCでは入力が容易である。著者の一人(濱口)の学校では、来年度の新入生から *BYOD*(Bring Your Own Device)を導入する計画である。そうなれ

ば、KMSR の授業での使用をより促すことが期待される。また、比較的安価に購入できる Bluetooth 接続のスマホキーボードがいくつか販売されているので、それらの利用で入力が容易になるかどうかについても今後検証して行きたいと考えている。今回は、KMSR の開発に合わせて試行するプロトタイプのものであったため、課題の採点には踏み込まなかったが、Maxima など自動採点するとしたとき、正解の許容範囲をどのように定めるかはシステム開発の大きな課題である。

**謝辞** 本研究は JSPS 科研費 18K02948, 18K02872, 19K21758 の助成を受けている。

## 参考文献

- [1] 大島利雄, Basel 問題の収束に関する一考察, 京都大学数理解析研究所講究録 (本講究録), 2021
- [2] 細谷大輔, 岡田裕紀, 鈴木雄大, Basel problem visualized by GeoGebra, 城西大学数学科数学教育紀要, 2020(掲載予定)
- [3] Gagern M., Kortenkamp U., Gebart J., Strobel M., CindyJS– Mathematical Visualization on Modern Devices–, ICMS 2016, LNCS **9725**, 319–334, Springer, 2016.
- [4] 高遠節夫, KeTCindyJS の開発と教育利用, 京都大学数理解析研究所講究録 2142, 2019