

# KeTCindyJSの開発と教育利用

東邦大学・理学部 高遠 節夫

Setsuo Takato, Faculty of Sciences, Toho University

## 1 序—KeTCindyの進展

KeTCindyは、動的幾何ソフトの1つであるCinderella2(以下、Cindyと呼ぶ)をGUIとして用いることによって、TeX文書に挿入する図のためのグラフィックコードを生成するツールであり([1],[3],[4],[5], Tpic specialなどのグラフィックコードセット(以下、GCSと呼ぶ)のいわばプリプロセッサとして働き、作図を容易ならしめるものである。Cindyは他の動的幾何にはない汎用的なプログラミング言語CindyScriptを持っていて、KeTCindyの実体はCindyScriptのマクロパッケージである。まず、統計ソフトRのスクリプトを書き出し、次にJavaによってRを実行することで描画コードを生成する。ごく簡単な例として、2点を結ぶ線分を描くにはCindyScriptのエディタに次のように記述すればよい。

`Listplot("1",[A,B]);` //点A,BはCindy画面上に動的にとっておくと、GCSがTpicの場合、KeTCindyは次の描画コードを生成する。

```
{\unitlength=1cm\begin{picture}(10,10)(-5,-5)%  
\special{pn 8}%  
\special{pa -1336 792}\special{pa 787 787}%  
\special{pa 532 -1324}\special{pa -1336 792}%  
\special{fp}%  
\end{picture}}%
```

KeTCindyには、簡単な幾何図形に限らず

関数のグラフ(媒介変数, 陰関数を含む)

微分方程式の解曲線(ルンゲクッタ)

自由曲線(ベジェ曲線)

作表

空間図形(直線, 平面, 多面体)

などを描く多くのコマンドが実装されている, また, CindyScriptでは, 幾何要素のみならず文字列やリストの処理も可能である. このことを利用して

曲線の交点

数値積分(大島スプライン曲線を利用)

曲線の長さや面積

色コード変換(RGB, CMYK)

などのコマンドも追加した。さらに、以下の機能拡張もなされている。

- (1) 数式処理ソフトの Maxima や R を K<sub>E</sub>T<sub>C</sub>indy から呼び出して、文字列として結果を返し、K<sub>E</sub>T<sub>C</sub>indy で利用することができる。例えば

```
Mxfun("1","diff",["sin(x)","x"]);
```

を実行すると、Maxima を呼び出して、結果の文字列"cos(x)"を返してくれる。

- (2) gcc(GNU C Compiler) を呼び出すことができる。これにより、曲面描画の隠線処理をはるかに高速化することができる。

- (3) T<sub>E</sub>X によるスライド作成機能を追加した。Beamer の簡易版とも言えるが、K<sub>E</sub>T<sub>C</sub>indy と連携することで、スクリプトの記述を簡単化するとともに、animate.sty によるアニメーションのみならず、授業ではより有効な場面が多いパラパラ動画を埋め込むこともできる。

これらに関するコマンドを合わせると、K<sub>E</sub>T<sub>C</sub>indy のライブラリは総計 500 近いコマンド群から構成されている。

2018 年には、TeXLive 開発メンバーの Norbert Preining 氏、山本宗宏氏の協力により、K<sub>E</sub>T<sub>C</sub>indy が CTAN(Comprehensive T<sub>E</sub>X Archive Network)

<https://www.ctan.org>

にアップロードされた。CTAN に行って ketcindy で検索すれば、ketcindy のパッケージを無料でダウンロードできる。なお、TeXLive2019 からは、K<sub>E</sub>T<sub>C</sub>indy パッケージが最初から組み込まれている。

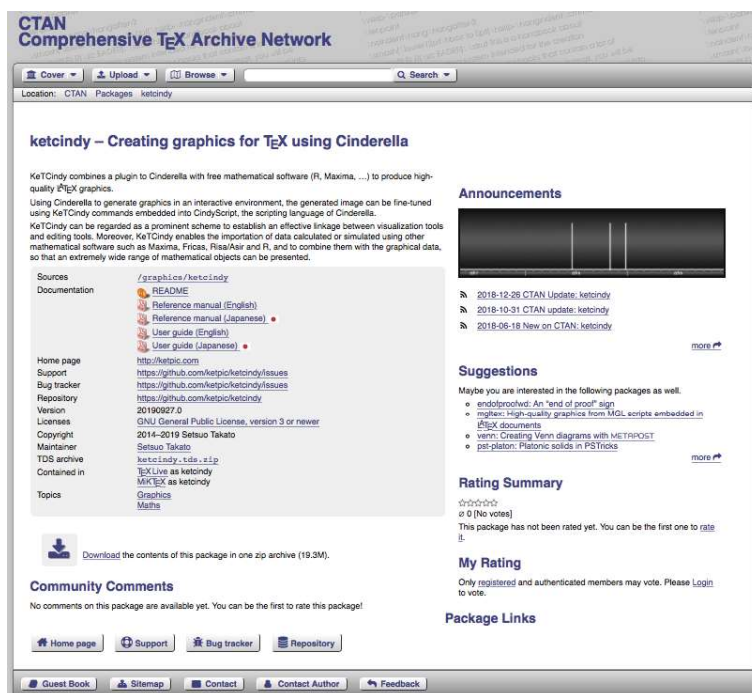


図 1. CTAN の ketcindy ページ

注) Download ボタンを押せば、安定版が CTAN からダウンロードできる、バグ修正などが随時なされた最新版を得るには、Repository ボタンから github のページに行けばよい。

2019年4月には,  $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{Cindy}$  のサンプルページ

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

を立ち上げた. 最近では, ブラウザの検索窓に「samples of ketcindy」あるいは「samples ketcindy」などを入力すればヒットするようになっている. このページには, 多くの教材サンプルとともに, コマンドの reference もアップされている.

また, 2019年3月には, Cinderella Japan 代表の入谷昭氏が中心となって, iBook(Apple book)「 $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{Cindy}$  インタラクティブに美図版作成」(無料)が出版された.

GCSとして従来利用してきた Tpic special は, 海外で主流となった pdfLaTeX ではサポートされていない. そこで, pict2e と tikz も利用できるようにした. それには, CindyScript に

`Usegraphics("pict2e");` または `Usegraphics("tikz");`  
を追加すればよい. すると, 先の例 Listplot でそれぞれ次のコードが出力される.

```
{\unitlength=1cm\begin{picture}(10,10)(-5,-5)%  
\linethickness{0.008in}%%  
\polyline(-3.39244,-2.01145)(2.00000,-2.00000)%  
(1.35097,3.36242)(-3.39244,-2.01145)%  
\end{picture}}%
```

または

```
{\unitlength=1cm\begin{tikzpicture}[x= 1cm ,y= 1cm ]%  
\clip (-5,-5) rectangle (5.5,5.5);%  
\draw [line width=0.6](-3.39244,-2.01145)--(2.00000,-2.00000)%  
--(1.35097,3.36242)--(-3.39244,-2.01145);%  
\end{tikzpicture}}%
```

## 2 $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{CindyJS}$ の開発

CindyJS は, Cindy での操作を Web 上で実現する HTML ファイルを作成するもので, ミュンヘン工科大学の CindyJS.org

<https://cindyjs.org>

のメンバーにより, 2013年頃から開発が進められており ([2]), 完全とは言えないにしろ, Cindy と相当の互換性を持っている. しかし, Cindy 自体がそうであるように, 幾何以外の幅広い学習分野の教材を作成するには不十分である. 一方,  $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{Cindy}$  は,  $\text{T}_{\text{E}}\text{X}$  での印刷教材や授業用のプレゼン教材の作成には力を発揮するが, Web での利用を考えた場合はでき上がった PDF を配布するしかなく, インタラクティブ性には劣っている. そこで, CindyJS で  $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{Cindy}$  のライブラリを利用することを試みた. これが  $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{CindyJS}$  である.

前述したように,  $\text{K}_{\text{E}}\text{T}_{\text{C}}\text{Cindy}$  のライブラリは多くのコマンドを持ち, 行の総数は 30000 にも上るので, このまま Web で配布または HTML に直接追加するのは現実的でない. そこで, 次の工夫をした.

- (1) KeTCindy のライブラリからコマンドについての情報をリストアップしたファイルを作成するプログラムを作成した。たとえば、その情報は以下のように記録される。

Listplot,basic1,3995,4076,Divoptions,Strsplit,Toupper,...

ここで、basic1, 3995, 4076 はこの関数 Listplot がライブラリファイル basic1 の 3995 行から 4076 行までに記載され、Divoptions などの関数とその定義内で使われていることを意味する。

- (2) Cindy のメニューから「CindyJS に書き出す」を選んで HTML ファイルを作成してから、KeTCindyJS の作成ボタンを押すと、KeTCindyJS はその HTML の

`<script id="csinit">` または `<script id="csdraw">`

に書かれたすべてのコマンドを抜き出し、(1) の情報に基づき、HTML にコマンドの定義部分を追加する。

以上によって、HTML 全体の分量を抑えることが可能となり、かつ、情報ファイルはライブラリの更新の度に更新されるので、常に最新の情報が用いられることになる。

さらに、次の機能を追加した。

- (3) CindyJS と KeTCindyJS のライブラリをオフラインで読み込んで実行することもできるようにした。

KeTJS ボタン            オンライン実行の HTML 作成

KeTJSoff ボタン        オフライン実行の HTML 作成

- (4) コマンド Setketcindyjs を追加して、それに記述される設定に従って、HTML 内の定義や設定を修正できるようにした。

- (5) アニメーション作成用のボタンとコマンドを追加した。

Play (再生), Pause (一時停止), Rev (逆再生), Stop (停止)

コマンド Animationparam はこれらのボタンにより定まる時間パラメータを返す。

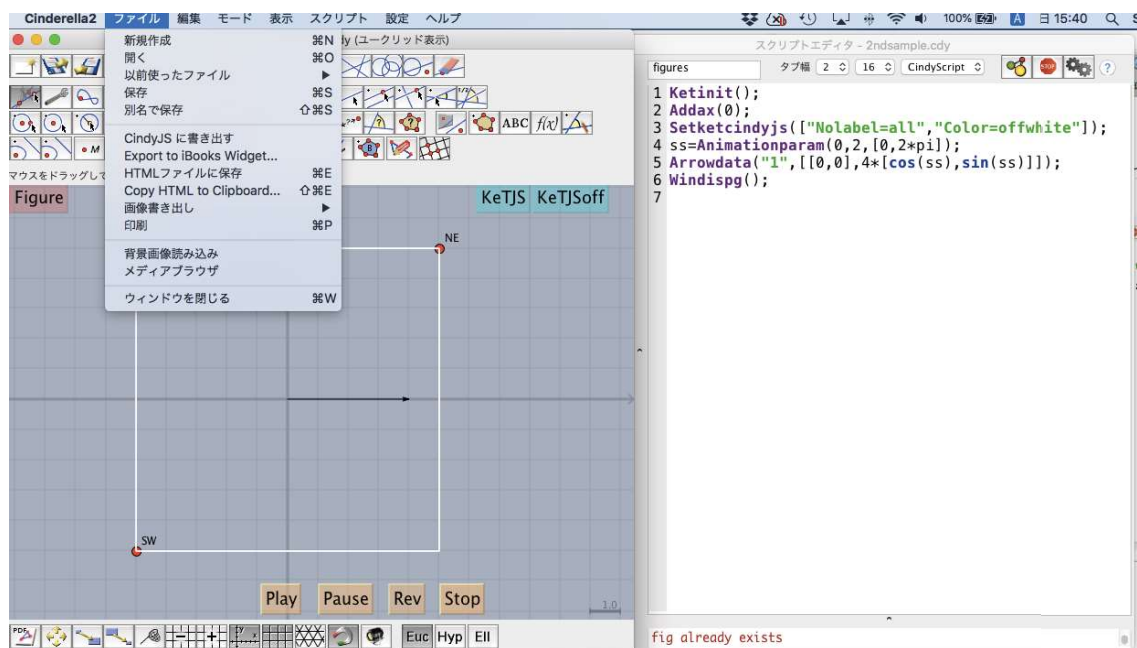


図 2. KeTCindyJS の画面と Script エディタ

- (6) コマンド `Textedit` により，画面上の入力窓から値を取得できるようにした。
- (7) コマンド `Ketcindyjsdata` により，Maxima で予め求めた一般公式を `KeTCindyJS` で使えるようにした。

前ページの図2を例として，`KeTCindyJS` による HTML 作成の流れを示す。

- (1) `KeTCindy` パッケージにあるサンプルフォルダまたは `KeTCindy` のサンプルページから `KeTCindy` 用の `template.cdy` を適当なフォルダにコピーしてダブルクリックする。
- (2) `CindyScript` エディタに以下のスクリプトを記述する。

```

Ketinit(); // KeTCindy の初期化
Addax(0); // 座標軸を描かない
Setketcindyjs(["Nolabel=all", "Color=offwhite"]);
// 点のラベルをつけない，背景色を offwhite にする
ss=Animationparam(0,2,[0,2*pi]);
// 時間パラメータの初期値，スピード，範囲
Arrowdata("1", [[0,0], 4*[cos(ss), sin(ss)]]);
// 原点から矢印を描く
Windisp(); // 画面に描画

```

注1) `TeX` を起動して pdf を作るには，`Figure` ボタンを押せばよい。

注2) `Play` ボタンを押すと，`Animationparam` に従ってアニメーションが始まる。

- (3) トップメニューから「ファイル > `CindyJS` に書き出す」を実行すると `CindyJS` の HTML ファイルができる。ただし，`Script` エディタに記述されたコマンドはすべて `KeTCindy` のものなので，幾何点以外は表示されない（図3左）

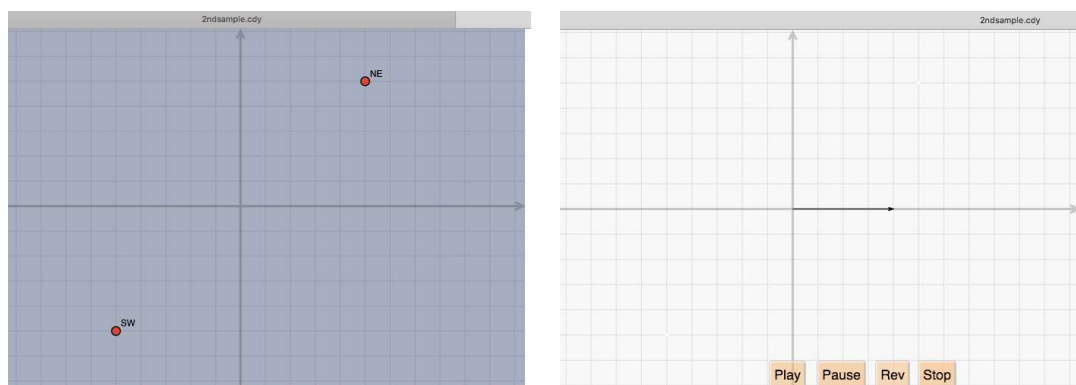


図3. `CindyJS` と `KeTCindyJS` による HTML

- (4) `KeTJS` ボタンまたは `KeTJSoff` ボタンを押すと，`KeTCindyJS` による HTML が作成される（図3右）。画面上の `Play` ボタンを押すとアニメーションが始まる。

次節では，`KeTCindyJS` によって作成されたいくつかの教材例を示すことにする。

### 3 教材例

本節に掲げるすべての例は、1節で紹介した K<sub>E</sub>T Cindy のサンプルページに行けば、PC (Windows, Mac, Linux) でもスマホやタブレットでも実行することができる。ただし、Windows の場合は Google Chrome か Firefox が推奨される (Internet Explorer では、場合によって設定を変更する必要がある)。参考のため、各教材例には、サンプルページ内の URL をつけておく。

#### 3.1 2次曲線

<https://s-takato.github.io/ketcindysample/s02graphs/offline/s0205coniccurvejssoffl.html>

3点を与えて2次曲線を描くもので、CindyScript の放物線に関する部分を示すと次のようになる。

```
pA=Ptpos(A); pB=Ptpos(B); pC=Ptpos(C);
if(chno==1,
  Parabolaplot("1", [pA,pB,pC], "[-5,10]");
  Lineplot("1", [pB,pC], ["da"]);
  Letter([A,"ne","F",B,"ne","A",C,"ne","B"]);
);
```

chno=1,2,3 は画面上のボタンから設定され、それぞれ放物線、楕円、双曲線の描画に対応している。A, B, C は幾何点 (画面上で自由に動く点) であるが、CindyJS のままでは、遠い位置からでも点が動いてしまうので、K<sub>E</sub>T CindyJS には制御範囲を限定するいくつかのコマンドを追加した。1行目にある Ptpos もその1つである。

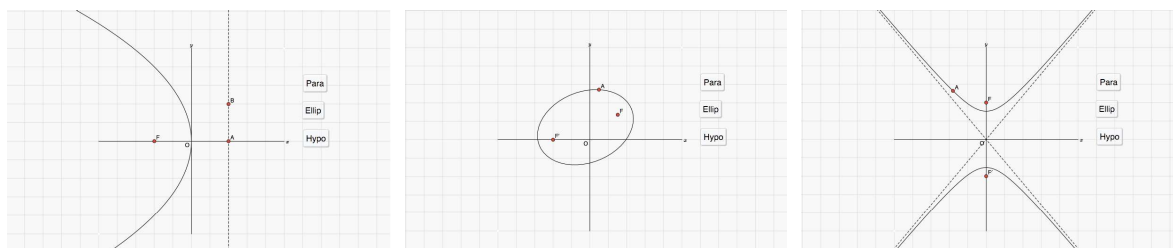


図 4. 2次曲線の描画

#### 3.2 一般角

<https://s-takato.github.io/ketcindysample/s06animation/offline/s0611generalanglejssoffl.html>

高校数学の復習として、一般角を理解してもらうために作ったアニメーション教材で、Play ボタンを押すと、動径が回転して一般角を螺旋で表示するようになっている。CindyScript の主要部分は以下の通りで、3行目の Animationparam によって、時間パラメータを取得する。すなわち、初期値は0で、1秒に60°進み、10000になったら停止するように設定している。

```

p0=[0,0]; r=2.5;
Circledata("1",[p0,r],["dr,1.5","Num=100"]);
an=Animationparam(0,60,[-10000,10000]);
th=an*pi/180;
p1=r*[cos(th),sin(th)];
Listplot("1",[p0,p1],["dr,1.5","Color=red"]);
fun="(0.5+0.1*abs(t)/(2*pi))*[cos(t),sin(t)]";
rng=Assign("t=[0,th]","th",th);
if(abs(th)>0,
  Paramplot("1",fun,rng);
  Arrowhead("1",Ptend(gp1),"gp1",[0.5]);
);
Expr([-r,r],"e","\theta="+sprintf(an,2)+"^\{\circ\}","Size=1.5");

```

著者としては、2周くらいの回転を見せるだけの意図（左図）であったが、学生たちは回転することを続けて「変だ」と言い始めた（中図）。しかし、さらに回転すると、右図のような模様が現れて、彼らは興味深く変化する模様を眺めていた。

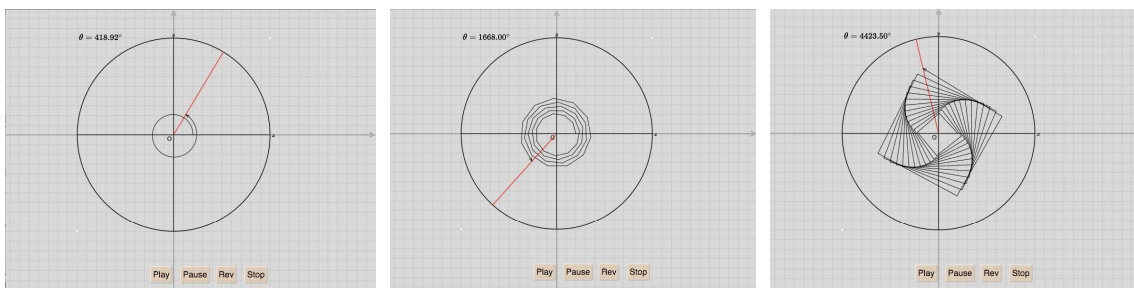


図5. 一般角

原因は、螺旋の分割数をデフォルト値50に固定していたという一種のプログラムミスであるが、学生たちに興味を持たせたことは、一般角の理解のために無意味ではなかったと考えている。

### 3.3 ネイピア数

<https://s-takato.github.io/ketcindysample/s16ketjsmisc/offline/s1612napierjssoffL.html>

$y = a^x$  の  $x = 0$  での接線の傾きが1となる  $a$  として説明するための教材である。

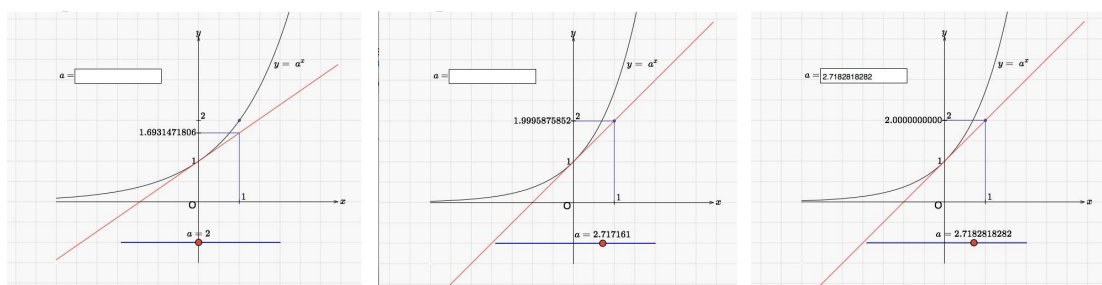


図6. ネイピア数



図6の左図は初期画面で  $a = 2$  としている。学生には、スライダ上の  $a$  を動かして点  $(1, 2)$  (青い点) を通るとききの  $a$  の値を求めることを指示する。中図はその結果であるが、スライダだけでは、 $a = 2.716171$  以上の精度を得ることができない。そこで、画面の左上の入力窓を使うことになる。KETCindyJSで入力窓を作るには、画面上にそのためのボタンと CindyScript に以下のスクリプトを追加するだけでよい。ここで、Texteditが 入力された文字列を取得するコマンドである。

```
//str=Textedit(60); //only ketjs
if(length(str)>0,
  aa=parse(str);
  Ptpos(A, [aa-2, -1]);
,
  aa=Ptpos(A)_1+2;
);
```

ほとんどの学生たちは、中図で求めた  $a = 2.716171$  を増減させながら  $a = 2.7182818282$  を得ていた。ただし、小数10位は CindyJS の計算精度のために誤差を生じている。

### 3.4 楕円ビリヤード

<https://s-takato.github.io/ketcindysample/s06animation/offline/s0612ellipticalbilliard3jssoffL.html>

楕円の焦点から発せられる光が、楕円壁で反射してもう1つの焦点に到達することを体験させるためのアニメーション教材である。ここでは、楕円壁までの到達時間や反射方向を求めるために、KETCindy から Maxima を呼び出している。例えば、到達時間を求めるには、Maxima のコマンド列をリストで作り、CalcbyM で Maxima を呼び出し、結果を文字列として取得する。

```
cmdL=[ //no ketjs on
  "assume(a>0,b>0)", [],
  "declare([u1,v1,x1,y1],real)", [],
  "f:x^2/a^2+y^2/b^2-1", [],
  "eq3:ev(f=0,[x=x1+u1*t,y=y1+v1*t])", [],
  "ans:solve(eq3,t)", [],
  "t1:ev(t,ans[1])", [],
  "t2:ev(t,ans[2])", [],
  "dif:ratsimp(t2-t1)", [],
  "t:if dif>=0 then t2 else t1", [],
  "x2:x1+u1*t", [],
  "y2:y1+v1*t", [],
  "t::x2::y2", []
];
CalcbyM("ans0",cmdL,[""]);
```



ただし、K<sub>E</sub>T CindyJS では Maxima の呼び出しができないので、この結果を次のコマンドで HTML に書き出すようになっている。

```
Ketcindyjsdata(["ans0",ans0,"ans",ans]);
```

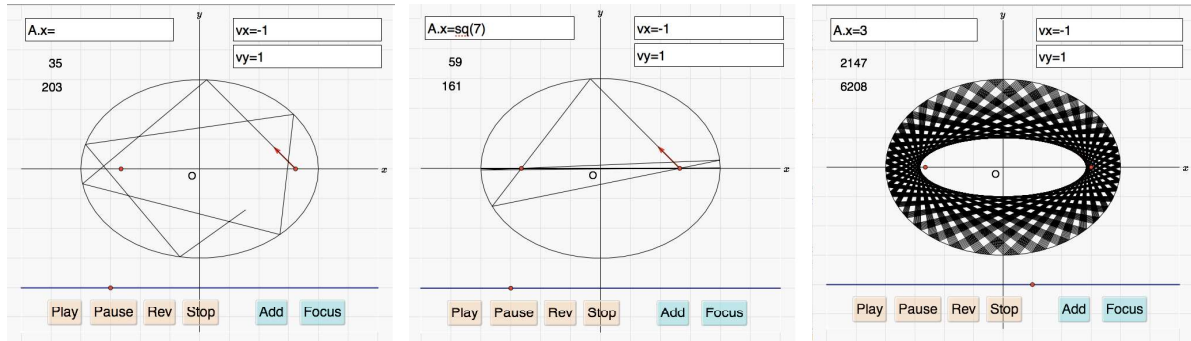


図 7. 楕円ビリヤード

始点が焦点でないときは焦点を通らない (左図)。また、焦点のときはもう 1 つの焦点に到達する (中図)。しかし、この場合は焦点を通過した後もとの焦点に戻るが、軌道は急速に  $x$  軸に接近する。一方、焦点を通らない場合 (右図) の反射を繰り返し行くと、軌道は同一焦点の楕円 (または双曲線) を包絡線としている。学生たちは、図形ができるのを興味深く観察していた。さらに時間の余裕があれば、これがより高度な数学 (力学系など) につながることに言及してもよい。

### 3.5 最速降下曲線

<https://s-takato.github.io/ketcindysample/s16ketjsmisc/offline/s1611brachistchrone2jsoffl.html>

最速降下曲線の数学的導出の前に、その動機づけとして作成した教材である。ここでも Maxima で計算した結果や微分方程式の数値解法が用いられている。図 7 の左図が初期画面で、 $(0, 0)$  から  $(5, -5)$  まで直線に沿って降下したときの時間 (単位 mm 秒) が図の右側に表示されている。曲線は 2 つの赤い点を制御点とするベジエ曲線で描かれていて、これらの点を動かすことで形状を変えることができる。学生への課題は「降下時間ができるだけ小さくなるように曲線を変形せよ」である (中図)。さらに、Play ボタンを押すと、降下する様子がアニメーションで示される (右図)。

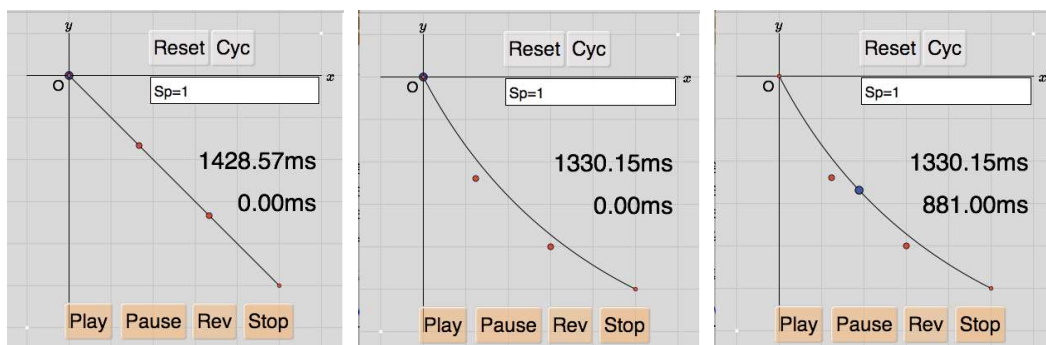


図 8. 最速降下曲線 (1)

効果時間をある程度小さくできた段階で、右上の Cyc ボタンを押すとサイクロイド曲線が表示され、求めた曲線が十分近いことを体感できる。

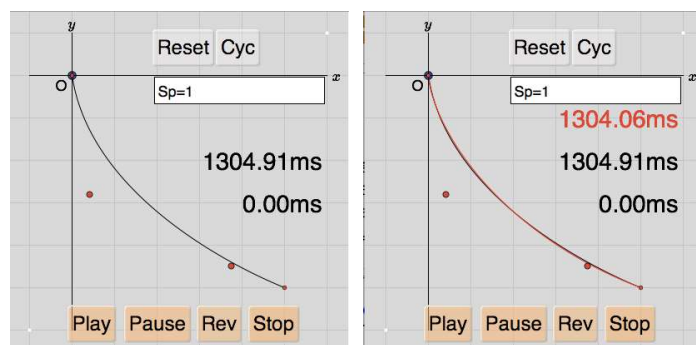


図 9. 最速降下曲線 (2)

## 4 まとめと課題

KeTCindyJS は種々のインタラクティブな教材を作成できる可能性を持ち、結果として、教員と学生のコミュニケーションを推進することが期待される。ここで、最も重要なのは教員の経験知であり、学生をつまづきを察し、どのような教材によってそれを解消できるか、学生の主体性を引き出すことができるかを見抜く力であろう。

現在のところ、KeTCindyJS から数式処理システムや C を直接呼び出すことはできないが、KeTCindyJS のファイルをモジュール化することで可能であると考えている。

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

## 参考文献

- [1] Takato S., What is and how to Use KeTCindy – Linkage Between Dynamic Geometry Software and Collaborative Use of KeTCindy and Free Computer Algebra Systems and  $\LaTeX$  Graphics Capabilities –, Mathematical Software –ICMS 2016, LNCS **9725**, 371–379, Springer, 2016.
- [2] Gagern M., Kortenkamp U., Gebart J., Strobel M., CindyJS– Mathematical Visualization on Modern Devices, –ICMS 2016, LNCS **9725**, 319–334, Springer, 2016.
- [3] Takato S, McAndrew, Vallejo J, Kaneko M., Collaborative use of KeTCindy and free Computer Algebra Systems, Mathematics in Computer Science 11 3-4 , 503-514, 2017.
- [4] 野田健夫, 高遠節夫, KeTCindy の C 呼び出し機能と曲線・曲面論の教材の作成, 京都大学数理解析研究所講究録 2067, pp132–141, 2018
- [5] 高遠節夫, 大島スプライン曲線の応用と KeTCindy プログラミング, 京都大学数理解析研究所講究録 2105, pp126–135, 2019