

# KeTCindy による Free Wolfram Engine(WE) の 呼び出し機能と教材作成

東邦大学理学部 高遠節夫, KeTCindy 開発センター 北原清志

## 1 はじめに

高校・高専・大学（以下カレッジ級という）の数学の授業において、多くの教員は印刷教材を配付するが、 $\text{T}_\text{E}\text{X}$  を用いて教材を作成する場合、本文と違和感のない図を作成して、適切な位置に配置するのに苦労することが多かった。著者のグループは、2006年頃から、Maple, Mathematica, Scilab などの数学ソフトウェアのライブラリを作り、 $\text{T}_\text{E}\text{X}$  の描画システム  $\text{T}_\text{p}\text{i}\text{c}$  のコードを生成するツール  $\text{K}_\text{E}\text{T}\text{p}\text{i}\text{c}$  を開発してきた [1][2][3]。また、図などを自由な位置に配置するために、 $\text{ketpic.sty}$ ,  $\text{ketlayer.sty}$  も作成した。2014年には、動的幾何ソフトの1つである Cinderella2 を  $\text{K}_\text{E}\text{T}\text{p}\text{i}\text{c}$  の GUI として用いることで、インタラクティブな図の作成を可能にする  $\text{K}_\text{E}\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$  へと発展させた [4][5][7]。Cinderella2 は、CindyScript という使いやすいプログラム言語を内蔵していて、 $\text{K}_\text{E}\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$  の主要部分は、CindyScript の総数 400 を超えるライブラリ群である。また、描画コードを生成する数学ソフトとしては、 $\text{R}$  を用い、描画コードは  $\text{T}_\text{p}\text{i}\text{c}$  の他に  $\text{p}\text{i}\text{c}\text{t}2\text{e}$  および  $\text{T}\text{i}\text{k}\text{Z}$  をサポートした。さらに、Cinderella2 には組み込まれていないファイル操作や外部プログラム実行のために開発した  $\text{KetCindyPlugin.jar}$  を Cinderella2 の  $\text{plugins}$  フォルダに入れて動かすようになっている。次の図は、Cinderella2 のスクリーンと CindyScript の画面である。

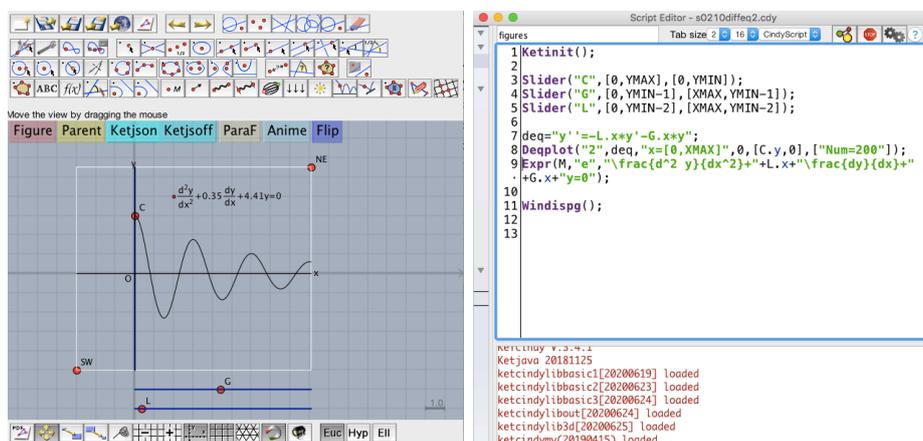


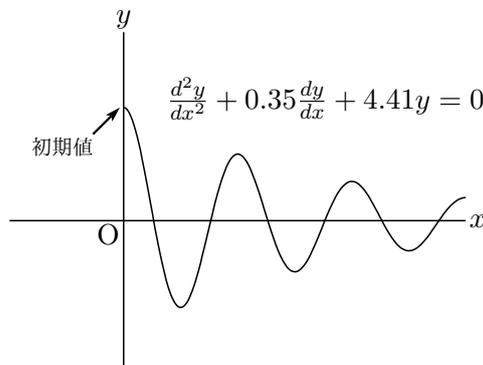
図1 Cinderella2 の画面

左図にある赤い点は Cinderella2 の幾何要素で、カーソルを用いて自由に動かすことができる。このうち、SW と NE は  $\text{T}_\text{E}\text{X}$  の描画領域を決定している。また C, G, L はスライダー上の点で、これらを動かすことにより、微分方程式の係数や初期値を変えることができる。左上にある **Figure** ボタンを押すことで、まず  $\text{R}$  が描画コードファイル（例えば  $\text{sample1.tex}$ ）を生成し、次に、Cinderella2 によって作られていた親ファイルを  $\text{T}_\text{E}\text{X}$  がコンパイルして、pdf ビューアが確認のための pdf を表示する。ここで、 $\text{sample1.tex}$  は  $\text{T}_\text{E}\text{X}$

のファイルであるため、読み込みには`\includegraphics`ではなく`\input`が用いられる。また、`TeX` ドキュメントの中で図を適切な位置に配置するには、`KpTpic` のスタイルファイルで定義された環境 `layer` を用いればよい。

```
\usepackage{keticpic,ketlayer}

\begin{layer}{120}{0}
\putnotese{82}{5}{\input{sample1}}
\arrowlineseg{96.5}{15}{5}{-135}
\putnotesw{93}{19}{\scriptsize 初期値}
\end{layer}
```



(本文)

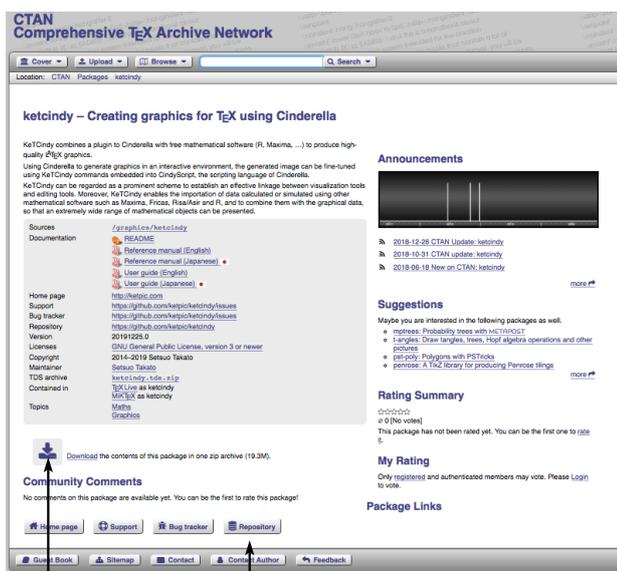
図2 layer 環境による図の配置

ここで、数値はすべて mm 単位で、`{120}{0}` は横方向 120mm、縦方向 0mm の範囲にグリッドを描くことを意味する、ただし、縦方向を 0 にしたときは、グリッドを表示しない。また、`\putnotese{82}{5}` により、(82, 5) を基点に南東方向 (se) に図を配置している。他の要素の位置を変えずに図や記号などを自由に配置する `layer` 環境は、教材作成にとって非常に有用であり、不可欠なものといえる。

2018 年、`KpTCindy` は CTAN (Comprehensive `TeX` Archive Network) に登録されて、スタイルファイルやライブラリー一式が CTAN からダウンロードできるようになった。同時に、`TeXLive2018` 以降には、`ketcindy` が最初から組み込まれている。また、2020 年 4 月に `Cinderella3` ( $\beta$  版) がリリースされた。

<https://beta.cinderella.de>  
`Cinderella3` の `plugins` フォルダには最初から `KetCindyPlugin.jar` が組み込まれたため、`KpTCindy` を簡単にインストールできるようになった。

- 1) CTAN から `KpTCindy` を取得。  
 CTAN `ketcindy` で検索すると `Package ketcindy - CTAN` がヒットする。  
 Download (正規登録版) または Repository (最新版) を押す。  
 注) Repository では、さらに Clone>Download Zip を押す。



Download ボタン    Repository ボタン

図3 CTAN/ketcindy のページ

- 2) R, Maxima と Sumatra(Windows のみ) をインストールする.
- 3) K<sub>E</sub>T<sub>C</sub>indy のパッケージにある `ketcindysettings.cdy` をダブルクリック, 以下の [1][2] を選択して, [3] の `Mkinit,Update,Work` を順に実行する.

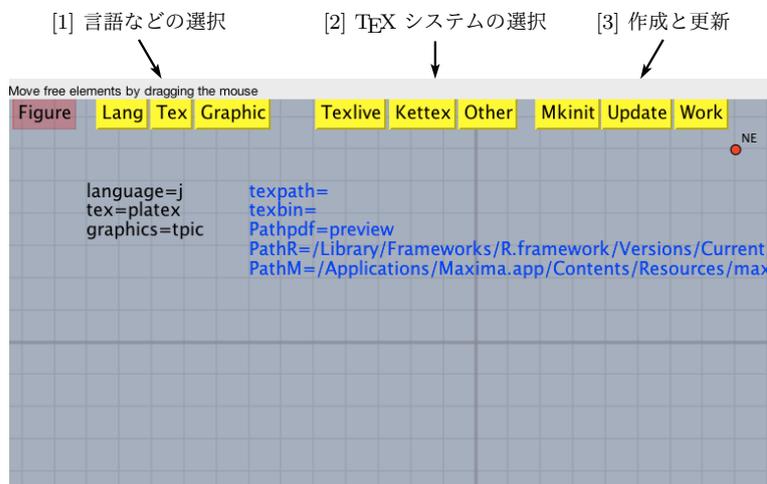


図 4 `ketcindysettings.cdy` の画面

以下, 2 節では K<sub>E</sub>T<sub>C</sub>indy の拡張機能として, スライド作成 (KeTSlide) と HTML ファイル作成 (KeTCindyJS) について述べる. また, 数式処理システム (以下, CAS という) は, 教材作成において時に有用であり, K<sub>E</sub>T<sub>C</sub>indy ではフリーの Maxima, Risa/Asir, Fricas を呼び出して計算結果を利用できる機能を既に組み込んでいる [8][9]. さらに, アプリ開発者などが無償で利用できるようになった Mathematica のコア Wolfram Engine (以下 WE という) を K<sub>E</sub>T<sub>C</sub>indy から呼び出せるようにした. 3 節では WE のインストールと K<sub>E</sub>T<sub>C</sub>indy からの利用, および 4 節では WE を用いた教材作成例を紹介する.

## 2 K<sub>E</sub>T<sub>C</sub>indy の拡張機能

図や文字を自由に配置できる layer 環境は, 単に印刷教材だけでなく, スライドの作成にも有用である. KeTSlide は, T<sub>E</sub>X を簡略化したテキストファイルから T<sub>E</sub>X のファイルを生成し, スライドを作成することができる K<sub>E</sub>T<sub>C</sub>indy の拡張機能の 1 つであり, Beamer にあるような段階的表示も可能である. 例えば, テキストファイル (slide.txt とする) と同名の cdy ファイル (slide.cdy とする) を作り, slide.txt に以下のように記述する.

```
new::初期値による違い
%repeat=2
\slidepage
layer::{120}{0}
%[1]::putnote::se{60}{25}::sample1,0.8
%[2]::putnote::se{60}{25}::sample2,0.8
end
$y'(0)=0$とする.
```

```

itemize
item::$y(0)=3$のとき
%[2,-]::item::$y(0)=2$のとき
end

```

slide.cdy の画面上にある Slide ボタンを押すと、まず K<sub>E</sub>T<sub>C</sub>indy が T<sub>E</sub>X ファイルを作り、次に T<sub>E</sub>X コンパイラが、同一ページに 2 枚のスライドがある pdf ファイルを作成する。

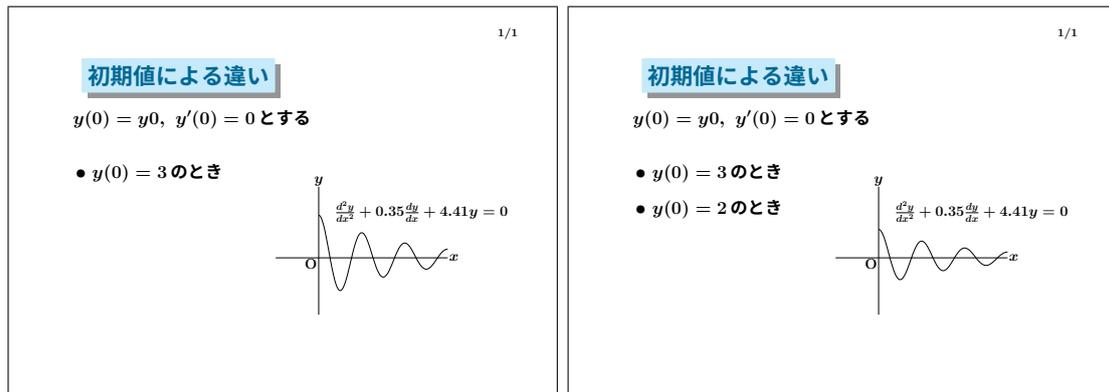


図5 スライドのページ

もう 1 つの主要な拡張機能は KeTCindyJS である。2016 年、Cinderella の開発者である J.Richter-Gebert 教授の所属するミュンヘン工科大学のグループが、Cinderella の機能をほぼ HTML として実現する CindyJS を発表した [6]。以下のページに行けば、CindyJS によって作成された豊富なサンプルを見ることができる

<https://cindyjs.org>

しかし、CindyJS だけではカレッジ級で用いられる種々の教材を作ることは難しい。それは、ちょうど Cinderella と K<sub>E</sub>T<sub>C</sub>indy との関係と同じである。そこで、CindyJS で作られる HTML に K<sub>E</sub>T<sub>C</sub>indy の関数を追加する機能を追加した [10]。ただし、すべての関数を追加すると 20000 行を超えるため、CindyScript に記述されている関数のみを追加するようにしている。そのため、K<sub>E</sub>T<sub>C</sub>indy を付加した HTML は、多くの場合 2000 行以内に収まっている。1 例として 2 次曲線を描画する HTML の作成を説明する。

```

Putpoint("A", [2,0],A.xy);
Putpoint("B", [2,2],B.xy);
Putpoint("C", [-2,0],C.xy);
if(chno==1,
  Parabolaplot("1", [C,A,B], "[-5,10]");
  Lineplot("1", [A,B], ["da"]);
);
if(chno==2,
  Ellipseplot("1", [C,A,B], "[0,2*pi]");
);
if(chno==3,
  Hyperbolaplot("1", [C,A,B], ["Asy=da"]);
);

```

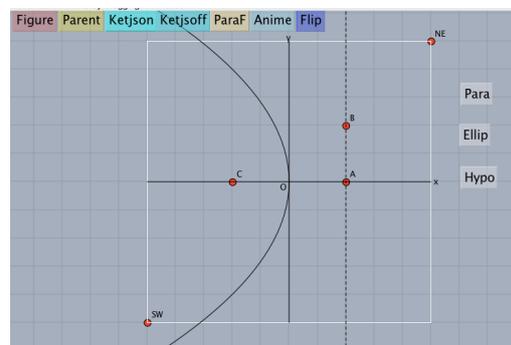


図6 KeTCindyJS の画面

画面の右側にある Para などのボタンは Cinderella で作られ、ボタンを押すことで chno の値がそれぞれ 1, 2, 3 に設定され、放物線、楕円、双曲線が表示される。トップメニューから「ファイル > HTML に書き出す」を選ぶと、CindyJS の HTML ファイルが生成され、さらに Ketjsoff (または Ketjson) のボタンを押すことで KeTCindyJS のファイルが作られる。このファイルでは、点 A,B,C を自由に動かすことができ、2次曲線の選択ボタンも有効である。

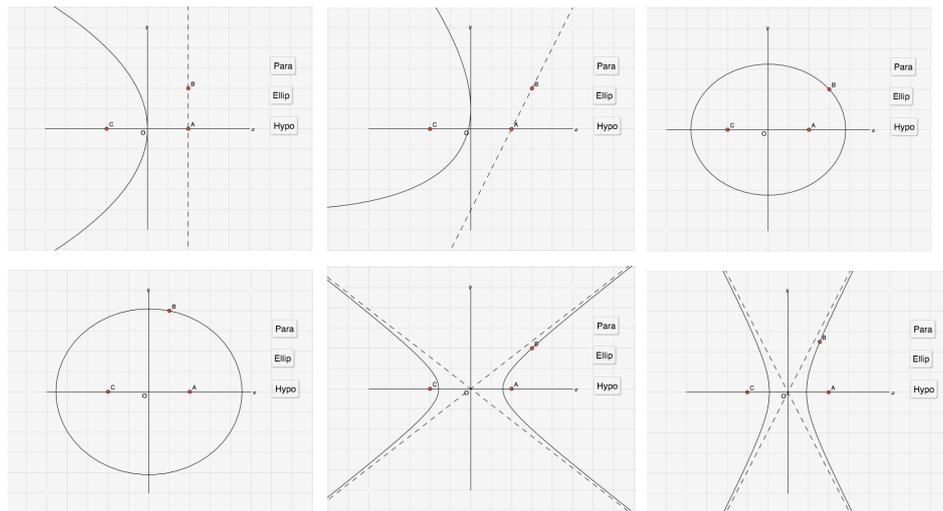


図 7 2次曲線描画の HTML

### 3 WE の KeTCindy からの呼び出し

2019年5月、Wolfram社はFree Wolfram Engine for Developers(WE)を発表し、MathematicaのコアであるWolfram Engineをローカルにダウンロードして、ソフトウェア開発の試作に使えるようにした。

<https://www.wolfram.com/engine/index.ja.php?source=footer>

ページの冒頭には、WEが使える用途として

- あなた自身の、あるいはあなたが勤めている会社の製品を開発する。
- 自宅、学校、職場で個人的なプロジェクトを実施する。
- 将来の製品プロジェクトのために、Wolfram言語を試す。

の3点が挙げられている。このページからOSを選んでクリックすればダウンロードが始まり、指示に従ってインストールすると、Wolfram ScriptとWolfram Engineがアプリケーションのフォルダに作られる。ただし、実行前にWolfram IDを作成しておく必要がある。

<https://account.wolfram.com/ja/login/create>

注) 既に Mathematica などを使っていて Wolfram ID がある場合は、それを使えばよい。ダウンロードしたファイルをアクティベートする時に ID とパスワードが必要となる。

KeTCindyにはCASの呼び出しのために以下の関数を実装している。

- (1) 1つのコマンドを実行する (Maxima では Mxfun) .
- (2) 複数のコマンド列を実行する (Maxima では CalcbyM) .
- (3) L<sup>A</sup>T<sub>E</sub>X 書式の文字列を返す (Maxima では Mxtex) .

これに対応して, WE では (1) Wlfun, (2) CalcbyW, (3) Wltex の3つであり, それぞれ次のように用いられる.

- (1) Wlfun("deq", "DSolve", ["y'[x]+y[x]==a Sin[x]", "y[x]", "x"]);  
DSolve によって微分方程式を解く. WE の計算結果は, 文字列として変数 wldeq に割り当てられる. この例では次のようになる.

```
wldeq="{y[x]->C[1]/E^x+(a*(-Cos[x]+Sin[x]))/2}"
```

- (2) まず, コマンド列を用意して, CalcbyW により WE を呼び出す.

```
cmdL=[
  "fun=x^2*Sin[x]", [],
  "df=D", ["fun", "x"],
  "intf=Integrate", ["fun", "x"],
  "df::intf", []
];
CalcbyW("ans", cmdL, [""]);
```

変数 ans は, 導関数と不定積分を表す2つの文字列からなるリストである.

```
ans=["x^2*cos[x]+2*x*sin[x]", "-((-2+x^2)*cos[x])+2*x*sin[x]"]
```

- (3) Wltex("1", wldeq);

変数 tx1 に L<sup>A</sup>T<sub>E</sub>X 書式の文字列が割り当てられる.

```
tx1="\left\{\left\{y(x)\to\frac{1}{2}a(\sin(x)-\cos(x))+c_1\dots\right\}\right\}
```

実際に L<sup>A</sup>T<sub>E</sub>X で出力すると  $\left\{\left\{y(x) \rightarrow \frac{1}{2}a(\sin(x) - \cos(x)) + c_1e^{-x}\right\}\right\}$  となる.

## 4 WE を用いた教材作成例

K<sub>E</sub>T<sub>C</sub>indy から WE を呼び出す関数の応用例を紹介する. 本教材の対象学生としては, 微分方程式の初歩を学習した学生を想定している. 微分方程式の解は初等関数で求まらない場合も多いので, その場合は数値的に解を求めることが重要である. 簡単な例として, 次の微分方程式を数値的に解くことを考えよう.

$$\frac{dx}{dt} = x, \quad x(0) = 1$$

微分方程式の解は  $x(t) = e^t$  であり数値的に解く必要はないのだが, 最初の例として取り上げる. この微分方程式の数値解を WE を呼び出すことにより計算する. プログラムは次のようになる.

```
cmdL=[
  "xf=First[x /. NDSolve[{x'[t]==x[t], x[0]==1}, x, {t, -1, 2}]]", [],
```

```

"trg=Range[-1,2,0.1]", [],
"xrg=Map[xf,trg]", [],
"trg::xrg", []
];
CalcbyW("ans",cmdL,[""]);
rg1=replace(ans_1,[[{"","["],["}",""]]]);
rg2=replace(ans_2,[[{"","["],["}",""]]]);
trg=parse(rg1);
xrg=parse(rg2);

```

cmdL は WE の文法で書かれた文字列からなるリストである。CalcbyW を実行すると文字列 cmdL が WE に渡され、実行結果は変数 ans の中に 2 種類の文字列リスト ans=[ans\_1,ans\_2] として格納されている。NDSolve が微分方程式を数値的に解く命令で、 $x'[t]==x[t]$  が微分方程式である。右辺の  $x[t]$  の部分を  $\text{Sin}[x[t]]$  のように変えればより複雑な微分方程式を解くことになる。trg は範囲  $-1 \leq t \leq 2$  を 0.1 刻みで分割した  $t$  の値のリスト、xrg は各  $t$  の値に対応する  $x$  の値のリストである。CalcbyW 以降の最初の命令 replace は、各文字列 ans\_i={...} ( $i = 1, 2$ ) を文字列 rgi=[...] に置き換える Cinderella の命令で、WE のリストを K<sub>F</sub>TCindy のリストに変換している。また、parse は文字列を数字列に変換する Cinderella の関数である。

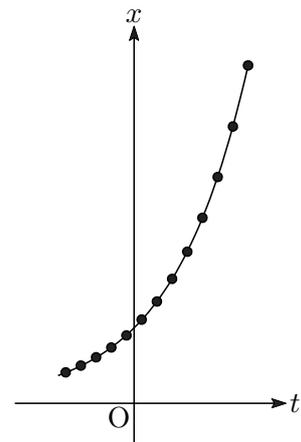


図 8 関数解と数値解

こうして時間区分列 trg と  $x$  の値の列 xrg が得られる。これらデータリストを用い、K<sub>F</sub>TCindy で真の解（関数解）と数値解の両方を同時に図示すると図 8 が得られる。

数値計算が有効な例として、歴史的に有名な振り子の等時性問題を扱う。振り子の等時性とは、「振り子のひもの長さが一定なら振幅の大きさに関わりなく周期は一定である」ことである。ひもの長さ  $a$  の振り子の往復運動と、半径  $a$  の摩擦のない円周上の質点の往復運動とは同じなので、今後は摩擦のない円周上の往復運動について考察する。このようにすると一般に、摩擦のない曲線  $y = f(x)$  上の質点の往復運動について議論することができる。例えば「放物線  $y = x^2$  上の質点の往復運動は等時性を持つか?」ということが考察可能になる。さて、振り子の等時性はガリレイによって発見されたが、のちに小さな誤差の範囲でのみガリレイの結論は正しいことが知られるようになった。等時性が成り立つ曲線はサイクロイドであることを証明したのはホイヘンスであり、1659 年のことであった。

曲線の方程式を  $y = f(x)$ , ( $-1 \leq x \leq 1$ ) とする。質点の質量を 1 とし、速度（曲線の接線方向速度）を  $v$ 、重力加速度を  $g$  とすればエネルギー保存則により次の式が成り立つ。

$$\frac{v^2}{2} + gy = gy_0$$

上式より質点の初期位置  $(x_0, y_0)$  において速度は 0 である ( $y_0 = f(x_0)$ )。本質的ではな

いので、以下では  $g = 1/2$  ととることにする。次の一連の関係式

$$v^2 = \left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2, \quad y = f(x), \quad \frac{dy}{dt} = f'(x) \frac{dx}{dt}$$

を上記方程式に代入して整理すれば  $x = x(t)$  に関する一階微分方程式が得られる。

$$\frac{dx}{dt} = \sqrt{\frac{f(x_0) - f(x)}{1 + f'(x)^2}}, \quad x(0) = x_0$$

この微分方程式の数値解を WE を呼び出すことにより計算するのだが、初めの簡単な例と比べると、右辺が複雑な関数になっていることが違うだけである。見かけはそうであるが、実は初期条件  $x(0) = x_0$  に問題があり、このまま解くと  $x(t) = x_0$  (つまり運動は起こらない) という力学的にはあり得ない解が得られる。そこで  $t = 0$  に初期条件をおくのではなく、 $t_1 > 0$  における  $x$  の値  $x_1$  を用いて、初期条件を  $x(t_1) = x_1$  とする必要がある。しかしここでは初期条件に関する問題についてこれ以上深入りしない。

数値計算結果を用いて、放物線と円の場合について、振幅が大きい赤点と振幅が小さい青点の2点の運動を図示したものが下図である。赤点が原点に到達したときの青点の位置は第3図からわかるが、どちらの場合も青点の方が先に進んでいる。つまり振幅の小さい青点の方が周期が短いことから、等時性はないことが見て取れる。

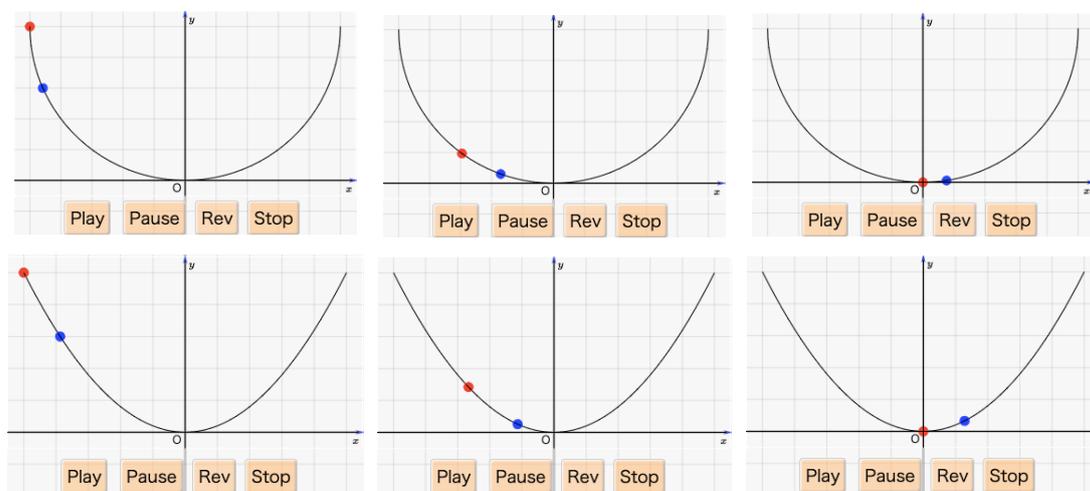


図9 放物線と円では等時性は成立たない

上図に対して、サイクロイドの場合は赤点と青点が同時に原点を通過している。したがって等時性が成立っていることがわかる。

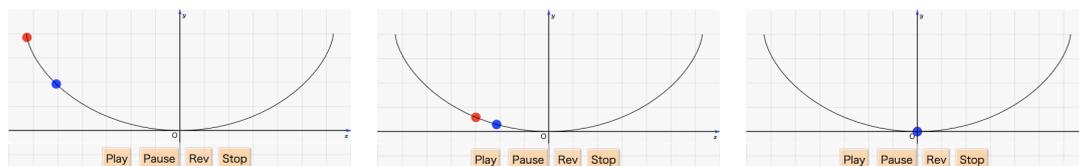


図10 サイクロイドでは等時性が成立つ

上記の3種類の図はブラウザ上で動画を動かし、適当な場所でポーズさせ、画面コピーをしたものである。それぞれの動画用HTMLファイルは大きさが50Kバイト程度と小さいので、学生に配布して自由に動かしてもらい、振動の様子を確かめることができる。動画用HTMLの作り方は、 $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ の拡張機能 $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}\text{J}\text{S}$ の作り方と同じである。少し違う点は、Playなどのボタン付きの $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ ファイルにプログラムを書かねばならないことである（`template4ketcindyjs.cdy`がテンプレートである。このファイルをコピーして名称を変更すればよい）。振動を表す微分方程式の数値解をWEを呼び出すことにより計算し動画を作るプログラムを見よう。数値解を作るところは以前と変わらないので、動画に相当する部分だけを取り出すと次の形になる。ここで $xrg, \dots, yrg1$ は2つの動点の位置座標のリストであり、既に作られているものとする。

```
Ketcindyjsdata(["xrg",xrg,"yrg",yrg,"xrg1",xrg1,"yrg1",yrg1]);//no ketjs
s=Animationparam(0.01,2,[0.01,26]);
u=ceil(s);
Pointdata("1",[5*[xrg_u, yrg_u]],["Size=7","Color=red"]);
Pointdata("2",[5*[xrg1_u, yrg1_u]],["Size=7","Color=blue"]);
```

$\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}\text{J}\text{S}$ 自体はCASを呼び出す機能をもっていないが、WEの呼び出しで得られたデータをHTMLファイルに書き込む命令として、`Ketcindyjsdata`を実装しており、この命令を用いることで位置座標リストがHTMLでも使用可能になる。なお、行末にコメントとして書かれている`//no ketjs`は、HTMLにはこの行を書き込まないという指示である。すなわち、`Ketcindyjsdata`はデータをHTMLに書き込んだ後はHTMLにとって必要のない命令なので、HTMLには書き込まないようにしている。`Animationparam`命令によりPlayボタンを押してからの経過時間が変数 $s$ に渡され、それを整数化して2点の表示を行なっている。繰り返しになるが、プログラムを書いている $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ ファイルから $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}\text{J}\text{S}$ のファイルを作るには、トップメニューから「ファイル>HTMLに書き出す」を選び、さらに`Ketjsoff`（または`Ketjson`）のボタンを押せばよい。

## 5 まとめと今後の課題

$\text{K}\epsilon\text{T}\text{p}\text{i}\text{c}$ は、 $\text{T}\epsilon\text{X}$ によって作成する教材に違和感のない図を簡単に挿入するために開発された。2014年には、動的幾何ソフトCinderellaをGUIとして用いることでインタラクティブに図を作成することができる $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ へと発展した。 $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ では描画コードとして`Tpic`, `pict2e`, `TikZ`をサポートし、それらのコードを出力する数学ソフトウェアとしてRを用いている。さらに、`CindyJS`の出力するHTMLコードに $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ の関数を選択的に付加してHTMLを作成する $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}\text{J}\text{S}$ を開発した。効果的な教材作成には、CASの利用が有用である。 $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ では、主にフリーのCASであるMaximaを利用してきたが、2019年、Wolfram社がFree Wolfram Engine for Developers(WE)を公開したことを受けて、 $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ にWEを呼び出すことができる機能を追加した。WEの関数のヘルプは体系的で使いやすく内容も豊富である。今後は、これらの関数の効果的な利用を探究するとともに、Maximaの利用との比較研究を行うことを考えている。

## 6 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] 金子真隆, 阿部孝之, 関口昌由, 山下哲, 高遠節夫, KETpic による曲面描画と教育利用, 数理解析研究所講究録 1624, pp. 1-10, 2009
- [2] 北原清志, 高遠節夫, 級数を扱う授業における KETpic で作成した挿図教材の使用, 数理解析研究所講究録 1624, pp. 90-105, 2009
- [3] 山下哲, 高遠節夫, KETpic による教材作成と Symbolic Thinking, 数理解析研究所講究録 1780, pp. 72-82, 2012
- [4] 高遠節夫, KeTCindy 開発チーム, KeTCindy の開発について, 数理解析研究所講究録 1978, pp. 173-182, 2015
- [5] 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 L<sup>A</sup>T<sub>E</sub>X Graphics Capabilities –, Mathematical Software –ICMS 2016, LNCS **9725**, 371–379, Springer, 2016
- [6] Gagern M., Kortenkamp U., Gebart J., Strobel M., CindyJS– Mathematical Visualization on Modern Devices,–ICMS 2016, LNCS **9725**, 319–334, Springer, 2016
- [7] 高遠節夫, TeX による教材作成環境の充実, 数理解析研究所講究録 2022, pp. 118-127, 2017
- [8] 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
- [9] Takato S, Vallejo J, Interfacing Free Computer Algebra Systems and C with KeTCindy, Computer Algebra Systems in Teaching and Research, Siedlce University of Natural Sciences and Humanities Volume 6, 172–185, 2017
- [10] 高遠節夫, KeTCindyJS の開発と教育利用, 数理解析研究所講究録 2142, 2019