

Sage 上での $\text{K}\epsilon\text{T}\pi\text{c}$ の利用について

呉工業高等専門学校・自然科学系分野 深澤 謙次 (Kenji Fukazawa)
Department of Natural Science,
Kure National College of Technology
東邦大学・薬学部 高遠 節夫 (Setsuo Takato)
Faculty of Pharmaceutical Sciences,
Toho University

1 はじめに

数学や物理学の研究者や教育者の中には、論文の作成に $\text{L}\text{A}\text{T}\text{E}\text{X}$ を用いる者が多くいるが、教材の作成となると $\text{L}\text{A}\text{T}\text{E}\text{X}$ ではなく、Microsoft Word などのワープロを使用する者も、特に物理教育者の中では、少なくない。その理由の 1 つは、 $\text{L}\text{A}\text{T}\text{E}\text{X}$ が図を扱うのが得意ではないことが考えられる。

教材にはきれいで正確な図が不可欠である。言葉や数式で説明してもなかなかわからないことが、図を 1 つ見せるだけで理解できることもある。したがって、 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 文書にきれいで正確な図を簡単に入れられるようにならない限り、教材の作成に $\text{L}\text{A}\text{T}\text{E}\text{X}$ を使うようにはならない。

TEX 文書にきれいで正確な図を挿入するためのツールとして開発されたものの 1 つに $\text{K}\epsilon\text{T}\pi\text{c}$ がある。 $\text{K}\epsilon\text{T}\pi\text{c}$ は数式処理システム (以下, CAS) 上で動作するパッケージであり、当初は著者の 1 人 (高遠) によって Maple 上で開発が始められた。現在の開発は主に Scilab 上で行われている。

近年、オープンソースソフトウェアの利用が広まり、一般の人々の間でも使われるようになってきた。現在、様々なソフトウェアがオープンソースソフトウェアやフリーソフトウェアとして公開されており、その中には CAS も含まれている。よく知られているフリーソフトウェアの CAS として Maxima, Scilab, Reduce などがある。また、CAS を含めて様々な数学ソフトウェアを利用するためのプラットフォームとして Sage がある。

Sage は代数, 幾何, 数論, 暗号学, 数値計算とそれらに関係した領域における研究と教育をサポートするフリーでオープンソースなソフトウェアであり, Sage の開発目的は Magma, Maple, Mathematica, MATLAB の代替となるフリーかつオープンソースなソフトウェアを提供することである。Sage 上で利用できる数学ソフトウェアには以下のようなものがある。

Axiom, FriCAS, Frobbly, GAP, Gnuplot, GP/Pari, Groebner Fan, KASH,
LiE, Macaulay2, Magma, Maple, Mathematica, MATLAB, Maxima, MuPAD,
mwrnk, Octave, PHC, QEPCAD, R, Scilab, Singular, ...

本研究では Sage 上での $\text{K}\epsilon\text{T}\pi\text{c}$ の利用について報告する。

2 K_εTpic とは

K_εTpic では T_εX 文書用の挿図を作成するために、Tpic を利用する。Tpic とは T_εX 用に開発された図形プリプロセッサ及びそれが出力する special コマンドセットの名称である。Tpic を用いて T_εX 文書に図を挿入するには、図を描くための一連の Tpic のコマンドの並びをファイルに書き込み、そのファイルを `\input` 文を用いて T_εX のマスターソースファイルに読み込めばよい。

K_εTpic はこの Tpic のソースファイルを作成するための CAS 上で動作するプログラム群として実装されている。K_εTpic を用いることで、ユーザーは Tpic のコマンドを知らなくても Tpic を利用した図が作成できる訳である。この結果、K_εTpic には以下のような特徴が生まれている。

- T_εX との親和性が良い（図の中に本文と同じ書体で数式が書ける）。
- 形と大きさに関して正確な図が描ける。
- 図の中に様々な装飾がつけられる。
- 豊かな表現力を持ったモノクロ線画が描ける。
- 修正が容易である。

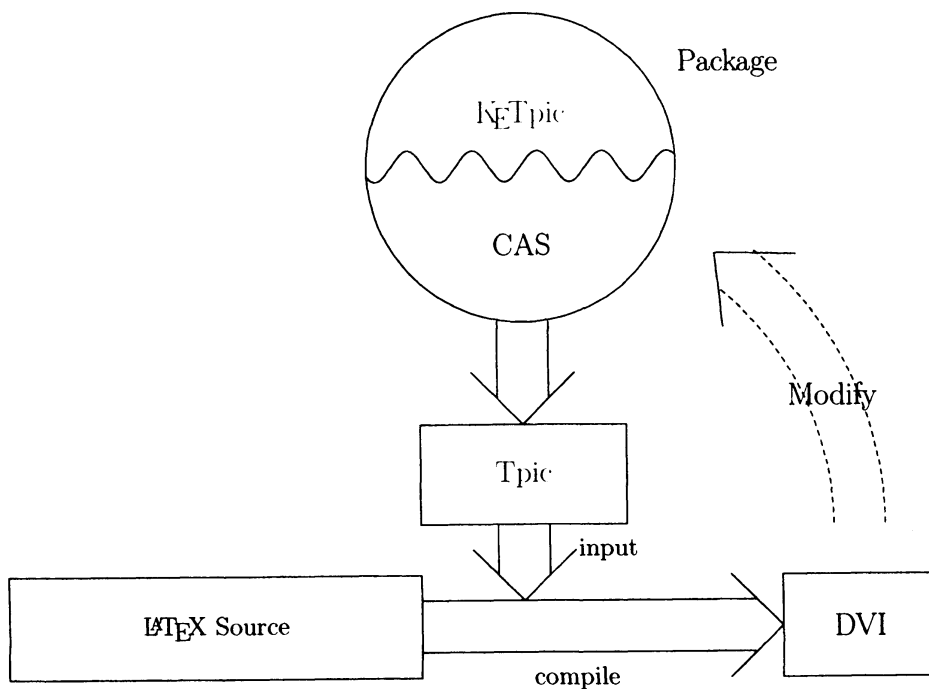


図 1: K_εTpic による作図手順

KETpic を用いて挿図を作成する手順を模式的に図示すると、図 1 のようになる。ユーザーは CAS 上で KETpic のコマンドを使って図を描くための一連のコマンドの並びを書き、 Tpic ファイルを作成する。このファイルを LATEX ソースファイルに読み込みコンパイルすると、挿図入りの dvi ファイルが得られる。図を修正したい場合は、CAS 上にもどり KETpic のコマンドを修正後、同じことを繰り返す。

現在、 KETpic を利用可能な CAS として Maple, Mathematica, Scilab, Maxima, R, Matlab がある。コマンドリファレンスなどは以下のサイトから自由にダウンロードできる。

<http://ketpic.com>.

3 Sage 上での KETpic

3.1 Sage 上での KETpic の利用方法

Sage 上で KETpic を利用するには以下の 2 通りの方法がある。

1. KETpic を Sage に移植する
2. Interpreter Interface を利用する

1. は KETpic を Python で書き直す必要があり、当然多くの手間と時間がかかる。また KETpic がバージョンアップする度にそれに追随しなければならない。一方で、この方法のメリットは特に何も思い付かない。

2. は Sage 上で Scilab を動作させ、Scilab 上で KETpic パッケージを利用するために、 KETpic 用の Interpreter Interface を定義しそれを使用する方法である。この方法では、1 度 KETpic 用の Interpreter Interface を定義すれば KETpic がバージョンアップしてもその度にいちいち追随する必要がないので、1. の方法に比べて大幅に少ない時間で済む。また、 KETpic がバージョンアップした場合、 KETpic の新しいパッケージをインストールするだけで新しい機能やバグフィクスされたバージョンが利用できるというメリットもある。その意味では、最近多くなってきたウェブアプリケーションに似ていると言える。これらの好ましいメリットに対して、この方法のデメリットはほとんどない。強いて上げれば、この方法を利用するための設定が今のところ簡単ではないということがある。そもそも、SAGE は様々な数学ソフトウェアを統一したユーザインタフェースで利用できるようにすることが目的の 1 つであるから、 KETpic を SAGE に移植するよりも、Scilab を SAGE 上で呼び出して、Scilab 版 KETpic を利用することを考える方が SAGE のデザイン哲学にも沿っており、また、現実的である。

どちらの方法を用いるにせよ、Sage 上では原理的には、Sage 上で動作する多くの数学ソフトウェアを利用して KETpic パッケージを用いた作図ができるはずである。

3.2 Interpreter Interface に必要なクラス

ここで Interpreter Interface に必要なクラスについて説明する。それらは spawn クラスと Expect クラスの2つである。spawn クラスはプロセスを fork して子プロセスを生成し、指定されたプログラムを実行するオブジェクトを作るためのクラスである。

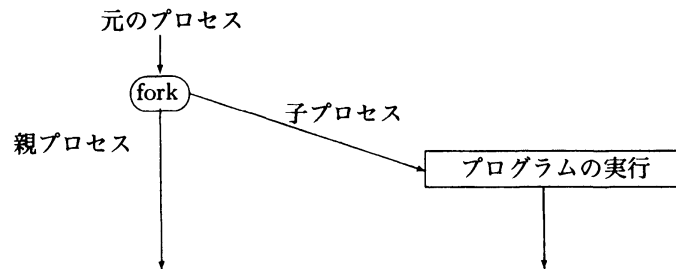


図 2: spawn クラスの動作

Expect クラスは spawn クラスを利用して指定したプログラム (interpreter) を子プロセスで実行するオブジェクトを作るためのクラスである。実行されたプログラムにはコマンドを文字列 (input stream) で送り、その出力は文字列 (output stream) で送り返される。この返された文字列の中から、正規表現と `_expect_expr`, `_before` メソッドを利用して必要なプログラムの出力部分を取得する。

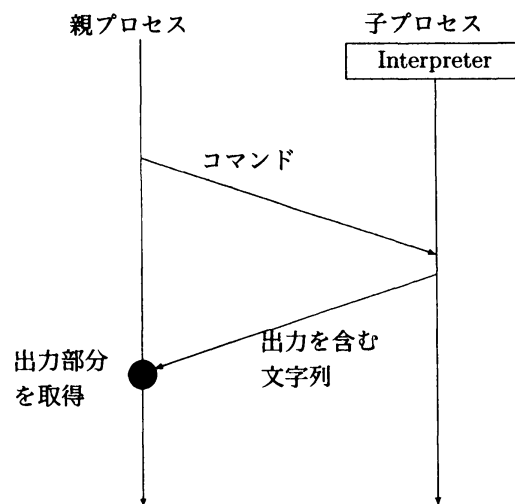


図 3: Expect クラスの動作

3.3 Interpreter Interface クラスの作成

Sage 上で KETpic for Scilab を利用するための Interpreter Interface を、作成する。ここでは、クラス名を `Ketsci` として Maxima の Interpreter Interface を真似してクラス

を定義すればよい。このクラスを用いて $\text{K}\epsilon\text{Tpic}$ コマンドを実行するには、基本的には

```
Ketsci().eval('コマンド')
```

とすればよいが、利便性のために $\text{K}\epsilon\text{Tpic}$ の各コマンド用のメソッドを定義しておいた方がよい。例えば、

```
ThisVersion()    → version()          #バージョンを表示する
Setwindow('...') → setwin('...')      #ウィンドウを設定する
Windisp(...)     → disp(...)         #画面上に図を表示する
Drwline('...')   → drw('変数名', '...') #データを実線で描く
•
•
•
```

ここでは、メソッド名を簡略化している。また、実行結果を変数に代入させる必要のあるコマンドについては次のようにメソッドを定義しなければならない。

```
Plotdata('...') → pd('変数名', '...') #グラフのデータを作成する
Listplot('...') → listpd('変数名', '...') #折れ線のデータを作成する
Lineplot('...') → linepd('変数名', '...') #延長した線分のデータを作成する
Paramplot('...') → parapd('変数名', '...') #パラメトリック関数のグラフのデータを作成する
•
•
•
```

このクラスを用いた簡単な実行例は次のようになる。

```
k = Ketsci()
k.setwin('[-2*\%pi,2*\%pi]', '[-1.5,1.5]')
k.pd('d', 'cos(x)', 'x')
k.disp(d)
k.open('fig.tex')

k.beginpic('1cm')
k.drw(d)
k.endpic()
k.close()
```

1行目で Ketsci クラスのオブジェクトを作成しており、その後はこのオブジェクトに対してメッセージを送ることで $\text{K}\epsilon\text{Tpic}$ コマンドを実行させている。

この例は簡単な \cos 関数のグラフを表示するものであり、図ファイルとして fig.tex を作成する。このファイルを $\text{L}\text{A}\text{T}\text{E}\text{X}$ ソースファイルに $\backslash\text{input}$ することで、グラフを含んだ dvi ファイルが得られる。 $\text{L}\text{A}\text{T}\text{E}\text{X}$ ソースファイルの簡単な例は以下のようなものである。

```

\documentclass[a4paper]{jsarticle}
\newlength{\Width}
\newlength{\Height}
\newlength{\Depth}
\begin{document}
\begin{center}
\input{fig}
\end{center}
\end{document}

```

3.4 Sage 上での $\text{KE}Tpic$ の利用例

ここでは *Sage* 上で $\text{KE}Tpic$ を利用した例をいくつか示す。

最初の例は *Schwarzschild* 時空中での質点の運動の軌跡を描いたものである。ただし、質点は $\theta = \pi/2$ の曲面上を運動すると仮定している。*Schwarzschild* 時空中は静的球対称な時空なので、この仮定をしても一般性を失わない。

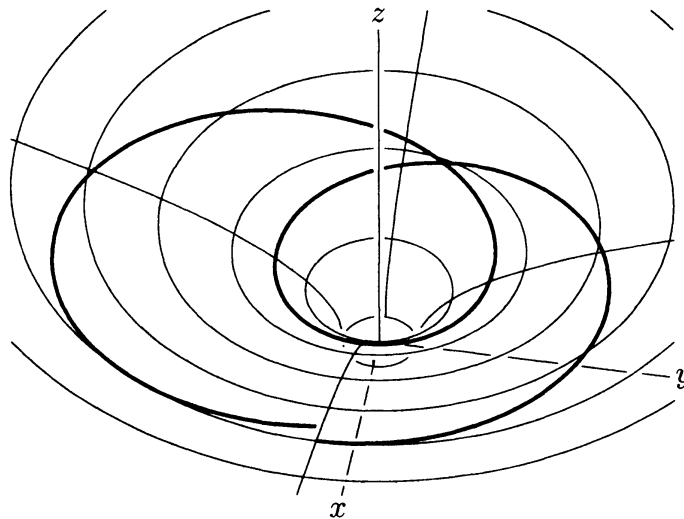


図 4: Schwarzschild 時空中での質点の運動の軌跡

次の例は *Schwarzschild* 時空中での座標時間の性質を図示したものである。この図では、座標時間 $t = t_0$ での時空面が xy 平面で表され、座標時間 $t = t_0 + \Delta t$ での時空面が曲面で表されている。そして、曲面の高さが空間の各点での 2 つの座標時間間隔を表している。図から、中心に近づく程時間間隔が大きくなり、中心では無限大になることがわかる。

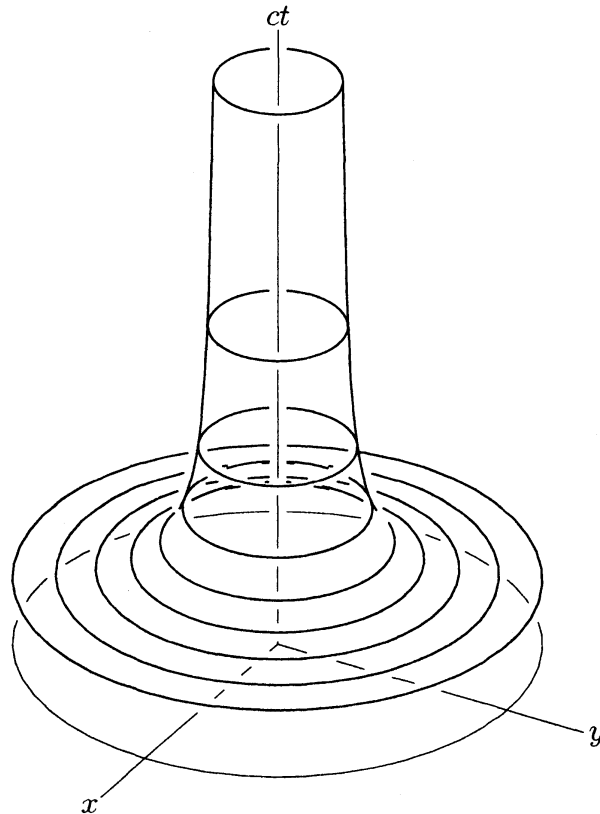


図 5: Schwarzschild 時空上での座標時間間隔

4 まとめ

本報告では *Sage* 上での *KETpic* の利用について議論した。 *Sage* 上で *KETpic* を利用するには

1. *KETpic* を *Sage* に移植する
2. *Interpreter Interface* を利用する

の方法が考えられるが、2. の方法が簡単で現実的である。そこで、 *KETpic for Scilab* を使うための *Interpreter Interface* を定義し、その実行例を示した。

問題点としては、 *Sage* 上で *KETpic* を利用するための設定が簡単ではないことがある。利用者を増やすためには、これを簡単にできるようにする必要がある。

また、 *Sage* 上で他の数学ソフトウェアを使って作成したデータを *KETpic* で利用できると、 *Sage* 上で *KETpic* を利用する大きなメリットになると考えられる。そのための方法を調べることも今後必要である。

謝辞

本研究は、科学研究費補助金基盤研究 C (課題番号 20500818) の補助を受けている。

参考文献

- [1] Y. Nakamura and S. Takato, Development of a graphical user interface for \LaTeX plotting software $\text{KE}\TeX\text{pic}$, *2009 International Conference on Computational Sciences and its Applications*, pp.109-114, *IEEE*, 2009.
- [2] M. Kaneko, T. Abe, M. Sekiguchi, Y. Tadokoro, K. Fukazawa, S. Yamashita and S. Takato, CAS-aided visualization in \LaTeX documents for mathematical education, to appear in *Teaching Mathematics and Computer Science*, Vol. VII, Issue II, 2009.
- [3] M. Kaneko, T. Abe, H. Izumi, K. Kitahara, M. Sekiguchi, Y. Tadokoro, S. Yamashita, K. Fukazawa and S. Takato, A simple method of the \TeX surface drawing suitable for teaching materials with the aid of CAS, *Lecture Notes in Computer Science*, 5102, pp. 35-45, *Springer-Verlag*, 2008.
- [4] M. Sekiguchi, T. Abe, H. Izumi, M. Kaneko, K. Kitahara, Y. Tadokoro, S. Yamashita, K. Fukazawa and S. Takato, Monochrome line drawings of 3D objects due to the programmability of $\text{KE}\TeX\text{pic}$, *2008 International Conference on Computational Sciences and its Applications*, pp. 277-283, *IEEE*, 2008.
- [5] M. Sekiguchi, M. Kaneko, Y. Tadokoro, S. Yamashita and S. Takato, A new application of CAS to \LaTeX -plottings, *Lecture Notes in Computer Science* 4488, pp.178-185, *Springer-Verlag*, 2007.
- [6] M. Sekiguchi, S. Yamashita and S. Takato, Development of a Maple macro package suitable for drawing fine \TeX -pictures, *Lecture Notes in Computer Science* 4151, pp.24-34, *Springer-Verlag*, 2006.