数理解析研究所講究録
第1674巻 2010年 31-39

KETpicの Maximaへの移植と SAGEへの移植の試み

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
木更津工業高等専門学校・基礎学系　阿部　孝之 (Takayuki Abe)
金子　真隆 (Masataka Kaneko)
Department of Fundamental Research,
Kisarazu National College of Technology
工学院大学・工学部　北原　清志 (Kiyoshi Kitahara)
Faculty of Engineering,
Kogakuin University
木更津工業高等専門学校・基礎学系　山下　哲 (Satoshi Yamashita)
Faculty of Fundamental Research,
Kisarazu 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 LaTeX を用いる者が多くいるが、教材の作成となると LaTeX 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、LaTeX 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、LaTeX 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 LaTeX 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 KETpic がある。KETpic 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。その後、Mathematica への移植が行われたが、 Maple や Mathematica は高価な商用の CAS であり、誰でもが気軽に購入して試せるソフトウェアではない。

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 などが知られている。現在、 KETpic は Scilab に移植されており、現在の開発は主に Scilab 上で行われている。

Scilab は Matlab と似た数値計算システムであるが、記号処理機能は備えていない。一方、 Maxima は Common Lisp の処理系上で動作するプログラムであり、他の CAS と同様に高度な記号処理機能を備えている。本研究では KETpic の Maxima への移植について報告する。

2 KETpic とは

KETpic では TEX 文書用の挿図を作成するために、Tpic を利用する。Tpic とは TEX 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 TEX 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 TEX のマスターソースファイルに読み込めばよい。

KETpic 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。KETpic 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、 KETpic には以下のような特徴が生まれている。

- TEX 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。

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。

KETpic のコマンドは大きく 2 種類に分けられる。

1. 描画データ（plot data）を作成するためのコマンド
2. Tpic ファイルを作成するためのコマンド

必ずしも全ての KETpic コマンドがどちらかに分類できる訳ではないが、初心者はどのコマンドがどちらに属するかを意識しておくとよい。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.

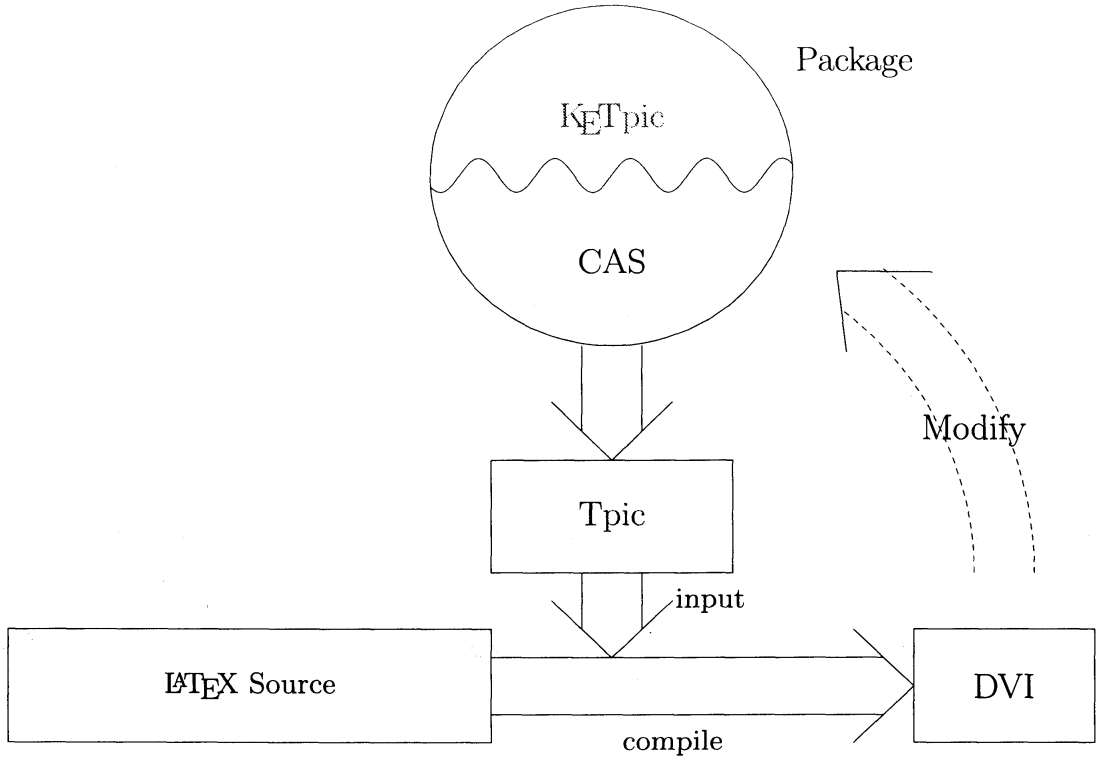

図 1: KETpic による作図手順

3 KETpic の Maxima への移植

KETpic の Maxima への移植は、基本的には他の CAS への移植と同じであるが、いくつか気をつけなければならない点がある。

1. Lisp 処理系への依存性に注意する。
2. 再帰関数は使わない（代りに `for` 文を使う）。

Maxima は Common Lisp の処理系上で動作するのだが、再帰関数を使うとなぜかエラーが生じることがある。これを `for` 文で書き直すと問題なく実行できるので、最初から再帰関数は使わないようにした方がよい。1. の Lisp 処理系への依存性は、特に KETpic を `save` コマンドで保存した状態で配布しようとしたときに障害となる。

Maxima 版 KETpic の使い方は Scilab 版 KETpic などと同じである。ただ、いくつかコマンドの引数や動作などに違いがある。Maxima 版 KETpic での `enclosing` コマンドは他の版とは異なり、指定された plotdata でできる全ての「最小の閉曲線」を計算し、その plotdata を出力するようになっている。ここで「最小の閉曲線」とは、内部に他の閉曲線を含まない閉曲線である。

次のスクリプトはコマンド enclosing を用いた例であり、3つの曲線からできる3つの最小の閉曲線を計算する。

```
1:  load("ketpic.mac")$
2:  setwindow([-3, 7], [-6, 4])$
3:  f1(x):= x^2 + 1$
4:  f2(x):= -f1(x)$
5:  p1: plotdata(f1(x), [x, -3, 3])$
6:  p2: plotdata(f2(x), [x, -3, 3])$
7:  c1: circledata([0,0], 4)$
8:  trp1: translatedata(rotatedata(p1, -\%pi/4), 2, -1)$
9:  trp2: translatedata(rotatedata(p2, -\%pi/4), 2, -1)$
10: trc1: translatedata(rotatedata(c1, -\%pi/4), 2, -1)$
11: ec: enclosing(trp1, trp2, trc1)$
```

enclosing コマンドを実行した後に enclose_view コマンドを実行すると、これらの最小の閉曲線を順に表示させることができる (図 2)。

```
12: enclose_view(ec, trp1, trp2, trc1)$
```

特定の閉曲線だけを取り出すには、その閉曲線の表示された順番を覚えておき、以下のようにすればよい。

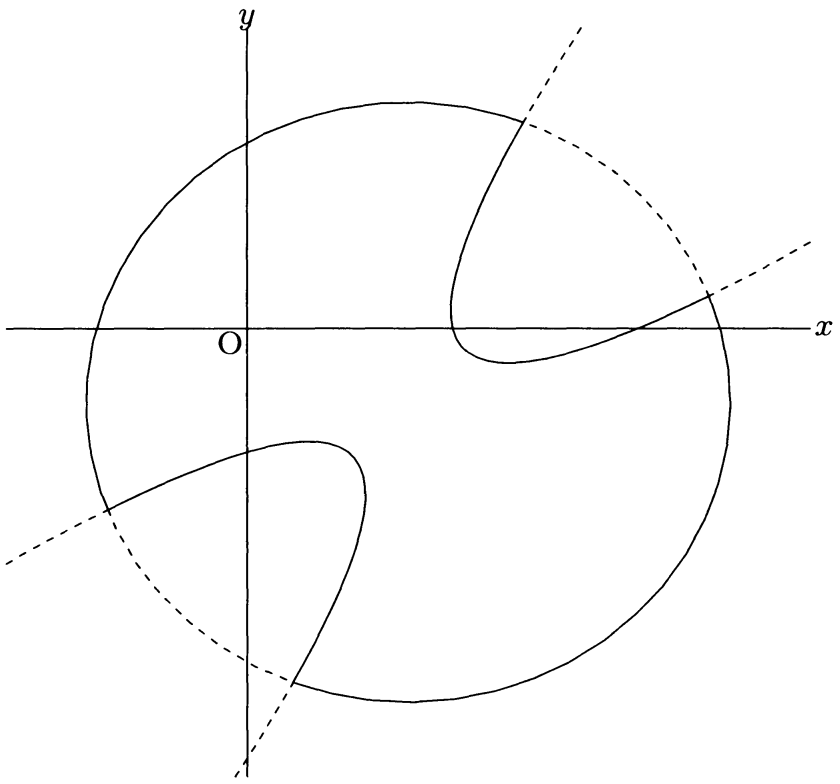

図 2: enclose_view コマンドの実行例 (表示される最初の図)

```
13:     pd: [ec[1]]$
```

この例では、最初の閉曲線だけを取り出す場合を示している。

また、特定の閉曲線に斜線を入れたい場合は enchatchdata コマンドが利用できる。

```
14:     ech: enchatchdata(ec, [2, 3])$
15:     openFile("fig.tex")$
16:     beginpicture("1cm")$
17:     setpen(0.5)$
18:     drwline(ech)$
19:     setpen(1)$
20:     dashline(trp1, trp2, trc1)$
21:     drwline([ec[1]])$
22:     endpicture(1)$
23:     closeFile()$
```

この例では、2番目と3番目の閉曲線に斜線を入れている (図 3)。

Maxima 版 KETpic の enclosing コマンドの有用性は、以下に示すように、すべての最小の閉曲線にそれぞれ異なった斜線を入れたい場合に明かになる (図 4)。

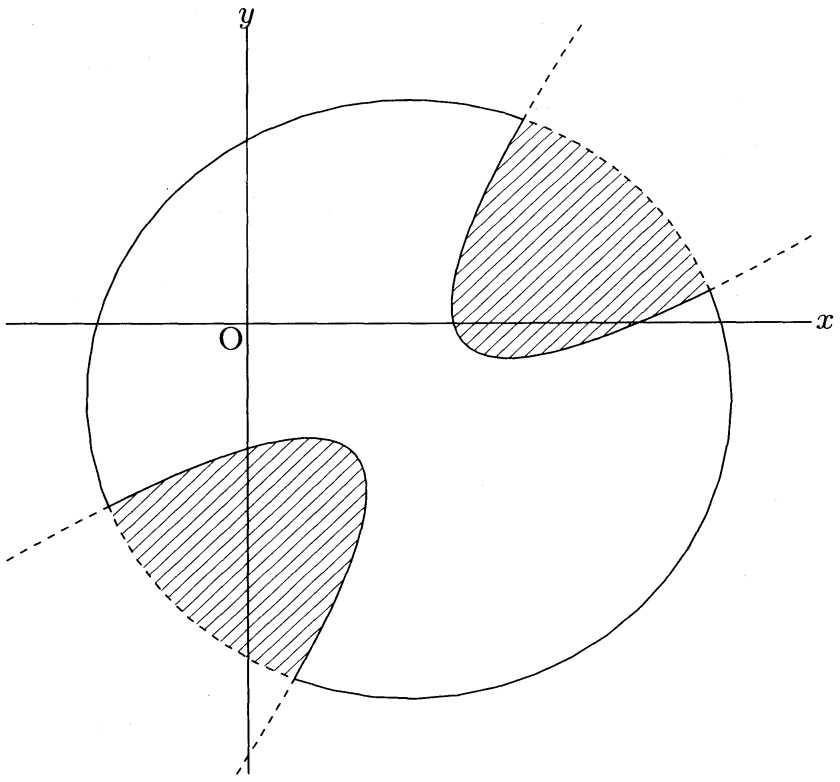

図 3: enchatchdata コマンドの実行例

```
24:     nesw: listplotdata(ptne(), ptsw())$
25:     ecs: enclosing(trp1, trp2, trc1, nesw)$
26:     openFile("fig.tex")$
27:     beginpicture("1cm")$
28:     setpen(0.5)$
29:     dashline_all(ecs, 1.5)$
30:     setpen(1)$
31:     drwline(trp1, trp2, trc1, nesw)$
32:     endpicture(1)$
33:     closeFile()$
```

ここで dashline_all はすべての最小の閉曲線にそれぞれ異なった斜線 (破線) を入れるコマンドであり、以下のように定義される。

```
1:    dashline_all(ec_all, interval):=
2:    block([_n_ec, _i, _dir, _hd, _ptn, _tmp],
3:      _n_ec: length(ec_all),
4:      for _i thru _n_ec do (
5:        _dir: 180 / (_n_ec + 1) * _i,
6:        _hd: [hatchdata("i",[[ec_all[_i]]],_dir,interval)],
7:        _ptn: 2 * [_i, _n_ec + 1 - _i] / (_n_ec + 1),
8:        _tmp: append([_hd], _ptn),
9:        apply(dashline, _tmp)    )  )$
```

他の版の enclosing コマンドを使用して同じ図を作成する場合は、それぞれの閉曲線ごとに斜線を入れていかなければならないので、最小の閉曲線の数が多い場合はかなり煩雑な作業になってしまう。破線を実線にしたい場合は 26 行以降の部分を以下のように変更すればよい。

```
26:     hd: hatch_all(ecs, 1.5)$
27:     openFile(figfile)$
28:     beginpicture("1cm")$
29:     drwline(hd, 0.2)$
30:     drwline(trp1, trp2, trc1, nesw, 1)$
31:     endpicture(1)$
32:     closeFile()$
```

ここで hatch_all はすべての最小の閉曲線にそれぞれ異なった角度の斜線 (実線) を入れるコマンドであり、以下のように定義される。

```
1:    hatch_all(ec_all, interval):=
2:    block([_hd, _n_ec, _i, _dir],
3:      _hd: [],
4:      _n_ec: length(ec_all),
5:      for _i thru _n_ec do (
6:        _dir: 180 / (_n_ec + 1) * _i,
7:        _hd: append(_hd,
8:            hatchdata("i",[[ec_all[_i]]],_dir,interval))),
9:      _hd  )$
```

最小の閉曲線の数が多い場合に Maxima 版 KETpic の enclosing コマンドを使用すると、最小の閉曲線の計算に時間が掛かってしまうので、その点には注意が必要である。

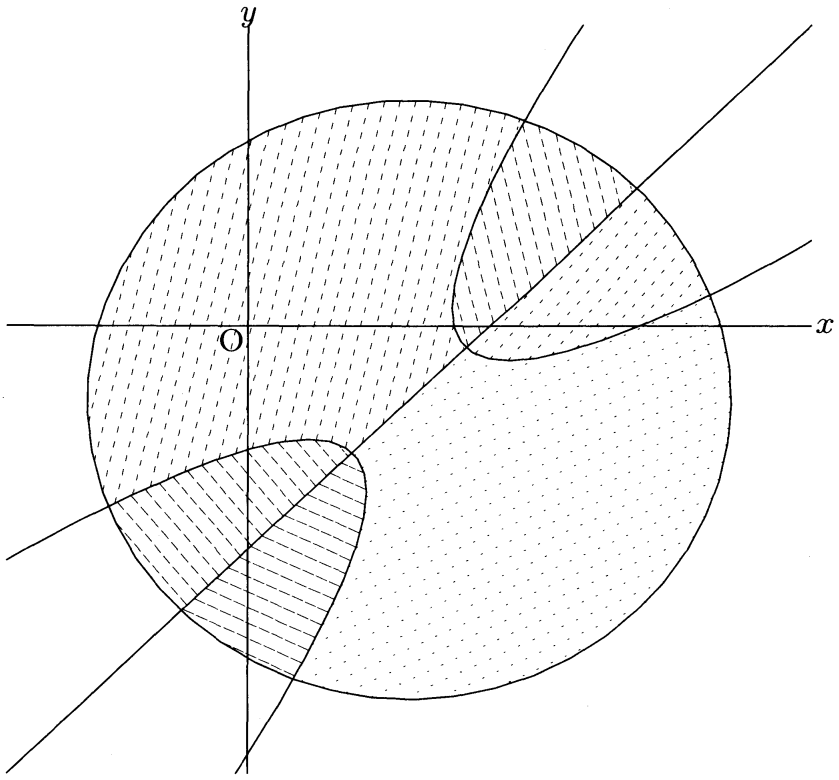

図 4: すべての最小の閉曲線に斜線 (破線) を入れる例

4 Maxima 版 KETpic の将来展望と SAGE 上での利用

現状では Maxima 版 KETpic は空間曲線描画機能まで移植が済んでいるが、空間曲面描画機能はまだ移植されていない。例えば、空間曲線描画機能の 1 つである skeleton 法を利用するとわかるが、Scilab 版 KETpic と比べて Maxima 版 KETpic では計算に時間が掛かってしまうため、 Maxima 版 KETpic で空間曲面描画機能を利用するのはあまり現実的ではない。むしろ、 Maxima 上で Scilab 版 KETpic を呼び出すことを考えた方

が実際的であると思われる。この方法には、Scilab 版 KETpic でなされた空間曲面描画機能の追加やバグフィックスに対して追随する必要がないというメリットもある。

Unix 系の OS 上では、Maxima の `system` 関数が利用できる。これを利用すると、外部プログラムを起動することができ、また、 Scilab は

```
scilab -f "File"
```

とすることで`"File"` に書かれた Scilab スクリプトの内容をバッチ的に実行することができるので、このことを利用すると以下のようにして Maxima 上で子プロセスとして Scilab を実行させることができる。

```
system("scilab -f <Scilab-script.sce>")$
```

この `<Scilab-script.sce>` ファイルの中に Scilab 版 KETpic のスクリプトを書けば、Scilab 版 KETpic を実行することができる。したがって、上記の `<Scilab-script.sce>` を生成する Maxima 関数を定義すれば Scilab 版 KETpic が利用できる。ただし、この方法はすべての OS 上で利用できる訳ではないので、動作環境に注意する必要がある。

SAGE 上で KETpic を利用できるようにするには、いくつかの方法が考えられる。1 つの方法は KETpic を SAGE に移植することであるが、この方法にはいくつかの欠点が考えられる。

- 移植には時間が掛かる。
- SAGE は動的プログラミング言語 Python を利用しているが、 Python は実効速度が遅いので、 KETpic を移植しても Scilab 版 KETpic と比べて速くなるとは期待できない。
- KETpic がバージョンアップする度に常に追随し続けなければならない。

これらの欠点に対して、SAGE に移植する長所は特にないと思われる。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、例えば Scilab を SAGE 上で呼び出せるように修正して、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。この方法では、 KETpic を移植する場合に存在する欠点がないという大きなメリットもある。今後は、この方法に沿って考えていく予定である。

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 KETpic, 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 KETpic, 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.