数理解析研究所講究録
第1865巻 2013年 43-48

Maxima 上での KETpic の実装について

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 Sciences,
Kure National College of Technology
東邦大学・薬学部　高遠 節夫 (Setsuo Takato)
Fucluty of Pharmaceutical Science,
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，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 は数学の配布用印刷教材の作成を目的に開発が始められ，数式処理システム（以下，CAS）上で動作するパッケージとして提供されている．現在の開発は主に Scilab 上で行われている．

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，この結果として，KETpic には以下のような特徴が得られている．

- 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.

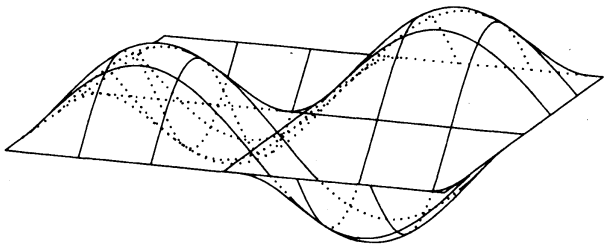

図1 KETpic による曲面の例

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2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．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.

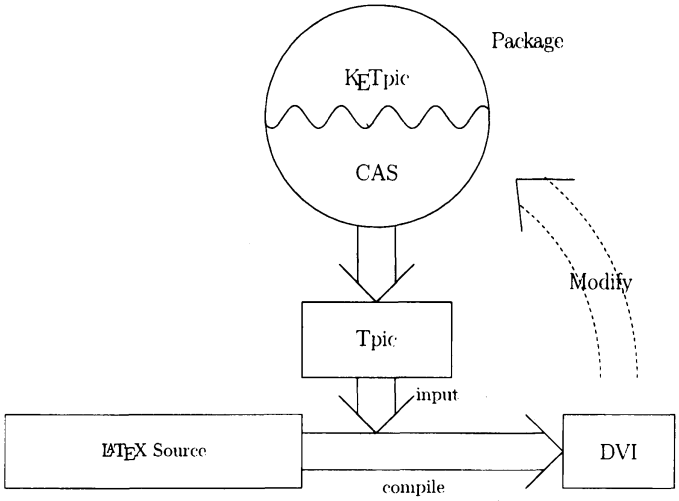

図2 KETpic による作図手順

KETpic では図の中に様々な装飾をつけるために曲線の交点などの補助データを内部で計算している．例えば，曲線で囲まれた領域にハッチングをつけるためにはそれらの曲線と直線との交点の座標を求めなければならない．そのために KETpic ではこれらの曲線の基本データ (プロットデータ) を作成し，基本データを基にして交点を見つけている．以上をまとめると以下のようになる．

1. 基本データ (プロットデータ) を作成する
2. 基本データを基に交点などの補助データを計算する

3. 補助データを基にハッチングなどの2次データを作成する

4. 以上のデータを基に，図の tpic(or pict2e) ファイルを作成する

5. アクセサリなどを追加する

基本データを基にして交点などの補助データを計算するこのような方法には他の CAS への移植に手間が掛かるという問題がある．これらは CAS に依存しない部分であるので，C 言語などで実装しライブラリ化して利用するという方法も考えられる．

一方， Maxima などの CAS には標準で様々な便利な関数が用意されており，利用することができる．例えば，Maxima で用意されている関数には以下のものがある．

`taylor`	$taylor(< expr >, < x >, < a >, < n >)$ Taylor or Laurent 級数に展開する
`solve`	$solve([< eqn\_1 >, ..., < eqn\_n >], [< x\_1 >, ..., < x\_n >])$ 方程式の解を求める
`mnewton`	$mnewton(< FuncList >, < VarList >, < GuessList >)$ Newton method で解を求める
`cspline`	$cspline(< points >, < option1 >, < option2 >, ...)$ 3次スプライン法による多項式補間を計算する
	...

本論文では，Maxima で用意されている関数を利用して曲線の交点を計算する方法について検討する．

2 曲線の交点

例として 図3 のような螺旋図を考える．この螺旋図では，螺旋を射影した平面上での “交点” を求め，曲線の一部にカットを入れることによって，螺旋に立体感を持たせている．ここでは，`taylor`, `solve`, `mnewton` を利用する方法を説明する．

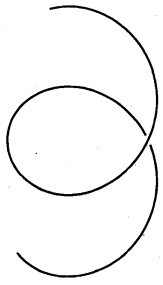

図3 KETpic による螺旋の例（平行投影）

2.1 螺旋図の作成の準備

螺旋図を作成するための準備をまとめると以下のようになる.

- 空間曲線（螺旋）の設定　　$\vec{r} = (\cos t,\ \sin t,\ t)$

  r3d(t) := [cos(t), sin(t), t]$　(Maxima code)

- 視点の方向の設定（平行投影）

  setangle(20,60)$　(KETpic code)

  ◇ 法線ベクトル（注視点は原点） $\vec{n}$

  ◇ 射影平面（原点を通り $\vec{n}$ に垂直）

- 射影平面上での曲線

  r_plane(t) := r3d(t) - inprod(n, r3d(t)) * n$　(Maxima code)

  r2d(t) := [inprod(r_plane(t), ex_pp),
  inprod(r_plane(t), ey_pp)]$　(Maxima code)

  法線方向 z(t) := inprod(r3d(t), n)$　(Maxima code)

ここで `inprod` は Maxima 上で用意された内積を計算する関数であり，射影平面上での x, y 方向の単位ベクトルを ex_pp, ey_pp としている．r2d(t) が射影平面上に射影された曲線を表す関数である.

2.2 射影平面上での交点

ここでは交点を求めるために，曲線を部分曲線に分割し全ての部分曲線の組み合わせについて，交点の有無を調べている．交点の有無を調べるために，各部分曲線をその始点のまわりで Taylor 展開しそれらに対して Maxima の `solve` 関数を用いることで交点の有無を判定している．`solve` の解が見つかった場合は，それを初期値として Maxima の `mnewton` 関数を利用してより正確な交点の値を求めている．以上ををまとめると以下のようになる.

1. 射影平面上の曲線 (r2d(t)) を部分曲線に分割

   分割条件:　部分曲線の長さが部分曲線の両端の距離の
   CRV_LNGTH_LIMIT 倍以下（default は 1.5）

   get_proper_regions(func, var, regions)　(自作の Maxima 関数)

2. 各部分曲線を始点のまわりで Taylor 展開

   eq1: `taylor`(x1(t1)-x2(t2), t1, rgn1_st, 3, t2, rgn2_st, 3)　(Maxima code)

   eq2: `taylor`(y1(t1)-y2(t2), t1, rgn1_st, 3, t2, rgn2_st, 3)　(Maxima code)

   (注) 曲線 1 r2d_1(t) = (x1(t), y1(t)) と曲線 2 r2d_2(t) = (x2(t), y2(t)) の範囲がそれぞれ rgn1, rgn2 であり，それらの始点が rgn1_st, rgn2_st で表されるとする．t1, t2 は変数である.

3. すべての部分曲線の組み合わせで交点の有無を調べる

ans: `solve`([eq1, eq2], [t1, t2])　(Maxima code)

4. 交点がある場合，3. で得られた解を初期値にして Newton method で交点を求める

para: `mnewton`([x(t1)-x(t2), y(t1)-y(t2)], [t1, t2], ans[1])$
(Maxima code)

pt1: `float`(`ev`([x(t), y(t)], t = para[1]))$　(Maxima code)
(例) [0.5950476556147, 1.969894606719056]

pt2: `float`(`ev`([x(t), y(t)], t = para[2]))$　(Maxima code)
(例) [0.5950476565923, 1.969894600543818]

h1: `float`(`ev`(z(t), t = para[1]))$　(Maxima code)
(例) 3.062407372837459

h2: `float`(`ev`(z(t), t = para[2]))$　(Maxima code)
(例) 7.762074521610913

4. の例の場合，pt1 と pt2 の座標がほぼ等しいことがわかり，$h1 < h2$ であるから pt1 が pt2 より視点から遠い位置にあることがわかる．

2.3　射影平面上の曲線のパラメータの範囲の決定

射影平面上の曲線の交点がすべて求まり，それぞれの交点の視点から遠い側の点が決まれば，その点に対応するパラメータ値を (ある範囲で) 曲線のパラメータの範囲から除くことで立体感のある螺旋図が作成できる．

上の例の場合，交点に対応するパラメータについて，視点側に対応する値 (t = para[2]) はそのままで，交点の視点から遠い側に対応する値 (t = para[1]) の近傍にカットを入れる．その結果，パラメータを $0 \leqq t \leqq \text{para}[1] - \Delta$ と変更することで 図3 が得られる．

3　考察とまとめ

本論文では，Maxima で用意されている関数を利用して曲線の交点を計算する方法について検討した．ここでは交点を求めるために，曲線を部分曲線に分割し全ての部分曲線の組み合わせについて，交点の有無を調べた．交点の有無を調べるためには，各部分曲線をその始点のまわりで Taylor 展開しそれらに対して Maxima の `solve` 関数を用いることで交点の有無を判定した．`solve` の解が見つかった場合は，それを初期値として Maxima の `mnewton` 関数を利用してより正確な交点の値を求めた．

この方法では Taylor 展開と `solve` 関数を利用しているため，曲線が接する場合に接点を正確に求めることは難しい．他の方法として3次スプライン法を利用することが考えられるが，交点の有無を判定する方法として Maxima の `solve` 関数を用いない方法が必要である．考えられる1つの方法は，区間 $[a, b]$ 内の N 個の点で2つの曲線の上下

の位置関係を調べる方法である．この方法で接点を正確に求めるには，適切な点の数 N をどう決めるかという問題と 2 曲線がどの程度近付いたら接すると見做すかという問題について解決しておかなければならない．これについてはいくつかの具体的な曲線に対して具体的に計算して決めるしかないと思われる。

参考文献

[1] CASTEX 応用研究会編, KETpic で楽々 TEX グラフ, イーテキスト研究所, 2011.