

Risa/Asir の関数描画機能の拡張について¹⁾

詫間電波高専 近藤 祐史 (Yuji Kondoh)
埼玉女子短大 三好 善彦 (Yoshihiko Miyoshi)
上智大学理工学部 齋藤 友克 (Tomokatsu Saito)

Abstract.

Visualization helps us to understand phenomena, to see into the background, to discover a new question and so on. Usually, we use a numerical method to draw graphs, but little of them is faithful.

In this paper, we extended the plotting functions on Risa/Asir. Our system can faithfully plot algebraic functions without loss of minute structures of algebraic functions. We can use Gröbner basis to detect the minute structures.

1. はじめに

数式処理の分野において関数の形状を表示することは、現象の理解、背景の洞察もしくは新しい課題の発見等のために欠くべからざるプロセスである。特に数式で与えられた問題の幾何的イメージを把握できれば、解の存在範囲や交点の個数配置などが容易に見る事ができ、最終的な答えを導き出すことに役立つことが多い。

関数のグラフを表示することは一見容易な問題に見えるが、与えられた精度で関数のグラフを確実に描くことは非常に困難である。

我々はこの問題を ifplot 問題と呼び、2 変数代数関数の場合に限って一般的なアルゴリズム [3] を報告した。また、これらのアルゴリズムをインプリメントした陰関数描画システム [2] を試作した。本稿では、本システムの現状と追加した機能について報告する。

2. Cell plotting

関数の描画とは、与えられた表示領域 D での関数 $f(x_1, x_2) = 0$ の零点を必要とされる精度で、出力装置 (ディスプレイ、プリンタ等) に出力することである。領域 D を必要とされる精度でメッシュに切り、セル S_i に分割することにより、

$$D = \bigcup_{i=1}^n S_i, \quad S_i \cap S_j = \phi, \quad i \neq j,$$

と表せる。描画とは各セル S_i において零点が存在するか否かの判定を行うことである。このような描画法を cell plotting と呼ぶことにする。

¹⁾ An extension of the plotting functions on Risa/Asir

Cell plotting において、領域に関する定義として基本単位系とセル S に零点が存在する
か否かの判定に対する定義として描画関数を次のように定める。

2.1. 描画基本単位系

D 上定義された χ の基本単位系 S とは、

$$\begin{cases} S_i \text{ は } D \text{ の compact subset,} \\ S \text{ は } S_i \text{ の有限集合,} \\ D = \bigcup_{i=1}^n S_i, \\ S_i \cap S_j = \phi \quad i \neq j \end{cases}$$

と定義する。ここで S^i は S の内点の集合であり、 n はセルの総数である。

2.2. 描画関数

描画関数 $\chi(D, S)(f)$ とは、

$$\begin{cases} 1. \chi: S \rightarrow \{0, 1\} \\ 2. \chi(S_i) = 0 \quad S_i \text{ の任意の元 } x \text{ に対し } f(x) \neq 0 \end{cases}$$

と定義する。この $\chi(D, S)$ を D における S による character とよぶ。

2.2.1. Faithful character

D における S による faithful character χ とは、

$$\begin{cases} 1. \chi \text{ は } S \text{ による character} \\ 2. \chi(S_i) = 1 \quad \exists x \in S_i \text{ s.t. } f(x) = 0 \end{cases}$$

と定義する。

3. 2変数多項式の faithful character

今、セル S を \mathbb{R} 上の矩形領域とし、 $f(x_1, x_2)$ を 2 変数有理係数多項式とする。このとき、 $f(x_1, x_2) = 0$ がセル S において解を持つ必要十分条件は、次の定理により与えられる。

定理 1. 2 変数代数方程式 $f(x_1, x_2) = 0$ は \mathbb{Q} において無平方とする。 f が領域 S において零点を持つ場合は次の 3 つの場合しかない。

1. f が S の境界上に零点を持つ。
2. Gröbner 基底 $\mathcal{G}(\partial f/\partial x_1 = 0, \partial f/\partial x_2 = 0, f = 0)$ が S 内に零点を持つ。
3. Gröbner 基底 $\mathcal{G}(f = 0, \partial f/\partial x_1 = 0)$ が S 内に零点を持つ。

この定理で場合 3 は場合 2 を含むため、セル S に零点がある否かを次のように判定する。

1. f を無平方にする。
2. S の境界線上で Sturm の定理による方法により零点があるか判定する。
3. $\mathcal{G}(f = 0, \partial f/\partial x_1 = 0)$ が S 内に零点を持つか否か判定する。

4. インプリメントと実行例

本システムは、数式処理ライブラリ Risa および Asir の Gröbner 基底計算部を利用する。また、Asir とのプロセス間通信のインタフェースを備えることにより、`asir_plot` と同様な使用ができるように実現した。また、新たに、

1. Postscript 出力
2. 点、線分、円、円弧などのオブジェクトの上書き機能

などの機能を付加した。そのため、線分などのオブジェクトの上書き機能を利用する方法として、Asir に `plotobj()` 関数を加えた。

本システムの実行例に、以下の代数関数を用いる。

$$f_1(x, y) = x^5 - 2yx^2 + y^5$$

$$f_2(x, y) = \frac{93392896}{15625}x^6 + \left(\frac{94359552}{625}y^2 + \frac{91521024}{625}y - \frac{249088}{125}\right)x^4 + \frac{1032192}{25}y^4$$

$$- 36864y^3 - \frac{7732224}{25}y^2 - 207360y + \frac{770048}{25})x^2 + 65536y^6 + 49152y^5$$

$$- 135168y^4 - 72704y^3 + 101376y^2 + 27648y - 27648$$

$$f_3(x, y) = f_2(x, y) - \frac{1}{5}$$

f_2 を 600 dpi(2400×2400 セル) で計算した PostScript 出力結果を Fig. 1 に示す。また、タイミングデータを Table 1,2 に示す。計算は FreeBSD 2.2-SNAP Pentium 166MHz 上で行った。Fig. 1 中の円は中心のセル内に孤立特異点またはセルの内部で閉じた微細構造があることを示す。

Table 1. 画面出力の場合の計算時間 (秒)

	f_1	f_2	f_3
境界部	5.93 + gc : 0.74	12.40 + gc : 1.98	11.49 + gc : 3.78
Gröbner 部	0.06 + gc : 0.01	1.09 + gc : 0.08	3.63 + gc : 0.47
total	5.99 + gc : 0.75	13.49 + gc : 2.06	15.12 + gc : 4.25

境界部：Sturm の定理による方法

Gröbner 部：Gröbner 基底とその解の計算

Table 2. Postscript 出力の場合の計算時間 (秒)

f_1	f_2	f_3
48.66 + gc : 21.45	95.38 + gc : 17.09	90.45 + gc : 19.57

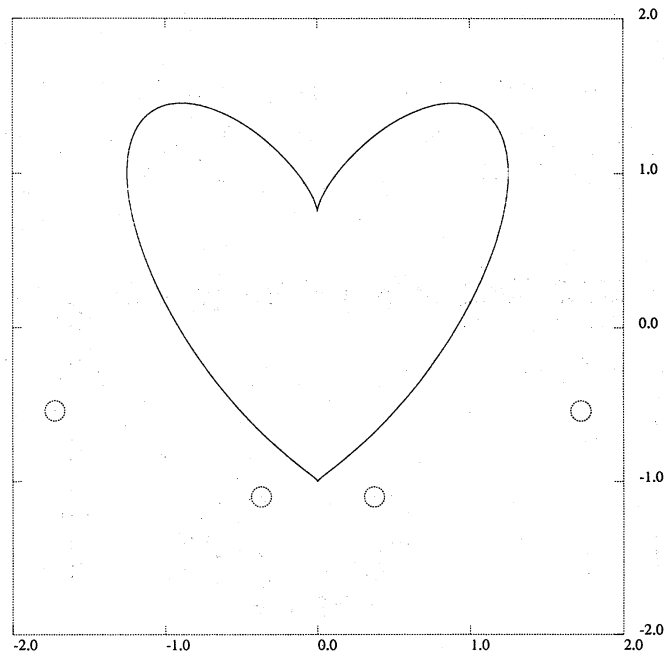


Fig. 1. f_2 の Postscript 出力 (600 dpi)

5. Cell Trace

セルの境界部の計算効率を上げるため、Cell Trace を試みた。この方法は、表示領域の境界上の零点を含むセルと $\{f = 0, \partial f / \partial x = 0\}$ の解と表示領域の境界上の $f = 0$ の解を含むセルから出発して、セル毎の追跡を行うものであり、以下の手順で境界部の Sturm を計算する。

1. $\{f = 0, \partial f / \partial x = 0\}$ の解と表示領域の境界上の $f = 0$ の解を含むセルを黒く塗り、その周りのセルを候補に入れる。
2. 候補のセルを一つ取り出し、その境界線上で Sturm により零点があるかどうか調べる。このとき、一度計算した境界線を再度計算しないように工夫する。零点があるのセルを黒く塗り、その周りのセルを候補に加える。(ただし、既に黒く塗られているセルは除く。)
3. 候補が無くなるまで 2. を繰り返す。

Cell Trace と素朴な方法との計算時間を Table 3 に示す。この結果より、Cell Trace は曲

Table 3. 素朴な方法と Cell Trace の境界部計算の時間 (秒)

	f_1	$f_2[-2, 2][-2, 2]$	$f_2[-4, 4][-4, 4]$
素朴な方法	6.95 + gc: 2.40	11.11 + gc: 1.61	7.01 + gc: 1.00
Cell Trace	5.68 + gc: 6.40	10.35 + gc: 2.99	5.04 + gc: 1.16

線が表示領域の一部しかないような場合には有効である。

6. 簡易 3 次元表示

本システムに付加したオブジェクトの上書き機能を利用することにより、 $z = f(x, y)$ の簡易 3 次元表示プログラムを開発した。このプログラムは、新たに付加した `plotobj()` 関数を用いることにより、Asir 言語で記述してある。Fig. 2 に、 $z = x^2 + y^2$ での実行例を示す。

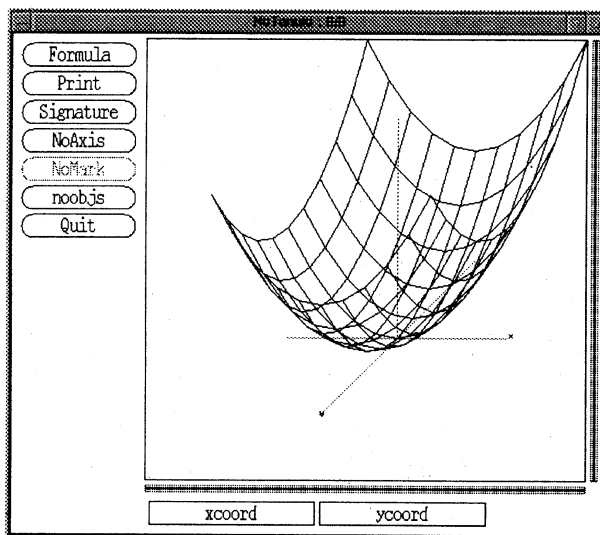


Fig. 2. $z = x^2 + y^2$ の簡易 3 次元表示

7. まとめ

我々は、数式処理システム Risa/Asir の描画機能を拡張し、有理数係数 2 変数代数関数の零点を表示装置の分解能の精度で正確に描画するシステムを開発した。また、本システムには、

1. Postscript 出力の faithful character への対応
2. 点、線分、円、円弧などのオブジェクトの上書き機能

などの機能を付加している。この機能を利用して簡易 3 次元表示を行うアプリケーションも開発した。今後は、Sturm を用いている境界部分計算の高速化を行う必要がある。

参 考 文 献

- [1] 近藤, 三好, 齋藤, 陰関数描画に関する一つの試み, 数理解析研究所講究録, vol.920, pp.168-174, 1995.
- [2] 近藤, 三好, 齋藤, Cell Plotting, 数式処理学会大会報告, 数式処理, vol.5, No.1, pp.52-53, 1996.
- [3] 齋藤, 三好, 近藤, ifplot アルゴリズム, 数理解析研究所講究録, vol.941, pp.223-228, 1996.

- [4] Saito, T., An extension of Sturm's theorem to two dimensions, Proceedings of the Japan Academy to appear.