数理解析研究所講究録
第1907巻 2014年 166-173

ifplotでの3次元描画の拡張

兵頭礼子
サレジオ高専
NORIKO HYODO
SALESIAN POLYTECHNIC

近藤祐史
香川高専
YUJI KONDOH
KAGAWA NATIONAL COLLEGE OF TECHNOLOGY

村尾裕一
電気通信大学
HIROKAZU MURAO
THE UNIVERSITY OF ELECTRO-COMMUNICATIONS

齋藤友克
アルファオメガ
TOMOKATSU SAITO
ALPHAOMEGA INC.

1 始めに

1.1 ifplotの現状

過去20年以上に渡りRisa/Asirのplot関連のプログラムには手が入っていない．新しい機能を付け加えることのみであった．関数の部分的な，バグ対応はあったが，全体としての見直しはしてこなかった．

その結果，関数群の間に不整合を生ずる場合も存在する状況であった．また，バグが存在することが確認されていても，修正が困難な状況であり修正できない場合も存在する．

さらに，3D-printerなどの新しいデバイスが普及してきたことにより，その対応をすることが必要と考えている．しかし，ifplotに3次元描画への拡張を付け加えるためには，単純な指標関数追加では対応ができない状況であることが現状である．そこで過去の追加も含め，全体の機能を整理し見直す必要があるとの結論にいたった．

1.2 方針は変えない

Risa/Asirの描画機能であるifplotは，1992年の一般に向けての発表以来[1]本質的に変化していない．これは，ifplotの描画理論[2, 3, 4]の変更がないことを意味している．著者らは，この描画理論の変更は必要ないと現在も考えている．

ifplot描画基本方針

- 空間をCellと呼ばれる領域Cに分割して描画はそのCellの選択により表示する
- 選択はCharacterと呼ばれる指標関数χにより定める

ifplotの変更は，この理論で提示された指標関数の追加にともなう修正である．例えば，区間数による描画，不等式の正領域の描画など新たな指標関数に対応する関数は適時追加していた．

指標関数の定義

Definition 1 (指標関数の定義)
D 上定義されている関数 f の *Cell* の族 $\{C_k\}$ による指標関数 χ とは,

1. $\chi : \{C_k\} \to \{0, 1\}$
2. $\chi(C_k) = 0$ ならば C_k の任意の元 x に対し $f(x) \neq 0$

とする.

2 3次元描画

2.1 3D-描画の実装の方針

1. 空間を Cell に分割する
2. 描画する Cell の選択は指標関数により定める
3. $\chi(C) = 1$ となる Cell ををを選択して空間に配置する
   デバイスの特徴により空間に自由に Cell は配置できない

この方針は 2D の陰関数描画とほぼ同じ方針である. ただし Cell の配置に制限がかかる点が異なっている.

しかし, 作成いらい年月が経過し追加作業は徐々に困難になってきた. そのため, 3D に対応する機能を導入するよう拡張は大変難しい状況である. そこで, 今回 Risa/Asir の描画関数を全面的に見直す作業を始めた. 今回の報告は, 3 次元描画に向けての Risa/Asir の ifplot 関数群の見直しの経過報告である.

2.2 配置できないもの, できるもの

3D-printer の構造上 2D の場合と異なるものは次の点である.

- 下に凸な物は配置できない (何らかのサポートが必要)
- 接する cell が存在すれば置くことができる
- 現時点では出力された物は不安定である

この解決方法としては, 天井と床を常に考慮して床から積み上げる. また, 垂直方向への区分的な描画をすることによりこの問題を回避することができる.

3 既存の関数群の問題点

ifplot の作成以来 20 年以上の時がが経過すると様々な不都合が顕在化している. このため描画が正しくない場合が多々報告されている.

3.1 実装上の問題

例えば $(x-y)^{10}$ を ifplot で表示すると図 3 ようになる (参照 [5]). これは, 描画のスピードを上げるため一部の計算に浮動小数点演算を利用していることに起因する誤差である. 当然この場合は $(x-y)^{10}$ を既約分解して $(x-y)$ の描画をすれば正しい図となる. もしくは, precise モードで描画をすれば $(x-y)^{10}$ のままでも正しい図となる. しかし, この方法では, 常にひと手間事前処理が必要となる. また利用者が何が起きて図が乱れているかを識別する能力が必要である. この手間を軽減するよう改良する.

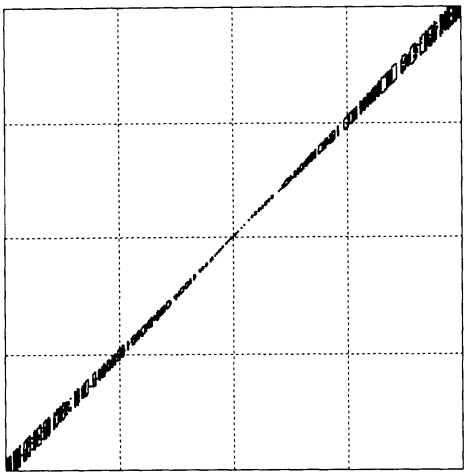

図 1: ifplot による $(x-y)^{10}$ の図

3.2 現在の描画関数

現在 Risa/Asir で用意されている描画関数の名称と機能は次のようになっている.

plot 1 変数関数の零点の描画

ifplot 2 変数多項式の零点の描画

ineqn 2 変数不等式 (多項式) の正領域の描画

ineqnand 既存の描画に 2 変数不等式 (多項式) の領域を and で重ねる

ineqnor 既存の描画に 2 変数不等式 (多項式) の領域を or で重ねる

ineqnxor 既存の描画に 2 変数不等式 (多項式) の領域を xor で重ねる

itvifplot 区間数による 2 変数多項式のゼロを跨ぐ区間の描画

conplot 2 変数関数の実数上での等高線を表示する (図 6 参照)

polarplot 極形式による関数の描画 (図 7)

plotover すでに存在しているウィンドウへ描画する (図 2 参照)

open_canvas 描画用ウィンドウ (キャンバス) を生成

clear_canvas 描画用ウィンドウ (キャンバス) をクリアする

draw_obj キャンバス上に点または線分を描画する

draw_string キャンバス上座標 [X,Y] に文字列 G を指定色で描画する

drawcircle キャンバス上指定した場所に円盤を描く

obj_cp 描画用ウィンドウを他の描画ウィンドウに指定方法でコピーする

となっている. これらは内部でどのように計算しているかは判別しがたい名称となっている. この機能を整理する.

3.3 内部計算方式をを明確にする

内部の計算方式を関数名称から明確に区別出きるようにする．この変更は，関数の値の評価と判定の部分の変更である．

浮動小数点計算評価 関数名の末尾に D をつける

有理数計算評価 関数名の末尾に Q をつける

有理数計算 +sturm 法評価 関数名の末尾に B をつける

この変更により関数名の前半部分が指標関数の区別となり最後の文字がその関数の評価の為の計算方式となる．

ただし `itvifplot` に関しては，区間数の性質に鑑み端点を有理数のみの実装とするため D のみの実装となることより末尾には何も付けない．

precise モードは廃止する．

3.4 付加機能

その他の機能としては，すでに実装されているものもあるが新規の関数群には次の機能を付加する．

- 3D 関連も見据えグラフの描画に色指定を直接するようにする．
- グラフ間の重ねあわせを可能とする

また，互換性のため既存の関数は名称，引数を含めそのまま存在させる．そこで新たな枠組みによる関数群を創設することとする．

3.5 ネットワーク依存の排除

3D-printer による出力を考慮すると内部的に ox-asir の描画は少々問題がある．

ox を使った場合別空間での処理となり通常は利点があるが，今回の描画関数は小さな処理のやりとりを繰り返して全体を構成する必要がある．

そこで新規の関数群は tcp を利用した従来の構造を廃止し，直接 Risa/Asir から描画するようにする．この変更は，3D を描画する機能は本体とのデータのやりとりが発生する可能性が高いためネットワークを経由しない方式とする．[1)]この変更は，ネットワークの構造をもっているとその安全性を保証するための処理が増加する傾向にあることも留意している．

4 新規関数群

4.1 実装の状況

2013 年末の段階で，新規関数群は tcp の機能を削除することをを除いて実装されている．この変更は Risa/Asir のレポジトリに反映されている．利用する為には，`INTERVAL` の宣言を付けて Risa/Asir を make する必要がある．

[1)] windows 版はもともと tcp 経由ではない

4.2 関数名称

今回の更新により次の関数群が追加整備された．しかし，現時点では tcp を利用した構造になっているが最終的には tcp を排除した構造になる予定である．

tcp を排除した構成になっていないため現在の状態では次の関数は修正されていない．open_canvas, clear_canvas, draw_obj, draw_string, rawcircle, obj_cp 新規関数群が tcp の機能を削除した段階で名称も含め新たな関数に修正される予定である．

従来の名称	浮動小数点計算	有理数計算	有理数計算 +sturm
ifplot	ifplotD(図 3 参照)	ifplotQ(図 4 参照)	ifplotB(図 5 参照)
plot	plotD	plotQ	plotB
ineqn	ineqnD(図 11 参照)	ineqnQ(図 12 参照)	ineqnB(図 13 参照)
ineqnand	ineqnandD	ineqnandQ	ineqnandB
ineqnor	ineqnorD	ineqnorQ	ineqnorB
ineqnxor	ineqnxorD	ineqnxorQ	ineqnxorB
itvplot	itvplot	–	–
conplot	conplotD(図 8 参照)	conplotQ(図 9 参照)	conplotB(図 10 参照)

4.3 従来の関数の引数

従来の関数の引数は，

```
FUNC [,GEOMETRY] [,XRANGE] [,YRANGE] [, ZRANGE] [,ID] [,NAME]
```

となっていた．その意味は

FUNC 多項式である
　但し plot は Risa/Asir の解析関数を含む多項式でもよい

GEOMETRY リストである形式はウィンドウのサイズをドット単位で表す

XRANGE 変数の範囲の指定で, [変数, 最小値, 最大値] で指定する

YRANGE 変数の範囲の指定で, [変数, 最小値, 最大値] で指定する

ZRANGE textttconplot のみ．変数の範囲の指定で, [変数, 最小値, 最大値 [, 刻み巾]] で指定する．刻み巾は指定がない場合は 16 を使う．新しい ifplot は，表示領域の最大最小を求めその内部を 32 等分する．

ID tcp によるプロセスの番号

NAME 描画されたキャンバスの名称．キャンバスに対する処理をする場合キャンバスを特定する為の番号
　ウィンドウの名称は NAME:N/M となる N は tcp によるプロセス番号 M はウィンドウ番号である．

順番は随意である．

4.3.1 新規関数の引数

新規関数群の引数は，変数カテゴリにより区分されている．その順番はカテゴリ内部では以下の順番になっている．

関数 1 変数か 2 変数の多項式を記述する．但し plot は有理多項式も許す．また Risa/Asir の内部関数の解析関数を含んでもよい．

数値　色，ID(従来の ID と同様)，IDX(従来のウィンドウ番号), 分割数が数のカテゴリにより想定されている．分割数は現時点では区間演算の `itvplot` のみである．tcp を廃止下場合は ID が削除される．
リスト　`XRANGE`，`YRANGE`，`ZRANGE`，`GEOMETRY` を表すリストである．
XRANGE，YRANGE のリストの先頭には，FUNC の変数が入っていなければならない，長さはおのの 3 であることを想定している．
ZRANGE はダミーの変数がリストの先頭に入っている言で判定している．長さは 3 もしくは 4 であることを想定してる．
EOMETRY は長さ 2 の数のリストを想定している．
どの引数も省略可能である．その場合は，`[関数の変数,-2,2]` とされる．GEOMERY は `[300,300]` である．
文字列　ウィンドウ名称をつける場合の文字列である．

4.3.2　plotover の修正

以前より存在する `plotover` に色コードを指定できるようにした．

```
[10] ifplotD(H,[500,500]);
[11] plotover(C,0,0,Gray60);
[12] plotover(D,0,0,Gray30);
```

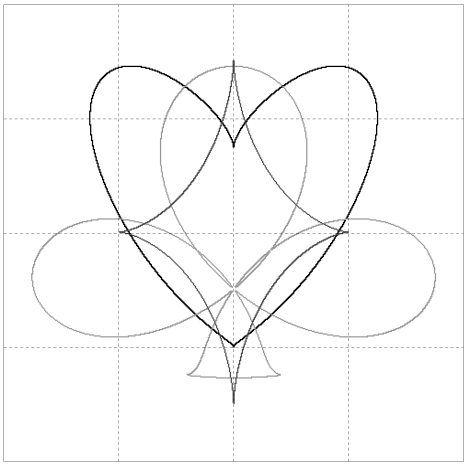

図 2: `plotvoer`

4.3.3　ifplot の修正

ifplot に有理数演算や sturm を付け加えた．

4.3.4　等高線表示関数の修正

等高線表示をする関数 `conplot` に対しても ifplot と同様の修正を加えた．`conplot{H,[z,-3000,3000,32]}` ここで `z` はダミー変数であり `H` に存在しない変数であればよい．特にステップの上下限を指定しない場合は，その領域における最大最小値を求め 16 等分して表示するよう変更した．

4.3.5　不等式の表示関数の修正

不等式表時関数の感数名から浮動小数展，有理数計算の区別を明確にした．

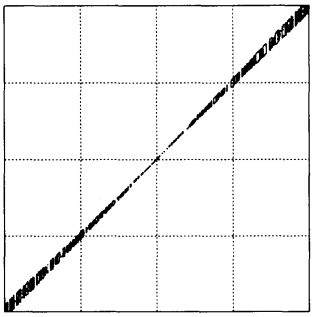

図 3: ifplotD

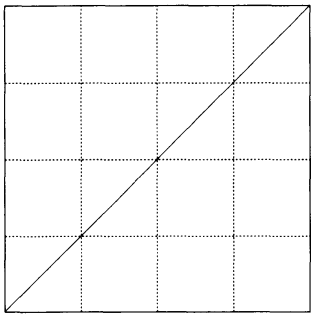

図 4: ifplotQ

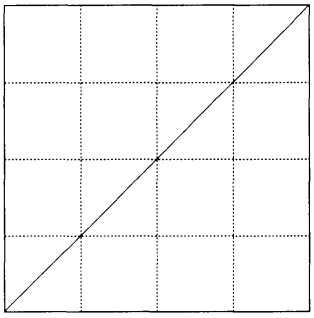

図 5: ifplotB

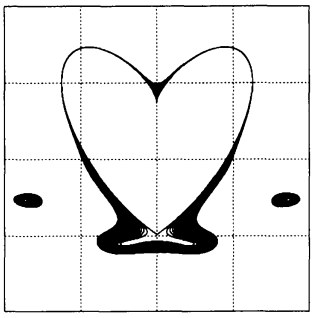

図 6: conplot

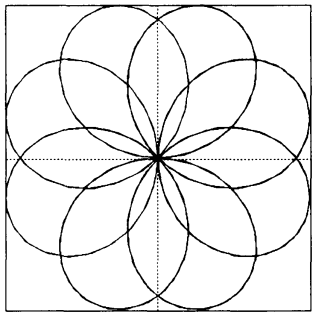

図 7: polarplot

4.3.6 極形式の表示

極形式での関数の表示は，陰関数形式では存在しない．現在実装されているものは plot と同様の機能である．実行例で polarplot(sin(4/3*t),[t,0,50])(図 7 参照) である．

参 考 文 献

[1] Noro,M.,Takeshima,T.: Risa/Asir – A Computer Algebra System, *Proceedings of ISSAC '92*, New York, 1992, pp. 387–396

[2] Saito,T.: An extension of sturm's theorem to two dimensions. *Proc. the Japan Academy*, Vol.73 A pages 18–19, 1997.

[3] 齋藤友克, 近藤祐史, 三好善彦, 竹島卓: Displaying real solution of mathematical equations. **数式処理**, Vol.6 pages 2–21, 日本数式処理学会, 1998.

[4] 齋藤友克, 近藤祐史, 三好善彦, 竹島卓: Faithful Plotting of Real Curves Defined by Bivariate Rational Polynomials **情報処理学会論文誌**, Vol.41(4) pages 1009–1017, 情報処理学会, 2000.

[5] 兵頭礼子, 近藤祐史, 村尾裕一, 齋藤友克: ifplot の改良**数式処理**,to appear.

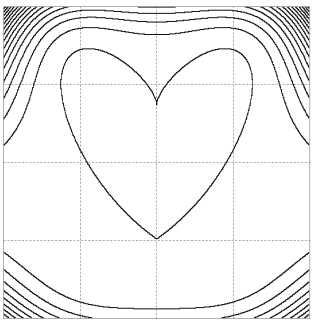

図 8: conplotD(H)

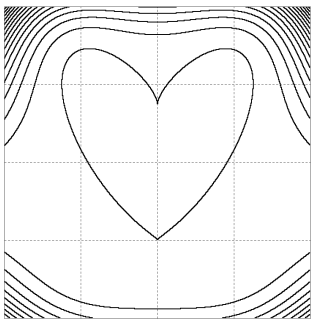

図 9: conplotQ(H)

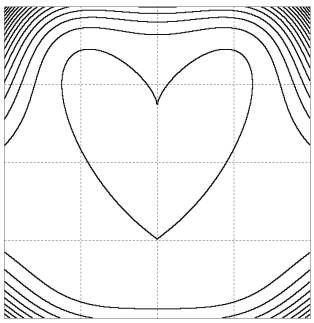

図 10: conplotB(H)

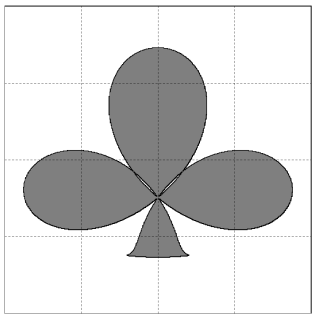

図 11: ineqnD(-C,Gray50)

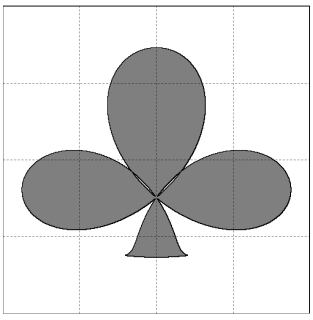

図 12: ineqnQ(-C,Gray50)

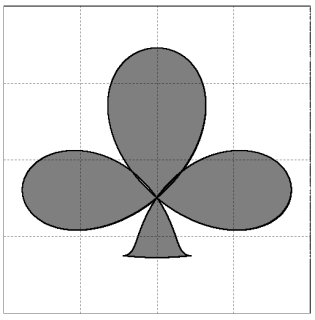

図 13: ineqnB(-C,Gray5)0