

近似代数を用いた代数曲面の描画方法

武庫川女子大学 生活環境学部 福井哲夫 (Tetsuo Fukui)

1. はじめに

曲面の描画はコンピュータグラフィックスや科学技術計算の結果の視覚化に応用されている。

描画方法は数値計算による手法が中心で、曲面上の有限なサンプリング点列から三角形あるいは多角形分割された平面のつなぎ合わせによる方法が代表的である [1]。その他、球面のようななめらかさを表現するために関数近似や多項式補間を行う方法などが考えられている [2]。

関数曲面描画特有の問題としては、(i) 隠線処理、(ii) 等高線描画、(iii) エッジ(境界) 曲線の描画、(iv) 描画速度の問題などが挙げられる。(ii)、(iii) の問題は陰関数描画の問題に帰着される [2]。

高さに関して 2 変数一価関数で与えられる曲面に対しては数値計算によるアルゴリズムがよく知られているが、一般の陰曲面については個別に解析して処理しているのが実状である [3]。特に、特異点を含むような場合は正確に描画することは難しい。

詫間電波高専の福間および筆者は 1993 年に、特異点を含むような一般の代数曲線の描画に際し、近似代数 [4] を用いた手法が有効であることを示した。そこで、これを 3 次元の場合に発展させて曲面の描画を試みた。代数曲面の描画は近似代数の応用として興味深い問題である。以下では、近似代数を用いた代数曲面の描画アルゴリズムを提案し、実際にインプリメントした結果を報告する。

2. 代数曲線 (2 次元) の描画

代数曲線を描画する上で問題となるのが、特異点および特異点の近傍の処理である。代数曲線が次のような 2 変数多項式方程式で与えられているとき、

$$f(x, y) = 0 \quad (1)$$

$$x_{\min} \leq x \leq x_{\max}, \quad y_{\min} \leq y \leq y_{\max} \quad (2)$$

従来の数値計算による描画方法は領域 (2) に $N_x \times N_y$ の画素に分割した格子を考え、(1) の条件を満たす、あるいは近似的に満たす点をプロットすることによって描画する方法を採用していることが多い。条件 (1) を調べる最も簡単な方法は、隣り合う格子点の $f(x, y)$ の値が異符号のとき、一方の点をプロットする方法である。このステップを全画素 $N_x \times N_y$ 回繰り返せば代数曲線が完成する [2]。

しかし、このような方法では孤立特異点や特異点の近傍は正しく描画されない場合がある。

代数曲線 (1) の特異点は条件

$$\begin{aligned} f_x(x, y) &= 0 \\ f_y(x, y) &= 0 \end{aligned} \quad (3)$$

を満たす点として定義され、特異点以外の点は通常点と呼ばれる。特異点の近傍以外の通常点は先の数値計算による方法で容易に描画される。したがって、一般に代数曲線は全ての特異点および極値点の座標が解れば、その近傍は精度を上げて処理することによって完全に描画できる。

そこで、福岡および筆者は特異点および極値点を求めるアルゴリズムとして近似代数を用いた方法を 1993 年 11 月本研究会にて提案した。ここでは概要を解説する。

2.1. 特異点および極値点を求めるアルゴリズム

簡単のために、無平方で 1 変数因子を含まない浮動小数係数の 2 変数多項式 $f(x, y)$ を考える。このとき特異点および極値点は

$$\begin{aligned} f(x, y) &= 0 \\ f_x(x, y) &= 0 \end{aligned} \quad (4)$$

を満たす有限個の解として求めることができる。これを近似代数によって求めるアルゴリズムを以下に示す。

アルゴリズム 1 特異点および極値点の座標計算

入力： 無平方で 1 変数因子を含まない浮動小数係数多項式 $f(x, y)$

出力： 特異点および極値点の座標

Step.1 $P_1 = f(x, y), P_2 = f_x(x, y), k = 2$

Step.2 P_k が x を含まないならば Step.4 へ

Step.3 $P_{k+1} = \text{roundoff}(\text{prem}(P_{k-1}, P_k), e), k = k + 1, \text{Step.2 へ}$

Step.4 $\text{RealSolve}(P_k = 0, \delta) \Rightarrow \{y_1, y_2, \dots\}$

Step.5 各解 y_i について $\text{RealSolve}(\text{app_GCD}(f(x, y_i), f_x(x, y_i), \epsilon) = 0, \delta) \Rightarrow \{x_{i1}, x_{i2}, \dots\}$

Step.6 $\text{return} \{(x_{11}, y_1), (x_{12}, y_1), \dots\}$

ここで、 $\text{roundoff}(f, e)$ は多項式 f の e 以下の係数を持つ項を切り捨てる関数である。 e はマシン誤差程度に取っている。また、 $\text{prem}(P, Q)$ は多項式 P の Q に対する擬剰余を求める関数である。 $\text{app_GCD}(f, g, e)$ は精度 e の近似 GCD である [5]。 $\text{RealSolve}(f = 0, \delta)$ は 1 変数高次方程式 $f = 0$ の近似実数解を求める関数であり、以下のように定義される。

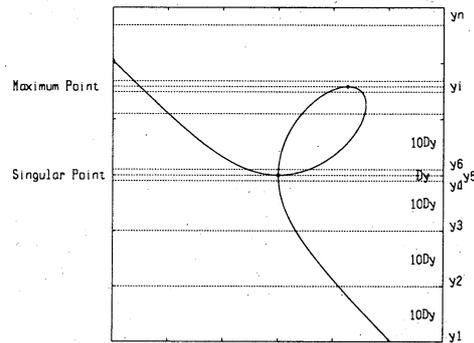


Fig. 1. 曲線描画の y 軸スライス基点

アルゴリズム 2 1 変数高次方程式の実数解：RealSolve

入力：浮動小数係数多項式 $P(x)$, 近似パラメータ δ

出力：近似実数解 (+ 近似多重度)

Step.1 精度 δ^2 の近似無平方分解 [5] (スケール変換を含む)

$$P(x) = Q_1^1 Q_2^2 \dots Q_k^k$$

Step.2 $Q_j(x) = 0$ を DKA 法 [6] で解いた実数解を $\{x_{j1}, x_{j2}, \dots\}$ とする。また、このときの近似多重度を j とする。ただし、DKA 法による誤差はスミスの定理 [7] を用いて δ 以下になるようにし、実数の判定条件は

$$|\text{Im}(z)| \leq \delta$$

を用いた。

Step.3 return $\{\dots, (x_{jm}, j), \dots\}$ 近似実数解 (+ 近似多重度)

2.2. 代数曲線描画アルゴリズムの概要

アルゴリズム 1 によって得られた特異点および極値点の y 座標を含む数列 $\{y_1 (= y_{\min}), y_2, y_3, \dots\}$ を作り、各 y_i に対し $f(x, y_i) = 0$ を *RealSolve* によって解き、各 $y = y_i$ 上にある曲線上の点を得る。次に、 $y = y_i$ と $y = y_{i+1}$ 上の点どうしをそれぞれの点の多重度をもとに直線をつないで描画する。数列 y_i の選び方は特異点および極値点の近傍では画素幅 D_y 、その他の通常点では $10D_y$ とした (Fig.1)。

3. 代数曲面 (3次元) の描画

3.1. 曲面描画方法の概要

3変数多項式方程式

$$f(x, y, z) = 0 \quad (5)$$

で表された代数曲面の描画領域

$$[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}] \quad (6)$$

内における描画を考える。描画解像度は単純に1辺がN画素の立方体で表現するものとする。このとき画素幅は

$$D_x = \frac{x_{\max} - x_{\min}}{N}, \quad D_y = \frac{y_{\max} - y_{\min}}{N}, \quad D_z = \frac{z_{\max} - z_{\min}}{N} \quad (7)$$

となる。

z を描画領域 (7) 内のある値 z_0 で固定すれば $f(x, y, z_0) = 0$ は2変数多項式方程式となり、 $z = z_0$ 平面上の代数曲線を表す。したがって、2章で述べたアルゴリズムがそのまま適用できる。このとき z_0 を z 軸方向のスライス点と呼び、曲線 $f(x, y, z_0) = 0$ をスライス曲線と呼ぶことにする。

スライス曲線を3次元表示するためには、描画点 (x, y, z) の2次元視平面座標系への射影変換が必要であるが、これについては文献 [8] に基づく数値計算によって行う。

さて、この代数曲面の描画方法で一番問題となるのが、2次元と同様に特異点 (曲線) およびその近傍の処理である。さらに、球のように z 軸方向に有界な曲面では、ある $z = z_0$ 平面と接する境界点 (以後、 z 極値点と呼ぶ) も求める必要がある。これら、特異点の z 座標あるいは特異曲線の z 軸方向の境界点および z 極値点を総称して z 軸方向のスライス基点と呼ぶことにする。スライス基点を完全に求めることは、特に特異曲線の問題を含んでいるので困難が予想される。そこで2次元の場合の近似代数によるアルゴリズムを拡張して z 軸方向のスライス基点の部分集合を“近似的”に求めるアルゴリズムを考えた。これを次節で述べる。

z 軸方向のスライス基点が求めれば代数曲面の描画は容易である。隠線処理については、描画点と視点を結ぶ直線と曲面との交点を判定することによって可能となる。具体的方法は後で述べる。

[代数曲面の描画アルゴリズム]

入力： 3変数多項式 $f(x, y, z)$,

描画領域 (6),

解像度 N

視点 (x_L, y_L, z_L)

出力： 曲面描画図形

Step.1 z 軸方向のスライス基点 z_i の計算

Step.2 z 列 $\{z_j\}$ (基点の近傍は D_z 、以外は $10D_z$ 間隔)

に対し、スライス曲線 $f(x, y, z_j) = 0$ の描画を繰り返す。

3.2. z 軸方向のスライス基点計算

必要ならば無平方分解等を行えばよいので、簡単のために、 $f(x, y, z)$ を 2 変数以下の因子を含まない無平方多項式とする。このとき代数曲面 (5) の特異点は

$$\begin{aligned} f_x(x, y, z) &= 0 \\ f_y(x, y, z) &= 0 \\ f_z(x, y, z) &= 0 \end{aligned} \quad (8)$$

で定義される。描画のためには z 軸方向のスライス基点を求める必要があるため、方程式 (8) ではなく、

$$\begin{aligned} f(x, y, z) &= 0 \\ f_x(x, y, z) &= 0 \\ f_y(x, y, z) &= 0 \end{aligned} \quad (9)$$

の解を考える。ここでは、スライス基点 (9) の解の部分集合を“近似的”に求めるための次のようなアルゴリズムを提案する。

アルゴリズム 3 z 軸方向のスライス基点計算

入力： 無平方で 2 変数因子を含まない浮動小数係数多項式 $f(x, y, z)$,
近似パラメータ e, δ, ϵ

出力： z 値配列 Z

Step.1 $Z = \{\phi\}$

Step.2 $rfx(y, z) = \text{Resultant}(f(x, y, z), \frac{\partial f}{\partial x}, x, e)$
 $rfy(y, z) = \text{Resultant}(f(x, y, z), \frac{\partial f}{\partial y}, x, e)$

Step.3 $rfxy(z) = \text{Resultant}(rfx(y, z), rfy(y, z), y, e)$

Step.4 もし、 $rfxy(z)$ が z を含まないならば、Step.6 へ

Step.5 $\{z_1, z_2 \dots\} = \text{RealSolve}(rfxy(z) = 0, \delta)$,
 $Z = Z + \{z_1, z_2 \dots\}$,
return Z

Step.6 もし、 $rfxy < e$ ならば return $\{\phi\}$

Step.7 $G(y, z) = \text{App_GCD}(rfx(y, z), rfy(y, z), \epsilon)$

Step.8 $gfxy(y, z) = \text{App_SquareFree}(G(y, z), e)$

Step.9 $\{z_1, \dots\} = \text{RealSolve}(\text{Resultant}(gfxy(y, z), \frac{\partial gfxy}{\partial y}, y, e) = 0, \delta)$
 $Z = Z + \{z_1, z_2 \dots\}$,

Step.10 $rfx(y, z) = \text{Quotient}(rfx(y, z), G(y, z), y)$
 $rfy(y, z) = \text{Quotient}(rfy(y, z), G(y, z), y)$, Step.3 へ

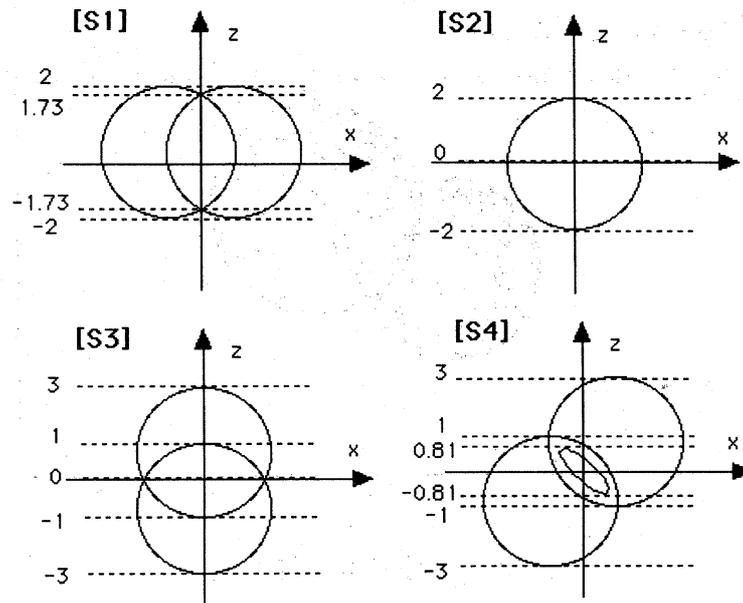


Fig. 2. 代数曲面の x - z 平面射影図と z 軸スライス基点

3.3. z 軸方向のスライス基点計算例

アルゴリズム 3 の適用例として、半径 2 の 2 つの球が重なった 4 つの場合を示す。調べた多項式 f とアルゴリズム 3 の出力結果を列記すると次のようになる。

$$[S1] f = \{(x+1)^2 + y^2 + z^2 - 4\} \{(x-1)^2 + y^2 + z^2 - 4\}$$

出力結果: $Z = -2, -1.7320508, 1.7320508, 2$

$$[S2] f = \{x^2 + (y+1)^2 + z^2 - 4\} \{x^2 + (y-1)^2 + z^2 - 4\}$$

出力結果: $Z = -2, 0, 2$

$$[S3] f = \{x^2 + y^2 + (z+1)^2 - 4\} \{x^2 + y^2 + (z-1)^2 - 4\}$$

出力結果: $Z = -3, -1, 0, 1, 3$

$$[S4] f = \{(x+1)^2 + (y+1)^2 + (z+1)^2 - 4\} \cdot \{(x-1)^2 + (y-1)^2 + (z-1)^2 - 4\}$$

出力結果: $Z = -3, -1, -0.816497, 0, 0.816497, 1, 3$

代数曲面 [S1]~[S4] の x - z 平面への射影図にスライス基点の点線を引いた図を Fig.2 に示す。 z 極値は正確に求まっているため全て描画可能である。特に [S4] では特異曲線 (2 球の交線) に接するスライス基点 (± 0.816497) も得られており、完全な描画が可能である。

3.4. 隠線の判定

視点を (x_L, y_L, z_L) とし、描画点を (x_k, y_k, z_k) とする。ここで $f(x_k, y_k, z_k) = 0$ である。視線の方程式は

$$\frac{x - x_L}{x_k - x_L} = \frac{y - y_L}{y_k - y_L} = \frac{z - z_L}{z_k - z_L} = t \quad (10)$$

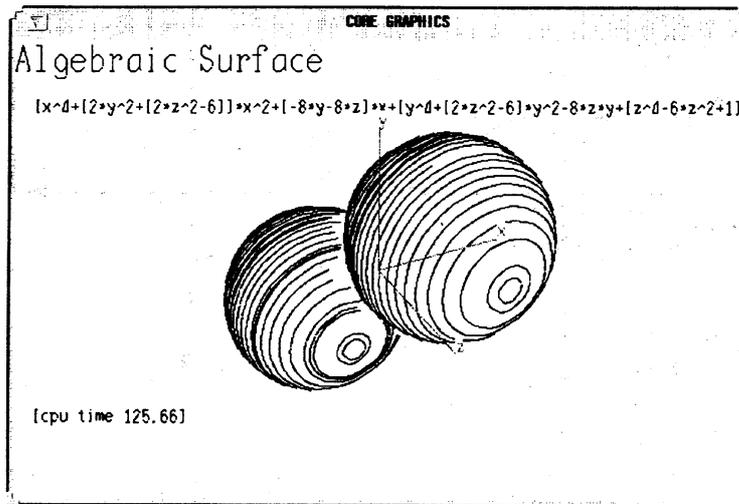


Fig. 3. 曲面 [S4] の描画例

で与えられる。ここで t はパラメータで、(10) 式を f に代入すれば t の多項式

$$\tilde{f}(t) \equiv f((x_k - x_L)t + x_L, (y_k - y_L)t + y_L, (z_k - z_L)t + z_L) \quad (11)$$

が得られる。曲面との交点は方程式 $\tilde{f}(t) = 0$ の解を *RealSolve* で解けばよい。

したがって、 $t < 1$ に解が存在し、かつ、交点 $(x(t), y(t), z(t))$ が描画領域 (6) の内部であれば隠線と判定し描画しない。

4. 描画の実際と近似パラメータ

4.1. 代数曲面描画アルゴリズムのインプリメント

本アルゴリズムを以下の動作環境にインプリメントして動作を確認した。

使用マシン：Sparc Station 5(Solaris2.3), microSPARCII 70MHz

処理系：GNU C++コンパイラ

(グラフィック：3次元コア・グラフィックス Ver.1.0[9])

(C++数式処理：CALTOOL Ver.0.6[10])

実行時間は、例えば曲面 [S4] の場合、描画に CPU Time 125.66(sec)、その内隠線判定にかかった時間は 87.23(sec) であった (Fig.3)。その他、14 種類の 2 次～4 次多項式についてテストした結果、描画時間は CPU Time 12.55～149.31 (sec)、平均 63.23(sec) (隠線判定にかかった割合約 72%) であった。

4.2. スライス面描画の近似パラメータ

ここでは近似パラメータの条件を議論する。

2章で述べたスライス曲線の描画アルゴリズムでは *RealSolve* が重要な役割を果たしており、根間距離 δ 以下の根を重解として返す。したがって、画素の精度で正確に曲面を描画するための近似パラメータ e, ϵ, δ の条件は、画素幅を D として、

$$D \geq \sqrt{\epsilon} \geq \delta \gg e \quad (12)$$

と選べばよい。

4.3. 隠線判定の近似パラメータ

視線と曲面の交点の方程式 $\tilde{f}(t) = 0$ の解 $(x(t), y(t), z(t))$ を根間距離が画素幅 D 以下で分離する条件は、パラメータ t に対する分離パラメータ δ を

$$\delta \leq \min \left(\frac{D_x}{|x_k - x_L|}, \frac{D_y}{|y_k - y_L|}, \frac{D_z}{|z_k - z_L|} \right) \quad (13)$$

と選べばよい。

5. まとめ

本発表をまとめると次のようになる。

- 代数曲面の描画方法として近似代数を用いた z 軸方向の基点によるスライス面描画方式のアルゴリズムを提案した。
- 本アルゴリズムはスライス面の特異点を画素精度で描画でき、近接点の多い曲面描画に強い。
- スライス面描画のための根間分離パラメータは画素幅程度にとり、隠線判定のための方程式を解く分離パラメータは精度を高くして描画する方が安定することを計算実験によって調べた。

今後の課題としては

- 描画安定のための近似パラメータ条件の理論的な分析、
- エッジ(境界)曲線の描画、
- 面塗りによる隠線処理(視平面垂線をスライス軸にとると有効)、
- 特異曲線の描画(近似代数の安定化が不可欠)

などが挙げられる。

参 考 文 献

- [1] 杉原厚吉、“グラフィックスの数理”、情報数学講座 13、共立出版(1995).
- [2] 森正武、“曲線と曲面”、新しい応用数の数学 5、教育出版(1974).
- [3] 岸正倫、“曲面のCG入門”、Information & Computing 66、サイエンス社(1992).
- [4] 野田松太郎、佐々木建昭、鈴木正幸、“数式処理と数値計算の融合による精度保証”、情報処理、31(9)、pp. 1204-1211(1990).

- [5] M.Noda and T.Sasaki, "Approximate GCD and its application to illconditioned algebraic equations", *Journal CAM*, **38**, pp. 335-351 (1991).
- [6] 戸田隼人, "数値計算法", 電子情報通信学会大学シリーズ **A4**, コロナ社 (1981).
- [7] 森正武、室田一雄、杉原正顕 "数値計算の基礎", 岩波講座応用数数学方法 **1**, 岩波書店 (1993).
- [8] 守川穰, "2&3次元グラフィックス・ソフトの基礎と応用", CQ出版 (1984).
- [9] 福井哲夫, "教育・研究用簡易2次元グラフィックスパッケージ-コア・システム-", 詫間電波高専研究紀要 **21**, pp. 75-86 (1993).
- [10] 福井哲夫, "近似代数計算のためのC++による数式処理クラスライブラリーの開発", 数理解析研究所講究録 **848**, 162-176 (1993).