

## 双3次パッチの描画への Aberth 法の適用について

藤本 忠博      大野 義夫

慶應義塾大学 理工学部

レイトレーシング法で双3次パッチを描画するときには、双3次パッチと光線との交点を求める必要がある。この問題を代数的に解決するものとして、交点計算を高次代数方程式(18次方程式)に帰着させる方法がある [1]。

本稿では、この方程式の解法に Aberth 法を適用した場合の効果を調べ、他の解法と比較した場合の利点を検討する。Aberth 法ではすべての解が同時に得られるため、解の誤差が少なく、高精度が要求される場合にも適切な画像が安定して得られることを実証する。

## On the Aberth Method for Rendering Bicubic Patches

Tadahiro Fujimoto      Yoshio Ohno

Faculty of Science and Technology, Keio University  
3-14-1 Hiyoshi, Kohoku-ku, Yokohama-shi, Kanagawa 223, Japan

When a bicubic patch is displayed by ray tracing technique, we have to find an intersection of a ray and a patch. One approach to solve this problem is to convert the intersection problem into an algebraic equation with degree 18.

In this paper, we discuss the effect of Aberth method in solving the equation, and show its advantages compared with other methods. With Aberth method, we can create a correct picture stably even when high accuracy is required, because in this method all roots are obtained simultaneously and the errors are small.

## 1 はじめに

コンピュータグラフィックスにおけるレンダリング技法の1つであるレイトレーシング法で双3次パッチを描画することは、平面ポリゴンや2次曲面などの描画に比べて非常に困難であることが知られている。レイトレーシング法では視点から発した光線と物体との交点計算を繰り返す必要があるが、双3次パッチの場合には、光線、すなわち直線との交点を求めることが容易でない。これまでのアプローチは、

1. 曲面パッチを再分割する方法
2. 直接の数値計算で解く方法

の2つに大別される。1.の方法の適用は比較的容易であるが、計算時間がかかる。一方、2.の方法では代数的な知識が必要とされるが、計算時間が短縮される可能性が高い。

本研究では、2.の方法の1つであり、J. T. Kajiyaによって提案されたアルゴリズム [1] について検討する。このアルゴリズムは、代数的なアプローチとしての初期のものであり、交点計算を18次方程式を解くことに帰着させる。高次方程式の解法には反復法を用いなければならないが、描画の正確さと安定性を保証するため、その近似解には高精度が要求される。本稿では、すべての解を同時に求める Aberth 法の適用による、解および描画の精度の高さを示し、解を1つずつ求める Newton-Raphson 法および Kajiya が用いた Laguerre 法と比較した場合の利点を検証する。

まず、2節で Kajiya のアルゴリズムについて述べる。次に、3節で代数方程式を解くための3つの反復法の特徴と性質を示し、4節で各方法を適用した場合の効果と Aberth 法の利点を述べる。

## 2 交点計算アルゴリズム

### 2.1 Kajiya のアルゴリズム

Kajiya のアルゴリズムを適用することにより、直線と双3次パッチとの交点計算は高次方程式を解くことに変換される。ここでは、Kajiya の交点計算アルゴリズムについて簡単に述べる。

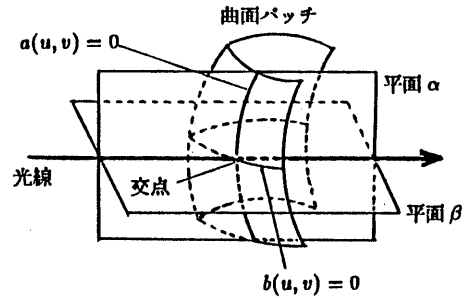


図 1: 2つの代数曲線

双3次パッチは、3次元空間上で2つのパラメータ  $u, v$  の双3次式として次のように定義される。

$$\begin{cases} x(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 p_{ij}^x u^i v^j \\ y(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 p_{ij}^y u^i v^j \\ z(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 p_{ij}^z u^i v^j \end{cases} \quad (1)$$

ただし、 $0 \leq u, v \leq 1$  とする。この双3次パッチと直線との交点を求めるため、直線を2平面の交わりとして次のように定義する。

$$\begin{cases} \alpha_x x + \alpha_y y + \alpha_z z + \alpha_w = 0 \\ \beta_x x + \beta_y y + \beta_z z + \beta_w = 0 \end{cases} \quad (2)$$

(1) 式を (2) 式に代入し同類項をまとめると次式が得られる。

$$\begin{cases} \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} u^i v^j = 0 \\ \sum_{i=0}^3 \sum_{j=0}^3 b_{ij} u^i v^j = 0 \end{cases} \quad (3)$$

(3) 式の2つの方程式は、直線を定義する2平面のそれぞれとパッチとが交差してできる、 $u, v$  を変数とする2つの代数曲線を表わす。そして、この2曲線の交点がパッチと直線の交点の  $u, v$  値となる (図1)。つまり、パッチと直線の交点の  $u, v$  値を求めるためには、(3) 式の共通解 ( $u, v$ ) を求めればよい。

(3) 式を解くために“終結式”を導入する。終結式とは、 $x$  に関する2つの方程式

$$\begin{cases} a(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 = 0 \\ b(x) = b_m x^m + b_{m-1} x^{m-1} + \dots + b_0 = 0 \end{cases} \quad (4)$$

が共通解  $x$  をもつときに0となるような式  $R(a, b)$  をい

い, たとえば, 次の行列式で表せる.

$$R(a, b) = \begin{vmatrix} a_n & a_{n-1} & \cdots & \cdots & a_0 & & & & & \\ & a_n & a_{n-1} & \cdots & \cdots & a_0 & & & & \\ & & & & \vdots & & & & & \\ & & & & a_n & a_{n-1} & \cdots & \cdots & a_0 & \\ b_m & b_{m-1} & \cdots & \cdots & & & & & & b_0 \\ & b_m & b_{m-1} & \cdots & \cdots & & & & & b_0 \\ & & & & \vdots & & & & & \\ & & & & b_m & b_{m-1} & \cdots & \cdots & & b_0 \end{vmatrix} \quad (5)$$

この終結式を用いた (3) 式の解法を, 以下に示す. まず, 2 変数  $u, v$  の双 3 次式を,  $u$  についての多項式を係数とする 1 変数  $v$  の式であるとみなす.

$$\begin{cases} a(u, v) = a[u](v) \\ b(u, v) = b[u](v) \end{cases} \quad (6)$$

そして, これらの係数 ( $u$  の多項式) から終結式  $r(u)$  をつくる. すると, 方程式  $r(u) = 0$  の解が (3) 式の共通解の  $u$  値となることは明らかである.  $r(u) = 0$  の解を求めた後, その値を (3) 式に代入し, ユークリッドアルゴリズムを用いることで共通解の  $v$  値が求められる.

このアルゴリズムは,

- $u, v$  パラメータに複素座標系を適用することで, 計算を形式化できる.
- 低次のパッチに対する処理時間が短縮される.

という利点をもつ.

## 2.2 解の選択

交点の座標として必要なのは,  $0 \leq u, v \leq 1$  の範囲内の実数のみである. しかし, この範囲に限定した解のみを求めることは困難なので, 複素数も含めたすべての解を求めた後に, 必要なものを選択するという方法をとる必要がある. 実際には,  $r(u) = 0$  の解を複素数も含めてすべて求め, その中から  $0 \leq u \leq 1$  の実数となるものを選び, その  $u$  値から  $v$  値を求めることとする.

## 3 高次代数方程式の解法

Kajiya のアルゴリズムでは, 方程式  $r(u) = 0$  を解くことが要求される. これは実数を係数とした最高で 18 次

方程式となるので, 解法には反復法を用いなければならない. より高速かつ正確な描画を実現するためには, 用いるべき反復法における収束時間および精度が問題となる. 本研究では Aberth 法の有効性を検証することを目的とするが, ここで, 代数方程式を解く反復法として 3 つの方法 (Newton-Raphson 法, Laguerre 法, Aberth 法) を取り上げ, その収束時間および精度について検討する.

### 3.1 Newton-Raphson 法

Newton-Raphson 法は, 非線形方程式を解く方法として最もよく知られているものであり, 方程式  $P(z) = 0$  に対する反復公式が次式で表される.

$$z^{(\nu+1)} = z^{(\nu)} - \frac{P(z^{(\nu)})}{P'(z^{(\nu)})} \quad (7)$$

この方法では近似解は 2 次収束するが, 初期値の選び方によっては収束しない場合もある. また, 高次方程式に対する解が 1 度に 1 つしか求まらないため, 1 つの解  $z_i$  を求めた後, 因数  $z - z_i$  によってもとの方程式を割り (減次), 次数の下がった方程式を再び Newton-Raphson 法で解く, という操作を繰り返すことですべての解を求める. ある解が複素数であったときには, その共役複素数も解の 1 つであるはずなので, 複素解が求まったらその共役複素数も解とし, それらの因数の積 (実係数の 2 次式) でもとの方程式を割ることとする.

### 3.2 Laguerre 法

Kajiya は, 代数方程式の解法に Laguerre 法の適用を提案している.

Laguerre 法は Newton-Raphson 法に類似した方法で, その反復は次式で表される.

$$\begin{aligned} z^{(\nu+1)} &= z^{(\nu)} - \frac{nP(z^{(\nu)})}{P'(z^{(\nu)}) \pm \sqrt{H(z^{(\nu)})}} \\ H(z) &= (n-1)[(n-1)P'(z)^2 - nP(z)P''(z)] \end{aligned} \quad (8)$$

Laguerre 法では, いかなる初期値を与えても近似解はその初期値から最も近い根に 3 次収束する. また複素根に対する収束性もすぐれている. しかし, この方法でも解は 1 つずつしか求まらないため, Newton-Raphson 法と同様に減次によってすべての解を求めることになる.

### 3.3 Aberth 法

Aberth 法は科学計算分野でよく用いられる方法であり、 $n$  次方程式  $P(z) = 0$  に対して次式により反復を行なう [2][3].

$$\begin{aligned} z_i^{(\nu+1)} &= z_i^{(\nu)} + \psi_i(z_1^{(\nu)}, \dots, z_n^{(\nu)}) \\ \psi_i(z_1, \dots, z_n) &= -\frac{P'(z_i)/P''(z_i)}{1 - \frac{P'(z_i)}{P''(z_i)} \sum_{j \neq i} \frac{1}{z_i - z_j}} \end{aligned} \quad (9)$$

Aberth 法では、任意の相異なる  $n$  個の複素数を初期値とすることで 3 次収束が保障される。また、Newton-Raphson 法や Laguerre 法には減次による誤差の累積が考えられるが、Aberth 法では同時に  $n$  個の解を求めるのでその欠点がなく、高い精度が期待される。

### 3.4 評価

前節で述べた各解法に対する評価を行なう。

ここで実際の反復回数および計算時間は、係数をランダムに設定した 100 個の方程式についての平均値である。

また、近似解は次式の条件を満たしたところで収束したものとみなすこととした。

$$|z^{(\nu+1)} - z^{(\nu)}| < \epsilon \quad (10)$$

ただし、許容誤差  $\epsilon$  には適値を設定する。

なお、反復の初期値としては“Aberth の初期値”を用いる [2][3]。これは、複素平面上ですべての解の重心を中心とし、すべての解を含むような円を考え、その円周上で等間隔に複素数を選ぶもので、初期値についての適切な情報が得られていない場合に有効である。

#### 3.4.1 次数について

まず、方程式の次数と反復回数および計算時間の関係を調べる。

$n$  次方程式に対する 1 回の反復に要する計算量 (浮動小数点演算回数) は、理論的には表 1 のようになる。そして、これに関する実際の計算時間は図 2 のようになる。次数が高くなるにつれて Aberth 法の計算時間が他のものにくらべて増えていくが、Newton-Raphson 法のほぼ 2 倍程度である。

図 3 および図 4 は、1 つの解を求めるのに要する反復回数および計算時間を示す。この図から、Laguerre 法の収

表 1:  $n$  次方程式に対する 1 回の反復あたりの計算量

演算	Newton 法	Laguerre 法	Aberth 法
加減	$6n + 2$	$9n + 7$	$11n + 4$
乗除	$8n + 4$	$12n + 16$	$12n + 12$
反転	2	1	$n + 1$
絶対値	—	2	—
平方根	—	3	—

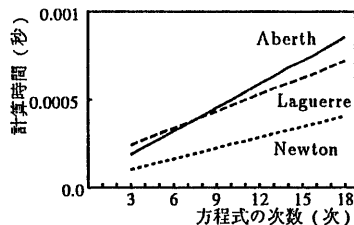


図 2: 1 回の反復あたりの計算時間

束が最も速いことがわかる。Aberth 法の収束時間は Newton-Raphson 法とほぼ同じである。

図 5 はすべての解を求めるための計算時間である。ここでは、Aberth 法の計算時間が他の 2 つにくらべてかなり大きくなっている。これは、Newton-Raphson 法と Laguerre 法では、減次によって次々と低い次数の方程式を解くことになるのに加えて、複素解が求まった場合には次数が 1 度に 2 つ下がるため、方程式を解く回数自体もかなり減るためである。18 次方程式では、Aberth 法は Laguerre 法にくらべて約 10 倍の計算時間を必要とするこ

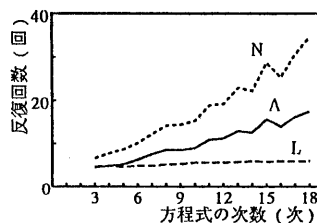


図 3: 1 つの解を求めるための反復回数

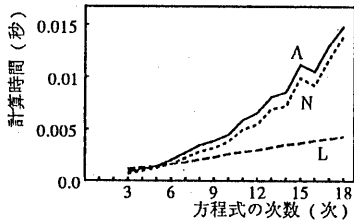


図 4: 1つの解を求めるための計算時間

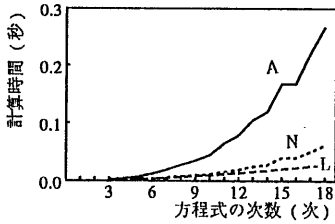


図 5: すべての解を求めるための計算時間

とになる。

なお、ここでは  $\epsilon = 10^{-6}$  とした。

### 3.4.2 許容誤差について

ここでは 18 次方程式のみを取り上げ、許容誤差  $\epsilon$  と反復回数、計算時間および解の精度との関係を調べる。

解の精度については、求められた近似解と真の解との差をはかるのが理想的であるが、ふつう真の解は未知である。そこで、解の精度をはかる方法として、方程式  $P(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n = 0$  の近似解  $z = z_j$  に対して、次式を満たした場合に精度のよい解であるとみなすことにする [4]。

$$|P(z_j)| < \delta(z_j) \quad (11)$$

$$\delta(z) = n \cdot u \cdot \max_i |a_i z^{n-i}|$$

ここで、 $\beta$  進法で仮数部  $s$  桁の浮動小数点演算の場合、 $u = \beta^{-(s-1)}$  とおく。

図 6 から図 8 は、1 つの解を求めるのに要する反復回数、計算時間および計算をした方程式のうち  $P(z) < \delta$  が

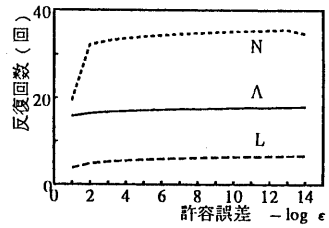


図 6: 1 つの解を求める場合の許容誤差  $\epsilon$  と反復回数との関係

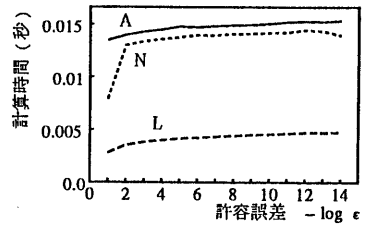


図 7: 1 つの解を求める場合の許容誤差  $\epsilon$  と計算時間との関係

満たされたものの割合の許容誤差に対する関係である。図 8 より、Aberth 法の精度が最も高いことがわかる。

図 9、図 10 は、すべての解を求めるのに要する計算時間、 $P(z) < \delta$  が満たされる割合である。図 10 から、Newton-Raphson 法と Laguerre 法では減次による誤差の累積のため、かなり精度が低くなることがわかる。

これに関して、Newton-Raphson 法と Laguerre 法における減次による誤差の累積を調べたのが図 11、図 12 である。ここでは、 $z_1$  から  $z_{18}$  の順で解を求めており、後で求められた解の精度が低くなることがわかる。これにより、減次による誤差の累積が実際に生じることが確かめられた。

## 4 双 3 次パッチの描画

以上の 3 つの解法を実際に Kajiya のアルゴリズムに適用して双 3 次パッチの描画を行なった場合の、それぞれの効果について検討する。

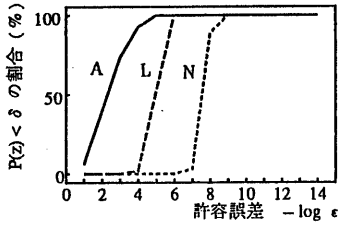


図 8: 1つの解を求める場合の  
許容誤差  $\epsilon$  と  $P(z) < \delta$  である解の割合との関係

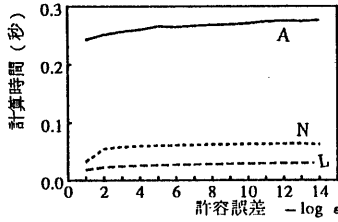


図 9: すべての解を求める場合の  
許容誤差  $\epsilon$  と計算時間との関係

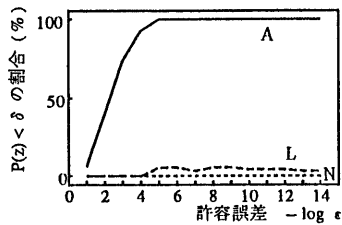


図 10: すべての解を求める場合の  
許容誤差  $\epsilon$  と  $P(z) < \delta$  である解の割合との関係

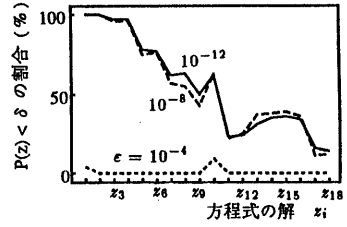


図 11: 減次による誤差の累積 (Newton-Raphson 法)

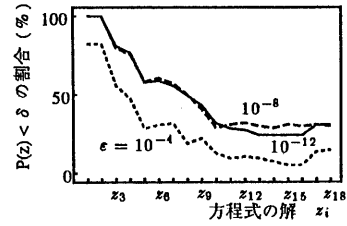


図 12: 減次による誤差の累積 (Laguerre 法)

“Aberthの初期値”は初期値の情報がないときには有効であるが、レイトレーシング法では隣接光の交点どうしは接近している可能性が高いため、初期値として前の光線に対する解を与えることで、収束性の向上が期待できる。

図 16の双3次パッチを描画したときの1ピクセルあたりの1つの解を求めるための平均反復回数, 1ピクセルあたりの平均計算時間, 全ピクセルについて  $P(z) < \delta$  を満たしたピクセルの割合は, 許容誤差  $\epsilon$  に対してそれぞれ図 13, 図 14, 図 15 のようになる。

図 6と図 13および図 9と図 14をくらべると, 実際の描画では反復回数および計算時間が非常に減少していることがわかる。これは, 初期値として前の光線に対する解を与えたためであり, 特に Aberth 法における計算時間の減少が著しい。Laguerre 法のほぼ2倍程度にまで短縮している。

また図 15より, Aberth 法における解の精度が他の2つの方法に比べて非常に高いことがわかる。

さらに, 図 16, 図 17の画像から, Aberth 法における解の精度の高さを検証することができる。図 16において,

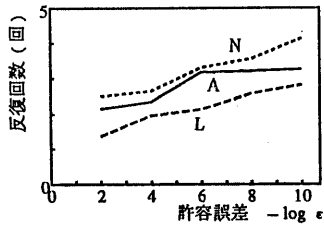


図 13: 許容誤差  $\epsilon$  と 1 ピクセルあたりの 1 つの解を求めるための平均反復回数との関係

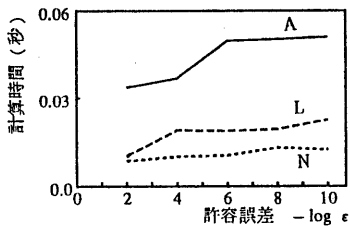


図 14: 許容誤差  $\epsilon$  と 1 ピクセルあたりの平均計算時間との関係

Aberth 法では  $\epsilon = 10^{-2}$  程度の許容誤差でもほぼ正確な画像が得られているのに対して、他の 2 つの方法では正確な計算結果が得られずに、パッチに穴があいてしまっているところが見られる。特に, Newton-Raphson 法ではかなりの計算誤差が見られる。さらに, 図 17 は図 16 の 1 部分を拡大して描画を行なったものであるが, 高い精度が要求されるため, Laguerre 法ではその精度が満たされず, 正確な画像が得られない。このように, 実際の画像からも Aberth 法の精度の高さが検証できる。

## 5 まとめ

レイトラシング法で双 3 次パッチを描画するための交点計算を高次代数方程式に帰着させた方法において, その解法に Aberth 法を適用した場合, 高精度が要求されるときにも, 適切な画像が安定して得られることが実証された。

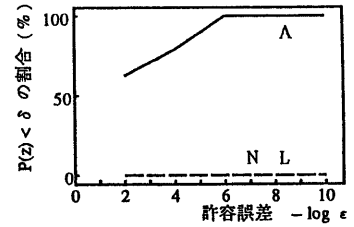
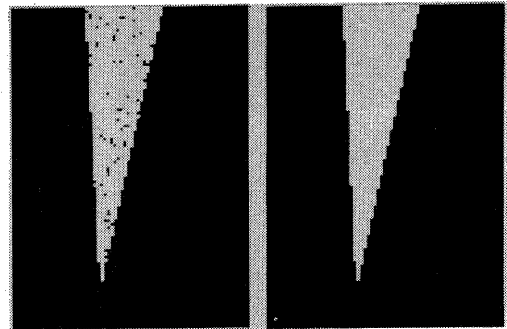


図 15: 許容誤差  $\epsilon$  と全ピクセルにおける  $P(x) < \delta$  である解の割合との関係



Laguerre 法

Aberth 法

図 17: 図 16 の双 3 次パッチの拡大図 (200 倍) (許容誤差  $\epsilon = 10^{-10}$  に設定)

## 参考文献

- [1] James T. Kajiya  
"Ray Tracing Parametric Patches"  
SIGGRAPH 1982, 245-254.
- [2] 山本 哲朗  
"ある代数方程式解法と解の事後評価法"  
数理科学, 1976 年 7 月, 52-57.
- [3] 伊理 正夫, 山下 浩, 寺野 隆雄, 小野 令美  
"大域的収束性をもつ代数方程式の解法"
- [4] 伊理 正夫, 藤野 和建  
"数値計算の常識"  
共立出版, 1985 年 4 月.

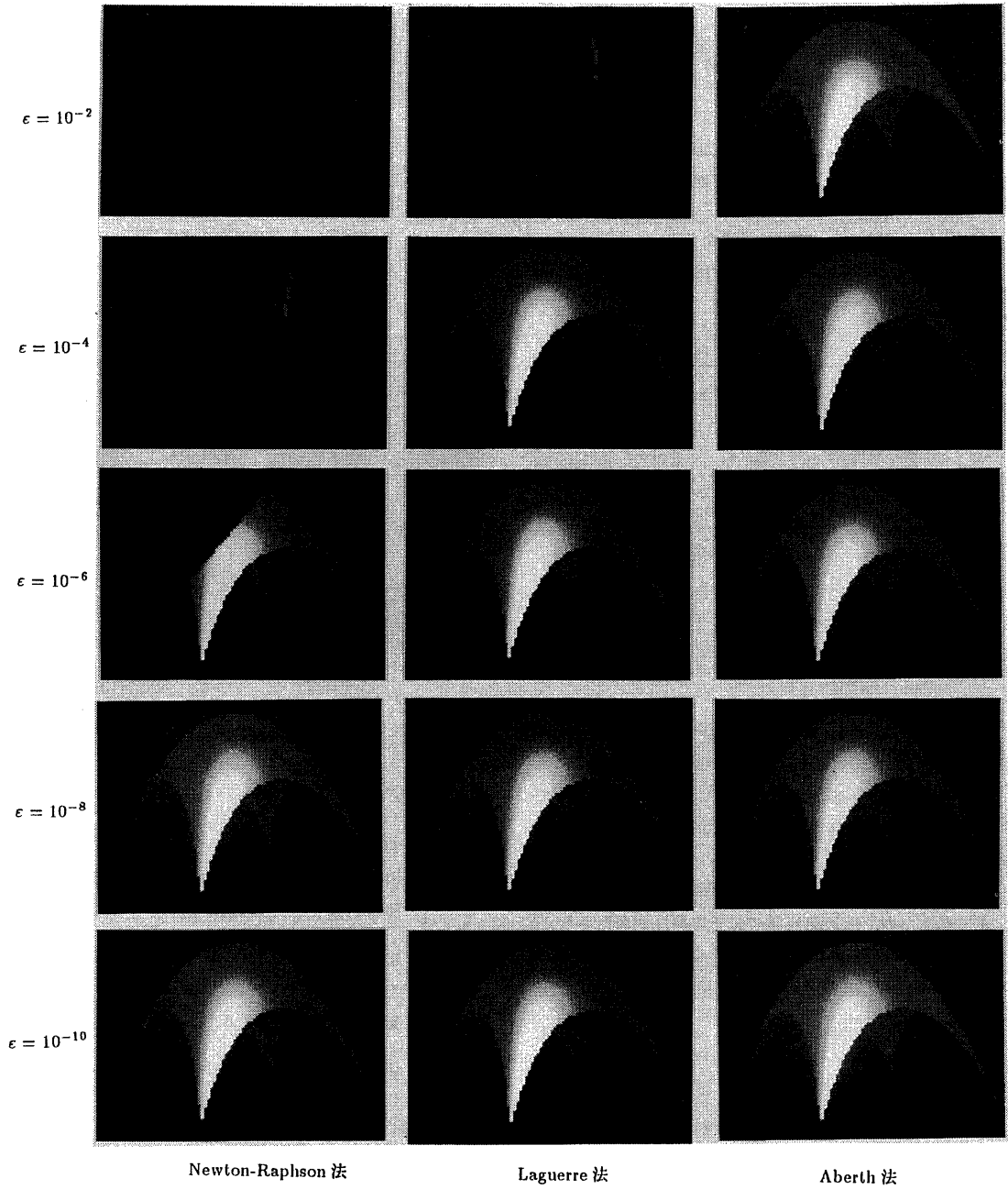


図 16: 双 3 次パッチ