

数式処理による反復演算について

Computation of interval arithmetic on computer algebra systems

関東学院大学 大墨礼子 *1
 NORIKO OSUMI
 KANTO GAKUIN UNIVERSITY

香川高等専門学校 近藤祐史 *2
 YUJI KONDOH
 NATIONAL INSTITUTE OF TECHNOLOGY, KAGAWA COLLEGE

防衛大学校 藤村雅代 *3
 MASAYO FUJIMURA
 NATIONAL DEFENSE ACADEMY

Abstract

The aim of this study is to find a method to draw Julia set with less error. In this talk, we compute Julia sets with interval number coefficients and compare results to ploop and Cartesian coordinates. From those experimental results, we will explore expansion and error suppressing method for general iterative computation.

1 反復演算とは

ジュリア集合は複素力学系の研究対象となる集合であり、その性質として、不安定な集合であるため、正確な描画は難しいとされている。複素関数 f は多項式または有理関数であるとする。関数 f の n 回の反復合成 $\underbrace{f \circ f \circ \dots \circ f}_n$ を f^n と書く。このとき、 $f^n(z) = z$ をみたす点 z を f の周期 n の周期点という。特に、 $n = 1$ のとき z を不動点という。点 z が f の周期 n の周期点のとき、 $\lambda = (f^n)'(z)$ の値を z の乗数という。関数 f の周期 n の周期点 z は、その乗数 λ によって

- $0 < |\lambda| < 1$ のとき吸引的
- $\lambda = 0$ のとき、超吸引的
- $|\lambda| > 1$ のとき、反発的
- $|\lambda| = 1$ のとき、中立的

*1 E-mail: osumi@kanto-gakuin.ac.jp

*2 E-mail: kondoh@di.kagawa-nct.ac.jp

*3 E-mail: masayo@nda.ac.jp

の4つに分類される。

点 z_0 を f の (超) 吸引不動点とすると、 $B(z_0) = \{z \in \mathbb{C} : f^n(z) \rightarrow z_0 (n \rightarrow \infty)\}$ を z_0 の吸引鉢、その z_0 を含む連結成分 $A(z_0)$ を z_0 の直接吸引鉢という。

関数 f の反復による関数列 $\{f^n\}$ が正規族をなさない点全体を f のジュリア集合といい $J(f)$ と記す(ただし、領域 $D \subset \widehat{\mathbb{C}}$ 上で定義される族 F に対して、 F の元からなる任意の列が広義一様収束する部分列を持つとき正規族であるという)。複素力学系で用いられる用語についての詳細は、[4] や [5] を参照されたい。

ジュリア集合の描画では多くの反復合成を行い、点の挙動を調べる必要がある。しかし、これらの描画には、ジュリア集合自身の不安定性に基づく問題もあるが、計算に浮動小数点数を用いることによっておこる問題が大きく影響している。本研究では、描画アルゴリズムのうち、単純な反復計算のみを行うレベルセット法を扱う。レベルセット法において浮動小数点数演算を用いると、丸め誤差の影響を大きく受けるため、描画されたものがどの程度正しいのかの判断が不可能である。誤差の影響を抑えるため、本研究では数式処理システム上で区間数を用いた計算を行い、その挙動の分析を行う。本研究の計算で用いる数式処理システム Risa/Asir の区間数にはゼロ書き換えモードの実装もされており、区間数と合わせてゼロ書き換えによる誤差の抑制を期待できる。

2 区間演算の反復計算への適用

区間演算は、区間を数の拡張と考え、その間の四則演算を定義する。実数で与えられる真値の上限と下限を浮動小数点数とし、浮動小数点数演算により実装する場合、厳密には機械区間演算と呼ばれる [1] が、本稿では、単に区間演算と呼ぶことにする。実装に際しては、上限下限の計算時に浮動小数点数の丸めモードを変更することにより、得られた区間に真値が必ず含まれる、つまり、下限以上で上限以下の区間に真の値があることを保証する。区間演算は、精度保証付き数値計算において利用されている。

数式処理システム Risa/Asir では区間演算の実装がなされている [2]。Risa/Asir 上での区間演算及び区間数とは、

$$A = \{x | \underline{a} \leq x \leq \bar{a}\} \quad x, \underline{a}, \bar{a} \in \mathbb{R}$$

なる A を区間数と呼び、 $A = [\underline{a}, \bar{a}]$ と表す。ただし、 $\underline{a} \leq \bar{a}$ とする。 \underline{a}, \bar{a} それぞれを区間数の下限、上限と呼ぶ。また、2つの区間数 $A = [\underline{a}, \bar{a}]$, $B = [\underline{b}, \bar{b}]$ の間の演算を次のように定義する。ここで、英大文字は区間数、英小文字は実数を表す。

$$\left\{ \begin{array}{l} A + B = [\underline{a}, \bar{a}] + [\underline{b}, \bar{b}] = [\underline{a} + \underline{b}, \bar{a} + \bar{b}], \\ A - B = [\underline{a}, \bar{a}] - [\underline{b}, \bar{b}] = [\underline{a} - \bar{b}, \bar{a} - \underline{b}], \\ A \cdot B = [\underline{a}, \bar{a}] \cdot [\underline{b}, \bar{b}] = [\min(\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}), \max(\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b})], \\ A / B = [\underline{a}, \bar{a}] / [\underline{b}, \bar{b}] = [\underline{a}, \bar{a}] \cdot [1/\bar{b}, 1/\underline{b}] \\ = [\min(\underline{a}/\bar{b}, \underline{a}/\underline{b}, \bar{a}/\bar{b}, \bar{a}/\underline{b}), \max(\underline{a}/\bar{b}, \underline{a}/\underline{b}, \bar{a}/\bar{b}, \bar{a}/\underline{b})], \\ \text{(ただし, } 0 \notin B\text{).} \end{array} \right.$$

Risa/Asir では特に設定しない場合は、区間の範囲 a, b は double の浮動小数点数となる。

また、Risa/Asir の区間演算では、ゼロ書き換えモードが実装されている。ゼロ書き換えとは、区間演算の結果がゼロが含まれる区間数となった場合、その区間をゼロに置き換える演算のことである。

本研究で描画アルゴリズムとしてレベルセット法を使用する。レベルセット法は、関数 f が多項式の場合、無限遠点は超吸引の不動点で、 $J(f) = \partial A(\infty)$ であるという性質を用いて充填ジュリア集合を描くアルゴリズムである。具体的には、描画領域の各格子点 z に対して、 f による軌道 $z, f(z), f^2(z), \dots$ を計算し、十分大きな正の値 $R > 2$ と十分大きな自然数 n で $|f^n(z)| > R$ となれば、 $z \in A_0(\infty)$ と判定する。そ

うでなければ、充填ジュリア集合の元と判定する。レベルセット法では、このような比較的単純なアルゴリズムで f の充填ジュリア集合の内点を描画することが出来る。さらに、 $|f^n(z)| > R$ となる最小の n の値によりプロットする色を変化させることで、充填ジュリア集合の境界であるジュリア集合を推測することが可能である。

本研究では、 $f_c(z) = z^2 + c$, ($c \in \mathbb{C}$) とし、各格子点は各軸とも -2 から 2 の範囲で $\frac{1}{150}$ の幅で刻むものとし、数式処理システム Risa/Asir を用いて計算および描画を行う。

2.1 区間演算を用いた計算

2.1.1 計算例: $c = -0.778264 + 0.1155285i$ の場合

$f_c(z) = z^2 + c$, $c = -0.778264 + 0.1155285i$ の場合を考える。この場合、充填ジュリア集合は連結集合であることを注意しておく。

まず、この例での充填ジュリア集合の概形は DEM 法を用いて図 1 に示した。DEM 法は今回の描画実験の対象ではないが、ジュリア集合の ϵ -近傍を描くアルゴリズムであり、数学的に実状に近い描画が可能である。

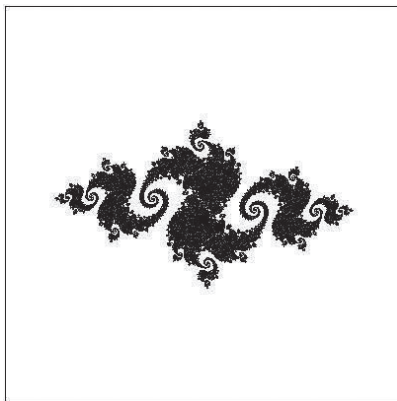


図 1: DEM 法による描画

次に、レベルセット法を用いて描画したものを図 2 に示す。図 2 の右の図に見られるように、レベルセット法では、繰り返し回数を上げると本来あるべき点が誤差の影響を大きく受けて消えてしまうという状況が起こっている。特に境界に近い点ではその影響が顕著である。

そこで、レベルセット法での計算に区間数を用いた描画実験を行う。区間数を用いる際には、描画領域の各格子点 z を区間数として演算する。単純に区間数を用いて計算した場合、充填ジュリア集合となる黒点は 1 つも検出されていない。次に、図 3 に double の区間数でゼロ書き換えを行った場合の結果と、bigfloat の区間数でゼロ書き換えを行った場合の結果を示す。

ゼロ書き換えモードを用いた区間数での演算では、double の区間数を用いたもので、充填ジュリア集合となる黒点は 11202 個、bigfloat を用いた区間数では 5398 個となる。

ここで、区間数およびゼロ書き換えモードを用いた場合の挙動について考える。ゼロ書き換えモードを用いると、上述の通り区間にゼロが含まれる場合はゼロとして演算する。そのため、座標軸上にある点の誤差を抑制することが可能となるのではないかと考えられる。

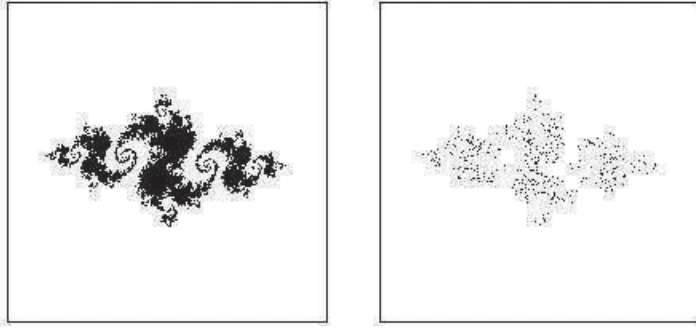


図 2: レベルセット法による描画 (左: 繰り返し回数 500 回, 右: 繰り返し回数 50000 回)

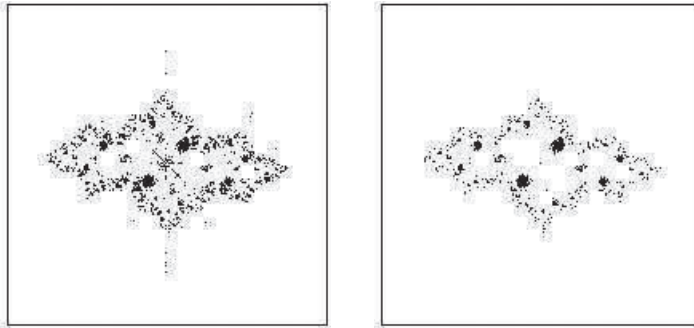


図 3: 左: double の区間数およびゼロ書き換えモードを用いた描画 (繰り返し回数 500 回)
右: bigfloat の区間数およびゼロ書き換えモードを用いた描画 (繰り返し回数 500 回)

2.1.2 計算例: $c = -1.75487766624669276$ の場合

$f(z) = z^2 + c$, $c = -1.75487766624669276$ の場合を考える. この場合も, 充填ジュリア集合は連結集合であり, 実軸の一部と原点の近くの開集合を含むことから, それらの逆像をすべて含んだ形状をしている. 充填ジュリア集合の概形は DEM 法を用いて図 4 に示した.

次に, 区間数を用いた場合と, 区間数およびゼロ書き換えモードを用いた描画の結果を示す.

図 5 の左の図に示されたように, 単純に区間数を用いる場合は, 実軸上に存在する充填ジュリア集合に含まれる区間が描画されない. 実軸上の点の挙動の計算は区間数がゼロを含む場合に相当すると考えられるため, ゼロ書き換えモードを用いれば補正できると推測した. 実際に, 区間数と合わせてゼロ書き換えモードを用いて描画を行ったところ, 図 5 の右の図に示す通り, 実軸上の黒点がより多く検出されて, 特に実軸の一部の描画がされていることがわかる. ただし, 虚軸上の点の計算が不正確であり, 本来の充填ジュリア集合には属さない部分の虚軸上の点も充填ジュリア集合の点としてプロットされていることが観測される.

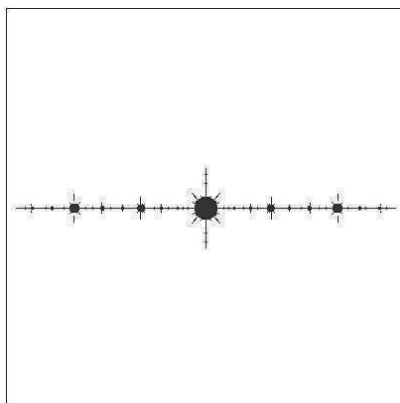


図 4: DEM 法による描画

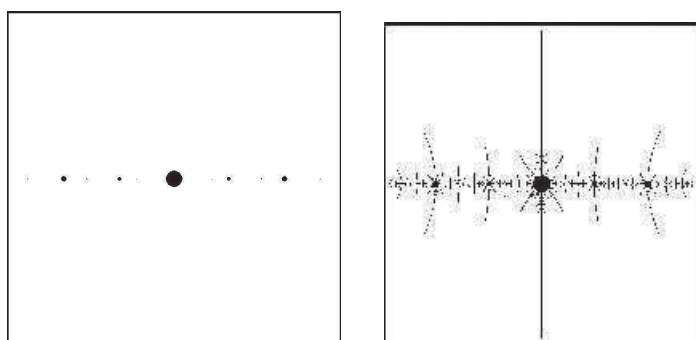


図 5: 左 : 区間数を用いた描画 (繰り返し回数 500 回)
 右 : 区間数およびゼロ書き換えモードを用いた描画 (繰り返し回数 500 回)

2.2 極座標形式での区間演算を用いた計算

区間数およびゼロ書き換えモードを用いた結果から、座標軸をはさんだ充填ジュリア集合の検出には区間数およびゼロ書き換えモードの適用が有効であることが考えられる。ここで、極座標系式において区間数を用いることを考える。複素関数を極座標形式に変換する場合には三角関数を用いるが、Risa/Asir の区間数では、三角関数に対応した区間数の実装はこれまでされていなかった。本研究では、三角関数に対応した区間数を実装し、これを用いて極座標系での描画実験を行った。実装された関数は、`evalitvd()`、`evalitvbf()` の 2 種類であり、いずれの関数も引数で三角関数の値を渡すことでその値を区間数として返すものである。`evalitvd()` が `double` の区間数を返すもの、`evalitvbf()` が `bigfloat` の区間数を返すものとなっている。使用例を以下に示す。

```
F=sin(1);
evalitvd(F);
evalitvbf(F);
```

これらの関数を用いて、描画対象の複素関数を極座標形式で区間数、区間数及びゼロ書き換えモードを用いて計算し、描画を行う。

2.2.1 計算例: $c = -1.75487766624669276$ の場合

ここでは、2.1.2 章において扱った $f(z) = z^2 + c$, $c = -1.75487766624669276$ を考える。

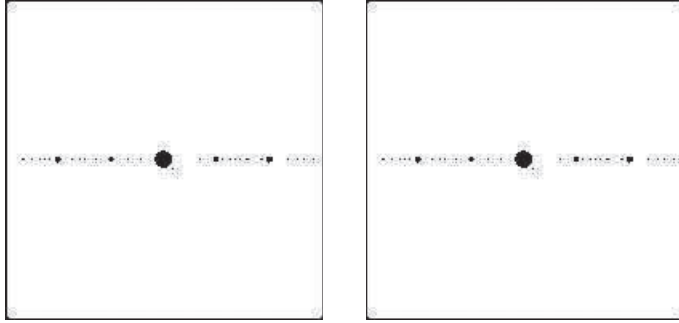


図 6: 右 : 極座標形式で区間数を用いた描画 (繰り返し回数 500 回)
左 : 極座標形式で区間数およびゼロ書き換えモードを用いた描画 (繰り返し回数 500 回)

図 6 に示す通り、極座標形式では単純な区間数の適用と、区間数およびゼロ書き換えモードの適用では差がみられない。また、原点の近くの開集合およびその逆像の描画には成功しているが、実軸の一部およびその逆像は描けていないことが観測できる。

今後、結果の詳細な分析が必要ではあるが、今回実装した三角関数に対応した区間数は、入力された角度の三角関数の値が区間となるため、ゼロ書き換えの際の誤差の抑え込みの位置が単純な区間数の適用の場合とは違っているためこのような描画結果となったことが推測される。

3 まとめ

本研究では、ジュリア集合のレベルセット法を用いた描画に区間演算を適用し、その結果を示した。今回新たに三角関数に対応した区間数の実装を行い、極座標形式での演算も行った。複素数の表現をデカルト座標系でなく極座標を用いた場合の演算では、ゼロ書き換えを用いた場合の結果が異なっており、それぞれ部分での誤差となるものが抑え込まれたのかの詳細な分析が今後の課題である。今後の最終目標は、有理数を用いた正確な描画の実現ではある。そのためのデータ型および式の展開方法の検討が次のステップであるが、いずれにせよ誤差の抑制は必ず必要となる。そのため、誤差の抑制法として、どのような場面で、どの形の区間演算を適用するのが最適化も十分調査していく必要がある。

謝辞

本研究の一部は JSPS 科研費 JP19K03531 の助成を受けたものです。

参 考 文 献

- [1] 大石進一, 精度保証付き数値計算, コロナ社, 2000.
- [2] 近藤祐史, 区間演算と数式処理の歴史, 数式処理 Vol.12, No.1, pp.23-31, 2005
- [3] 大墨礼子, 近藤祐史, 藤村雅代, 反復計算への区間演算の適用について, 京都大学数理解析研究所講究録 2138, pp.59-63, 2019
- [4] A. F. Beardon, *Iteration of Rational Functions*, Springer-Verlag, 1991.
- [5] J. Milnor, *Dynamics in one complex variables*, 3rd ed., Princeton Univ. Press, 2006.