反復計算への区間演算の適用について
An application of interval arithmetic to iterative computation

関東学院大学　大墨礼子

Noriko Osumi

Kanto Gakuin University *

香川高等専門学校　近藤祐史

Yuji Kondoh

National Institute of Technology, Kagawa College †

防衛大学校　藤村雅代

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 coeffients, and compare results to other kind of coefficients. Moreover, following this computer experiment, we will explore methods usable for general iterative computation.

1　はじめに

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．以降扱う複素関数 f は，多項式または有理式で与えられるとする．関数 f の n 回の反復合成 $\underbrace{f \circ f \circ \cdots \circ f}_{n}$ 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

*osumi@kanto-gakuin.ac.jp

†kondoh@di.kagawa-nct.ac.jp

‡masayo@nda.ac.jp

- $|\lambda| = 1$ のとき，中立的

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き，$B(z_0) = \{z \in \mathbb{C} \ : \ f^n(z) \to z_0 \ (n \to \infty)\}$ 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, [3] や [4] 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誤差なく，または誤差を抑えた描画を行うための検討材料として，区間数を用いた計算実験を行い，その挙動について調査する．

2　描画実験

本研究で描画アルゴリズムとして使用するレベルセット法は，関数 f が多項式の場合，無限遠点は超吸引的不動点で，$J(f) = \partial A(\infty)$ であるという性質を使い，充填ジュリア集合を描く．既存のソフトなどで最もよく使われる方法であるが，境界が不明瞭になることを等高線でカバーしている．

描画領域の各格子点 z に対して，f による軌道 $z, f(z), f^2(z), \dots$ を計算する．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．このような手順で, f の充填ジュリア集合を描画する.

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　浮動小数点数を用いた描画

ここで $c = -0.778264 + 0.1155285i$ として，$f_c(z)$ の ジュリア集合の計算を行うことを考える．十分大きな繰り返し回数を取るものとし，$|f^n(z)| < 10$ となる点を黒点で描画する．図 1 は 500 回繰り返した $(n = 500)$ 結果，図 2 は 10000 回繰り返した $(n = 10000)$ 結果である．図 2 では，図 1 に比べて検出されている点が減少し，充填ジュリア集合となる点が絞り込まれているように見えるが，浮動小数点数での演算のため，どこまで丸め誤差が影響を与えているか不明であり，正確な描画であるといえるかは難しいところである．

2.2　有理数を用いた実験

浮動小数点数を用いた計算では，繰り返し演算によって，大きく丸め誤差の影響を受ける．数学的に正確な描画を目指すには，浮動小数点部分を有理数とすれば，理論上は正確に描画が可能となるはずである．そこで，$c = -\frac{778264}{1000000} + \frac{1155285}{10000000}i$ とし，f_c のジュリア集合の描画を浮動小数点数の場合と同様に行った．有理数を用いた計算では，途中から計算が進まなくなってしまう．図 3 は，繰り返し回数が指定回数以下のものに対しても各点に黒色以外の着色を行って描画を行っている途中状態の図である．浮動小数点数の描画と比較すると，繰り返し演算回数が増加して，最初の黒点が出現する箇所付近で時間がかかっていると考えられる．コンピュータのリソース状況としては，メモリおよび CPU 使用率にも十分に余裕がある．演算が進

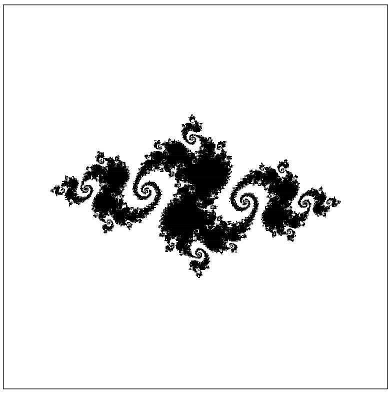

図 1: 繰り返し回数 500 回の描画

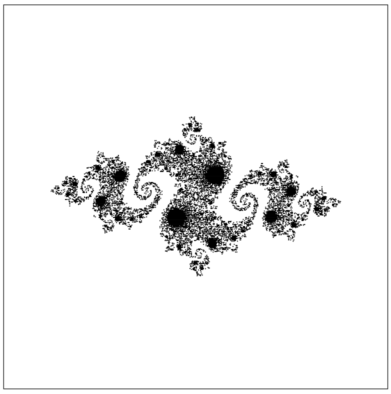

図 2: 繰り返し回数 10000 回の描画

図 3: 有理数を用いた描画

まなくなっている箇所付近では，非常に大きな有理数の係数を持つ計算が行われていることが確認できた．反復計算の結果，非常に長大な有理数 (分子分母ともに 400 桁以上) の演算を行う必要が出てくるため，その計算に時間がかかっていると考えられる．

2.3 区間演算を用いた計算

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．

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，

$$A = \{x | \underline{a} \leq x \leq \overline{a}\} \qquad x, \underline{a}, \overline{a} \in \mathbb{R}$$

なる A を区間数と呼び，$A = [\underline{a}, \overline{a}]$ と表す．ただし，$\underline{a} \leq \overline{a}$ とする．$\underline{a}$, $\overline{a}$ それぞれを区間数の下限，上限と呼ぶ．また，2 つの区間数 $A = [\underline{a}, \overline{a}]$, $B = [\underline{b}, \overline{b}]$ 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．

$$
\begin{cases}
A+B &= [\underline{a},\overline{a}]+[\underline{b},\overline{b}] = [\underline{a}+\underline{b},\overline{a}+\overline{b}],\\
A-B &= [\underline{a},\overline{a}]-[\underline{b},\overline{b}] = [\underline{a}-\overline{b},\overline{a}-\underline{b}],\\
A\ \cdot\ B &= [\underline{a},\overline{a}]\ \cdot\ [\underline{b},\overline{b}] = [\min(\underline{a}\underline{b},\underline{a}\overline{b},\overline{a}\underline{b},\overline{a}\overline{b}),\max(\underline{a}\underline{b},\underline{a}\overline{b},\overline{a}\underline{b},\overline{a}\overline{b})],\\
A\ /\ B &= [\underline{a},\overline{a}]\ /\ [\underline{b},\overline{b}] = [\underline{a},\overline{a}]\ \cdot\ [1/\overline{b},1/\underline{b}]\\
&= [\min(\underline{a}/\overline{b},\underline{a}/\underline{b},\overline{a}/\overline{b},\overline{a}/\underline{b}),\max(\underline{a}/\overline{b},\underline{a}/\underline{b},\overline{a}/\overline{b},\overline{a}/\underline{b})],\\
& \quad (\text{ただし，}\quad 0\notin B).
\end{cases}
$$

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．この区間数を用いて，描画領域の各格子点 z を区間数として演算する．この場合, 区間数による演算により実軸に平行な方向と虚軸に平行な方向の計算誤差の制御ができるが, ジュリア集合上では各点の近傍の点の軌道が複雑な振る舞いをするため, 上記の 2 方向の評価を用いる方法がどこまで描画に有効なのかは, 今後に調べるべき課題である．描画結果を図 4 に示す．図 4 では繰り返し回数が 10 回以下となる点は白色で，黒色以外の点にも繰り返し回数に応じで黒色以外の着色を行い表示している．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．反復回数の最大値は 39 回であり, 上限回数の 500 回よりはるかに小さい回数で判定が終了している．

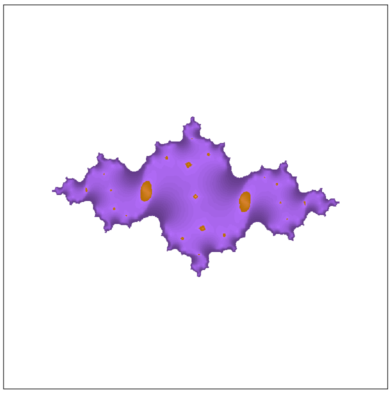

図 4: 区間数を用いた描画 (繰り返し回数 500 回)

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．このゼロ書き換えモードを用いて演算を行った結果を示す．ゼロ書き換えモードを用いた演算では，区間数としてデフォルトの double の区間数だけでなく，オプションを指定することで使用できる bigfloat の区間数でも演算を行った．図 5 に double の区間数でゼロ書き換えを行った場合の結果を，図 6 に 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 10717 個，bigfloat を用いた区間数では 5243 個となった．

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ついて区間演算を用いた．特にゼロ書き換えを用いる場合，浮動小数点数を用いた計算とは大きく異なる結果を得たが，その詳細な分析に関しては今後の課題である．また，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，また異なる

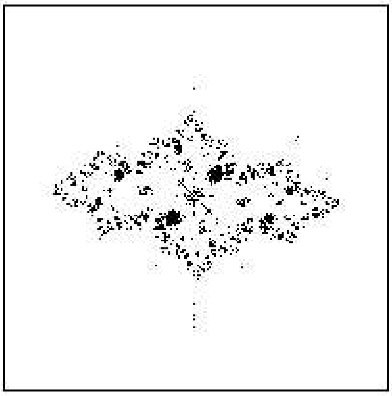

図 5: double の区間数およびゼロ書き換えモードを用いた描画 (繰り返し回数 500 回)

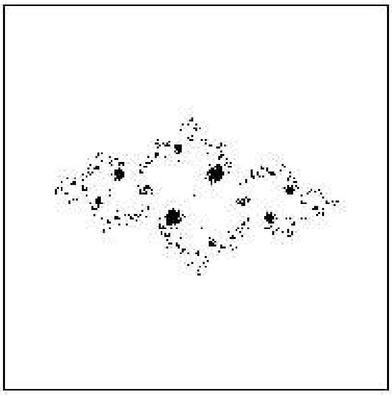

図 6: bigfloat の区間数およびゼロ書き換えモードを用いた描画 (繰り返し回数 500 回)

結果となる可能性がある．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が，誤差を極力抑える方法としての区間演算の適用の可能性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 JP15K04943 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] A. F. Beardon, *Iteration of Rational Functions*, Springer-Verlag, 1991.

[4] J. Milnor, *Dynamics in one complex variables*, 3rd ed., Princeton Univ. Press, 2006.