

Laplace 方程式の Dirichlet 境界値問題に対する精度保証

総合数理学部 現象数理学科

佐々木 彩奈

2021 年 2 月 28 日

1 はじめに

今回、私は卒業研究として大畑 [4] に沿って、Laplace 方程式の Dirichlet 境界値問題

$$(1.1) \quad \Delta u(x) = 0 \quad (x \in \Omega)$$

$$(1.2) \quad u(x) = \varphi(x) \quad (x \in \partial\Omega)$$

において、領域 Ω を 2 次元円盤領域としたときについての精度保証付き数値計算を行った。大畑 [4] では基本解の方法によって作成した近似解に対して、2 階楕円型方程式において重要である最大値原理を基にした精度保証付き数値計算を行っている。誤差の計算には、素朴な区間を用いた区間演算の他に、平均値の定理を基に作成する平均値形式や Taylor の定理を基に作成する Taylor 形式が用いられており、それらの計算に現れる区間演算にはドイツの S. M. Rump 教授によって作成された、MATLAB の区間演算ライブラリである INTLAB が用いられている。

2 円盤領域での Laplace 方程式に対する精度保証

2.1 Laplace 方程式に対する基本解の方法

Laplace 方程式の Dirichlet 境界値問題 (1.1), (1.2) に対する近似解の作成について記す。Laplace 方程式 (1.1) の基本解

$$u(x) = -\frac{1}{2\pi} \log |x|$$

を平行移動したものの線型結合によって得られる関数

$$(2.1) \quad u_N(x) = \sum_{j=1}^N Q_j \log |x - y_j|$$

も (1.1) の解となる。ただし、 $Q_j \in \mathbb{R}$, $x \in \bar{\Omega}$ であり、点 $\{y_j\}_{j=1, \dots, N}$ は領域 Ω の外部の点である。次に、(2.1) の係数 Q_j を Dirichlet 境界条件 (1.2) を満たすように定める。つまり、 $\partial\Omega$ 上に $\{x_k\}_{k=1, \dots, N}$ を取り、

$$(2.2) \quad u_N(x_k) = \sum_{j=1}^N Q_j \log |x_k - y_j| = \varphi(x_k) \quad (k = 1, \dots, N)$$

を満たすように Q_j を定める。式 (2.2) を行列を用いて書き直すと

$$(2.3) \quad \begin{pmatrix} \log|x_1 - y_1| & \log|x_1 - y_2| & \cdots & \log|x_1 - y_N| \\ \log|x_2 - y_1| & \log|x_2 - y_2| & \cdots & \log|x_2 - y_N| \\ \vdots & \vdots & \ddots & \vdots \\ \log|x_N - y_1| & \log|x_N - y_2| & \cdots & \log|x_N - y_N| \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{pmatrix} = \begin{pmatrix} \varphi(x_1) \\ \varphi(x_2) \\ \vdots \\ \varphi(x_N) \end{pmatrix}$$

となる。ここで、領域 Ω を 2 次元円盤領域 $\Omega = \{x \in \mathbb{R}^2 \mid |x| < \rho^2\}$ とし、拘束点 $\{x_k\}$ を領域 $\partial\Omega$ 上、電荷点 $\{y_j\}$ を半径 R の円状にそれぞれ配置することとする。このとき、桂田 [1] の補題 3.2.3 により、 $R^N - \rho^N \neq 1$ であるならば、連立一次方程式 (2.3) は可解となる。

2.2 調和関数についての最大値原理を用いた誤差評価

前節で作成した近似解

$$(2.1) \quad u_N(x) = \sum_{j=1}^N Q_j \log|x - y_j|$$

は 2 次元領域 Ω において調和関数であることが確認できる。また、Laplace 方程式 (1.1) の厳密解 u についても 2 次元領域 Ω において調和関数であることが分かる。したがって、厳密解と近似解の誤差を表す関数 $f(x)$ を

$$f(x) = u(x) - u_N(x)$$

と定めるとき、 f も Ω において調和関数となる。調和関数に対して、以下の最大値原理が成り立つ。

定理 1. (調和関数についての最大値原理)

Ω は \mathbb{R}^2 の有界領域、 $\partial\Omega$ は領域 Ω の境界であるとする。このとき、 $u \in C^2(\Omega) \cap C(\bar{\Omega})$ が Ω で調和ならば、

$$(2.4) \quad \max_{x \in \bar{\Omega}} u(x) = \max_{x \in \partial\Omega} u(x), \quad \min_{x \in \bar{\Omega}} u(x) = \min_{x \in \partial\Omega} u(x)$$

が成り立つ。

この最大値原理により、上記の $f(x)$ について以下が成り立つことがわかる。

$$\max_{x \in \bar{\Omega}} f(x) = \max_{x \in \partial\Omega} f(x), \quad \min_{x \in \bar{\Omega}} f(x) = \min_{x \in \partial\Omega} f(x).$$

これにより、

$$(2.5) \quad \max_{x \in \bar{\Omega}} |f(x)| = \max_{x \in \partial\Omega} |f(x)|$$

となることが分かる。今回はこの性質を利用して、 $\partial\Omega$ での誤差の最大値を求めることによって 2 次元円盤領域 $\Omega = \{x \in \mathbb{R}^2 \mid |x| < \rho^2\}$ での誤差の評価を行う。

2.3 数値実験

以下に、今回行った数値実験についての説明と実行結果を記す。使用したプログラムについては 4 章に載せる。

2.3.1 精度保証なしでの数値計算

1つ目の数値実験として、精度保証を行わずに基本解の方法によって近似解を作成し、ランダムに選んだ $\partial\Omega$ 上の点 tx において、厳密解と近似解の値の差

$$e = \max |u_N(tx) - u(tx)|$$

を計算し、その最大値を求めることによって誤差を見積もった。使用したプログラムは4.1節に載せている。

拘束点 $\{x_\ell\}$ と電荷点 $\{y_\ell\}$ の個数を N として、

$$\theta_\ell = \frac{2(\ell-1)\pi}{N}, \quad x_\ell = \rho e^{i\theta_\ell}, \quad y_\ell = R e^{i\theta_\ell}$$

と定める。ただし、 \mathbb{R}^2 と \mathbb{C} を同一視した表記を用いている。例えば、 $x_\ell = \rho e^{i\theta_\ell}$ の \mathbb{R}^2 での座標は $(\rho \cos \theta_\ell, \rho \sin \theta_\ell)$ である。このとき、 $\rho = 1$ を固定して境界条件を与える関数や R の値を変えて、 N の値と誤差の関係を表すグラフを描画したところ、以下ようになった。

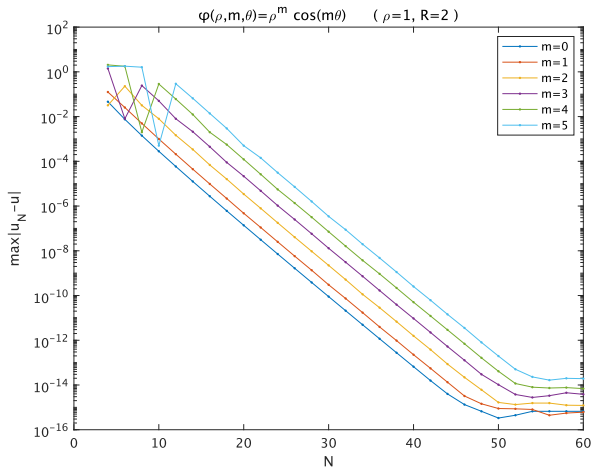


図 1: 実行結果 ($\varphi(x) = \rho^m \cos m\theta, R = 2$)

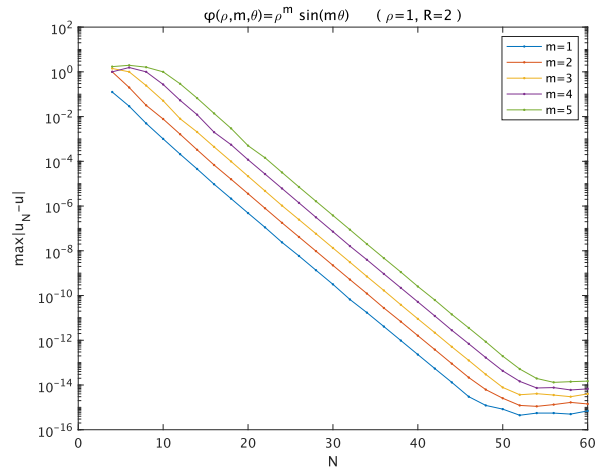


図 2: 実行結果 ($\varphi(x) = \rho^m \sin m\theta, R = 2$)

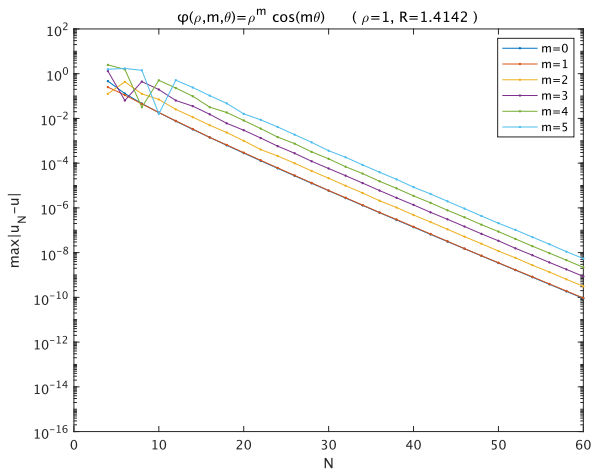


図 3: 実行結果 ($\varphi(x) = \rho^m \cos m\theta, R = \sqrt{2}$)

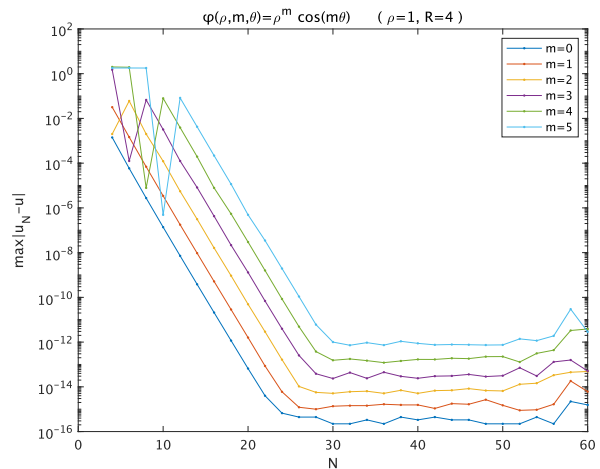


図 4: 実行結果 ($\varphi(x) = \rho^m \cos m\theta, R = 4$)

上記の結果において、誤差が指数関数的に減少している部分が見られる。この部分において、図1、図2の条件下では $O(N^{-18})$ 程度、図3の条件下では $O(N^{-11})$ 程度、図4の条件下では $O(N^{-23}) \sim O(N^{-25})$ 程度にそれぞれ比例して誤差が減少している様子が確認できた。

この方法における誤差の計算ではランダムに選んだ点における誤差の最大値を求めており、確実に領域 Ω における誤差の最大値が得られているとは限らないことに注意が必要である。一方、次節以降の計算では区間演算を用いて計算しているため、実際の誤差は得られた結果以下となるように改善されている。

2.3.2 区間演算による精度保証

2つ目の数値実験として、前節と同様に通常の数値計算によって近似解 u_N を作成した後に、境界上における厳密解

$$u(x) = \varphi(x) \quad (x \in \partial\Omega)$$

との差

$$f(x) = |u(x) - u_N(x)|$$

を区間演算を用いて計算することで誤差の計算を行った。使用したプログラムは4.2節に載せている。

2.3.1と同様に拘束点 $\{x_\ell\}$ と電荷点 $\{y_\ell\}$ の個数を N として、

$$\theta_\ell = \frac{2(\ell-1)\pi}{N}, \quad x_\ell = \rho e^{i\theta_\ell}, \quad y_\ell = R e^{i\theta_\ell}$$

と定める。加えて、誤算の計算に用いる区間 I_ℓ を

$$I_\ell = [\theta_\ell, \theta_{\ell+1}]$$

とおく。各区間 I_ℓ での誤差の最大値 $\max_{x \in I_\ell} |f(x)|$ を求めた後に、更にそれらの中での最大値

$$e = \max_{1 \leq \ell \leq N} \max_{x \in I_\ell} |f(x)|$$

を求めることで全体の誤差を求めている。 $\rho = 1$ を固定して境界条件を与える関数や R の値を変えて、 N の値と誤差の関係を表すグラフを描画したところ、以下ようになった。

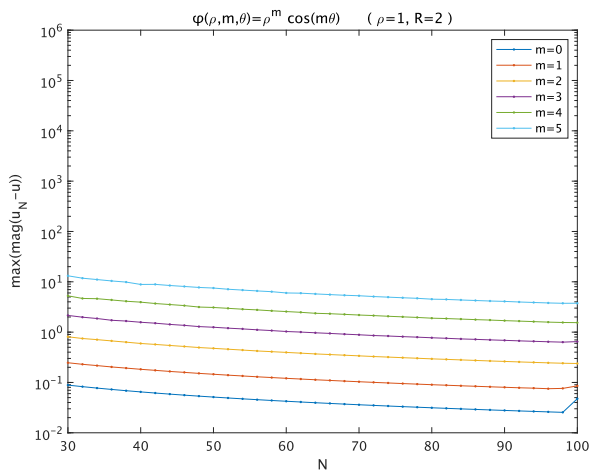


図5: 実行結果 ($\varphi(x) = \rho^m \cos m\theta, R = 2$)

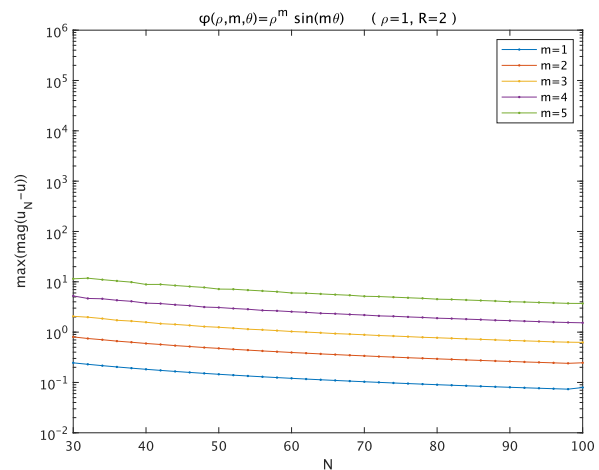


図6: 実行結果 ($\varphi(x) = \rho^m \sin m\theta, R = 2$)

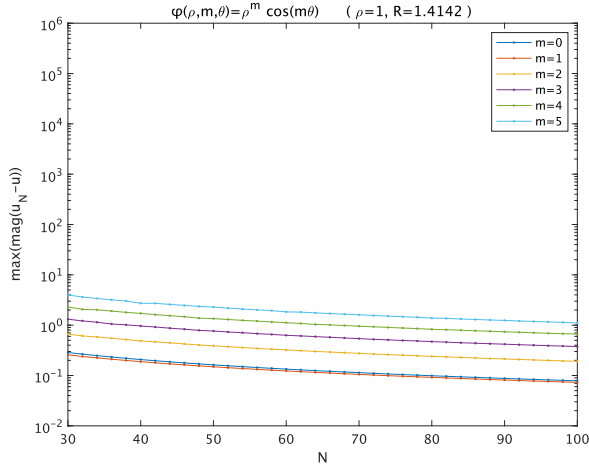


図 7: 実行結果 ($\varphi(x) = \rho^m \cos m\theta, R = \sqrt{2}$)

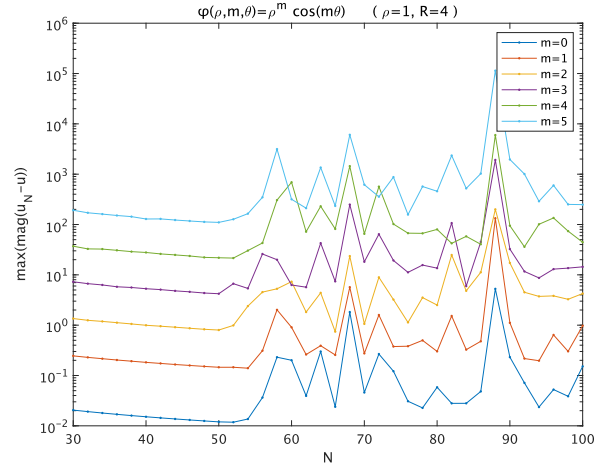


図 8: 実行結果 ($\varphi(x) = \rho^m \cos m\theta, R = 4$)

上記の結果において、誤差が指数関数的に減少している部分が見られる。この部分において、図5～図8のいずれの条件下でも $O(N^{-1})$ 程度に比例して誤差が減少している様子が確認できた。

2.3.3 より狭い区間での区間演算による精度保証

3つ目の数値実験として、前節までと同様に通常の数値計算によって近似解を作成した後に、2.3.2で誤差評価に用いた区間 I_ℓ をさらに N_2 等分した区間 $I_{\ell,k}$ 、すなわち

$$m = (\ell - 1) \cdot N_2 + k, \quad \theta_m = \frac{2(m - 1)\pi}{N \cdot N_2}, \quad I_{\ell,k} = I_m = [\theta_m, \theta_{m+1}]$$

を用いて誤差の計算を行った。使用したプログラムは4.3節に載せている。

2.3.2と同様に拘束点 $\{x_\ell\}$ と電荷点 $\{y_\ell\}$ の個数を N として、

$$\theta_\ell = \frac{2(\ell - 1)\pi}{N}, \quad x_\ell = \rho e^{i\theta_\ell}, \quad y_\ell = R e^{i\theta_\ell}, \quad I_\ell = [\theta_\ell, \theta_{\ell+1}]$$

と定める。プログラムでは、区間 I_ℓ を N_2 等分した区間 $I_{\ell,k}$ での誤差の最大値 $\max_{x \in I_{\ell,k}} |f(x)|$ を求め、それらを用いて各区間 I_ℓ での最大値 $\max_{1 \leq k \leq N_2} \max_{x \in I_{\ell,k}} |f(x)|$ を求め、更にそれらの中での最大値

$$e = \max_{1 \leq \ell \leq N} \max_{1 \leq k \leq N_2} \max_{x \in I_{\ell,k}} |f(x)|$$

を求めることで全体の誤差を求めている。境界条件を与える関数 $\varphi(x)$ を

$$\varphi(x) = \rho^5 \cos 5\theta$$

とし、 $N = 50$, $\rho = 1$ を固定して $R = \sqrt{2}, 2, 4, 8$ とした時の N_2 の値と誤差の関係を表すグラフを描画したところ、以下ようになった。

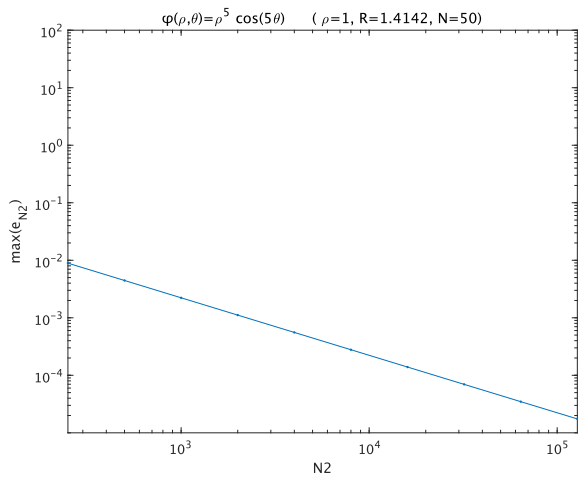


図 9: 実行結果 ($R = \sqrt{2}$)

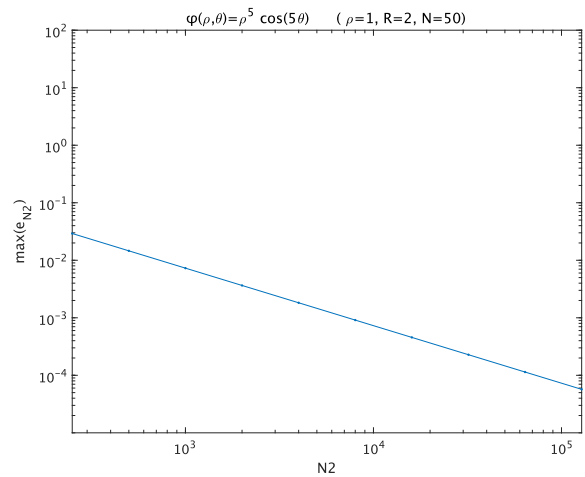


図 10: 実行結果 ($R = 2$)

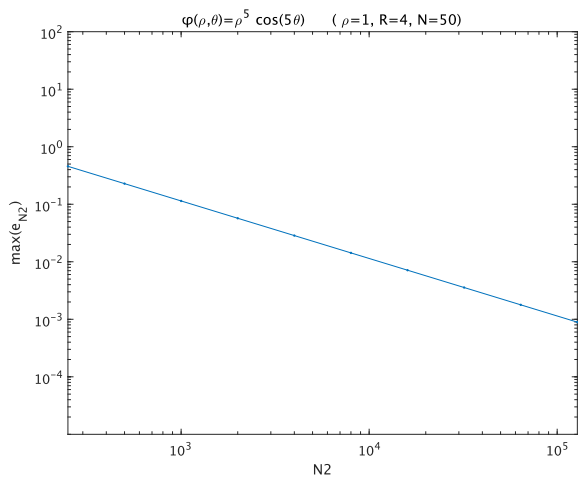


図 11: 実行結果 ($R = 4$)

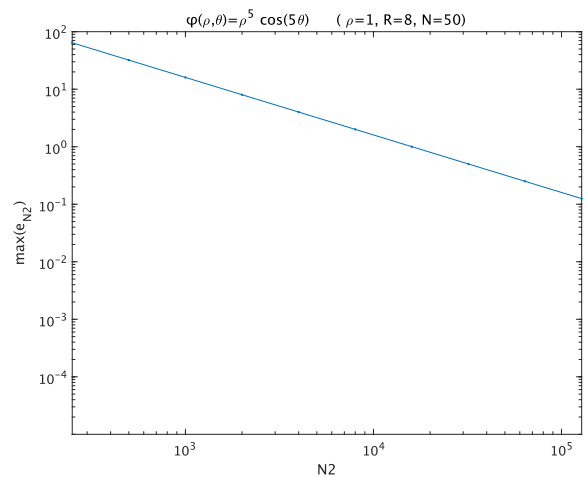


図 12: 実行結果 ($R = 8$)

上記の結果において $R = \sqrt{2}, 2, 4, 8$ のいずれの場合でも $O(N_2^{-1})$ 程度に比例して誤差が指数関数的に減少している様子が確認できた。

2.3.4 平均値形式を用いた精度保証

4つ目の数値実験として、2.3.3 までと同様に通常の数値計算によって近似解を作成した後に、平均値の定理から導かれる平均値形式によって誤差の計算を行った。使用したプログラムは4.4 節に載せている。

誤差の計算に用いた平均値形式について記す。平均値形式は以下の平均値の定理を基に構成されている。

定理 2. (平均値の定理)

写像 $f : [a, b] \rightarrow \mathbb{R}$ が微分可能であるとき、

$$(2.6) \quad f(a) - f(b) = f'(c)(a - b)$$

を満たす $c \in (a, b)$ が存在する。

ここで、関数 f についての区間拡張 f_{\square} を定義する。実数に関する演算規則によって定義された関数 f について、その演算規則を区間演算に置き換えて定義される関数を f の区間拡張といい、 f_{\square} と表す。この f と f_{\square} について、

$$(2.7) \quad f(X) \subset f_{\square}(X), \quad f'(X) \subset f'_{\square}(X)$$

が成り立つ。この区間拡張を平均値の定理の $f'(c)$ に施すことで、次の定理が得られる。ただし、実区間全体の集合を \mathbb{IR} を表すこととする。

定理 3. (平均値形式)

$D \subset \mathbb{R}$ とする。写像 $f : D \rightarrow \mathbb{R}$ が1回連続微分可能としたとき、 $X \in \mathbb{IR}$, $X \subset D$, $x, \tilde{x} \in X$ に対して

$$(2.8) \quad f(x) \in f(\tilde{x}) + f'_{\square}(X)(x - \tilde{x})$$

が成り立つ。

Proof.

$x < \tilde{x}$ として証明する。平均値の定理より、

$$f(x) - f(\tilde{x}) = f'(c)(x - \tilde{x})$$

すなわち

$$f(x) = f(\tilde{x}) + f'(c)(x - \tilde{x})$$

を満たす $c \in [x, \tilde{x}]$ が存在する。ここで、 $c \in [x, \tilde{x}] \subset X$ であるから、

$$f'(c) \in f'(X) \subset f'_{\square}(X)$$

が成り立つ。よって、

$$f(x) = f(\tilde{x}) + f'(c)(x - \tilde{x}) \in f(\tilde{x}) + f'_{\square}(X)(x - \tilde{x})$$

が成り立つ。 □

ここで、区間 X を十分小さくとるとき、 $f'_{\square}(X)$ はある程度狭い区間、 $x - \tilde{x}$ は十分に小さい値となる。したがって右辺によって与えられる区間はかなり小さい区間となることが期待される。

2.3.3 と同様に拘束点 $\{x_{\ell}\}$ と電荷点 $\{y_{\ell}\}$ の個数を N として、

$$\theta_{\ell} = \frac{2(\ell - 1)\pi}{N}, \quad x_{\ell} = \rho e^{i\theta_{\ell}}, \quad y_{\ell} = R e^{i\theta_{\ell}}, \quad I_{\ell} = [\theta_{\ell}, \theta_{\ell+1}]$$

と定め、区間 I_ℓ を N_2 等分した区間を $I_{\ell,k}$ とする。ここで、1つの区間 $I_{\ell,k}$ を Θ 、その中心と半径をそれぞれ $\text{mid}(\Theta)$, $\text{rad}(\Theta)$ と表す。すなわち、

$$\text{mid}(\Theta) = \frac{\theta_\ell + \theta_{\ell+1}}{2}, \quad \text{rad}(\Theta) = \frac{\theta_{\ell+1} - \theta_\ell}{2}$$

このとき、 $\text{mid}(\Theta) \in \Theta$ であるから、平均値形式 (2.8) により、

$$\forall \theta \in \Theta, \quad f(\theta) \in f(\text{mid}(\Theta)) + f'_{\square}(\Theta)(\theta - \text{mid}(\Theta))$$

が成り立つ。ここで、 $\theta - \text{mid}(\Theta)$ は θ を点区間とみなした区間演算を表している。上の式から、

$$f(\Theta) \subset f(\text{mid}(\Theta)) + f'_{\square}(\Theta)(\Theta - \text{mid}(\Theta))$$

が成り立つことが分かる。また、区間 Θ について

$$\Theta - \text{mid}(\Theta) = [-\text{rad}(\Theta), \text{rad}(\Theta)]$$

が成り立つので、

$$f(\Theta) \subset f(\text{mid}(\Theta)) + f'_{\square}(\Theta) [-\text{rad}(\Theta), \text{rad}(\Theta)]$$

となることが得られる。今回はこの右辺

$$f_{\text{mean}}(\Theta) := f(\text{mid}(\Theta)) + f'_{\square}(\Theta) [-\text{rad}(\Theta), \text{rad}(\Theta)]$$

を計算することによって誤差の値が含まれる区間を計算し、近似解の精度保証を行った。この節の説明の最後に、平均値形式の計算に用いた $u_n(x)$ と $\varphi(x)$ の極座標による表示と、それぞれの θ についての1階導関数を載せておく。ただし、

$$z(\theta) = e^{i\theta} = (\cos \theta, \sin \theta) \quad (\theta \in [0, 2\pi])$$

と定め、境界条件を与える関数 $\varphi(x)$ は

$$\varphi(x) = \rho^5 \cos 5\theta$$

とする。

$$\begin{aligned} u_N(z(\theta)) &= \sum_{j=1}^N Q_j \log \sqrt{\rho^2 + R^2 - 2\rho R \cos(\theta - \theta_j)}, \\ \frac{\partial}{\partial \theta} u_N(z(\theta)) &= \sum_{j=1}^N Q_j \frac{-\rho R \sin(\theta_j - \theta)}{\rho^2 + R^2 - 2\rho R \cos(\theta_j - \theta)}, \\ \varphi(z(\theta)) &= \rho^5 \cos 5\theta, \quad \frac{\partial}{\partial \theta} \varphi(z(\theta)) = -5\rho^5 \sin 5\theta. \end{aligned}$$

$N = 50, \rho = 1$ を固定して $R = \sqrt{2}, 2, 4, 8$ とした時の N_2 の値と誤差の関係を表すグラフを描画したところ、以下のようになった。

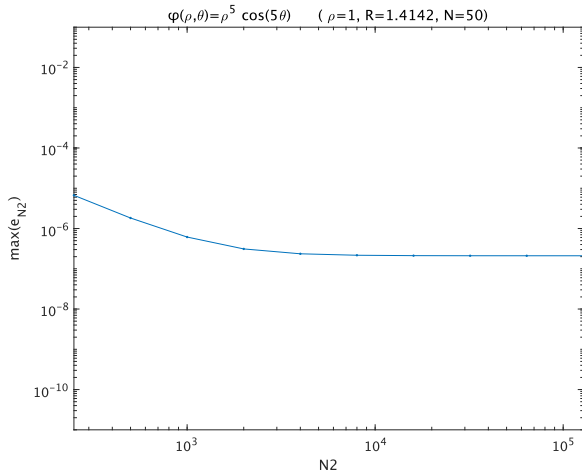


図 13: 実行結果 ($R = \sqrt{2}$)

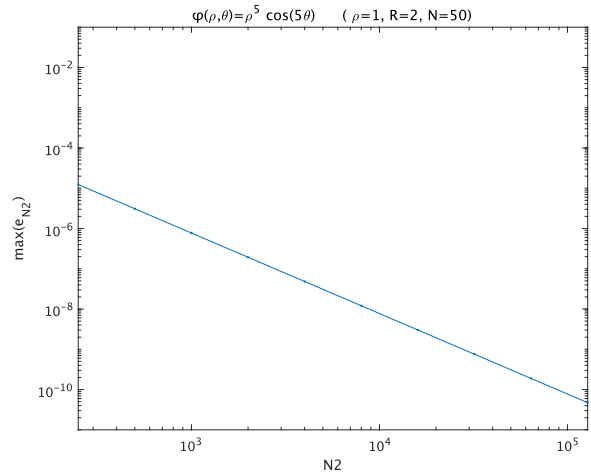


図 14: 実行結果 ($R = 2$)

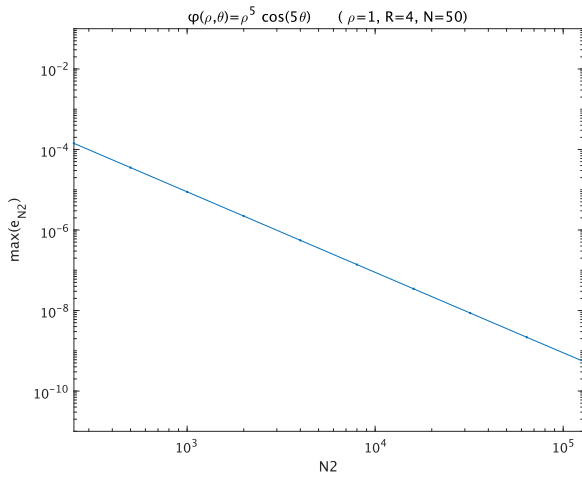


図 15: 実行結果 ($R = 4$)

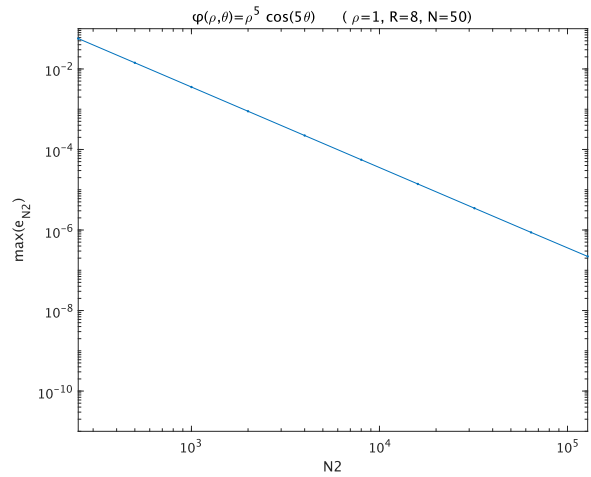


図 16: 実行結果 ($R = 8$)

上記の結果において誤差が指数関数的に減少している部分が見られる。この部分において、 $R = \sqrt{2}$ のときには $O(N_2^{-1.2})$ 程度に比例して、 $R = 2, 4, 8$ のときには $O(N_2^{-2})$ 程度に比例して、誤差が減少している様子が確認できた。

2.3.5 Taylor 形式を用いた精度保証

5つ目の数値実験として、2.3.4 までと同様に通常の数値計算によって近似解を作成した後に、Taylor の定理から導かれる Taylor 形式によって誤差の計算を行った。使用したプログラムは 4.5 節に載せている。

誤差の計算に用いた Taylor 形式について記す。Taylor 形式は以下の Taylor の定理を基に構成されている。

定理 4. (Taylor の定理)

D は \mathbb{R} の区間とする。写像 $f : D \rightarrow \mathbb{R}$ が区間 $X \subset D$ において k 階微分可能とする。このとき、 $x, \tilde{x} \in X$ に対して

$$(2.9) \quad f(x) = \sum_{n=0}^{k-1} \frac{1}{n!} f^{(n)}(\tilde{x})(x - \tilde{x})^n + \frac{1}{k!} f_{[\]}^{(k)}(c)(x - \tilde{x})^k$$

が成立するような $c \in X$ が存在する。

$f^{(k)}(c)$ について区間拡張を施すことで、次の Taylor 形式が得られる。

定理 5. (Taylor 形式)

D は \mathbb{R} の区間とする。写像 $f : D \rightarrow \mathbb{R}$ が区間 $X \subset D$ において k 階連続微分可能とする。このとき、 $x, \tilde{x} \in X$ に対して

$$(2.10) \quad f(x) \in \sum_{n=0}^{k-1} \frac{1}{n!} f^{(n)}(\tilde{x})(x - \tilde{x})^n + \frac{1}{k!} f_{[\]}^{(k)}(X)(x - \tilde{x})^k$$

が成立する。

ここで、区間 X を十分小さくとるとき、 $f_{[\]}^{(k)}(X)$ はある程度狭い区間、 $x - \tilde{x}$ は十分に小さい値となる。したがって右辺によって与えられる区間はかなり小さい区間となることが期待される。

2.3.4 と同様に拘束点 $\{x_\ell\}$ と電荷点 $\{y_\ell\}$ の個数を N として、

$$\theta_\ell = \frac{2(\ell - 1)\pi}{N}, \quad x_\ell = \rho e^{i\theta_\ell}, \quad y_\ell = R e^{i\theta_\ell}, \quad I_\ell = [\theta_\ell, \theta_{\ell+1}]$$

と定め、区間 I_ℓ を N_2 等分した区間を $I_{\ell,k}$ とする。ここで、1つの区間 $I_{\ell,k}$ を Θ 、その中心と半径をそれぞれ $\text{mid}(\Theta)$, $\text{rad}(\Theta)$ と表す。このとき、 $\text{mid}(\Theta) \in \Theta$ であるから、Taylor 形式 (2.10) により、

$$\forall \theta \in \Theta, \quad f(\theta) \in \sum_{n=0}^{k-1} \frac{1}{n!} f^{(n)}(\text{mid}(\Theta))(\theta - \text{mid}(\Theta))^n + \frac{1}{k!} f_{[\]}^{(k)}(\Theta)(\theta - \text{mid}(\Theta))^k$$

が成り立つ。よって、

$$f(\Theta) \subset \sum_{n=0}^{k-1} \frac{1}{n!} f^{(n)}(\text{mid}(\Theta))(\Theta - \text{mid}(\Theta))^n + \frac{1}{k!} f_{[\]}^{(k)}(\Theta)(\Theta - \text{mid}(\Theta))^k$$

が成り立つことが分かる。また、区間 Θ について

$$\Theta - \text{mid}(\Theta) = [-\text{rad}(\Theta), \text{rad}(\Theta)]$$

が成り立つので、

$$f(\Theta) \subset \sum_{n=0}^{k-1} \frac{1}{n!} f^{(n)}(\text{mid}(\Theta)) [-\text{rad}(\Theta), \text{rad}(\Theta)]^n + \frac{1}{k!} f_{[\]}^{(k)}(\Theta) [-\text{rad}(\Theta), \text{rad}(\Theta)]^k$$

となることが得られる。今回は $k = 2$ としたときのこの右辺

$$\begin{aligned} f_{\text{Taylor}}(\Theta) &:= f(\text{mid}(\Theta)) + f'(\text{mid}(\Theta)) [-\text{rad}(\Theta), \text{rad}(\Theta)] + \frac{f''_{[\]}(\Theta)}{2} [-\text{rad}(\Theta), \text{rad}(\Theta)]^2 \\ &= f(\text{mid}(\Theta)) + \left\{ f'(\text{mid}(\Theta)) + \frac{f''_{[\]}(\Theta)}{2} [-\text{rad}(\Theta), \text{rad}(\Theta)] \right\} [-\text{rad}(\Theta), \text{rad}(\Theta)] \end{aligned}$$

を計算することによって誤差の値が含まれる区間を計算し、近似解の精度保証を行った。この節の説明の最後に、Taylor 形式の計算に用いた $u_n(x)$ と $\varphi(x)$ の極座標による表示と、それぞれの θ についての 1 階導関数及び 2 階導関数を載せておく。ただし、

$$z(\theta) = e^{i\theta} = (\cos \theta, \sin \theta) \quad (\theta \in [0, 2\pi])$$

と定め、境界条件を与える関数 $\varphi(x)$ は

$$\varphi(x) = \rho^5 \cos 5\theta$$

とする。

$$\begin{aligned} u_N(z(\theta)) &= \sum_{j=1}^N Q_j \log \sqrt{\rho^2 + R^2 - 2\rho R \cos(\theta - \theta_j)}, \\ \frac{\partial}{\partial \theta} u_N(z(\theta)) &= \sum_{j=1}^N Q_j \frac{-\rho R \sin(\theta_j - \theta)}{\rho^2 + R^2 - 2\rho R \cos(\theta_j - \theta)}, \\ \frac{\partial^2}{\partial \theta^2} u_N(z(\theta)) &= \sum_{j=1}^N Q_j \frac{(\rho^2 + R^2) \rho R \cos(\theta - \theta_j) - 2\rho^2 R^2}{(\rho^2 + R^2 - 2\rho R \cos(\theta - \theta_j))^2} \\ \varphi(z(\theta)) &= \rho^5 \cos 5\theta, \quad \frac{\partial}{\partial \theta} \varphi(z(\theta)) = -5\rho^5 \sin 5\theta, \quad \frac{\partial^2}{\partial \theta^2} \varphi(z(\theta)) = -25\rho^2 \cos 5\theta. \end{aligned}$$

$N = 50, \rho = 1$ を固定して $R = \sqrt{2}, 2, 4, 8$ とした時の N_2 の値と誤差の関係を表すグラフを描画したところ、以下ようになった。

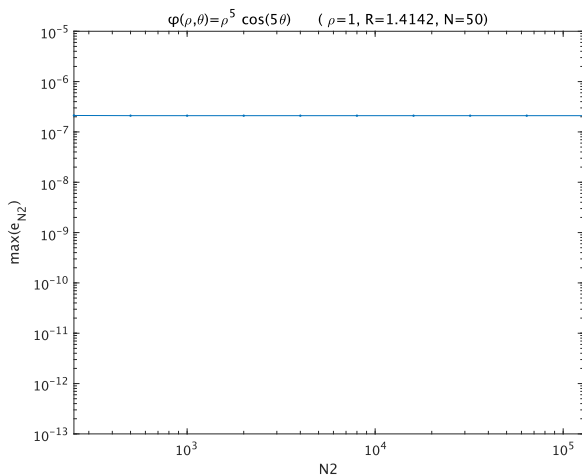


図 17: 実行結果 ($R = \sqrt{2}$)

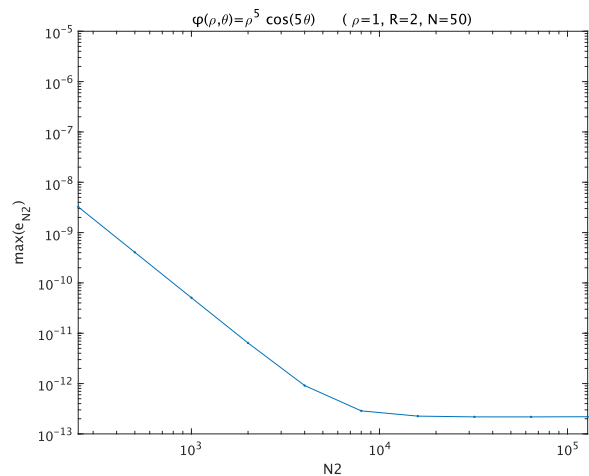


図 18: 実行結果 ($R = 2$)

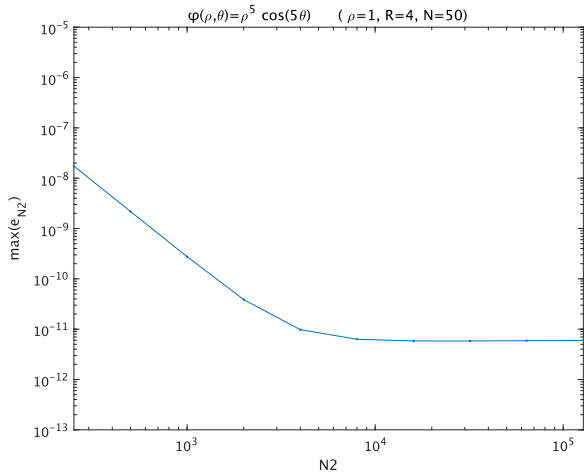


図 19: 実行結果 ($R = 4$)

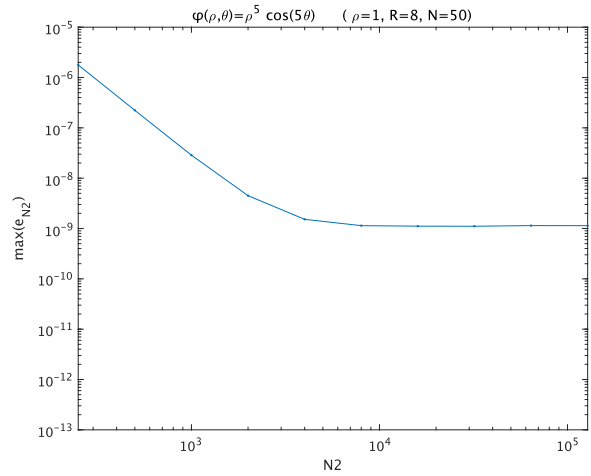


図 20: 実行結果 ($R = 8$)

上記の結果において誤差が指数関数的に減少している部分が見られる。この部分において、 $R = 2, 4$ のときには $O(N_2^{-2.7})$ 程度に比例して、 $R = 8$ のときには $O(N_2^{-2.5})$ 程度に比例して、誤差が減少している様子が確認できた。

2.3.6 数値実験のまとめ

2.3.3 節、2.3.4 節、2.3.5 節で行った数値実験の結果を改めて描画したところ、以下のようになった。

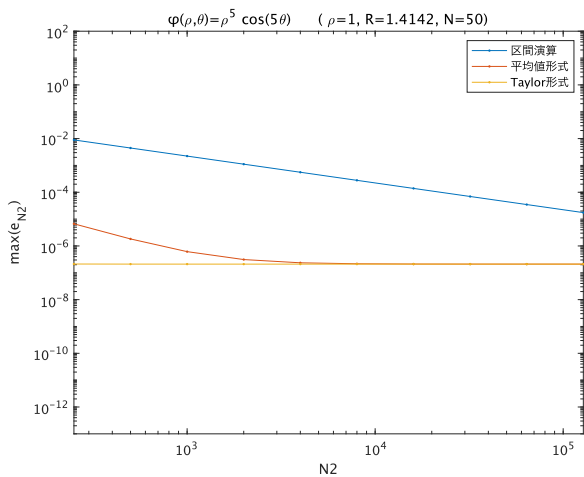


図 21: 実行結果 ($R = \sqrt{2}$)

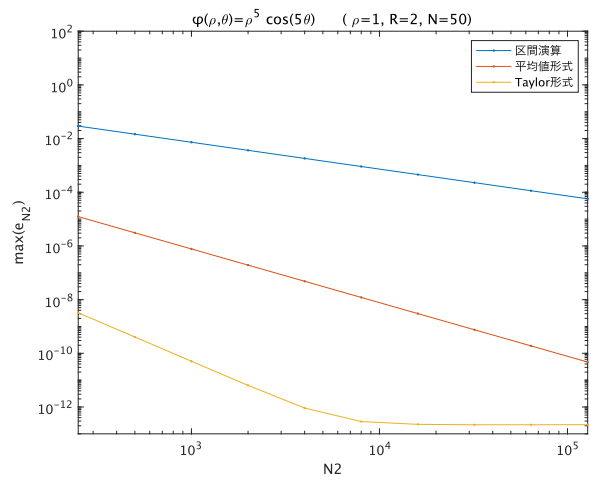


図 22: 実行結果 ($R = 2$)

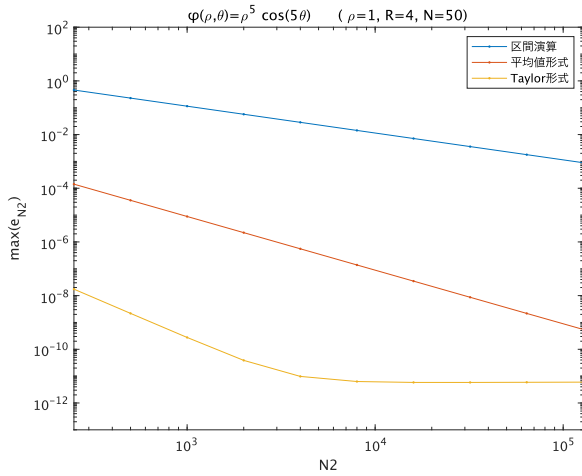


図 23: 実行結果 ($R = 4$)

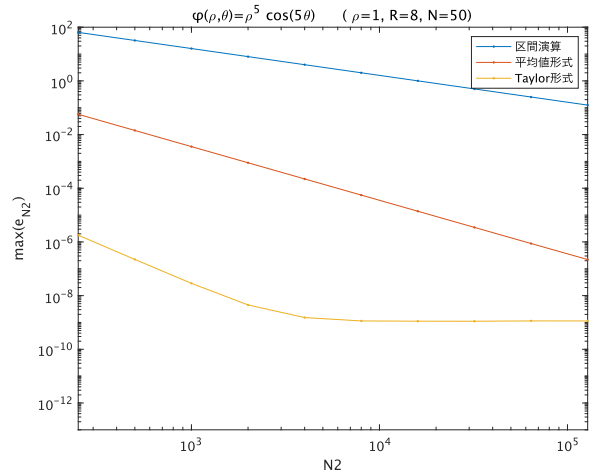


図 24: 実行結果 ($R = 8$)

以上の結果から、 $N_2 = 10^3$ としたとき、平均値形式では区間演算の時に比べて誤差が 10^{-4} 程度となることが確認できた。また、Taylor 形式において誤差の値が速く収束した $R = \sqrt{2}$ の場合を除いて、 $N_2 = 10^3$ としたときには、Taylor 形式では平均値形式の時に比べて誤差が $10^{-5} \sim 10^{-6}$ 程度となることが確認できた。また、今回扱った方法の中では 2 階 Taylor 形式が最良であることや、今回試した電荷点を配置する円領域の半径 R の中では $R = 2$ の時に誤差が収束する値が最小となることも示された。また、各方法による計算にかかった時間をグラフによって表すと以下のようなになった。

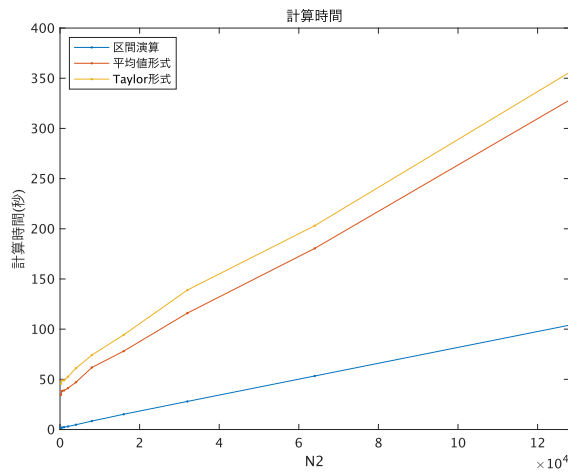


図 25: 各方法による計算に必要となる時間

この結果から、平均値形式や Taylor 形式を用いた計算では素朴な区間演算による計算に比べて多くの時間を必要とする一方で、Taylor 形式を用いた計算と平均値形式を用いた計算については必要とする時間に大差はないことが分かった。

3 まとめと課題

今回の卒業研究では主に桂田 [1]、大畑 [4] に沿って円盤領域における Laplace 方程式の Dirichlet 境界値問題に対して、区間演算や平均値形式、Taylor 形式を用いることで精度保証を行った。

境界条件を与える関数 $\varphi(x)$ について、その導関数を与えることで近似解の精度保証を行うことは出来たが、多くの関数 $\varphi(x)$ に対して実験を行うことは出来ていない。今後の課題は自動微分を用いることでより多くの $\varphi(x)$ に対しての数値実験を行うことや、Taylor 形式においてより高階の導関数を用いた計算を行うこと、円盤領域以外の領域 Ω に対する数値実験を行うことが挙げられる。

4 プログラム

2.3節の数値実験に使用した MATLAB プログラムを載せる。プログラム 4.1 はコマンドラインに関数名を入力することで、プログラム 4.2~プログラム 4.5 は INTLAB を起動した後にコマンドラインに関数名を入力することでプログラムが実行される。各プログラムの実行結果については 2.3.1~2.3.5 に示している。

4.1 近似解の計算, ランダムな点における誤差の計算

```
1  function FSM()
2  rho=1;
3  R=2;
4  kappa=0;
5
6  Nmin=4;
7  Nstep=2;
8  Nmax=60;
9
10 Nlist=[Nmin:Nstep:Nmax];
11 N2=100; % 境界上にランダムに取る点の個数
12
13 for m=0:5
14     en=zeros((Nmax-Nmin)/Nstep+1,1);
15     for N=Nmin:Nstep:Nmax
16
17         theta=transpose(linspace(0,2*pi*(N-1)/N,N));
18         omega=exp(1i*theta);
19         y=R*omega;
20         delta=exp(1i*kappa*(2*pi/N));
21         x=rho*delta*omega;
22
23         A=log(abs(x-transpose(y)));
24         b=varphi(rho,m,theta);
25         Q=A\b;
26
27         % 境界上に N2 個の点を取り、その点を用いて誤差の最大値を見積もる
28         max=0;
29
30         for j=1:N2
31             tt=rand()*2*pi;
32             u=varphi(rho,m,tt);
33             tx=rho*delta*exp(1i*tt);
34             A2=log(abs(tx-transpose(y)));
35             un=A2*Q;
```

```

36         if(abs(un-u)>max)
37             max=abs(un-u);
38         end
39     end
40
41     en((N-Nmin)/Nstep+1)=max;
42
43 end
44 semilogy(Nlist,en,'- .')
45 hold on;
46 end
47 legend('m=0','m=1','m=2','m=3','m=4','m=5')
48 title([' $\phi(\rho,m,\theta)=\rho^m \cos(m\theta)$ ',...
49       ' (  $\rho=$ ,num2str(rho), ...
50       ', R=',num2str(R),' )'])
51 ylim([1e-16 1e+2])
52 xlabel('N')
53 ylabel('max|u_N-u|')
54 hold off;
55 end
56
57 function val=varphi(rho,m,theta)
58 % 境界条件の関数 cos,sin(r,m,  $\theta$ )
59     val=rho^m*cos(m*theta);
60     %val=rho^m*sin(m*theta);
61 end

```

4.2 近似解の計算, 区間演算を用いた誤差の計算

```

1  function FSM_IA1()
2  rho=1;
3  R=2;
4  kappa=0;
5
6  Nmin=30;
7  Nstep=2;
8  Nmax=100;
9
10 Nlist=[Nmin:Nstep:Nmax];
11
12 for m=0:5
13     en=zeros((Nmax-Nmin)/Nstep+1,1);
14     for N=Nmin:Nstep:Nmax
15

```



```

16         disp(['m=', num2str(m), ', N=', num2str(N)]);
17
18         % 基本解の方法で Q を求める
19         theta=transpose(linspace(0, 2*pi*(N-1)/N, N));
20         omega=exp(1i*theta);
21         y=R*omega;
22         delta=exp(1i*kappa*(2*pi/N));
23         x=rho*delta*omega;
24
25         A=log(abs(x-transpose(y)));
26         b=varphi(rho, m, theta);
27         Q=A\b;
28
29         % 区間演算で誤差の最大値を見積もる
30         theta=[theta; 2*pi];
31         Ij=infsup(theta(1:N), theta(2:N+1));
32
33         u=varphi(rho, m, Ij);
34         tx=rho*delta*exp(1i*Ij);
35         A2=log(abs(tx-transpose(y)));
36         un=A2*Q;
37
38         err=max(mag(u-un));
39         en((N-Nmin)/Nstep+1)=err;
40
41     end
42     semilogy(Nlist, en, '- .')
43     hold on;
44 end
45 legend('m=0', 'm=1', 'm=2', 'm=3', 'm=4', 'm=5')
46 title([' $\phi(\rho, m, \theta) = \rho^m \cos(m\theta)$ ', ...
47       ' ( $\rho =$ , num2str(rho), ...
48       ',  $R =$ , num2str(R), ')'])
49 ylim([1e-2 1e+6])
50 xlabel('N')
51 ylabel('max(mag(u_N-u))')
52 hold off;
53 end
54
55 function val=varphi(rho, m, theta)
56 % 境界条件の関数 cos, sin(r, m,  $\theta$ )
57     val=rho^m*cos(m*theta);
58     %val=rho^m*sin(m*theta);
59 end

```

4.3 近似解の計算, より狭い区間での区間演算を用いた誤差の計算

```
1 function FSM_IA2()
2 format long;
3
4 rho=1;
5 R=2;
6 kappa=0;
7
8 N2min=250;
9 N2max=128000;
10 Nnum=log2(N2max/N2min)+1;
11
12 N2list=zeros(Nnum,1);
13 IA2time=zeros(Nnum,1);
14
15 N=50;
16
17 en=zeros(Nnum,1);
18
19 % 基本解の方法でQを求める
20 theta=transpose(linspace(0,2*pi*(N-1)/N,N));
21 omega=exp(1i*theta);
22 y=R*omega;
23 delta=exp(1i*kappa*(2*pi/N));
24 x=rho*delta*omega;
25
26 A=log(abs(x-transpose(y)));
27 b=varphi(rho,theta);
28 Q=A\b;
29
30 % [0,2pi] を N*N2 等分した区間を用いて誤差の最大値を見積もる
31 N2=N2min;
32 theta=[theta; 2*pi];
33 for l=1:Nnum
34
35     disp(['N2=',num2str(N2)]);
36
37     tic
38     maxerr=0;
39     for j=1:N
40         Ij2=transpose(linspace(theta(j),theta(j+1),N2+1));
41         Ij=infsup(Ij2(1:N2),Ij2(2:N2+1));
42         tx=rho*delta*exp(1i*Ij);
```

```

43     u=varphi(rho,Ij);
44     A2=log(abs(tx-transpose(y)));
45     un=A2*Q;
46     err=max(mag(u-un)); % I_{j,k}(kのみ動く)での最大値
47     if err>maxerr
48         maxerr=err; % 今までのjでの誤差の最大値より大きければ更新
49     end
50 end
51 IA2time(log2(N2/N2min)+1)=toc;
52 en(log2(N2/N2min)+1)=maxerr;
53 N2list(log2(N2/N2min)+1)=N2;
54 N2=N2*2;
55
56 end
57
58 loglog(N2list,en,'- .')
59 xlim([0 128000])
60 title(['\phi(\rho,\theta)=\rho^5 \cos(5\theta)',...
61       '(\rho=',num2str(rho),...
62       ', R=',num2str(R),', N=',num2str(N),')'])
63 xlabel('N2')
64 ylabel('max(e_{N2})')
65 end
66
67 function val=varphi(rho,theta)
68 % 境界条件の関数
69 val=rho^5*cos(5*theta);
70 end

```

4.4 近似解の計算, 平均値形式を用いた誤差の計算

```

1  function FSM_MEAN()
2  format long;
3
4  r=1;
5  R=2;
6  kappa=0;
7
8  N2min=250;
9  N2max=128000;
10 Nnum=log2(N2max/N2min)+1;
11
12 N2list=zeros(Nnum,1);
13 meantime=zeros(Nnum,1);

```

```

14
15 N=50;
16
17 en=zeros(Nnum,1);
18
19 % 基本解の方法でQを求める
20 theta=transpose(linspace(0,2*pi*(N-1)/N,N));
21 omega=exp(1i*theta);
22 y=R*omega;
23 delta=exp(1i*kappa*(2*pi/N));
24 x=r*delta*omega;
25
26 A=log(abs(x-transpose(y)));
27 b=phi(r,theta);
28 Q=A\b;
29
30 % [0,2pi] を N*N2 等分した区間で平均値形式を用いて誤差の最大値を見積もる
31 N2=N2min;
32 theta=[theta; 2*pi];
33 for l=1:Nnum
34
35     disp(['N2=',num2str(N2)]);
36
37     tic
38     maxerr=0;
39     for j=1:N
40         %disp(['j=',num2str(j)]);
41         Ij2=transpose(linspace(theta(j),theta(j+1),N2+1));
42         Ij=infsup(Ij2(1:N2),Ij2(2:N2+1));
43         midtheta=mid(Ij);
44         radtheta=rad(Ij);
45         un1=zeros(N2,1);
46         for j2=1:N
47             un1=un1...
48                 +Q(j2)*log(sqrt(r^2+R^2-2*r*R*cos(midtheta-theta(j2))));
49         end
50         f1=un1-phi(r,midtheta);
51         un2=zeros(N2,1);
52         for j2=1:N
53             un2=un2...
54                 +(-Q(j2)*r*R*sin(theta(j2)-Ij))...
55                 ./ (r^2+R^2-2*r*R*cos(theta(j2)-Ij));
56         end
57         f2=un2-dphi(r,Ij);

```

```

58         fmean=f1+f2.*infsup(-radtheta,radtheta);
59         err=max(mag(fmean));
60         if err>maxerr
61             maxerr=err; % 今までの j での誤差の最大値より大きければ更新
62         end
63     end
64
65     meantime(log2(N2/N2min)+1)=toc;
66     en(log2(N2/N2min)+1)=maxerr;
67     N2list(log2(N2/N2min)+1)=N2;
68     N2=N2*2;
69
70 end
71
72 %xlim([N2min N2max])
73 loglog(N2list,en,'- .')
74 title(['  $\phi$  ( $\rho, \theta$ )= $\rho^5 \cos(5\theta)$ '],...
75       '      (  $\rho$ =' ,num2str(r), ...
76       ' , R=' ,num2str(R), ' , N=' ,num2str(N), ' )'])
77 xlabel('N2')
78 ylabel('max(e_{N2})')
79 end
80
81 function val=phi(r,theta)
82 val=r^5*cos(5*theta);
83 end
84 function val=dphi(r,theta)
85 val=-5*r^5*sin(5*theta);
86 end

```

4.5 近似解の計算, Taylor 形式を用いた誤差の計算

```

1  %function FSM_Taylor()
2  format long;
3
4  r=1;
5  R=2;
6  kappa=0;
7
8  N2min=250;
9  N2max=128000;
10 Nnum=log2(N2max/N2min)+1;
11
12 N2list=zeros(Nnum,1);

```

```

13  taylortime=zeros(Nnum,1);
14
15  N=50;
16
17  en=zeros(Nnum,1);
18
19  % 基本解の方法でQを求める
20  theta=transpose(linspace(0,2*pi*(N-1)/N,N));
21  omega=exp(1i*theta);
22  y=R*omega;
23  delta=exp(1i*kappa*(2*pi/N));
24  x=r*delta*omega;
25
26  A=log(abs(x-transpose(y)));
27  b=phi(r,theta);
28  Q=A\b;
29
30  % [0,2pi] を N*N2 等分した区間を用いて誤差の最大値を見積もる
31  N2=N2min;
32  theta=[theta; 2*pi];
33  for l=1:Nnum
34
35      disp(['N2=',num2str(N2)]);
36
37      tic
38      maxerr=0;
39      for j=1:N
40          %disp(['j=',num2str(j)]);
41          Ij2=transpose(linspace(theta(j),theta(j+1),N2+1));
42          Ij=infsup(Ij2(1:N2),Ij2(2:N2+1));
43          midtheta=mid(Ij);
44          radtheta=rad(Ij);
45          un1=zeros(N2,1);
46          for j2=1:N
47              un1=un1...
48                  +Q(j2)*log(sqrt(r^2+R^2-2*r*R*cos(midtheta-theta(j2))));
49          end
50          f1=un1-phi(r,midtheta);
51          un2=zeros(N2,1);
52          for j2=1:N
53              un2=un2...
54                  +(-Q(j2)*r*R*sin(theta(j2)-midtheta))...
55                  ./(r^2+R^2-2*r*R*cos(theta(j2)-midtheta));
56          end

```

```

57     f2=un2-dphi(r,midtheta);
58     un3=zeros(N2,1);
59     for j2=1:N
60         un3=un3...
61             +Q(j2)*((r^2+R^2)*r*R*cos(Ij-theta(j2))-2*r^2*R^2)...
62             ./((r^2+R^2-2*r*R*cos(Ij-theta(j2)))).^2;
63     end
64     f3=un3-ddphi(r,Ij);
65     ftaylor=f1+(f2+f3/2.*infsup(-radtheta,radtheta))...
66         .*infsup(-radtheta,radtheta);
67     err=max(mag(ftaylor));
68     if err>maxerr
69         maxerr=err; % 今までの j での誤差の最大値より大きければ更新
70     end
71 end
72
73 taylor_time(log2(N2/N2min)+1)=toc;
74 en(log2(N2/N2min)+1)=maxerr;
75 N2list(log2(N2/N2min)+1)=N2;
76 N2=N2*2;
77
78 end
79
80 loglog(N2list,en,'- .')
81 title([' $\phi(\rho, \theta) = \rho^5 \cos(5\theta)$ ',...
82     '    ( $\rho =$ ',num2str(r), ...
83     ',  $R =$ ',num2str(R), ',  $N =$ ',num2str(N),')'])
84 xlabel('N2')
85 ylabel('max(e_{N2})')
86 %end
87
88 function val=phi(r,theta)
89     val=r^5*cos(5*theta);
90 end
91 function val=dphi(r,theta)
92     val=-5*r^5*sin(5*theta);
93 end
94 function val=ddphi(r,theta)
95     val=-25*r^5*cos(5*theta);
96 end

```

参考文献

- [1] 桂田祐史, 代用電荷法に関する一考察, 東京大学大学院理学系研究科数学専攻修士論文, 1986.
- [2] 大石進一, 精度保証付き数値計算, コロナ社, 2000.
- [3] G. I. Hargreaves, Interval Analysis in MATLAB, Numerical Analysis Report No.416, Department of Mathematics, University of Manchester, 2002.
- [4] 大畑佑樹, 楕円型偏微分方程式に対する精度保証付き数値計算, 明治大学大学院先端数理科学研究科 2019 年度修士学位請求論文, 2020.