

計算物理学

2005 年 8 月 22 日

概要

このテキストは学部生が自身の卒論テーマを取り扱う時に、必要とされる最も基本的な内容についてまとめたものである。しかし、このテキストの内容だけでは不十分であり、自主的に知識を深める努力が求められる。その際、このテキストをベースにいただければ幸いである。

内容は、4年までに取得したであろうコンピュータ解析の講義内容の復習から入る。理論的、プログラムの簡単なものから取り扱い、卒論から修士レベルの内容まで取り扱っていく。形式は講義と同様に、問題を出題する形で進めていく。学生はこの問題に取り組みながら、勉強を進めればよい。

テキストは理論よりも、理論をどのようにプログラム化するかに重きを置いている。理論を取り扱った本は山のように出ているが、実際にプログラム化するために書かれた本は、あまりないように感じたからだ。

目次

第 1 章	N 次方程式 (非線形方程式) の解法	2
1.1	2 分法 (Bitwise Chopping method)	2
1.2	ニュートン法 (Newton method)	5
第 2 章	数値積分法	8
2.1	台形公式 (Trapezoid rule)	8
2.2	シンプソン公式 (Simpson rule)	11
第 3 章	常微分方程式の解法	13
3.1	オイラー法 (Euler method)	13
3.2	ルンゲ・クッタ法 (Runge-Kutta method)	16
第 4 章	物理シミュレーション 1	19
4.1	質点の運動	19
4.2	波動方程式	22
4.3	固有値問題	25
第 5 章	連立方程式の解法	28
5.1	ガウスの消去法	28
5.2	ガウス・ザイデル法	32
第 6 章	物理シミュレーション 2	35
6.1	ポアソン方程式	35

第 1 章

N 次方程式 (非線形方程式) の解法

1.1 2 分法 (Bitwise Chopping method)

まずは 2 分法からはじめよう。次のような 1 変数非線形方程式の解 x を数値的に求めるには、どうすればよいだろうか。

$$f(x) = 0 \quad (1.1)$$

関数 $f(x)$ は区間 $a \leq x \leq b$ で連続、 $a < b$ かつ $f(a)f(b) < 0$ であれば、少なくとも 1 つは $f(x) = 0$ を満たす解が存在することが予想できる。 a と b との midpoint $c = (a + b)/2$ の場合について見ても、 $f(c)f(a) < 0$ であれば区間 (a, c) の内に、 $f(c)f(b) < 0$ であれば区間 (c, b) の内に少なくとも 1 つは解が存在することは中間値の定理からも明らかである。2 分法はこのような解の存在する区間を $1/2$ ずつに縮小しながら解 x を誤差 ϵ の範囲で求める方法である。

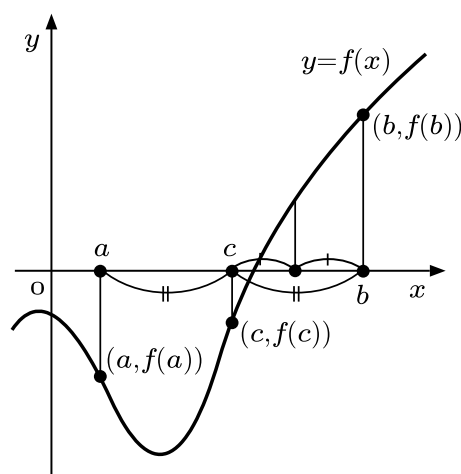


Fig.1.1: 2 分法

2 分法は、関数 $f(x)$ が連続でさえあれば、どんな関数でも収束する強力な方法であるが、その収束は他のスキームに比べ遅い。

1.1.1 2分法のアルゴリズム

1. 適当な初期値 a と b を決める。ここで $a < b$ であり $f(a)f(b) < 0$ 、また $f(x)$ は区間 $a \leq x \leq b$ で連続であるとする。
2. 中点 $c = (a+b)/2$ を求める。 $f(c) \leq 0$ なら a を c と入れ換える ($a = c$)。 $f(c) > 0$ なら b を c と入れ換える ($b = c$)。
3. 各値を表示する。
4. 計算回数が 100 回に達した場合は、“not convergent” と表示して終了する。
5. $|b - a|$ を求め設定した要求精度 ϵ (10^{-5}) よりも小さくなったら処理を終了する。
6. 4 と 5 に該当しない限りは、2 と 3 をくり返し計算する。

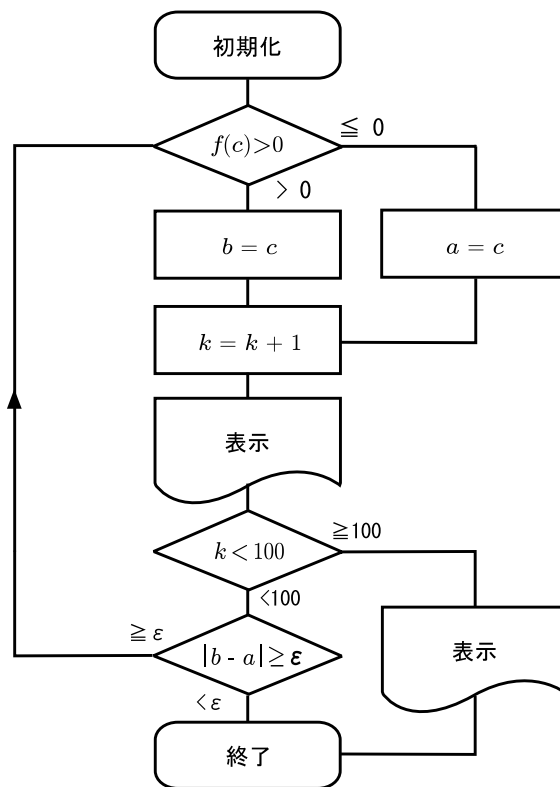


Fig.1.2: 2分法のフローチャート

ここに載せたフローチャート以外にも様々なフローチャートが考えられる。作成するプログラムは問題の要求を満たしていれば良い。どうしてもよいか分からない人は、上記のフローチャートを参考にしてプログラムを作成すること。

1.1.2 2分法の問題

$f(x) = x - \sin(\pi x)$, $a = 0.5, b = 1.0, \epsilon = 10^{-5}$ として、2分法を用いて $f(x) = 0$ の解を求めよ。反復ごとに各変数を出し、値が収束している様子を観察せよ。要求精度を $\epsilon = 10^{-5}$ とし $|b - a| < \epsilon$ となったら計算を終了すること。また、100回計算しても収束しなかった場合は “not convergent” と表示して終了するようにすること。

n	a	b	$f(a)$	$f(b)$
1	0.500000	0.750000	-0.500000	0.042893
2	0.625000	0.750000	-0.298880	0.042893
3	0.687500	0.750000	-0.143970	0.042893
4	0.718750	0.750000	-0.054261	0.042893
5	0.734375	0.750000	-0.006576	0.042893
6	0.734375	0.742188	-0.006576	0.017940
7	0.734375	0.738281	-0.006576	0.005627
8	0.736328	0.738281	-0.000489	0.005627
9	0.736328	0.737305	-0.000489	0.002565
10	0.736328	0.736816	-0.000489	0.001037
11	0.736328	0.736572	-0.000489	0.000274
12	0.736450	0.736572	-0.000107	0.000274
13	0.736450	0.736511	-0.000107	0.000083
14	0.736481	0.736511	-0.000012	0.000083
15	0.736481	0.736496	-0.000012	0.000036
16	0.736481	0.736488	-0.000012	0.000012

Tab.1.1: 2分法の問題の計算結果

実行すると計算回数 16 回、 $a = 0.736481, b = 0.736488$ 程度に収束しているはずである。実行結果は、計算環境によって若干変わってくるため参考程度に留めること。

1.2 ニュートン法 (Newton method)

ニュートン法は、非線形方程式 $f(x) = 0$ の解法として最もよく知られた強力な方法の1つである。ニュートン法は $f(x)$ を Taylor 展開することで得られる。 $x = x_n$ と置き、 $f(x)$ を展開し、2次以上の項を無視すると、 $f(x) = 0$ の近似方程式として $f(x_n) + f'(x_n)(x - x_n) = 0$ を得る。よって、 $f'(x_n) \neq 0$ ならば、近似解を求める次の反復式を得る。

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (1.2)$$

ここで $f'(x_n)$ は $f(x_n)$ の x 微分である。ニュートン法では各反復ごとに微分値 $f'(x_n)$ を求める必要がある。そのため何らかの理由により、微分値が求まらない、求めにくい場合はニュートン法は不向きである。そこで各反復ごとに微分値を必要としないセカント法なども提案されている。

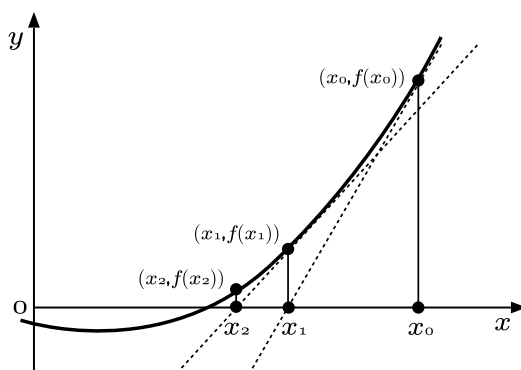


Fig.1.3: ニュートン法

(4.1) 式の反復式により、 $f(x) = 0$ の解に収束する点列 $\{x_n\}$ を生成する方法がニュートン法である。Fig.1.3 のように、適切な初期値 x_0 を選ぶ。すると $(x_0, f(x_0))$ の接線と x 軸との交点 x_1 が求まる。この反復計算を繰り返すことで、 x_n は解 x に収束していくことになる。

1.2.2 ニュートン法の問題

ニュートン法を用いて次の方程式の解を求めよ。

$$f(x) = x^5 + 6x^4 - 3x^3 + x^2 - 5 = 0$$

ただし、 $\epsilon = 10^{-5}$ とする。反復ごとに、現在の回数 n とそのときの解 x_n を表示するようにせよ。また、計算回数が 100 回に達した場合には、“not convergent” と表示し終了するようにすること。

n	x_n	$f(x_n)$
1	3.896785	2114.690403
2	3.031590	683.467245
3	2.357802	219.534882
4	1.839774	69.520768
5	1.453447	21.163660
6	1.188562	5.721480
7	1.045306	1.077666
8	1.003273	0.072407
9	1.000018	0.000405
10	1.000000	0.000000
11	1.000000	0.000000

Tab.1.2: ニュートン法の問題の計算結果

初期値 x_0 が $x_0 = 5.0$ の場合、繰り返し回数が 11 回となり、解 x は $x_{11} = 1.000000$ となる。2 分法との収束速度や精度を比較すると面白いかもしれない。

第2章

数値積分法

2.1 台形公式 (Trapezoid rule)

次にある関数 $f(x)$ を区間 $[a, b]$ に渡って数値的に積分することを考えよう。特に積分は解析的に解けないものがほとんどと言ってよく。数値的に積分を行うことがしばしば必要となってくるわけである。まず、最も簡単な台形公式について考えていく。

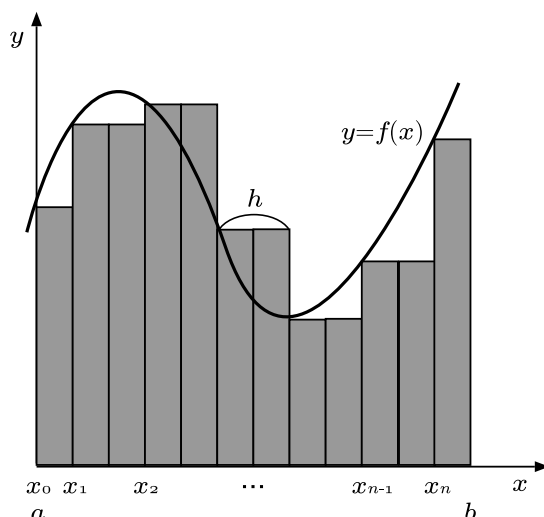


Fig.2.1: 台形公式

台形公式では、区間 $[a, b] = [x_0, x_n]$ における定積分を幅 h の n 個の長方形に近似し、積分値を見積もる方法である。

$$I = \int_a^b f(x)dx = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x)dx \quad (2.1)$$

とする。ここで $x_{k+1} = x_k + h$ である。次に点 x_k, x_{k+1} において $f(x)$ がとる値を y_k, y_{k+1} とすると、1 次の Taylor 展開より (2.1) 式は次のように書き直せる。

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx = \int_{x_k}^{x_{k+1}} \{y_k - y'(x - x_k)\}dx = \frac{h}{2}(y_k + y_{k+1}) \quad (2.2)$$

(2.1), (2.2) 式より

$$I = \sum_{k=0}^{n-1} I_k = \sum_{k=0}^{n-1} \frac{h}{2} (y_k + y_{k+1}) \quad (2.3)$$

となり、台形公式が以下のように求まる。

$$I = \frac{h}{2} (y_0 + 2y_1 + 2y_2 + \cdots + 2y_{n-1} + y_n) \quad (2.4)$$

2.1.1 台形公式のアルゴリズム

1. 積分範囲 $[a, b]$ 、要求精度 ϵ を決める。
2. n のループ文を組む。 n は 1 ~ 100 まで 1 ずつ変化する。
 1. $h = (b - a) / 2n$, $S = 0$
 2. i のループ文を組む。 i は 1 ~ n まで 1 ずつ変化する。
 1. $s = s + y_i$
 3. $s = y_0 + y_n$, $s = s \times (h/2)$ を計算する。また、各値を表示する。
 4. $n \neq 1$ のとき、 $|SS - S| < \epsilon$ ならループを抜ける。
 5. $SS = S$
3. もし n が 100 に達していれば、“not convergent” と表示して終了する。

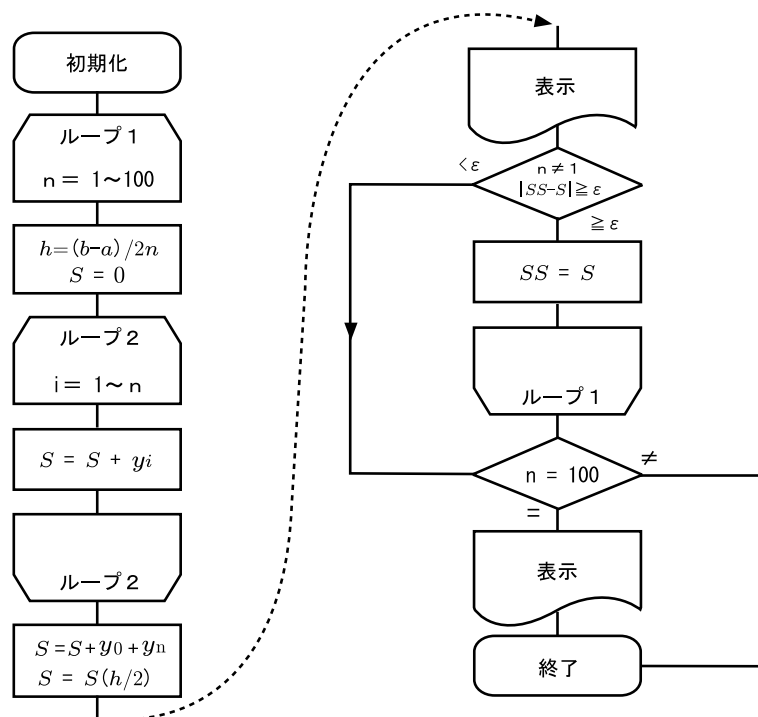


Fig.2.2: 台形公式のフローチャート

2.1.2 台形公式の問題

台形公式を用いて次の関数 $f(x)$ の数値積分を行え。

$$f(x) = \exp(-x^2/2) \quad (0.5 \leq x \leq 1.0)$$

ただし、要求精度 $\epsilon = 10^{-5}$ とし、積分区間を $2n$ に等分したときの積分値を S_n とした時、 $|S_n - S_{n+1}| < \epsilon$ となったら止まるようにすること。また、 $n > 100$ に達した場合は “not convergent” と表示して終了する。

n	h	S_n
1	0.250000	0.374838
2	0.125000	0.375484
3	0.083333	0.375604
4	0.062500	0.375645
5	0.050000	0.375665
6	0.041667	0.375675
7	0.035714	0.375682

Tab.2.1: 台形公式の問題の計算結果

実行すると以下のような結果が得られるはずである。等分数 n が増えるにつれ積分値 S_n が収束していく様子が見てとれる。 $n = 7$ のとき、前積分値との差が 10^{-5} 未満であるため処理を終了している。

2.2 シンプソン公式 (Simpson rule)

台形公式の導出式において、taylor 展開を 2 次まで求めればシンプソン公式を導くことができる。シンプソン公式では面積は次のように求まる。

$$I = \frac{h}{3}(y_0 + 4y_1 + 2y_2 + \cdots + 4y_{n-1} + y_n) \quad (2.5)$$

台形公式では、各分割領域での関数値を 1 次近似 (直線近似) によって積分値を求めているが、シンプソン公式ではもう少し精度良く 2 次近似 (曲線近似) によって求めているわけである。

2.2.1 シンプソン公式のアルゴリズム

1. 積分範囲 $[a, b]$ 、要求精度 ϵ を決める。
2. n のループ文を組む。 n は 1 ~ 100 まで 1 ずつ変化する。
 1. $h = (b - a) / 2n$, $S = 0$
 2. i のループ文を組む。 i は 1 ~ $2n$ まで 2 ずつ変化する。
 1. $s = s + 4y_i$
 3. i のループ文を組む。 i は 2 ~ $2n$ まで 2 ずつ変化する。
 1. $s = s + 2y_i$
 4. $s = y_0 + y_n$, $s = s \times (h/3)$ を計算する。また、各値を表示する。
 5. $n \neq 1$ のとき、 $|SS - S| < \epsilon$ ならループを抜ける。
 6. $SS = S$
3. もし n が 100 に達していれば、“not convergent” と表示して終了する。

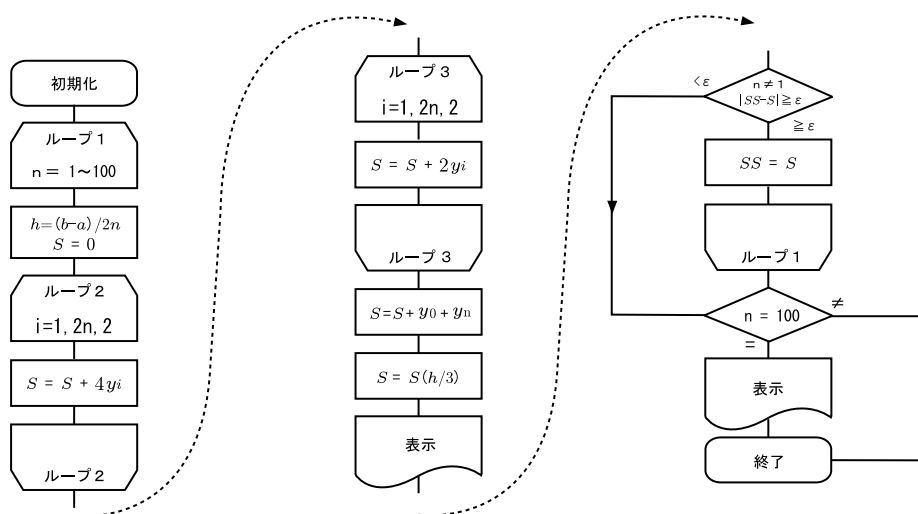


Fig.2.3: シンプソン公式のフローチャート

2.2.2 シンプソン公式の問題

シンプソン公式を用いて、次の関数 $f(x)$ の数値積分を行い、真値と比較せよ。

$$f(x) = \frac{x}{1+x^2} e^x \quad (0 \leq x \leq 1)$$

この積分の真値は $e^1/2 - 1$ となる。要求精度は $\epsilon = 10^{-5}$ とし、積分区間を $2n$ 等分したときの数値積分値を S_n としたとき、 $|S_n - S_{n+1}| < \epsilon$ となったら止まるようなプログラムを作成せよ。 $n > 100$ に達した場合は、“not convergent” と表示し終了する。

n	h	S_n	$ S^* - S_n $
1	0.500000	0.357517	0.001624
2	0.250000	0.358992	0.000149
3	0.166667	0.359108	0.000032
4	0.125000	0.359130	0.000011
5	0.100000	0.359136	0.000004

Tab.2.2: シンプソン公式の問題の結果

ここで S^* は真値を表しており、反復回数を増すごとに真値との差が小さくなっているのが分かる。また、台形公式に比べ少ない計算回数で要求精度を満たす数値解を得ることができた。

第 3 章

常微分方程式の解法

3.1 オイラー法 (Euler method)

次に常微分方程式 $dy/dx = f(x, y)$ を初期条件 $y(x_0) = y_0$ のもとで数値的に解くことを考える。実際にコンピュータが取り扱える計算は四則演算だけなので、微分や積分といった概念を直接取り扱うことはできない。そこで、何らかの方法を用いて微分積分を近似することを考える。オイラー法では、微分の定義に戻り幅 h を有限な大きなで近似する。

$$\frac{dy}{dx} = \lim_{h \rightarrow 0} \frac{y(x+h) - y(x)}{h} \simeq \frac{y(x+h) - y(x)}{h} \quad (h \ll 1) \quad (3.1)$$

以上から下式を得る。

$$y_{n+1} = y_n + hf(x_n, y_n) \quad (3.2)$$

オイラー法は前進差分とも呼ばれ、1 次の精度をもつ最も基本的な数値解法の 1 つである。(3.2) 式から常微分方程式を数値的に解く。Fig.3.1 から分かるようにオイラー法では解曲線（点線）に対して折れ線（実線）を用いて近似していく。

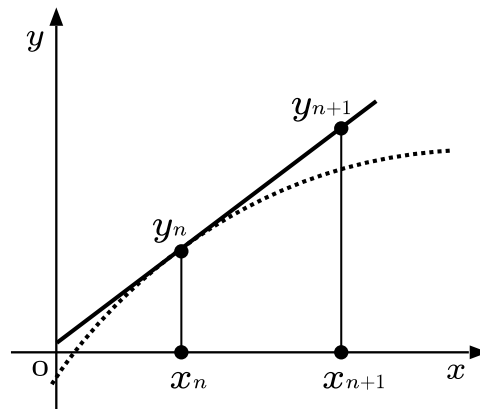


Fig.3.1: オイラー法

3.1.1 オイラー法のアルゴリズム

1. 初期値 (x_0, y_0) を代入する。
2. $f(x, y)$ を計算する。
3. y_{n+1} を計算する。
4. 各値を出力する。
5. $x = x + h$ を計算する。
6. y_n, y_{n+1} の変数交換
7. 条件を満たしていたら計算を終了する。
8. そうでない場合は、2に戻り計算を繰り返す。

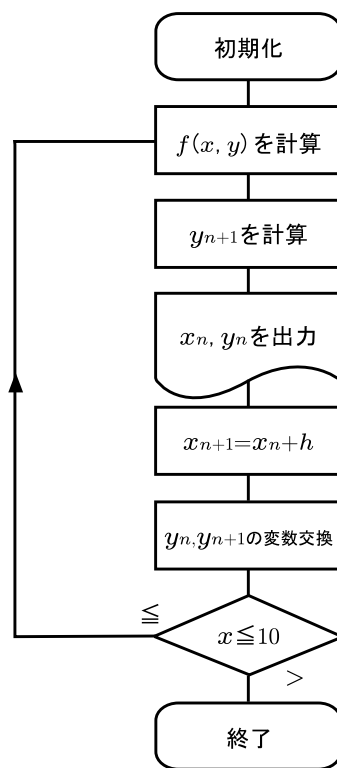


Fig.3.2: オイラー法のフローチャート

3.1.2 オイラー法の問題

常微分方程式 $dy/dx = 1 + y$ において、初期条件 $x_0 = 1, y_0 = 0$ であるとき、オイラー法を用いて区間 $[1, 10]$ の間で数値的に解け。刻み幅 h は $h = 0.1$ とする。この問題の厳密解は $y = \exp(x - 1) - 1$ で与えられる。数値解と厳密解との比較も行え。

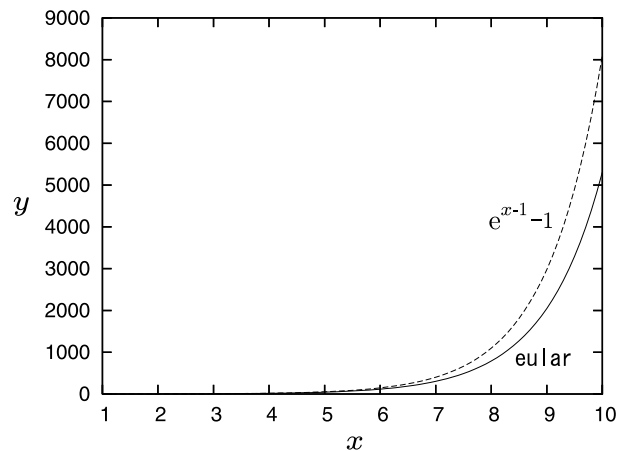


Fig.3.3: オイラー法の結果

計算の初期段階では、厳密解とオイラー法による数値解は大差ないが、後半に入ると累積した誤差は無視できない大きさとなっているのが分かる。この場合、厳密解と良い一致を得るためには刻み幅 h をさらに細かくとる必要がある。刻み幅 h を細かくとると計算回数が増大することになり、何らかの実用的な計算を行う場合には、膨大な計算コストとなる。そういった意味でオイラー法は実用的ではないが、最も基本的な数値解法の1つであり、理解が容易なことから重要な解法であるといえる。

3.2 ルンゲ・クッタ法 (Runge-Kutta method)

4 次の精度をもつ数値解法であるルンゲ・クッタ法では次のように解く。

$$k_1 = hf(x_n, y_n) \quad (3.3)$$

$$k_2 = hf(x_n + h/2, y_n + k_1/2) \quad (3.4)$$

$$k_3 = hf(x_n + h/2, y_n + k_2/2) \quad (3.5)$$

$$k_4 = hf(x_n + h, y_n + k_3) \quad (3.6)$$

$$k_5 = (k_1 + 2k_2 + 2k_3 + k_4)/6 \quad (3.7)$$

$$y_{n+1} = y_n + k_5 \quad (3.8)$$

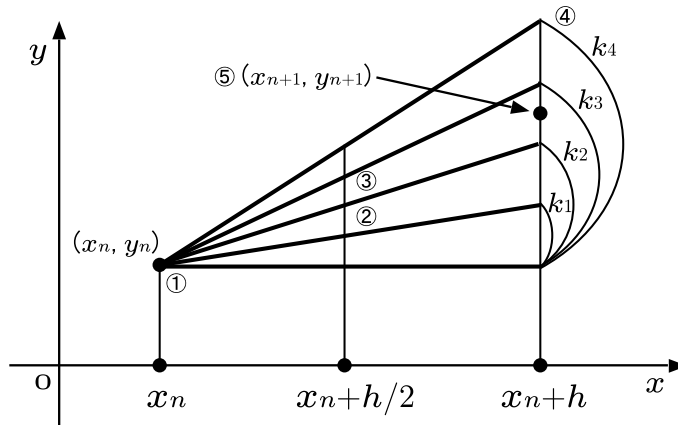


Fig.3.4: ルンゲ・クッタ法

上記の操作にどのような意味があるのか簡単に説明する。

1. 点 (x_n, y_n) から傾き $f(x_n, y_n)$ で x が h だけ進むと y は k_1 だけ変化する。
2. この線上の点 $(x_n + h/2, y_n + k_1/2)$ で傾きを求め、 (x_n, y_n) から h だけ進むと y は k_2 だけ変化する。
3. 同様に、 (x_n, y_n) から傾き $f(x_n + h/2, y_n + k_2/2)$ だけ h だけ進み、 k_3 を得る。
4. さらに (x_n, y_n) から $f(x_n + h, y_n + k_3)$ の傾きで h だけ進み、 k_4 を得る。
5. 以上、1~4 の加重平均として、 y_n に対する増分 k_5 を求め、点 (x_{n+1}, y_{n+1}) を得る。

この数値解法は、打ち切り誤差が小さく、また係数も簡単であることから数値計算法では良く用いられる方法の1つである。導出は Taylor 展開からなされるが、大変な仕事量である。また、オイラー法とルンゲ・クッタ法の間位置する2次の精度をもったホイーン法などもある。

3.2.1 ルンゲ・クッタ法のアルゴリズム

1. 初期値 (x_0, y_0) を代入する。
2. $k_1 \sim k_4$ を計算する。
3. y_{n+1} を計算する。
4. 各値を出力する。
5. $x = x + h$ を計算する。
6. y_n, y_{n+1} の変数交換
7. 条件を満たしていたら計算を終了する。
8. そうでない場合は、2に戻り計算を繰り返す。

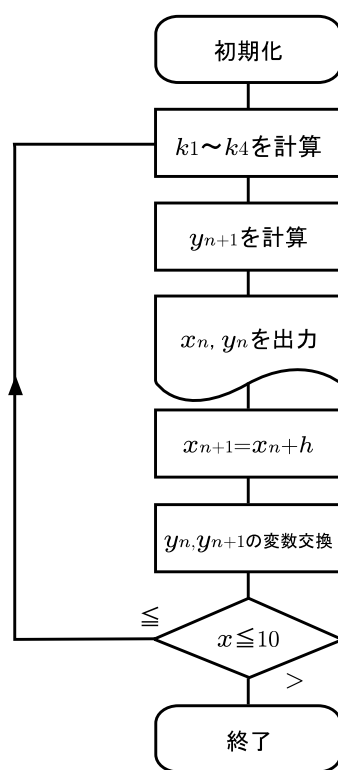


Fig.3.5: ルンゲ・クッタ法のフローチャート

3.2.2 ルンゲ・クッタ法の問題

常微分方程式 $dy/dx = 1 + y$ において、初期条件 $x_0 = 1, y_0 = 0$ であるとき、ルンゲ・クッタ法を用いて区間 $[1, 10]$ の間で数値的に解け。刻み幅 h は $h = 0.1$ とする。この問題の厳密解は $y = \exp(x - 1) - 1$ で与えられる。数値解と厳密解との比較も行え。

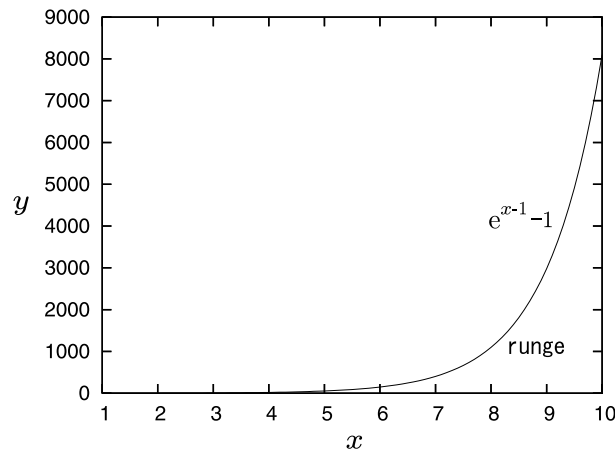


Fig.3.6: ルンゲ・クッタ法の結果

Fig.3.3 と比べても分かる通り、ルンゲ・クッタ法を用いた数値解と厳密解は良い一致を得ている。オイラー法の場合と同様の刻み幅 h でありながら、計算結果に大きな違いが出た。これはルンゲ・クッタ法の精度がオイラー法より高いことに依る。しかし、一般的に精度の高い数値解法は、計算のステップ数が増加する。また、計算に必要とされる変数の数も増加するため、一概に精度が良い数値解法を採用すれば良いというわけではない。

また、刻み幅 h も細かくとれば必ずしも良いわけではない。細かくとりすぎると桁落ち等の誤差が増大し精度が落ちるため、最適な h を求め採用するのが望ましい。

第 4 章

物理シミュレーション 1

4.1 質点の運動

今までの議論を踏まえ、質点の自由落下運動をシミュレーションすることを考える。質量 m の物体の重力のもとにおける運動方程式は次式で表される。

$$m \frac{d^2 y}{dt^2} = -mg \quad (4.1)$$

ここで、 g は重力加速度である。(4.1) 式は、次のように書き換えることができる。

$$\frac{dv_y}{dt} = -g \quad (4.2)$$

$$\frac{dy}{dt} = v_y \quad (4.3)$$

この位置、速度や時間を離散化し、方程式を差分化することで各時間毎に数値解を求めていく方法が数値シミュレーション方法である。ここでは、オイラー法を用いて (4.2) 式を差分化する。

$$\frac{v_y(t + \Delta t) - v_y(t)}{\Delta t} = -g \quad (4.4)$$

よって

$$v_y(t + \Delta t) = v_y(t) - g\Delta t \quad (4.5)$$

同様に (4.3) についても差分化する。ここで、 $v_y(t)$ を v_y^n 、 $y(t)$ を y^n と表記すると以下のように書き直せる。 n は離散化された時間ステップを表している。また、 Δt は時間の刻み幅である。

$$v_y^{n+1} = v_y^n - g\Delta t \quad (4.6)$$

$$y^{n+1} = y^n + v_y^n \Delta t \quad (4.7)$$

4.1.1 質点の運動のアルゴリズム

1. 定数 $\Delta t, g$ を設定する。
2. 位置、速度の計算
3. 各値を出力する。
4. $t = t + \Delta t$
5. 求めたい時刻 t に達したら終了
6. そうでない場合は、2 に戻り計算を繰り返す。

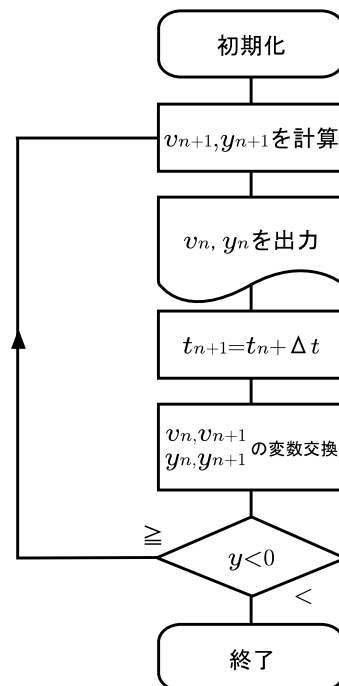
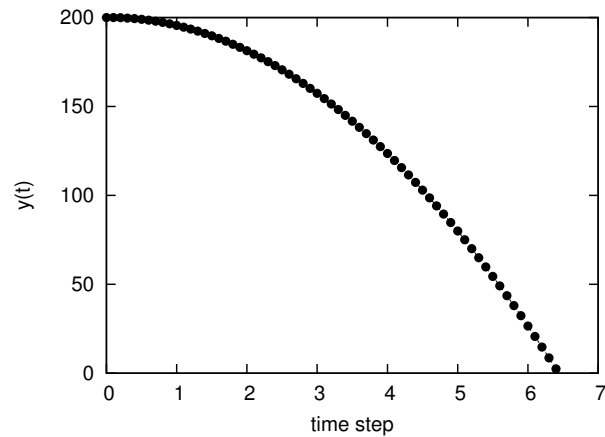
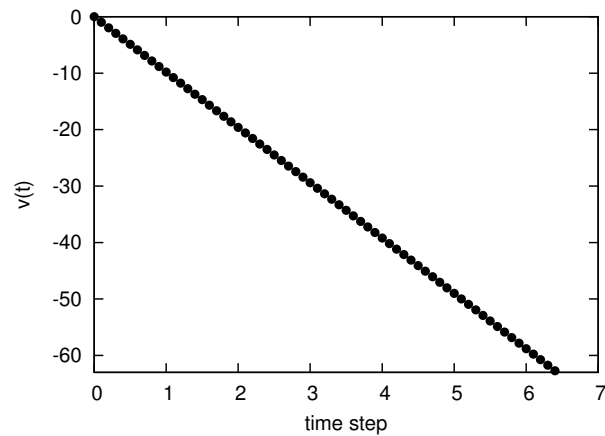


Fig.4.1: 質点の運動のフローチャート

4.1.2 質点の運動の問題

地面から高さ $h = 200$ の地点から物体を自由落下させたときの位置 ($y - t$) と速度 ($v_y - t$) を表示せよ。ここで、刻み幅 Δt は $\Delta t = 0.1$ とする。また、重力加速度 g は $g = 9.8$ とする。また、初期速度 v_0 は $v_0 = 0$ とする。質点が地面に衝突 ($y < 0$) したら処理を止めるようにする。

Fig.4.2: 質点の位置変化 ($y - t$)Fig.4.3: 質点の速度変化 ($v - t$)

質点の位置変化は2次関数的に変化しているのが分かる。また、質点の速度変化は加速度が一定なため直線的となる。余裕があれば、問題をさらに空気抵抗がある場合や外力が加わった場合などに拡張せよ。

4.2 波動方程式

双曲型偏微分方程式の中で最も簡単な方程式は、次の 1 次元波動方程式である。

$$\frac{\partial f}{\partial t} + c \frac{\partial f}{\partial x} = 0 \quad (4.8)$$

(4.4) 式を数値的に解く方法として次の方法がある。(4.4) 式の Taylor 展開を x と t について適用し、精度 $O(\Delta t^2), O(\Delta x^2)$ まで正確な差分を作ると

$$\frac{f(x, t + \Delta t) - f(x, t - \Delta t)}{2\Delta t} + c \frac{f(x + \Delta x, t) - f(x - \Delta x, t)}{2\Delta x} = 0 \quad (4.9)$$

を得る。この差分式を通常蛙跳び (leap frog) 法と呼ぶ。ここで、時間については 1 つ精度を落として

$$\frac{f(x, t + \Delta t) - f(x, t)}{2\Delta t} + c \frac{f(x + \Delta x, t) - f(x - \Delta x, t)}{2\Delta x} = 0 \quad (4.10)$$

を採用することにする。ここで、 $x = x_j = j\Delta x$, $t = t_n = n\Delta t$ とし、 $f(x, t)$ を f_j^n と表記すると (4.10) 式は以下のように書き直せる。

$$f_j^{n+1} = f_j^n - \frac{\chi}{2}(f_{j+1}^n - f_{j-1}^n) \quad (4.11)$$

ここで、 $\chi = c\Delta t/\Delta x$ である。この差分は一般的に中央差分法と呼ばれるものである。さらに精度を上げた 2 次精度の差分法として Lax-Wendroff 法がある。

$$f_j^{n+1} = f_j^n - \frac{\chi}{2}(f_{j+1}^n - f_{j-1}^n) + \frac{\chi^2}{2}(f_{j+1}^n - 2f_j^n + f_{j-1}^n) \quad (4.12)$$

また、(4.10) 式に対して次のような差分法を風上差分法と呼ぶ。

$$f_j^{n+1} = f_j^n - \chi'(f_j^n - f_{j-1}^n) \quad (c \geq 0) \quad (4.13)$$

$$f_j^{n+1} = f_j^n - \chi'(f_{j+1}^n - f_j^n) \quad (c < 0) \quad (4.14)$$

ここで、 $\chi' = |c|\Delta t/\Delta x$ である。風上差分法では c の正負によって適用する差分式が異なる。

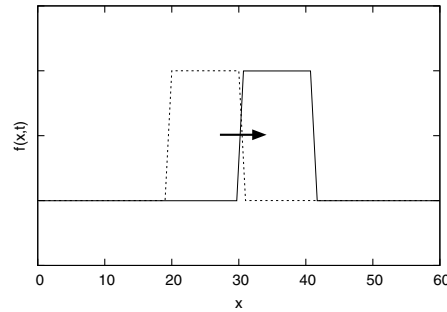


Fig.4.4: 矩形波の伝播 (厳密解)

4.2.1 波動方程式のアルゴリズム

1. 定数 $\Delta t, \Delta x, c$ を設定する。
2. 配列変数 f_j を 0 ～ 60 まで宣言する。
3. 初期の波の分布を設定する。また、 $f_0 = f_{60} = 0$ とする。
4. n のループを組む。 n は 0 ～ 100 まで 1 ずつ変化する。
 1. 各値を出力する。
 2. j のループを組む。 j は 1 ～ 59 まで 1 ずつ変化する。
 1. 各 j において次の時間の f_j^{n+1} を計算する。
 3. f_j^n と f_j^{n+1} の変数交換
5. 終了する。

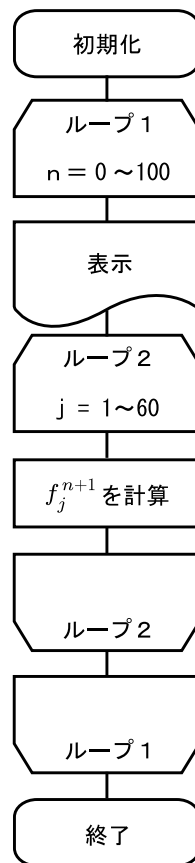


Fig.4.5: 波動方程式のフローチャート

4.2.2 波動方程式の問題

初期条件として j が 20 と 30 の間で $f_j^0 = 1$ とし、それ以外では $f_j^0 = 0$ とし、各差分法 (中央差分法・Lax-Wendroff 法・風上差分法) を用いて、矩形波の伝播をシミュレーションせよ。空間のメッシュ数 j は $j = 0 \sim 60$ とし、 $f_0 = f_{60} = 0$ とする。また、 $c = 1.0, \Delta x = 1.0, \Delta t = 0.1$ とし、時間ステップは 100 まで計算し、グラフ化せよ。

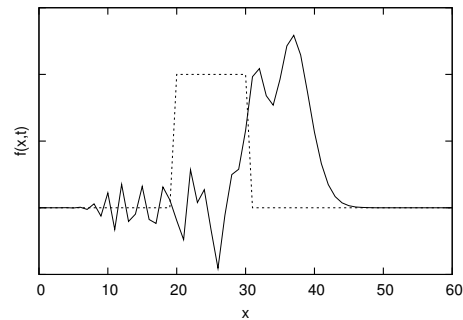


Fig.4.6: 中央差分法

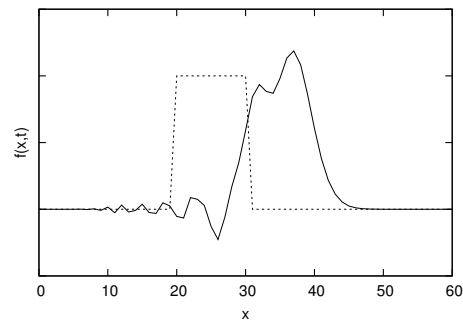


Fig.4.7: Lax-Wendroff 法

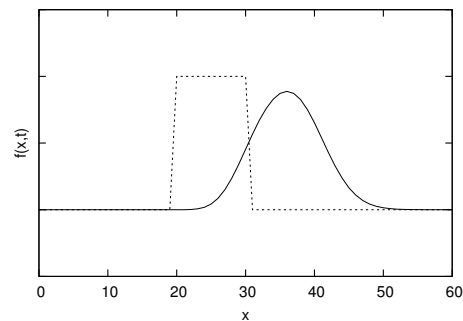


Fig.4.8: 風上差分法

4.3 固有値問題

次に量子力学におけるシュレディンガー方程式の固有値問題を数値的に解くことを考える。シュレディンガー方程式は

$$-\frac{\hbar^2}{2m}\nabla^2 f + V(x)f = Ef \quad (4.15)$$

である。ここで、 $\hbar$ はプランク定数 h を 2π で割ったもの、 m は質量、 $V(x)$ はポテンシャル、 E はエネルギー固有値、 f は波動関数である。1次元の場合を考えると、(4.15) 式は以下になる。

$$\frac{d^2 f}{dx^2} + k(x)^2 f = 0 \quad (4.16)$$

ここで、 $k(x)$ は

$$k(x)^2 = \frac{2m}{\hbar} \{E - V(x)\} = E^* - V(x)^* \quad (4.17)$$

である。この方程式に与えられた境界条件及びポテンシャル $V(x)$ のもとで解くのであるが、エネルギー固有値 E は決められていない。このような場合、与えられた境界条件を満たす解は E がある特定の値 (これを固有値と呼ぶ) の時のみ存在することが知られている。

このような境界値問題を数値的に解く方法の1つとして、shooting 法がある。これは一方の境界から出発した解ともう一方の境界から出発した解とを系の中のある点をつなぐ方法である。この方法を用いて最も簡単なポテンシャル $V(x)$ の場合について解いていく。ポテンシャル $V(x)$ を次のように定義する。

$$V^* = \infty \quad (x < 0, x > L) \quad (4.18)$$

$$V^* = 0 \quad (0 < x < L) \quad (4.19)$$

また、境界条件は $V(0) = V(L) = 0$ とする。このポテンシャル $V(x)$ における解析解は、良く知られているように

$$E_n^* = \left(\frac{\pi n}{L}\right)^2 \quad (4.20)$$

$$f_n = \sqrt{\frac{2}{L}} \sin \frac{n\pi}{L} x \quad (4.21)$$

となる。ここで、 n は $n = 1, 2, 3, \dots$ と変化する。これがポテンシャルの井戸に閉じ込められた粒子の波動関数である。また E の値をエネルギー順位と呼び、飛び飛びの値だけがとることを許される。

4.3.1 固有値問題のアルゴリズムと問題

以下の手順に従って、shooting 法のプログラムを作成し、(4.19) 式のポテンシャルのもとで固有値 E_1 を求めよ。下記にある f_j^L, f_j^R は境界の左側から求め始めた波動関数 f_j^L と右側から求め始めた波動関数 f_j^R である。一般的に境界条件が同じでも f_j^L と f_j^R は同じ値にならないが、真の解ならば $f_j^L = f_j^R$ が成り立たなければならない。成り立たない場合は E^* を少し変え、最適な E^* を求めていく。また、この場合 k_j は $k_j^2 = E^*$ である。

1. 各定数 $\Delta x, E^*, \Delta E^*, L, e$ を設定する。ここで $\Delta x = 1.0$ 、 $E^* = 0$ 、 $\Delta E^* = 10^{-7}$ 、井戸の幅 $L = 50.0$ 、要求精度 $e = 10^{-3}$ とする。
2. 実数型配列変数 f_j^L と f_j^R を $j = 0 \sim 50$ までの範囲で宣言する。また、判別用に整数型で g_j を $j = 0 \sim 50$ までの範囲で同じく宣言する。
3. 適当な f_1^L, f_{49}^R を選ぶ。ここでは、 $f_1^L = f_{49}^R = 1.0$ とする。求めたい波動関数が偶関数の場合は $f_1^L = f_{49}^R = 1.0$ 、奇関数の場合は $f_1^L = 1.0$ 、 $f_{49}^R = -1.0$ とすればよい。
4. 境界条件 $f_0^L = f_{50}^L = f_0^R = f_{50}^R = 0$ を適用する。 $g_j = 0$ ($0 \leq j \leq 50$) とする。
5. j のループを組む。 j は $1 \sim 49$ まで 1 ずつ変化する。
 1. 左側から波動関数 f^L を計算する。

$$f_{j+1}^L = (2 - k_j^2 \Delta x^2) f_j^L - f_{j-1}^L$$

6. j のループを組む。 j は $49 \sim 1$ まで 1 ずつ変化する。

1. 右側から波動関数 f^R を計算する。

$$f_{j-1}^R = (2 - k_j^2 \Delta x^2) f_j^R - f_{j+1}^R$$

7. j のループを組む。 j は $0 \sim 50$ まで 1 ずつ変化する。

1. $|f_j^L - f_j^R| \leq e$ なら $g_j = 1$ とする。

8. g_j の $j = 0 \sim 50$ までの和を求め、和が 51 ならそのときの E^* を表示し計算を終了する。そうでないなら、 $E^* = E^* + \Delta E^*$ とし、3 に戻り計算を繰り返す。

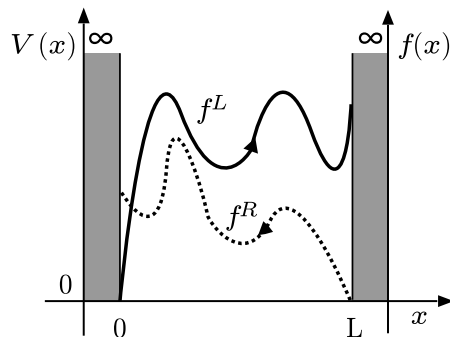


Fig.4.9: shooting 法の概念図

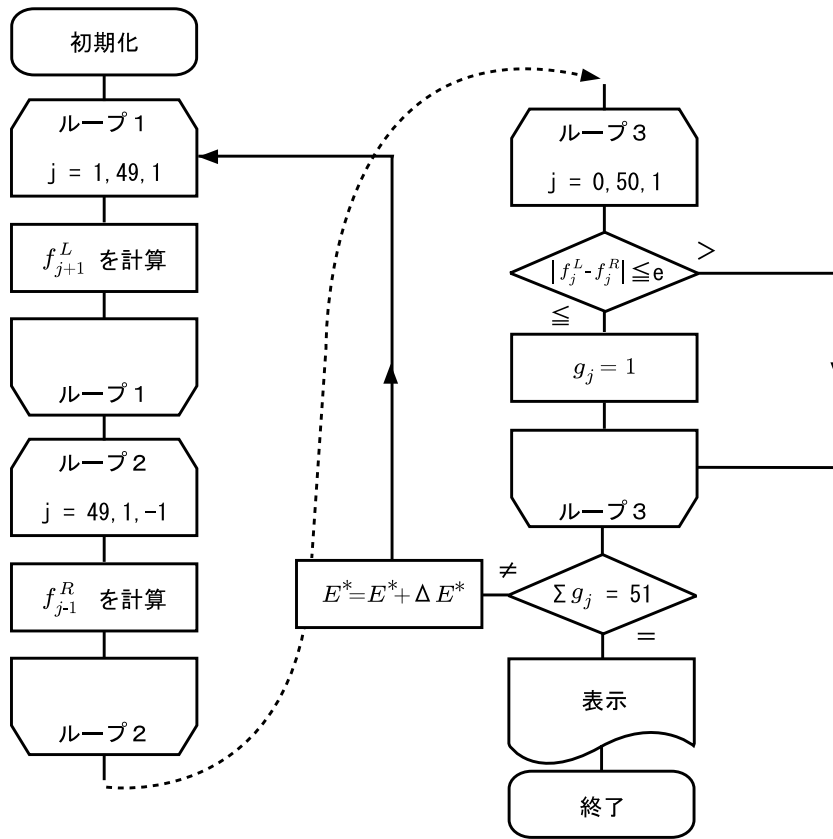


Fig.4.10: shooting 法のフローチャート

shooting 法によるエネルギー順位 E_1 の数値解は $E_1 = 0.003946$ となる。次に解析的にエネルギー順位 E_1 を求め比較してみよう。エネルギー順位 E_n^* の解析解は (4.20) 式より

$$E_n^* = \left(\frac{\pi n}{L} \right)^2$$

である。今は $n = 1$ の時の値を知りたいので、 $n = 1, L = 50.0$ を上式に代入すると、 E_1 の解析解は

$$E_1^* = \left(\frac{\pi}{50} \right)^2 = 0.003948$$

以上より、数値解と解析解とは良い一致を得た。同様にして、 E^* の値を増やしていけば、さらに高いエネルギー順位を求めていくことができる。

第 5 章

連立方程式の解法

次に連立方程式を数値的に解くことを考える。微分方程式などに比べ概念的には容易だが、数値的に扱う際のスキームはこちらの方がはるかに難解であることから、こちらの章を後にした。2 階の微分方程式は差分化すると、連立方程式の形になるため、ここで学ぶスキームは応用範囲が極めて広い。また、多くの参考文献もあることから平行して読み進めることをお薦めする。

5.1 ガウスの消去法

ガウスの消去法は、大きく 2 つのステップに分けられる。1 つ目のステップが前進消去ステップ、2 つ目が後退代入ステップと呼ばれている。次のような $x_0, x_1, \dots, x_{n-1}$ を未知数とする n 元連立 1 次方程式を数値的に解くことを考えよう。

$$\begin{cases} a_{00}x_0 + a_{01}x_1 + \cdots + a_{0,n-1}x_{n-1} = b_0 \\ a_{10}x_0 + a_{11}x_1 + \cdots + a_{1,n-1}x_{n-1} = b_1 \\ \vdots \\ a_{n-1,0}x_0 + a_{n-1,1}x_1 + \cdots + a_{n-1,n-1}x_{n-1} = b_{n-1} \end{cases} \quad (5.1)$$

解析的には各方程式を組み合わせて未知数を消去していく方法が知られている。ガウスの消去法は、これを系統的に行っていくスキームである。

5.1.1 前進消去ステップ

1. n 元連立 1 次方程式の n 個の式を上から順に、第 0 式, 第 1 式, $\dots$, 第 $n-1$ 式と呼ぶことにする。まず、第 0 式を a_{00} で割れば

$$x_0 + a_{01}^{(0)}x_1 + a_{02}^{(0)}x_2 + \cdots + a_{0,n-1}^{(0)}x_{n-1} = b_0^{(0)} \quad (5.2)$$

を得る。次に (5.2) 式を a_{10} 倍して、第 1 式から減算する。次に次に (5.2) 式を a_{20} 倍して、第 2 式から減算する、という操作を第 $n-1$ 式まで繰り返す。

$$\text{第 1 式} - a_{10}(\text{第 0 式}) = b_1^{(0)}$$

$$\text{第 2 式} - a_{20}(\text{第 0 式}) = b_2^{(0)}$$

$$\vdots$$

$$\text{第 } n-1 \text{ 式} - a_{n-1,0}(\text{第 0 式}) = b_{n-1}^{(0)}$$

すると、以下のような方程式を得ることができる。

$$\left\{ \begin{array}{lcl} x_0 + a_{01}^{(0)}x_1 + a_{02}^{(0)}x_2 + \cdots + a_{0,n-1}^{(0)}x_{n-1} & = & b_0^{(0)} \\ a_{11}^{(0)}x_1 + a_{12}^{(0)}x_2 + \cdots + a_{1,n-1}^{(0)}x_{n-1} & = & b_1^{(0)} \\ & \vdots & \\ a_{n-1,0}^{(0)}x_1 + a_{n-1,1}^{(0)}x_2 + \cdots + a_{n-1,n-1}^{(0)}x_{n-1} & = & b_{n-1}^{(0)} \end{array} \right. \quad (5.3)$$

2. 次に第 2 式以下の x_1 を消去する。まず、第 1 式を $a_{11}^{(0)}$ 式で割ると次式を得る。

$$x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1,n-1}^{(1)}x_{n-1} = b_1^{(1)} \quad (5.4)$$

(5.4) 式を $a_{21}^{(0)}$ 倍して、第 2 式から減算し $\cdots$ 、 $a_{n-1,1}^{(0)}$ 倍して第 $n-1$ 式から減算という操作を繰り返せば

$$\text{第 2 式} - a_{21}^{(0)}(\text{第 1 式}) = b_2^{(1)}$$

$$\text{第 3 式} - a_{31}^{(0)}(\text{第 1 式}) = b_3^{(0)}$$

$$\vdots$$

$$\text{第 } n-1 \text{ 式} - a_{n-1,1}^{(0)}(\text{第 1 式}) = b_{n-1}^{(0)}$$

以下のような方程式を得る。

$$\left\{ \begin{array}{lcl} x_0 + a_{01}^{(0)}x_1 + a_{02}^{(0)}x_2 + \cdots + a_{0,n-1}^{(0)}x_{n-1} & = & b_0^{(0)} \\ & x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1,n-1}^{(1)}x_{n-1} & = b_1^{(1)} \\ & a_{22}^{(1)}x_2 + \cdots + a_{2,n-1}^{(1)}x_{n-1} & = b_2^{(1)} \\ & \vdots & \\ & a_{n-1,1}^{(1)}x_2 + \cdots + a_{n-1,n-1}^{(1)}x_{n-1} & = b_{n-1}^{(1)} \end{array} \right. \quad (5.5)$$

3. 同様の操作を繰り返すことで、以下のような方程式を得る。

$$\left\{ \begin{array}{lcl} x_0 + a_{01}^{(0)}x_1 + a_{02}^{(0)}x_2 + \cdots + a_{0,n-1}^{(0)}x_{n-1} & = & b_0^{(0)} \\ & x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1,n-1}^{(1)}x_{n-1} & = b_1^{(1)} \\ & \vdots & \\ & & x_{n-1} = b_{n-1}^{(n-1)} \end{array} \right. \quad (5.6)$$

このように順番に未知数を消していくため、このステップは前進消去ステップと呼ばれている。

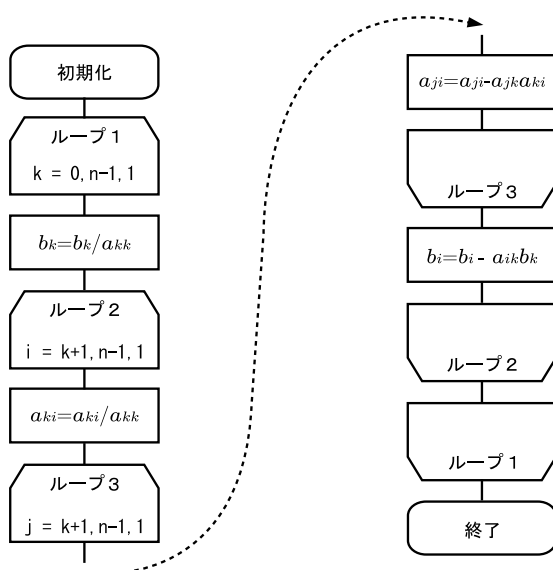


Fig.5.1: 前進消去ステップのフローチャート

5.1.2 後退代入ステップ

前進消去ステップの最後で、 x_{n-1} の値が b_{n-1} として得られている。この値を 1 つ上の方程式に代入すれば、 x_{n-2} の値が求まる。これを $x_{n-3}, \dots, x_1, x_0$ と繰り返すことで、連立方程式の解を求めることができる。

具体的には次の処理を行えば、 x_k の値が b_k に代入される。ここで、 k は $k = n-2, n-3, \dots, 2, 1, 0$ と変化する。

$$b_k = b_k - \sum_{j=k+1}^{n-1} a_{kj}x_j \quad (5.7)$$

このように逆順で解を求めていく、このステップを後退代入ステップという。

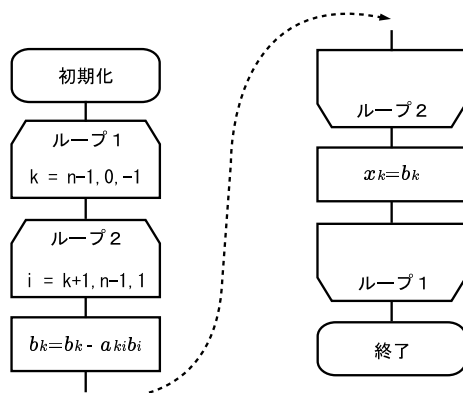


Fig.5.2: 後退代入ステップのフローチャート

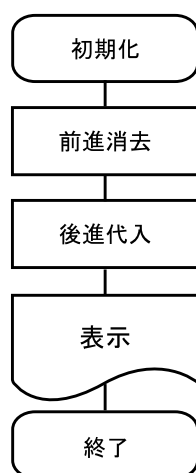


Fig.5.3: ガウスの消去法のフローチャート

5.1.3 ガウスの消去法の問題

ガウスの消去法を用いて、次の1次元連立方程式を解け。

$$\begin{cases} 4x_0 + x_1 - 3x_2 + 5x_3 = 18 \\ 2x_0 + 3x_1 + x_2 + 2x_3 = 21 \\ 5x_0 - 4x_1 - 2x_2 + x_3 = 5 \\ -2x_0 + 2x_1 + 8x_2 + 2x_3 = 16 \end{cases} \quad (5.8)$$

この連立方程式の解は以下ようになる。

$$x_0 = 4$$

$$x_1 = 3$$

$$x_2 = 2$$

$$x_3 = 1$$

5.2 ガウス・ザイデル法

ガウスの消去法では、系統的に未知数を求めていったが、ここでは適当な初期値を仮定し、必要な精度になるまで繰り返し解を求めることで未知数を求める方法を取り扱う。このように必要な精度になるまで繰り返し解を求めていく方法を反復法という。

ここでは、実際に例を考えながら、アルゴリズムの流れを追おう。以下のような連立方程式を考える。

$$\begin{cases} 3x + y + z &= 8 \\ 2x + 6y + 3z &= 23 \\ x + 2y + 4z &= 17 \end{cases} \quad (5.9)$$

上式を (x, y, z) についてそれぞれ解いた形に直す。

$$\begin{cases} x &= (8 - y - z)/3 \\ y &= (23 - 2x - 3z)/6 \\ z &= (17 - x - 2y)/4 \end{cases} \quad (5.10)$$

適当な初期値を決める。ここでは $(x, y, z) = (1, 1, 1)$ とすると

$$\begin{cases} x &= (8 - 1 - 1)/3 = 2 \\ y &= (23 - 2 - 3)/6 = 2.666\dots \\ z &= (17 - 1 - 2)/4 = 2.41667\dots \end{cases} \quad (5.11)$$

このように初期値から値を次々に代入し、新しく求まった値を使い、次の値を求めるということを繰り返し行う。すると以下のように (x, y, z) が真値に近づいていくのが分かる。

回数	x	y	z
1	2.000000	2.666667	2.416667
2	0.972222	2.300926	2.856481
3	0.947531	2.089249	2.968493
4	0.980753	2.022169	2.993727
5	0.994701	2.004903	2.998873
6	0.998741	2.000983	2.999823
7	0.999731	2.000178	2.999978
8	0.999948	2.000028	2.999999
9	0.999991	2.000004	3.000000
10	0.999999	2.000000	3.000000
11	1.000000	2.000000	3.000000

Tab.5.1: ガウス・ザイデル法の流れ

これは、以下の式を計算していると要約することができる。

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=0}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n-1} a_{ij} x_j^{(k)} \right) \quad (5.12)$$

ここで肩の (k) は計算回数を表している。ガウス・ザイデル法では

$$\left| x_i^{(k+1)} - x_i^{(k)} \right| < \epsilon \quad (i = 0, 1, \dots, n-1) \quad (5.13)$$

となったら反復計算を止め、 $x_i^{(k+1)}$ を解とする。 ϵ は 要求誤差と呼ばれる十分小さな値である。

5.2.1 ガウス・ザイデル法のアルゴリズム

1. 初期化
2. 要求精度 ϵ を判別するための変数 err に 0 代入
3. i のループを組む。 i は $0 \sim N-1$ まで 1 ずつ変化する。
 1. 変数 sum に b_i を代入
 2. j のループを組む。 j は $0 \sim N-1$ まで 1 ずつ変化する。
 1. sum に $sum - a_{ij}x_j$ を代入する。
 3. sum に sum/a_{ii} を代入する。
 4. x_i に $x_i + sum$ を代入する。
 5. err に $err + |sum|$ を代入する。
4. 結果を表示する。
5. err が要求精度 ϵ より小さければ終了する。
6. そうでなければ、2 に戻り計算を繰り返す。

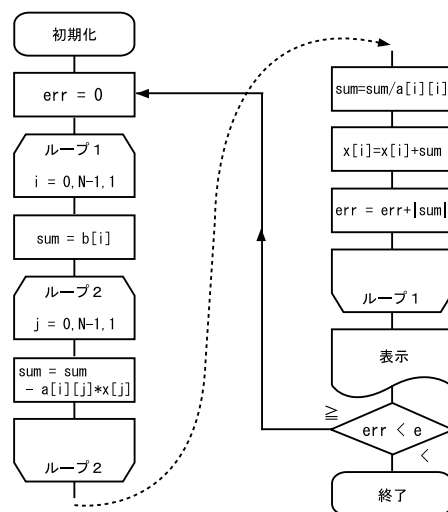


Fig.5.4: ガウス・ザイデル法のフローチャート

5.2.2 ガウス・ザイデル法の問題

ガウス・ザイデル法を用いて、次の1次元連立方程式を解け。

$$\begin{cases} 4x_0 + x_1 - 3x_2 + 5x_3 &= 18 \\ 2x_0 + 3x_1 + x_2 + 2x_3 &= 21 \\ 5x_0 - 4x_1 - 2x_2 + x_3 &= 5 \\ -2x_0 + 2x_1 + 8x_2 + 2x_3 &= 16 \end{cases} \quad (5.14)$$

要求精度 ϵ は 10^{-6} とする。また、初期値は $(x_0, x_1, x_2, x_3) = (1, 1, 1, 1)$ とした。

回数	x_0	x_1	x_2	x_3
1	2.000000	4.666667	1.083333	1.716667
5	3.852630	3.147325	1.956955	1.062604
10	4.009345	2.990894	2.002894	0.996082
15	3.999430	3.000556	1.999823	1.000239
20	4.000035	2.999966	2.000011	0.999985
25	3.999998	3.000002	1.999999	1.000001
29	4.000000	3.000000	2.000000	1.000000

Tab.5.2: ガウス・ザイデル法の問題の結果

第 6 章

物理シミュレーション 2

これまでの議論を踏まえて、ポアソン方程式を数値的に解くことを考える。ポアソン方程式は楕円型微分方程式の典型であり、様々な物理量の平衡状態を記述する方程式である。その応用範囲は広く、電磁場や熱伝導、温度分布など主に工学系の分野で用いられています。したがって、ポアソン方程式を数値的に解く技術を身に付けることは、実用的な問題を解く際に強力な武器となると言える。

6.1 ポアソン方程式

ポアソン方程式は下式で表される。

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = \rho \quad (6.1)$$

ここでは、簡単のために 2 次元の場合について考えていきます。下式を数値的に解くことを考えていく。

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \rho \quad (6.2)$$

ラプラス方程式は 2 階の微分方程式の形をしているので、前進差分と後退差分を用いて方程式を差分化する。

$$\begin{aligned} \frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) &\simeq \frac{\partial}{\partial x} \left\{ \frac{\phi(x + \Delta x, y) - \phi(x, y)}{\Delta x} \right\} \\ &\simeq \frac{1}{\Delta x^2} \left\{ \phi(x + \Delta x, y) - \phi(x, y) - \left(\phi(x, y) - \phi(x - \Delta x, y) \right) \right\} \\ &= \frac{1}{\Delta x^2} \left\{ \phi(x + \Delta x, y) - 2\phi(x, y) + \phi(x - \Delta x, y) \right\} \end{aligned} \quad (6.3)$$

同様に、 y 方向についても

$$\frac{\partial}{\partial y} \left(\frac{\partial \phi}{\partial y} \right) \simeq \frac{1}{\Delta y^2} \left\{ \phi(x, y + \Delta y) - 2\phi(x, y) + \phi(x, y - \Delta y) \right\} \quad (6.4)$$

ここで簡単のために刻み幅を $\Delta = \Delta x = \Delta y$ とすると (6.2) 式は (6.3), (6.4) 式より以下のように差分化することができる。

$$\frac{1}{\Delta^2} \left\{ \phi(x + \Delta, y) + \phi(x, y + \Delta) - 4\phi(x, y) + \phi(x - \Delta, y) + \phi(x, y - \Delta) \right\} = \rho \quad (6.5)$$

次に $\phi(x, y)$ を形式的に $\phi_{i,j}$ と表記しよう。すると (6.5) 式は以下のように書き直すことができる。

$$\frac{1}{\Delta^2} \left\{ \phi_{i+1,j} + \phi_{i,j+1} - 4\phi_{i,j} + \phi_{i-1,j} + \phi_{i,j-1} \right\} = \rho_{i,j} \quad (6.6)$$

(6.6) 式を $\phi_{i,j}$ について解けば、実用的な計算式が求まる。

$$\phi_{i,j} = \frac{1}{4} \left\{ \phi_{i+1,j} + \phi_{i,j+1} + \phi_{i-1,j} + \phi_{i,j-1} - \rho_{i,j} \right\} \quad (6.7)$$

ここで (i, j) は離散化された空間の位置座標を表しており、空間は以下のようなモデル図によって表される。点は格子点と呼ばれ、この格子点上に各物理量が記憶されることとなる。imx は x 方向の最大格子点数、jmx が y 方向の最大格子点数である。空間内には $(\text{imx} + 1) \times (\text{jmx} + 1)$ 個の格子点が存在することになる。

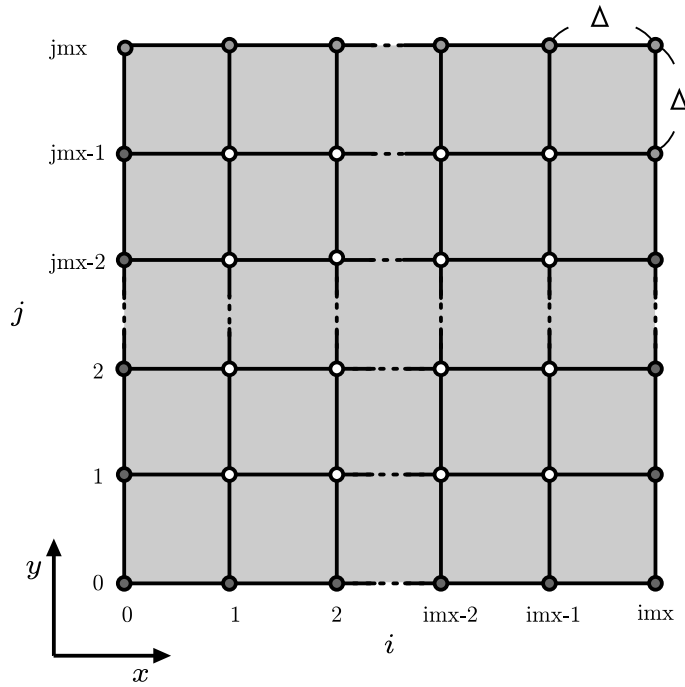


Fig.6.1: 空間格子モデル

6.1.1 ラプラス方程式解法のアルゴリズム

(6.6) 式は (i, j) についてそれぞれ展開すると膨大な連立方程式になっているため、従来のような代数的なスキームでは解くことができない。そこでここでは、前章で学んだガウス・ザイデル法を採用することにする。

また、ここでは境界条件として $\phi_{i,0} = \phi_{i,j_{\text{mx}}} = 0$ 、 $\phi_{0,j} = \phi_{i_{\text{mx}},j} = 0$ とした。つまり、Fig.6.1 の外周部の格子点 (灰色) の物理量を 0 とする。ループ文の範囲がそれぞれ $1 \sim i_{\text{mx}} - 1, 1 \sim j_{\text{mx}} - 1$ なのは、境界条件を除いた格子点 (白点) での反復計算を行うためである。

1. 初期化
2. 境界条件を適用する。境界上の格子点の物理量を 0 にする。
3. 要求精度 ϵ を判別するための変数 err に 0 代入
4. i のループを組む。 i は $1 \sim i_{\text{mx}} - 1$ まで 1 ずつ変化する。
 1. j のループを組む。 j は $1 \sim j_{\text{mx}} - 1$ まで 1 ずつ変化する。
 1. sum に $\phi_{i,j}$ を代入する。
 2. (6.7) 式
 2. err に $\text{err} + |\text{sum}|$ を代入する。
5. err が要求精度 ϵ より小さければ結果を出力して終了する。
6. そうでなければ、2 に戻り計算を繰り返す。

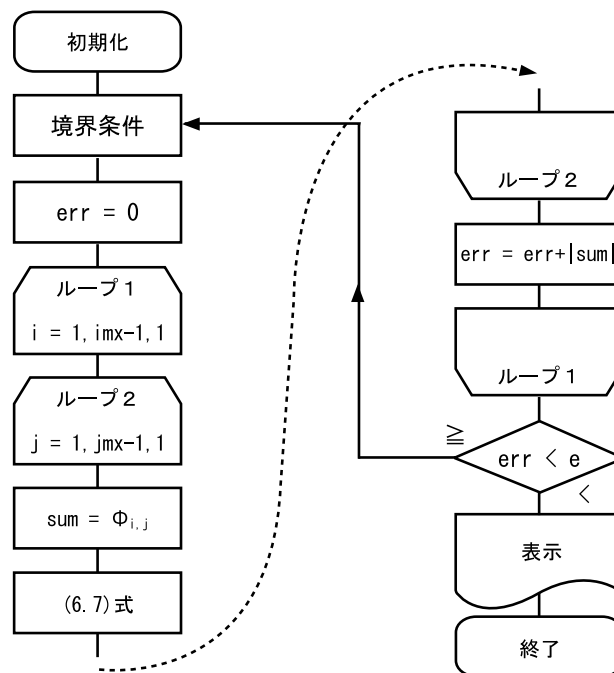


Fig.6.2: ポアソン方程式解法のアルゴリズム

6.1.2 ポアソン方程式の問題

次の条件の元でポアソン方程式を数値的に解き、その時のポテンシャル ϕ をグラフ化せよ。格子点数 $\text{imx} = \text{jmx} = 50$ 、刻み幅 $\Delta = 0.1$ 、要求精度 $\epsilon = 10^{-8}$ とする。また、電荷密度 ρ は $\rho_{i,j} = -1$ とする。境界条件は、 $\phi_{i,0} = \phi_{i,\text{jmx}} = 0$ 、 $\phi_{0,j} = \phi_{\text{imx},j} = 0$ を適用する。すなわち、境界上の格子点の物理量は 0 とする。

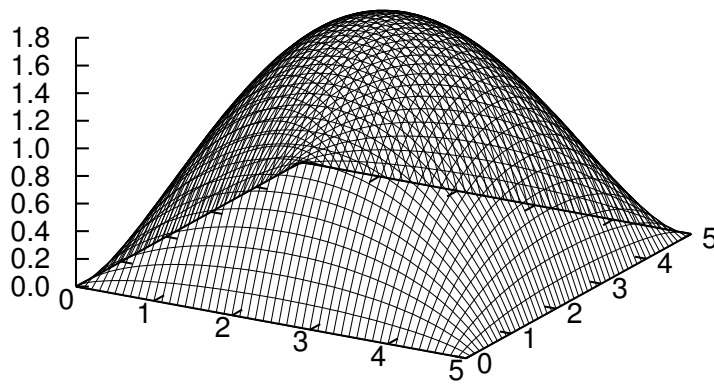


Fig.6.3: ポアソン方程式の問題の結果

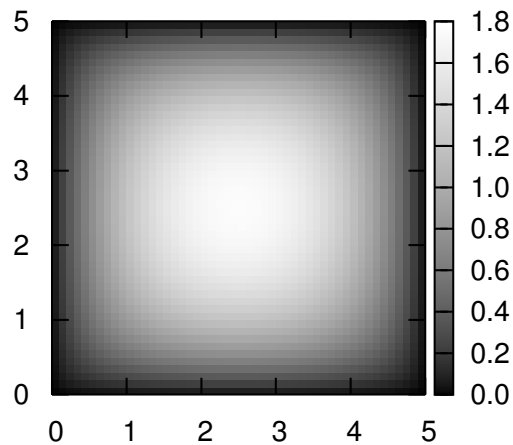


Fig.6.4: ポアソン方程式の問題の結果 (等値面)