前回資料の訂正

ヤコビ法のアルゴリズム

- 入力: $N, A = (a_{ij}), \boldsymbol{b} = (y_1, \ldots y_N), 0 < \varepsilon << 1$

1. 初期値 $\boldsymbol{x}^{(0)} = (x_1^{(0)}, \ldots, x_N^{(0)})^{\mathrm{T}}$ を設定する
2. $\boldsymbol{r} \leftarrow \boldsymbol{b}$;
3. while (1)
   a. sum $\leftarrow$ 0; error $\leftarrow$ 0; $\boldsymbol{b} \leftarrow \boldsymbol{r}$;
   b. for $i \in [1..N]$ do
      i. for $j \in [1.. i - 1]$ do $y_i \leftarrow y_i - a_{ij} x_j$;
      ii. for $j \in [i + 1 .. N]$ do $y_i \leftarrow y_i - a_{ij} x_j$;
      iii. $z_i \leftarrow y_i / a_{ii}$;

ヤコビ法のアルゴリズム

iv. sum $\leftarrow$ sum + $|z_i|$; error $\leftarrow$ error +$|z_i - x_i|$;

c. if (error < $\varepsilon \times$ sum) then break;

d. for $i \in [1..N]$ do $x_i \leftarrow z_i$;

2. return $\boldsymbol{z} = (z_1, \ldots, z_N)^{\mathsf{T}}$;

SOR法のアルゴリズム

- 入力: $N, A = (a_{ij}), \boldsymbol{b} = (y_1, \ldots y_N), 0 < \varepsilon << 1, \omega$

1. 初期値 $\boldsymbol{x}^{(0)} = (x_1^{(0)}, \ldots, x_N^{(0)})^{\mathrm{T}}$ を設定する
2. $\boldsymbol{r} \leftarrow \boldsymbol{b}$;
3. while (1)
    a. sum $\leftarrow$ 0; error $\leftarrow$ 0; $\boldsymbol{b} \leftarrow \boldsymbol{r}$;
    b. for $i \in [1..N]$ do
        i. for $j \in [1.. i - 1]$ do $y_i \leftarrow y_i - a_{ij} x_j$;
        ii. for $j \in [i + 1 .. N]$ do $y_i \leftarrow y_i - a_{ij} x_j$;
        iii. new_x $\leftarrow (\omega \times y_i) / a_{ii} + (1 - \omega)\, x_i$;

SOR法のアルゴリズム

iv. sum ← sum + |new_x|;

v. error ← error +|new_x-x_i|; x_i ← new_x;

c. if (error < ε × sum) then break;

3. return $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathsf{T}}$;

計算機数学II (2018)
2: 方程式の根

照井 章（筑波大学 数理物質系 数学域）
Akira Terui (Institute of Mathematics, University of Tsukuba)

第2章の内容

- 1変数方程式の根の計算
  - 2分法
  - ニュートン法

2-1 方程式の根と2分法

方程式の根

- $f(x)$: RからRへの連続関数
- 方程式 $f(x) = 0$ … (2.1) の根の近似値を、実数の四則演算を用いて、与えられた精度で求める
- 任意の x に対して $f(x)$ の正確な値を計算できるものとする

方程式の根

- 例 : $f(x) = x^2 + 2x - 1$ … (2.2)
- 2次方程式⇒解の公式で $x = -1 \pm \sqrt{2}$
  - さらに、例えば10進5桁の精度で近似値を求めたいならば、$\sqrt{2}$ の開平算で計算可能
- しかし、5次以上の代数方程式は一般に根号を用いて表すことはできない
- 代数方程式でない場合には解の公式も一般には存在しない ; 例 : $f(x) = \exp(x) - x^2 + \sin(x)$

方程式の根の計算

例 : $f(x) = x^2 - 11$ … (2.3)

方程式 $f(x) = 0$ の正の根 α を求める

1. $3^2 = 9 < 11 < 4^2 = 16$ より $3 < \alpha < 4$

   $(3+4)/2 = 3.5$

2. $11 < 3.5^2 = 12.25$ より $3 < \alpha < 3.5$

   $(3+3.5)/2 = 3.25$

方程式の根の計算

例 : $f(x) = x^2 - 11$ … (2.3)

方程式 $f(x) = 0$ の正の根 α を求める

3. $3.25^2 = 10.5625 < 11$ より $3.25 < \alpha < 3.5$

   $(3.25+3.5)/2 = 3.375$

4. $11 < 3.375^2 = 11.390625$ より $3.25 < \alpha < 3.375$

   $(3.25+3.375)/2 = 3.3125$

方程式の根の計算

例 : $f(x) = x^2 - 11$ … (2.3)

方程式 $f(x) = 0$ の正の根 α を求める

5. $3.3125^2 = 10.9726... < 11$ より $3.3125 < \alpha < 3.375$

   …

ここまでで $\alpha > 3.3$ が確認できる

根を含む区間を1/2倍ずつ絞っていく
⇒**2分法** (bisection method)

2分法

- 与えられた方程式の根のうち1個の近似値を求める
- $f(x)$ は連続関数であるので、中間値の定理が成り立つ
  - $[a, b]$ $(a < b)$: 実数上の閉区間
    $f(a) < 0$ かつ $f(b) > 0 \Rightarrow \exists\ c\ \in [a, b]$ s.t. $f(c) = 0$
- 計算の最初の段階で、上の条件をみたすa, bを与えるものとする

2分法のアルゴリズム（暫定版）

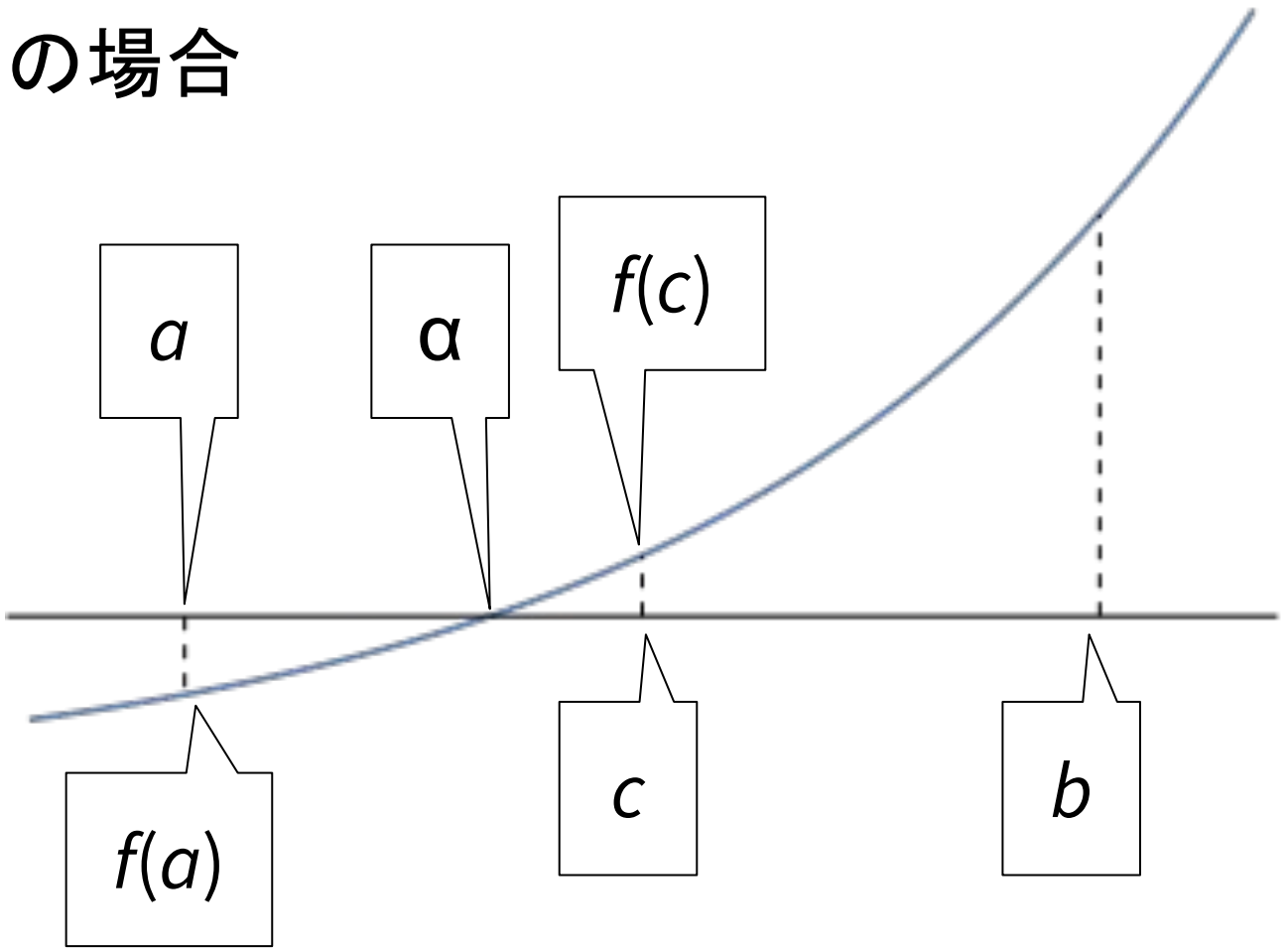
入力: $f(x)$: 連続関数、$a < b$: $f(a) < 0$ かつ $f(b) > 0$

出力: $c \in \mathbb{R}$ s.t. $f(c) \fallingdotseq 0$

1. while (1)
   a. $c \leftarrow (a + b)/2$;
   b. $fc \leftarrow f(c)$;
   c. if ($fc > 0$) then $b \leftarrow c$
      else if ($fc < 0$) then $a \leftarrow c$
      else if ($fc = 0$) then break;
2. return c;

2分法の動作

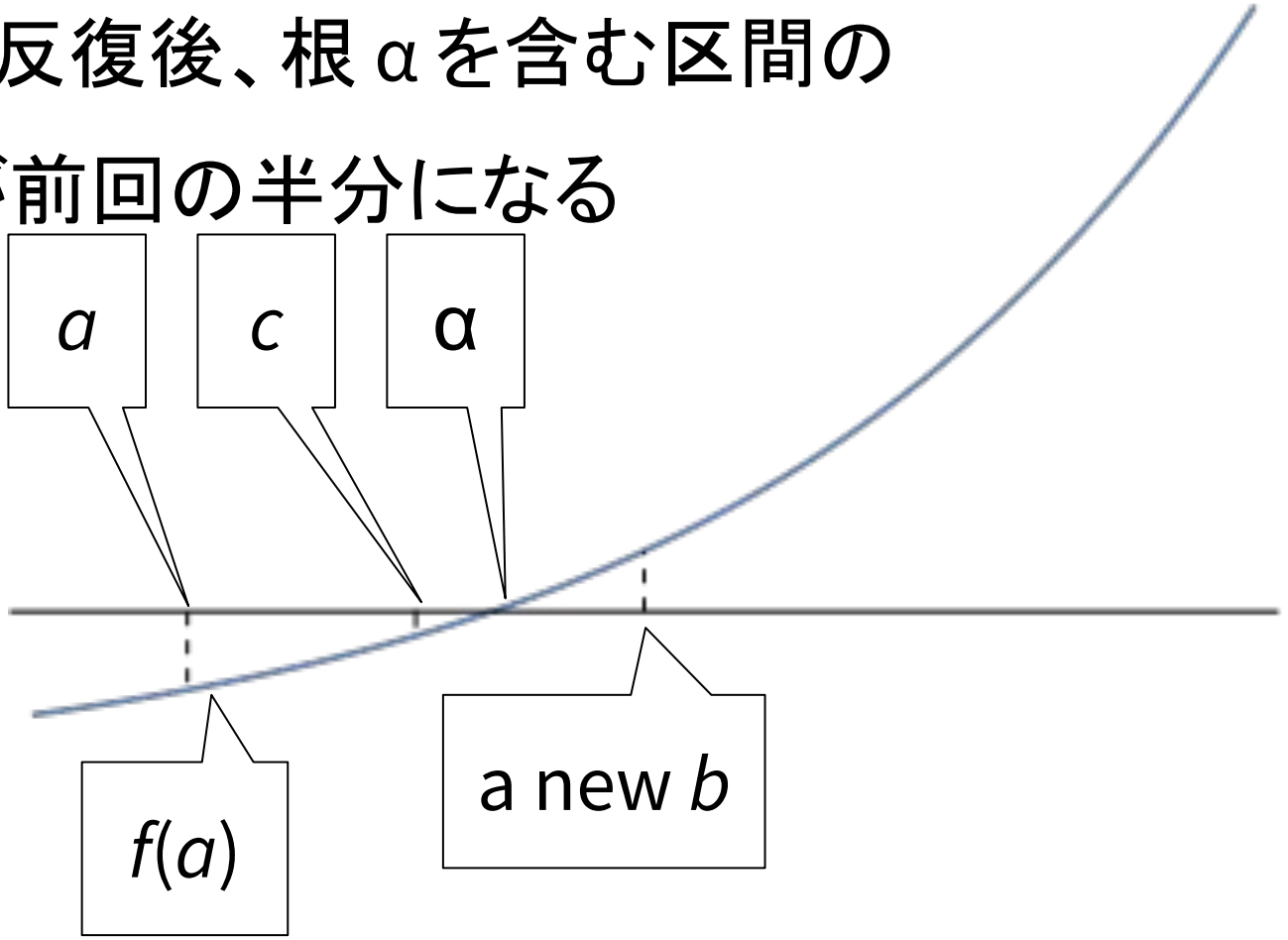
$f(c) > 0$ の場合



2分法の動作

1回の反復後、根αを含む区間の

長さが前回の半分になる



2分法の収束判定条件

- 収束判定条件 (convergence criterion):

  与えられた精度まで根が計算できたと見極めて、計算を途中で打ち切るための条件
- ここでは、計算された近似値と真の根との誤差で判定する

2分法の収束判定条件

- 収束判定のしきい値 : $0 < \varepsilon << 1$
- 真の根 : α
- あるステップで $[a, b]$ を計算したとき
- $c = (a+b)/2$ とすると、根は $[a, c]$ または $[c, b]$ の間に存在する
  $|\alpha - c| < |a - b| / 2$
- $|a - b| / 2 < \varepsilon$ を満たした段階で計算を打ち切り、c を答えにすると $|\alpha - c| < \varepsilon$ を満たす

2分法のアルゴリズム（収束判定条件つき）

入力: $f(x)$: 連続関数, $a < b$: $f(a) < 0$ かつ $f(b) > 0$, $\varepsilon > 0$

出力: $c \in \mathbb{R}$ s.t. $|c - (c$ に最も近い根$)| < \varepsilon$

1. while (1)
   a. $c \leftarrow (a + b)/2$;
   b. if $|a - b| / 2 < \varepsilon$ then break;
   c. $fc \leftarrow f(c)$;
   d. if ($fc > 0$) then $b \leftarrow c$; else if ($fc < 0$) then $a \leftarrow c$;
      else if ($fc = 0$) then break;
2. return c;

2分法の計算量

- ($f(c)$ の計算量) × (while ループの回数)
- while ループ1回繰り返しごとに、区間 $[a, b]$ の幅は半分になる
- ゆえに、whileループの繰り返しの回数は $|a - b| / 2^{N+1}$ を満たす最小の N
- すなわち $N = \lfloor \lg_2(|a - b| / \varepsilon) - 1 \rfloor$ … (2.6)’

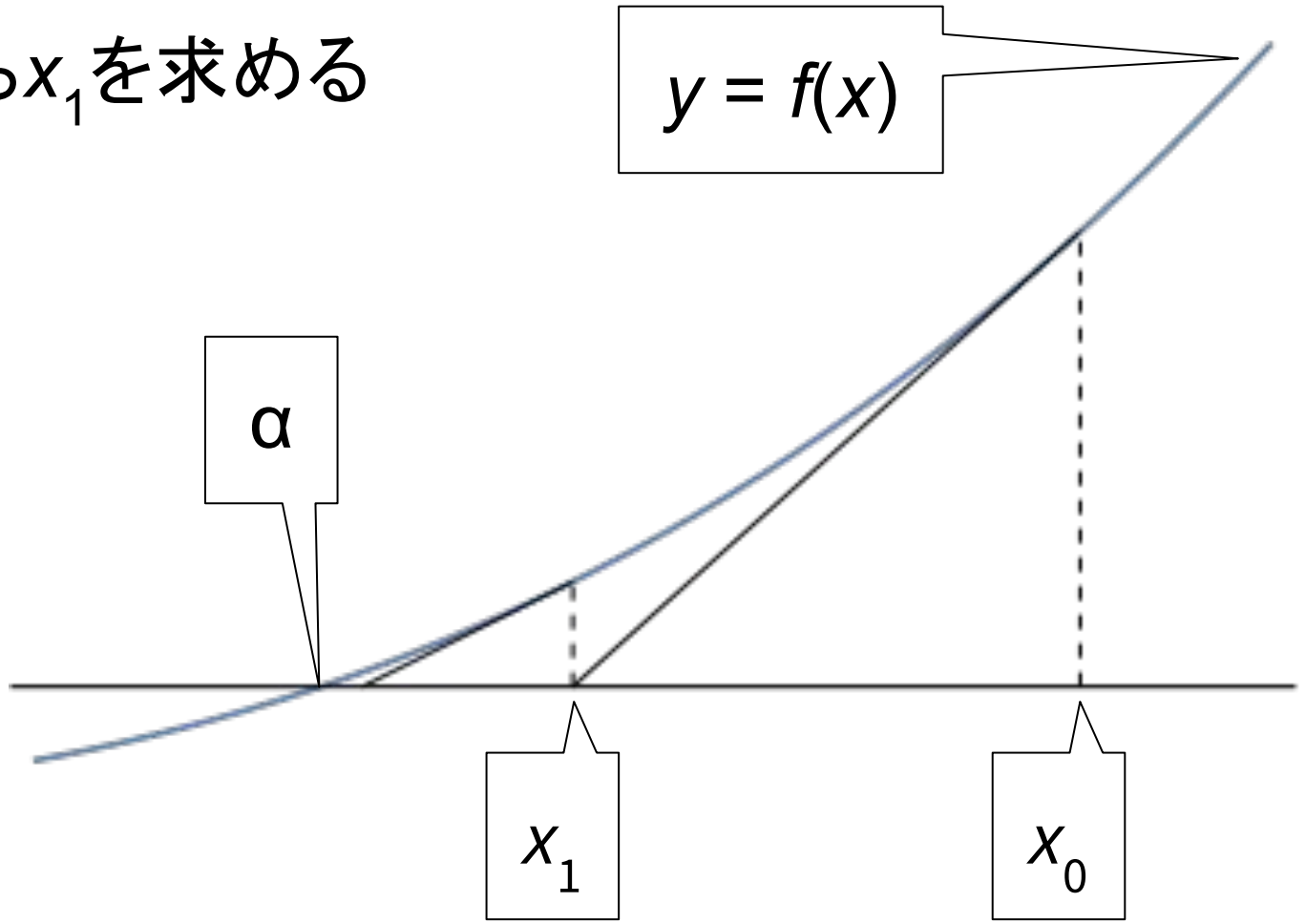
2分法の計算例

- $f(x) = e^{-x} - x^2$
- 収束の様子: 教科書 表 2-1
- 収束の速さはNewton法（後述）ほど速くはない
- 10進n桁分の精度を得るのに約3n回の反復
- 収束は安全で確実

2-2 Newton法 (Newton's method)

Newton法の漸化式の導出

x_0からx_1を求める



Newton法の漸化式の導出

- 求める根 α 付近の点 $x = x_0$ をとる
- $x = x_0$ と $x = \alpha$ の間で $f(x) \neq 0$ とする
- $x = x_0$ における $f(x)$ の傾き: $f'(x_0)$
- $x = x_0$ における $f(x)$ の接線:

$$y = f'(x_0)\,(x - x_0) + f(x_0) \qquad (2.7)$$

Newton法の漸化式の導出

- $x = x_0$ における $f(x)$ の接線:

  $y = f'(x_0)\,(x - x_0) + f(x_0)$ (2.7)

- 式 (2.7) で $y = 0$ とおいて x について解くと

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (2.8)$$

- 式 (2.8) で $x_{n+1} = x_n$ に収束すれば $f(x_n) = 0$ を満たす

Newton法の収束判定条件

$$\left| \frac{x_{n+1} - x_n}{x_{n+1}} \right| < \varepsilon \qquad (2.9)$$

x_{n+1} - x_n (差分) の x_{n+1} に対する割合

もし式 (2.9) が成り立っていれば、x_{n+1} は x_n と相対誤差 ε 程度まで一致している

Newton法のアルゴリズム

入力: $f(x)$: 関数, x: 初期値, ε: 収束判定条件

出力: α: 方程式 $f(x)=0$ の根の近似値

1. while (1)
   a. new_x ← x - f(x)/$f'(x)$;
   b. if |new_x - x| < ε |new_x| then break;
   c. x ← new_x;
2. return new_x;

Newton法の計算例

- $f(x) = e^{-x} - x^2$
- 収束の様子: 教科書 表 2-3
- 2分法で5桁の精度まで求めるのに17回の反復を要したのに対し、Newton法ではわずか3回

Newton法の計算量

- 1回の反復で行う関数計算の回数:
  - Newton法: $f(x)$, $f'(x)$ 2回
  - 2分法: $f(x)$ 1回
- 例題で、5桁の精度まで求めるのに実行される関数計算の回数:
  - Newton法: **6**回（反復回数3回）
  - 2分法: **17**回（反復回数17回）

Newton法の計算量

- Newton法は、**初期値を適切に選べば**、2分法に比べて収束が速く、計算量も小さく抑えられる

Newton法の収束性

- 根: α
- n 回反復後の誤差: $\varepsilon_n = |x_n - \alpha|$
- $n+1$ 回反復後の誤差: $\varepsilon_{n+1} = |x_{n+1} - \alpha|$
- ε_n と ε_{n+1} の関係を調べる

Newton法の収束性

式 (2.8) より

$$\varepsilon_{n+1} = x_{n+1} - \alpha = x_n - \frac{f(x_n)}{f'(x_n)} - \alpha = \varepsilon_n - \frac{f(x_n)}{f'(x_n)} \tag{2.10}$$

$x_n = \alpha + (x_n - \alpha) = \alpha + \varepsilon_n$ より、x_n で $f(x)$, $f'(x)$ の Taylor展開を行うと

Newton法の収束性

$$f(x_n) = f(\alpha + \varepsilon_n) = f(\alpha) + \varepsilon_n f'(\alpha) + \frac{\varepsilon_n^2}{2} f''(\alpha) + \frac{\varepsilon_n^3}{6} f'''(\xi_1)$$

$$f'(x_n) = f'(\alpha + \varepsilon_n) = f'(\alpha) + \varepsilon_n f''(\alpha) + \frac{\varepsilon_n^2}{2} f'''(\xi_2)$$

(2.11)

これを式 (2.10) に代入すると

Newton法の収束性

$$\varepsilon_{n+1} = \varepsilon_n - \frac{\varepsilon_n f'(\alpha) + \frac{\varepsilon_n^2}{2} f''(\alpha) + \frac{\varepsilon_n^3}{6} f'''(\xi_1)}{f'(\alpha) + \varepsilon_n f''(\alpha) + \frac{\varepsilon_n^2}{2} f'''(\xi_2)}$$

$$= \left(\frac{\varepsilon_n^2}{2}\right) \left(\frac{f''(\alpha) + \varepsilon_n f'''(\xi_2) - \frac{\varepsilon_n}{3} f'''(\xi_1)}{f'(\alpha) + \varepsilon_n f''(\alpha) + \frac{\varepsilon_n^2}{2} f'''(\xi_2)} \right) \tag{2.12}$$

ε_nが十分小さければ、分母、分子の ε_n 以降の項を無視して

$$\varepsilon_{n+1} \simeq \left(\frac{f''(\alpha)}{2f'(\alpha)}\right) \varepsilon_n^2 \tag{2.13}$$

(ただし $f'(\alpha) \neq 0$)

Newton法の収束性

- $\varepsilon_{n+1} = O(\varepsilon_n^{\,2})$
- x_n が α に十分近いとき「x_n は α に**2次収束**する」という
- 「$\varepsilon_n = 10^{-p} \Rightarrow \varepsilon_{n+1} = 10^{-2p}$」
- 「x_n の精度が10進 n 桁であれば、x_{n+1}の精度は10進$2n$桁になる」

Newton法の短所と対策

問題点

- 初期値が真の根から離れているような場合
- $|f'(x_n)| << 1$ の場合
- 修正項 $f(x)/f'(x)$ の絶対値が相対的に大きくなり、x_n が発散したり振動したりすることがある

Newton法の短所と対策

対策（例）

- 最初に2分法で真の根に十分近い近似値を求めておき
- そこからNewton法を使って速く収束させる

第2章のまとめ

- 1変数方程式の数値解法
  - 2分法
  - Newton法
    - 2次収束

計算機数学II (2018)
3: 曲線の推定

照井 章（筑波大学 数理物質系 数学域）
Akira Terui (Institute of Mathematics, University of Tsukuba)

第3章の内容

- 曲線の推定（補間法）
  - ラグランジュ(Lagrange)補間
  - スプライン(Spline)補間
  - 最小二乗法

3-1 曲線の推定

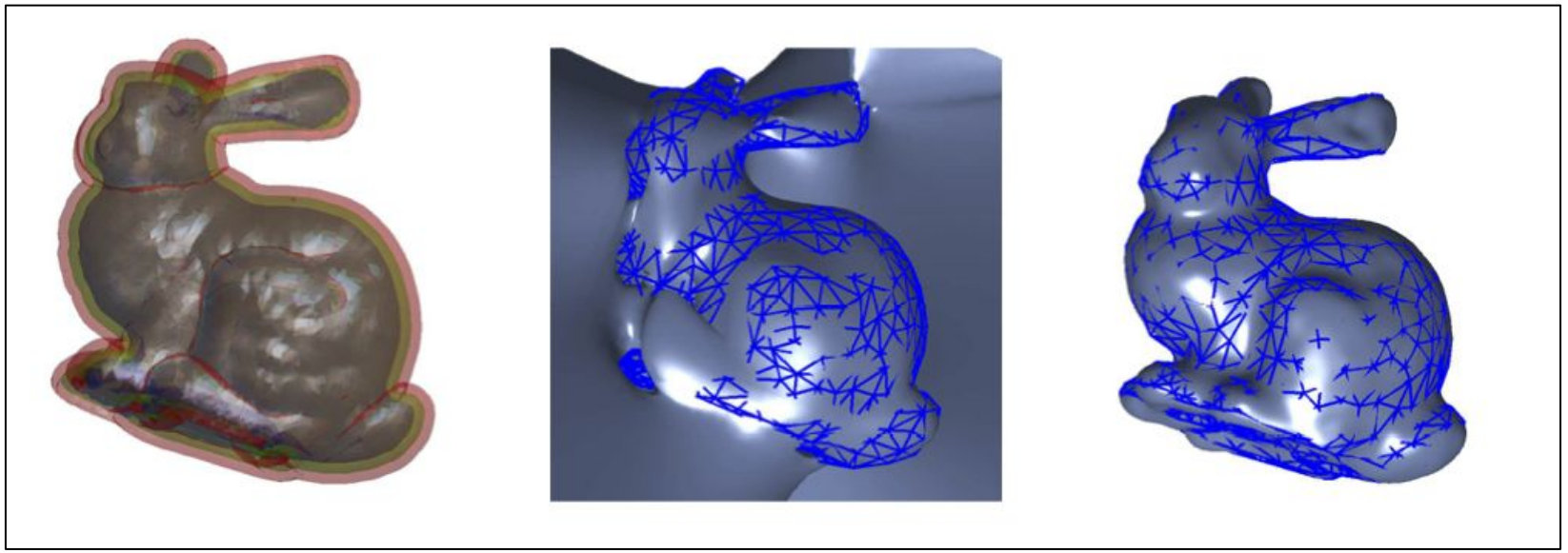
曲線の推定（その1）

xy 平面上にN+1個の点

$$(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)$$

が与えられたとき、これらの点すべてを通る曲線を推定する

曲線の推定（その1）



from M. Rouhani, A. Sappa, E. Boyer (2015),

“Implicit B-Spline Surface Reconstruction”,

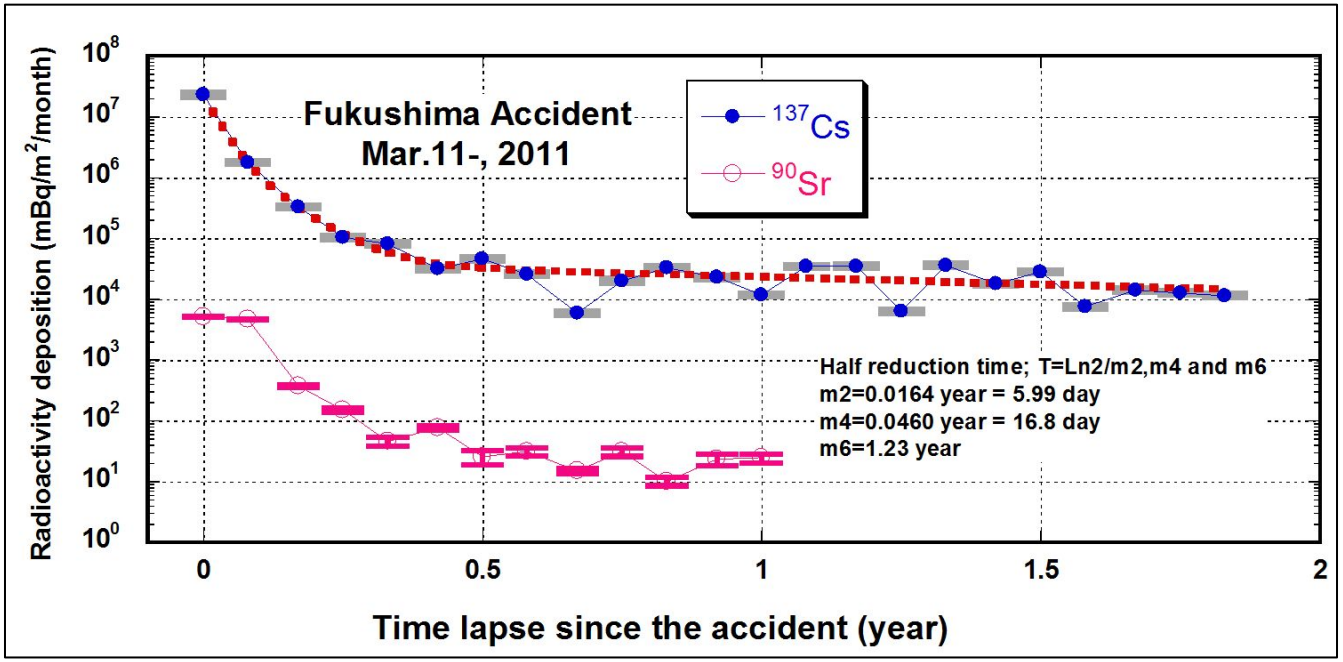
IEEE Trans. Image Process., 24, 22–32.

曲線の推定（その2）

実験や統計で得られたデータ点に最も近いモデルを推測する

☞ 曲線のあてはめ (curve-fitting)

曲線の推定（その2）



福島第一原発事故後のセシウム137の降下量
気象研究所「環境における人工放射能の研究2013」

曲線の推定

その1

- ラグランジュ(Lagrange)補間
- スプライン(Spline)補間

その2

- 最小二乗法

3-2 ラグランジュ補間

ラグランジュの補間多項式

Given: xy 平面上の点

$$(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N), \qquad x_0 < x_1 < \cdots < x_N$$

Find:

$$p_N(x) = a_0 + a_1 x + \cdots + a_N x^N \qquad (3.1)$$

Satisfying

$$p_N(x_j) = y_j \quad (j=0,1, \ldots, N) \qquad (3.2)$$

$N = 1$ の場合

Given: $(x_0, y_0), (x_1, y_1)$

Find: $p_1(x) = a_0 + a_1 x$（直線）

a_0, a_1 は、連立方程式

$$\begin{aligned} a_0 + a_1 x_0 &= y_0 \\ a_0 + a_1 x_1 &= y_1 \end{aligned} \tag{3.3}$$

を解くと

$$a_0 = (x_1 y_0 - x_0 y_1)/(x_1 - x_0),\ a_1 = (y_1 - y_0)/(x_1 - x_0)$$

$N = 1$ の場合

ゆえに

$$
\begin{aligned}
p_1(x) &= \frac{x_1 y_0 - x_0 y_1}{x_1 - x_0} + \frac{y_1 - y_0}{x_1 - x_0} x \\
&= \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_0 - x_1} y_1 \qquad (3.4)
\end{aligned}
$$

$N = 2$ の場合

Given: $(x_0, y_0), (x_1, y_1), (x_2, y_2)$

Find: $p_1(x) = a_0 + a_1 x + a_2 x^2$ ($a_2 \neq 0$であれば放物線)

By solving

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 &= y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 &= y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 &= y_2 \end{aligned} \tag{3.5}$$

$N = 2$ の場合

we find

$$p_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0 + \frac{(x-x_2)(x-x_0)}{(x_1-x_2)(x_1-x_0)}y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2 \tag{3.6}$$

一般のN+1の場合

$$l_j(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_N)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_N)}$$

(j = 1, … , N)

- 分母には $x - x_j$ の項がない (3.7)
- 分母には $x_j - x_j$ の項がない
- $l_j(x)$ は x の N 次多項式

$$l_j(x_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (3.8)$$

一般の$N+1$の場合

- そこで、$p_N(x)$ を

$$p_N(x) = y_0 l_0(x) + y_1 l_1(x) + \cdots + y_N l_N(x) \qquad (3.9)$$

とおくと、$p_N(x)$ はたかだか N 次の多項式で、

かつ $p_N(x_j) = y_j$ ($j = 1, \ldots, N$) … (3.8) を満たす

$p_N(x)$: ラグランジュの補間多項式
Lagrange's interpolating polynomial

ラグランジュの補間多項式による曲線の例

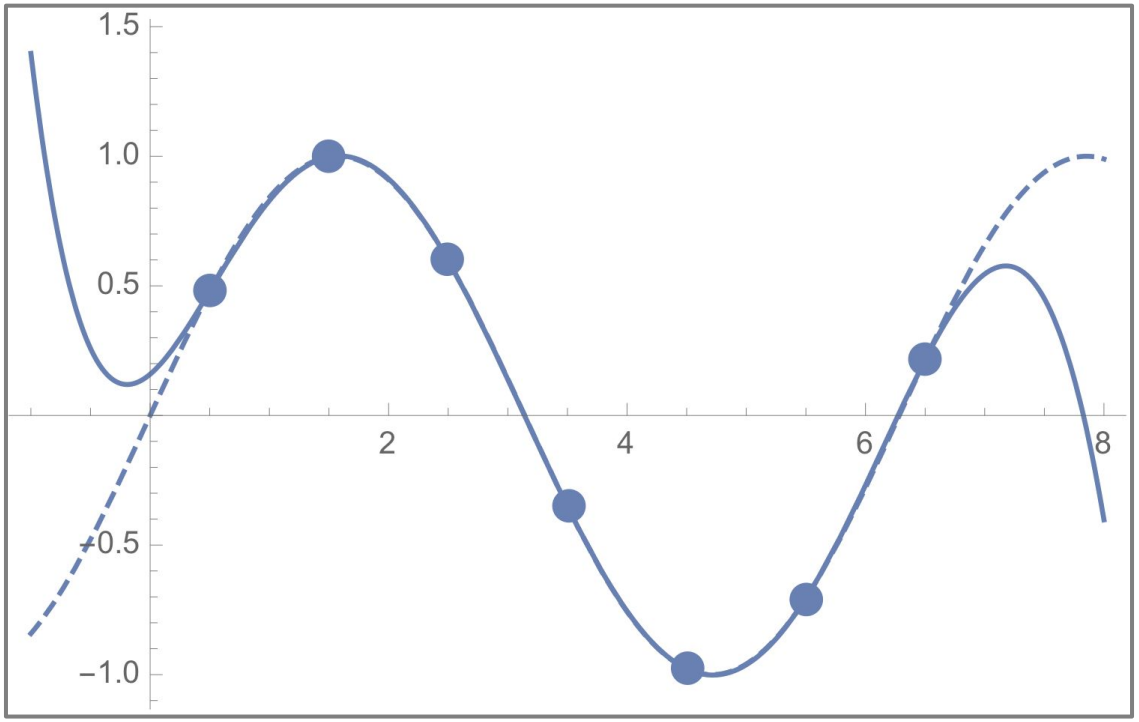


図 3-5: $(0.5+j, \sin(0.5+j))$, $j = 0,1, \ldots, 6$ の7点を通る曲線
$y = p_6(x)$（実線）および $y = \sin(x)$（破線）

曲線の推定⇒補間

- 曲線の推定は次のような問題に応用できる:
  - $f(x)$: unknown
  - Given: $y_j = f(x_j)$ for $j = 0, 1, \ldots, N$
  - Estimate value(s) of $f(x)$ at $x \neq x_j$
- 補間 (interpolation): 与えられた点以外での関数値を推定すること

補間＝関数値の推定

- ラグランジュ補間 (Lagrange's interpolation):
  ラグランジュの補間多項式による補間
- 図 3-5 では、与えられた7個の点ではすべて
  $y = \sin(x)$ を満たしており、両端の点に挟まれた領域では補間多項式による曲線（実線）が本来の曲線（点線）にほぼ重なっている

ラグランジュの補間多項式と中国剰余算法

- 中国剰余算法

  春学期「計算機数学I」、第10回

  https://www.math.tsukuba.ac.jp/~terui/compmath1-2018
- 実は、この時紹介した解の構成法は、ラグランジュの補間多項式に類似