

これまでのまとめ (学年末試験に向けて)

山本昌志*

2006年2月10日

1 はじめに

学年末試験に向けて，学習のポイントを示す．後期の中間試験から，これまで以下について数値計算の学習を行った．

- 連立一次方程式 (反復法)
 - － ヤコビ法
 - － ガウスザイデル法
 - － SOR 法
- 補間法
 - － ラグランジュの補間法
 - － スプライン補間法
- 数値積分
 - － 台形公式
 - － シンプソンの公式
 - － モンテカルロ積分
- 差分法による偏微分方程式の数値計算
 - － ラプラス方程式
 - － 波動方程式

ここでは，これらについて，簡単にまとめてある．これが学習の最重要ポイントであるのでしっかり勉強してほしい．このプリントを見ながら，分からない部分は授業中配ったプリントで補い，理解することが学習のこつである．

数値計算のいろいろな方法を学習したが，ほとんど全ての方法は1つの式に要約できる．この式を導き，それを使いこなせれば，ここでの授業での学習は完璧である．学年末試験では，使いこなせるか否かを調べることは難しいので，式を導くことを中心に出題する．

*独立行政法人 秋田工業高等専門学校 電気工学科

2 連立一次方程式 (反復法)

実用的なプログラムでは、非常に大きな連立方程式を計算しなくてはならない。数百万円に及ぶことも珍しくない。これを、ガウス・ジョルダン法で計算するの時間的にほとんど不可能である。そこで、これよりは格段に計算の速い方法が用いられる。ここでは、その一つとして反復法を簡単に説明する。

当然ここでも、連立方程式

$$Ax = b \quad (1)$$

を満たす x を数値計算により、求めることになる。ここで、真の解 x とする。

ここで、ある計算により n 回目で求められたものを x_n とする。そして、計算回数を増やして、

$$\lim_{n \rightarrow \infty} x_n = x \quad (2)$$

になったとする。この様に計算回数を増やして、真の解に近づける方法を反復法という。

この様な方法は、行列 A を $S - T$ と分解するだけで、容易に作るができる。たとえば、

$$Sx_{k+1} = Tx_k + b \quad (3)$$

とすればよい。ここで、 x_k が α に収束するとする。すると、式 (3) と式 (1) を比べれば、 α と x は等しいことがわかる。すなわち、式 (3) で元の方方程式 (1) を表した場合、 x_k が収束すれば、必ず真の解 x に収束するのである。別の解に収束することはなく、真の解に収束するか、発散するかのいずれかである。

2.1 ヤコビ法

係数行列 A の対角行列を反復計算の行列 S としたものがヤコビ (Jacobi) 法である。係数行列を

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & 0 & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & 0 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & 0 \end{bmatrix} \quad (4)$$

と分解し、右辺第 1 項が行列 S で第 2 項が $-T$ となる。 x_{k+1} の解の計算に必要な S の逆行列は、それが対角行列なので、

$$S^{-1} = \begin{bmatrix} a_{11}^{-1} & 0 & 0 & \dots & 0 \\ 0 & a_{22}^{-1} & 0 & \dots & 0 \\ 0 & 0 & a_{33}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{nn}^{-1} \end{bmatrix} \quad (5)$$

と簡単に求めることができる。 $k+1$ 番目の近似解は $x_{k+1} = S^{-1}(b + Tx_k)$ となるなので、容易に求めることができる。実際、 k 番目の解を

$$x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots, x_n^{(k)}$$

とすると、 $k+1$ 番目の解は

$$\begin{aligned}
 x_1^{(k+1)} &= a_{11}^{-1} \left\{ b_1 - \left(a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + a_{14}x_4^{(k)} + \cdots + a_{1n}x_n^{(k)} \right) \right\} \\
 x_2^{(k+1)} &= a_{22}^{-1} \left\{ b_2 - \left(a_{21}x_1^{(k)} + a_{23}x_3^{(k)} + a_{24}x_4^{(k)} + \cdots + a_{2n}x_n^{(k)} \right) \right\} \\
 x_3^{(k+1)} &= a_{33}^{-1} \left\{ b_3 - \left(a_{31}x_1^{(k)} + a_{32}x_2^{(k)} + a_{34}x_4^{(k)} + \cdots + a_{3n}x_n^{(k)} \right) \right\} \\
 &\vdots \\
 x_n^{(k+1)} &= a_{nn}^{-1} \left\{ b_n - \left(a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + a_{n3}x_3^{(k)} + \cdots + a_{nn-1}x_{n-1}^{(k)} \right) \right\}
 \end{aligned} \tag{6}$$

と計算できる．これを繰り返して連立方程式の解を求める方法が，ヤコビ法である．

2.2 ガウス・ザイデル法

ヤコビ法の特徴では， $x^{(k+1)}$ の近似値は，すべてその前の値 $x^{(k)}$ を使うことにある．大きな行列を扱う場合，全ての $x^{(k+1)}$ と $x^{(k)}$ を記憶する必要があり，大きなメモリーが必要となり問題が生じる．今では，個人で大きなメモリーを使い計算することは許されるが，ちょっと前まではできるだけメモリーを節約したプログラムを書かなくてはならなかった．

そこで， $x^{(k+1)}$ の各成分の計算が終わると，それを直ちに使うことが考えれば，メモリーは半分で済む．即ち， x_i^{k+1} を計算するときに，

$$\begin{aligned}
 x_i^{k+1} &= a_{ii}^{-1} \left\{ b_i - \left(a_{i1}x_1^{(k+1)} + a_{i2}x_2^{(k+1)} + a_{i3}x_3^{(k+1)} + \cdots + a_{ii-1}x_{i-1}^{(k+1)} + \right. \right. \\
 &\quad \left. \left. a_{ii+1}x_{i+1}^{(k)} + a_{ii+2}x_{i+2}^{(k)} + a_{ii+3}x_{i+3}^{(k)} + \cdots + a_{in}x_n^{(k)} \right) \right\}
 \end{aligned} \tag{7}$$

とするのである．実際の計算では， $k+1$ 番目の解は

$$\begin{aligned}
 x_1^{(k+1)} &= a_{11}^{-1} \left\{ b_1 - \left(a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + a_{14}x_4^{(k)} + \cdots + a_{1n}x_n^{(k)} \right) \right\} \\
 x_2^{(k+1)} &= a_{22}^{-1} \left\{ b_2 - \left(a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)} + a_{24}x_4^{(k)} + \cdots + a_{2n}x_n^{(k)} \right) \right\} \\
 x_3^{(k+1)} &= a_{33}^{-1} \left\{ b_3 - \left(a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{34}x_4^{(k)} + \cdots + a_{3n}x_n^{(k)} \right) \right\} \\
 &\vdots \\
 x_n^{(k+1)} &= a_{nn}^{-1} \left\{ b_n - \left(a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + a_{n3}x_3^{(k+1)} + \cdots + a_{nn-1}x_{n-1}^{(k+1)} \right) \right\}
 \end{aligned} \tag{8}$$

と計算できる．これを繰り返して連立方程式の解を求める方法が，ガウス・ザイデル法である．

2.3 SOR 法

ガウス・ザイデル法をもっと改善する方法がある．ガウス・ザイデル法の解の修正は， $x_{k+1} - x_k$ であったが，これをもっと大きなステップにしようというのである．通常の場合，ガウス・ザイデル法では近似解はいつも同じ側にあり，単調に収束する．そのため，修正を適当にすれば，もっと早く解に近づく．修正

幅を，加速緩和乗数 ω を用いて， $\omega(x_{k+1} - x_k)$ とする事が考えられた．これが，逐次加速緩和法 (SOR 法: Successive Over-Relaxation) である．

具体的な計算手順は，次のようにする．ここでは，ガウス・ザイデル法の式 (8) を用いて，得られた近似解を $\tilde{x}_i^{(k+1)}$ としている．

$$\begin{aligned}
 \tilde{x}_1^{(k+1)} &= a_{11}^{-1} \left\{ b_1 - \left(a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + a_{14}x_4^{(k)} + \cdots + a_{1n}x_n^{(k)} \right) \right\} \\
 x_1^{(k+1)} &= x_1^{(k)} + \omega \left(\tilde{x}_1^{(k+1)} - x_1^{(k)} \right) \\
 \tilde{x}_2^{(k+1)} &= a_{22}^{-1} \left\{ b_2 - \left(a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)} + a_{24}x_4^{(k)} + \cdots + a_{2n}x_n^{(k)} \right) \right\} \\
 x_2^{(k+1)} &= x_2^{(k)} + \omega \left(\tilde{x}_2^{(k+1)} - x_2^{(k)} \right) \\
 \tilde{x}_3^{(k+1)} &= a_{33}^{-1} \left\{ b_3 - \left(a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{34}x_4^{(k)} + \cdots + a_{3n}x_n^{(k)} \right) \right\} \\
 x_3^{(k+1)} &= x_3^{(k)} + \omega \left(\tilde{x}_3^{(k+1)} - x_3^{(k)} \right) \\
 &\vdots \\
 \tilde{x}_n^{(k+1)} &= a_{nn}^{-1} \left\{ b_n - \left(a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + a_{n3}x_3^{(k+1)} + \cdots + a_{nn-1}x_{n-1}^{(k+1)} \right) \right\} \\
 x_n^{(k+1)} &= x_n^{(k)} + \omega \left(\tilde{x}_n^{(k+1)} - x_n^{(k)} \right)
 \end{aligned} \tag{9}$$

これを繰り返して連立方程式の解を求める方法が，SOR 法である．

ここで，問題なのが加速緩和係数 ω の値の選び方である．明らかに，それが 1 の場合，ガウス・ザイデル法となりメリットは無い．また，1 以下だと，ガウス・ザイデル法よりも収束が遅い．ただし，ガウス・ザイデル法で収束しないような問題には使える．

従って，1 以上の値にしたいわけであるが，余り大きくすると，発散するのは目に見えている．これについては，2 を越えると発散することが分かっている．最適値となると，だいたい 1.9 くらいが選ばれることが多い．

3 補間法

実験やシミュレーションを行うと，離散的にデータが得られるのは普通である．例えば，半導体の電圧・電圧特性の測定では， $(V_0, I_0), (V_1, I_1), (V_2, I_2), (V_3, I_3), \dots, (V_N, I_N)$ のようなデータが得られる．通常，このデータはグラフ化して解析を進める．このデータの場合，2 次元のグラフ上に測定点が並ぶことは，今まで学習してきたとおりである．

実験等を通して得られる結果は離散的であるが，実際の現象は連続的なことが多い．この離散的な値を用いて，測定点の間の値，ここでは電流と電圧の関係を求めるのが補間法の役割である．ここで学習したラグランジュ補間もスプライン補間も，全てのグラフ上の測定点を通る曲線の方程式を求めている．

2 次元のグラフ上の点は，数学では座標 (x, y) の点として与えられる．以降の説明では，電圧・電流などのように特定の問題にとらわれないよう，一般化した座標 (x, y) で話を進める．

3.1 ラグランジュ補間

平面座標上に $N + 1$ 個の点がある場合、その全ての点を通る曲線は N 次関数で表せることは、数学で学習したとおりである¹。2 個の場合、1 次関数、すなわち直線で、その 2 点を通る関数を決めることができる。3 点の場合は、2 次関数である。

この性質を利用すると、 $N + 1$ 個の点がある場合、 N 次関数で補間できることが分かる。ラグランジュ補間とは、まさにこのことそのものである。数学の授業で、ある 3 点 (x_0, y_0) , (x_1, y_1) , (x_2, y_2) を通る 2 次関数 $y = ax^2 + bx + c$ の a, b, c を求めたことがあると思うが、それと同じである。そこでは、それぞれの x と y の値を代入して、連立方程式をつくり a, b, c を求めたはずである。

コンピューターを用いて、 $N + 1$ 個の点を通る N 次方程式を表す $N + 1$ 個の係数を連立方程式を解くことにより求めることは可能である。しかし、最終目的の N 次関数の値を求めると言う意味では不経済である。補間という目的からすると、関数を形成する係数なんか、全く興味の対象外なのである。そこで、係数が分からなくても、 N 次関数を示すものとして、ラグランジュ補間が使われる。

2 次元座標上に $N + 1$ 個の点、 (x_0, y_0) , (x_1, y_1) , (x_2, y_2) , \dots , (x_N, y_N) のラグランジュ補間は、

$$\begin{aligned}
 y = & \frac{(x - x_1)(x - x_2)(x - x_3) \cdots (x - x_N)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3) \cdots (x_0 - x_N)} y_0 + \frac{(x - x_0)(x - x_2)(x - x_3) \cdots (x - x_N)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3) \cdots (x_1 - x_N)} y_1 \\
 & + \frac{(x - x_0)(x - x_1)(x - x_3) \cdots (x - x_N)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3) \cdots (x_2 - x_N)} y_2 + \frac{(x - x_0)(x - x_1)(x - x_2) \cdots (x - x_N)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2) \cdots (x_3 - x_N)} y_3 \\
 & \cdots + \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_N)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_N)} y_k + \cdots \\
 & + \frac{(x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{N-1})}{(x_N - x_0)(x_N - x_1)(x_N - x_2) \cdots (x_N - x_{N-1})} y_N
 \end{aligned} \tag{10}$$

となる。

この式 (10) を見ると、

- 各項の分母は定数で、分子は N 次関数である。このことから、全ての項は N 次関数になっていることが理解できる。したがって、この式は N 次関数 (N 次多項式) である。
- x に $x_0, x_1, x_2, \dots, x_N$ を代入すると、 y の値は $y_0, y_1, y_1, \dots, y_N$ になることが分かる。これは、データ点 (x_0, y_0) , (x_1, y_1) , (x_2, y_2) , \dots , (x_N, y_N) の全てを通過していることを示している。

となっている。

式 (10) をもうちょっと格好良く書けば、

$$L(x) = \sum_{k=0}^N L_k(x) y_k \tag{11}$$

¹ x が同じで y が異なる複数の点が存在するような特別な場合を除く。

ただし,

$$\begin{aligned}
 L_k(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_N)}{(x_k-x_0)(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_N)} \\
 &= \frac{x-x_0}{x_k-x_0} \times \frac{x-x_1}{x_k-x_1} \times \frac{x-x_2}{x_k-x_2} \times \cdots \times \frac{x-x_{k-1}}{x_k-x_{k-1}} \times \frac{x-x_{k+1}}{x_k-x_{k+1}} \times \cdots \times \frac{x-x_N}{x_k-x_N} \\
 &= \prod_{j=0, j \neq k}^N \frac{x-x_j}{x_k-x_j}
 \end{aligned} \tag{12}$$

となる.

ラグランジュ補間の考え方は単純で、その計算も簡単である。しかし、補間の点数が増えてくると、ラグランジュの補間には問題が生じる。ラグランジュの補間では、補間の点数が増えてくると大きな振動が発生して、もはや補間とは言えなくなる。ラグランジュの補間には常にこの問題が付きまうので、データ点数が多い場合は使えなくなる。

3.2 スプライン補間

3.2.1 区分多項式

ラグランジュの補間は、データ点数が増えてくると関数が振動し問題が発生する。そこで、補間する領域をデータ間隔 $[x_i, x_{i+1}]$ に区切り、その近傍の値を使い低次の多項式で近似することを考える。区分的な関数を使うことになるが、上手に近似をしないと境界でその導関数が不連続になる。導関数が連続になるように、上手に近似する方法がスプライン補間 (spline interpolation) である。

補間をするデータは、先と同じように $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ とする。そして、区間 $[x_j, x_{j+1}]$ で補間をする関数を $S_j(x)$ とする。この様子を図 1 に示す。

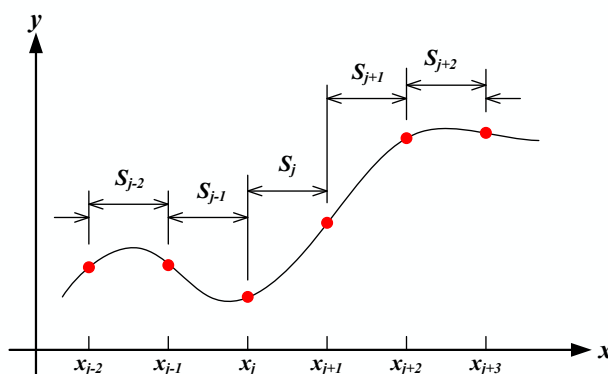


図 1: スプライン補間の区分

3.2.2 係数が満たす式

3 次のスプライン補間を考えるので、

$$S_j(x) = a_j(x - x_j)^3 + b_j(x - x_j)^2 + c_j(x - x_j) + d_j \quad (j = 0, 1, 2, 3, \dots, N - 1) \quad (13)$$

となる。スプライン補間を行う場合、この a_j, b_j, c_j, d_j を決めることが、主な作業である。

これらの $4N$ 個の未知数を決めるためには、 $4N$ 個の方程式が必要である。そのために、3 次のスプライン補間に以下の条件を課すことにする。

- 全てのデータ点を通る。各々の $S_j(x)$ に対して両端での値が決まるため、 $2N$ 個の方程式ができる。
- 各々の区分補間式は、境界点の 1 次導関数は連続とする。これにより、 $N - 1$ 個の方程式ができる。
- 各々の区分補間式は、境界点の 2 次導関数は連続とする。これにより、 $N - 1$ 個の方程式ができる。

以上の条件を課すと $4N - 2$ 個の方程式が決まる。未知数は $4N$ 個なので、2 個方程式が不足している。この不足を補うために、いろいろな条件が考えられるが、通常は両端 x_0 と x_N での 2 次導関数の値を 0 とする。すなわち、 $S_0''(x_0) = S_{N-1}''(x_N) = 0$ である。これを自然スプライン (natural spline) と言う。自然スプライン以外には、両端の 1 次導関数の値を指定するものもある。

これで全ての条件が決まった。あとは、この条件に満たす連立方程式を求めるだけである。

4 数値積分

4.1 台形公式

定積分、

$$S = \int_a^b f(x) dx \quad (14)$$

の近似値を数値計算で求めることを考える。積分の計算は面積の計算であるから、図 2 のように台形的面積の和で近似ができるであろう。積分の範囲 $[a, b]$ を N 等分した台形で近似した面積 T は、

$$\begin{aligned} T &= h \frac{f(a) + f(a+h)}{2} + h \frac{f(a+h) + f(a+2h)}{2} + h \frac{f(a+2h) + f(a+3h)}{2} + \dots \\ &\quad + h \frac{f(a+(N-1)h) + f(a+Nh)}{2} \\ &= \frac{h}{2} \sum_{j=0}^{N-1} [f(a+jh) + f(a+(j+1)h)] \end{aligned} \quad (15)$$

となる。これが数値積分の台形公式である。なんのことはない、積分を台形的面積に置き換えているだけである。

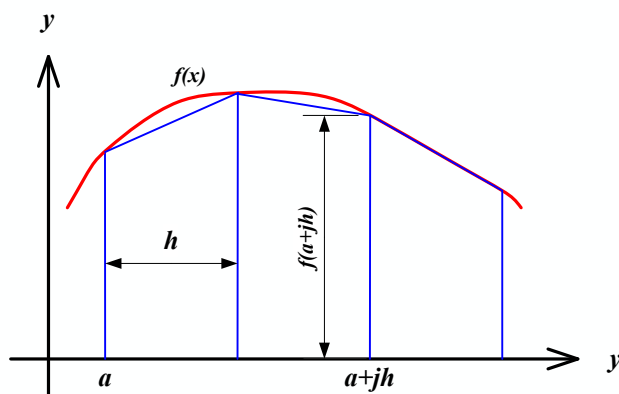


図 2: 積分と台形の面積の比較

4.2 シンプソンの公式

台形公式の考え方は簡単であるが，精度はあまりよくない．そこで，よく似た考え方で精度が良いシンプソンの公式を説明する．台形公式は，分割点の値を一次関数（直線）で近似を行い積分を行った．要するに折れ線近似である．ここで，1 次関数ではなく，高次の関数で近似を行えばより精度が上がることは，直感的に分かる．

2 次関数で近似を行うことを考える．2 次関数で近似するためには，3 点必要である．3 つの分点をそれぞれ， (x_j, x_{j+1}, x_{j+2}) とする．そして，この 2 次関数を $P(x)$ とする． $P(x)$ はラグランジュ補間に他ならないので，

$$P(x) = \frac{(x - x_{j+1})(x - x_{j+2})}{(x_j - x_{j+1})(x_j - x_{j+2})} f(x_j) + \frac{(x - x_j)(x - x_{j+2})}{(x_{j+1} - x_j)(x_{j+1} - x_{j+2})} f(x_{j+1}) + \frac{(x - x_j)(x - x_{j+1})}{(x_{j+2} - x_j)(x_{j+2} - x_{j+1})} f(x_{j+2}) \quad (16)$$

となる．図 3 に示すとおりである．

これを，区間 $[x_j, x_{j+2}]$ で積分する．紙面の都合上，式 (17) の右辺を各項毎に積分を行う．まず，右辺

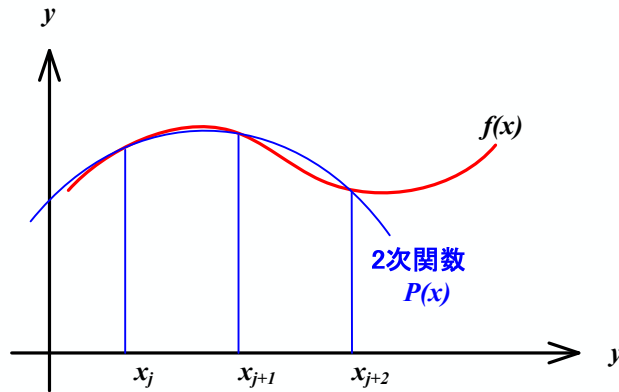


図 3: 元の関数を区間 $[x_j, x_{j+2}]$ を 2 次関数で近似する

第 1 項であるが，それは以下ようになる．

$$\begin{aligned}
 \text{式 (17) 右辺第 1 項の積分} &= \int_{x_j}^{x_{j+2}} \frac{(x - x_{j+1})(x - x_{j+2})}{(x_j - x_{j+1})(x_j - x_{j+2})} f(x_j) dx \\
 &= \int_0^{2h} \frac{(x_j + \xi - x_{j+1})(x_j + \xi - x_{j+2})}{(x_j - x_{j+1})(x_j - x_{j+2})} f(x_j) d\xi \\
 &= \int_0^{2h} \frac{(\xi - h)(\xi - 2h)}{(-h)(-2h)} f(x_j) d\xi \\
 &= \frac{f(x_j)}{2h^2} \int_0^{2h} (\xi^2 - 3h\xi + 2h^2) d\xi \\
 &= \frac{f(x_j)}{2h^2} \left[\frac{\xi^3}{3} - 3h\frac{\xi^2}{2} + 2h^2\xi \right]_0^{2h} \\
 &= \frac{h}{3} f(x_j)
 \end{aligned} \tag{17}$$

同様に，第 2,3 項を計算すると

$$\text{式 (17) 右辺第 2 項の積分} = \frac{4h}{3} f(x_{j+1}) \tag{18}$$

$$\text{式 (17) 右辺第 3 項の積分} = \frac{h}{3} f(x_{j+2}) \tag{19}$$

となる．以上より，近似した 2 次関数 $P(x)$ の範囲 $[x_j, x_{j+2}]$ の積分は，

$$\int_{x_j}^{x_{j+2}} P(x) dx = \frac{h}{3} \{f(x_j) + 4f(x_{j+1}) + f(x_{j+2})\} \tag{20}$$

となる．

これは，ある区間 $[x_j, x_{j+2}]$ の積分で，その巾は $2h$ である．区間 $[a, b]$ にわたっての積分 S は，式 (21)

を足し合わせればよい．ただし， $j = 0, 2, 4, 6$ と足し合わせる．

$$\begin{aligned}
 S &= \frac{h}{3} \{f(x_0) + 4f(x_1) + f(x_2)\} + \frac{h}{3} \{f(x_2) + 4f(x_3) + f(x_4)\} + \frac{h}{3} \{f(x_4) + 4f(x_5) + f(x_6)\} + \cdots \\
 &\quad \cdots + \frac{h}{3} \{f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)\} \\
 &= \frac{h}{3} \{f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + 4f(x_5) + 2f(x_6) + \cdots \\
 &\quad \cdots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)\}
 \end{aligned} \tag{21}$$

これが，シンプソンの公式と呼ばれるもので，先ほどの台形公式よりも精度が良い．精度は， N^4 に反比例する．

この式から，分割数 N は偶数でなくてはならないことがわかる．

4.3 モンテカルロ法

積分の境界が複雑な場合，乱数を使うモンテカルロ積分が適している．例えば，関数 $f(x_1, x_2, x_3, \dots, x_M)$ の体積積分を考える．この体積積分は

$$\int_{\Omega} f(x_1, x_2, x_3, \dots, x_M) dx_1 dx_2 dx_3 \cdots dx_M = V_{\Omega} \langle f \rangle \tag{22}$$

となる．ここで， V は M 次元体の領域 Ω の体積， $\langle f \rangle$ はその内部の関数の平均値である．これは 3 次元体の質量を考えると簡単である．その質量は，密度の体積積分となる．これは体積に平均密度を乗じた値に等しい．当たり前の式である．このことから，式 (23) の積分の値が欲しければ，体積と平均値が分かればよいことになる．

積分を計算するために，まずは体積である．これは体積の計算が容易な単純な形状の内部に，領域 Ω を包み込み，その内部にランダムに配置されたサンプル点の数を数えれば良いのである．単純な形状内部に配置されたランダムな点の数を N とする．そして，その内部にある積分領域 Ω に含まれる点の数を N_{Ω} とする．さらに，単純な形状の体積を V_r ，領域 Ω のそれを V_{Ω} とすると，

$$V_{\Omega} \simeq \frac{N_{\Omega}}{N} V_r \tag{23}$$

の関係がある．右辺はコンピューターにより容易に計算できる．ランダムな点の数 N が多くなればなるほど，近似の精度は良くなる．

残りは，体積内部の平均 $\langle f \rangle$ である．これも簡単で，領域 Ω 内部にあるサンプル点の平均より求めることができる．即ち，

$$\langle f \rangle \simeq \frac{1}{N_{\Omega}} \sum_i f(r_i) \quad \text{領域 } \Omega \text{ の内部のみ} \tag{24}$$

となる．ここで， r_i は i 番目のサンプル点の $(x_1, x_2, x_3, \dots, x_M)$ 座標である．また， N_{Ω} は領域内部のサンプル数である．この計算も簡単で，コンピューターは難なく，平均値の近似値を計算することができる．

以上より，モンテカルロ法を用いると，体積 V と平均値 $\langle f \rangle$ の近似値が計算できることが分かる．従って，式 (23) の近似値を求めることができる．

5 差分法による偏微分方程式の数値計算

5.1 ラプラス方程式

2次元のラプラス方程式

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (25)$$

を数値計で解くことを考える．まずは，いつものように，解 $\phi(x, y)$ をテイラー展開する． x および y 方向に微小変位 $\pm h$ があつた場合，

$$\phi(x+h, y) = \phi(x, y) + \frac{\partial \phi}{\partial x} h + \frac{1}{2!} \frac{\partial^2 \phi}{\partial x^2} h^2 + \frac{1}{3!} \frac{\partial^3 \phi}{\partial x^3} h^3 + \frac{1}{4!} \frac{\partial^4 \phi}{\partial x^4} h^4 + \dots \quad (26)$$

$$\phi(x-h, y) = \phi(x, y) - \frac{\partial \phi}{\partial x} h + \frac{1}{2!} \frac{\partial^2 \phi}{\partial x^2} h^2 - \frac{1}{3!} \frac{\partial^3 \phi}{\partial x^3} h^3 + \frac{1}{4!} \frac{\partial^4 \phi}{\partial x^4} h^4 - \dots \quad (27)$$

となる．これらの式の辺々を足し合わせると，

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{x,y} = \frac{1}{h^2} [\phi(x+h, y) - 2\phi(x, y) + \phi(x-h, y)] - O(h^2) \quad (28)$$

が得られる．このことから，2階の偏導関数の値は微小変位 h の場所の関数の値を用いて， h^2 の精度で近似計算ができることが分かる．すなわち，式 (29) の $O(h^2)$ を除いた右辺を計算すればよいのである．同じことを y 方向についても行うと

$$\left. \frac{\partial^2 \phi}{\partial y^2} \right|_{x,y} = \frac{1}{h^2} [\phi(x, y+h) - 2\phi(x, y) + \phi(x, y-h)] - O(h^2) \quad (29)$$

が得られる．

これらの式 (29) と (30) を元の2次元ラプラス方程式 (26) に代入すれば，

$$\phi(x+h, y) + \phi(x-h, y) + \phi(x, y+h) + \phi(x, y-h) - 4\phi(x, y) = 0 \quad (30)$$

となる．これが，2次元ラプラス方程式の差分の式である．この式を眺めると，座標 (x, y) のポテンシャルの値 $\phi(x, y)$ は，周りの値の平均であることがわかる．

実際にこの式を数値計算する場合，計算領域を間隔 h で格子状²に区切り，その交点での値を求めることになる．ここでは， x および y 方向には等間隔 h で区切り計算を進めるが，等間隔である必要はない．多少，式 (31) は異なるが同じような計算は可能である．これまでの説明が理解できていれば， x と y 方向の間隔が異なっても，式 (31) に対応する差分の式が作れるはずである．

実際，数値計算をする場合， $\phi(x, y)$ や $\phi(x+h, y)$ の形は不便なので，形式を改める．各格子点でのポテンシャルを

$$\begin{aligned} \phi(x, y) &= \phi(ih, jh) \\ &= U_{ij} \end{aligned} \quad (31)$$

²この格子のことをメッシュ(mesh)と言う事もある．

とする．このようにすると，式 (31) は

$$U_{i+1j} + U_{i-1j} + U_{ij+1} + U_{ij-1} - 4U_{ij} = 0 \quad (32)$$

となり，数値計算し易い形になる．

ラプラス方程式は式 (33) の連立方程式を解くだけである．格子に領域を分割することにより，難しげな偏微分方程式が連立方程式に還元されたわけである．

連立方程式を解くわけであるが，このままでは，式の数と未知数の数が異なる．格子点でのポテンシャルの値を求めるためには，境界条件を設定する必要がある．それにより，式の数と未知数の数が同一になり，格子点でのポテンシャルを求めることができる．

5.2 波動方程式

弦の長さが 1，そこを伝わる波の速度を 1 として，弦の横波の様子を数値計算で解くことを考える．1 次元波動方程式を数値計で解くことを考える．計算に移る前に，解くべき方程式と条件をきちんと書いておく．解くべき方程式と条件は，

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2} & (0 \leq x \leq 1, \quad 0 \leq t) \\ u(x, 0) = \phi(x), \quad \frac{\partial u}{\partial t}(x, 0) = \psi(x) & (0 \leq x \leq 1) \\ u(0, t) = u(1, t) = 0 \end{cases} \quad (33)$$

となる．弦を伝わる波の速度は 1，弦の長さも 1 としている．この最初の式は波動方程式であるが，2 番目を初期条件，3 番目を境界条件と言う．

波動方程式の他に，初期条件と境界条件がある．力学的状態は，ある時刻，ここでは $t = 0$ の時の変位とその変位の速度が決まれば，それ以降を決めることができる．振動の場合は，これに加えて更に，振動の境界条件を決める必要がある．これらが決まって初めて，波動方程式とともに，振動の状態，ある時刻と位置の変位の値が決まるわけである．図 4 に初期条件と境界条件の様子を示す．

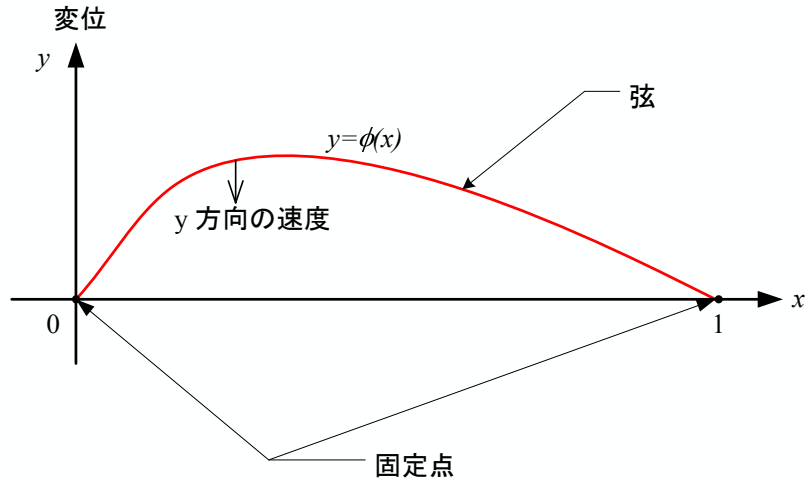


図 4: 時刻 $t = 0$ のときの弦の様子 (スナップショット) . 初期条件と境界条件が表されており , y 方向の速度が $\psi(x)$ になっている .

まずは , 波動方程式を差分方程式に書き直すことから始める . これも , いつものように , 解 $u(x, t)$ をテイラー展開する . x 方向の微小変位を Δx , 時間軸方向の微小変位を Δt とする . すると ,

$$\begin{aligned} u(x + \Delta x, t) &= u(x, t) + \frac{\partial u}{\partial x} \Delta x + \frac{1}{2!} \frac{\partial^2 u}{\partial x^2} (\Delta x)^2 + \frac{1}{3!} \frac{\partial^3 u}{\partial x^3} (\Delta x)^3 + \frac{1}{4!} \frac{\partial^4 u}{\partial x^4} (\Delta x)^4 + \dots \\ u(x - \Delta x, t) &= u(x, t) - \frac{\partial u}{\partial x} \Delta x + \frac{1}{2!} \frac{\partial^2 u}{\partial x^2} (\Delta x)^2 - \frac{1}{3!} \frac{\partial^3 u}{\partial x^3} (\Delta x)^3 + \frac{1}{4!} \frac{\partial^4 u}{\partial x^4} (\Delta x)^4 - \dots \end{aligned} \quad (34)$$

となる . これらの式の辺々を足し合わせると ,

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{x,t} = \frac{1}{\Delta x^2} [u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)] - O(\Delta x^2) \quad (35)$$

が得られる . このことから , 2 階の偏導関数の値は微小変位 Δx の場所の関数の値を用いて , $(\Delta x)^2$ の精度で近似計算ができることが分かる . すなわち , 式 (36) の右辺の第 1 項を計算すればよいのである . ラプラス方程式と同じである . 同様なことを時間軸方向についても行うと

$$\left. \frac{\partial^2 u}{\partial t^2} \right|_{x,t} = \frac{1}{\Delta t^2} [u(x, t + \Delta t) - 2u(x, t) + u(x, t - \Delta t)] - O(\Delta t^2) \quad (36)$$

が得られる .

これらの式 (36) と (37) を元の波動方程式 (34) に代入すれば ,

$$\frac{1}{\Delta x^2} [u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)] = \frac{1}{\Delta t^2} [u(x, t + \Delta t) - 2u(x, t) + u(x, t - \Delta t)] \quad (37)$$

となる．これが，1次元波動方程式の差分の式である．この式を計算し易いように，もう少し変形すると，

$$u(x, t + \Delta t) = 2u(x, t) - u(x, t - \Delta t) + \frac{\Delta t^2}{\Delta x^2} [u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)] \quad (38)$$

とすることができる．この式の右辺は，時刻 t と $t - \Delta t$ の値である．そして，左辺は時刻 $t + \Delta t$ の値である．このことから，式 (38) を用いると，時刻 t と $t - \Delta t$ の値から， $t + \Delta t$ の値が計算できることになる．

実際に式 (38) を数値計算する場合， x 方向には Δx ，時間軸方向には Δt 毎に分割する．ラプラス方程式を格子点で分割したのと同じである．格子点に分割し数値計算する場合， $u(x, t)$ や $u(x + \Delta x, t)$ と表現するよりは， u_{ij} と表現したほうが便利である．そこで，

$$\begin{aligned} u(x, t) &= \phi(i\Delta x, j\Delta t) \\ &= u_{ij} \end{aligned} \quad (39)$$

と表現を改める．このようにすると，式 (38) は

$$u_{ij+1} = 2u_{ij} - u_{ij-1} + \alpha (u_{i+1j} - 2u_{ij} + u_{i-1j}) \quad (40)$$

となり，数値計算し易い形になる．ただし，

$$\alpha = \frac{\Delta t^2}{\Delta x^2} \quad (41)$$

である．