# 補間(interpolation)と数値積分

### ー数値計算(07/11/30)ー

**関西学院大学理工学部 西谷滋人**



## 補間(interpolation)

### 補間と近似

最も単純な2次元データについて補間と近似を考える．補間はたんに点をつなぐことを，近似はある関数にできるだけ近くなるようにフィットすることを言う．本章では補間とそれに密接に関連した積分について述べる．

### 多項式

データを単純に多項式で補間する方法を先ず示そう．$N+1$点を$N$次の多項式でつなぐ．この場合の補間関数は，

$$F(x)=\sum_{i=0}^{N} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N$$

である．データの点を$x_i, y_i$とすると

$$a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_N x_0^N = y_0$$

$$a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_N x_1^N = y_1$$
$$\vdots$$
$$a_0 + a_1 x_{N+1} + a_2 x_{N+1}^2 + \cdots + a_N x_{N+1}^N = y_{N+1}$$

が，係数 $a_i$を未知数と見なした線形の連立方程式となっている．係数行列は

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & x_1^N \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_{N+1}^N \end{bmatrix}$$

となる．$a_i$と$y_i$をそれぞれベクトルとみなすと

$$\boxed{\phantom{x}}$$

により未知数ベクトル$a_i$が求まる．これは単純に，前に紹介した Gauss の消去法や LU 分解で解ける．

## 多項式補間(polynomial interpolation)

```
> restart;
  X:=[0,1,2,3]:
  Y:=[1,2,3,-2]:
> with(LinearAlgebra):with(plots):
> list1:=[X,Y];
```

$$list1 := [[0, 1, 2, 3], [1, 2, 3, -2]] \qquad (2.3.1)$$

```
> A:=Matrix(4,4):
  for i from 1 to 4 do
  for j from 1 to 4 do
    A[i,j]:=X[i]^(j-1);
  end do;
  end do:
> A;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix}$$

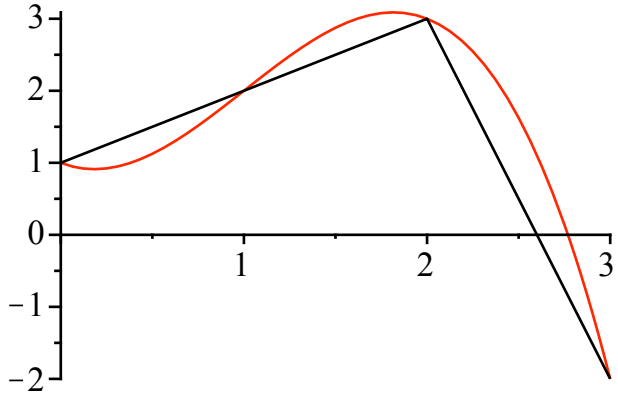
```
> a1:=MatrixInverse(A).Transpose(Matrix(Y));
```

$$a1 := \begin{bmatrix} 1 \\ -1 \\ 3 \\ -1 \end{bmatrix} \qquad (2.3.2)$$

```
> f1:=unapply(add(a1[ii,1]*x^(ii-1),ii=1..4),x);
```

$$f1 := x \to 1 - x + 3x^2 - x^3 \qquad (2.3.3)$$

```
> f1p:=plot(f1(x),x=0..3):
  l1p:=listplot(Transpose(Matrix(list1))):
  display(f1p,l1p);
```




## Lagrange(ラグランジュ) の内挿公式

多項式補間は手続きが簡単であるため，計算間違いが少ないように思えるが，あまり"みとうし"のよい方法とはいえない．その点，Lagrange(ラグランジュ) の内挿公式は見通しがよい．これは，

$$F(x) = \sum_{k=0}^{N} \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)} f_k = \sum_{k=0}^{N} \frac{\frac{(x - x_0)(x - x_1) \cdots (x - x_N)}{(x - x_k)}}{\frac{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_N)}{(x_k - x_k)}} f_k$$

と表わされる．数学的に 2 つ目の表記は間違っているが，先に割り算を実行すると読み取って欲しい．これは一見複雑に見えるが，単純な発想から出発している．求めたい関数$F(x)$を

$$F(x) = y_1 l_1(x) + y_2 l_2(x) + y_3 l_3(x)$$

とすると

$$l_1(x_1) = 1 \; l_2(x_2) = 0 \; l_3(x_3) = 0$$
$$l_1(x_1) = 0 \; l_2(x_2) = 1 \; l_3(x_3) = 0$$
$$l_1(x_1) = 0 \; l_2(x_2) = 0 \; l_3(x_3) = 1$$

となるように関数$l_i(x)$を決めればよい．これは，

## Newton(ニュートン) の内挿公式

Newton(ニュートン) の内挿公式は,

$$F(x) = F(x_0) + (x - x_0) f_1[x_0 - x_1] + (x - x_0)(x - x_1) f_2[x_0, x_1, x] + \cdots + \prod_{i=0}^{N-1} (x - x_i) f_N[x_0, x_1, \cdots, x_N]$$

となる．ここで$f_n[\ ]$は次のような関数を意味していて,

$$f_1[x_0, x_1] = \frac{y_0 - y_1}{x_0 - x_1}$$

$$f_2[x_0, x_1, x_2] = \frac{f_1[x_0, x_1] - f_1[x_1, x_2]}{x_0 - x_2}$$

$$\vdots$$

$$f_N[x_0, x_1, \cdots, x_n] = \frac{f_N[x_0, x_1, \cdots, x_{n-1}] - f_N[x_0, x_1, \cdots, x_n]}{x_0 - x_n}$$

差分商と呼ばれる．得られた多項式は，Lagrange の内挿公式で得られたものと当然一致する.

### ニュートンの補間公式の導出

## 数値積分 (Numerical integration)

積分,

$$I = \int_a^b f(x)\, \mathrm{d}x$$

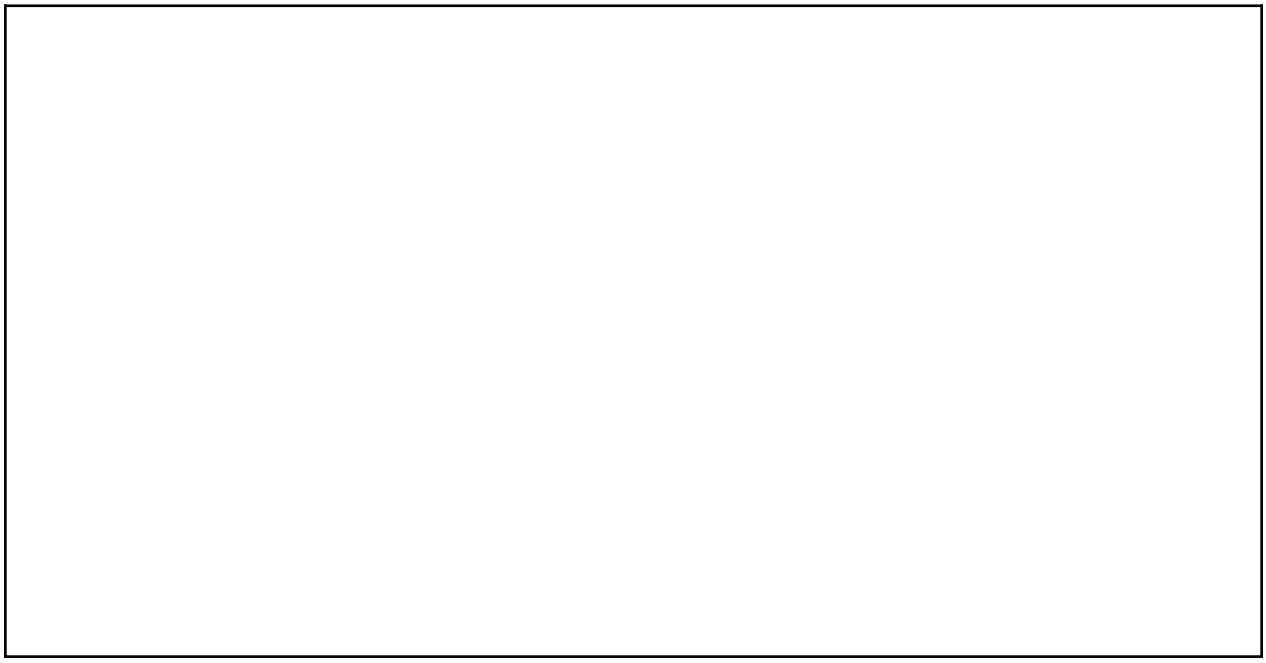
を求めよう．1次元の数値積分法では連続した領域を細かい短冊に分けて，それぞれの面積を寄せ集めることに相当する．分点の数を $N$ とすると,

$$x_i = \frac{b-a}{N} i = h \times i, h = \frac{b-a}{N}$$

ととれる．そうすると，もっとも単純には,

$$I_N = \sum_{i=0}^{N-1} f\left(x_i\right) h = \sum_{i=0}^{N-1} f\left(i \times h\right) h$$

となる.



## 中点則 (midpoint rule)

中点法 (midpoint rule) は，短冊を左端から書くのではなく，真ん中から書くことに対応し,

$$I_N = \sum_{i=0}^{N-1} f\left(\left(i + \frac{1}{2}\right) \times h\right) h$$

となる.

## 台形則 (trapezoidal rule)

さらに短冊の上側を斜めにして，短冊を台形にすれば精度が上がりそうに思う.
その場合は，短冊一枚の面積$S_i$は,

$$S_i = \frac{f\left(x_i\right) + f\left(x_{i+1}\right)}{2} h$$

で求まる．これを端から端まで加えあわせると,

$$I_N = \sum_{i=0}^{N-1} S_i = h\left\{\frac{1}{2} f\left(x_0\right) + \sum_{i=0}^{N-1} f\left(x_i\right) + \frac{1}{2} f\left(x_N\right)\right\}$$

が得られる.

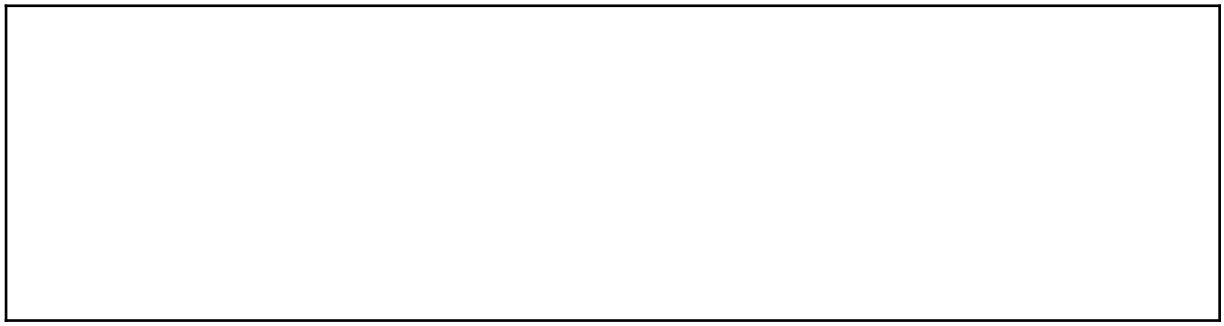
## Simpson(シンプソン) 則

Simpson(シンプソン) 則では，短冊を 2 次関数,

$$f(x) = ax^2 + bx + c$$

で近似することに対応する．こうすると，

$$S_i = \int_{x_i}^{x_{i+1}} f(x)\,dx = \int_{x_i}^{x_{i+1}} ax^2 + bx + c\,dx$$



$$\frac{h}{6}\left\{ f(x_i) + 4f\left(x_i + \frac{h}{2}\right) + f(x_{i+h}) \right\}$$

となる．これより，

$$I_N = \frac{h}{6}\left\{ f(x_0) + 4\sum_{i=0}^{N-1} f\left(x_i + \frac{h}{2}\right) + 2\sum_{i=1}^{N-1} f(x_i) + f(x_N) \right\}$$

として計算できる．ただし，関数値を計算する点は台形則などの倍となっている．

教科書によっては，分割数 N を偶数にかぎって，点を偶数番目 (even) と奇数番目 (odd) に分
けて，

$$I_N = \frac{h}{3}\left\{ f(x_0) + 4\sum_{i=even}^{N-2} f\left(x_i + \frac{h}{2}\right) + 2\sum_{i=odd}^{N-1} f(x_i) + f(x_N) \right\}$$

としているテキストもあるが，同じ計算になるので誤解せぬよう．

## Derivation of Simpson's rule

```
> restart;
  f:=x->a*x^2+b*x+c;
  e1:=expand(subs(x1=x0+h,int(f(x),x=x0..x1)));
```

$$f := x \to a x^2 + b x + c$$

$$e1 := a\,x0^2\,h + a\,x0\,h^2 + \frac{1}{3}\,a\,h^3 + b\,x0\,h + \frac{1}{2}\,b\,h^2 + c\,h$$

```
> e2:=expand(h/6*(f(x0)+4*f(x0+h/2)+f(x0+h)));
```

$$e2 := a\,x0^2\,h + a\,x0\,h^2 + \frac{1}{3}\,a\,h^3 + b\,x0\,h + \frac{1}{2}\,b\,h^2 + c\,h$$

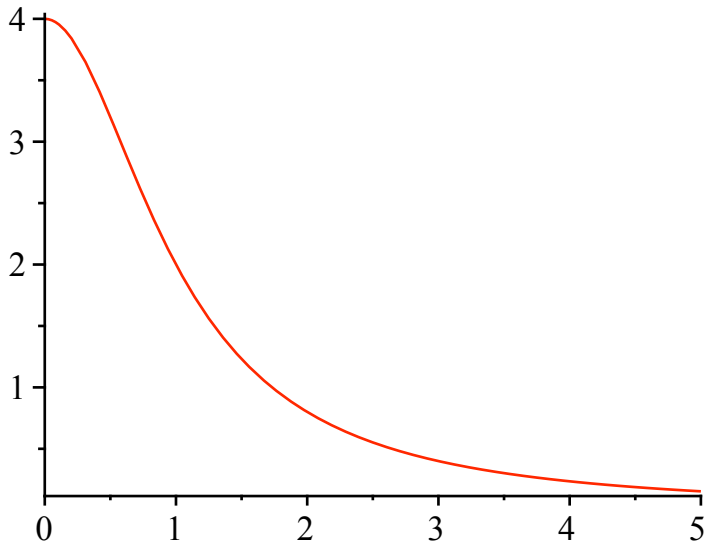
```
> evalb(e1=e2);
```

$$true$$

# Numerical integration

```
> restart;
  f1:=x->4/(1+x^2);
  plot(f1(x),x=0..5);
```

$$f1 := x \rightarrow \frac{4}{1+x^2}$$




```
> int(f1(x),x=0..1);
```

$$\pi$$

```
> int(1/(1+x^2),x);
```

$$\arctan(x)$$

```
> N:=8:
  x0:=0:
  xn:=1:
  Digits:=20:
```

## Midpoint rule(中点法)

```
> h:=(xn-x0)/N:
  S:=0:
  for i from 0 to N-1 do
    xi:=x0+(i+1/2)*h;
    dS:=h*f1(xi);
    S:=S+dS;
  end do:
  evalf(S);
```

3.1428947295916887799

## Trapezoidal rule(台形公式)

```
> h:=(xn-x0)/N:
  S:=f1(x0)/2:
  for i from 1 to N-1 do
    xi:=x0+i*h;
    dS:=f1(xi);
    S:=S+dS;
  end do:
  S:=S+f1(xn)/2:
  evalf(h*S);
```

3.1389884944910890093

## Simpson's rule(シンプソンの公式)

```
> M:=N/2:
  h:=(xn-x0)/(2*M):
  Seven:=0:
  Sodd:=0:
  for i from 1 to 2*M-1 by 2 do
    xi:=x0+i*h;
    Sodd:=Sodd+f1(xi);
  end do:
  for i from 2 to 2*M-1 by 2 do
    xi:=x0+i*h;
    Seven:=Seven+f1(xi);
  end do:
  evalf(h*(f1(x0)+4*Sodd+2*Seven+f1(xn))/3);
                    3.1415925024587069144
```

## 課題

1 多項式補間

次の4点

x y

0 1

1 2

2 3

3 -2

を通る多項式を以下のそれぞれの手法で求めよ.

(a) 逆行列

(b)ラグランジュ補間

(c)ニュートンの差分商公式

2 数値積分

次の関数

$$y = \frac{1}{1+x^2}$$

を$x=0\cdots1$で等間隔に$N+1$点とり，$N$個の区間にわけて数値積分で求める．$N$を2, 4, 8, 16, 32, 64, 128, 256と取ったときのそれぞれの手法の収束せいを比較せよ.

(a)中点法

(b)台形公式

(c)シンプソン公式

ヒント：Maple script にあるそれぞれの数値積分法を関数 (procedure) に直して，for-loop で回せば楽．出来なければ，一つ一つ手で変えても OK. 両対数プロット (loglogplot) すると見やすい.