2 変数関数のスプライン関数による取り扱いについて

中野武雄 nakano@apm.seikei.ac.jp

2010-1-27

概 要

スパッタリング法のモンテカルロシミュレーションでは、粒子の散乱角をエネルギーと衝突パラメータの関数として求める。本文書は、この取り扱いをスプライン関数で行うプログラムルーチン作成に当たってのメモである。

1 B スプライン

B スプラインとは、スプライン関数の基底関数 (basis) を構成するものである。 n 次の B スプラインは $n+2$ 個のノード間でのみ値を持ち、その他の領域では 0 になるような関数である。また、有限領域の両端では $n-1$ 次の微係数が 0 という性質を持つ。

この B スプラインは、切断べき級数の差分商によって与えられる。 M 次の切断べき級数は

$$(x-q_i)_+^M = \begin{cases} 0 & (x < q_i) \\ (x-q_i)^M (x \geq q_i) \end{cases} \tag{1}$$

と定義される関数である。

一方差分商は以下のように定義される。いま関数 $f(x)$ が異なる座標 $q_i, q_{i+1}, \ldots, q_{i+N}$ に対して定義されているとする。

1. 関数 $f(x)$ に関する 1 階差分商 $f(q_i, q_{i+1})$ を以下のように定義する。

$$f(q_i, q_{i+1}) = \frac{f(q_{i+1}) - f(q_i)}{q_{i+1} - q_i} \tag{2}$$

すなわちこれは $y_i = f(q_i)$ としたときのデータ点 (q_i, y_i) と (q_{i+1}, y_{i+1}) を結ぶ直線の傾きである。

2. $r+1$ 階の差分商 $f(q_i, q_{i+1}, \ldots, q_{i+r+1})$ は r 階の差分商から以下の漸化式で求められる。

$$f(q_i, q_{i+1}, \ldots, q_{i+r+1}) = \frac{f(q_{i+1}, q_{i+2}, \ldots, qi+r+1) - f(q_i, q_{i+1}, \ldots, qi+r)}{q_{i+r+1} - q_i} \tag{3}$$

これらを用いて、$K-1$ 次の B スプライン $B_{i,k}(x)$ を以下のように定義する。

$$B_{i,k}(x) = (q_{i+K} - q_i) M_K(x : q_i, q_{i+1}, \ldots, q_{i+K}) \tag{4}$$

ここで $M_K(x : q_i, q_{i+1}, \ldots, q_{i+K})$ は

$$M_K(x : q) = (q-x)_+^{K-1} \tag{5}$$

に対して $q_i, q_{i+1}, \ldots, q_{i+K}$ の接点で K 階の差分商を取ったものである。このとき各接点は $q_i \leq q_{i+1} \leq \cdots \leq q_{i+K}$ を満たしているものとする。

すると 0 次のスプライン関数 $B_{i,1}(x)$ は

$$B_{i,1}(x) = \begin{cases} 1 & (q_i \leq x < q_{i+1}) \\ 0 & (x < q_i, q_{i+1} \leq x) \end{cases} \tag{6}$$

となり、 $K-1$ 次の B スプラインは漸化式

$$B_{i,K}(x) = \frac{x - q_i}{q_{i+K-1} - q_i} B_{i,K-1}(x) + \frac{q_{i+K} - x}{q_{i+K} - q_{i+1}} B_{i+1,K-1}(x) \tag{7}$$

と表すことができる。

2　1 次元スプライン関数の導出

前節で定義された B スプライン関数を用いて、 K-1 次のスプライン関数を構成する。スプライン関数 $s(x)$ は

$$s(x) = \sum_{i=0}^{N-1} \alpha_i B_{i,K}(x) \tag{8}$$

で定義され、与えられた N 個の点 $(x_0, y_0), (x_1, y_1), \ldots, (x_{N-1}, y_{N-1})$ を通るものとする。このとき $B_{i,K}$ が $q_i, \ldots, q_{i+K}$ の K 個のデータ点から構成されるから、式 8 を構成するためには $N+K$ 個の接点が必要となる。さらに N 個の各データ点を $s(x)$ が通るようにするには、 B スプラインは

$$B_{i,K}(x_i) \neq 0, \qquad (i = 0, 1, \ldots, N-1) \tag{9}$$

を満たす必要がある。各 B スプラインを構成する接点から式 9 を書き換えると

$$\begin{gathered} q_0 < x_0 < q_{0+K} \\ q_1 < x_1 < q_{1+K} \\ \vdots \\ q_{N-1} < x_{N-1} < q_{N-1+K} \end{gathered} \tag{10}$$

となる。この条件は Shoenberg-Whittny の条件と呼ばれる。通常 $K-1$ 次のスプライン関数の接点は、簡単のために以下のように取ることが多い。

$$\begin{aligned} q_0 &= q_1 = \cdots = q_{K-1} = x_0 \\ q_{i+K} &= (x_i + x_{i+K})/2 \qquad (i = 0, 1, \ldots, N-1-K) \\ q_N &= q_{N+1} = \cdots = x_{N+K-1} = x_{N-1} \end{aligned} \tag{11}$$

接点が決定できたら、連立 N 次方程式

$$\sum_{i=0}^{N-1} \alpha_i B_{i,K}(x_j) = y_j (j = 0, 1, \ldots, N-1) \tag{12}$$

を解いて α_i を求めれば良い。

2.1 2 次 (K=3) のスプライン関数における実作業

前提として、$x_i(i = 0 \ldots N-1)$ は等間隔であり、各々に対する $y_i = f(x_i)$ は既知とする。
まず $q_i(i = 0 \ldots N+2)$ を以下のように定める。

$$\begin{aligned} q_0 &= q_1 = q_2 = x_0 \\ q_{i+3} &= (x_i + x_{i+3})/2 \qquad (i = 0 \ldots N-4) \\ q_N &= q_{N+1} = q_{N+2} = x_{N-1} \end{aligned} \tag{13}$$

$N \times N$ の行列 $\boldsymbol{B}$ を用意し、以下のように B_{ij} を決定する。

```
# 1st kind
for(j=0;j<N;j++){
    for(i=0;i<N;i++){
        if (x[j].ge.q[i] .and. x[j].le.q[i+1]){
            B[i][j] = 1
        }
        else{
            B[i][j] = 0
        }
    }
}

# 2nd kind
for(j=0;j<N;j++){
    for(i=0;i<N;i++){
        if (q[i+1].eq.q[i]){
            BP = B[i][j]
        }
        else{
            BP = (x[j]-q[i])/(q[i+1]-q[i])*B[i][j]
        }

        if (i+1.ge.N) {
            BL = 1
        }
        else{
            BL = B[i+1][j]
        }
        if (q[i+2].eq.q[i+1]){
            BL = BL
        }
        else{
            BL = (q[i+2]-x[j]/(q[i+2]-q[i+1])*BL
```

```
        }
        B[i][j] = BP + BL
    }
}

# 2nd kind
for(j=0;j<N;j++){
    for(i=0;i<N;i++){
        if (q[i+2].eq.q[i]){
            BP = B[i][j]
        }
        else{
            BP = (x[j]-q[i])/(q[i+2]-q[i])*B[i][j]
        }

        if (i+1.ge.N) {
            BL = 1
        }
        else{
            BL = B[i+1][j]
        }
        if (q[i+3].eq.q[i+1]){
            BL = BL
        }
        else{
            BL = (q[i+3]-x[j]/(q[i+3]-q[i+1])*BL
        }
        B[i][j] = BP + BL
    }
}
```

このようにして B_{ij} が求まったら、

$$\alpha_i = B_{ji}^{-1} y_j \tag{14}$$

によって係数ベクトル $\boldsymbol{\alpha}$ を求めれば良い。行列 B は疎 (sparse) な行列となるので、 Stanford 大学で開発された SPARSE ライブラリを用いるのがメモリの有効利用などの点で有利であろうと考えている（まだ実装はしていない)。

3　2 次元スプライン関数の導出

3.1　一般的な議論

モンテカルロシミュレーションでは、

- 散乱角が衝突のエネルギー E と衝突パラメータ b の関数となる。

- 衝突ガス粒子の速度、角度が乱数 p とスパッタ粒子の速度 v_g の関数となる。

これら二変数の適当な間隔ごとに関数値を求め、この $(E_i, b_j, \theta_{ij}) \quad (0 \leq i, j \leq N-1)$ や (p_i, v_{pi}, v_{gi}) を元にスプライン関数を構成することを考える。以降では変数は $(x, y, z(x, y))$ に置き換える。

まず x 方向の接点 q_i と y 方向の接点 r_j を式 11 に従って求める。スプライン関数 s(x,y) は

$$s(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \alpha_{ij} B_{i,K}(x) B_{j,K}(y) \tag{15}$$

と与えられるから、これを前節と同様の方法で解けば良い。

3.2 2 次のスプライン関数 $(K = 3)$ の実装

Berkley SPARSE ライブラリではなく、通常の行列解法を用いたバージョンを作成した。サブルーチン bd でマトリックス b を作り、これをベクトル z(nx*ny) に対して $ba = z$ のように解いて係数ベクトル a を求める。関数値は spf によって得る。

これを 2 変数関数

$$z = \cos(x) \sin(y) \tag{16}$$

に対して使ってみた結果を添付図に示す。与えたデータ点は x, y ともに 0.5 刻みとした。

```
      subroutine bd(mdim, m, ndimx, nx, ndimy, ny, ichix, ichiy,
     $     x, y, qx, qy, b, bx, by, bxj, byj)
c
c     create the matrix of N x N dimension for the determination of
c     parameters of spline function order ICHI using Deboor Cox argorism
c
c     assumption.. the data point is placed periodically
c     (i.e. x[j] = ja)
c
c     input
c       mdim  .. declared row dimension of B in calling program
c       m     .. order of the matrix (namely, nx * ny)
c       ndimx .. declared row demension for x
c       nx    .. number of data points along x axis
c       ndimy .. declared row demension for y
c       ny    .. number of data points along y axis
c       ichix .. order of spline function for x axis
c       ichiy ..                              y
c       x(n)  .. data points along x axis
c       y(n)  .. data points along y axis
c
c     output
c       qx, qy .. node data
c       b      .. created matrix to be solved
```

```
c
c     work
c       bx    .. B spline for X axis
c       by    ..              Y
c       bxj   .. B spline for X axis
c       byj   .. B spline for Y axis
c
      integer mdim, m, ndimx, nx, ndimy, ny, ichix, ichiy
      real b(mdim, m), x(nx), y(ny), qx(nx+ichix), qy(nx+ichiy)
      real bx(ndimx, nx), by(ndimy, ny)
      real bxj(nx), byj(ny)
===
      real function spf (nx, ny, ichix, ichiy, qx, qy, a, x, y)
c
c     calculate the value of 2D spline function at (x, y).
c
c     input
c       nx, ny        .. number of data points
c       ichix, ichiy  .. order of spline function
c       qx, qy        .. node point
c       a             .. parameter vector for spline function
c       x, y          .. input data
c
      integer nx, ny, ichix, ichiy
      real qx(nx+ichix), qy(ny+ichiy)
      real a(nx*ny), x, y
```

参考文献

[1] 桜井 明 (監), 菅野敬祐, 吉村和美, 高山文雄「C によるスプライン関数」東京電機大学出版局 (1993)