

# ラグランジュ補間型基底を用いた有限要素法の 行列要素の代数的構造について

村上 弘<sup>1,a)</sup>

**概要：**有限要素法（FEM）の近似に於いて、ラグランジュ補間型の基底関数を用いると、FEMで離散化された行列要素は一定の代数的な構造を持つ。そのことを利用すると行列要素を再現するために保持すべき記憶量が削減できることや、行列-ベクトル積の計算量を削減しうることを簡単な偏微分作用素の例について示す。

**キーワード：**有限要素法, ラグランジュ補間基底, 記憶削減, 高速アルゴリズム, Trummer の問題

## On algebraic structures of FEM matrix elements using Lagrange interpolants as the basis functions

HIROSHI MURAKAMI<sup>1,a)</sup>

**Abstract:** When Lagrange interpolants are used as the basis functions, matrix elements in FEM approximation have certain algebraic structures. By the use of the structures, for a simple PDE case as an example, we show it is possible to reduce the amount of storage to reproduce the matrix elements and also reduce the number of arithmetic operations to calculate the multiplication of the FEM matrix to a vector.

**Keywords:** FEM, Lagrange interpolants, reduction of memory, fast algorithm, Trummer's problem

### 1. はじめに

将来の HPC システムで大規模な問題を安価に解くためには、問題を解くのに必要な記憶の量や記憶の転送量をなるべく低減することが、性能のボトルネックを回避するために必要になると考えられている。その要件を満たしながら、高速で精度の良い計算が可能とするような解法の定式化、算法、計算技法の開発が今後重要になると思われる。

今回、簡単な線形の偏微分方程式（PDE）の境界値問題を FEM により高精度に解く例を設定し、それに対して必要記憶量を低減できると思われる計算技法の一つについて考察をした。PDE の境界値問題から FEM 離散化で生じる

連立 1 次方程式あるいは固有値問題は、有限要素ごとの行列にベクトルを乗じる操作を繰り返して反復法により解くものとする。

各有限要素ごとの FEM 離散化行列を構成する際に、未知関数の展開基底を Lagrange 補間の基底多項式（多次元ではそのテンソル積）にとるならば、Lagrange 補間公式が示している Lagrange 補間基底多項式の持つ単純な代数的性質が FEM 離散化行列に反映されることがわかる。すなわち（簡単な分数の恒等式によって）相異なる Lagrange 型基底関数 2 個の積は別の関数 2 個の和の形で表される。線形 2 階の微分方程式の各種演算子の Lagrange 型基底を用いた FEM による離散化行列は、要素内の基底の個数を  $N$  とするとき、一般には  $O(N^2)$  個の非零要素を持つ密行列となるが、Lagrange 型基底（多次元ではそのテンソル積）の性質から行列要素は  $O(N)$  個の量から容易に組み立

<sup>1</sup> 首都大学東京・数理情報科学専攻  
Department of Mathematics and Information Sciences,  
Tokyo Metropolitan University

<sup>a)</sup> mrkmhrsh@tmu.ac.jp

てられる構造を持つことが導ける。さらに、この行列要素が持つ特別な構造を利用すると、FEMの方程式を反復法により解く際に必要な、「離散化行列とベクトルとの乗算」の演算量のオーダーを下げる事ができる。

**注: FEMで扱う微分方程式の例題**

以下では簡単のために、FEMで離散化する偏微分方程式の作用素として、たとえば $U(\mathbf{x})$ を位置 $\mathbf{x}$ に依るポテンシャル関数として $(-\Delta + U(\mathbf{x}))$ を考える。また簡単のため、解析対象の領域は座標軸に平行な直方体状であるとする(図1, 図2)。直方体状の解析領域の各辺の長さを任意に分割し、分割で得られた直方体状の部分領域(有限要素)の内部では連続な未知関数を(直積型の)L-補間式で近似する。(曲線座標による曲要素への一般化も可能ではあるが、計算式は煩雑になる)。FEMで離散化した大規模連立方程式を反復法により解くものと仮定すると、行列をベクトルに乗じる処理は多くの回数繰り返す必要がある。

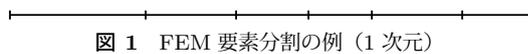


図1 FEM要素分割の例(1次元)

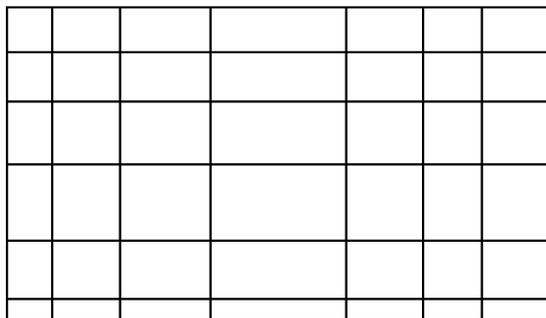


図2 FEM要素分割の例(2次元)

**2. 行列要素の構造**

(一次元)実区間 $I$ 内の相異なる $n$ 個の分点 $\alpha_j$ ,  $j = 1, 2, \dots, n$ 上での任意の関数 $f(x)$ に対するLagrangeの(多項式)補間公式は

$$\tilde{f}(x) = \sum_{j=1}^n \frac{P^{(\alpha)}(x)}{(x - \alpha_j)P^{(\alpha)'(\alpha_j)}} f(\alpha_j)$$

で、 $\tilde{f}(x)$ は一般には $(n - 1)$ 次多項式となる。ここで $P^{(\alpha)}(x) \equiv \prod_{\ell=1}^n (x - \alpha_\ell)$ は全分点を1次因子を持つ積である。この補間法は、例えば分点分布を等間隔にとると、よく知られているように、 $n$ を増やすにつれて数値不安定性が拡大するが、等間隔ではない「性質の良い分点分布」を採用すれば、分点数 $n$ を増すとき関数 $f(x)$ の補間近似 $\tilde{f}(x)$ による近似度は向上する。また通常は隣接するFEM要素の境界上での解の連続性の実現を容易にするために、区間 $I$ の両端を接点に含めておく(この場合には閉じた数値積分公式であるGauss-Lobatto則の分点を分点分布とし

て採用すると性質が良い。標準区間 $[-1, 1]$ に対するこの数値積分則の分点数 $n$ の分点は、 $(n - 2)$ 個の $(n - 1)$ 次Legendre多項式の導関数 $P'_{n-1}(x)$ の零点(すべて実数で $(-1, 1)$ にある)と2個の両端点 $\pm 1$ からなる。関数の線形展開に用いるLagrange基底は $n$ 個の相異なる $(n - 1)$ 次多項式である(図3)。また、 $n$ を基底の(あるいは展開の)位数と呼ぶ。

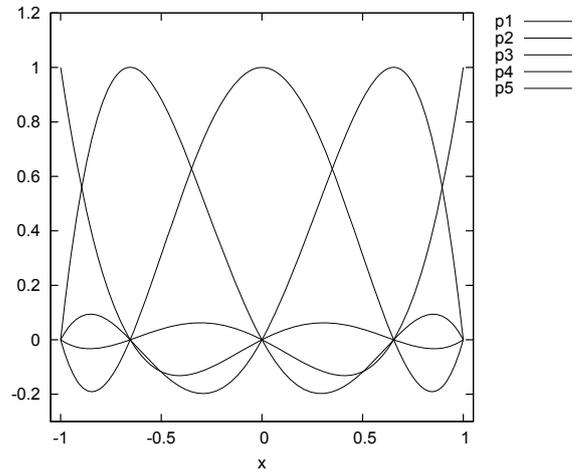


図3 L-基底多項式の例. 5点 Gauss-Lobatto 則の分点を採用

なお以下では式の簡略化のため、補間基底から規格化因子 $1/P^{(\alpha)'(\alpha_j)}$ を省略したLagrange基底を $\varphi_k^{(\alpha)}(x) \equiv P^{(\alpha)}(x)/(x - \alpha_k)$ ,  $k = 1, 2, \dots, n$ と定義する。本来のLagrange基底による行列要素が必要なときには、添字についてスケールを行なえばよい。

本報告で紹介する方法の本質は、簡易な代数的恒等式： $\frac{1}{x-a} \cdot \frac{1}{x-b} = \frac{1}{a-b} (\frac{1}{x-a} - \frac{1}{x-b})$ 、(ただし $a \neq b$ の場合)により、同一要素内の異なる添字を持つ2個のLagrange基底関数 $\varphi_i^{(\alpha)}(x)$ と $\varphi_j^{(\alpha)}(x)$  ( $i \neq j$ )の積が、 $\varphi_i^{(\alpha)}(x)\varphi_j^{(\alpha)}(x) = \frac{1}{\alpha_i - \alpha_j} (\Phi_i^{(\alpha)}(x) - \Phi_j^{(\alpha)}(x))$ 、(ここで $\Phi_k^{(\alpha)}(x) \equiv \{P^{(\alpha)}(x)\}^2 / (x - \alpha_k)$ である)となって別の関数2個の和の形で表されることである。

このLagrange型基底関数が持つ代数的な性質を利用することで、FEMによる離散化行列が行列の要素数と比べて少ない個数の量から構成されること、さらに行列を任意ベクトルに乗じる計算の演算量のオーダーが下げられることを以下で示す。

**注意**

同様に、Sinc-型の関数基底系 $\psi_k(x) \equiv \sin x / (x - \alpha_k)$ (ただし $\alpha_k = k\pi$ ,  $k = 0, \pm 1, \pm 2, \dots$ )を採用すると、これは通常多項式によるLagrange補間式の中に現れる多項式 $P^{(\alpha)}(x)$ を、分点を等間隔にして極限移行した関数 $\sin x$ に置き換えたもので一種のLagrange基底系とみなせる。この場合にも、2個の異なるSinc-型基底 $\psi_i(x)$ と $\psi_j(x)$ (ただし $i \neq j$ )の積が $\psi_i(x)\psi_j(x) = \frac{1}{\alpha_i - \alpha_j} (\Psi_i(x) - \Psi_j(x))$

(ここで  $\Psi_k(x) \equiv \sin^2 x / (x - \alpha_k)$  である) となり, 2 個の関数の和として書けることがわかる. なお, 領域の実効的大きさを制限するために遠方  $|x| > R$  において急減衰する窓関数  $W(x, R)$  (たとえば  $W(x, R) = \exp(-x^2/R^2)$  など) を導入して, 基底関数を  $\psi_k(x) \equiv W(x, R) \cdot \sin x / (x - \alpha_k)$  とする場合も同様になる. 大きな 1 個だけの有限要素を用いて遠方での振る舞いが自然な方程式を解析する用途には有効に利用できるのではないかとと思われる. ただし今回は興味深いこの基底関数系の拡張についてはこれ以上の追求をしない.

## 2.1 一次元の場合

### ポテンシャルの FEM 行列要素 (一次元)

ポテンシャル  $U(x)$  の基底関数  $\varphi_k^{(\alpha)}$  による要素行列は  $V_{i,j} \equiv \int_I \varphi_i^{(\alpha)}(x) U(x) \varphi_j^{(\alpha)}(x) dx = \int_I \frac{\{P^{(\alpha)}(x)\}^2 U(x)}{(x - \alpha_i)(x - \alpha_j)} dx$ , ただし  $x$  は要素内の座標である. そのとき

$$L_i^{(e)} = \int_I \frac{\{P^{(\alpha)}(x)\}^2 U(x)}{(x - \alpha_i)^{1+e}} dx, \quad e = 0, 1$$

で定義された量を用いると,

$$V_{i,j} = \begin{cases} \frac{1}{\alpha_i - \alpha_j} (L_i^{(0)} - L_j^{(0)}) & , \text{if } i \neq j \\ L_i^{(1)} & , \text{if } i = j \end{cases}$$

と書ける. このことからポテンシャルの FEM 行列要素である  $n(n+1)/2$  個の量  $V_{i,j}$  は  $L_k^{(0)}$  と  $L_k^{(1)}$  の合計  $2n$  個の量から構成できることがわかる.

$L^{(0)}$  に対する被積分関数は  $V_{i,j}$  の元の定義の被積分関数よりも多項式の次数が 1 だけ高いので,  $L^{(0)}$  の値の精度を保つために数値積分には少しだけ高次の公式を用いる必要があるように見えるが, 最終的に計算される値は  $V_{i,j}$  なので,  $L^{(0)}$  の積分を行う際には  $V_{i,j}$  の元の式の積分に必要なだけの精度の公式を用いれば充分である.

### ラプラシアン of FEM 行列要素 (一次元)

まず, ラプラシアン of FEM 行列要素は用いる基底だけに依るので, 領域が複数の有限要素に分割されている場合, 要素内の局所座標を標準区間  $[-1, 1]$  に 1 次変換したときに同一の L-型基底関数の組が割りあてられた有限要素に対するラプラシアンの行列は一致する. よって, "例えば" Gauss-Lobatto 則の分点を用いた L-型基底関数を採用している場合には, 有限要素内の基底の位数  $n$  の種類が比較的少なければ, その種類分の標準区間の有限要素に対応するラプラシアン of FEM 行列だけを保持して要素の実際の寸法に合わせて値をスケールして用いれば記憶量の削減ができる.

分点数  $n$  が大きい有限要素に対しては以下のようにすると, その有限要素に対するラプラシアン行列の作用を実現するための記憶量を削減できる. 一次元での (反対符号の) ラプラシアンの行列要素  $K_{i,j}$  は,  $x$

を要素内座標として  $K_{i,j} = \int_I (\frac{d}{dx} \varphi_i^{(\alpha)} \cdot \frac{d}{dx} \varphi_j^{(\alpha)}) dx = \int_I \{ \frac{d}{dx} (\frac{P^{(\alpha)}(x)}{x - \alpha_i}) \cdot \frac{d}{dx} (\frac{P^{(\alpha)}(x)}{x - \alpha_j}) \} dx$  と書ける. そのとき  $A_k^{(3)} \equiv -2 \int_I \frac{\{P^{(\alpha)}(x)\}^2}{x - \alpha_k} dx = -2 \int_I P^{(\alpha)}(x) \varphi_k^{(\alpha)}(x) dx$ ,  $A_k^{(2)} \equiv \int_I (\frac{P^{(\alpha)}(x)}{x - \alpha_k})^2 dx = \int_I \{\varphi_k^{(\alpha)}(x)\}^2 dx$ ,  $A_k^{(1)} \equiv \int_I \frac{d}{dx} P^{(\alpha)}(x) \cdot \frac{d}{dx} (\frac{P^{(\alpha)}(x)}{x - \alpha_k}) dx = \int_I \{ \frac{d}{dx} P^{(\alpha)}(x) \cdot \frac{d}{dx} \varphi_k^{(\alpha)}(x) \} dx$ ,  $B_k \equiv \int_I \{ \frac{d}{dx} (\frac{P^{(\alpha)}(x)}{x - \alpha_k}) \}^2 dx = \int_I \{ \frac{d}{dx} \varphi_k^{(\alpha)}(x) \}^2 dx$ , で定義される定数  $A_k^{(3)}$ ,  $A_k^{(2)}$ ,  $A_k^{(1)}$ ,  $B_k$  を用いると, ラプラシアンの行列要素はそれらの組み合わせとして

$$K_{i,j} = \begin{cases} (\frac{1}{\alpha_i - \alpha_j})^3 (A_i^{(3)} - A_j^{(3)}) + (\frac{1}{\alpha_i - \alpha_j})^2 (A_i^{(2)} + A_j^{(2)}) \\ \quad + \frac{1}{\alpha_i - \alpha_j} (A_i^{(1)} - A_j^{(1)}), \text{ if } i \neq j, \\ B_i, \text{ if } i = j. \end{cases}$$

となる. それゆえ, 行列  $K$  の  $n(n+1)/2$  個の要素の値は,  $A_k^{(0)}$ ,  $A_k^{(1)}$ ,  $A_k^{(2)}$ ,  $B_k$ ,  $k=1, 2, \dots, n$  の合計  $4n$  個の量だけから構成できる. いま量  $C_{i,j}^{(\alpha)}$  を  $C_{i,j}^{(\alpha)} = \begin{cases} \frac{1}{\alpha_i - \alpha_j} & , \text{if } i \neq j \\ 0 & , \text{if } i = j \end{cases}$  とすると,  $K_{i,j} = (C_{i,j}^{(\alpha)})^3 (A_i^{(3)} - A_j^{(3)}) + (C_{i,j}^{(\alpha)})^2 (A_i^{(2)} + A_j^{(2)}) + C_{i,j}^{(\alpha)} (A_i^{(1)} - A_j^{(1)}) + \delta_{i,j} B_i$  であるから, 任意ベクトル  $\mathbf{u}$  に行列  $K$  を乗じた結果  $\mathbf{w}$  を計算するには,  $C_{i,j}^{(\alpha)}$  を格納せずに  $w_i = \sum_j K_{i,j} u_j = A_i^{(3)} \sum_j (C_{i,j}^{(\alpha)})^3 u_j - \sum_j (C_{i,j}^{(\alpha)})^3 (A_j^{(3)} u_j) + A_i^{(2)} \sum_j (C_{i,j}^{(\alpha)})^2 u_j + \sum_j (C_{i,j}^{(\alpha)})^2 (A_j^{(2)} u_j) + A_i^{(1)} \sum_j C_{i,j}^{(\alpha)} u_j - \sum_j C_{i,j}^{(\alpha)} (A_j^{(1)} u_j) + B_i u_i$  を計算すればよい. いま  $p_j^{(3)} = A_j^{(3)} u_j$ ,  $p_j^{(2)} = A_j^{(2)} u_j$ ,  $p_j^{(1)} = A_j^{(1)} u_j$ , さらに  $q_i^{(3)} = \sum_j (C_{i,j}^{(\alpha)})^3 u_j$ ,  $q_i^{(2)} = \sum_j (C_{i,j}^{(\alpha)})^2 u_j$ ,  $q_i^{(1)} = \sum_j C_{i,j}^{(\alpha)} u_j$  と置くと,  $w_i = A_i^{(3)} q_i^{(3)} + A_i^{(2)} q_i^{(2)} + A_i^{(1)} q_i^{(1)} - \sum_j \{ (C_{i,j}^{(\alpha)})^3 p_j^{(3)} - (C_{i,j}^{(\alpha)})^2 p_j^{(2)} + C_{i,j}^{(\alpha)} p_j^{(1)} \} + B_i u_i$  として効率良く計算できる. いまの一次元の場合には, この計算方法による演算量は  $O(n^2)$  なのであらかじめ陽に形成し保持しておいた行列  $K$  をベクトルへ乗じるとオーダーは同じであるが,  $n$  が非常に大きい場合に行列  $K$  自身を格納や保持しないので必要な記憶量を小さく抑えて計算ができる.

## 2.2 二次元の場合

### テンソル積型基底 (二次元)

要素が局所座標の直積でパラメタ付けされるなら, 基底を各座標の Lagrange 基底のテンソル積  $\varphi_{k_1, k_2}^{(\alpha, \beta)}(x, y) = \varphi_{k_1}^{(\alpha)}(x) \varphi_{k_2}^{(\beta)}(y)$  とする. ただし  $\varphi_{k_1}^{(\alpha)}(x) = P^{(\alpha)}(x) / (x - \alpha_{k_1})$ ,  $\varphi_{k_2}^{(\beta)}(y) = P^{(\beta)}(y) / (y - \beta_{k_2})$ , そうして  $P^{(\alpha)}(x) = \prod_{\ell=1}^{n_1} (x - \alpha_\ell)$ ,  $P^{(\beta)}(y) = \prod_{\ell=1}^{n_2} (y - \beta_\ell)$  である.

### ポテンシャルの FEM 行列要素 (二次元)

二次元のポテンシャル  $U(x, y)$  の FEM 行列要素は,  $V_{i,j} = V_{i_1, i_2; j_1, j_2} = \int \int_{area} \varphi_{i_1}^{(\alpha)}(x) \varphi_{i_2}^{(\beta)}(y) U(x, y) \varphi_{j_1}^{(\alpha)}(x) \varphi_{j_2}^{(\beta)}(y) dx dy = \int \int_{area} \frac{\{P^{(\alpha)}(x)\}^2 \{P^{(\beta)}(y)\}^2 U(x, y)}{(x - \alpha_{i_1})(x - \alpha_{j_1})(y - \beta_{i_2})(y - \beta_{j_2})} dx dy$  であるが, 定義  $L_{i,j}^{(e_1, e_2)} = \int \int_{area} \frac{\{P^{(\alpha)}(x)\}^{2+e_1} \cdot \{P^{(\beta)}(y)\}^{2+e_2}}{(x - \alpha_i)^{1+e_1} (y - \beta_j)^{1+e_2}} U(x, y) dx dy$  を用いれば  $V_{i_1, i_2; j_1, j_2}$  は  $L$  の組み合わせにより以下のように

書ける.

$$= \begin{cases} \frac{1}{\alpha_{i_1-\alpha_{j_1}} \cdot \frac{1}{\beta_{i_2-\beta_{j_2}}} (L_{i_1,i_2}^{(0,0)} - L_{j_1,i_2}^{(0,0)} - L_{i_1,j_2}^{(0,0)} + L_{j_1,j_2}^{(0,0)}) \\ \quad , \text{if } i_1 \neq j_1, i_2 \neq j_2, \\ \\ \frac{1}{\beta_{i_2-\beta_{j_2}}} (L_{p,i_2}^{(1,0)} - L_{p,j_2}^{(1,0)}) , \text{if } i_1=j_1=p, i_2 \neq j_2, \\ \\ \frac{1}{\alpha_{i_1-\alpha_{j_1}}} (L_{i_1,q}^{(0,1)} - L_{j_1,q}^{(0,1)}) , \text{if } i_1 \neq j_1, i_2=j_2=q, \\ \\ L_{p,q}^{(1,1)} , \text{if } i_1=j_1=p, i_2=j_2=q. \end{cases}$$

上の代数的関係式を用いると、約  $n^4/2$  個の値  $V_{i_1,i_2;j_1,j_2}$  は全部で  $(2n)^2$  個ある  $L_{i,j}^{(e_1,e_2)}$  の値だけから組み立てられる。行列要素が持つこの構造を用いれば、必要な記憶量を減らすことができる。ポテンシャルの行列要素の場合は、この代数的構造は簡単な分数式の恒等式と基底が Lagrange 補間の基底関数のテンソル積であることだけから導かれるものである。

### ラプラシアン の FEM 行列要素 (二次元)

二次元以上の場合の微分作用素の Galerkin 離散化については、特に偏微分作用素が定数係数の場合には簡単で高速になる。いま曲線座標系を用いていないので、各次元方向の行列  $M_{i_1,j_1}^{(\alpha)} = \int_{\delta x} \varphi_{i_1}^{(\alpha)}(x) \varphi_{j_1}^{(\alpha)}(x) dx$  と  $K_{i_1,j_1}^{(\alpha)} = \int_{\delta x} \{ \frac{d}{dx} \varphi_{i_1}^{(\alpha)}(x) \cdot \frac{d}{dx} \varphi_{j_1}^{(\alpha)}(x) \} dx$ 、および  $M_{i_2,j_2}^{(\beta)} = \int_{\delta y} \varphi_{i_2}^{(\beta)}(y) \varphi_{j_2}^{(\beta)}(y) dy$  と  $K_{i_2,j_2}^{(\beta)} = \int_{\delta y} \{ \frac{d}{dy} \varphi_{i_2}^{(\beta)}(y) \cdot \frac{d}{dy} \varphi_{j_2}^{(\beta)}(y) \} dy$  をあらかじめ作成しておく、2次元の(符号が逆の)ラプラシアン  $-\Delta = -(\nabla_x^2 + \nabla_y^2)$  を FEM で離散化した行列は  $K_{i,j} = K_{i_1,i_2;j_1,j_2} = K_{i_1,j_1}^{(\alpha)} \cdot M_{i_2,j_2}^{(\beta)} + M_{i_1,j_1}^{(\alpha)} \cdot K_{i_2,j_2}^{(\beta)}$  と表せる。この構造を利用すると行列  $K$  と任意のベクトルの積は高速に計算できる。 $x$  軸方向の行列  $M^{(\alpha)}$  と  $K^{(\alpha)}$  とがそれぞれとりうる種類は有限要素の  $x$  方向の寸法の違いを正規化する座標の線形 1 次変換による因子を除けば有限要素の  $x$  軸方向の基底関数の種類だけしかなく、 $y$  軸方向の行列  $M^{(\beta)}$  と  $K^{(\beta)}$  がそれぞれとりうる種類もまた有限要素の  $y$  方向の寸法の違いを正規化する座標の線形 1 次変換による因子を除けば有限要素の  $y$  軸方向の基底関数の種類だけしかない、あられる種類がごく少数なら、それらを正規化された座標での行列要素としてあらかじめ計算して保持しておいても記憶量的には大きな負担にはならない。

### 2.3 三次元の場合

#### テンソル積型の基底 (三次元)

二次元の場合と同様に、基底関数を各局所座標の Lagrange 基底のテンソル積  $\varphi_{k_1,k_2,k_3}^{(\alpha,\beta,\gamma)}(x,y,z) = \varphi_{k_1}^{(\alpha)}(x) \varphi_{k_2}^{(\beta)}(y) \varphi_{k_3}^{(\gamma)}(z)$  とする。ただし  $\varphi_{k_1}^{(\alpha)}(x) = P^{(\alpha)}(x)/(x - \alpha_{k_1})$ ,  $\varphi_{k_2}^{(\beta)}(y) = P^{(\beta)}(y)/(y - \beta_{k_2})$ ,  $\varphi_{k_3}^{(\gamma)}(z) = P^{(\gamma)}(z)/(z - \gamma_{k_3})$  で、 $P^{(\alpha)}(x) = \prod_{\ell=1}^{n_1} (x - \alpha_\ell)$ ,

$P^{(\beta)}(y) = \prod_{\ell=1}^{n_2} (y - \beta_\ell)$ ,  $P^{(\gamma)}(z) = \prod_{\ell=1}^{n_3} (z - \gamma_\ell)$  である。

#### ポテンシャルの FEM 行列要素 (三次元)

三次元のポテンシャル  $U(x,y,z)$  の FEM 離散化行列要素は  $V_{i,j} = V_{i_1,i_2,i_3;j_1,j_2,j_3} = \int \int \int_{vol} \varphi_{i_1}^{(\alpha)}(x) \varphi_{i_2}^{(\beta)}(y) \varphi_{i_3}^{(\gamma)}(z) U(x,y,z) \varphi_{j_1}^{(\alpha)}(x) \varphi_{j_2}^{(\beta)}(y) \varphi_{j_3}^{(\gamma)}(z) dx dy dz$   
 $= \int \int \int_{vol} \frac{\{P^{(\alpha)}(x)\}^2 \{P^{(\beta)}(y)\}^2 \{P^{(\gamma)}(z)\}^2 U(x,y,z)}{(x-\alpha_{i_1})(x-\alpha_{j_1})(y-\beta_{i_2})(y-\beta_{j_2})(z-\gamma_{i_3})(z-\gamma_{j_3})} dx dy dz$  となる。

いま簡約された行列  $L$  を  $L_{i,j,k}^{(e_1,e_2,e_3)}$   
 $= \int \int \int_{vol} \frac{\{P^{(\alpha)}(x)\}^2 \{P^{(\beta)}(y)\}^2 \{P^{(\gamma)}(z)\}^2 U(x,y,z)}{(x-\alpha_i)^{1+e_1} (y-\beta_j)^{1+e_2} (z-\gamma_k)^{1+e_3}} dx dy dz$  と定義すると、ポテンシャルの行列要素  $V_{i_1,i_2,i_3;j_1,j_2,j_3}$  は  $L$  の組合せで以下のように表せる。

$$= \begin{cases} \frac{1}{\alpha_{i_1-\alpha_{j_1}} \cdot \frac{1}{\beta_{i_2-\beta_{j_2}}} \cdot \frac{1}{\gamma_{i_3-\gamma_{j_3}}} (L_{i_1,i_2,i_3}^{(0,0,0)} - L_{j_1,i_2,i_3}^{(0,0,0)} - L_{i_1,j_2,i_3}^{(0,0,0)} - L_{i_1,i_2,j_3}^{(0,0,0)} + L_{j_1,i_2,j_3}^{(0,0,0)} + L_{j_1,i_2,i_3}^{(0,0,0)} + L_{i_1,j_2,i_3}^{(0,0,0)} - L_{j_1,j_2,i_3}^{(0,0,0)}) \\ \quad , \text{if } i_1 \neq j_1, i_2 \neq j_2, i_3 \neq j_3, \\ \\ \frac{1}{\beta_{i_2-\beta_{j_2}} \cdot \frac{1}{\gamma_{i_3-\gamma_{j_3}}} (L_{p,i_2,i_3}^{(1,0,0)} - L_{p,i_2,j_3}^{(1,0,0)} - L_{p,j_2,i_3}^{(1,0,0)} + L_{p,j_2,j_3}^{(1,0,0)}) \\ \quad , \text{if } i_1=j_1=p, i_2 \neq j_2, i_3 \neq j_3, \\ \\ \frac{1}{\alpha_{i_1-\alpha_{j_1}} \cdot \frac{1}{\gamma_{i_3-\gamma_{j_3}}} (L_{i_1,q,i_3}^{(0,1,0)} - L_{i_1,q,j_3}^{(0,1,0)} - L_{j_1,q,i_3}^{(0,1,0)} + L_{j_1,q,j_3}^{(0,1,0)}) \\ \quad , \text{if } i_1 \neq j_1, i_2=j_2=q, i_3 \neq j_3, \\ \\ \frac{1}{\alpha_{i_1-\alpha_{j_1}} \cdot \frac{1}{\beta_{i_2-\beta_{j_2}}} (L_{i_1,i_2,r}^{(0,0,1)} - L_{j_1,i_2,r}^{(0,0,1)} - L_{i_1,j_2,r}^{(0,0,1)} + L_{j_1,j_2,r}^{(0,0,1)}) \\ \quad , \text{if } i_1 \neq j_1, i_2 \neq j_2, i_3=j_3=r, \\ \\ \frac{1}{\alpha_{i_1-\alpha_{j_1}}} (L_{i_1,q,r}^{(0,1,1)} - L_{j_1,q,r}^{(0,1,1)}) \\ \quad , \text{if } i_1 \neq j_1, i_2=j_2=q, i_3=j_3=r, \\ \\ \frac{1}{\beta_{i_2-\beta_{j_2}}} (L_{p,i_2,r}^{(1,0,1)} - L_{p,j_2,r}^{(1,0,1)}) \\ \quad , \text{if } i_1=j_1=p, i_2 \neq j_2, i_3=j_3=r, \\ \\ \frac{1}{\gamma_{i_3-\gamma_{j_3}}} (L_{p,q,i_3}^{(1,1,0)} - L_{p,q,j_3}^{(1,1,0)}) \\ \quad , \text{if } i_1=j_1=p, i_2=j_2=q, i_3 \neq j_3, \\ \\ L_{p,q,r}^{(1,1,1)} \quad , \text{if } i_1=j_1=p, i_2=j_2=q, i_3=j_3=r. \end{cases}$$

上記の代数的関係により、約  $n^6/2$  個の量  $V_{i_1,i_2,i_3;j_1,j_2,j_3}$  は合計  $(2n)^3$  個の量  $L_{i,j,k}^{(e_1,e_2,e_3)}$  から構成できる。

#### ラプラシアン の FEM 行列 (三次元)

曲線座標系を用いていないので、2次元の場合と同様に各次元方向の行列  $M_{i_1,j_1}^{(\alpha)}$  と  $K_{i_1,j_1}^{(\alpha)}$ 、および  $M_{i_2,j_2}^{(\beta)}$  と  $K_{i_2,j_2}^{(\beta)}$ 、および  $M_{i_3,j_3}^{(\gamma)}$  と  $K_{i_3,j_3}^{(\gamma)}$  を作成しておけば、ラプラシアン  $-\Delta = -(\nabla_x^2 + \nabla_y^2 + \nabla_z^2)$  のテンソル積型基底に対する FEM 離散化行列は

$$K_{i,j} = K_{i_1,i_2,i_3;j_1,j_2,j_3} = K_{i_1,j_1}^{(\alpha)} \cdot M_{i_2,j_2}^{(\beta)} \cdot M_{i_3,j_3}^{(\gamma)} + M_{i_1,j_1}^{(\alpha)} \cdot K_{i_2,j_2}^{(\beta)} \cdot M_{i_3,j_3}^{(\gamma)} + M_{i_1,j_1}^{(\alpha)} \cdot M_{i_2,j_2}^{(\beta)} \cdot K_{i_3,j_3}^{(\gamma)}$$

と表せて、この構造を用いて有限要素内の行列  $K$  とベクトルの積が高速に計算できる。いま  $x$  軸方向に対応する行列  $M^{(\alpha)}$  と  $K^{(\alpha)}$ 、および  $y$  軸方向に対応する行列  $M^{(\beta)}$  と  $K^{(\beta)}$ 、および  $z$  軸方向に対応する行列  $M^{(\gamma)}$  と  $K^{(\gamma)}$  は、それぞれすべての有限要素に設定した各軸方向の L-型基底の種類が比較的少数ならば、(要素の寸法の違いからくる各座標の線形 1 次変換による因子を考慮して) 共用できる。

### 3. 作用素行列とベクトルの高速な乗算法

前節では FEM の行列要素が持つ構造を示して、簡約された行列要素を導入した。行列要素が持つ構造の知識は、必要な数値積分の回数を節約しながら全ての行列要素を生成するのに利用できる。被積分関数の振舞が悪く積分を精度高く計算するための分点数を多く必要とする場合には極めて有効であろう。

あるいは記憶の圧縮法としても使える。たとえば高次の FEM 要素を多く用いて全系の行列方程式を反復法で解く場合に、行列ベクトル積を多数回計算するが、その際に本方法を用いて FEM 行列要素を簡約された形式で保持すれば、本来の定義による FEM 行列要素を直接保持する場合に比べて、記憶の転送量が低減できる。さらに、簡約された行列要素を保持するならば、与えられたベクトルと FEM 行列との積を速く計算することもできる。

#### Trummer の問題

分点の組  $\{\alpha\}$  に対応する反対称行列  $C^{(\alpha)}$  を  $C_{i,j}^{(\alpha)} \equiv \begin{cases} 1/(\alpha_i - \alpha_j) & , \text{if } i \neq j \\ 0 & , \text{if } i = j \end{cases}$  と定義する。  $n \times n$  行列  $C^{(\alpha)}$  の要素は  $n$  個の分点の値の組から必要に応じて簡単に計算できるので、行列  $C^{(\alpha)}$  は生成も保持もしない。

任意に与えられたベクトルに  $C^{(\alpha)}$  の形の行列を乗じた結果を高速に求める問題は「Trummer の問題」と呼ばれ、演算量  $O(n \log^2 n)$  の高速アルゴリズムが既に知られている [5] (さらに分点集合  $\{\alpha\}$  の分布を Chebyshev 零点のように特殊に選べれば演算量は  $O(n \log n)$  になる)。これは理論的には大変興味深い、「高速アルゴリズム」を適用したときの数値的な性質が悪くないかなどについての検証が必要である。そのため、以下では主として通常の古典的な行列ベクトル積のみを考察するが、高速アルゴリズムを使用する場合の計算量のオーダーも  $C^{(\alpha)}$  の形の行列による添字の縮約に着目すれば容易にわかる。

#### 3.1 一次元の場合

例としてポテンシャルの行列  $V$  と与えられたベクトル  $u$  との積の計算を示す。定義に従って行列要素  $V_{i,j}$  の式はクロネッカのデルタ  $\delta$  を用いて  $V_{i,j} = C_{i,j}^{(\alpha)} (L_i^{(0)} - L_j^{(0)}) + \delta_{i,j} L_i^{(1)}$  と書けて、  $C_{k,k}^{(\alpha)} = 0$  を用いると、行列  $V$  とベクトル  $u$  の積は次のように計算できる。  $\sum_j V_{i,j} u_j =$

$\sum_{j(\neq i)} V_{i,j} u_j + V_{i,i} u_i = \sum_j C_{i,j}^{(\alpha)} (L_i^{(0)} - L_j^{(0)}) u_j + L_i^{(1)} u_i = (\sum_j C_{i,j}^{(\alpha)} L_i^{(0)} u_j) - (\sum_j C_{i,j}^{(\alpha)} L_j^{(0)} u_j) + L_i^{(1)} u_i$ . この第 1 項目は  $\sum_j C_{i,j}^{(\alpha)} L_i^{(0)} u_j = L_i^{(0)} (\sum_j C_{i,j}^{(\alpha)} u_j) = L_i^{(0)} a_i$  と計算すれば  $O(n^2)$  回の演算でできる。第 2 項目は  $\sum_j C_{i,j}^{(\alpha)} (L_j^{(0)} u_j) = \sum_j C_{i,j}^{(\alpha)} b_j$  とすれば  $O(n^2)$  回の演算でできる。第 3 項目は  $O(n)$  回の演算で計算できる。結局全部で  $O(n^2)$  回の演算で積を求めることができる。この一次元の場合の計算量のオーダーは、あらかじめ生成しておいた行列をベクトルに乗じた場合と同じである (理論的には、Trummer の問題の高速アルゴリズムを採用すれば、全体の計算量が減らせて  $O(n^2)$  を  $O(n \log^2 n)$  に下げられる)。  $C_{i,j}^{(\alpha)}$  の値を格納しないで必要に応じて計算すると、上記の計算過程の作業記憶量は  $O(n)$  になる。

#### 3.2 二次元の場合

二次元ポテンシャル  $V_{i,j}$  の行列の表式は、  $C^{(\alpha)}$  や  $C^{(\beta)}$  の対角が零であることを用いて、  $V_{i,j} = C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} (L_{i_1,i_2}^{(0,0)} - L_{j_1,j_2}^{(0,0)} - L_{i_1,j_2}^{(0,0)} + L_{j_1,i_2}^{(0,0)}) + \delta_{i_1,j_1} C_{i_2,j_2}^{(\beta)} (L_{i_1,i_2}^{(1,0)} - L_{j_1,j_2}^{(1,0)}) + C_{i_1,j_1}^{(\alpha)} \delta_{i_2,j_2} (L_{i_1,i_2}^{(0,1)} - L_{j_1,j_2}^{(0,1)}) + \delta_{i_1,j_1} \delta_{i_2,j_2} L_{i_1,i_2}^{(1,1)}$  となる。これからポテンシャルの行列  $V_{i,j}$  と与えられたベクトル  $u_j$  との積の具体的な計算過程が容易に書き下せて、(有限要素内の各座標軸方向の基底の個数が  $O(n)$  とすると) 現れる項の計算量がどれも  $O(n^3)$  であることが示せるので、すべての項の計算を合計した演算量も  $O(n^3)$  である (これも Trummer 問題に対する高速アルゴリズムを用いるならば、計算量は  $O(n^2 \log^2 n)$  とできる)。あらかじめ生成しておいた行列  $V$  をベクトル  $u$  に乗じる普通の方法による演算量は  $O(n^4)$  になるので、それに比べると計算量のオーダーが減ることがわかる。

ラプラシアン of 離散化行列  $K_{i,j} = K_{i_1,i_2;j_1,j_2}$  をベクトル  $u_j = u_{j_1,j_2}$  に乗じる計算の場合には、行列要素が持つ構造  $K_{i_1,i_2;j_1,j_2} = K_{i_1,j_1}^{(\alpha)} M_{i_2,j_2}^{(\beta)} + M_{i_1,j_1}^{(\alpha)} K_{i_2,j_2}^{(\beta)}$  を利用すれば、(有限要素内の各座標軸方向の基底の個数が  $O(n)$  とすれば) 次数  $O(n)$  の行列同士の転置を伴う積と和に帰着できて、次数  $O(n)$  の行列乗算に古典的な方法を用いて全部で演算量  $O(n^3)$  で計算ができる。

#### 3.3 三次元の場合

同様に三次元の場合も、ポテンシャルの離散化行列  $V_{i,j}$  と任意ベクトル  $u$  の積の計算式で出現する項の計算量は (有限要素内の各軸方向の基底の個数がどれも  $O(n)$  であるとき) すべて  $O(n^4)$  であることが示せる (この場合も Trummer 問題に対する高速アルゴリズムを用いるなら、演算量を  $O(n^3 \log^2 n)$  に下げられる)。あらかじめ生成しておいた行列  $V$  をベクトル  $u$  と乗じる普通の計算方法による演算量は  $O(n^6)$  なので、それに比べて計算量のオーダーが減ることがわかる。

ラプラシアン の 離散化行列  $K_{i,j} = K_{i_1,i_2,i_3;j_1,j_2,j_3}$  をベクトル  $u_j = u_{j_1,j_2,j_3}$  に乗じる計算の場合には、行列要素が持つ構造  $K_{i_1,i_2,i_3;j_1,j_2,j_3} = K_{i_1,j_1}^{(\alpha)} M_{i_2,j_2}^{(\beta)} M_{i_3,j_3}^{(\gamma)} + M_{i_1,j_1}^{(\alpha)} K_{i_2,j_2}^{(\beta)} M_{i_3,j_3}^{(\gamma)} + M_{i_1,j_1}^{(\alpha)} M_{i_2,j_2}^{(\beta)} K_{i_3,j_3}^{(\gamma)}$  を用いれば、(有限要素内の各座標軸方向の基底の個数が  $O(n)$  とすれば) やはり次数  $O(n)$  の行列と  $O(n) \times O(n^2)$  の行列の転置を伴う積と和に帰着できて、次数  $O(n)$  の行列乗算に古典的な方法を用いると全部で演算量  $O(n^4)$  で計算ができる。

#### 4. 隣接する有限要素間での解の連続性条件

詳しい説明は省略するが、隣接する有限要素間の境界上で近似関数の連続性条件を満たすには、共有されている有限要素の頂点、辺内、あるいは面内で、同じ値あるいは多項式として表現していることを保証するため、各有限要素内に台を持ち境界上で値を持つ基底関数に対する係数(ベクトルの要素)は、隣接する有限要素の同様の基底関数に対する係数との間で一種の拘束(張り合わせ)条件を満たす必要がある(今の場合、基底関数は各座標の Lagrange 補間多項式の直積であり、Lagrange 補間多項式のうち有限要素の上端と下端で値が 0 でないものはそれぞれ 1 個ずつとしている)。

一次元問題の場合は、隣接する有限要素の頂点での値が一致することだけが必要である。

二次元問題の場合には、ある頂点を共有するすべての有限要素(最大 4 個)はその頂点での値に対応する展開係数が一致すること。およびある辺を共有する二つの有限要素の間でその辺上での展開基底の位数が異なる場合には、位数が低い側の有限要素での辺上での展開係数を位数が高い側の辺上での有限要素の展開係数として移すために、高位側の補間点に於ける低位側の近似多項式の値を評価して求める。

三次元の場合にはより複雑で、ある頂点を共有するすべての有限要素はその頂点での値に対応する展開係数が一致すること。さらにある辺を共有する有限要素の中でその辺上での展開位数が最も低い有限要素に対応する展開係数から他の有限要素の同じ辺上での補間点での値を多項式の評価計算で求める。さらにまたある面を共有する二つの有限要素  $E^{(a)}$ ,  $E^{(b)}$  のその面内(辺を含まず)での展開係数は、2つの軸方向、それを第 1 軸方向と第 2 軸方向とするときに、その各軸方向への展開位数がそれぞれ  $n_1^{(a)}, n_2^{(a)}$  と  $n_1^{(b)}, n_2^{(b)}$  であるならば、 $n_1 = \min(n_1^{(a)}, n_1^{(b)})$ ,  $n_2 = \min(n_2^{(a)}, n_2^{(b)})$  とするとき、両側の有限要素の各面内に値を持つ基底に対する展開係数は、位数  $n_1$  と  $n_2$  の基底のテンソル積型基底で表される近似多項式の値をそれぞれの補間点に於いて評価して求める。

#### 5. おわりに

FEM の高次多項式基底として Lagrange 型基底関数を用いると、基底関数の積が別の関数の和の形に書けることから、微分方程式の作用素を FEM で離散化した行列の要素がある代数的な構造を持つことが(微分作用素が  $\Delta + U(\mathbf{x})$  のような場合には簡単に)示せて、行列要素の値は少ない個数の中間の値から組み立てることができる。このため(一般には数値積分を用いて求める)積分で定義される行列要素は、そのすべての行列要素を求める代わりに、少ない個数の中間の値だけを求めることで済ませることができる。そしてさらに FEM により離散化で生じた行列とベクトルの積の計算でも、必要記憶量と演算量が低減できる。本文中では主にポテンシャルの離散化を例にとり説明した。

数値的な性質に関しては、高次の展開基底でも Lagrange 補間を用いたことによる近似の数値不安定性はたとえば基底として Gauss-Lobatto 則の分点分布を用いると生じないであろうと思われる。また本技法の原理である(記憶の圧縮や計算量の低減のために)「積を和(差)に置き換える」ことにより、差の計算での相殺から生じる桁落ちで丸め誤差の拡大を起こす可能性があるようにも思われるが、それも分点分布の適切性と、必要ならば演算に用いる有効桁数を多少増やせば回避できるであろうと思われる。

想定されている電子計算機の演算性能と記憶容量や記憶転送の能力のバランスの今後一層の悪化を考慮すれば、このような記憶量を削減して計算を行う方法を考察することには意義があると考えられる。

数値実験を実際にいくつかの例について行ない有効性を調べてみることは、今後の課題である。

#### 参考文献

- [1] Aho, A.V., Hopcroft, J.E. and Ullman, J.D.: *The Design and Analysis of Computer Algorithms*, Addison-Wesley, (1974).
- [2] Axelsson, O. and Baker, V.A.: *Finite Element Solution of Boundary Value Problems*, Academic Press, 1984.
- [3] Babuska, I., Katz, I.N., Szabo, B.A.: "The P Version of the Finite Element Method", *SIAM J. Numer. Anal.*, **v.18** (1981), pp.515-545.
- [4] Chan, T.F., Glowinski, R., Periaux, J. and Widlund, O.B. Eds.: *Domain Decomposition Methods*, SIAM, Philadelphia, (1989).
- [5] Gerasoulis, A., Grigoriadis, M.D. and Sun, Liping: "A Fast Algorithm for Trummer's Problem", *SIAM J. Sci. Stat. Comput.*, **v.8**, No.1 (1987), s135-s138.
- [6] Szabo, B.A. and Babuska, I.: *Finite Element Analysis*, Wiley, New York, (1991).
- [7] Vajtersic, M.: *Algorithms for Elliptic Problems*, Kluwer Academic Publishers, London, (1992).
- [8] Weiwei, S. and Zamani, N.G.: "A fast algorithm for solving the tensor product collocation equations", *J. Franklin Inst.* (UK), **v.326** (1989), pp.295-307.