

テンソル積型の高次多項式基底を用いた FEMの計算技法について

村上 弘^{1,a)}

概要：有限要素法により典型的な線形二階の楕円型偏微分方程式の数値解を求めるにあたり、直方体状の有限要素を用いて要素内基底関数には各辺に沿った高次多項式のテンソル積を用いる。この方式を採用した場合の p -版 FEM に付随する技法を紹介する。この方式を用いて全体の連立線形方程式を反復法で解く場合に、多次元の問題では行列-行列積の演算が豊富になりデータ参照の局所性が高い。

キーワード：有限要素法, 基底, 記憶削減, 行列積, 高速計算法

Calculation Techniques of FEM Using Tensorial Product of Higher Order Polynomial Basis

HIROSHI MURAKAMI^{1,a)}

Abstract: We try to explain techniques for p -version FEM which solves numerically some typical 2nd order linear elliptic PDEs. The method uses rectilinears as finite elements and the expansion basis of the solution inside each element are tensorial products of higher order polynomials along edges. When the global linear equation is solved in some iterative manner for multi-dimensional PDE problems, this method is rich in matrix-matrix products to give a high locality of data references.

Keywords: FEM, basis, memory reduction, matrix multiplication, fast algorithm

1. はじめに

有限要素法 (FEM) により線形偏微分方程式の境界値問題を離散化すると、偏微分方程式を各要素ごとに離散化した小規模な係数行列を全ての要素について集計した係数を持つ大規模な連立一次方程式が得られる。

離散化で得られた大規模な連立一次方程式を Gauss 消去法などの直接法を用いて解くには、係数行列全体をまず構成しそれに消去変形を適用する過程で発生するフィルインも含めた記憶を保持する必要がある、問題の領域が三次元以上になると記憶量と計算量に対する負荷が非常に大きくなる。そこで主に記憶量の制約からは直接解法ではなく反

復解法が用いられる。

反復解法では計算の主要部は離散化された方程式の (全域的な) 係数行列を (全域的な) 係数ベクトルに乗じる「行列ベクトル積」の処理であり、各要素ごとの (局所的な) 離散化行列を (局所的な) 係数ベクトルに作用させた結果を集めて合計する Element-by-Element 法を使うことができる。E-by-E 法を用いて (全域的な) 「行列ベクトル積」の計算を行なう際に、(局所的な) 離散化行列は作用の直前に構成して乗算の計算直後に記憶場所を解放するのであれば、係数行列全体の記憶の保持は不要となり、作業記憶量に関する制約が緩和される。しかし各反復ごとに全要素で局所的な離散化行列を計算して構成するので、演算量がかなり多くなる。

今回解説する FEM の方法では、直方体形状の要素を採用してその各辺に沿った座標の高次多項式の組のテンソル

¹ 首都大学東京・数理情報科学専攻
Department of Mathematics and Information Sciences,
Tokyo Metropolitan University
^{a)} mrkmhrsh@tmu.ac.jp

積（直積）を要素内での展開基底として採用する。そのような要素形状と基底の構造の特徴を利用して計算法を組み立てると、(局所的な) 離散化行列を陽的には構成しないでそれを(局所的な) 係数ベクトルへ乗じた結果を得ることができ、しかも演算量も抑えられる。すると全体行列の連立方程式を反復法で E-by-E 法を利用して解くことが容易となり、必要記憶量を抑えた高速な計算が可能となる。

E-by-E 法では要素内の計算は独立に行なえるので、それを利用した並列分散化は比較的容易であり、要素内でのテンソル積型基底に基づく作用素を離散化した行列を乗じる作用は行列同士の積 (MATMUL) あるいはその高度な実装である BLAS3 のルーチン xGEMM を用いて実現できるので、計算機システムの性能を効率よく引き出すことが可能になる。

2. h -版 FEM と p -版 FEM

伝統的な「 h -版の FEM」では、解析対象とする全体領域をまず重なりのない「要素」と呼ばれる部分領域に分割し、要素内での関数の振る舞いを展開して記述するための局所的な基底関数を割り当てる。基底関数としては通常は低次の Lagrange 補間多項式を採用する。各要素内で偏微分作用素の離散化を Galerkin 法で行なう。各要素内で生成した離散化行列（要素行列）を解の連続性を考慮し、境界条件を含めながら集計して全体行列を構成する（組み立てる）。組み立てた全体行列を係数とする連立一次方程式あるいは固有値方程式を解くと、各要素内で区分的に基底関数により展開された近似解が得られる。近似を適応的に改良するには要素分割のサイズを小さくする。

他方で p -版 FEM では、基底関数に高次の多項式を使用して適応的に基底の次数を上げて近似を改良する。特徴としては、高精度な近似解が基底関数に高次の多項式を用いることで得られることや、必要ならば要素内の基底関数の次数を適応的に増減することが容易であること、解が滑らかならば p -版による近似解は h -版のものよりも少ない自由度で収束すること、さらに以下で説明する高速化の技法を用いた場合には、係数行列を生成や記憶せずに離散化された係数行列とベクトルの乗算ができ、しかも係数行列を生成してからベクトルに乘じるよりも少ない計算量で実行できるという利点がある。

3. 基底多項式について。

多項式の位数 (order) が k であるとは、次数が $(k-1)$ であることを意味することとする。基底の位数が k であるとは、位数が k 以下の任意の多項式が表現可能であることとする。空間が一次元であれば、位数 k の基底は k 個の多項式からなる。

3.1 積分 Legendre 多項式 (integrated Legendre)

3.1.1 定義

標準区間 $[-1, 1]$ 上の積分 Legendre (ILGN) 多項式 $\phi_j(x)$ を

$$\phi_1(x) = \frac{1-x}{2}, \quad \phi_2(x) = \frac{1+x}{2},$$

$$\phi_{\ell+2}(x) = \int_{-1}^x P_{\ell}(x) dx \quad (\ell \geq 1)$$

と定義する。ここで $P_{\ell}(x)$ は標準区間 $[-1, 1]$ に於ける ℓ 次の規格化された Legendre 多項式をあらわす。一般の区間 $[a, b]$ 上での定義は線型変換で得られる。ILGN 多項式は Legendre 多項式の線型結合として書くことができる。最

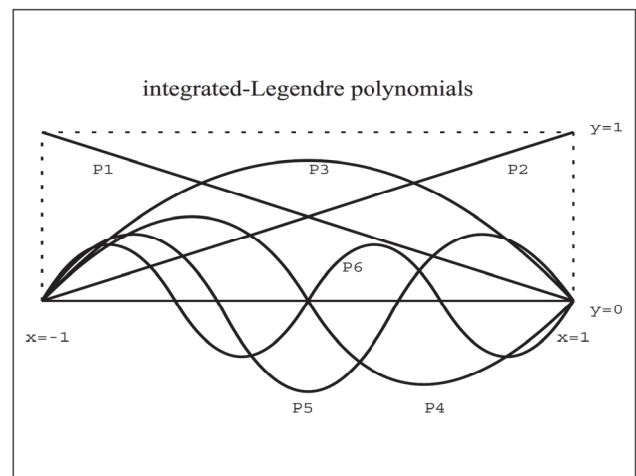


図 1 積分 Legendre 多項式 (ILGN) の例

初の 6 個の積分 Legendre 多項式のグラフを図 1 に示す。

3.1.2 積分 Legendre (ILGN) 基底系の性質

境界値：

ILGN 基底は、最初の二個は一次式であり $\phi_1(x)$ は区間の左端で値 1、右端では値 0 をそれぞれとり、 $\phi_2(x)$ は区間の左端で値 0、右端では値 1 をそれぞれとる。それより後の基底関数はどれも区間両端で値が零である。この性質により、ILGN 基底を用いた FEM では隣接する FEM 要素の境界に於ける近似関数の値の連続性を実現することが容易になる。

階層性：

ILGN 基底系は、低位の基底に新たな基底多項式を順次追加することで高位の基底が作られているので階層的 (hierarchical) である。低位の基底を用いて表わされている関数を高位の基底を用いて表わす場合には、低位の基底に対する展開係数は同一のまま追加した基底多項式に対する係数は零となる。

この ILGN 基底の持つ階層性の性質と、左端と右端で零でない値を持つ多項式はそれぞれ一個ずつであることから、(一般の空間次元で) 各要素に対する基底の位数を任意に割り当てても要素の境界上における近似関数の値の連続

性を実現することが容易になる。

また階層性を利用すると、解の領域中の各場所での振る舞いに対応して要素を細かく分割あるいは粗く分割する (h-版 FEM) かわりに、要素に割り当てる基底系の位数を上げるあるいは下げる (p-版 FEM) ことで適応的な計算を比較的容易に実現できる。(但し、ILGN 基底系は階層的であっても直交基底系ではないので、ある関数の基底による展開係数を二乗ノルムで最良となるように決めるとき、多項式を追加して基底を拡大し高位にするとそれ以前の低位の部分の展開係数は保たれずに変化する。また、多項式を削除して基底を高位から低位に縮小するときも、最良となるように決めた近似では低位の部分の展開係数は変化する。)

演算子の離散化：

一次元の Laplacian や恒等演算子を ILGN 基底を用いて FEM で離散化した行列は極めて疎な行列になり、 k 次の行列の非零要素の数は $O(k)$ である。ILGN 基底のテンソル積を基底として三次元の Laplacian を FEM で離散化した行列は、一次元の Laplacian と恒等演算子の離散化行列の組合せとして書ける。このことから Poisson 方程式に現われる三次元 Laplacian の離散化は ILGN 基底のテンソル積を用いるのが便利である。三次元 Laplacian の離散化行列をベクトルへ乗じた結果を求める演算量は基底の位数が (k_1, k_2, k_3) の要素では $O(k_1 \times k_2 \times k_3)$ となり、高速に計算が行なえる。

一次元演算子の行列要素：

標準区間 $[-1, 1]$ 上での、典型的な一次元演算子を FEM で離散化した行列の要素は、ILGN 多項式の定義から容易に計算される。

Laplacian の離散化は

$$\begin{aligned}
 L_{p,q} &= \frac{1}{2} & (p, q) &= (1, 1), (2, 2) \\
 &= \frac{-1}{2} & (p, q) &= (1, 2), (2, 1) \\
 &= 1 & p > 2, q > 2 \text{ and } p=q \\
 &= 0 & \text{otherwise}
 \end{aligned}$$

であり、最初の二つの行と二つの列 ($p-1, 2$ と $q-1, 2$) 以外では、対角の位置だけに非零要素 1 を持つ疎行列である。

恒等演算子の離散化は

$$\begin{aligned}
 M_{p,q} &= \frac{2}{3} & (p, q) &= (1, 1), (2, 2) \\
 &= \frac{1}{3} & (p, q) &= (1, 2), (2, 1) \\
 &= \frac{-1}{\sqrt{6}} & (p, q) &= (1, 3), (2, 3), (3, 1), (3, 2) \\
 &= \frac{1}{3\sqrt{10}} & (p, q) &= (1, 4), (4, 1) \\
 &= \frac{-1}{3\sqrt{10}} & (p, q) &= (2, 4), (4, 2) \\
 &= \frac{2}{(2p-1)(2p-5)} & p > 2, q > 2 \text{ and } p=q \\
 &= \frac{-1}{(2p-5)\sqrt{(2p-3)(2p-7)}} & p > 2, q > 2 \text{ and } |p-q|=2 \\
 &= 0 & \text{otherwise}
 \end{aligned}$$

であり、最初の二つの行と二つの列以外では、対角の位置 ($p=q$) および対角から 2 離れた位置 ($|p-q|=2$) だけに非零要素を持つ疎行列である。

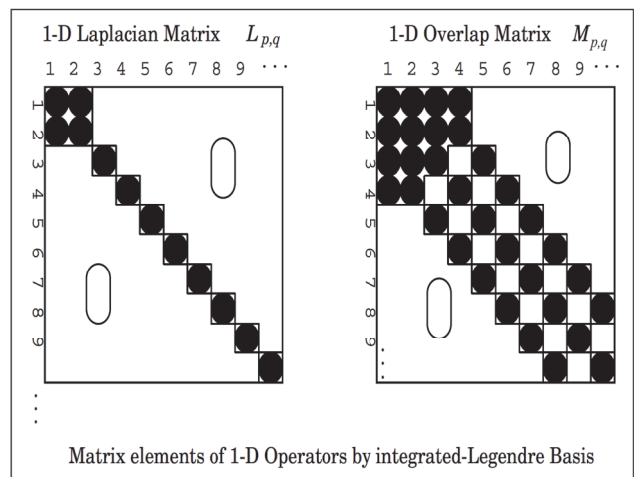


図 2 ILGN 基底による離散化行列の例

3.2 Lobatto-Gauss (LBGS) 多項式

3.2.1 定義

位数 k の Lobatto-Gauss (LBGS) 多項式 $\eta_j^{(k)}(x)$, $j = 1, 2, \dots, k$ は、 k 点 Lobatto-Gauss (LBGS) 数値積分公式の k 個の分点 $\{x_j\}$ 上の Lagrange 内挿の基底多項式である。区間が $[-1, 1]$ 以外での定義は座標 x の線型変換から得られる。以後では基底の位数 k を省略して $\eta_j(x)$ と書くことがある。

3.2.2 LBGS 積分公式

標準区間での被積分関数 $F(x)$ に対する k 点 LBGS 積分公式は：

$$\int_{-1}^1 F(x) dx \approx \sum_{j=1}^k w_j F(x_j)$$

である。被積分関数 $F(x)$ が $(2k-3)$ 次以下の多項式であればこの k 点公式は正確な積分値を与える。積分区間が $[-1, 1]$ 以外の公式も線型変換により得られる。

標準区間 $[-1, 1]$ 上の LBGS 積分公式の分点 x_j ,

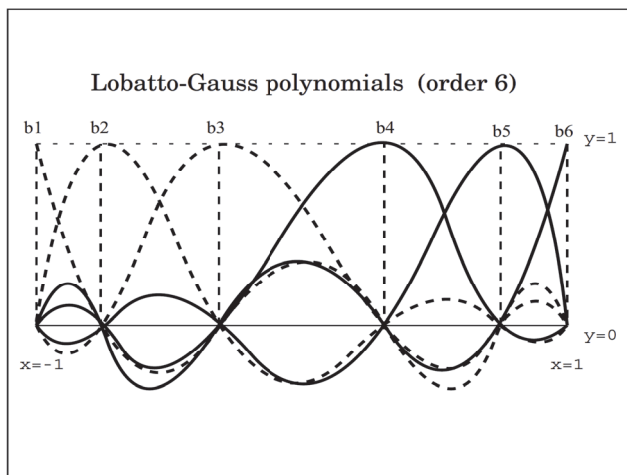


図 3 Lobatto-Gauss 多項式の例

$j=1, 2, \dots, k$ は、区間の両端点 $x_1 = -1$ と $x_k = 1$ を含み、それ以外の分点は $(k-2)$ 次の多項式 $\frac{d}{dx} P_{k-1}(x)$ の左端から $(j-1)$ 番目の零点である。ここで $P_m(x)$ は標準区間 $[-1, 1]$ に対応する m 次の Legendre 多項式をあらわす。そうして標準区間 $[-1, 1]$ 上の k 点 LBGS 積分公式の各分点に付随する重みは：

$$w_1, w_k = \frac{2}{k(k-1)},$$

$$w_j = \frac{2}{k(k-1)} \{P_{k-1}(x_j)\}^{-2} \quad (j=2, \dots, k-1),$$

である。

3.2.3 LBGS 基底系の性質

分点上の値：

LBGS 多項式は Lagrange 内挿の基底多項式であることから性質 $\eta_j^{(k)}(x_i) = \delta_{i,j}$ を持つ。ここで $\delta_{i,j}$ は Kronecker の delta 記号を表す。

内挿公式：

位数 k の LBGS 多項式基底を用いた関数 $F(x)$ の内挿公式は、上のことから内挿の係数 $C_j = F(x_j)$ により $F(x) \simeq \sum_{j=1}^k C_j \eta_j(x)$ となる。

準直交性：

FEM で固有値問題を解析するとき、LBGS 多項式の持つ準直交性が便利に使える場合がある。いま $\eta_i(x)$, $\eta_j(x)$ を位数 k の LBGS 多項式とすると、1 次元の Gram 行列 (質量行列, Mass matrix) $M_{i,j} = \int_{-1}^1 \eta_i(x) \eta_j(x) dx$ は正確に計算すると密行列である。被積分関数 $\eta_i(x) \eta_j(x)$ は $(k-1)$ 次の多項式二個の積であるから $(2k-2)$ 次の多項式である。 k 点 LBGS 積分公式は $(2k-3)$ 次以下の多項式に対しては正確であるが、あいにく被積分関数である多項式の次数はそれよりも 1 次だけ高いから、 k 点の数値積分は正確な値を与えない。しかしそれを理解した上で、(位数が一つ不足の) k 点公式を使用して Gram 行列 M の定義の積分を近似してみると：

$$\widehat{M}_{i,j} = \sum_{p=1}^k w_p \eta_i(x_p) \eta_j(x_p) = \sum_{p=1}^k w_p \delta_{i,p} \delta_{j,p} = w_i \delta_{i,j}$$

となり、この近似では (積分公式の重みを対角に並べた) 対角行列 \widehat{M} になる。つまり精度の足りない数値積分で計算すれば Gram 行列が対角になる。このことを基底の「準直交性」と呼ぶことにする。この Gram 行列の対角近似で導入される近似誤差は、基底の位数を一つ下げたのと同程度で、解が滑らかで基底の位数が高ければ十分小さいと期待できる。

またこれは次のように考えることもできる。いま $f(x)$ と $g(x)$ を高々 $(k-2)$ 次の多項式とする。積 $f(x)g(x)$ の次数は高々 $(2k-4)$ で $(2k-3)$ 以下だから 区間 I 内での積分の値 $J = \int_{-1}^1 f(x)g(x) dx$ は k 点の LBGS 数値積分公式による値 $J' = \sum_{p=1}^k w_p f(x_p)g(x_p)$ と一致する。いま $f(x)$ と $g(x)$ をそれぞれ (それらよりも一次だけ次数の高い) 位数 k ($(k-1)$ 次) の LBGS 基底を用いて展開したものを $f(x) = \sum_{i=1}^k c_i \eta_i(x)$, $g(x) = \sum_{j=1}^k d_j \eta_j(x)$ とする。すると本来の積分の値は $J = \sum_{i,j} c_i d_j \int_{-1}^1 \eta_i(x) \eta_j(x) dx$ である。ここで $\int_{-1}^1 \eta_i(x) \eta_j(x) dx = M_{i,j}$ は基底の Gram 行列である。行列記法で書けば $J = \mathbf{c}^T \mathbf{M} \mathbf{d}$ となる。これに対して準直交性を用いて密行列 M の対角近似 $M_{i,j} \approx \widehat{M}_{i,j} = w_i \delta_{i,j}$ を取り入れたものは $J \approx \sum_{i,j} c_i d_j (w_i \delta_{i,j}) = \sum_i w_i c_i d_i$ となる。行列記法で書けば、 $J = \mathbf{c}^T \widehat{\mathbf{M}} \mathbf{d}$ である。一方で、LBGS 数値積分公式により与えられる正確な積分の値を求めてみると $J' = \sum_{p=1}^k w_p \sum_{i,j} c_i d_j \eta_i(x_p) \eta_j(x_p) = \sum_{i,j} c_i d_j \sum_{p=1}^k w_p \eta_i(x_p) \eta_j(x_p) = \sum_{i,j} c_i d_j \sum_{p=1}^k w_p \delta_{i,p} \delta_{j,p} = \sum_{i,j} c_i d_j (w_i \delta_{i,j}) = \sum_i w_i c_i d_i$ となる。

つまり、 \mathbf{c} , \mathbf{d} をそれぞれ高々 $(k-2)$ 次の多項式を位数 k の LBGS 基底関数で展開して得られた係数のベクトルであるとする (注：位数 k の基底による展開は一般の状況では $(k-1)$ 次の多項式を表現する)。その場合には $\mathbf{c}^T \mathbf{M} \mathbf{d} = \mathbf{c}^T \widehat{\mathbf{M}} \mathbf{d}$ となるので、離散化係数のベクトルの (一般化された) 内積について、その重みである Gram 行列 M を準直交性による対角近似行列 \widehat{M} で置き換えても内積の値は変わらないことがわかる。

さて FEM で線形偏微分方程式の固有値問題を近似すると通常は一般化固有値問題が得られる。LBGS 基底を用いて離散化した Gram 行列 (質量行列) に対して準直交性の近似を採用すれば (その近似による誤差の導入を認めるものとして)、FEM 要素ごとの対角行列を重ね合わせた全体行列もまた対角行列になるので標準固有値問題へ簡単に還元できる (ただし、全体の FEM 要素分割と基底関数の割り当ての方法を特別に選ぶ必要がある)。

注記：

準直交性を持つ Lagrange 基底は LBGS 基底だけではない。いま数値積分公式 $\int_I F(x) dx \approx \sum_p w_p F(x_p)$ を任

意に選び、その分点分布 $\{w_p\}$ 上の Lagrange 内挿の基底多項式 $\varphi_p(x)$ を作る。基底の Gram 行列を定義する積分 $M_{p,q} = \int_I \varphi_p(x)\varphi_q(x) dx$ をその数値積分公式で近似して得られる行列は、公式の重みを対角に並べた行列 $\widehat{M}_{p,q} = w_p\delta_{p,q}$ になることがわかる。この近似の程度は選んだ数値積分公式によるが、公式の分点分布が考えている区間 I に含まれていることや分点での重みがすべて正であることは強く望ましい性質である。隣接要素の境界上で表現したい関数の値が連続になる条件を容易に扱えるように分点分布に区間の両端を含めるならば、準直交性の近似度は LBGS 基底の場合に最大になる。(もしもそのような端点での条件が不要ならば、 k 点の Gauss-Legendre の数値積分公式に対応した Lagrange 内挿の基底多項式をとれば、2 個の基底多項式の積は $(2k-2)$ 次であるが k 点の Gauss-Legendre 数値積分公式は $(2k-1)$ 次の多項式に対して正確であることから、この場合には $M_{p,q} = \widehat{M}_{p,q}$ が成立するので、Gram 行列は対角行列で、基底多項式は互いに直交する。)

4. 行列ベクトル積の高速計算の技法

FEM の要素が直方体状で、要素内の関数の展開基底は要素の各辺の方向を座標とする 1 次元の基底関数のテンソル積型であるとする。その場合に演算子を離散化した“行列”を“係数ベクトル”に乗じた結果を、行列を実際には生成しないで計算する技法を示す。これは行列を生成してからベクトルに乗じる素朴な方法よりも演算量が少ない利点がある。つまりこの技法はそのような特別な要素と基底を用いた演算子の FEM による離散化行列に対する「行列ベクトル積」の高速乗算法であり、しかも演算子の離散化行列を生成や記憶しておく必要がない。連立一次方程式や固有値問題の反復解法では、反復のたびごとに係数行列とベクトルの乗算が必要になるが、この技法を用いるとその乗算を高速にしかも行列自体を生成しないで行なえるので、大規模な問題の扱いが容易になる。

典型的な演算子を離散化してこの方法を例示する。簡単のために用いる FEM 要素はすべて x_1, x_2, x_3 軸に沿った直方体とする(曲線座標により写像された曲がった要素を扱う場合には、たとえばポテンシャルの項であれば被積分項に写像の Jacobian を含めれば同様に定式化できる)。

以下では次の表記法を説明の記述に用いる：
 $e = (e_1, e_2, e_3)$ ：要素は区間の直積。

$\mathbf{x} = (x_1, x_2, x_3)$ ：座標。

$i = (i_1, i_2, i_3), j = (j_1, j_2, j_3)$ ：テンソル積型基底関数の添字。

$b_i(\mathbf{x}) = b_{i_1}^{(1)}(x_1)b_{i_2}^{(2)}(x_2)b_{i_3}^{(3)}(x_3)$ ：テンソル積型基底関数。

4.1 ポテンシャル項

例えば空間三次元の Poisson 方程式のポテンシャル項が

座標 \mathbf{x} の関数 $U(\mathbf{x})$ として与えられたとする。このポテンシャルの FEM による要素 e に対応する離散化行列は：

$$V_{i,j} = \int_e b_i(\mathbf{x}) U(\mathbf{x}) b_j(\mathbf{x}) dx$$

である。直方体要素内の関数 $F(\mathbf{x})$ の積分は(適当な)積型数値積分公式により、次のように近似される：

$$\int_e F(\mathbf{x}) dx = \sum_{q_1, q_2, q_3} w(q_1, q_2, q_3) F(q_1, q_2, q_3),$$

ここで q_1, q_2, q_3 は積型数値積分公式の分点であり、 $w(q_1, q_2, q_3)$ は積型数値積分公式の重みである。以降では q_1, q_2, q_3 をそれぞれ数値積分の分点の座標の意味と分点の番号の意味の両方に用いている。

これらの用意の上で、ポテンシャルを離散化して得られた行列 $V_{i,j}$ を係数ベクトル C_j に作用させると：

$$\begin{aligned} \sum_j V_{i,j} C_j &= \sum_{q_1, q_2, q_3} w(q_1, q_2, q_3) \\ &\quad \times b_{i_1}^{(1)}(q_1) b_{i_2}^{(2)}(q_2) b_{i_3}^{(3)}(q_3) \\ &\quad \times U(q_1, q_2, q_3) \\ &\quad \times b_{j_1}^{(1)}(q_1) b_{j_2}^{(2)}(q_2) b_{j_3}^{(3)}(q_3) \\ &\quad \times C_{j_1, j_2, j_3}. \end{aligned}$$

上式の値は、以下の三つのステップで計算される：

STEP-1. 積分の分点に於ける値を求める：

$$C'(q_1, q_2, q_3) \leftarrow \sum_{j_1, j_2, j_3} b_{j_3}^{(3)}(q_3) b_{j_2}^{(2)}(q_2) b_{j_1}^{(1)}(q_1) C_{j_1, j_2, j_3}.$$

添字 j_1, j_2, j_3 は順次に縮約される(添字 \mathbf{j} から \mathbf{q} への変換)。

STEP-2. 中間のスケーリング：

$$C''(q_1, q_2, q_3) \leftarrow w(q_1, q_2, q_3) U(q_1, q_2, q_3) C'(q_1, q_2, q_3).$$

ここでは添字は縮約されない。

STEP-3. 添字 \mathbf{q} から \mathbf{i} への逆変換：

$$C'''(i_1, i_2, i_3) \leftarrow \sum_{q_1, q_2, q_3} b_{i_3}^{(3)}(q_3) b_{i_2}^{(2)}(q_2) b_{i_1}^{(1)}(q_1) C''(q_1, q_2, q_3).$$

添字 q_1, q_2, q_3 は順次に縮約される。

STEP-1 はさらに次の三つのステップで計算される：

STEP-1a. C_{j_1, j_2, j_3} を $C0(j_1; [j_2, j_3])$ とみなして、

$$C1([j_2, j_3]; q_1) \leftarrow \sum_{j_1} C0(j_1; [j_2, j_3]) B1(j_1; q_1).$$

STEP-1b. $C1([j_2, j_3]; q_1)$ を新たに $C1(j_2; [j_3, q_1])$ で

あるとみなして、

$$C2([j_3, q_1]; q_2) \Leftarrow \sum_{j_2} j_2 C1(j_2; [j_3, q_1]) B2(j_2; q_2).$$

STEP-1c. $C2([j_3, q_1]; q_2)$ を再び $C2(j_3; [q_1, q_2])$ であるとみなして、

$$C3([q_1, q_2]; q_3) \Leftarrow \sum_{j_3} C2(j_3; [q_1, q_2]) B3(j_3; q_3).$$

この $C3([q_1, q_2]; q_3)$ が STEP-1 の $C'(q_1, q_2, q_3)$ である。

添字を三個持つ量を行列と見なして行と列の添字をセミコロンの分離して書き、角括弧に囲まれた組を行列の添字一個のように扱うことを強調している。

添字 j_1 の縮約：

$$C1([j_2, j_3]; q_1) \Leftarrow \sum_{j_1} C0(j_1; [j_2, j_3]) B1(j_1; q_1)$$

は、以下のように二個の行列の積として実行することができる：

$$C1 \Leftarrow \text{MATMUL}(\text{TRANSPOSE}(C0), B1).$$

(また、添字 j_2 の縮約と添字 j_3 の縮約についても同様である。) この縮約演算は行列同士の積であり、例えば BLAS3 ルーチンの xGEMM を用いて実装すれば高い性能が期待できる。

いま各方向の数値積分の分点の個数を Q_1, Q_2, Q_3 であらわし、さらに各方向の基底関数系 b_1, b_2, b_3 の位数をそれぞれ B_1, B_2, B_3 とあらわすことにする。そのとき STEP-1 での全演算量は、積和の回数で数えて：

$$(Q_1 B_1 B_2 B_3) + (Q_1 Q_2 B_2 B_3) + (Q_1 Q_2 Q_3 B_3).$$

STEP-2 では、 $U(q_1, q_2, q_3)$ と $w(q_1, q_2, q_3)$ の値があらかじめ用意されていて利用可能であるとすると、このステップの演算回数は $(Q_1 Q_2 Q_3)$ のオーダーである。STEP-3 の計算は STEP-1 と同様におこなえて、その演算量は積和の回数で $(B_1 Q_1 Q_2 Q_3) + (B_1 B_2 Q_2 Q_3) + (B_1 B_2 B_3 Q_3)$ のオーダーである。

注記：

$Q_1 = Q_2 = Q_3, B_1 = B_2 = B_3$ ではないときには、演算量の面だけからすると添字の縮約の最適な順が i_1, i_2, i_3 ではないことがあるが、一応この順で説明する。また数値積分の分点の個数 Q_1, Q_2, Q_3 は基底関数の位数 B_1, B_2, B_3 や要素内でのポテンシャル関数の振舞いも考慮に入れて選ぶ。

STEP-3 での順次の縮約についても同様である。

4.2 関数の分点での値の計算

いま三次元直方体要素内で、関数 $F(\mathbf{x})$ がある基底関数系 $b_j(\mathbf{x})$ の展開係数 G_j により与えられているとする：

$$F(\mathbf{x}) = \sum_j b_j(\mathbf{x}) G_j.$$

そのとき、積型積分公式のすべての分点上で $F(\mathbf{x})$ の値を求める計算は基底関数系 $b_j(\mathbf{x})$ がテンソル積型であれば、次のように書ける：

$$F(q_1, q_2, q_3) \Leftarrow \sum_{j_1, j_2, j_3} b_{j_1}^{(1)}(q_1) b_{j_2}^{(2)}(q_2) b_{j_3}^{(3)}(q_3) G_{j_1, j_2, j_3} \\ = \{ \sum_{j_1} b_{j_1}^{(1)}(q_1) \{ \sum_{j_2} b_{j_2}^{(2)}(q_2) \{ \sum_{j_3} b_{j_3}^{(3)}(q_3) G_{j_1, j_2, j_3} \} \} \}.$$

これを高速に計算する方法は、上記の STEP-1 の計算の場合と同様である。

4.3 Laplacian

三次元では(符号が逆の $-\nabla^2$ である) Laplacian の有限要素 \mathbf{e} に対応する離散化行列の要素は：

$$L_{i,j} = \int_{\mathbf{e}} \nabla b_i(\mathbf{x}) \nabla b_j(\mathbf{x}) d\mathbf{x} \\ = \int_{\mathbf{e}} \{ b_{i_1}^{(1)}(x_1) b_{i_2}^{(2)}(x_2) b_{i_3}^{(3)}(x_3) b_{j_1}^{(1)}(x_1) b_{j_2}^{(2)}(x_2) b_{j_3}^{(3)}(x_3) \\ + b_{i_1}^{(1)}(x_1) b_{i_2}^{(2)}(x_2) b_{i_3}^{(3)}(x_3) b_{j_1}^{(1)}(x_1) b_{j_2}^{(2)}(x_2) b_{j_3}^{(3)}(x_3) \\ + b_{i_1}^{(1)}(x_1) b_{i_2}^{(2)}(x_2) b_{i_3}^{(3)}(x_3) b_{j_1}^{(1)}(x_1) b_{j_2}^{(2)}(x_2) b_{j_3}^{(3)}(x_3) \} d\mathbf{x},$$

ここでプライムは微分を表す。

直方体要素では上記の三次元積分は、以下のように一次元積分の組み合わせた形に簡約できる：

$$L_{i,j} = L_{i_1, j_1}^{(1)} M_{i_2, j_2}^{(2)} M_{i_3, j_3}^{(3)} \\ + M_{i_1, j_1}^{(1)} L_{i_2, j_2}^{(2)} M_{i_3, j_3}^{(3)} \\ + M_{i_1, j_1}^{(1)} M_{i_2, j_2}^{(2)} L_{i_3, j_3}^{(3)}.$$

ここで行列 $L^{(1)}$ と $M^{(1)}$ は一次元の場合の Laplacian と恒等演算子の離散化であり、以下の一次元積分で定義される：

$$L_{i_1, j_1}^{(1)} = \int_{e_1} b_{i_1}^{(1)}(x_1) b_{j_1}^{(1)}(x_1) dx_1, \\ M_{i_1, j_1}^{(1)} = \int_{e_1} b_{i_1}^{(1)}(x_1) b_{j_1}^{(1)}(x_1) dx_1.$$

また行列 $L^{(2)}, M^{(2)}$ や $L^{(3)}, M^{(3)}$ の定義も同様である。(基底関数を多項式に採ったので $L^{(1)}, L^{(2)}, L^{(3)}$ および $M^{(1)}, M^{(2)}, M^{(3)}$ の行列要素は解析的に求めることができる。特に基底関数が積分 Legendre 多項式 (ILGN) の場合には、既に述べたように非常に疎な行列になるので、行列の乗算は極めて容易であり高速に実行できる。)

離散化行列 $L_{i,j}$ の展開係数ベクトル C_j への乗算は、STEP-1 で述べたのと同様に各項ごと独立に計算される：

$$\sum_j L_{i,j} C_j \\ = \{ \sum_{j_3} M_{i_3, j_3}^{(3)} \{ \sum_{j_2} M_{i_2, j_2}^{(2)} \{ \sum_{j_1} L_{i_1, j_1}^{(1)} C_{j_1, j_2, j_3} \} \} \} \\ + \{ \sum_{j_3} M_{i_3, j_3}^{(3)} \{ \sum_{j_2} L_{i_2, j_2}^{(2)} \{ \sum_{j_1} M_{i_1, j_1}^{(1)} C_{j_1, j_2, j_3} \} \} \} \\ + \{ \sum_{j_3} L_{i_3, j_3}^{(3)} \{ \sum_{j_2} M_{i_2, j_2}^{(2)} \{ \sum_{j_1} M_{i_1, j_1}^{(1)} C_{j_1, j_2, j_3} \} \} \}.$$

4.4 恒等演算子

恒等演算子の離散化は、基底関数の Gram 行列（質量行列）である。有限要素 c の寄与は：

$$M_{i,j} = \int_e b_i(\mathbf{x})b_j(\mathbf{x}) dx = M_{i_1,j_1}^{(1)} M_{i_2,j_2}^{(2)} M_{i_3,j_3}^{(3)}$$

となる。この三次元恒等演算子の場合の離散化行列を高速に乗じる方法は、三次元 Laplacian の場合と同様になる。

5. 要素間での連続性の保証について

隣接する FEM 要素の境界上で、近似展開された関数値の連続性を保証するには、隣接するそれぞれの要素での基底による展開係数に拘束条件を課す必要がある。

いま FEM 要素は直方体状であるとする。そうして要素に於ける関数展開の基底は直方体状要素の各辺に沿った局所座標軸の標準区間の多項式基底のテンソル積（直積）で、しかも各局所座標軸に対するそれぞれの基底が（ILGN 基底のように）階層的で、さらに区間両端での値を表現する自由度に相当する二個と境界では値が零である区間内部の自由度に分けられるとする。そのときには、連続性を保証するための拘束条件は要素の表面での自由度に対するものとなるので、設定が容易になる。

5.1 直方体状の要素に対する基底の種類

5.1.1 一次元要素

1次元の場合には要素に対する ILGN 基底には、

- 左端で値が 1 で右端で値が 0 である $\phi_1(x)$ と、左端で値が 0 で右端で値が 1 である $\phi_2(x)$ の二個の自由度。
- 要素の端点では値を持たない $\phi_\ell(x)$ ($\ell > 2$) の自由度。

関数の近似展開の値の連続性を保証するためには、頂点を共有する二個の要素について、共有された頂点に対する自由度の係数は一致することが必要である。

注意：

一般には、二個の要素についてそれぞれの基底の定義に用いた局所座標の標準区間 $[-1, 1]$ の向きが、元の空間の中では逆になっている可能性がある。そのような場合には ILGN 基底の場合には $\phi_1(-x) = \phi_2(x)$, $\phi_2(-x) = \phi_1(x)$, $\phi_\ell(-x) = (-1)^{\ell-1}\phi_\ell(x)$ ($\ell > 2$) の関係を用いて係数の番号あるいは符号を変換する必要がある。 k 位の LBGS 基底の場合には内挿の分点の番号を局所座標の増大順に付けていれば、向きが逆になれば大小が入れ替わるので $\eta_\ell^{(k)}(-x) = \eta_{k-\ell+1}^{(k)}(x)$ ($\ell=1, 2, \dots, k$) の関係を用いて係数の番号を付け替える。もしも最初から領域の要素分割を行なってすべての要素の局所座標の向きを整合するように揃えられるのであれば、この注意は不要である。領域を基板目状に要素分割するのであれば簡単に、すべての要素で局所座標の向きが整合的となるようにできる。

5.1.2 二次元要素

二次元の場合には、長方形要素に対する基底には

- 長方形の四個の頂点に対応する、
 $\phi_1^{(1)}(x)\phi_1^{(2)}(y)$ (左下頂点), $\phi_2^{(1)}(x)\phi_1^{(2)}(y)$ (右下頂点),
 $\phi_1^{(1)}(x)\phi_2^{(2)}(y)$ (左上頂点), $\phi_2^{(1)}(x)\phi_2^{(2)}(y)$ (右上頂点)
 の四個の基底。
- 長方形の四個の辺の内部（頂点を含まない）に対応する、
 $\phi_1^{(1)}(x)\phi_j^{(2)}(y)$ (左側の辺), $\phi_2^{(1)}(x)\phi_j^{(2)}(y)$ (右側の辺),
 $\phi_i^{(1)}(x)\phi_1^{(2)}(y)$ (下側の辺), $\phi_i^{(1)}(x)\phi_2^{(2)}(y)$ (上側の辺)
 の四種類の基底。ただし、ここでは添字 i, j は 3 以上の番号を表す。
- 長方形の一個の面の内部（頂点も辺も含まない）に対応した、 $\phi_i^{(1)}(x)\phi_j^{(2)}(y)$ の 1 種類の基底。ただし、ここでは添字 i, j は 3 以上の番号を表す。

がある。

二次元 FEM 要素の境界上で基底展開による関数の値が連続であるためには、

- 互いに頂点を共有する FEM 要素に対して、その共有する頂点の自由度に対する展開係数が一致する。
- 互いに辺を共有する FEM 要素に対して、その共有する辺内部（頂点を含めない）の自由度に対する展開係数が一致する。

ことが必要である。（要素の間の局所座標の向きのとりがたが逆になっている場合に対する注意は同様に適用される。）二次元の場合には、共有する辺の内部に対する自由度は、その両側の二個の要素がそれぞれテンソル基底としてその辺の向きに対して割り当てた基底の位数が異なる場合には、それが ILGN 基底であれば階層性からより低位の側の基底の位数を k_{\min} とするとき、その方向の基底の番号 j が $2 < j \leq k_{\min}$ を満たすような辺の内部に対する自由度は割り当てが、それ以外は自由度には含めない。E-by-E 法で要素個別の行列ベクトル積を計算するときには、（テンソル積型の基底の係数に対応する）係数ベクトルの冗長な自由度に対しては零を入れておいて離散化行列を乗じる計算を行い、得られた積のベクトルの冗長な自由度に対応する値は（Galerkin 法なので）単に無視して捨てる。

例えば二個の二次元要素の ILGN 基底の位数がそれぞれ (10, 20) と (15, 5) になっていてそれらが第二座標軸に平行な辺を共有していたとすると、その辺上では真の位数は低位側の位数 $\min(20, 5) = 5$ であり、位数 5 を持つ側の第二番目の要素は共有する辺に対して全ての自由度があるが、第一番目の要素は位数 5 を越えた部分の $20 - 5 = 15$ 個の冗長な自由度を辺の内部に持つ。両方の要素が共有している「辺の内部」の自由度の個数は、頂点の自由度二個を除いた $5 - 2 = 3$ である。

5.1.3 三次要素

三次元の場合には、直方体要素に対する基底には、

- 直方体の八個の頂点に対応して、
 $\phi_1^{(1)}(x)\phi_1^{(2)}(y)\phi_1^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_1^{(2)}(y)\phi_1^{(3)}(z)$,
 $\phi_1^{(1)}(x)\phi_2^{(2)}(y)\phi_1^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_2^{(2)}(y)\phi_1^{(3)}(z)$,
 $\phi_1^{(1)}(x)\phi_1^{(2)}(y)\phi_2^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_1^{(2)}(y)\phi_2^{(3)}(z)$,
 $\phi_1^{(1)}(x)\phi_2^{(2)}(y)\phi_2^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_2^{(2)}(y)\phi_2^{(3)}(z)$
 の八個の基底がある。
- 直方体の十二個の辺の内部（頂点を含まない）に対応して
 $\phi_1^{(1)}(x)\phi_1^{(2)}(y)\phi_\ell^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_1^{(2)}(y)\phi_\ell^{(3)}(z)$,
 $\phi_1^{(1)}(x)\phi_2^{(2)}(y)\phi_\ell^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_2^{(2)}(y)\phi_\ell^{(3)}(z)$,
 $\phi_1^{(1)}(x)\phi_j^{(2)}(y)\phi_1^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_j^{(2)}(y)\phi_1^{(3)}(z)$,
 $\phi_1^{(1)}(x)\phi_j^{(2)}(y)\phi_2^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_j^{(2)}(y)\phi_2^{(3)}(z)$,
 $\phi_i^{(1)}(x)\phi_1^{(2)}(y)\phi_1^{(3)}(z)$, $\phi_i^{(1)}(x)\phi_2^{(2)}(y)\phi_1^{(3)}(z)$,
 $\phi_i^{(1)}(x)\phi_1^{(2)}(y)\phi_2^{(3)}(z)$, $\phi_i^{(1)}(x)\phi_2^{(2)}(y)\phi_2^{(3)}(z)$
 の十二種類の基底がある。ただし、ここでは添字 i, j, ℓ は 3 以上の番号を表す。
- 直方体の六個の面の内部（頂点および辺を含まない）に対応して
 $\phi_i^{(1)}(x)\phi_j^{(2)}(y)\phi_1^{(3)}(z)$, $\phi_i^{(1)}(x)\phi_j^{(2)}(y)\phi_2^{(3)}(z)$,
 $\phi_1^{(1)}(x)\phi_j^{(2)}(y)\phi_\ell^{(3)}(z)$, $\phi_2^{(1)}(x)\phi_j^{(2)}(y)\phi_\ell^{(3)}(z)$,
 $\phi_i^{(1)}(x)\phi_1^{(2)}(y)\phi_\ell^{(3)}(z)$, $\phi_i^{(1)}(x)\phi_2^{(2)}(y)\phi_\ell^{(3)}(z)$
 の六種類の基底がある。ただし、ここでは添字 i, j, ℓ は 3 以上の番号である。
- 最後に直方体の内部（頂点、辺、面を含まない）に対応して $\phi_i^{(1)}(x)\phi_j^{(2)}(y)\phi_\ell^{(3)}(z)$ の一種類の基底がある。ただし、ここでは添字 i, j, ℓ は 3 以上の番号を表す。

上記のように分類した基底の種別をそれぞれ、頂点 (Vertex), 辺 (Edge), 面 (Face), 体 (Body) などと呼ぶことにする。

三次元 FEM 要素の境界上で基底展開による関数の値が連続であるためには、

- 互いに頂点を共有する要素に対して、その共有する頂点の自由度に対する展開係数が一致すること。
- 互いに辺を共有する要素に対して、その共有する辺内部（頂点を含めない）の自由度に対する展開係数が一致すること。
- 互いに面を共有する要素に対して、その共有する面内部（頂点を含めない）の自由度に対する展開係数が一致すること。

が必要である。（要素の間の局所座標の向きのとりにかたが逆になっている場合に対する注意は、同様に適用される。）

共有する辺の内部の自由度は、辺を共有する要素のその辺の向きに各要素が割り当てた基底の位数の最小値から頂点の自由度の 2 を引いて決まる。

共有する面の内部の自由度は、その面を共有する二個の

要素をその面を張る座標軸の二つの方向に対してそれぞれ両方の要素が割り当てた基底の位数の最小値を求めて、それらから頂点の自由度の 2 を引いたものの積になる。その二つの方向のどちらかに関して位数が最小値よりも大きい基底は、自由度としては冗長である。

いまたとえば、面を共有する二個の要素に割り当てた位数がそれぞれ (10, 30, 20), (15, 25, 40) であるとする。共有した面が第二軸と第三軸で張られたものであるとすると、 $\min(30, 25) = 25$, $\min(20, 40) = 20$ であるから、共有する面では第二軸目の位数が 25, 第三軸目の位数が 20 のいずれかを越えた部分の自由度は冗長である。この共有する面の内部（頂点や辺を除く）での自由度の個数は、 $(25 - 2) \times (20 - 2) = 23 \times 18$ になる。

5.2 大域的係数ベクトルと局所的係数ベクトルの間の対応

領域全体での関数の自由度を集めた（大域的な）ベクトルを作るには、（まず境界条件を与える自由度を除外して）頂点 (Vertex), 辺 (Edge), 面 (Face), 体 (Body) に分類されるそれぞれの「図形」について、共有されている「図形」は一つのもので重複を排除して考え、それぞれの「図形」についてそれを共有している FEM 要素を調べて、あらかじめ FEM 要素に割り当てられたテンソル積型基底の位数に関する情報を総合することにより、その「図形」に対応する（冗長な自由度の個数を除いた）真の自由度を決定する。そのようにして、図形ごとに真の自由度が決まれば、全ての図形ごとに自由度を集めた大域的なベクトルが構成できる。

いま「頂点」が全部で n_V 個あり、「辺」が全部で n_E 個あり、「面」が n_F 個、「体」が n_B 個あるとする、つまり、「頂点」は V_1, V_2, \dots, V_{n_V} , 「辺」は E_1, E_2, \dots, E_{n_E} , 「面」は F_1, F_2, \dots, F_{n_F} , 「体」は B_1, B_2, \dots, B_{n_B} である。（ただし、ここではいま説明の簡単化のために、PDE の境界条件を入れる際の自由度の変化を省略して説明している。）「頂点」 V_i の自由度はどれも FEM 要素の基底の割り当てにはよらずに 1 である。「辺」 E_i はそれを共有する FEM 要素の間で基底の割当て方を調べてやり、「辺」方向の位数の最低値からその真の自由度が決まる。それを $f(E_i)$ とする。同様に「面」 F_i の真の自由度を $f(F_i)$, 「体」 B_i の自由度は、その「体」に対応する FEM 要素の基底の位数を (k_1, k_2, k_3) とすれば、真の自由度は $(k_1 - 2)(k_2 - 2)(k_3 - 2)$ で与えられる。それを $f(B_i)$ とする。すると、例えば「頂点」の真の自由度、「辺」の真の自由度、「面」の真の自由度、「体」の真の自由度の順に、しかもそれぞれの図形に付けた番号の順番に真の自由度を並べて書けば、 $f(V_1), f(V_2), \dots, f(V_{n_V}), f(E_1), f(E_2), \dots, f(E_{n_E}), f(F_1), f(F_2), \dots, f(F_{n_F}), f(B_1), f(B_2), \dots, f(B_{n_B})$ となる。これらの自由度の合計を N として、これらの真の自

由度に対応して係数を格納し、FEMの大域的な係数ベクトルを作るとその次元は N になる。適切なデータ構造を構築して管理することにより、例えば m 番目の「面」である F_m の真の自由度の値やその大域的な係数ベクトルの中での格納開始位置を知ることができて、該当する係数を取り出したり、あるいは書き込むことなどができる。(大域的ベクトルは物理的にはメモリ上の連続領域にかためて置いたりあるいは一台のマシン上に集中して置く必要は必ずしもなくて、並列分散処理向けに適切に分散して配置することができる。E-by-E法の分散並列処理をする場合に、メモリ空間上で離れたFEM要素に対応する係数のデータをあまり近接させると、係数のデータ更新の際にアクセスが集中して排他制御により待ちが生じて、並列化効率が落ちる可能性がある。たとえば「頂点」の係数をかためて配置するのは良くない可能性がある。)

6. Element-by-Element 法

大域的なベクトルに対して E-by-E 法で乗算を行なうときには、まず大域的なベクトル Y_{global} を零で初期化しておいてから、全ての要素 e のそれぞれに対して以下の三つのステップの処理を行なう。

STEP-1 要素 e に属するすべての「図形」(V, E, F, B) に対して、大域的なベクトル X_{global} の中から「図形」に対する格納位置を求めて、「図形」の真の自由度に対応する係数を読み出し、局所的なテンソル積型基底による係数のベクトル X_{local} を構成する。その際に、「図形の」真の自由度と対応していない冗長な「テンソル積型基底に対する係数」は零に設定する。

STEP-2 STEP-1 で構成した局所的なベクトル X_{local} に局所的な離散化行列 A_{local} を乗じることで、局所的なベクトル Y_{local} を作る。

STEP-3 要素 e に属するすべての「図形」(V, E, F, B) に対して、「図形」の真の自由度に対応する係数が大域的なベクトル Y_{global} の中で格納されている位置を求めて、その位置に STEP-2 で得たテンソル積型基底による係数のベクトル Y_{local} のうちで「図形」の真の自由度と対応する係数は加算するが、そうではない係数は単に無視する。

STEP-2 での局所的な離散化行列 A_{local} を局所的なベクトル X_{local} に乗じた結果の局所的なベクトル Y_{local} を作るここでは、本論文中で既に解説した高速計算の技法を用いることができる。また処理を並列実行している場合には、STEP-3 で Y_{global} への加算を行なうところで排他制御が必要である。

7. おわりに

現在あるいは将来の HPC システムを用いて大規模な問題を低コストで解くためには必要な記憶量や記憶の転送量を低減する計算法が極めて重要である。

線形偏微分方程式の境界値問題を有限要素法で離散化近似して得られる大規模な連立一次方程式を解く場合には、問題の領域が三次元以上の場合、消去法を用いて解くとその過程で大量のフィルインが発生して膨大な記憶量が必要になる。そこで連立一次方程式は反復解法を用いて解くとする。

反復解法における計算の主要部分は大規模な係数行列とベクトルの積であり、それは各要素ごとに小規模な離散化行列を小規模な展開係数ベクトルに乗じた結果を集めて合計する Element-by-Element 法で実現できる。Element-by-Element 法は分散並列処理に向けた方法である。その際に各反復のたびに新たに乗算を行なうために各要素について小規模な離散化行列を構成し直し、乗算が済んだら直ちに廃棄して記憶を解放するという方法を採用すれば、必要な作業記憶量は抑えられる。しかし反復のたびに全要素に対する小規模な離散化行列を構成すると必要な演算量が増えることが問題になる。

そこでいま有限要素法に用いる要素の形状を直方体に限定して、要素内での関数の展開基底に直方体要素の各辺に沿った座標の高次多項式のテンソル積(直積)を採用する。するとこの要素形状と基底の構造の特殊性を用いた主に密行列同士の行列-行列積(BLAS3)演算で構成された計算の技巧により、E-by-E法で必要な小規模な離散化行列をベクトルへ乗じる計算が離散化行列を具体的に構成することなしにできるので作業に必要な記憶量を減らせる。しかも演算量も少なくなり計算の局所性も高いので、E-by-E法の適用が容易になる。

今回紹介したFEMの高速計算の技法は、1990年頃までには既に知られていたものである。また曲線座標変換による曲がった平行四辺形や四面体を要素にする場合にも技法の適用は一応可能である。

参考文献

- [1] V. Sonnad, H. Murakami, S. Foresti, S. Hassanzadeh and B. Jiang: "The Equations of Elasticity and Their Solution by Finite Element Methods", in *Modern Techniques in Computational Chemistry: MOTECC-90*, Chap.22, pp.983-999, ESCOM Leiden, 1990.
- [2] V. Sonnad, H. Murakami, S. Foresti, S. Hassanzadeh and B. Jiang: "The Equations of Elasticity and Their Solution by Finite Element Methods", in *Modern Techniques in Computational Chemistry: MOTECC-91*, Chap.29, pp.1091-1108, ESCOM Leiden, 1991. これは上とほぼ同じもの。
- [3] V. Sonnad, R. Panda, B. Jiang, H. Murakami, S. Has-

- sanzadeh and S. Foresti: “The Equations of Fluid Flow and Their Solutions by Numerical Methods”, in *Modern Techniques in Computational Chemistry: MOTTECC-91*, Chap.28, pp.1055–1090, ESCOM Leiden, 1991.
- [4] Barna Szabo and Ivo Babuska: *Finite Element Analysis*, Wiley-Interscience, New York, 1991.
- [5] H. Murakami, V. Sonnad and E. Clementi: “A Three-Dimensional Finite Element Approach Towards Molecular SCF Computations”, *Int.J.Quantum.Chem.*, Vol.42, No.4 (1992), pp.785–817.
- [6] S. Foresti, S. Hassanzadeh, H. Murakami and V. Sonnad: “Parallel rapid operator for iterative finite element solvers on a shared memory machine”, *Parallel Computing*, Vol.19, No.1 (1993), pp.1–7.
- [7] Ch. Schwab: *p and hp-Finite Element Methods*, Oxford Univ. Press, 1999.
- [8] 千塚明, 土田英二, 日本計算工学会 (編): 「アダプティブ有限要素法」, 丸善 (計算力学レクチャーシリーズ), 2003.
- [9] V. Korbeev, U. Langer and L. S. Xanthis: “On fast domain decomposition solving procedures for *hp*-discretization of 3-D elliptic problems”, *Computational Methods in Applied Mathematics*, Vol.3, No.4, (2003), pp.536-559.
- [10] Szabo B., Rank E. and Duester A.: “The *p*-version of the finite element method”, Chap.5 in Vol.1 of *Encyclopedia of Computational Mechanics*, John Wiley & Sons, 2004.
- [11] Leszek Demkowicz: *Computing with hp-ADAPTIVE FINITE ELEMENTS: Vol.I*, Chapman and Hall/CRC, 2006.
- [12] Leszek Demkowicz: *Computing with hp-ADAPTIVE FINITE ELEMENTS: Vol.II*, Chapman and Hall/CRC, 2007.
- [13] George Karniadakis and Spencer Sherwin: *Spectral /Hp Element Methods for Computational Fluid Dynamics*, 2nd Ed., Oxford Univ. Press, 2013.