A Structure of FEM Matrix by Lagrange Basis Polynomials

Murakami Hiroshi
Tokyo Metropolitan University
murakami@tmu.ac.jp

ABSTRACT

When the formula of Lagrange basis polynomials (LBPs) and an algebraic identity of fractions are used in combination, a product of two different LBPs is re-expressed as a sum of two functions. When N is the number of basis polynomials for an element of FEM, the discretized matrix of an operator such as potential function may be dense and has $O(N^2)$ non-zero entries by its original definition, but it can be re-expressed in the combination of entries of reduced matrices which have only $O(N)$ values in total. Therefore, the amount of storage to represent the discretized operator would be reduced.

KEYWORDS

FEM, Matrix Element, Structure, Lagrange basis polynomials

1 INTRODUCTION

For modern high performance computer systems to solve problems of large sizes with low costs, it is important to reduce both amounts of storage requirement and storage transfer to overcome the bottle-neck of the performance. Thus it would be important to make research on solution methods which reduce them.

We try a kind of p-version FEM (Axelsson and Baker, 2001)(Babuška et al., 1981) to solve a simple boundary value problem of linear PDE with high accuracy. And we consider to apply a method which potentially reduces the amount of storage for that FEM problem. To solve the simultaneous linear equation or eigenvalue problem which comes from the FEM discretized boundary value problem of PDE, we assume some iterative solver is used whose computation kernel is the matrix-vector products with element-by-element FEM approach.

When Lagrange basis polynomials (LBPs) (for multi-dimensional case, tensor products of LBPs for each dimension (Weiwei and Zamani, 1989)) are used to construct local matrix of FEM for each FE, we can show that a simple algebraic property of LBPs is reflected to the structure of the local FEM element matrix. By an identity of fractions, a product of two different LBPs is re-expressed by a sum of two other polynomials. If the number of basis functions is N, the local FEM matrix which is a discretization of linear PDE operator by LBPs, is dense and has $O(N^2)$ non-zero entries in general. But from the property of LBPs (tensor products of LBPs for the multi-dimensional case), we can show that the local FEM matrix has a structure such that it can be constructed from only $O(N)$ values. Also, if we use this special structure, the arithmetic complexity for n can be reduced to make the matrix-vector multiplication of local FEM matrix and vector, which is used to solve the assembled global equation of FEM iteratively by the element-by-element approach.

. .

2 PDE PROBLEM BY FEM

In below, we consider a discretization of a simple PDE operator $(-\Delta + U(\mathbf{x}))$ by FEM. Here $U(\mathbf{x})$ is a potential function depends only on the spatial coordinate $\mathbf{x}$. For the simplicity, the region of the analysis is assumed rectangular whose sides are parallel to coordinate axes (Fig.1,Fig.2). We subdivide arbitrarily each edge of the region, and inside the FE made by the subdivision the unknown continuous function is approximated by a linear combination of LBPs. If the simultaneous linear equations of large size is solved by some iterative method whose coefficient matrix (global FEM matrix) is a collection of all small matrices (local matrices) discretized by FEM, then matrix-vector multiplication of the global FEM matrix and the global vector is repeated many times.

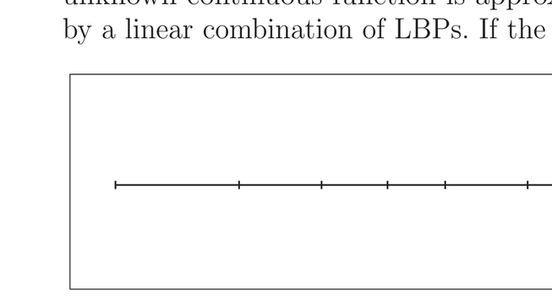

Figure 1: Conceptual FEM element subdivision(1-Dim)

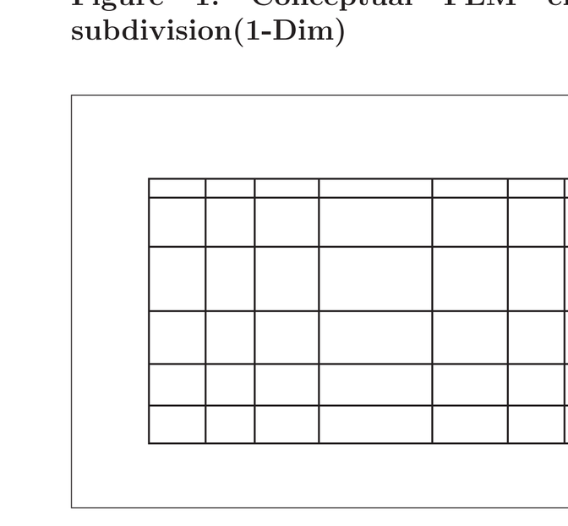

Figure 2: Conceptual FEM element subdivision(2-Dim)

3 STRUCTURE OF MATRIX ELEMENTS

The Lagrange interpolation formula inside an interval I at the n-distinct abscissas $\{\alpha_j \,|\, j = 1,\ldots,n\}$ gives the polynomial $\widetilde{f}(x)$ which approximates a function $f(x)$ as:

$$\widetilde{f}(x) = \sum_{j=1}^{n} \frac{P^{(\alpha)}(x)}{(x-\alpha_j)P^{(\alpha)\prime}(\alpha_j)} f(\alpha_j) \,.$$

Here, $P^{(\alpha)}(x)$ is a degree n polynomial corresponds to the set of n abscissas:

$$P^{(\alpha)}(x) \equiv \prod_{k=1}^{n} (x-\alpha_k) \,.$$

When the distribution of abscissas is selected well, the approximation of the original function f by the polynomial $\widetilde{f}$ would go better as the number of abscissas n increases.

To make the continuous connection of solution at the boundaries of adjacent local elements easily, the set of abscissas contains both ends of the interval I. (In this case, a distribution of abscissas with good property is the set of abscissas for Gauss-Lobatto quadrature formula, which is a closed quadrature formula. The abscissas of n points G-L quadrature formula over the standard interval $[-1,1]$ are, both-ends of the interval ± 1 and $(n-2)$ zeros of the derivative of the Legendre polynomial of $(n-1)$-th degree $P'_{n-1}(x)$, all of which are inside the standard interval.) The LBPs for the function expansion are n different degree $(n-1)$ polynomials (Fig.3). n is called the order of basis of the expansion.

The basis functions of expansion are the LBPs which are n different polynomials of degree $(n-1)$. To make the simple derivations of formulae, in the followings the factors $1/P^{(\alpha)\prime}(\alpha_j)$ are removed from definitions of the LBPs. Thus the basis polynomials are taken as:

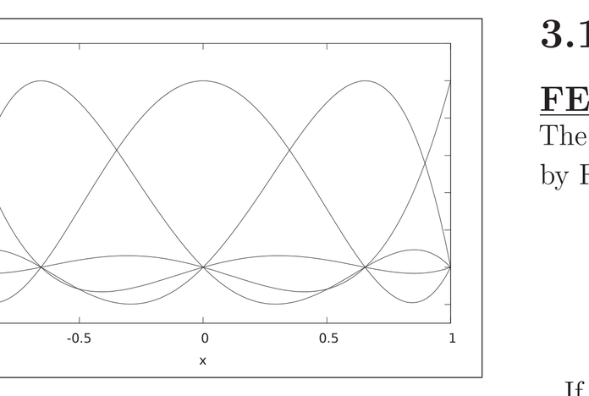

Figure 3: LBPs by 5-points G-L abscissas

$$\varphi_k^{(\alpha)}(x) \equiv \frac{P^{(\alpha)}(x)}{x-\alpha_k}, \quad \text{for } k = 1,\ldots,n \,.$$

If required, matrix elements based on the original LBPs can be obtained by scalings associated with the index k of the basis polynomials.

The essence of our method is as follows. By using an identity:

$$\frac{1}{x-a} \times \frac{1}{x-b} = \frac{1}{a-b}\left(\frac{1}{x-a} - \frac{1}{x-b}\right) \ ,(\text{if } a \neq b)\,,$$

the product of two different LBPs of degrees $n-1$ in a FE can be re-expressed as a sum of two different new functions:

$$\begin{aligned} & \varphi_i^{(\alpha)}(x)\,\varphi_j^{(\alpha)}(x) \\ = & \frac{P^{(\alpha)}(x)}{x-\alpha_i} \times \frac{P^{(\alpha)}(x)}{x-\alpha_j} \\ = & \frac{1}{\alpha_i-\alpha_j}\left[\frac{\{P^{(\alpha)}(x)\}^2}{x-\alpha_i} - \frac{\{P^{(\alpha)}(x)\}^2}{x-\alpha_j}\right] \\ = & \frac{1}{\alpha_i-\alpha_j}\left\{\Phi_i^{(\alpha)}(x) - \Phi_j^{(\alpha)}(x)\right\} . \end{aligned}$$

Here $\Phi_k^{(\alpha)}(x) \equiv \{P^{(\alpha)}(x)\}^2/(x-\alpha_k)$, $k = 1,2,\ldots,n$ are n different polynomials of degree $2n-1$.

3.1 Case of 1-Dimension

FEM Matrix of Potential

The local matrix of the potential $U(x)$ discretized by FEM using $\varphi_k^{(\alpha)}(x)$ for the basis functions is:

$$\begin{aligned} V_{i,j} &= \int_I \varphi_i^{(\alpha)}(x)\,U(x)\,\varphi_j^{(\alpha)}(x)dx \\ &= \int_I \frac{\{P^{(\alpha)}(x)\}^2}{(x-\alpha_i)(x-\alpha_j)}\,U(x)dx \,. \end{aligned}$$

If we define values:

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

then $V_{i,j}$ can be re-expressed as:

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

Thus the symmetric matrix $V_{i,j}$, a discretization of the potential $U(x)$, has a structure whose $n(n+1)/2$ entries can be constructed from only $2n$ values of $L_k^{(0)}$ and $L_k^{(1)}$ for $k = 1,2,\ldots,n$.

The integrand of the definition of $L^{(0)}$ has the degree higher by one than the integrand of the original definition of $V_{i,j}$, therefore to keep accuracies of the values of $L^{(0)}$, it seems that the quadrature formula of slightly higher accuracy have to be used. However, what finally to be calculated are the values of $V_{i,j}$, so $L^{(0)}$ may be calculated by some quadrature formula that is just sufficient to integrate $V_{i,j}$ by the original formula. The errors of $L^{(0)}$ which are introduced by this reduced integration will be canceled when $V_{i,j}$ is calculated from $L^{(0)}$.

FEM Matrix of Laplacian

In our situation (without curved coordinate transformation), the FEM matrix of Laplacian depends only on basis functions selected. Therefore, when the region of analysis is divided into FEs, and each local coordinate inside the FE is transformed linearly to the standard interval $[-1,1]$, then those FEs whose set of LBPs are the same have the same matrix of discretized Laplacian except the scaling factors. Therefore, when we use LBPs whose abscissas are for example G-L type, if the types of basis functions of the FEs are not so many, we can reduce the amount of storage by keep FEM matrices of the Laplacian for the standard interval correspondingly, and use them selectively with proper scaling depending on the size of the FE.

We can reduce the amount of the storage for those FE whose order of basis function n is large. The matrix elements of the Laplacian of 1-D are written as:

$$\begin{aligned} K_{i,j}^{(\alpha)} &= \int_I \left\{\frac{d}{dx}\varphi_i^{(\alpha)}(x)\cdot\frac{d}{dx}\varphi_j^{(\alpha)}(x)\right\} dx \\ &= \int_I \left\{\frac{d}{dx}\left(\frac{P^{(\alpha)}(x)}{x-\alpha_i}\right)\cdot\frac{d}{dx}\left(\frac{P^{(\alpha)}(x)}{x-\alpha_j}\right)\right\} dx \\ &= \int_I \left\{\frac{\frac{d}{dx}P^{(\alpha)}(x)}{x-\alpha_i} - \frac{P^{(\alpha)}(x)}{(x-\alpha_i)^2}\right\}\left\{\frac{\frac{d}{dx}P^{(\alpha)}(x)}{x-\alpha_j} - \frac{P^{(\alpha)}(x)}{(x-\alpha_j)^2}\right\} dx\,. \end{aligned}$$

Then we have:

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

Here, we have defined:

$$\begin{cases} A_k^{(1)} &\equiv \displaystyle\int_I \left\{\frac{d}{dx}\varphi_k^{(\alpha)}(x)\cdot\frac{d}{dx}P^{(\alpha)}(x)\right\} dx, \\ A_k^{(2)} &\equiv \displaystyle\int_I \{\varphi_k^{(\alpha)}(x)\}^2 dx, \\ A_k^{(3)} &\equiv \displaystyle -2\int_I \{\varphi_k^{(\alpha)}(x)P^{(\alpha)}(x)\} dx, \\ B_k &\equiv \displaystyle\int_I \left\{\frac{d}{dx}\varphi_k^{(\alpha)}(x)\right\}^2 dx\,. \end{cases}$$

Therefore, $n(n+1)/2$ elements of the symmetric matrix of discretized Laplacian $K_{i,j}^{(\alpha)}$ can be built from $4n$ values of $A_k^{(1)}$, $A_k^{(2)}$, $A_k^{(3)}$ and B_k for $k = 1,\ldots,n$.

If we define the anti-symmetric matrix $C_{i,j}^{(\alpha)}$ as

$$C_{i,j}^{(\alpha)} \equiv \begin{cases} \dfrac{1}{\alpha_i-\alpha_j}, & (\text{if } i \neq j), \\ 0, & (\text{if } i = j), \end{cases}$$

Then we have:

$$\begin{aligned} K_{i,j}^{(\alpha)} \ = \ & (C_{i,j}^{(\alpha)})^3(A_i^{(3)} - A_j^{(3)}) + (C_{i,j}^{(\alpha)})^2(A_i^{(2)} + A_j^{(2)}) \\ & + (C_{i,j}^{(\alpha)})(A_i^{(1)} - A_j^{(1)}) + \delta_{i,j}B_i\,. \end{aligned}$$

(here $\delta_{i,j}$ is the Kronecker's symbol)

Therefore, to calculate the vector $\mathbf{w}$ which is a matrix-vector multiplication of the matrix $K^{(\alpha)}$ and the vector $\mathbf{u}$, we need not store $C_{i,j}^{(\alpha)}$ and calculates:

$$\begin{aligned} w_i \ = \ & \sum_j K_{i,j}^{(\alpha)} u_j \\ = \ & A_i^{(3)} \sum_j (C_{i,j}^{(\alpha)})^3 u_j - \sum_j (C_{i,j}^{(\alpha)})^3 (A_j^{(3)} u_j) \\ & + A_i^{(2)} \sum_j (C_{i,j}^{(\alpha)})^2 u_j + \sum_j (C_{i,j}^{(\alpha)})^2 (A_j^{(2)} u_j) \\ & + A_i^{(1)} \sum_j C_{i,j}^{(\alpha)} u_j - \sum_j C_{i,j}^{(\alpha)} (A_j^{(1)} u_j) \\ & + B_i u_i\,. \end{aligned}$$

If we set auxiliary values:

$$\begin{cases} p_j^{(3)} \equiv A_j^{(3)} u_j, \quad q_i^{(3)} \equiv \sum_j (C_{i,j}^{(\alpha)})^3 u_j, \\ p_j^{(2)} \equiv A_j^{(2)} u_j, \quad q_i^{(2)} \equiv \sum_j (C_{i,j}^{(\alpha)})^2 u_j, \\ p_j^{(1)} \equiv A_j^{(1)} u_j, \quad q_i^{(1)} \equiv \sum_j C_{i,j}^{(\alpha)} u_j, \end{cases}$$

and calculate the vector $\mathbf{w}$ efficiently:

$$\begin{aligned} w_i \ = \ & \sum_j \left\{ -(C_{i,j}^{(\alpha)})^3 p_j^{(3)} \ + \ (C_{i,j}^{(\alpha)})^2 p_j^{(2)} \ - \ C_{i,j}^{(\alpha)} p_j^{(1)} \right\} \\ & + A_i^{(3)} q_i^{(3)} \ + \ A_i^{(2)} q_i^{(2)} \ + \ A_i^{(1)} q_i^{(1)} \\ & + B_i u_i\,. \end{aligned}$$

For this case of 1-D, by this method the number of arithmetic operations is $O(n^2)$, therefore it has the same complexity to matrix-vector multiplication of formed matrix $K^{(\alpha)}$ and the vector $\mathbf{u}$. However, if n is very large, this method does not store nor hold the matrix $K^{(\alpha)}$ explicitly, therefore the amount of storage requirement to make the calculation is less as long as the matrix $C_{i,j}^{(\alpha)}$ is not formed nor stored.

3.2 Case of 2-Dimension

Tensorial Product Base Functions

When the local FEM element is parameterized by a direct product of the local coordinates, the basis functions are taken as tensorial products of LBPs in each coordinate as:

$$\varphi_{\mathbf{k}}^{(\alpha,\beta)}(x,y) = \varphi_{(k_1,k_2)}^{(\alpha,\beta)}(x,y) \equiv \varphi_{k_1}^{(\alpha)}(x)\,\varphi_{k_2}^{(\beta)}(y)\,,$$

where

$$\begin{cases} \varphi_{k_1}^{(\alpha)}(x) \equiv \dfrac{P^{(\alpha)}(x)}{x-\alpha_{k_1}}, \quad P^{(\alpha)}(x) \equiv \displaystyle\prod_{k=1}^{n_1}(x-\alpha_k)\,, \\ \varphi_{k_2}^{(\beta)}(y) \equiv \dfrac{P^{(\beta)}(y)}{y-\beta_{k_2}}, \quad P^{(\beta)}(y) \equiv \displaystyle\prod_{k=1}^{n_2}(y-\beta_k)\,. \end{cases}$$

FEM Matrix of Potential

Matrix elements of 2-D potential $U(x,y)$ are:

$$\begin{aligned} & V_{\mathbf{i},\mathbf{j}} = V_{(i_1,i_2),(j_1,j_2)} \\ & = \iint_{area} \varphi_{i_1}^{(\alpha)}(x)\,\varphi_{i_2}^{(\beta)}(y)\,U(x,y)\varphi_{j_1}^{(\alpha)}(x)\,\varphi_{j_2}^{(\beta)}(y)dxdy \\ & = \iint_{area} \frac{\{P^{(\alpha)}(x)\}^2\,\{P^{(\beta)}(y)\}^2\,U(x,y)}{(x-\alpha_{i_1})(x-\alpha_{j_1})\,(y-\beta_{i_2})(y-\beta_{j_2})}dxdy. \end{aligned}$$

If we define:

$$L_{i,j}^{(e_1,e_2)} = \iint_{area} \frac{\{P^{(\alpha)}(x)\}^2}{(x-\alpha_i)^{1+e_1}}\,\frac{\{P^{(\beta)}(y)\}^2}{(y-\beta_j)^{1+e_2}}U(x,y)dxdy\,,$$

here $e_1 = 0$ or 1 and $e_2 = 0$ or 1.

Then matrix elements of the potential can be written by combinations of values of entries of $L^{(e_1,e_2)}$ as:

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

Therefore, by the above algebraic relations, all $n^2(n^2+1)/2$ entries of the symmetric matrix $V_{(i_1,i_2),(j_1,j_2)}$ can be built from only $4n^2$ values of $L_{i,j}^{(e_1,e_2)}$. The storage requirement can be reduced if this knowledge on the structure of the matrix elements is used.

The algebraic structure of this matrix elements is derived from only the simple identity of the fractions and the use of the LBPs.

FEM Matrix of Laplacian

Galerkin discretization of linear partial differential operator can be made easy and faster for two or more dimensions when the operator has only constant coefficients. Since we do not use the curved coordinate system, the matrices for each dimension

$$\begin{cases} M_{i_1,j_1}^{(\alpha)} = \displaystyle\int_{I_x} \varphi_{i_1}^{(\alpha)}(x)\varphi_{j_1}^{(\alpha)}(x)\,dx, \\ K_{i_1,j_1}^{(\alpha)} = \displaystyle\int_{I_x} \{\frac{d}{dx}\varphi_{i_1}^{(\alpha)}(x)\cdot\frac{d}{dx}\varphi_{j_1}^{(\alpha)}(x)\}\,dx, \end{cases}$$

and

$$\begin{cases} M_{i_2,j_2}^{(\beta)} = \displaystyle\int_{I_y} \varphi_{i_2}^{(\beta)}(y)\varphi_{j_2}^{(\beta)}(y)\,dy, \\ K_{i_2,j_2}^{(\beta)} = \displaystyle\int_{I_y} \{\frac{d}{dy}\varphi_{i_2}^{(\beta)}(y)\cdot\frac{d}{dy}\varphi_{j_2}^{(\beta)}(y)\}\,dy, \end{cases}$$

are prepared, then the matrix of FEM discretized Laplacian in 2-dim $-\Delta = -(\nabla_x^2 + \nabla_y^2)$ is:

$$K_{\mathbf{i},\mathbf{j}}^{(\alpha,\beta)} = K_{(i_1,i_2),(j_1,j_2)}^{(\alpha,\beta)} = K_{i_1,j_1}^{(\alpha)}\cdot M_{i_2,j_2}^{(\beta)} + M_{i_1,j_1}^{(\alpha)}\cdot K_{i_2,j_2}^{(\beta)}\,.$$

If we use this structure, we can make the fast multiplication of the matrix $K^{(\alpha,\beta)}$ and a vector. When linear scaling factors from element widths are ignored, types of matrices $M^{(\alpha)}$ and $K^{(\alpha)}$ in x-direction are types of basis functions inside FEs in x-direction, and types of matrices $M^{(\beta)}$ and $K^{(\beta)}$ in y-direction are types of basis functions inside FEs in y-direction. So, if the types of basis functions of each directions in FEs are not so many, we can calculate and keep a set of small number of matrices K and M for 1-D for the standard interval for each direction correspond to the types of basis functions inside elements with not so much amount of storage requirement, and use the stored 1-D matrices selectively with proper scalings.

3.3 Case of 3-Dimension

Tensorial Product Base Functions

Similar to the 2-D case, tensorial products of LBPs in each local coordinate are taken as basis functions:

$$\begin{aligned} & \varphi_{\mathbf{k}}^{(\alpha,\beta,\gamma)}(x,y,z) = \varphi_{(k_1,k_2,k_3)}^{(\alpha,\beta,\gamma)}(x,y,z) \\ \equiv \ & \varphi_{k_1}^{(\alpha)}(x)\,\varphi_{k_2}^{(\beta)}(y)\,\varphi_{k_3}^{(\gamma)}(z) \end{aligned}$$

$$\begin{cases} \varphi_{k_1}^{(\alpha)}(x) \equiv \dfrac{P^{(\alpha)}(x)}{x-\alpha_{k_1}}, \quad P^{(\alpha)}(x) \equiv \displaystyle\prod_{k=1}^{n_1}(x-\alpha_k)\,, \\ \varphi_{k_2}^{(\beta)}(y) \equiv \dfrac{P^{(\beta)}(y)}{y-\beta_{k_2}}, \quad P^{(\beta)}(y) \equiv \displaystyle\prod_{k=1}^{n_2}(x-\beta_k)\,, \\ \varphi_{k_3}^{(\gamma)}(z) \equiv \dfrac{P^{(\gamma)}(z)}{z-\gamma_{k_3}}, \quad P^{(\gamma)}(z) \equiv \displaystyle\prod_{k=1}^{n_3}(z-\gamma_k)\,. \end{cases}$$

FEM Matrix of Potential

The matrix of 3-D potential $U(x,y,z)$ is

$$\begin{aligned} & V_{\mathbf{i},\mathbf{j}} = V_{(i_1,i_2,i_3),(j_1,j_2,j_3)} \\ = & \iiint_{vol} \varphi_{i_1}^{(\alpha)}(x)\,\varphi_{i_2}^{(\beta)}(y)\,\varphi_{i_3}^{(\gamma)}(z)\,U(x,y,z) \times \\ & \qquad \varphi_{j_1}^{(\alpha)}(x)\,\varphi_{j_2}^{(\beta)}(y)\,\varphi_{j_3}^{(\gamma)}(z)\,dxdydz \\ = & \iiint_{vol} \frac{\{P^{(\alpha)}(x)\}^2}{(x-\alpha_{i_1})(x-\alpha_{j_1})}\,\frac{\{P^{(\beta)}(y)\}^2}{(y-\beta_{i_2})(y-\beta_{j_2})} \times \\ & \qquad \frac{\{P^{(\gamma)}(z)\}^2}{(z-\gamma_{i_3})(z-\gamma_{j_3})}\,U(x,y,z)\,dxdydz\,. \end{aligned}$$

If we define the reduced matrices:

$$L_{i,j,k}^{(e_1,e_2,e_3)} = \iiint_{vol} \frac{\{P^{(\alpha)}(x)\}^2}{(x-\alpha_i)^{1+e_1}} \times \frac{\{P^{(\beta)}(y)\}^2}{(y-\beta_j)^{1+e_2}} \times \frac{\{P^{(\gamma)}(z)\}^2}{(z-\gamma_k)^{1+e_3}}\,U(x,y,z)\,dxdydz\,,$$

then the matrix elements can be expressed by combinations of matrix elements of reduced matrices:

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

Therefore, the symmetric matrix $V_{(i_1,i_2,i_3),(j_1,j_2,j_3)}$, which has $n^3(n^3+1)/2$ entries, can be built from only $8n^3$ values of $L_{i,j,k}^{(e_1,e_2,e_3)}$ by using the above algebraic relations.

FEM Matrix of Laplacian

Similar to the 2-D case, If we have prepared 1-D matrices for each direction:

$$\begin{cases} M_{i_1,j_1}^{(\alpha)} = \displaystyle\int_{I_x} \varphi_{i_1}^{(\alpha)}(x)\varphi_{j_1}^{(\alpha)}(x)\,dx, \\ K_{i_1,j_1}^{(\alpha)} = \displaystyle\int_{I_x} \{\frac{d}{dx}\varphi_{i_1}^{(\alpha)}(x)\cdot\frac{d}{dx}\varphi_{j_1}^{(\alpha)}(x)\}\,dx, \end{cases}$$

and

$$\begin{cases} M_{i_2,j_2}^{(\beta)} = \displaystyle\int_{I_y} \varphi_{i_2}^{(\beta)}(y)\varphi_{j_2}^{(\beta)}(y)\,dy, \\ K_{i_2,j_2}^{(\beta)} = \displaystyle\int_{I_y} \{\frac{d}{dy}\varphi_{i_2}^{(\beta)}(y)\cdot\frac{d}{dy}\varphi_{j_2}^{(\beta)}(y)\}\,dy, \end{cases}$$

and

$$\begin{cases} M_{i_3,j_3}^{(\gamma)} = \displaystyle\int_{I_z} \varphi_{i_3}^{(\gamma)}(z)\varphi_{j_3}^{(\gamma)}(z)\,dz, \\ K_{i_3,j_3}^{(\gamma)} = \displaystyle\int_{I_z} \{\frac{d}{dz}\varphi_{i_3}^{(\gamma)}(z)\cdot\frac{d}{dz}\varphi_{j_3}^{(\gamma)}(z)\}\,dz, \end{cases}$$

then the FEM discretized matrix of 3-D Laplacian $-\Delta = -(\nabla_x^2 + \nabla_y^2 + \nabla_z^2)$ by tensorial basis functions is:

$$\begin{aligned} K_{\mathbf{i},\mathbf{j}}^{(\alpha,\beta,\gamma)} \ = \ & K_{(i_1,i_2,i_3),(j_1,j_2,j_3)}^{(\alpha,\beta,\gamma)} \\ = \ & K_{i_1,j_1}^{(\alpha)}\cdot M_{i_2,j_2}^{(\beta)}\cdot M_{i_3,j_3}^{(\gamma)} \\ & + M_{i_1,j_1}^{(\alpha)}\cdot K_{i_2,j_2}^{(\beta)}\cdot M_{i_3,j_3}^{(\gamma)} \\ & + M_{i_1,j_1}^{(\alpha)}\cdot M_{i_2,j_2}^{(\beta)}\cdot K_{i_3,j_3}^{(\gamma)}\,. \end{aligned}$$

By using this structure, fast multiplication of matrix $K^{(\alpha,\beta,\gamma)}$ and a vector is available.

When there are not so many types of the 1-D basis functions, we can prepare and store 1-D matrices for each direction $M^{(\alpha)}$, $K^{(\alpha)}$, $M^{(\beta)}$,$K^{(\beta)}$, $M^{(\gamma)}$, $K^{(\gamma)}$, and use them selectively with scalings.

4 FAST MAT-VEC MULTIPLICATION

In the previous section, structures of FEM matrix are shown and reduced matrices are introduced.

By the use of the structure, the number of quadratures would be reduced to obtain the whole matrix elements (of the potential function), especially when accurate quadratures are difficult and many abscissas are used.

The knowledge on the structure also could be used to reduce the amount of storage requirement. For example, when there are many FEM elements of higher order, and the assembled global linear equation of FEM has to be solved by some iterative method, and many matrix-vector multiplications need to be made. If we store matrix compactly in the reduced form by the knowledge of the structure, the amount of storage transfer is reduced than we store original matrix explicitly.

Moreover, if the reduced matrix elements are stored and used, the matrix-vector multiplication can be made faster than by the trivial method which uses explicitly formed elements of the original matrix.

Below, we define an anti-symmetric matrix C:

$$C_{i,j} \equiv \begin{cases} 1/(\alpha_i-\alpha_j) & ,(\text{if } i \neq j), \\ 0 & ,(\text{if } i = j). \end{cases}$$

Elements of matrix C can be calculated easily on demand from a set of n values of α_k, so it is not necessary to form the matrix C explicitly.

The multiplication of the matrix C to a given vector is known as *Trummer's problem*, and a fast algorithm is known whose arithmetic complexity is $O(n(\log n)^2)$(Gerasoulis et al., 1987). Theoretically very interesting, we do NOT use the fast algorithm for *Trummer's problem* since it may not be fast for the actual size of our application and also the numerical behavior is not clear.

4.1 Case of 1-Dimension

Here, we show the calculation of the product of the potential matrix $V_{i,j}$ to a given vector $\mathbf{u}$ to make a vector $\mathbf{v}$.

Since

$$V_{i,j} = \begin{cases} C_{i,j}^{(\alpha)} \times \left(L_i^{(0)} - L_j^{(0)}\right) & ,(\text{if } i \neq j), \\ L_i^{(1)} & ,(\text{if } i = j)\,, \end{cases}$$

by the definition $C_{k,k}^{(\alpha)} = 0$, the matrix-vector product $\mathbf{v}$ can be calculated as:

$$\begin{aligned} v_i \ &= \ \sum_j V_{i,j}\,u_j \\ &= \left(\sum_{j(\neq i)} V_{i,j}\,u_j\right) + V_{i,i}\,u_i \\ &= \left\{\sum_j C_{i,j}^{(\alpha)} \times (L_i^{(0)} - L_j^{(0)})\,u_j\right\} + L_i^{(1)}\,u_i \\ &= \left(\sum_j C_{i,j}^{(\alpha)} L_i^{(0)}\,u_j\right) - \left(\sum_j C_{i,j}^{(\alpha)} L_j^{(0)}\,u_j\right) + L_i^{(1)}\,u_i\,. \end{aligned}$$

- The first term of the above expression is calculated as below in $O(n^2)$ arithmetic operations.

$$\sum_j C_{i,j}^{(\alpha)} L_i^{(0)} u_j = L_i^{(0)}\left(\sum_j C_{i,j}^{(\alpha)} u_j\right) = L_i^{(0)} a_i\,.$$

- The second term is calculated as below, in $O(n^2)$ arithmetic operations.

$$\sum_j C_{i,j}^{(\alpha)}\left(L_j^{(0)} u_j\right) = \sum_j C_{i,j}^{(\alpha)} b_j\,.$$

- The third term $L_i^{(1)} u_i$ is calculated in $O(n)$ arithmetic operations.

In total, matrix-vector product of the potential matrix can be made $O(n^2)$ arithmetic operations in total, which is the same complexity when the matrix V is formed and multiplied. If $C^{(\alpha)}$ is not formed nor stored, the storage requirement of the above process is only $O(n)$.

4.2 Case of 2-Dimension

Potential Term

The matrix elements $V_{\mathbf{i},\mathbf{j}}$ of the 2-D potential is written by using $C^{(\alpha)}$ and $C^{(\beta)}$ which have zeros in diagonals as:

$$\begin{aligned} V_{\mathbf{i},\mathbf{j}} \ = \ & C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} (L_{i_1,i_2}^{(0,0)} - L_{i_1,j_2}^{(0,0)} - L_{j_1,i_2}^{(0,0)} + L_{j_1,j_2}^{(0,0)}) \\ & + \delta_{i_1,j_1} C_{i_2,j_2}^{(\beta)} (L_{i_1,i_2}^{(1,0)} - L_{j_1,j_2}^{(1,0)}) \\ & + C_{i_1,j_1}^{(\alpha)} \delta_{i_2,j_2} (L_{i_1,i_2}^{(0,1)} - L_{j_1,j_2}^{(0,1)}) \\ & + \delta_{i_1,j_1} \delta_{i_2,j_2} L_{i_1,i_2}^{(1,1)} \end{aligned}$$

Here we will some part of the calculation of the matrix-vector product of the matrix $V_{\mathbf{i},\mathbf{j}}$ and a vector $\mathbf{u_j}$.

- The calculation of the term below (when $L^{(0,0)}$ has no index of j) requires $O(n^3)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} L_{i_1,i_2}^{(0,0)} u_{j_1,j_2} \\ = \ & L_{i_1,i_2}^{(0,0} \left(\sum_{j_1} C_{i_1,j_1}^{(\alpha)} \left(\sum_{j_2} C_{i_2,j_2}^{(\beta)} u_{j_1,j_2}\right)\right)\,. \end{aligned}$$

- The calculation of the term below (when $L^{(0,0)}$ has an index j_2) requires $O(n^3)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} L_{i_1,j_2}^{(0,0)} u_{j_1,j_2} \\ = \ & \sum_{j_2} C_{i_2,j_2}^{(\beta)} L_{i_1,j_2}^{(0,0)} \left(\sum_{j_1} C_{i_1,j_1}^{(\alpha)} u_{j_1,j_2}\right) \\ = \ & \sum_{j_2} C_{i_2,j_2}^{(\beta)} \left(L_{i_1,j_2}^{(0,0)} b_{i_1,j_2}\right) \\ = \ & \sum_{j_2} C_{i_2,j_2}^{(\beta)} c_{i_1,j_2}\,. \end{aligned}$$

- The calculation of the term below (when $L^{(0,0)}$ has an index j_1) requires $O(n^3)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} L_{j_1,i_2}^{(0,0)} u_{j_1,j_2} \\ = \ & \sum_{j_1} C_{i_1,j_1}^{(\alpha)} L_{j_1,i_2}^{(0,0)} \left(\sum_{j_2} C_{i_2,j_2}^{(\beta)} u_{j_1,j_2}\right) \\ = \ & \sum_{j_1} C_{i_1,j_1}^{(\alpha)} \left(L_{j_1,i_2}^{(0,0)} \widetilde{b}_{j_1,i_2}\right) \\ = \ & \sum_{j_1} C_{i_1,j_1}^{(\alpha)} \widetilde{c}_{j_1,i_2}\,. \end{aligned}$$

- The calculation of the term below (when $L^{(0,0)}$ has induces j_1 and j_2) requires $O(n^3)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} \left(L_{j_1,j_2}^{(0,0)} u_{j_1,j_2}\right) \\ = \ & \sum_{j_1,j_2} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} d_{j_1,j_2} \\ = \ & \sum_{j_1} C_{i_1,j_1}^{(\alpha)} \left(\sum_{j_2} C_{i_2,j_2}^{(\beta)} d_{j_1,j_2}\right)\,. \end{aligned}$$

- (We omit to show other calculations of terms.)

The number of arithmetic operations is $O(n^3)$ in total (If the fast algorithm for Trummer's problem is used, it is $O(n^2(\log n)^2)$). If V is formed explicitly, it has already about $n^4/2$ entries, and the matrix-vector product requires $O(n^4)$ arithmetic operations.

Laplacian Term

The multiplication of the FEM discretized 2-D Laplacian matrix

$$K_{\mathbf{i},\mathbf{j}}^{(\alpha,\beta)} = K_{(i_1,i_2),(j_1,j_2)}^{(\alpha,\beta)}$$

to the vector $\mathbf{v_j} = \mathbf{v}_{(j_1,j_2)}$, is made by the use the matrix structure:

$$K_{(i_1,i_2),(j_1,j_2)}^{(\alpha,\beta)} = K_{i_1,j_1}^{(\alpha)} M_{i_2,j_2}^{(\beta)} + M_{i_1,j_1}^{(\alpha)} K_{i_2,j_2}^{(\beta)}\,.$$

The number of arithmetic operations is $O(n^3)$.

4.3 Case of 3-Dimension

Potential Term

Here also shows only a part of the calculation of a matrix-vector product of $V_{\mathbf{i},\mathbf{j}}$ and $\mathbf{u_j}$.

- The calculation of the term below (when $L^{(0,0,0)}$ has no index with j) requires $O(n^4)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2,j_3} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} L_{i_1,i_2,i_3}^{(0,0,0)} u_{j_1,j_2,j_3} \\ = \ & L_{i_1,i_2,i_3}^{(0,0,0)} (\sum_{j_1} C_{i_1,j_1}^{(\alpha)} (\sum_{j_2} C_{i_2,j_2}^{(\beta)} (\sum_{j_3} C_{i_3,j_3}^{(\gamma)} u_{j_1,j_2,j_3})))\,. \end{aligned}$$

- The calculation of the term below (when $L^{(0,0,0)}$ has one index with j) requires $O(n^4)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2,j_3} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} L_{i_1,i_2,j_3}^{(0,0,0)} u_{j_1,j_2,j_3} \\ = \ & \sum_{j_2,j_3} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} L_{i_1,i_2,j_3}^{(0,0,0)} \left(\sum_{j_1} C_{i_1,j_1}^{(\alpha)} u_{j_1,j_2,j_3}\right) \\ = \ & \sum_{j_2,j_3} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} L_{i_1,i_2,j_3}^{(0,0,0)} b_{i_1,j_2,j_3} \\ = \ & \sum_{j_3} C_{i_3,j_3}^{(\gamma)} L_{i_1,i_2,j_3}^{(0,0,0)} \left(\sum_{j_2} C_{i_2,j_2}^{(\beta)} b_{i_1,j_2,j_3}\right) \\ = \ & \sum_{j_3} C_{i_3,j_3}^{(\gamma)} \left(L_{i_1,i_2,j_3}^{(0,0,0)} c_{i_1,i_2,j_3}\right) \\ = \ & \sum_{j_3} C_{i_3,j_3}^{(\gamma)} d_{i_1,i_2,j_3}\,. \end{aligned}$$

- The calculation of the term below (when $L^{(0,0,0)}$ has two induces with j) requires $O(n^4)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2,j_3} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} L_{i_1,j_2,j_3} u_{j_1,j_2,j_3} \\ = \ & \sum_{j_2,j_3} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} L_{i_1,j_2,j_3}^{(0,0,0)} \left(\sum_{j_1} C_{i_1,j_1}^{(\alpha)} u_{j_1,j_2,j_3}\right) \\ = \ & \sum_{j_2,j_3} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} \left(L_{i_1,j_2,j_3}^{(0,0,0)} e_{i_1,j_2,j_3}\right) \\ = \ & \sum_{j_2,j_3} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} f_{i_1,j_2,j_3} \\ = \ & \sum_{j_2} C_{i_2,j_2}^{(\beta)} \left(\sum_{j_3} C_{i_3,j_3}^{(\gamma)} f_{i_1,j_2,j_3}\right) \\ = \ & \sum_{j_2} C_{i_2,j_2}^{(\beta)} g_{i_1,j_2,i_3}\,. \end{aligned}$$

- The calculation of the term below (when $L^{(0,0,0)}$ has three induces with j) requires $O(n^4)$ arithmetic operations.

$$\begin{aligned} & \sum_{j_1,j_2,j_3} C_{i_1,j_1}^{(\alpha)} C_{i_2,j_2}^{(\beta)} C_{i_3,j_3}^{(\gamma)} \left(L_{j_1,j_2,j_3}^{(0,0,0)} u_{j_1,j_2,j_3}\right) \\ = \ & \left(\sum_{j_1} C_{i_1,j_1}^{(\alpha)} \left(\sum_{j_2} C_{i_2,j_2}^{(\beta)} \left(\sum_{j_3} C_{i_3,j_3}^{(\gamma)} h_{j_1,j_2,j_3}\right)\right)\right)\,. \end{aligned}$$

- (We omit to show other calculations of terms.)

Laplacian Term

For the mat-vec multiplication of the FEM discretized 3-D Laplacian $K_{\mathbf{i},\mathbf{j}}^{(\alpha,\beta,\gamma)} = K_{(i_1,i_2,i_3),(j_1,j_2,j_3)}^{(\alpha,\beta,\gamma)}$ and a vector $\mathbf{u_j} = \mathbf{u}_{j_1,j_2,j_3}$, when the structure of matrix elements $K_{(i_1,i_2,i_3),(j_1,j_2,j_3)}^{(\alpha,\beta,\gamma)} = K_{i_1,j_1}^{(\alpha)} M_{i_2,j_2}^{(\beta)} M_{i_3,j_3}^{(\gamma)} + M_{i_1,j_1}^{(\alpha)} K_{i_2,j_2}^{(\beta)} M_{i_3,j_3}^{(\gamma)} + M_{i_1,j_1}^{(\alpha)} M_{i_2,j_2}^{(\beta)} K_{i_3,j_3}^{(\gamma)}$ is used, (when the number of basis functions for each direction is $O(n)$), the calculation reduces to the multiplications with transpose and sums of matrices of size $O(n)$ and $O(n)$ by $O(n^2)$ matrices, and the total number of arithmetic operations is $O(n^4)$ by the classical matrix multiplication method.

5 CONCLUSION

We made a research on a technique which reduces amounts of storage requirement and data transfer, to overcome the bottle-neck of performance of modern high speed computer systems.

We have shown algebraic structures of local FEM matrix of operator (like potential function) which is discretized by using LBPs as the FEM expansion basis functions.

When N is the number of (tensorial) basis polynomials in an element of FEM, the discretized matrix of an operator such as potential function may be dense and has $O(N^2)$ non-zero entries by its original definition, but those entries of the matrix can be expressed by combinations of entries of reduced matrices which have only $O(N)$ values in total. Therefore, the amount of storage to hold the discretized operator could be reduced. Also the number of numerical integration can be reduced since only reduced matrices have to be calculated by integrations.

When the multiplication of the assembled global FEM matrix to a vector is made by the element-by-element approach to solve global linear equation by some iterative manner, we have shown that the arithmetic complexity of the matrix-vector multiplication in a local FE is reduced when reduced matrices are used than when the local matrix in explicit form is used.

REFERENCES

O. Axelsson and V. A. Baker. 2001. *Finite Element Solution of Boundary Value Problems.* SIAM. https://doi.org/10.1137/1.9780898719253

I. Babuška, B. A. Szabó, and I. N. Katz. 1981. The P Version of the Finite Element Method. *SIAM J. Numer. Anal.* 18, 3 (1981), 515–545. https://doi.org/10.1137/0718033

A. Gerasoulis, M.D. Grigoriadis, and Liping Sun. 1987. A Fast Algorithm for Trummer's Problem. *SIAM J. Sci. Stat. Comput.* 8, 1 (Jan. 1987), s135–s138. https://doi.org/10.1137/0908017

S. Weiwei and N. G. Zamani. 1989. A Fast Algorithm for Solving the Tensor Product Collocation Equations. *J. Franklin Inst.* 326, 2 (1989), 295–307. https://doi.org/10.1016/0016-0032(89)90076-8