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 and an algebraic identity of fractions are used in combination, a product of two different Lagrange basis polynomials can be re-expressed as a sum of two functions. The matrix which is a discretization of an operator such as potential function is dense and has $O(N^2)$ non-zero entries in general, here N is the number of basis polynomials for an element of FEM. But, entries of the matrix can be re-expressed by simple combinations of entries of reduced matrices which have only $O(N)$ entries in total, and we can reduce the amount of storage to hold the discretized matrix of the operator.

KEYWORDS

FEM, Matrix Element, Structure, Lagrange basis polynomials

ACM Reference Format:
Murakami Hiroshi. 2018. A Structure of FEM Matrix by Lagrange Basis Polynomials. In *Proceedings of HPC Conference (HPCAsia2018).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

When we solve large size problems with low costs by using modern high performance computer systems, it is important to reduce both amounts of storage requirement and storage transfer to overcome the bottle-neck of the performance. Thus in this paper we research on methods to reduce them.

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

When Lagrange basis polynomials (for multi-dimensional case, tensor products of basis polynomials for each dimension (Weiwei and Zamani, 1989)) are used to construct local matrix for each finite element, we can show that a simple algebraic property of Lagrange basis polynomials is reflected to structures of local matrices. From a simple mathematical identity of fractions, a product of two different Lagrange basis polynomials can be re-expressed as a sum of two other polynomials.

If the number of basis functions is N, then the local matrix which is obtained by FEM discretization of linear PDE operator by Lagrange basis polynomials, is dense in general and has $O(N^2)$ non-zero entries. But we can show that from the property of Lagrange basis polynomials (tensor products of Lagrange basis polynomials for the multi-dimensional case), the local matrix has a structure to be constructed from only $O(N)$ values. When the assembled equation of FEM is solved by some iterative method with element-by-element approach, we can also show that the arithmetic complexity of multiplication of a local matrix to a local vector is reduced if the structure of the matrix is used.

2 FEM DISCRETIZATION OF PDE

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 which depends only on the spatial coordinate $\mathbf{x}$. For the sake of simplicity, we assume the region of the analysis is rectangular whose sides are parallel to coordinate axes (Fig.1). We subdivide each edge of the region arbitrarily, and inside each finite element which is made by the subdivision the unknown

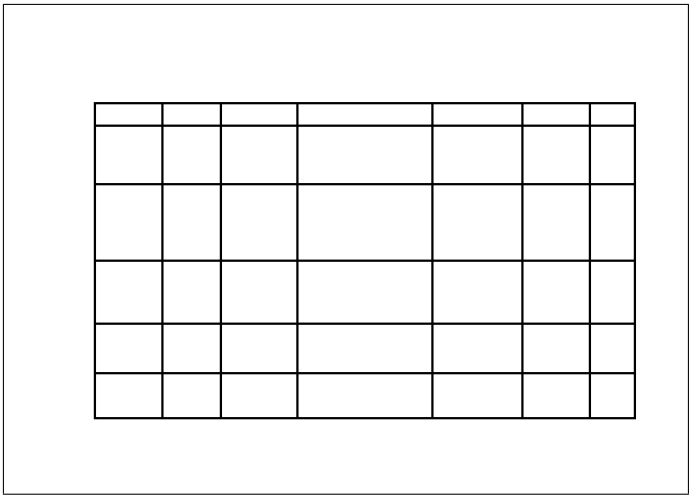

Figure 1: Conceptual FEM element subdivision(2-D)

HPCAsia2018, January 29–31, Akihabara, Tokyo, Japan

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

continuous function is approximated by a linear combination of Lagrange basis polynomials. When some iterative method is used to solve simultaneous linear equations of large size, 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.

3 STRUCTURE OF MATRIX ELEMENTS

3.1 Lagrange Basis Polynomials

The Lagrange interpolation formula inside an interval I at n-distinct abscissas $\alpha = \{\alpha_j \,|\, j = 1, 2, \ldots, n\}$ 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, $P^{(\alpha)}(x) \equiv \prod_{k=1}^{n}(x-\alpha_k)$ is a degree n polynomial corresponds to the set of n abscissas α.

The approximation of the original function f by the polynomial $\widetilde{f}$ would go better as the number of abscissas n increases if the set of abscissas is selected appropriately.

Usually, both-ends of the interval I are included in the set of abscissas to make easy continuous connections of the solution at boundaries to adjacent local elements. (In this case, a good set of abscissas is the one for Gauss-Lobatto quadrature formula. Abscissas of n-points Gauss-Lobatto 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 them are inside the standard interval.) Lagrange basis polynomials for function expansion are n different polynomials of degree $(n-1)$ (Fig.2).

In the followings, from definitions of the basis polynomials we remove factors $1/P^{(\alpha)\prime}(\alpha_j)$ to make formulae simpler. Thus our basis polynomials are $\varphi_k^{(\alpha)}(x) \equiv P^{(\alpha)}(x)/(x-\alpha_k)$,

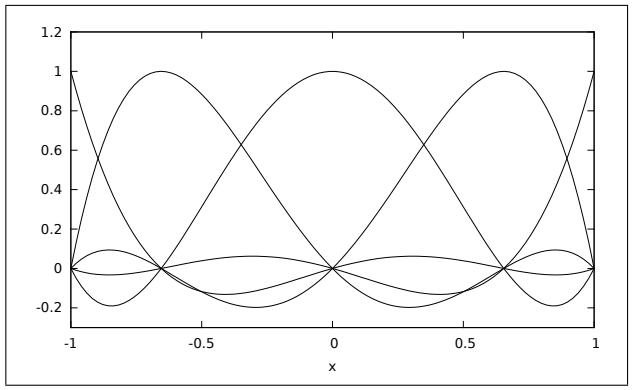

Figure 2: Lagrange basis polynomials by 5-points Gauss-Lobatto abscissas

for $k = 1, \ldots, n$. If required, matrix elements based on the original Lagrange basis polynomials can be easily obtained by just scalings associated with the index k of the basis polynomials.

The essence of our method is as follows. By using a simple and trivial algebraic 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 basis polynomials inside a finite element can be re-expressed as a sum of two different new functions:

$$\varphi_i^{(\alpha)}(x)\,\varphi_j^{(\alpha)}(x) = \frac{1}{\alpha_i - \alpha_j}\left\{\Phi_i^{(\alpha)}(x) - \Phi_j^{(\alpha)}(x)\right\}.$$

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

3.2 Case of 2-D FEM

3.2.1 Tensor Product Base Functions (2-D). If the local finite element is parameterized by a direct product of local coordinates, basis functions are taken as tensor products of Lagrange basis polynomials 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

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

3.2.2 FEM Matrix Elements of 2-D Potential Function. Matrix elements of 2-D potential $U(x,y)$ are:

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

If we define:

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

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

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

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

Therefore, from the above algebraic relations, if we set $N \equiv n_1 n_2$ all $N(N+1)/2$ entries of the symmetric matrix $V_{(i_1,i_2),(j_1,j_2)}$ can be built from only $4N$ values of $L_{i,j}^{(e_1,e_2)}$. If this knowledge on the structure of matrix elements is used, we can reduce the storage requirement to keep them.

In Table 1, under an assumption of $n_1 = n_2 = n$, we show values of $N(N+1)/2$ which is the number of entries of local matrix of discretized potential function, and values of $4N$ which is the total number of entries of 4 reduced matrices for comparisons.

This algebraic structure of matrix elements is derived only from the use of tensor products of Lagrange basis polynomials for the expansion of a function in a rectangular element, and the use of a simple identity for a product of fractions.

Table 1: Numbers of entries of a local FEM matrix and 4 reduced matrices (for 2-D potential function)

n	4	5	6	7	8	9	10
$N = n^2$	16	25	36	49	64	81	100
$N(N+1)/2$	136	325	666	1,225	2,080	3,321	5,050
$4N$	64	100	144	196	256	324	400

3.2.3 FEM Matrix Elements of 2-D 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

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

and

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

are prepared, then the matrix of FEM discretized Laplacian in 2-dimension $-\Delta = -(\nabla_x^2 + \nabla_y^2)$ 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 this structure is used, we can make the fast multiplication of the matrix $K^{(\alpha,\beta)}$ and a vector. When linear scaling factors from widths of the element are ignored, types of matrices $M^{(\alpha)}$ and $K^{(\alpha)}$ in x-direction are types of basis functions inside finite elements in x-direction, and types of matrices $M^{(\beta)}$ and $K^{(\beta)}$ in y-direction are types of basis functions inside finite elements in y-direction. So, if there are not so many types of basis functions of each directions in finite elements in total, we can calculate and keep a small number of matrices K and M for 1-dimension for the standard interval for each direction corresponds to types of basis functions inside elements with not so much amount of storage requirement, and use the stored 1-dimensional matrices selectively with proper scalings.

For the case of 3-D problem also, structures of local FEM matrices can be derived in similar manner.

4 FAST MATRIX-VECTOR MULTIPLICATION

In the previous section, a structure of FEM local matrix are shown and reduced matrices are introduced.

By the use of the structure, the number of numerical integrations to obtain whole matrix elements (of the potential function) is reduced, to be useful especially when accurate numerical integrations are difficult and many quadrature abscissas are required for integration.

The knowledge of the structure could be used also to reduce the amount of storage requirement. For example, when there are many higher order FEM local elements, and the assembled global linear equation of FEM is solved by some iterative method which uses many matrix-vector multiplications repeatedly. When local FEM matrices are stored by using their structures in reduced forms, the amount of storage transfer is reduced from the one when we store local matrices in explicit forms.

Moreover we can show that, the multiplication of the local matrix to a given vector can be made faster when reduced matrices are stored and used than the multiplication by the trivial method which uses the local matrix stored in the explicit form.

We define an anti-symmetric matrix C as:

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

All matrix elements of C can be calculated easily from a set of n values of α_k on demand without forming C explicitly.

The problem to multiply the matrix C in the above form to a given vector as fast as possible has been known as the Trummer's problem, and a fast algorithm has been known whose arithmetic complexity is $O(n(\log n)^2)$ (or $O(n \log n)$ if the special distribution of abscissas as Chebyshev zeros is chosen)(Gerasoulis et al., 1987).

Theoretically very interesting, however this fast algorithm may not be fast for the actual size of our application and the numerical behavior is also not clear. Therefore, in this paper we will not consider the fast algorithm for Trummer's problem even it might be useful when we used very high order Lagrange basis functions for FEM.

4.1 Case of 2-D FEM

4.1.1 Potential Function Term (2-D). The matrix elements $V_{\mathbf{i},\mathbf{j}}$ of the 2-dimensional potential can be written by using matrices $C^{(\alpha)}$ and $C^{(\beta)}$ which have zeros in diagonals as:

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

In below, we will show some parts of the calculation required in matrix-vector product of the matrix $V_{\mathbf{i},\mathbf{j}} = V_{(i_1,i_2),(j_1,j_2)}$ and a vector $\mathbf{u_j} = u_{j_1,j_2}$. (For simplicity, in below we placed $n_1 = n_2 = n$, and let $N = n_1 n_2 = n^2$. All induces i_1, i_2, j_1, j_2 run from 1 to n.)

- For the case $L^{(0,0)}$ has no index of j_1 nor j_2, the term below can be calculated in $O(n^3)$ arithmetic operations.

$$\begin{aligned}&\sum_{j_1,j_2} C^{(\alpha)}_{i_1,j_1} C^{(\beta)}_{i_2,j_2} L^{(0,0)}_{i_1,i_2} u_{j_1,j_2}\\ =\;& L^{(0,0}_{i_1,i_2} \big(\sum_{j_1} C^{(\alpha)}_{i_1,j_1} \big(\sum_{j_2} C^{(\beta)}_{i_2,j_2} u_{j_1,j_2}\big)\big)\,.\end{aligned}$$

- For the case $L^{(0,0)}$ has an index j_2, the term below can be calculated in $O(n^3)$ arithmetic operations.

$$\begin{aligned}&\sum_{j_1,j_2} C^{(\alpha)}_{i_1,j_1} C^{(\beta)}_{i_2,j_2} L^{(0,0)}_{i_1,j_2} u_{j_1,j_2}\\ =\;& \sum_{j_2} C^{(\beta)}_{i_2,j_2} L^{(0,0)}_{i_1,j_2} \big(\sum_{j_1} C^{(\alpha)}_{i_1,j_1} u_{j_1,j_2}\big)\\ =\;& \sum_{j_2} C^{(\beta)}_{i_2,j_2} \big(L^{(0,0)}_{i_1,j_2} b_{i_1,j_2}\big)\\ =\;& \sum_{j_2} C^{(\beta)}_{i_2,j_2} c_{i_1,j_2}\,.\end{aligned}$$

- For the case $L^{(0,0)}$ has an index j_1, the term below can be calculated in $O(n^3)$ arithmetic operations.

$$\begin{aligned}&\sum_{j_1,j_2} C^{(\alpha)}_{i_1,j_1} C^{(\beta)}_{i_2,j_2} L^{(0,0)}_{j_1,i_2} u_{j_1,j_2}\\ =\;& \sum_{j_1} C^{(\alpha)}_{i_1,j_1} L^{(0,0)}_{j_1,i_2} \big(\sum_{j_2} C^{(\beta)}_{i_2,j_2} u_{j_1,j_2}\big)\\ =\;& \sum_{j_1} C^{(\alpha)}_{i_1,j_1} \big(L^{(0,0)}_{j_1,i_2} \widetilde{b}_{j_1,i_2}\big)\\ =\;& \sum_{j_1} C^{(\alpha)}_{i_1,j_1} \widetilde{c}_{j_1,j_2}\,.\end{aligned}$$

- For the case $L^{(0,0)}$ has both induces j_1 and j_2, the term below can be calculated in $O(n^3)$ arithmetic operations.

$$\sum_{j_1,j_2} C^{(\alpha)}_{i_1,j_1} C^{(\beta)}_{i_2,j_2} \big(L^{(0,0)}_{j_1,j_2} u_{j_1,j_2}\big) = \sum_{j_1,j_2} C^{(\alpha)}_{i_1,j_1} C^{(\beta)}_{i_2,j_2} d_{j_1,j_2}$$

$$= \sum_{j_1} C^{(\alpha)}_{i_1,j_1} \big(\sum_{j_2} C^{(\beta)}_{i_2,j_2} d_{j_1,j_2}\big)\,.$$

- (We omit to show calculations of other terms, whose arithmetic complexity are less in the order of n.)

By the use of the structure, the number of arithmetic operations to multiply the matrix V to a vector $\mathbf{u}$ is $O(n^3) = O(N^{3/2})$ in total. If the symmetric matrix V is explicitly formed without using the structure, then V is dense and has $n^2(n^2+1)/2 = N(N+1)/2$ entries to store, and for the matrix-vector multiplication the required number of arithmetic operations is $O(n^4) = O(N^2)$. Therefore, the structure can be used to reduce the order of arithmetic operations. (If the fast algorithm for the Trummer's problem were applied, the number of arithmetic operations could be reduced to $O(n^2(\log n)^2) = O(N(\log N)^2)$.)

4.1.2 Laplacian Term (2-D). If we multiply the matrix of FEM discretized 2-D Laplacian $K^{(\alpha,\beta)}_{\mathbf{i},\mathbf{j}} = K^{(\alpha,\beta)}_{(i_1,i_2),(j_1,j_2)}$ to the vector $\mathbf{u_j} = \mathbf{u}_{(j_1,j_2)}$, by the use the matrix structure $K^{(\alpha,\beta)}_{(i_1,i_2),(j_1,j_2)} = K^{(\alpha)}_{i_1,j_1} M^{(\beta)}_{i_2,j_2} + M^{(\alpha)}_{i_1,j_1} K^{(\beta)}_{i_2,j_2}$, the number of arithmetic operations required is $O(n^3) = O(N^{3/2})$.

5 CONCLUSION

In order to overcome the performance bottleneck of modern high-speed computer systems, we have made a research on a technique to reduce both amounts of storage requirement and data transfer.

We have shown an algebraic structure of local FEM matrix of an operator (such as potential function) when Lagrange basis polynomials are used for FEM discretization.

When N is the number of (tensor) basis polynomials in an element of FEM, the discretized matrix of an operator such as potential function is dense and has $O(N^2)$ non-zero entries in its explicit form. But those (N^2) entries can be expressed by combinations of entries of reduced matrices which in total have only $O(N)$ entries. Therefore, we can reduce the amount of storage to hold the discretized operator. Also the number of numerical integration can be reduced since only integrals for reduced matrices have to be calculated.

(In this paper, we have shown for 2-D case only. For 3-D case also, we use 3-D tensor products of Lagrange basis polynomials for FEM expansion, and let N is the number of 3-D tensor basis polynomials in a FEM element, then the local matrix is dense and has $O(N^2)$ non-zero entries, which can be re-expressed by combinations of entries of reduced matrices which have in total only $O(N)$ entries to be stored.)

When we solve the global linear equation by some iterative method, the global FEM matrix is assembled from local FEM matrices and it is multiplied to a vector by the element-by-element approach. We have shown that the arithmetic complexity to make matrix-vector multiplication in a local element can be reduced when we use reduced matrices than we use the local matrix formed explicitly.

In future work, we have to make some numerical experiments of 2-D or 3-D FEM problems to show both amounts of storage requirement and computation could be reduced.

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