数理解析研究所講究録
第548巻 1985年330-49

Reordering Algorithm for Skyline Method

Okayama University Takeo Taniguchi
谷口 健男

1. Introduction

Let

$$Ax = b \qquad (1)$$

be an N*N sparse, symmetric nonsingular system of linear equations. As a direct solver for eq.1 we find BAND MATRIX METHOD, SKYLINE METHOD and WAVE-FRONT METHOD, and their efficiency wholly depends on the elimination ordering for eq.1. On the other hand, the theoretical investigation on their ordering problem from the aspect of combinatorial problems clarified that the searching of the optimum orderings for them becomes NP-complete problems. This result indicates that to find the optimum elimination ordering generally requires long execution-time, and, therefore, the ordering method to be proposed may become the one aiming near optimum elimination ordering. Actually we find several effective methods which are applicable for above solvers[1-5]. Among them Reverse Cuthill-McKee(RCM) and Gibbs-Poole-Stockmeyer(GPS) algorithms are often applied for not only the band solver but also the skyline method. That is, both of them are fundamentally

proposed for the band solver.

In this paper the author tries to propose new profile reducer. For this final purpose the optimum elimination orderings for above three solvers are firstly investigated from the theoretical viewpoint, and he clarifies the differences between their orderings. By using these theoretical results new reordering algorithm for the skyline method is proposed, and its efficiency is surveyed through a number of numerical tests.

2. Preliminaries

Firstly some terminology of the theory of graph. An undirected graph $G=(X,E)$ consists of a set X of nodes together with a set E of lines, which are unordered pairs of distinct nodes of X. A subgraph $G'=(X',E')$ of G is one for which $X' \subset X$ and $E' \subset E$. The nodes x and y are adjacent if $\{x,y\} \in E$. A graph is complete when every pair of nodes in G are adjacent. A clique of G is a subgraph which is complete. The degree of a node x is equal to the number of lines gathered at x and it is denoted by deg.x. The distance between two nodes, d(x,y), is the number of lines on the shortest path connecting them. If $d(x,y)<\infty$ for $x,y \in X$, G is a connected one. A level structure of G is a partition

$$L = \{L_0, L_1, L_2, \ldots, L_\ell\}$$

of the node set X such that

$$\text{adj.}L_i \subset L_{i-1} \cup L_{i+1}$$

$$\text{adj.}L_0 \subset L_1 \quad \text{and} \quad \text{adj.}L_\ell \subset L_{\ell-1}$$

in which "adj." is the abbreviation of adjacent.

Let a_{ij} be (i,j) entry of A in eq.1. Then,

$$B = \max_{\substack{i=1 \\ j \geq i}}^{n} (j - i) \tag{2}$$

$$P = \sum_{i=1}^{n} \max_{\substack{a_{ij} \neq 0 \\ i \geq j}} (i - j) \tag{3}$$

Eq's 2 and 3 are the definitions of the half-bandwidth and profile of A, respectively, and they determine the efficiency of the band and skyline solvers. That is, the aim of reordering is to minimize the values of B and P. For these solvers all data neccessary for solving eq.1 are stored in CPU. Wave-front method never treat the matrix A as a whole in CPU but constructs only submatrix in CPU by getting data of A row-wisely from the auxiliary memory apparatus. After the elimination operation for the submatrix only data in the submatrix subjected to the elimination are returned to the auxiliary memory and data for new row of A are taken into CPU. For the simplicity we assume that all data in the submatrices treated in CPU are stored in two-dimensional array. Then, for the effective usage of CPU there arise following two purposes of elimination ordering;

$$\text{CPU memory} \propto \max F_i \tag{4}$$

$$\text{Execution-time} \propto N \tag{5}$$

, where F_i is the dimension of the submatrix constructed in CPU and N is total number of nonzero entries of A after the forward elimination process. Here, we note that new nonzeros appear in A through the elimination process and they are called "fill-ins". This phenomenon is explained by following equation; The i-th row elimination of A modifies a_{jk} to $\tilde{a}_{jk}$ which is expressed as

$$\tilde{a}_{jk} = a_{jk} - a_{ji}a_{ik}/a_{ii} \qquad (6)$$

That is, (j,k) entry becomes nonzero if both of a_{ji} and a_{ik} are not zeros. Furthermore, this equation indicates that the subgraph consisted of only nodes which are gathered at the eliminated node becomes complete after the elimination.

Since the number of nonzeros, N, in eq.5 is the sum of nonzero entries in original A and the fill-ins, eq.5 is replaced as following;

$$\text{Execution-time} \propto F \qquad (7)$$

, where F is the number of fill-ins.

For the matrix A, prepare n nodes labeled from "1" to "n", respectively, and connect two nodes, v_i and v_j, by a line for every nonzero a_{ij} in the upper triangular matrix of A. By this procedure we obtain a graph G(n,m) from a matrix A(n*n) whose upper triangular area contains m nonzero entries.

Let L be a level structure of G. If each level L_i of L consists of only one connected subgraph, G has the 0th level convexity. If successive two levels of L construct a connected subgraph, the graph is the 1st level convexity.

3. Minimum B, P and F Problems

It is obvious that the minimization of eq's 2 to 7 determines the efficiency of the band, skyline and wave-front solvers. In this section the minimum B, P and F problems are theoretically treated, and we show that the strict elimination ordering for the solvers are different each other.

3.1 Minimum Half-Bandwidth Problem

By using the graph G(n,m) instead of A(n*n) the minimum B problem is expressed as following;

$$\min B \doteqdot \min \{ \max_{\substack{v_i \in \text{adj}.v_j \\ j \geq i}} (i - j) \} \tag{8}$$

That is, the maximum difference of labels of two nodes which are adjacent each other must be minimized. Assume that a level structure, L, is constructed for a graph G, and all nodes are successively labeled from the first level of L. Then, the half-bandwidth is given as following;

$$B \doteqdot \max |L_i|, \text{ for } i = 0,1,2,\ldots,\ell \tag{9}$$

, where $|L_i|$ is the number of nodes in the i-th level of L. Therefore, eq.8 is replaced by

$$\min B \doteqdot \min \{ \max |L_i| \} \tag{10}$$

That is, min B depends on how to construct a level structure and it is governed by a level which includes the maximum number of nodes.

The complexity of min B expressed by eq.10 for arbitrary graph is well discussed in Ref.6, and the study concludes that the design of effective algorithm for eq.10 is possible only for low-level convex graphs. More precisely, the effective design becomes possible if the level structure which is constructed from one end of the diameter has low-level convexity. This indicates that the optimum ordering for min B may be fundamentaly based on the level structure. GPS and RCM algorithms are effective only for graphs with above characteristic, and Ref.7 is also effective

for them even though its strategy is quite different from above two algorithms.

3.2 Minimum Profile Problem

Let's denote the number of zero entries in the profile by Loss. Then, the definition of P is replaced as following;

$$P = m + Loss \tag{11}$$

Since "m" is the number of nonzeros, the minimization of P is equivalent to the minimization of Loss.

$$\min P \equiv \min Loss \tag{12}$$

By the elimination process a part of Loss becomes fill-ins and the residuals keep to be zero. That is,

$$\min P \equiv \min (F + Z) \tag{13}$$

in which F and Z are the number of fill-ins and zeros, respectively.

Let h_i be the i-th column height of A;

$$h_i = \max_{\substack{v_j \in adj.v_i \\ i \geq j}} (i - j) \tag{14}$$

h_i depends on the number of nodes being eliminated between v_j and v_i. Assume that all nodes adjacent to v_i are successively eliminated. Then, h_i has the minimum values, because h_i consists of only nonzeros. If any node, namely $v_k \notin adj.v_i$, is eliminated between v_j and v_i, then a zero element is included in h_i. It is obvious that this zero entry in h_i is Z in eq.13.

Actual matrix we treat is a sparse one, and, therefore, even if the ordering is one of best ones, most of h's neccessarily include many zero entries before the elimination. Thus, above

consideration suggests that the elimination should be continued so as to fulfill all zero entries in P by fill-ins.

We may assume that any row (or column) in A has only several nonzeros originally. As denoted in Preliminaries, the nodes in a subgraph which are already eliminated and faces to non-eliminated area construct a clique. Then, the number of fill-ins in the subgraph depends on the square of the number of nodes in the subgraph. This suggests that in order to minimize the fill-ins cliques appearing through the elimination processes must always be kept small. That is, any clique should locate so as to cross the direction of the diameter of the graph. Summarizing above consideration we obtain following expression for min P;

$$\min P \equiv \min F \text{ with } Z=0 \qquad (15)$$

As far as we treat any convex graph cliques may be located so as to keep above restriction, but for non-convex graphs like fingure- or star-type graphs cliques may locate so as to cross the diameter of the graph but a part of the clique often locates along the direction of the longitudinal axis of the convex area. Ref.8 surveyed on this problem, and it proposed following new expression for min P;

$$\min P \Rightarrow \min F + \min Z \qquad (16)$$

This expression indicates that (1) any non-convex graph is firstly replaced by a gatherings of convex subgraphs by cutting at the roots of convex areas, (2) each subgraph is independently treated so as to satisfy eq.15 and (3) at the assemblage of ordered nodes in subgraph into the original one min Z is considered.

Now, we consider on min Z in eq.16. From the discussion on h_i of eq.14 it is obvious that the value of Z is determined by the number of nodes in convex-subgraph and the number of nodes facing to the cutting line in the first procedure. That is,

$$\min Z = \min_{i \neq j} \{ a_i N_j \} \qquad (17)$$

, where the suffix indicates the label of convex subgraphs and a and N indicate the number of nodes on the cutting line and the number of nodes in a subgraph, respectively.

Summarizing the results of the consideration on min P problem for convex graphs we may treat the problem of min F with Z=0, and for non-convex graphs we have to treat it as min F + min Z. Therefore, the ordering for convex graphs may be a similar one of the ordering of min B, because the concept of level structure may be an effective tool for this case, too. But, for the case of non-convex graphs it is obvious that both optimum orderings are quite different each other. By using above results the author proposes new algorithm in the next chapter.

3.3 Minimum Fill-in Problem

In a glance this problem is very similar to eq.15 but we find that eq.15 subjects to the restriction of "Z=0". We examine the meaning of this restriction.

In order to eliminate all nodes in a graph with Z=0 any node for the successive elimination is always selected among nodes which are adjacent to the eliminated area. Thus, min F problem treated here may ignore the appearance of Z.

As discussed in Section 3.2, the number of fill-ins depends

on the number of nodes included in each clique appearing at a vertex-elimination. Therefore, for min F in this section how to minimize the number of nodes in each clique is the main object of the node-reordering. On this subject we find excelent study by A. George[9,10]. The nested dissection method and its extended method are very effective for min F.

As obvious from above consideration on min F problem the node-ordering for min F becomes quite different from the one with the restriction of Z=0.

4. New Profile Reducer

According to the results in previous section the node-reordering algorithm for the minimum profile must include following steps;

1. Selection of the starting node.
2. Separation of a graph into convex subgraphs.
3. Min F ordering algorithm for convex graph.
4. Min Z ordering algorithm.

At first each procedure is examined from the viewpoint of the algorithm design.

[Selection of the starting node]

It is obvious that the starting node should be one end of the diameter of the graph. But, it is already known that the search of the diameter requires a lot of numerical operations[11]. On the other hand, Gibbs et al proposed an effective method for searching an end of the longitudinal axis of any graph[2,3]. In this paper their method is introduced for the determination of

the starting node.

By the application of their method all nodes in G are ordered into a level structure

$$L = \{ L_0 \Rightarrow v_1, L_1, L_2, \ldots, L_\ell \}$$

, where v_1 is the starting node. This level structure and its characteristics are usefully introduced in successive three procedures.

[Partition of a graph]

Since the node-ordering is started from one end of the longitudinal axis of the graph and the level structure is generated from the starting node, the judgement whether the graph is convex may be done only for the level structure.

Now, we assume a graph consisted of only triangular mesh. According to the definition of any graph the non-convexity is recognized if all nodes in a level set, namely L_i, of the level structure are not connected each other. But, in preceding section it is clarified that the actual judgement whether the non-convex graph should be treated so as to minimize F and also Z must be done by taking consideration the properties of the convex area, i.e. the length of the cutting line and the nodes included in the area.

Assume that in level sets $\{ L_i, L_{i+1}, \ldots, L_{i+k} \}$ we find the discontinuity between nodes. If all nodes in L_i and L_{i+k} have connectivity to the nodes in L_{i-1} and L_{i+k+1}, respectively, then we can recognize that these level sets construct a ring-type structure. If some nodes in L_{i+k} have no connectivity to

the member of L_{i+k+1}, then the subgraph includes a convex area.

Let express L_j as $L_j^1 \cup L_j^2$ for $i \leqq j \leqq i+k$. By using above level sets we can easily calculate the number of nodes in the convex area as $\Sigma|L_j^1|$ and $\Sigma|L_j^2|$, respectively. From the viewpoint of algorithm design the method of the judgement must be simple. That is, we may aim the decreasing of the profile value instead of the minimum profile. Then, if

$$k + 1 \geqq \left(\sum_{s=i}^{i+k} |L_s^a| \right)^p \tag{18}$$

is satisfied for the smallest convex area, we treat the area as a convex area for the ordering. Otherwise, we ignore the convexity and we aim the reordering of min F for the area. The value of "p" in eq.18 is equated to 1/2 and 1/4 for a planar and spatial graphs, respectively. These values were decided by the results of a number of numerical experiments.

For a graph with ring-type structure all nodes in $\{ L_j^1 \}$ or $\{ L_j^2 \}$ for $i \leqq j \leqq i+k$ are ordered before the ordering of another.

[Min F ordering method]

By using above level structure and above judgement any graph can be replaced to gatherings of convex subgraphs. Note that all nodes in each subgraph are already set into a level structure and that the level structure is constructed so as to cross the longitudinal axis of the graph. Therefore, all nodes of any convex area may be successively ordered from one level to the adjacent one. This procedure is repeated for all subgraphs, and this procedure is finished.

[Min Z ordering method]

According to the result in previous section min Z ordering should be done by considering eq.17. But, for the simplicity we consider only the number of nodes in convex subgraphs;

$$\min Z \Rightarrow \min N_j \qquad (19)$$

Assume that three convex subgraphs are gathered and that they construct a star-type nonconvex graph. Then, all nodes in the smallest area are ordered between the others. For this ordering we compare only the numbers of nodes in the convex subgraphs, and the algorithm is simplified.

According to above considerations we propose new profile reducer. Since the reverse ordering is effective for the profile reducing, we use it in this new method, too[2].

The input data neccessary for this new method are as follows;

1. Total number of nodes
2. Expected maximum degree
3. Node-node incidence
4. Parameter

Among four procedures neccessary for the new method the first, the second and the last two procedures are presented as ALGORITHM START, ALGORITHM BREAK and ALGORITHM PROF, respectively. Note that ALGORITHM START is the one given by George[2]. The details of these algorithms are given as followings;

ALGORITHM START

Step 1 : Find a node r of minimum degree.

Step 2 : Generate the level structure rooted at r;

$L(r) = \{L_0(r), L_1(r), L_2(r), \ldots, L_\ell(r)\}$

Step 3 : Find all connected components in $L_\ell(r)$.

Step 4 : For each component C in the level set, find a node x of minimum degree and generate its rooted level structure. If $\ell(x)>\ell(r)$, and put $r \Leftarrow x$ and go to step 3.

Step 5 : r is a starting node.

The aim of ALGORITHM BREAK is the searching of the convex areas, and for this purpose above level structure is reversely used.

ALGORITHM BREAK

Step 6 : Find all connected components in each level set from the last level, reversely. If all levels consist of only one connected component, then reorder the level structure and go to step 14.

Step 7 : Divide L(r) into sub-level structures, namely $\{L^0, L^1, \ldots\}$, by considering the connectivity between connected components in successive level sets in L(r). We assume that L^0 includes the starting node.

Step 8 : For the first level set of each sub-level structure, search the connectivity to other sub-level structures. If there exists no connectivity and the sub-level structure doesn't satisfy eq.18, restore all nodes in the sub-level structure to the original places in L(r).

Step 9 : Step 8 is repeated for all sub-level structures except L^0.

ALGORITHM PROF

Step 10: Find a sub-level structure, namely L^1 , whose last level

set is connected to a level set of L^0. This procedure is continued in the ordering of level sets of L_i^0.

Step 11: If the first level set of L^i has connectivity to any level set of L^0, then go to step 13.

Step 12: If the sub-level structure include more nodes than level sets of L^0 till the level set where the sub-level structure is connected, insert the sub-level structure before L^0 and go to step 10.

Step 13: The sub-level structure is inserted behind the level set of L^0 which was found in step 10 in succession, and go to step 10.

Step 14: L^0 is the new node-ordering fot the skyline method.

5. Numerical Examples

In order to survey the efficiency of the proposed method more than 10 test examples are solved by using the new method(NEW), Reverse-Cuthill-McKee(RCM) and Gibbs-Poole-Stockmeyer(GPS) algorithms, and their results are compared for the profile values, the execution-time and several other items. The reason of the usage of RCM and GPS for this comparison is that both of them are most commonly used at present. But, since original RCM lacks the procedure to select an appropriate starting node, it may often require long execution-time for the reordering. Therefore, we modify as following; Determine the starting node by using ALGORITHM START given in this paper and apply RCM only from the starting node. By this modification RCM generates only one level structure. Henceforce, we call this modified RCM as MRCM.

Table 1 Details of test examples

Case	No. of nodes	Dim.	Mesh	Subgraphs
1	42	2	0	1
2	47	2	1	2
3	59	2	0	3
4	99	2	0	1
5	136	2	0	1
6	148	2	0	1
7	193	2	2	2
8	232	2	0	1
9	374	2	1	3
10	414	2	1	2
11	630	2	1	2
12	157	3	0	3
13	211	3	0	3
14	312	3	0	5

Details of the test examples used in this paper are summarized in Table 1. In Table 1 the column "Dim." indicates whether the graph is a planar or a spatial one, the column "Mesh" indicates whether the graph has a ring-type structure. The column of "Subgraphs" indicates how many sub-level structures are generated by the proposed method. As obvious from the result of the theoretic investigation of the minimum profile problem the topology or the configuration of a graph becomes complicated as the graph contains more meshes and also sub-level structures. That is, these two columns indicate the complexity of the node-reordering.

14 test examples are used for the comparison of NEW, GPS and MRCM. 11 cases among them are planar graphs and other three are spatial ones. 5 examples have ring structure, and the maximum number of rings is two. The graphs whose column of "Subgraphs"

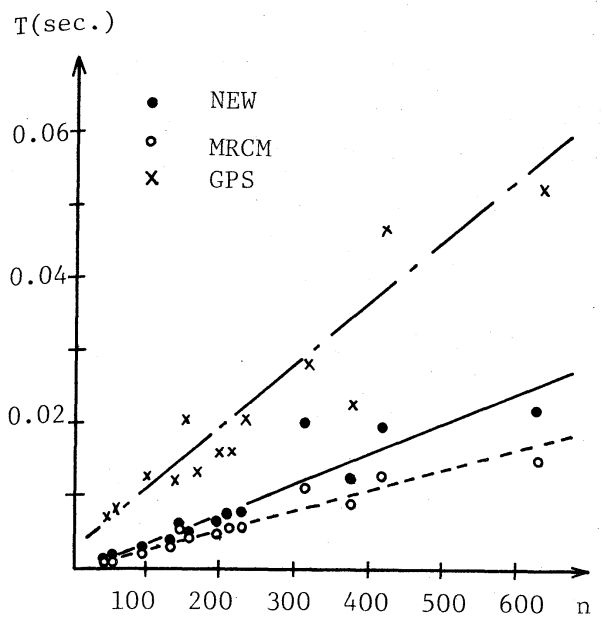

Fig.1 Comparison of execution-time

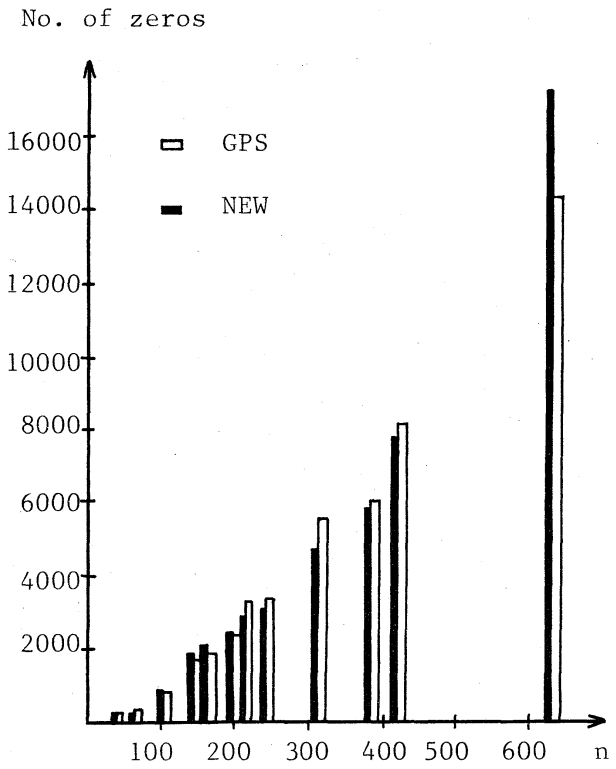

Fig.2 Comparison of zeros in profile

is equal to "1" are supposed to be convex, and, therefore, 9 test examples are non-convex graphs. Maximum sub-level structures appears in a graph of case 14.

The results of the execution-time only for the node-ordering and the number of zero entries in the profile are shown in Fig's 1 and 2, respectively. For the former three methods are compared , and for the latter only the results by GPS and NEW are compared, because MRCM is one of the fastest methods but it can give rather poor results with respect to the profile values[3].

Table 2 is given for the examination of the efficiency of NEW, GPS and MRCM as the ordering method not only for the profile but also the wave-front and the band matrix methods. The efficiency

Table 2. Comparison of results by NEW, GPS and MRCM

Case	Profile Solver						Wavefront Solver						Band Solver		
	Wavg			Wrms			Wmax			Wrms			B		
	MRCM	GPS	NEW	MRCM	GPS	NEW	MRCM	GPS	NEW	MRCM	GPS	NEW	MRCM	GPS	NEW
1	8.0	6.1	6.4	8.5	6.4	6.8	12	9	10	11.1	8.2	8.9	14	8	13
2	5.8	5.8	5.6	6.1	6.0	5.8	8	8	7	7.3	7.2	6.8	9	7	15
3	6.2	5.5	4.5	6.7	5.8	4.7	10	9	8	9.7	8.3	6.7	15	8	19
4	9.8	8.6	8.9	10.2	8.9	9.3	14	11	13	13.5	11.5	12.4	17	11	15
5	9.2	8.3	8.4	9.9	8.8	9.0	16	14	15	12.3	11.4	11.4	25	18	26
6	13.0	11.5	12.4	13.7	12.0	13.2	18	15	20	17.6	15.6	17.3	33	16	21
7	14.9	12.7	12.8	15.7	13.1	13.5	24	18	21	20.1	16.9	17.8	36	19	75
8	15.2	13.8	13.7	15.7	14.1	14.1	21	19	19	20.3	18.7	18.7	32	19	27
9	16.9	16.4	15.6	17.7	17.1	16.4	25	22	24	23.6	22.8	21.9	36	23	152
10	20.1	19.6	18.9	20.8	20.5	19.6	29	29	29	27.2	27.3	26.2	44	29	40
11	30.5	22.9	27.4	32.7	23.5	29.4	50	29	47	43.0	31.5	40.0	81	29	50
12	14.2	12.3	14.0	15.6	12.9	15.0	26	19	27	26.9	22.2	26.0	45	18	37
13	16.7	15.6	13.7	18.1	16.2	14.6	31	23	26	31.1	28.1	25.4	51	22	60
14	19.2	17.9	13.9	20.5	19.0	14.7	35	30	24	35.1	32.9	25.1	56	35	109

of the ordering method may be measured by the memory size neccessary for the data structure after the ordering and the execution-time for the ordering. According to Everstine[12] they are estimated by using two parameters, Wavg and Wrms, for the profile method and Wmax and Wrms for the wavefront method, and by using the half-bandwidth, B, for the band matrix method.

From the results of above numerical experiments we can remark following items;

1. New method requires only about 1.25 times of the execution-time by MRCM, and its execution-time is less than half of the one by GPS. This result indicates that NEW is also one of the fastest reordering methods.

2. New method can generally give almost the same profile values as GPS can, and comparing to the results by MRCM it can give better orderings. If the configuration of graphs is complex, new one can show its characteristics for the ordering, and the results become best.

3. As the ordering method for the profile method new one is as effective as GPS can, but it can give poor results for the ordering method for the band matrix method, especially for the non-convex graphs. The reason is that NEW is not designed for the decreasing of the bandwidth but for the decreasing of the matrix profile. As the ordering method for the wavefront method NEW is as effective as GPS is.

6. Concluding Remarks

The theoretical investigation on the minimum profile problem could give several important results, and by their introduction new profile reducer was proposed in this paper. Its efficiency was examined through a number of numerical experiments, and the results show that with respect to the profile value new method is so effective as Gibbs-Poole-Stockmeyer algorithm, and with respect to the execution-time for the ordering it requires only half of GPS does. Especially, the new method can give good results for graphs with complex configuration.

The propriety of the application of the profile reducer to the ordering of other solvers, for example the wavefront method, was also theoretically investigated, and we obtained the conclusion that different ordering method is required for them.

Numerical error appearing at the application of the profile method was also experimentally investigated, and we obtained that there exist slight difference between the band solver and the profile solver.

All of the computations in this paper were done by using a computer, ACOS 1000 Model 20, of Data Processing Center in Okayama University.

Acknowledgement

The author wishes to express his sincere appreciation for the support of Mr. A. Soga, a graduate student of Okayama University, in the preparing of this paper.

References

[1] Cuthill, E. and J. McKee,"Reducing the bandwidth of sparse symmetric matrices", Proc. of ACM National Conference, 1969, pp.157-172

[2] George, J.A.,"Solution of linear systems of equations : direct methods for finite element problems", In Sparse Matrix Techniques (ed. by V.A. Barker) Lecture Note in Mathematics, pp.52-101, Springer.

[3] Gibbs, N.E., W.G. Poole and P.K. Stockmeyer,"An algorithm for reducing the bandwidth and profile of a sparse matrix", SIAM j. Numer. Anal., 13, pp. 236-250

[4] Pina, H.L.G.,"An algorithm for front-width reduction", International J. Numer. Meth. Eng., 1981, 17, pp.1539-1546

[5] Razzaque, A.,"Automatic reduction of front-width for finite

element analysis", Int. J. Numer. Meth. Eng., 1980, 15, pp.1315-1324

[6] Konishi, I., N. Shiraishi and T. Taniguchi,"Reducing the banwidth of structural stiffness matrices", J. Structural Mechanics, 1976, 4.2 pp.197-226

[7] Shiraishi, N. and T. Taniguchi,"New renumbering algorithm for minimizing the bandwidth of sparse matrices", Advances in Engineering Software, 1980, 2.4 pp.173-179

[8] Taniguchi, T."New algorithm for the reduction of matrix profile", International Conference on Numerical Methods in Engineering, 1985 (In printing)

[9] George, J.A."Nested dissection of a regular finite element mesh", SIAM j. Numer. Anal., 1973, 10.2 pp.345-363

[10] George, A."Sparse matrix aspects of the finite element method ", Proc. 2nd International Symposium on Computing Methods in Applied Science and Engineering, 1976

[11] Shiraishi, N. and T. Taniguchi,"Design of algorithms for reducing the bandwidth and profile of sparse matrix", Proc. of 2nd International Congress of Numerical Methods for Engineering, 1980, pp.187-196

[12] Everstine, G.C.,"A comparison of three resequencing algorithms for the reduction of matrix profile and wavefront', Int. J. Numer. Meth. Eng., 1979, 14, pp.837-852