Memoirs of the School of Engineering, Okayama University, Vol. 15-1, November 1980

Fundamental Study of the Fill-in Minimization Problem

Takeo TANIGUCHI* and Katsu NUMATA**

(Received October 7, 1980)

Synopsis

In this paper the fill-in minimization problem which arises at the application of the sparse matrix method for a large sparse set of linear equations is discussed from the graph-theoretic viewpoint and also through the numerical experiments. Therefore, this investigation consists of two parts, and in the former part the author shows, at first, that the elimination process of a sparse matrix is equivalently replaced to the vertex eliminations for a graph obtained from the matrix, and by use of some concepts in the theory of graph he proves that the vertex elimination process for the minimum fill-in is equivalent to the vertex eliminations for vertices in each subgraph which is obtained by the appropriate dissection of whole graph, and that there are only two types of vertex eliminations through the process. This results in the proposal of a new model of the vertex elimination process.

The latter part of this investigation is used for the verification of the results from the theoretic investigation. Through the numerical experiments he concludes that the new model of the vertex elimination process is valid, at least, for a graph like a regular finite element mesh. Furthermore, he shows that this model coincides with Nested Dissection Method which can give the minimum value of fill-in, at present.

---

* : Department of Civil Engineering

** : Tomoegumi Steel Works Co., LTD.

1. Introduction

Since the powerful numerical methods using discretized systems as Finite Element Method and Finite Difference Method result oftenly in a large set of linear equations,

$$Mx = b \tag{1}$$

, where a matrix M is generally sparse, how to solve eq.(1) effectively becomes important and we find some efficient solvers, most of which utilize the sparsity of the matrix.

As the discretized system becomes large, the ratio of zero entries to the total number of elements of M increases, because total entries increase by $O(n^2)$, though the non-zeros by $O(n)$ for n by n matrix M. Any solver utilizing this sparsity aims to exclude as many zeros from the numerical operations and also from the input data, and for this purpose there arise new problems, i.e. the minimum bandwidth problem for BAND SOLVER [1], the minimum profile problem for PROFILE SOLVER [2] and the minimum fill-in problem for SPARSE MATRIX METHOD [3].

On the other hand it is well known that through the elimination process for eq. (1) additional non-zeros are produced, that is zeros in M are replaced to non-zeros during the elimination [4]. This is proved as following: Assume that the i-th row elimination alters m_{jk} element of M to m^*_{jk}, if $i<j$ and $i<k$.

$$m^*_{jk} = m_{jk} - m_{ij}\, m_{ki} / m_{ii} \tag{2}$$

Even if $m_{jk} = 0$ in M, m^*_{jk} in the modified matrix M* becomes non-zero for $m_{ij} \neq 0$ and $m_{ki} \neq 0$. Such a new non-zero entry is called "fill-in". Thus, zero elements in M are classified into "Fill-in" and "Zero" which is still zero after the eliminations. Therefore, the minimum value of the input data for the solvers is mainly governed by the sum of the non-zeros and fill-ins. SPARSE MATRIX METHOD is a solver using only them, though the others require additional zero entries, and it can be said that this solver is the most efficient one among sparse matrix solvers.

Though the number of non-zeros is determined when M is given, the number of fill-ins alters according to the elimination ordering, and the fill-in minimization method is now required. Some effective methods are already proposed for this purpose [4, 5, 6], and George's Nested Dissection Method among them can give the best ordering, though it has a strict restriction for its application[5,6]. Taniguchi shows independently that the ordering for the minimum fill-in induces neccessarily the appropriate dissection of system and that the method of the dissection is dependent to the system [3].

In this paper we consider on the fill-in minimization method from the graph-theoretic viewpoint and also through the numerical experiments, and aim to obtain some important informations which are valid for the actual proposal of new ordering.

2. Elimination Process on Graph

First of all some terminology of the theory of graph [5]. An undirected graph $G = (X,E)$ consists of a finite set X of nodes or vertices together with a set E of edges, which are unordered pairs of distinct nodes of X. A subgraph $G' = (X',E')$ of G is one for which $X' \subseteq X$ and $E' \subseteq E$. The nodes x and y are adjacent if $\{x,y\} \in E$. For $Y \subset X$, the adjacent set of Y, denoted by adj(Y), is $\{x \in X/Y \mid {}^{\exists} y \in Y \ni \{x,y\} \in E\}$. The degree of a node x in G, denoted by deg(x), is the number $|adj(x)|$, where $|\ .\ |$ is the cardinality of the set. A graph is complete if every pair of nodes is adjacent, and if a subgraph G' of G is complete, G' is called a clique. The distance between distinct nodes x and y is the number of edges locating on the shortest path connecting them, and we denote it by d(x,y).

Now replace the elimination process of a sparse matrix M to that on a graph G. Let M be a (n * n) sparse matrix with m non-zero entries in its upper triangular area except the main diagonal. At first prepare n nodes which are numerically ordered from 1 to n, respectively. Here we denote a vertex labeled i by v_i. Then, give an edge connecting v_i and v_j for every (i,j) non-zero entry of the upper triangular matrix of M. By this procedure we obtain a graph G for a matrix M. Then, the row by row elimination of M is equivalent to vertex elimination of G [4].

Let's consider eq.(2) on this graph. Eq.(2) shows that the i-th row elimination can give influence only to a submatrix of M which consist of some j-th rows with $j>i$ and $m_{ij} \neq 0$. This is explained on G representing M as that the i-th vertex elimination gives influences only to a subgraph which consists of vertices v_j with $j>i$ and $d(v_i,v_j) = 1$.

Let's denote two graphs of before and after the v_i elimination by G and G*, respectively. Then, even if $d(v_j,v_k) > 1$ but $d(v_i,v_j) = d(v_i,v_k) = 1$ in G, G* has the relation of $d(v_j,v_k) = 1$ through v_i elimination. Since $m_{jk} \neq 0$ is equivalent to the existence of an edge connecting v_j and v_k, a fill-in is recognized as the introduction of a new edge. Furthermore, we find that v_i elimination replaces the subgraph $\{adj(v_i)\}$ to a clique. We call this subgraph as a FVG (which is the abbrebiation of Frontal Vertex Group).

Now we investigate on the general vertex elimination process. Suppose a modified graph G* which is obtained after some stages of vertex eliminations in accordance with an arbitrary ordering of the vertex eliminations. Then, all vertices in original graph may be classified into three kind of vertices:

1). $v \in G_E$
2). $v \in G_N$
3). $v \in FVG$

, where G_E is a subgraph consisting of only vertices which are already eliminated, and G_N is a subgraph consisting of only vertices not eliminated and also not included in any FVG.

It is obvious that a vertex for next elimination is selected among $\{v \mid v \in G_N \cup FVG\}$. Therefore, any vertex v in G must belong to one of following types:

Type 1. $v \in G_N$

Type 2. $v \in FVG$, and $\{adj(v)\} \cap G_N \neq \phi$

Type 3. $v \in FVG$, and $\{adj(v)\} \cap G_N = \phi$

Type 4. $v \in FVG$'s, and $\{adj(v)\} \cap G_N = \phi$

Type 5. $v \in FVG$'s, and $\{adj(v)\} \cap G_N \neq \phi$

Among above five types a vertex elimination of Type 3 never create any fill-in.

Now, let's give physical interpretations for these five types. Type 1-elimination for a vertex v' creates a new FVG consisting of vertices $\{V \mid d(v',v)=1\}$, and Type 2-elimination of v' includes some vertices $\{v \mid d(v',v)=1, v \notin FVG\}$ into the FVG. Type 3 can decrease the number of vertices in the FVG by one. Type 4 and Type 5 reconstruct several FVG's into one FVG, and the latter can, furthermore, include some additional vertices in the new FVG. Since Type 5-elimination is thought as the combination of Type 2 and Type 4, the vertex elimination is classified into only 4 types, i.e. Type 1, 2, 3 and 4. Summarizing them,

Type 1 : Creation of new FVG.

Type 2 : Extension of FVG.

Type 3 : Schrinkage of FVG.

Type 4 : Coupling of several FVG's.

Considering above four types of vertex eliminations, we obtain that

1). at first Type 1-eliminations appear and the others appear after Type 1,

2). after Type 1-elimination is applied at least twice, Type 4 can appear, and

3). Type 3-elimination continues after Type 4.

3. Vertex Elimination Process for Minimum Fill-in

In this section we treat the vertex elimination process for minimizing the fill-in which is a special case of the process given in the previous section. Therefore, the process for the minimum fill-in is also constructed as an appropriate combination of four types of vertex eliminations.

From the results in previous section it is obvious that all vertices in G can be eliminated by using only Type 1 and Type 2 if and only if Type 1 is applied only once, and that otherwise, four types may appear in the elimination process. Now, we prove that the elimination of the former case cann't generally give the minimum fill-in. This proof is done by showing an example which cann't give the minimum fill-in by use of only one Type 1-elimination. A good example is a tree graph with , at least, one vertex whose degree is more than 3. Therefore, this results in that the elimination process includes, at least, two Type 1-eliminations, and that during the elimination process four types appear.

This result gives following very important information; since, at least, more than two FVG's appear through the elimination, any FVG must stop its growth by the successive Type 2-eliminations. We denote this FVG stopped its growth by $\overline{\text{FVG}}$. At this stage any $v \in \overline{\text{FVG}}$ cann't be eliminated as Type 2 but as Type 3 or Type 4. Therefore, some vertices $\{v \mid G_N \ni v \in \text{adj}(\overline{\text{FVG}})\}$ must be eliminated before the elimination of vertices in $\overline{\text{FVG}}$, that is another FVG must be newly created by the introduction of Type 1-elimination.

Here, we should notice that the influence of the elimination of a vertex v is restricted only to $\{\text{adj}(v)\}$. Therefore, the elimination of vertices in a subgraph enclosed by one $\overline{\text{FVG}}$ gives no influence to the residual of G, and, furthermore, only one vertex in the subgraph is eliminated as Type 1 and the other vertices in the subgraph as Type 2. From this consideration we obtain that the vertex elimination in any subgraph is proceeded successively from the vertex as Type 1. There appear as many $\overline{\text{FVG's}}$ as the number of Type 1 vertex eliminations for the original graph G^0, and and all the vertices enclosed in these $\overline{\text{FVG's}}$ are eliminated as Type 1 and Type 2. We denote the residual graph at this stage by G^1.

If we consider G^1 as a new graph, then the vertex elimination for G^1 can be done again by using two types, i.e. Type 1 and Type 2. By the repetition of replacing the residual graph to a new graph, the discussions for G^0 are directly applied to successive graphs, i.e. G^1, G^2, G^3, ..., and all vertices in the original graph are eliminated by using two types of vertex eliminations. From this model of the vertex elimination process we can notice that

1). how to find $\overline{\text{FVG's}}$ for each G^i is the most important factor for minimizing the fill-in, and
2). especially, the determination of $\overline{\text{FVG's}}$ for G^0 gives the largest influence to the value, because it decides the outlines of successive graphs.

Now consider on the characteristics of $\overline{\text{FVG}}$. A vertex $v_0 \in G$ is, at first, eliminated as Type 1, and $v_1 \in \text{adj}(v_0)$ may be successively eliminated as Type 2. Therefore, as far as Type 2 is applied as the successive elimination, a vertex for the next elimination must be selected among $\{v \mid v \in \text{adj}(v_0, v_1, v_2, \ldots)\}$. In general, as Type 2 eliminations are continued, the number of vertices in FVG, denoted by $|\text{FVG}|$, becomes large. The number of fill-in, denoted by $|F|$, appearing at any Type 2 elimination is decided by following expression.

$$|F| \propto |\text{FVG}| \times \deg(v)$$

, that is $|F|$ is governed by the product of the number of vertices in FVG and the degree of a vertex being eliminated. This suggests that successive Type 2 are continued not to increase the vertices in FVG by selecting a vertex with the minimum degree among $\{v \mid v \in \text{FVG}\}$. Therefore, FVG appearing by further elimination of $v \in \overline{\text{FVG}}$ includes too many vertices, and any vertex $v \in \text{adj}(\overline{\text{FVG}})$ should be included in another FVG, and thus, this $\overline{\text{FVG}}$ should be the boundary of these subgraphs.

From above expression of fill-in we can notice that max $|FVG|$ of G should not exceed the maximum width of the original graph, where "the width of a graph" is the shortest path crosses G and separates G into two subgraphs. Above discussions are valid not only for G^0 but also successive graphs G^i.

Though this new model of the vertex elimination process is directly obtained from the graph-theoretic considerations, the model is very similar to Nested Dissection Method by A. George [5, 6]. The reasons are

1). the graph is subdivided into a gathering of subgraphs,

2). all the vertices in a subgraph are successively eliminated from a vertex which is appropriately selected, and

3). the maximum dissection line locates at the widest portion of the graph.

Therefore, we can conclude that the discussions given in this section show a proof that George's method can give the minimum or near minimum value of fill-in.

4. Numerical Experiments

The investigation in Section 3 clarify general characteristics of the vertex elimination process for the minimum fill-in, but the results obtained there don't directly lead to actual procedure but only give effective suggestions. Therefore, the aims of this section are not only to give the proof of the results in previous sections but also to obtain important informations which are valid for the proposal of actual procedure for the determination of vertex elimination ordering.

The results of the vertex ordering given in this section are not by the computer but by hands so as to satisfy following conditions:

1). The vertex ordering should basically follow the results in Section 3.

2). The ordering method obtained must be simple.

3). The number of fill-ins must be less or equal to the minimum value obtained by using other methods proposed already.

As the other methods in item 3) the author uses a). Minimum Degree Algorithm, b). Minimum Defficiency Algorithm [4] and c). Nested Dissection Method [5, 6]. The numerical experiments for a) and b) are done by using computer but not for c).

The graphs used for these experiments are simple grid graphs with square and rectangular surrounding configurations whose edges are subdivided by $n \times n$ and $n \times m$, respectively. This kind of simple test models are chosen in order to clarify the characteristics of $\overline{FVG}$'s and also to make ease of hand reordering. The connectivity relation between vertices is that of the nine point of difference scheme, and some experiments are done for the five point of differnce scheme.

Some results according to above three conditions are presented in Fig.1, 2 and 3, and Table 1 and 2 summarize all the results of above procedure, a), b) and c).

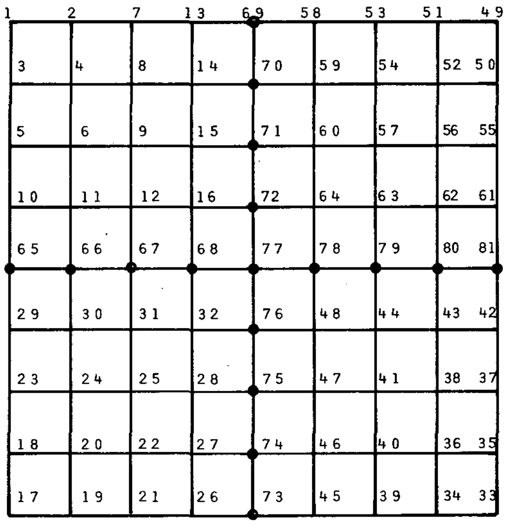

Fig. 1 Vertex Elimination Ordering of (9 × 9) Graph and Location of $\overline{\text{FVG's}}$. $|F| = 288$

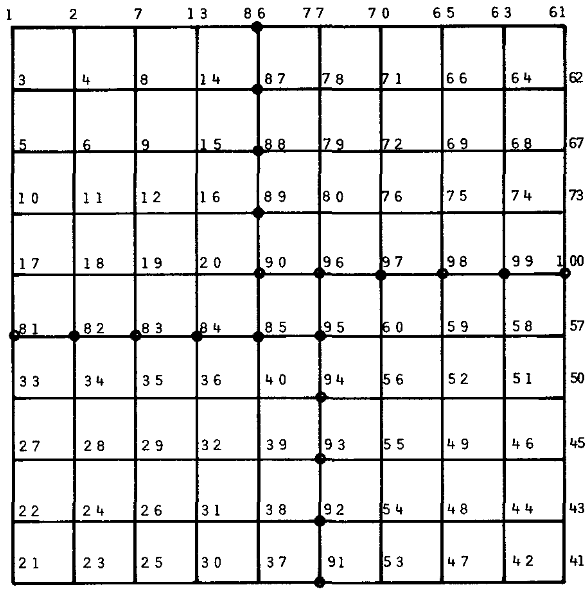

Fig. 2 Vertex Elimination Ordering of (10 × 10) Graph and Location of $\overline{\text{FVG's}}$. $|F| = 420$

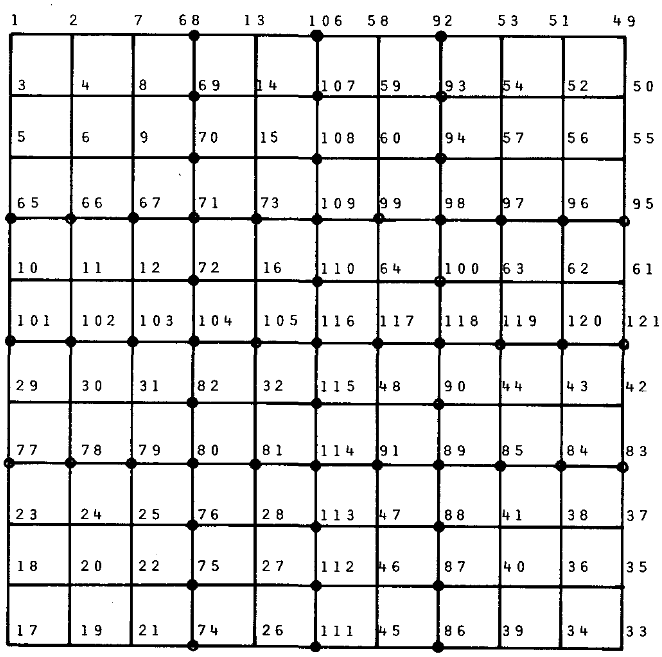

Fig. 3 Vertex Elimination Ordering of (11 × 11) Graph and Location of $\overline{\text{FVG's}}$. $|F| = 573$

References

[1] T. Taniguchi & N. Shiraishi," New Renumbering Algorithm for Minimizing the Bandwidth of Sparse Matrices", Advances in Engineering Software, Vol.2, No.4 (1980),

[2] N. Shiraishi & T. Taniguchi," Design of Algorithms for Reducing the Bandwidth and Profile of a Sparse Matrix", International Congress on Numerical Methods for Engineering (1980), (In printing)

[3] T. Taniguchi," A Graph-Theoretic Study of the Minimum Fill-in Problem for Sparse Matrix Method", Memoirs of the School of Engineering, Okayama University, Vol.13 (1979), 239-248

[4] D.J. Rose," A Graph-Theoretic Study of the Numerical Solution of Sparse Positive Definite Systems of Linear Equations", Graph Theory and Computing(Ed. R.C. Read) (1972), 183-217

[5] A. George," Nested Dissection of a Regular Finite Element Mesh", SIAM Journal of Numerical Analysis, Vol.4, No.2 (1973), 345-363

[6] A. George,"Solution of Linear Systems of Equations : Direct Methods for Finite Element Problems", Lecture Notes in Mathematics, 572, Sparse Matrix Techniques Copenhagen 1976 (Ed. by V.A. Barker), (1976), 53-101

Graph (n×n)	No. of Nodes	HAND-1	HAND-2	MIN.DEF.	MIN.DEG.	N.D.
5×5	25	28	28	28	-	/
6×6	36	64	64	64	-	/
7×7	49	104	104	110	-	/
8×8	64	191	191	207	218	/
9×9	81	288	288	316	349	300
10×10	100	420	420	439	473	/
11×11	121	573	585	610	677	/
12×12	144	748	795	799	898	/
13×13	169	968	1041	1018	1280	/
14×14	196	1235	1336	1320	1367	/
15×15	225	1519	1691	1681	1861	/
16×16	256	1860	2079	2002	2316	/
17×17	289	2268	2528	2406	2898	2280
18×18	324	2716	3056	2820	3080	/
19×19	361	3173	3727	3302	3858	/
20×20	400	-	-	-	4667	/
25×25	625	-	-	-	8644	/
30×30	900	-	-	-	13724	/

Table 1. Results of Numerical Experiments for (n × n) Graphs.

Graph (n×m)	No. of Nodes	HAND-1	HAND-2	MIN.DEF.	MIN.DEG.	N.D.	M.N.D.
5×9	45	76	76	79	79	/	/
9×17	153	732	736	775	1020	/	836
10×20	200	1090	-	1133	1266	/	/
13×21	273	1651	-	1656	-	/	/
17×25	425	-	-	-	-	/	3900

Table 2. Results of Numerical Experiments for (n × m) Graphs.

Notes of Table 1 and 2.

1). Connectivity between nodes is the nine point difference scheme.

2). HAND-1 ; Hand reordering according to the results in this paper.

HAND-2 ; Hand reordering in Ref. [3].

MIN.DEF. ; Minimum Defficiency Algorithm.

MIN.DEG. ; Minimum Degree Algorithm.

N.D. ; Nested Dissection Method by Alan George.

M.N.D. ; Modified Nested Dissection Method by the authors.

From these tables and figures we can notice that

1). the new orderings obtained in this paper is not only simple but also very systematic,
2). they satisfy all the results obtained in previous section,
3). they can always give the minimum fill-in values for all test examples comparing with other methods,
4). the vertex orderings for all test examples are very similar to the ones by Nested Dissection Method, and the difference of the value of fill-in is caused by the difference of the dissections for four edges of the graphs, and
5). the arrangement of $\overline{\text{FVG's}}$ is rather simple till the width of a graph is less than 11, and new $\overline{\text{FVG's}}$ appear in order to subdivide a subgraph when the width is equal to 11. This fact indicates that further subdivisions by new $\overline{\text{FVG's}}$ appear when the width of a graph grows, and it coincides with the increase of "i" for G^i in the previous section.

From these results we can conclude that Nested Dissection Method by A. George is sufficient for grid graphs with square surrounding configurations, and that the method is easily applied not only to square graphs with n vertices on each edge but also to any rectangular graph subdivided into n by m with a slight modification, though original nested dissection method has a strict restriction for its applications.

5. Concluding Remarks

Through the theoretical and also experimental investigations on the fill-in minimization problem we obtained following results:

1). New model of the vertex elimination process is theoretically obtained, and it is proved by the numerical experiments only for simple graphs.
2). In the elimination process there appear only two types of vertex eliminations, and the elimination is successively done from a vertex which is appropriately selected in each subgraph enclosed by $\overline{\text{FVG}}$.
3). $\overline{\text{FVG's}}$ of a graph almost coincide with the dissection lines by Nested Dissection Method.
4). New model is thought as an extension of the dissection method.
5). $|\overline{\text{FVG's}}|$ are determined by the characteristics of the graph, though the dissection lines of George's method are independent from the characteristics.

This investigations can clarify only a portion of the fill-in minimization problem, and therfore, further study on this problem are required. Especially, the enumeration of $\overline{\text{FVG's}}$ for various types of graphs are desired for the proposal of valid method for the determination of the vertex elimination.