

また、この行を用いて、それ以下の小行列の上三角化を進めるとき、もし $k_{ik} = 0$, かつ $k_{jk} = 0$ ならば $\tilde{k}_{jk} = 0$. すなわち、各列において最初の非零要素より下側に位置する零要素のみ非零化する。このように、各列の最上非零要素と主対角要素が囲まれる領域をプロファイルと呼ぶ。(7)の類似性

K の帯領域¹⁾、プロファイル内において最も多くの非零要素をみられる。それらのうちの非零化要素は fill-in であり、その数は消去順序によって変化する。実際の計算に必要な要素は (節点数 + 部材数 + fill-in 数) であり、その和の最小化を図るには fill-in 数の最小化を行わねばならない。

いま n 点よりなる 1 個の構造系を考え、その行列を $K(n \times n)$ と書き、 K のグラフ表現を行う。 K に対し $\{1, 2, 3, \dots, n\}$ と番号を付けた n 個の点を準備する。

$$\begin{cases} \text{もし } k_{ij} \neq 0 \text{ ならば } V(i), V(j) \text{ 両点に 1 本の線が結ぶ。} \\ \text{もし } k_{ij} = 0 \text{ ならば } V(i), V(j) \text{ 両点で線が結ばない。} \end{cases} \quad (4)$$

ここで $V(i)$ は " i " と番号のついた点を示す。(4)の操作で K の上三角部 ($i=1 \sim n, j=i \sim n$) について行えば、 K の 1 つのグラフ表現 $G(n, m)$ を得る。ここで m は系に含まれる部材数 (力学的伝達路数) であり、 $K(n \times n)$ の非零小行列の数である。

K に対する消去演算をグラフ G において考えとみる。ここでグラフの距離、 d , を定義する。

$d(V(i), V(j)) = 1$ とは $k_{ij} \neq 0$, すなわち i, j 両点は 1 本の線が結ばれていることを示す。(3)ではグラフ的には

$$\begin{aligned} G \text{ において } d(V(i), V(j)) > 1 \text{ とも } d(V(i), V(k)) = d(V(j), V(k)) = 1 \text{ ならば} \\ G \text{ において } d(V(i), V(j)) = 1 \text{ とする。} \end{aligned} \quad (5)$$

ここで $\bar{G}$ は消去後のグラフを示す。すなわち $\bar{G}$ では i, j 両点間に新たな線が発生する。(5)式によれば、今消去しようとする点を含む部分グラフ G_S について、

$$\bar{G}_S = \{V(j) \mid d(V(i), V(j)) = 1\} = G_S \quad (\text{完全グラフ}) \quad (6)$$

すなわち、この点の消去後 $\{V(j) \mid d(V(i), V(j)) = 1\}$ の全ての点は互に直接結ばれる。 G_S と $G_S(\alpha, \beta)$ とあれば、消去後 $G_S \cup \{V(i)\}$ において、その線数 $\bar{\beta}$ は、

$$\bar{\beta} = \alpha(\alpha+1)/2 \quad (7)$$

従ってこの点消去において発生する新たな線 (fill-in) の数は

$$\text{fill-in 数} = \bar{\beta} - \beta = \alpha(\alpha+1)/2 - \beta \quad (8)$$

(8)式より、一般的に云、隣接する節点数 (これを次数: deg. と略す) の多い点 $V(i)$ を消去すれば、数多くの fill-in が発生する可能性があることを導かれる。しかしながら $V(i)$ を含む G_S が完全グラフに近ければ fill-in の発生は少ない。

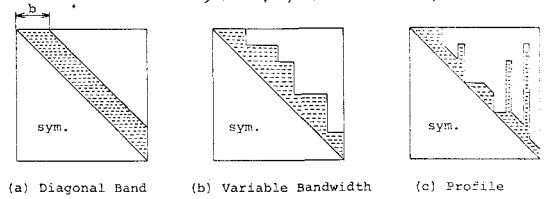
3. ラベリング問題 帯領域のプロファイルの最小化¹⁾のグラフ問題は NP-complete 問題の一つであり、有効なラベリングアルゴリズムの開発はほとんど不可能と考えられる。特に、節点数の分布が不均一²⁾であったり、あるいは円周²⁾である系の外周形状が複雑である場合、位相探索³⁾された帯領域減少法では不十分な結果しか期待しえない。しかしながら、

- (1). このような不十分なアルゴリズムでも、外周形状が簡単な系に対しては良い結果を導く。
- (2). 設定基本系の 1 種と考えられるトリグラフ²⁾に対するプロファイル最小化法はほぼ完成している。
- (3). トリグラフの各枝に重みを持たせると外周形状が複雑なグラフを得る。従って、プロファイル法は、このような系に対して有効になる。

以上の考察より、もし系の外周近形状が簡単な場合には帯行列法、複雑な場合にはプロポイル法を用いるべきであると考えられる。この理論を確かめるために、まず、帯行列・プロポイル型行列のグラフ表現を行う。

3.1 帯行列およびプロポイル型行列のグラフ表現

簡単な帯行列、プロポイル型行列は図-1に示される。図-2は帯行列のグラフ表現の例であり、



(a) Diagonal Band (b) Variable Bandwidth (c) Profile

いずれも(c)に示される4エーシ状グラフと考えられる。

図-1 帯行列およびプロポイル型行列

元の系が1個の連結グラフとして表現されることより、プロポイル行列は2つのタイプに分けられる。(図-3)

(I) 行列の主対角線と2帯行列を形成し、いくつかの行におり2は列方向への突出領域内にも非零要素が存在する。

(II) 行列のいくつかの行には、突出領域内に1が非零要素が存在しない。

(I),(II)をグラフ表現したものが図-4(b), (a)にそれぞれ示され、それぞれメッシュ構造、分岐を有するトリートメント構造をもつ。このように、非対角部の非零要素は、グラフの構造に大きな影響を及ぼす。

3.2 プロポイル法のラベリングとグラフ表現

ラベリング法は次のような3段階よりなる。

STEP 1. 系を切断して簡単な部分系の集合に置換する。

STEP 2. 各部分系を独立に扱い、ラベリングを行う。

STEP 3. ラベルした各部分系に対し一連のラベルを付け直す。

この3段階の為に、1.最適切断箇所設定法、2.簡単な部分系に対するラベリング法、3. renumbering 法が必要となる。

3.2.1 最適切断箇所設定法

一般的なグラフに対するアルゴリズム

これは現在実装してないが、トラス構造などの系に対しては下記のアルゴリズムで切断が可能である。なお、B.P.とは各点の等距離集合 D_i 中の境界点が初めて3個以上になる点に属し、 D_{i-1} から $d=1$ になる点(境界点)を云う。なお、 G の直径は既知とする。

STEP 1. 直径の1端より始め2 B.P.を探す。

STEP 2. G の直径を結ぶ境界点のみよりなる2つのパスのうち、一方を(+), 他方を(-)とする。

STEP 3. B.P. が(+)上にある場合は(-)に、(-)上にある場合は(+)に向って2 B.P. を出発点とする最短パスを探す。各B.P. の最短パスの短いものを切断線とする。

STEP 4. 切断線上にあるB.P. を出発点として、1, 2, 3の操作を繰り返して G を G_i の集合に置換する。

3.2.2 部分グラフに対するラベリング法

$V(A)$ の out-deg. とは $V(A)$ と隣接している D_{i-1} の点散

$V(A)$ ($\exists D_i$) の in-deg. とは $V(A)$ と隣接している D_{i-2} の点散をいう。アルゴリズムは次のようになる。

STEP 1. 直径の両端点の中で、out-deg. 最小の点を出発点として "n" と番号付ける。

STEP 2. D_i の点に番号を付ける場合は、 D_{i-1} の out-deg. の小さいものに隣接する D_i の点から付ける。

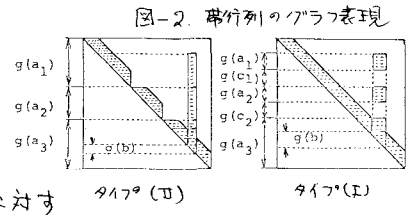


図-2 帯行列のグラフ表現

図-3 プロポイル型行列

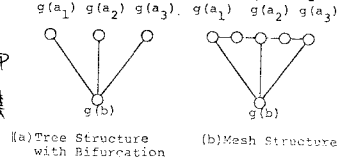


図-4 プロポイルのグラフ表現

STEP 3. out-deg. 最小のもの加いかつある場合は、その中 2 in-deg.

最小のものに番号を付ける。

STEP 4. 番号を付けた節点の in-deg. を消す。

STEP 5. 2, 3, 4 の操作を繰り返す。ただし番号は $n, n-1, \dots, 2, 1$ と付ける。

このアルゴリズムの適用例を図-5 に示す。この場合 Profile 領域は 2903, 2 のうち 2 要素は 2210。なお Band 3670 は図式帯幅減少法を用いた得られた帯領域の要素数 3670 であることを示す。

3.2.3 Renumbering 法

トリーグラフに対する場合

法則。このとき適用される。

図-5 に示されたプロセッサの適用例を図-6 に示す。左図は図式帯幅減少法による結果を示している。

4. 消去順序問題のグラフ理論的考察

最適節点消去順序

に關する研究は、D.J. Rose 氏によるものと除いておきたい。

彼らの考えた方法に於ては以下の 2 つがあげられる。

1. Minimum Degree Algorithm (Min. Deg. Al.)

2. Minimum Deficiency Algorithm (Min. Def. Al.)

1. は系をたらし出すグラフの各点の次数のうち、最小のものより順次消去を進めるという考えであり、(7)式に基準を置いた方法といえ、2. は(8)式に基準を置き、次に消去すべき点は(9)式の値を最小とする節点のうちより選ばれるという考えである。2. の方が、より良い結果を与えるが、一般には最適解を与えることはできず、例えは図-7A に示すように得られた消去順序は元の節点番号により大きく支配され、かつ非常に複雑な順序を与える。

しかしながら、現時点においては、Optimum Elimination Ordering

が全くわからない以上、Minimum Deficiency Algorithm を用いて、典型的なグラフの消去順序を求め、その結果に対して Min. Def. Al. の特性を考慮しながら、今一度 reordering を行う。この reordering の操作は手をもつて行う。以下に示す 2 つの目的：

1). fill-in 数は少くとも Min. Def. Al. を用いた時のその値より少ないか、等しいこと。

2). Reordering は、ある統一されたプロセッサで行い得ること。言い換えると、Min. Def. Al. の与える消去順序より複雑でないこと。

を満足するものがなければならぬ。

Band 3670
Profile 2903

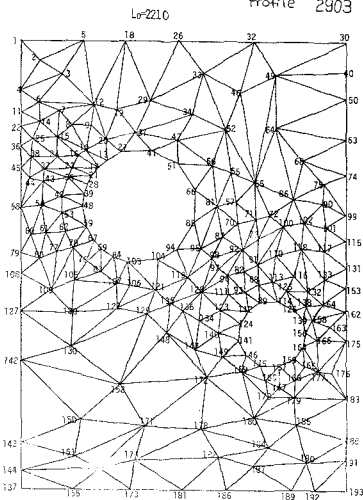


図-5. 3.2.2 のアルゴリズム適用例。

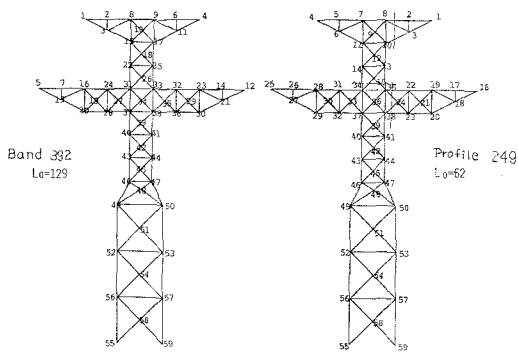
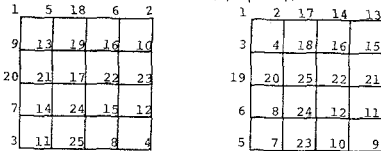
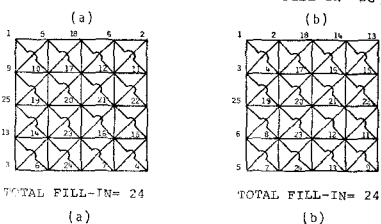
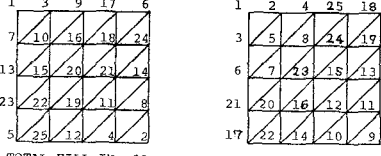


図-6. 3.2.2 のアルゴリズム適用例。



(a) (b)



(a) (b)

図-7. 節点消去順序の比較

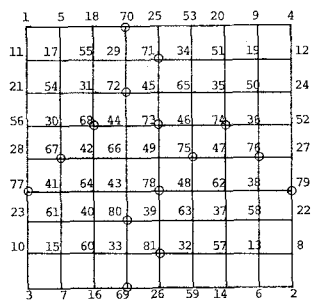
このような数値実験のために選ばれた例題が図-7に示される3種類のグラフである。上述のように Optimum Ordering が全く不明であることもあり、外形が最も簡単な正方形としており、点密度一定のグラフであり、変化させたのは線の配置のみである。

図-7の3つの結果に共通する注目すべき事柄は、1~25の消去順序のうち、最後の9個の番号(17~25)は全2. グラフを4分割する点に付くらしいということである。特に、例1及び例3のグラフにおいて正しく4等分割される。

この Min. Def. Al. による結果に対し、そのアルゴリズム固有の特性を考慮しながら、今一度手動で Reordering を行った結果は図-7bに示される。Reordering Procedure は、Min. Def. Al. により4分割された部分グラフを4つの部分に独立に経験に従って消去順序を与えるという方法がある。ちなみに、部分グラフ内には一連の番号が与えられる。この3例において、Reorder した結果の方が Fill-in が明らかに少ないという結果を得た。なお、この3例に限れば、Reorder した結果得られた消去順序は Optimum の一つであることが確認される。これより、Min. Def. Al. を適用した時現われる複推定 ordering は、少なくとも Min. Fill-in を与える ordering の特性ではなく、この手法の性質の一つであることがわかる。

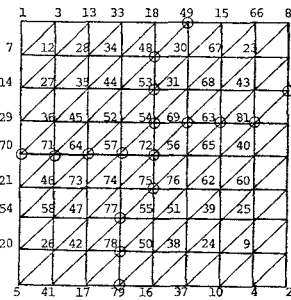
ここに示した Reordering Procedure を更に大きなグラフに適用した結果は図-8に示される。これも、前例と全く同様に、Reordering した結果の方が Min. Def. Al. の結果よりよいことは明らかである。

以上の諸例でもわかるように、対象とするグラフを4分割した部分グラフ内の全2の点消去の



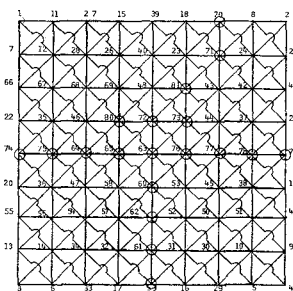
TOTAL FILL-IN = 246

(a) Elimination Ordering by Min. Def. Al.



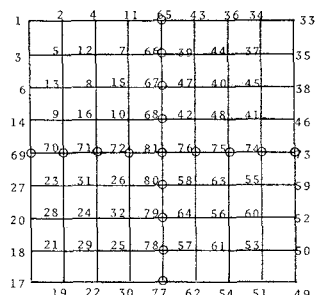
TOTAL FILL-IN = 276

(a) Elimination Ordering by Min. Def. Al.



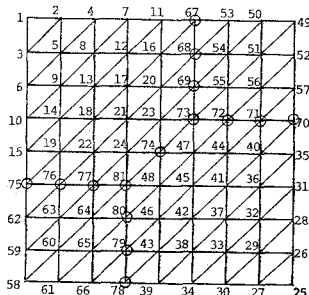
TOTAL FILL-IN = 316

(a) Elimination Ordering by Min. Def. Al.



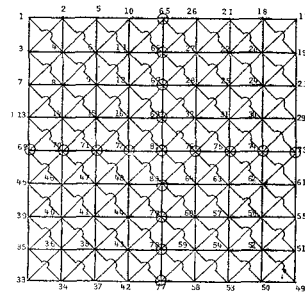
TOTAL FILL-IN = 250

(b) Elimination Ordering by Proposed Method



TOTAL FILL-IN = 267

(b) Elimination Ordering by Proposed Method



TOTAL FILL-IN = 272

(b) Elimination Ordering by Proposed Method

図-8. Min. Def. Al. による結果をもとに Reordering した3例題

終了後、部分グラフ間の境界を構成する節点が消去される。言い換えると、グラフの1点よりその点に隣接、もしくは近傍に位置する諸点を順次消去されるが、消去された領域の前面に位置する諸点（これは、それらのみで部分安全グラフを構成し、この G_c を Frontal Vertex Group (FVG と略す) と呼ぶことにする) の数が増える個数になる。このとき、それ以上その領域を拡大するような節点消去が行われず、他の領域内の点に次の消去順序が与えられる。この fill-in 最小化の本質であるように思われる。以下、この推論の裏付けを行う。

いま1つのグラフ G を考える。この G を対象として最適な節点消去がある段階まで進んだ時を考えると、 G は消去された点グループと未消去節点グループの2つの部分に分けられる。消去された節点グループに隣接する未消去節点グループに属する点は、1個以上の FVG を構成する。

G の中のいくつかの点グループを G より除くことにより、 G を2個の部分グラフに分断できるとき、この点グループを Separator, S , と呼ぶことにする。 G において S は数多く選り出すことができ、最大の Separator のうちの最小のものを Minimum Separator (M.S.) と呼ぶことにする。M.S. はグラフの最大中の最小値といえるものがある。このような M.S. と FVG には次のような関係が成り立つ。

$$\text{M.S. 内に含まれる節点数} \geq \text{1個の最大FVGに含まれる節点数} \quad (9)$$

すなわち、Vertex Elimination の Optimum である限りにおいて、Elimination Process において発生する FVG は、必ず対象となる G の最大中に含まれる節点数以下であることは、(9)式は示す。

上に示した諸例において、例えば 9×9 のメッシュ系 (図-8) においては、M.S. は9個の点を含むことになり、4隅より消去が進んだとき FVG のうちの最大の FVG は明らかに9個以下の点しか含まず、最大の FVG が (9) 式を満たす限界 (9個の点) に達した時点で他の領域に消去が移動する。このような臨界状態の最大の FVG の前面に位置する諸点が、全体系を4個の部分グラフに分断する節点グループを構成している。

このように (9) 式は上記諸例に共通した特性、すなわち G 内の消去は、 G を4分割して、各部分グラフについて独立に消去を行えばよいという事実の1つの裏付けを与えているが、それら部分グラフの消去が完了した時、すなわち残る点はいずれも2個以上の FVG に含まれることより残る Vertex Elimination は必然的に、その点の属する全ての FVG より新たに大きな1個の FVG を構成せしめ、従って、このような状態においては、(9)式は成り立たない。すなわち、このようにして複数の FVG の合併によって生じた1個の FVG に含まれる節点数は、明らかに M.S. に含まれる点数より多い。しかしながら、このような FVG の合併においては、新たな Fill-in の発生がみられるのは、合併する複数の FVG に含まれる節点のうち、互に共有しない諸点間においてであり、このような互に共有しない点のみを考慮すれば、このような FVG の合併時には (9) 式は成り立つといえる。このように、(9) 式は、本節が取り扱った諸例について成り立ち、従って4分割するという基本的な考え方の正当性を保障する。

この基本的な考え方を更に一般的グラフに適用した結果を、図-9に示す。それらの図中において () 内に示した値が Min. Def. Al. に与えられる Fill-in 数であり、Reordering により発生する Fill-in の方が少い。更に一般的系への適用例が図-10に示される。

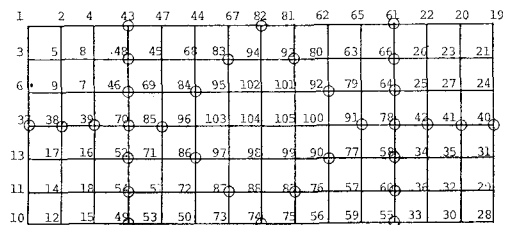
本節で示した様々な例題の結果、および(9)式を基にすれば、Fill-inを最小化(あるいは減少)させる目的での Vertex Elimination Ordering の基本的戦略は、

1. G を (9) 式を満たすように 11 個の部分グラフの集合に置換する。
2. 各部分グラフについて独立に節点消去順序の最適化を行う。
3. 部分グラフ間に位置する FIVs に含まれる点の消去順序を定める。

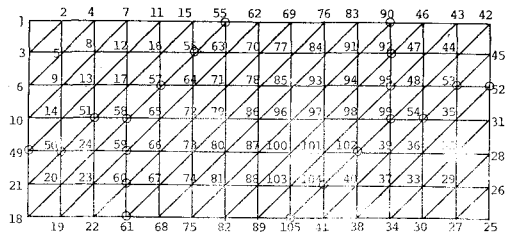
の3つよりなると考えられる。この考え方に従えば、各部分グラフ内の全ての点は順次引まつて消去されることになり、特に対称系が Δ 要素系 $\square$ 要素より構成される場合には、本節で扱った "簡単な外周近形状" を有する系に対する Profile 法の Δ Labeling Algorithm の適用が可能となる。また $\square$ 要素の系についても あるいは程度消去が進んだ段階で上記アルゴリズムの適用が可能となる。

5. あとがき 本報文中において、グラフ理論を基に、連立一次方程式の消去逆算プロセスにおける帯行列法、プロファイル法のための最適節点番号付け問題、スパースマトリクス法のための最適節点消去順序問題を取り扱った。前者に対しては、系の位相構造によって用いる解法を変えるという考え方が提案され、この提案によりいわゆる NP-complete 問題も扱い可能な問題に置換えらることを示した。同時に、実際の Labeling Algorithm の提案が行われた。後者に対しては、一種の最適化実験の結果に於けるグラフ理論的考察により、Fill-in 最小化のための基本的戦略を与えた。最小帯幅問題、最小プロファイル問題、最小プロファイル問題に共通する系の位相幾何学的なクォータは、(9)式で示されるようにグラフの中であることを判明した。しかしながら、依然多くの問題点未解決のままであり、今後の解決にあたり、本研究と同様にグラフ理論に基づくアプローチが必要である。

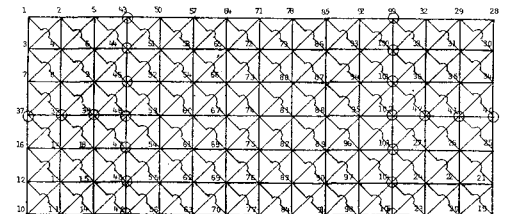
参考文献 1). Ch. H. Papadimitriou, "The NP-Completeness of the Bandwidth Minimization Problem", Computing 16, pp. 963-970, 1976
 2). 白石成久・谷口健男, "構造解析のための節点番号付け最適化手法について", 電算機利用に関心, ショーラウ, pp. 5-8, 1976
 3). 白石成久・谷口健男・殿本幸, "複雑な形状を有する構造物に対する節点番号付けアルゴリズム", 第3回電算機利用に関する学術大会, pp. 81-84, 1978
 4). D. J. Rose, "A Graph Theoretic Study of the Numerical Solution of Sparse Positive Definite Systems of Linear Equations", Graph Theory & Computing, pp. 183-217, 1972



TOTAL FILL-IN = 295 (312)



TOTAL FILL-IN = 335 (347)



TOTAL FILL-IN = 352 (396)

図-9. Reorderingによる消去順序

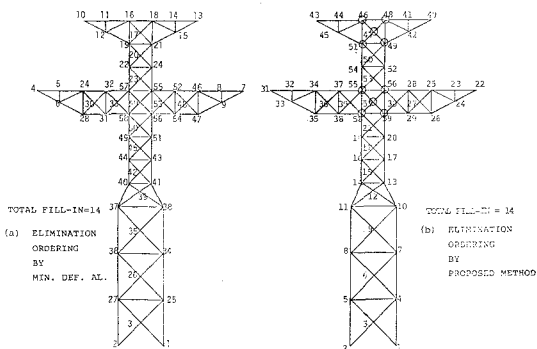


図-10. Min. Def. Al. と Reordering 法の比較

Application of Topological Characteristics of Structural Model to Matrix Methods

Takeo Taniguchi* Naruhito Shiraishi**

This study is concerned with the graph-theoretic approach to the minimization problems of bandwidth, profile and fill-in which are necessary for the effective utilization of the band matrix method, the profile method and the sparse matrix method, respectively. These minimization problems are evidently the combinatorial ones and it is already known that they belong to the Non-deterministic Polynomial-Time Complete Problem, that is, it may be almost impossible to give effective algorithms for these minimization procedures.

Though the profile method is a variation of the band matrix method, they are effectively applied to quite different two types of matrices; the band matrix method is used only for the band-formed matrix, and the profile method can treat non-banded one. A non-zero element in a structural coefficient matrix represents the existence of physical path connecting two nodes, and, therefore, the matrix is equivalently replaced by a graph. Hence, above two methods are effectively applied to different structural systems whose graphs are quite different, each other. Above consideration leads to the conclusion that we should select an effective numerical method in accordance with the type of graph obtained from the structural model. By the introduction of the additional procedure of selecting a numerical method before node-labeling for a structural model we can restrict the types of structures be treated by one numerical method and, hence, the authors could propose effective algorithm of node-labeling for the band matrix method and the profile method.

On the other hand, the optimum vertex elimination procedure which give the minimum value of fill-in is not known at present. The authors applied, at first, the minimum deficiency algorithm proposed by D.J. Rose to some graphs and they could give better results by reordering obtained vertex elimination processes taking into consideration of the properties of the algorithm. The results of these numerical experiments are summarized as follows:

1. Optimum vertex elimination may be done by treating a given graph as a gathering of some subgraphs.
2. The maximum width of a graph decides the maximum length of the cutting lines which tear a graph into subgraphs.
3. All the vertices in each subgraph are successively eliminated.

The most valuable result of this investigation is that the width of a graph representing a structural model is the common factor for the minimization problems of the bandwidth, profile and fill-in.

.....
* Department of Civil Engineering, Okayama University

** Department of Civil Engineering, Kyoto University