

スパースな連立方程式におけるピボット操作の順序づけ*

坂元 明雄** 白川 功*** 尾崎 弘***

Abstract

For a considerably large system, the coefficient matrix of a linear system of equations is frequently very sparse. Thus, in the computer-aided analysis of such a system of equations, of prime importance is the exploitation of the sparsity of the matrix in terms of reductions in memory space, processing time, and round-off errors. Motivated by this problem, in the present paper we first define a sparsity-oriented pivotal operation, and then discuss the problem of sparsity-preserving ordering of the pivotal operations with the use of a newly defined p-graph (pivoting graph).

1. ま え が き

行列において零要素の割合が多い場合、その行列はスパース行列 (sparse matrix) といわれる。スパースな係数行列をもつ系の解析においてそのスパースィティ (sparsity) を活用することは、計算時間の短縮、メモリの節約、あるいは丸め誤差の削減などの観点からきわめて有用である。これらの手法に関しては、近年いろいろな観点から考察されてきた¹⁾⁻⁴⁾。

このようなスパースィティを活用する問題は、大きく二つの問題に分けられる。その一つは、連立方程式を解く際の Gauss 消去に付随して生じる係数行列の LU 分解⁵⁾ の過程における、スパースィティを保存するようなピボット (pivot) 操作の順序決定問題である⁶⁾⁻⁹⁾。また他の一つは、原始インミタンス行列を係数行列にもつ回路方程式から、指定されたいくつか (通常二つ) の端子対に関する縮約方程式の係数行列を求める際に行なわれる、端子対縮約操作におけるスパースィティの活用問題である^{10), 11)}。

本論文では、スパースィティの保存性という点でこれらの二つの操作を同時に含むような一つの新しいピボット操作を定義し、まずこの操作の過程におけるスパースィティ保存の点で最適なピボット順序を決定す

る問題を定式化する。さらに新たに p-グラフを導入してピボット操作の擬似最適順序を得るための一つの手法を考察する。

2. 最適順序決定問題

n 個の未知数に関する独立な n 個の線形方程式、

$$W_0 |x\rangle = |b\rangle, \det W_0 \neq 0, \quad (1)$$

が与えられたとする。ただし、 $|x\rangle$ および $|b\rangle$ はそれぞれ n 次の列ベクトルとし、係数行列 W_0 の各対角要素は (もし必要ならば W_0 の行およびそれに対応する $|b\rangle$ の行の置換により) 零でないものとする。

正方向行列 W_0 の LU 分解とは、 W_0 を下三角行列 L および上三角行列 U の積の形に分解することである。ただし L は各対角要素がすべて 1 であるような下三角行列である。いま、 $W_0 = [w_{ij}^{(0)}]$ を初期値とし、 $k=1, 2, \dots, n$ に対して順次次の操作を行なう。

$$\left\{ \begin{array}{l} \langle w_h^{(k)} | = \langle w_h^{(k-1)} |, \\ h=1, 2, \dots, k, \end{array} \right. \quad (2.1)$$

$$\left\{ \begin{array}{l} |w_l^{(k)}\rangle = |w_l^{(k-1)}\rangle, \\ l=1, 2, \dots, k-1 \quad (k \geq 2), \end{array} \right. \quad (2.2)$$

$$\left\{ \begin{array}{l} w_{ik}^{(k)} = w_{ik}^{(k-1)} / w_{kk}^{(k-1)}, \\ i=k+1, k+2, \dots, n, \end{array} \right. \quad (2.3)$$

$$\left\{ \begin{array}{l} w_{ij}^{(k)} = w_{ij}^{(k-1)} - w_{ik}^{(k)} \cdot w_{kj}^{(k-1)}, \\ i, j \geq k+1. \end{array} \right. \quad (2.4)$$

ただし上式において、 $\langle w_h^{(k)} |$ および $|w_l^{(k)}\rangle$ はそれぞれ W_k の第 h 行および第 l 列を表わす。このようにして得られた行列 $W_n = [w_{ij}^{(n)}]$ に対して $L = [l_{ij}]$ および $U = [u_{ij}]$ は次式で与えられる。

* Ordering of Pivotal Operations on Sparse System of Equations, by Akio Sakamoto (Postgraduate student of Osaka University), Isao Shirakawa, and Hiroshi Ozaki (Faculty of Engineering, Osaka University)

** 大阪大学大学院

*** 大阪大学工学部

$$l_{ij} = \begin{cases} w_{ij}^{(n)} & ; i > j, \\ 1 & ; i = j, \end{cases} \quad (3)$$

$$u_{ij} = w_{ij}^{(n)}, \quad i \leq j. \quad (4)$$

式 (2.4) の操作は LU 分解における段階 k のピボット操作と呼ばれる。

一方、式 (1) において $|b\rangle = [b_1, b_2, \dots, b_p, 0, \dots, 0]^t$ (t は転置) である場合、原方程式 (1) から $|x'\rangle = [x_1, x_2, \dots, x_p]^t$ および $|b'\rangle = [b_1, b_2, \dots, b_p]^t$ に関する縮約方程式 $W'|x'\rangle = |b'\rangle$ の係数行列 W' を求めることがしばしば要求される。このときは W_0 に対して変数 $x_k (k \geq p+1)$ に関する次の操作を行えば $n-1$ 次の正方行列 $W_0^{(k)} = [w_{ij}^{(k)}]$ が求められる。

$$w_{ij}^{(k)} = w_{ij}^{(0)} - w_{ik}^{(0)} \cdot w_{kj}^{(0)} / w_{kk}^{(0)}, \quad i, j \neq k. \quad (5)$$

この操作を順次他の変数 $x_h (h \geq p+1)$ に対して $n-p$ 回行なうことにより p 次正方行列 W' を得ることができる。各段階におけるこの操作を端子対縮約操作という。

上に述べた LU 分解および行列の縮約のいずれの場合でも、その操作順序のいかにかわらざるを得ない行列は数式的には同一であることに注意すれば、これらの操作過程でスパースリティをできるだけ保存するような操作順序を決定することは、計算機実行の面からきわめて重要である。 LU 分解および行列縮約において零要素が非零要素にかわるのは、それぞれ式 (2.4) および (5) のピボット操作および端子対縮約においてのみであり、それらはそれぞれ $w_{ik}^{(k)} \cdot w_{kj}^{(k)} \neq 0$ および $w_{i,k}^{(h)} \cdot w_{k,j}^{(h)} \neq 0$ (ただし変数 x_k で行列縮約をした後変数 x_k に関して行列縮約を行なうものとする) のときである。そこで、任意の n 次正方行列 $W = [w_{ij}]$ ($\det W \neq 0, w_{ii} \neq 0$) に対して次のようなブール値を各要素にもつ行列 $X(W) = [x_{ij}]$ を定義する。

$$x_{ij} = \begin{cases} 1 & ; w_{ij} \neq 0, \\ 0 & ; w_{ij} = 0. \end{cases} \quad (6)$$

定義 1 m 次正方行列 W に対して、その行 k および列 k に関してなんらかの操作を行なってその行および列を取り去って得られた $m-1$ 次正方行列を $W^{**} = [w_{ij}^{**}]$ とする。このとき $X(W) = [x_{ij}]$ と $X(W^{**}) = [x_{ij}^{**}]$ との間に、

$$x_{ij}^{**} = x_{ij} + x_{ik} \cdot x_{kj}, \quad (i, j \neq k), \quad (7)$$

なる関係が成立するとき、この操作を k に関する広義のピボット操作という。ただし、ここでの加算および

乗算はブール代数におけるものとする。

先に述べた二つの操作のいずれもは明らかにこの広義のピボット操作の一つであり、以下では特にことわらないかぎりこの広義のピボット操作を単にピボット操作ということにする。

定義 2 正方行列 W の列 k および行 k に対して、それぞれ集合 A_k および集合 B_k を次式で定義する。

$$A_k = \{j | x_{jk} = 1, j \neq k\}, \quad (8)$$

$$B_k = \{i | x_{ki} = 1, i \neq k\}. \quad (9)$$

定義 3 集合 B_k の各元 i に対して、集合 L_{ki} を次式で定義する。

$$L_{ki} = \{j | x_{ji} = 0, j \in A_k, i \in B_k\} \\ = A_k - \{i\} - A_i. \quad (10)$$

$X(W)$ が対称行列であれば明らかに $A_k = B_k$ である。また定義 1 から、 $j \in L_{ki}$ なる元 j に対して $x_{ji} = 0$ かつ $x_{ji}^{**} = 1$ が成立することがわかる。すなわちこの場合 $w_{ji} = 0$ が、 k に関するピボット操作の結果 $w_{ji}^{**} \neq 0$ となることを意味している。ここで、 k に関するピボット操作によって零から新たに非零になる要素の個数を $\delta(W^{**})$ で表わすことにすれば、 $\delta(W^{**})$ は次の式で与えられる。

$$\delta(W^{**}) = \sum_{i \in B_k} |L_{ki}|. \quad (11)$$

ただし $|S|$ は集合 S の元の個数を意味する。

行列 W の n 個の行および列のうち、ピボット操作の行なわれるべき行および列の数が q 個のとき、それぞれは第 1 行 (列) から第 q 行 (列) までであるとして一般性を失わない。このとき、順次 q 回のピボット操作を行なう仕方は全部で $q!$ 通りある。いま、 q 次対称群 P の一つの元に対応する次のような系列の集合 S を考えよう。

$$S = \left\{ s = (\sigma_1, \sigma_2, \dots, \sigma_q) \mid \begin{pmatrix} 1 & 2 & \dots & q \\ \sigma_1 & \sigma_2 & \dots & \sigma_q \end{pmatrix} \in P \right\} \quad (12)$$

このとき各 $s = (k_1, k_2, \dots, k_q) \in S$ は一つのピボット操作の順序に対応する。ある $s \in S$ に対応する順序でピボット操作を行なう過程において、零から非零に変わる要素の個数 $\Delta(s)$ は、

$$\Delta(s) = \delta(W_0^{**k_1}) + \delta(W_0^{**k_1 k_2}) + \dots \\ + \delta(W_0^{**k_1 k_2 \dots k_q}), \quad (13)$$

で与えられる。ここで、

$$\delta(W_0^{**k_1 k_2 \dots k_h}) = \delta(\{ \dots (\dots (W_0^{**k_1})^{**k_2}) \dots \}^{**k_h}), \\ h = 2, 3, \dots, q, \quad (14)$$

である。

以上のことから最適順序決定問題は、次の式を満たすような最適順序 s^* を求める問題に帰着される。

$$D(s^*) = \min_{s \in S} D(s). \quad (15)$$

すなわち最適順序 s^* は、最もスパースィティを保存するようなピボット順序である。ただしこのような s^* はただ一つであるとは限らない。

3. p-グラフ

与えられた正方形行列 W に対して p-グラフ $G=(V(G), E(G))$ を次のように定義する。ただし $V(G)$ は G の節点の集合、 $E(G) \subset V(G) \times V(G)$ は G の枝の集合を表わす。

定義 4 m 次正方形行列 W の p-グラフ $G=(V(G), E(G))$ は次のような有向グラフである。

- (i) 各節点 $v_i \in V(G)$ ($i=1, 2, \dots, m$) は W の第 i 列に対応する。
- (ii) W の各非零要素 $w_{ij} \neq 0$ ($i \neq j$) には、 v_i から v_j に向かう枝 $(v_i, v_j) \in E(G)$ が対応する。
- (iii) 各 $v_i \in V(G)$ および各 $(v_i, v_j) \in E(G)$ は、それぞれ集合 A_i および集合 L_{ij} を重みとしてもつ。

例 1

行列 W_0 に対する行列 $X(W_0)$ が次式で与えられているとする。ただし ‘ \cdot ’ は 0 を表わす。

$$X(W_0) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & \cdot & \cdot & \cdot & 1 & 1 \\ 2 & \cdot & 1 & \cdot & 1 & \cdot \\ 3 & 1 & \cdot & 1 & 1 & \cdot \\ 4 & \cdot & 1 & \cdot & 1 & 1 \\ 5 & 1 & \cdot & \cdot & 1 & 1 \\ 6 & 1 & 1 & 1 & \cdot & \cdot \end{pmatrix}. \quad (16)$$

この W_0 に対応する p-グラフ G_0 は図 3.1 で示される。図中では $\{ \cdot \}$ および $[\cdot]$ は、それぞれ節点および枝の重みの集合を表わす。

行列 W の p-グラフを G とし、 W^{**} の p-グラフを $G^{**}=(V(G^{**}), E(G^{**}))$ とする。また節点の集合 $\Gamma_k(G) = \{v_i | (v_k, v_i) \in E(G)\}$ および $\Gamma_k^{-1}(G) = \{v_j | (v_j, v_k) \in E(G)\}$ を定義する。まず、 G^{**} が G から求められることを示そう。

補題 1 $G^{**}=(V(G^{**}), E(G^{**}))$ は $G=(V(G), E(G))$ に対して次の式で与えられる。

$$\begin{aligned} V(G^{**}) &= V(G) - \{v_k\}, & (17) \\ E(G^{**}) &= (E(G) - \bigcup_{v_k \in \Gamma_k^{-1}(G)} \{(v_k, v_k)\}) \\ &\quad - \bigcup_{v_i \in \Gamma_k(G)} \{(v_k, v_i)\} \cup \{ \bigcup_{i \in B_k} H_i \}. & (18) \end{aligned}$$

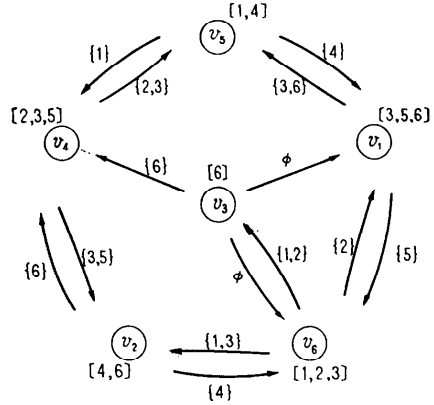


図 3.1 p-graph G_0

ただし、

$$H_i = \begin{cases} \phi & ; L_{ki} = \phi, \\ \{(v_j, v_i) | j \in L_{ki}\} & ; L_{ki} \neq \phi. \end{cases} \quad (19)$$

である。

証明 k に関してピボット操作を行なうのであるから、式 (17) は明らかである。一方 G^{**} は、 G において v_k に接続している枝以外のすべての枝を含むから、

$$\begin{aligned} E(G^{**}) \supset E(G) - \bigcup_{v_k \in \Gamma_k^{-1}(G)} \{(v_k, v_k)\} \\ - \bigcup_{v_i \in \Gamma_k(G)} \{(v_k, v_i)\}, \end{aligned} \quad (20)$$

は明らかに成立する。さらに $E(G)$ に含まれない枝で $E(G^{**})$ に新たに加えられるべき枝は、 W において零であったが k に関するピボット操作の結果非零になった要素に対応する枝である。このような枝は $U_i \in B_k$ H_i で与えられるから、結局式 (18) が得られる (証明終)。

補題 2 W^{**} における各集合 A_i^{**}, B_i^{**} および L_{ij}^{**} は、 W における集合 A_i および B_i から次式で求められる。

$$A_i^{**} = \begin{cases} A_i \cup A_k - \{i, k\} & ; i \in B_k, \\ A_i & ; i \notin B_k, \end{cases} \quad (21)$$

$$B_i^{**} = \begin{cases} B_i \cup B_k - \{i, k\} & ; i \in A_k, \\ B_i & ; i \notin A_k, \end{cases} \quad (22)$$

$$L_{ij}^{**} = A_i^{**} - \{j\} - A_j^{**}. \quad (23)$$

(証明略)

例 2

式 (16) の W_0 の第 1 行 (列) に関してピボット操作を行なった結果、図 3.2 に示すような W_0^{*1} の p-グラフ G_0^{*1} が得られる。

p-グラフ G において、節点 $v_k \in V(G)$ に関する値 $M_k(G)$ を次のように定義する。

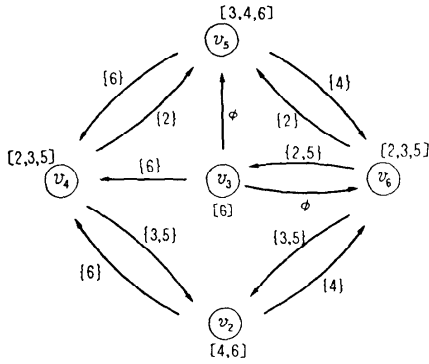


図 3.2 p-graph \$G_0^{*1}\$

$$M_k(G) = \sum_{v_i \in \Gamma_k(G)} |L_{ki}|. \tag{24}$$

\$W_0\$ の p-グラフ \$G_0\$ が与えられたとき式 (11) から、次のことがわかる。

$$M_{ki}(G_0) = \delta(W_0^{*k_i}), \tag{25}$$

$$M_{k_2}(G_0^{*k_1}) = \delta(W_0^{*k_1 k_2}), \tag{26}$$

\$W_0\$ の各行および列のうち、ピボット操作の行なわれべき行および列に対応する \$G_0\$ の節点の集合を \$U_0\$ とする。さらに、\$v_i \in U_0\$ なる \$v_i\$ で \$M_i(G_0) = 0\$ であるような \$G_0\$ の節点の集合を \$U_1\$ とする。明らかに \$U_1 \subset U_0\$ である。いま、\$U_1 = \{v_{i_1}, v_{i_2}, \dots, v_{i_r}\}\$ とすると、系列 \$(i_1, i_2, \dots, i_r)\$ の任意の置換 \$s_a = (a_1, a_2, \dots, a_r)\$ を考えるとき、

$$\delta(W_0^{*a_1}) + \delta(W_0^{*a_1 a_2}) + \dots + \delta(W_0^{*s_a}) = 0, \tag{27}$$

である。ただし、

$$\delta(W_0^{*s_a}) = \delta(W_0^{*a_1 a_2 \dots a_r}), \tag{28}$$

とする。同様に、\$v_j \in U_0 - U_1\$ なる \$v_j\$ で \$M_j(G_0^{*s_a}) = 0\$ であるような \$G_0^{*s_a}\$ の節点の集合を \$U_2 = \{v_{j_1}, v_{j_2}, \dots, v_{j_s}\}\$ とし、系列 \$(j_1, j_2, \dots, j_s)\$ の任意の置換 \$s_b = (b_1, b_2, \dots, b_s)\$ を考えれば、

$$\delta((W_0^{*s_a})^{*b_1}) + \delta((W_0^{*s_a})^{*b_1 b_2}) + \dots + \delta(W_0^{*s_a s_b}) = 0, \tag{29}$$

である。ただし、

$$\delta(W_0^{*s_a s_b}) = \delta((W_0^{*s_a})^{*s_b}), \tag{30}$$

である。この操作をくり返すことによって、次の式を満足するような系列 \$s_c = (c_a, c_b, \dots, c_d)\$ が得られる。すなわち、

$$\delta(W_0^{*a_1}) + \delta(W_0^{*a_1 a_2}) + \dots + \delta(W_0^{*s_a}) + \delta(W_0^{*s_a b_1}) + \delta(W_0^{*s_a b_2}) + \dots + \delta(W_0^{*s_a s_b}) + \dots + \delta(W_0^{*s_c}) = 0, \tag{31}$$

かつ、任意の \$v_i \in V(G_0^{*s_c}) \cap U_0\$ に対して、

$$M_i(G_0^{*s_c}) \neq 0. \tag{32}$$

このような \$s_c\$ に関して、次の定理が成立する。

定理 式 (15) を満足する最適順序 \$s^*\$ の集合を \$S^* = \{s_i^* | i \in S\}\$ とする。また、式 (31) および (32) を満足する系列を \$s_c\$ とすれば、\$s_c\$ および、\$G_0^{*s_c}\$ に対する任意の最適順序 \$s_{j_c}^+\$ を部分系列とする \$s_j^+\$、すなわち、

$$s_j^+ = (s_c, s_{j_c}^+), \tag{33}$$

の集合 \$S^+ = \{s_j^+ | j \in S\}\$ は \$S^*\$ の部分集合である。

証明 この定理を証明するためには、もし \$M_i(G) = 0\$ ならば \$M_i(G^{**}) = 0\$ (\$k \neq i\$) であること、すなわちすべての \$v_k \in \Gamma_i(G)\$ に対して \$L_{ik} = \phi\$ であるような \$G\$ の任意の節点 \$v_i\$ は、\$G^{**}\$ においてもすべての \$v_k \in \Gamma_i(G^{**})\$ に対して \$L_{ik}^{**} = \phi\$ であることを示せば十分である。いまそのような \$v_i\$ が \$v_i \in \Gamma_k(G)\$ であるとすれば、\$A_i^{**} = A_i\$ であるから \$G^{**}\$ においてすべての \$v_k \in \Gamma_i(G^{**})\$ に対して \$L_{ik}^{**} = \phi\$ であることは式 (23) から明らかである。したがって \$v_i \in \Gamma_k(G)\$ の場合のみを考えればよい。このときもし \$L_{ki} = \phi\$ であれば \$A_i^{**} = A_i - \{k\}\$ となり、上の場合と同様に \$G^{**}\$ においてすべての \$v_k \in \Gamma_i(G^{**})\$ に対して \$L_{ik}^{**} = \phi\$ である。一方、\$L_{ki} \neq \phi\$ であれば、\$v_k \in \Gamma_i(G)\$ なる \$v_k\$ に対して \$(v_k, v_k) \in E(G)\$ である。なぜなら \$(v_k, v_k) \in E(G)\$ ならば \$w_{kk} = 0\$ であり、いま \$v_i \in \Gamma_k(G)\$ であるから \$w_{ki} \neq 0\$ である。すなわち \$k \in A_k\$ かつ \$k \in A_i\$ であるから式 (10) より \$k \in L_{ik}\$。これは \$L_{ik} = \phi\$ に反する。したがって \$v_k \in \Gamma_i(G)\$ なる \$v_k\$ に対して \$(v_k, v_k) \in E(G)\$ であり、これは \$L_{kk}\$ なる集合が定義されることを意味する。そこで \$j \in L_{ki}\$ なる \$j\$ に対して \$j \in L_{kk}\$ であるとする。このとき \$\{A_i \cup \{j\} - \{k\}\} \subset A_i^{**}\$ かつ \$j \in A_k\$ であるから、\$L_{ik}^{**} = A_i^{**} - \{k\} - A_k^{**}\$ には \$j\$ は含まれない。また \$j \in L_{ki}\$ なる \$j\$ に対して \$j \in L_{kk}\$ であるときは、\$A_i^{**}\$ と \$A_k^{**}\$ にはともに \$j\$ が含まれるからやはり \$j \in L_{ik}^{**}\$ となる。したがって \$v_i \in \Gamma_k(G)\$ のときも、\$v_k \in \Gamma_i(G)\$ なるすべての \$v_k\$ に対して \$G^{**}\$ では \$L_{ik}^{**} = \phi\$ となる (証明終)。

この定理から、最適順序 \$s^* \in S^*\$ を求める問題は、式 (33) における系列 \$s_{j_c}^+\$ を求める問題と考えることができる。

4. 擬似最適順序

\$U_c = U_0 \cap V(G_0^{*s_c})\$ とし、\$q = |U_c|\$ とする。最適

順序 s_j^+ の部分系列 $s_{j^*}^+$ を求めるとき、たとえば branch and bound 法などを用いても、最悪の場合すべての $q!$ とおりの系列 $s_i = (\sigma_1, \sigma_2, \dots, \sigma_q), v_i \in U_s$ について各 $\Delta(s_i, s_i)$ を求めなくてはならない。そこで必ずしも最適である保障はないが、できるだけ少ない手数でしかも最適に近い系列を得る方法が要求される。このスパーシティを保存するピボット操作の擬似最適順序を決定する問題に関しては、次の三つの手法が考えられている^{8),9)}。

手法 1 行列 W_0 の式 (9) で定義される各行 k に対する各集合 B_k に、

$$|B_{k_1}| \leq |B_{k_2}| \leq \dots \leq |B_{k_p}| \quad (34)$$

なる関係があるとき、ピボット順序を $s = (k_1, k_2, \dots, k_p)$ とする。

手法 2 行列 W_0 の $|B_k|$ の値が最小である行を k_1 とする。次に $W_0^{*k_1}$ において $|B_k|$ の値が最小となる行を k_2 とする。以下同様にして、 $W_0^{*k_1 k_2 \dots k_i}$ において $|B_k|$ の値の最小となる行を順次 k_{i+1} とし、ピボット順序 $s = (k_1, k_2, \dots, k_p)$ を決定する。

手法 3 行列 W_0 で $\delta(W_0^{**})$ の値を最小にする行を k_1 とする。次に $W_0^{*k_1}$ において $\delta(W_0^{*k_1 k})$ の値が最小となる行を k_2 とする。以下同様にして、 $W_0^{*k_1 k_2 \dots k_i}$ において $\delta(W_0^{*k_1 k_2 \dots k_i k})$ の値の最小となる行を順次 k_{i+1} とし、ピボット順序 $s = (k_1, k_2, \dots, k_p)$ を決定する。

上の三つの手法のうち、手法 1 は行列 W_0 にのみ着目してピボット順序を決定するものであり、ピボット操作の過程における非零要素の影響を考慮しない最も単純な方法である。また、手法 1 および 2 では各行の非零要素の個数に着目するが、手法 3 においてはピボット操作による非零要素の増加に着目している。これらの手法の優劣は、与えられた行列の形態（すなわち非零要素の分布）あるいは行列のサイズなどにより異なるが、一般には手法 3 がより有効である。ところが、手法 3 で k_{i+1} を定めるとき、 $\delta(W_0^{*k_1 k_2 \dots k_i k})$ の最小値を与える k が 2 個以上存在したとき、それらのどの k を選ぶかは任意であるが、その選び方が最終的に $\Delta(s)$ の値に影響する場合がある。このような観点から、ここでは s_j^+ に対する擬似最適順序 $s_{j^*} = (s_1, s_{j^*})$ を与える一つのアルゴリズムを考察する。

操作 1 与えられた正方行列 W_0 に関して、 $G \leftarrow G_0, U \leftarrow U_0, \pi \leftarrow \lambda$ (null sequence), $\Delta \leftarrow 0$ として次の操作へ。

操作 2 もし $U = \phi$ であれば操作完了。 $U \neq \phi$ の

とき次の操作へ。

操作 3 集合 $V_1 = \{v_k | M_k(G) = 0, v_k \in U\}$ を求める。 $V_1 = \phi$ のとき操作 4 へ。 $|V_1| = t \neq 0$ のときは、各 $v_{k_i} \in V_1$ に対する k_i の任意の系列を $s = (k_1, k_2, \dots, k_t)$ とし、 $\pi \leftarrow (\pi, s), G \leftarrow G^{**}, U \leftarrow U - V_1$ として操作 2 へ。

操作 4 次式の値を与える $v_k \in U$ を元とする集合 V_2 を求める。

$$\min_{v_k \in U} \left\{ \sum_{v_i \in \Gamma_k(G)} |L_{k_i}| + |\Gamma_k(G)| \right\}. \quad (35)$$

$|V_2| > 1$ のとき次の操作へ。 $|V_2| = 1$ ならば操作 7 へ。

操作 5 次式の値を与える $v_k \in V_2$ を元とする集合 V_3 を求める。

$$\max_{v_k \in V_2} \left\{ \bigcup_{v_i \in \Gamma_k^{-1}(G)} L_{i,k} \right\}. \quad (36)$$

$|V_3| > 1$ のとき次の操作へ。 $|V_3| = 1$ ならば操作 7 へ。

操作 6 次式の値を与える $v_k \in V_3$ を元とする集合 V_4 を求める。

$$\min_{v_k \in V_3} \left\{ \sum_{v_j \in \Gamma_k(G)} \left\{ \sum_{h \in L_{k_j}} |L_{h,j}^{**}| \right\} \right\}. \quad (37)$$

V_4 の任意の元を v_k として次の操作へ。

操作 7 $\pi \leftarrow (\pi, k), \Delta \leftarrow \Delta + M_k(G), G \leftarrow G^{**}, U \leftarrow U - \{v_k\}$ として操作 2 へ。

以上の操作が完了したとき、 $s_{j^*} = \pi, \Delta(s_{j^*}) = \Delta$ となっている。これらの過程において、操作 2 および 3 は $s_{j^*} = (s_1, s_{j^*})$ の部分系列 s_1 を求めるための手続きであり、いったん操作 4 にはいるとそれ以降の操作は部分系列 s_{j^*} を求めるための手続きになる。

式 (35) で $|\Gamma_k(G)|$ は p -グラフ G において節点 v_k から出ている枝の個数であり、 $\sum_{v_i \in \Gamma_k(G)} |L_{k_i}| = M_k(G)$ はそれらの枝の重みの元の個数の総和である。また式 (36) は、節点 v_k にはいつている枝の重みの和集合の元の個数を比較することを意味し、式 (37) の場合とは逆にその値が最も大きい節点でピボット操作を行なうほうがよいことを示している。なぜなら、節点 v_k でピボット操作を行なうことにより、 v_k に接続しているすべての枝は G から取り除かれる。このとき、 v_k から出る枝の重みの元の個数だけ G^{**} では枝が増加するが、 v_k にはいる枝の重みは G^{**} における枝の増加に寄与しない。したがって、式 (35) では最小値を与える節点を選び、式 (36) では最大値を与える節点を選ぶようにする。式 (37) は、 v_k でピボット操作を行なった結果 G^{**} に新たに加わる枝 (v_k, v_j) の重み $L_{k_j}^{**}$ の元の個数の総和を求め、その

値が最小である v_8 を選ぶことを意味している。したがって操作6では、2段階後の非零要素の増加も考慮していることになる。

例 3

正方行列 W_0 に対する $X(W_0)$ が次の式で与えられたとする。

$$X(W_0) = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 \\ \cdot & 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 \\ \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & 1 \\ \cdot & 1 & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 & 1 & 1 & 1 & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & 1 \\ 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & 1 \end{pmatrix} \end{matrix} \quad (38)$$

いま $U_0 = \{v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ として、先に述べた手法3で擬似最適順序を求めると、 $s = (3, 6, 4, 8, 9, 5, 7)$ となり $\Delta(s) = 14$ である。これに対して、上で考察したアルゴリズムにより得られる擬似最適順序は、 $s_j^* = (3, 6, 8, 9, 4, 5, 7)$ となり $\Delta(s_j^*) = 12$ である。なお、この場合 $s_j^* \in S^*$ である。この例で3番目の節点 v_8 を選ぶ場合について考えてみる。いま $G = G_0^{*3,6}$ で $M_4(G)$ の最小値は4、 $M_4(G) = M_6(G) = M_9(G) = 4$ である。手法3では v_8 として v_4, v_8, v_9 の任意のものを選ぶから $v_8 = v_4$ とする。一方上のアルゴリズムでは、 $V_2 = V_3 = \{v_4, v_8, v_9\}$ であり、操作6で $V_4 = \{v_8, v_9\}$ となる。したがって $v_8 = v_8$ とする。この段階では $M_4(G) = M_6(G)$ であるから、 Δ の値はともに4だけ増加する。ところが $G_1 = G_0^{*3,6,4}$ を考えたとき、その後どのような系列を選んでも非零要素の増加量は4以上になるが、 $G_2 = G_0^{*3,6,8}$ では、非零要素の増加量が2であるような系列（この場合 $(9, 4, 5, 7)$ ）を得ることができる。

与えられた係数行列 W_0 に対する $X(W_0)$ が対称行列である場合について、それらの擬似最適順序 s_j^* を求めるプログラムを作成し、branch and bound 法による最適順序 s_j^+ を求めるプログラムと比較した結果が表4.1である。表中の n は $X(W_0)$ のサイズ、 q はピボット操作の行なわれるべき行および列の個数 $|U_0|$ を表す。なお使用機種はNEAC-2200-500である。

5. むすび

本論文ではまず正方行列の LU 分解および行列縮

表 4.1 The comparison of the near-optimal ordering with the optimal ordering (' : minute; " : second).

datum No.	n	q	optimal		near-optimal	
			CPU time	Δ	CPU time	Δ
1	8	6	30"	4	30"	4
2	10	8	3'56"	4	30"	4
3	11	9	12'55"	20	31"	20
4	11	9	22'37"	14	31"	14
5	11	9	37'53"	22	31"	22
6	12	10	24'30"	28	36"	28
7	14	12	—	—	32"	20
8	17	15	—	—	37"	20
9	28	26	—	—	48"	66

約を、スパースィティを保存するという点で同時に含むような広義のピボット操作を定義し、このピボット操作において最適順序決定問題を定式化した。さらに擬似最適順序を得る一つの手法について考察した。

実際に計算機によりスパースな係数行列を取り扱う場合、本論文の方法で得られた擬似最適順序に従って操作を行えば、計算手数やメモリなどが節約される。特に係数行列のサイズが大きいくほど、非零要素の占める割合は小さくなり、計算機に実行させる際にそれらの非零要素をリスト構造にして記憶させることが多い⁶⁾。このとき本論文で述べた手法を用いれば、あらかじめどの非零要素が非零要素に変換されるかがわかるので、それらの要素のためのメモリをまえて用意しておくことができる。

また、回路方程式などのように係数行列の各要素の値があるパラメータ（たとえば周波数）の関数となっているような場合、これらの多数のパラメータの値に関して縮約行列を求めなければならないことが多い。このようなときまえて、その操作順序を上手法で定め、それに従って計算すれば全体の計算手数をかなり節約することができる。

おわりに、有益なご示唆をいただいた査読者に感謝の意を表す。

参考文献

- 1) F.H. Branin, Jr.: "Computer methods of network analysis", *Computer Oriented Circuit Design*, F. F. Kuo and W. G. Magnuson, Jr., ed., Prentice-Hall, N. J., (1969).
- 2) J. K. Reid, ed.: "Large Sparse Sets of Linear Equations", *Academic Press*, N. Y., (1971).
- 3) R. P. Tewarson: "Computations with sparse matrices", *SIAM Review*, 12, pp. 527-543 (1970).

- 4) R. A. Willoughby, ed.: "Proceedings of the Symposium on Sparse Matrices and Their Applications", Rep. RA 1 (11707), IBM, Yorktown Heights, N. Y., (1969).
- 5) G. E. Forsythe and C. B. Moler: "Computer Solution of Linear Algebraic Systems", Prentice-Hall, N. J., (1967).
- 6) R. D. Berry: "An optimal ordering of electronic circuit equations for a sparse matrix solution", *IEEE Trans. Circuit Theory, CT-18*, pp. 40-50 (1971).
- 7) F. G. Gustavson, W. M. Liniger and R. A. Willoughby: "Symbolic generation of an optimal Crout algorithm for sparse systems of linear equations", Rep. RC-1852, IBM, Yorktown Heights, N. Y., (1967).
- 8) E. C. Ogbuobiri, W. F. Tinney and J. W. Walker: "Sparsity-directed decomposition for Gaussian elimination on matrices", *IEEE Trans. Power App. Syst., PAS-89*, pp. 141-150 (1970).
- 9) W. F. Tinney and J. W. Walker: "Direct solutions of sparse network equations by optimally ordered triangular factorization", *Proc. IEEE*, 55, pp. 1801-1809 (1967).
- 10) H. Ariyoshi, I. Shirakawa and H. Ozaki: "Decomposition of a graph into compactly connected two-terminal subgraphs", *IEEE Trans. Circuit Theory, CT-18*, pp. 430-435 (1971).
- 11) 戸川, 白川, 尾崎: "Z 行列におけるブロックごとの端子対縮約のための回路網分解", *信学論(A)*, 54-A, pp. 338-344 (1971).
(昭和 46 年 9 月 30 日受付, 12 月 14 日再受付)