UDC 621.3.012: 512.83
621.311.1

論 文

47-B 8

大規模回路網トポロジ処理手法

電子技術総合研究所

佐 藤 弘 之

1. ま え が き

電力系統の過渡的過電圧ならびに過電流特性の電子計算機による模擬は、電源容量ならびに系統規模の著しい増大傾向に伴う送電設備または各種機器の大容量超高圧化と、それらの合理的設計による絶縁裕度の低減をはかるという観点から、ますます重要な技術的検討項目となりつつある。

電子計算機による電力系統または機器の等価回路網に対する過渡特性計算は、従来からよく使用されている実測または縮少モデルによる実験、または過渡現象計算盤、模擬送電線、などのアナログ形計算装置と比較してモデルの定数または係数の精度ならびにその数値の選択範囲に制約のないこと、モデルの理想化ならびにモデルサイズを弾力的に考慮できること、アナログ素子では原理的に不可能とされる模擬方法も容易に表現可能であること、計算結果の再現性がすぐれていること、入出力データの変換が可能であること、計算費用が低廉であることなど多数の長所を備えている。

しかし、電力系統ならびに各種機器により構成されるモデル回路網の表現を簡略化するにつれて過渡的過電圧倍数が過酷になり、波形ひずみが現われやすいなどの傾向がある。その抑止対策として非常に高速度な現象もかなり忠実に模擬しようとする、一般に回路網モデルはきわめて大規模でしかも複雑なものとなりやすい。

回路網の過渡特性解析のための数学的モデルには、数値計算または定式化に便利のように選ばれた状態変数に関する多元連立一階常微分方程式が使用される。大規模な回路網モデルでは数学的モデルの定式化過程にグラフ理論を応用するのが普通である。しかし、そ

の過程における演算操作処理はかなり複雑なため、電子計算機による完全自動的定式化処理を施さない限り数学的モデルの信頼性を確保することが困難である。

グラフ理論は回路素子を抽象化した回路網グラフの行列表示法により理論的に展開されているが、これをそのままの形式で計算機処理方式を設計することは高速メモリーの容量制約のためシステム規模の拡大をはかるのが困難となりやすい。

しかし、グラフ理論で使用されるほとんど全ての行列は回路網規模が大きくなるにつれて行列全体に対する零要素の比が多くなり、非零要素の比は少くなる傾向がある。しかも、非零要素の数値もたいていの場合 ± 1 の二種類のみ考慮できればよい。過渡解析の数学的モデルの定式化に必要とされる回路網グラフの構造と、状態変数との関係を記述する行列は、それらを導出するのに複雑な演算操作と多数の高速メモリーを必要とするが、きわめて基本的な二、三の行列の複合形式で表現することができる。^{(1)~(7)}

したがって、大規模回路網の過渡特性解析を実用的規模の計算機で可能な限り、高速メモリーの範囲内で計算させるためには、上述した接続データならびに行列の特長をとり入れた疎行列処理手法(Sparse matrix programming technique)をじゅうぶんに活用してデータ用に必要なメモリー数が非常に少なく、そしてその利用法が簡単な回路網トポロジ処理システムを開発することがきわめて重要な課題である。

本文には大規模な電力系統ならびに機器の実用的過渡特性計算に対して、所要メモリー数が非常に少なくしかも計算処理時間の短い回路網トポロジ処理システムとそのデータ処理方式について述べる。

2. 回路網グラフの計算機処理

回路網の過渡特性解析方法にはループ電流を状態変数としたタイセット解析法、ノード対電圧を状態変数

Sparse Matrix Technique for Applying Topological Methods to Large-Scale Electrical Networks. By H. SATO, Member (Electrotechnical Laboratory, Energy Division)
佐藤弘之：正員，電子技術総合研究所エネルギー部

としたカットセット解析法, ならびにその両者の長所をとり入れた混合解析法がある。混合解析法は状態変数の数が他と比較して少なくなるばかりでなく状態方程式の定式化を無理なく円滑に進めることができる利点がある。混合解析法における状態方程式の定式化の際には回路網グラフの接続と状態変数との関係を記述するのにハイブリッド行列 (Hybrid matrix) A ならびにダイアコプティカル行列 (Diakoptical matrix) Γ が使用される。⁽⁴⁾⁽⁵⁾ これらの行列は回路網を電気的特性に応じて分割し, それぞれの部分回路網に対してカットセット解析法ならびにタイセット解析法を適用し, 両者の特性を結合して総合特性を求めるのに便利のように基本カットセット行列 Q_f ならびに対応する基本タイセット行列 B_f のそれぞれの部分行列を組み合わせたものである。

基本カットセット行列 Q_f は理論的にはノード・ブランチ接続行列 (Node-branch incidence matrix) A に非特異行列を作用させることにより容易に導くことができる。⁽³⁾⁽⁶⁾⁽⁷⁾ また, 基本タイセット行列 B_f は次の関係式により Q_f と従属な関係にある。⁽¹⁾⁽³⁾⁽⁴⁾⁽⁶⁾⁽⁷⁾

$$Q_f B_f^t = 0 \quad \dots\dots\dots (1)$$

ただし, t : 行列の転置

したがって, 混合解析法における行列 A ならびに Γ は基本カットセット行列 Q_f が既知であれば容易に導くことができる。

回路網グラフの接続関係を記述する種々の行列 A , Q_f , B_f , A , Γ などは理論展開が便利であるために行列表示されることが多い。しかし, 計算機による小規模な回路網モデルの解析には演算操作処理ならびに計算機プログラムが簡素化しやすいために零要素も含む行列表示法を採用してもさしつかえないが, 回路網規模の拡張は行列要素の数, ならびに演算処理時間を急激に増加させるため計算機のメモリー容量ならびに計算経費の両面からの困難を招きやすくなる。

回路網グラフの接続関係を記述する諸行列に要求される条件としては, データの記憶領域がきわめて少なく, 計算処理時間が比較的短いうえに実際に使用する時には行列表示法による諸行列の場合とほとんど同程度の簡便さで利用できなければならない。

3. ノード・ブランチ接続行列のリスト表示法

回路網グラフにおけるノードとブランチの間の接続関係を記述するのに最も簡単でしかも便利な形式はノードを行の番号に, そしてブランチを列の番号にそれぞれ対応させて各ブランチの接続を表わすノード・ブ

ランチ接続行列 A_a であろう。回路網のノード数を n , そしてブランチ数を b とすれば接続行列 A_a は $(n \times b)$ 元の方形行列である。行列 A_a の各要素 a_{ij} は次のように定義される。

$a_{ij}=1$ ブランチ j がノード i に接続し, i からでる向きするとき

$a_{ij}=-1$ ブランチ j がノード i に接続し, i にはいる向きするとき

$a_{ij}=0$ ブランチ j がノード i に接続しないとき
回路網の接続は各ブランチが接続している両端のノードと, その向きが記述できればじゅうぶんである。したがって, 第1図のように j 番目のブランチがノード m_1 からでて m_2 にはいる向きに接続しているとき次に示す形式でこのブランチの接続関係を表現することができる。

$$j(m_1^{(j)}, m_2^{(j)}) \quad j=1, 2, \dots, b \quad \dots\dots\dots (2)$$

上述の表現法により回路網の全てのブランチをリスト表示したとき, 接続行列 A_a の各要素 a_{ij} の値は次式により得られる。

$$a_{ij} = \begin{cases} -(-1)^k & i = m_k^{(j)} \quad (k=1 \text{ または } 2) \\ 0 & i \neq m_k^{(j)} \quad (k=1 \text{ または } 2) \end{cases} \quad \dots\dots\dots (3)$$

よって, j 番目のブランチの接続を $k=1$ または 2 に対応させて二次元配列 LIST(K, J) のようにリスト表示すれば, 接続行列 A_a の各要素 a_{ij} の値は上式をコード化した第2図の関数形サブプログラムに

第1図 ブランチの接続

Fig. 1. Connection of branches.

```

FUNCTION      NODEA (LIST,COLUMN,LL,I,J)
INTEGER      LIST(2,LL),COLUMN(LL),PARITY
C
ENTRY        NODES (I,J)
N=0
L=COLUMN(I)
DO 100 K=1,J
IF (LIST(K,L).EQ.0) GO TO 120
CONTINUE
GO TO 130
100 CONTINUE
120 N=PARITY(I)
130 NODEA=N
RETURN
END

```

第2図 ノード・ブランチ接続行列要素発生プログラム

Fig. 2. Node branch incidence matrix generator.

より得られる。なお、一次元配列 COLUMN は、ブランチ番号の交換に便利のため導入された順序ベクトル (Ordering vector) であり、そして、関数 PARITY は $(-1)^k$ を意味する。これにより、接続行列 A_e の必要とする $(n \times b)$ 個のメモリ数を $(2 \times b)$ 個に減少させ、しかも、実際に接続行列要素を使用したいときには、Entry name NODES (I, J) によりあたかも行列表示されているように利用することができる。

4. カットセット行列のリスト表示法

回路網グラフの全てのノードを含み、ループを作らないようなブランチの集合が木である。ノード・ブランチ接続行列 A_e から回路網の電圧基準ノードに対応する行を除いた接続行列を A と表わす。行列 A の最初の $(n-1)$ 列が木となるようにブランチ番号を交換する。そして木の部分行列を A_1 ならびに補木の部分行列を A_2 とする。そのとき、接続行列 A は次式のように分割される。⁽¹⁾⁽³⁾⁽⁴⁾⁽⁶⁾⁽⁷⁾

$$A = (A_1 | A_2) \dots\dots\dots (4)$$

基本カットセット行列 Q_f は次式に示すように接続行列 A に木部分行列 A_1 の逆行行列を左側から乗ずることにより得られる。⁽¹⁾⁽³⁾⁽⁶⁾

$$Q_f = A_1^{-1} A = (Q_1 | Q_2) \dots\dots\dots (5)$$

上式のようにして基本カットセット行列が求められると、これに対応して同一のブランチ順序の基本タイセット行列 B_f は (1) 式の関係から容易に導かれる。すなわち、基本カットセット行列 Q_f の木部分行列 Q_1 ならびに基本タイセット行列 B_f の補木部分行列 B_2 はそれぞれ $(n-1)$ および $(b-n+1)$ 元の単位行列であるので、基本タイセット行列の木部分行列 B_1 は次式により得られる。

$$B_1 = -Q_2' \dots\dots\dots (6)$$

上述した (5) 式および (6) 式により、ノード・ブランチ接続行列 A の左側から回路網解析法に便利なように選ばれた木部分行列 A_1 の逆行行列を乗ずれば基本カットセット行列 Q_f が得られ、そしてこれに対応する基本タイセット行列 B_f は容易に導けることが示されている。

したがって、基本カットセット行列 Q_f は行列表示されたノード・ブランチ接続行列 A に行ならびに列の交換を施しながらガウスジョルダン (Gauss Jordan) の消去過程を適用して導くこともできる。しかし、このような行列表示による方法はメモリーの節約の観点から非常に不利な方法である。これに対し回路網グラフの木部分集合要素を道とした道行列 P の転置行列はノード・ブランチ接続行列 A の木部分逆行行列 A_1^{-1}

に等しいから、⁽³⁾ 木グラフのトポロジ的性質を利用してきわめて少数の計算機メモリーで大規模回路網の基本カットセット行列 Q_f の各要素を求めることができる。以下に、リスト表示法による木グラフの選択法ならびに道行列リストの作成法について述べる。

〈4・1〉 木の選択法 一般に与えられた回路網グラフから多数の異なる木を選ぶことができる。したがって、それらの中から混合解析法に最も便利なものが選ばればよい。

回路網の連結グラフにおけるある一つの可能な木は適当なブランチ順序で並べられたノード・ブランチ接続行列 A から $(n-1)$ 元の特異正方正部分行列を選択することができれば、その列に対応するブランチを要素とする集合である。したがって、電子計算機により全く機械的に木の部分集合を求めることだけを目的とするならば、ノード・ブランチ接続行列に列ならびに行の交換を施しながらガウスの消去過程を適用して $(n-1)$ 元の主対角要素に零を含まず、しかも左下側三角部分行列要素の全てを零とする非特異正方正部分行列を得ることができればじゅうぶんである。

全てのノードを含むノード・ブランチ接続行列 A_e の各列にはそれぞれ二つの非零要素が含まれていることはすでに述べた。この接続行列に直接ガウスの消去過程を適用すると行列の非零要素の数が増減するが、その転置行列 A_e' に同じ消去過程を適用した場合には、行列の各行に含まれる非零要素の数は決して増加することがない。

したがって、この性質を利用して回路網グラフのノード・ブランチ接続行列 A_e のリスト表示に下記の操作を施して木集合の要素を求めることができる。一般に与えられた回路網をいくつかの部分回路網に分け、それぞれ他を開放除去または短絡除去して、ノードまたはループが不変でしかも他との相互関係をとり除き解析を単純化することができる。混合解析法ではブランチの電気的特性に応じてノード形 (アドミタンス部分回路網) ならびにループ形 (インピーダンス部分回路網) の二種類の部分回路網に分解する。⁽³⁾⁽⁴⁾ そして、ノード形部分回路網の解析の際にはループ形部分回路網を開放除去する。これに対しループ形部分回路網の解析の際にはノード形部分回路網を短絡除去する。ループ形部分回路網を開放除去したときにはこれに属するブランチを全て除去したノード・ブランチ接続行列を考慮すればよい。

また、ノード形部分回路網を短絡除去したときには、ループ形部分回路網に相互接続するノードを全て短絡するか、もしくはノード形部分回路網の木要素で

その全てのノードを橋絡してもループは不変に保たれる。したがって、ノード形部分回路網に属するブランチほど優先したブランチ番号が付けられていれば、ノード・ブランチ接続行列にガウスの消去過程を適用するだけでループ形の開放除去ならびにノード形の短絡除去はそれぞれ自動的に行なわれる。しかるに、大規模な回路網では最初から上述した優先順位に従うブランチ番号を付けるのは不便であるので、順序ベクトルを導入し計算機より自動的に行なうのが便利である。(付録参照)

木の選択アルゴリズムを接続リストに適用するとそれは変形されるので、もとの接続リストを保存するために作業領域に複写しなければならない。その複写に際して、木選択アルゴリズムでの演算処理速度の向上のために順序ベクトルを参照して配列変更を施すのが便利である。以上の準備が終わったならば、木選択アルゴリズムを開始する。最初から $(b-1)$ 番目のブランチまでピボット選択のためにポインターを移動させる。

いま、ポインターが I であるとしよう。このとき、転置されたノード・ブランチ接続行列 A_a^t の第 I 行の各要素は次式に示すように K および $\bar{K}$ 列にそれぞれ $+1$ および -1 が、そして他は零である。

$$A_a^t(I, \cdot) = I \begin{array}{|c|c|c|c|} \hline & K & & \bar{K} \\ \hline & & 1 & -1 \\ \hline \end{array} \quad \dots\dots\dots (7a)$$

ただし、

$$K = \text{LIST}(1, I) \quad \dots\dots\dots (7b)$$

$$\bar{K} = \text{LIST}(2, I) \quad \dots\dots\dots (7c)$$

そして、第 J 行 (ただし、 $J > I$) の各要素は、上述した場合と同様に t および h 列にそれぞれ $+1$ および -1 、そして他は零である。

$$A_a^t(J, \cdot) = J \begin{array}{|c|c|c|c|} \hline & t & & h \\ \hline & & 1 & -1 \\ \hline \end{array} \quad \dots\dots\dots (8a)$$

ただし、

$$t = \text{LIST}(1, J) \quad \dots\dots\dots (8b)$$

$$h = \text{LIST}(2, J) \quad \dots\dots\dots (8c)$$

もし、 $t = K$ であれば、第 J 行の全ての要素の符号を変えたものに第 I 行を加えて第 t 列の要素を消去する。この操作は次の二つの連続した命令式によってリスト表示形式のまま実行できる。

$$\text{LIST}(1, J) = \text{LIST}(2, J) \quad \dots\dots\dots (9a)$$

$$\text{LIST}(2, J) = \text{LIST}(1, J) \quad \dots\dots\dots (9b)$$

また、 $h = K$ であれば第 I 行を第 J 行に加えるだけで $A^t(J, h)$ を消去することができる。したがって次の代

入命令式により等価な操作を実現できる。

$$\text{LIST}(2, J) = \text{LIST}(2, I) \quad \dots\dots\dots (10)$$

次に $K = t$ ならびに $\bar{K} = h$ 、または $K = h$ ならびに $\bar{K} = t$ のとき、第 J 行の要素は全て消去される。したがって、次の二つの命令によってリスト表現を変える。

$$\text{LIST}(1, J) = 0 \quad \dots\dots\dots (11a)$$

$$\text{LIST}(2, J) = 0 \quad \dots\dots\dots (11b)$$

ただし、 $\text{LIST}(1, J)$ が零の場合には $\text{LIST}(2, J)$ も零であると約束しておけば第 2 番目の命令は不用である。

上述した操作をくり返して消去過程が終了したら、接続リストの内容を調べて零でないものが木の要素、そして零であるものが補木の要素に対応している。

順序ベクトルは上述した処理の完了後に最初の $(n-1)$ 要素に木のブランチ番号を、そして他の $(b-n+1)$ 要素に補木のブランチ番号を入れ換えておくことと便利である。

上述した木選択アルゴリズムのうち、消去過程のフロー図の例を第 3 図に示す。また、この手法により木を求める過程の例を示す回路網グラフの例に

第 3 図 木選択アルゴリズム(消去過程)フロー図

Fig. 3. Flow diagram of tree selection algorithm.

第 4 図 回路網グラフの例

Fig. 4. An example of network graph.

第 1 表 木選択アルゴリズムの消去

Table 1. Elimination of tree-selection algorithm.

ブランチ番号	消 去 段 階						備考
	0	1	2	3	4	5	
1	5	1	5	1	5	1	t_1
2	1	3	1	3	1	3	t_2
3	3	5	3	1	0	3	c_1
4	1	2	1	2	2	3	t_3
5	2	3	2	3	0	3	c_2
6	2	4	2	4	4	3	t_4
7	4	5	4	1	4	3	c_3
8	4	6	4	6	4	6	t_5
9	1	6	1	6	6	3	c_4
10	6	5	6	1	6	3	c_5

第 2 表 順序ベクトルの例

Table 2. Example of ordering vector.

1	2	4	6	8	3	5	7	9	10
木					補木				

より示す。第1表に回路網グラフのノードとブランチの接続リストならびに第1～5段階の消去過程によるリストの変化を示した。最終消去ステップが終わると第2表に示すようにリストの第1列が非零のものを木要素として順序ベクトルの前半に、また零のものを補木要素として後半に各ブランチ番号で記入する。

〈4・2〉 道行列のリスト表示法 木部分グラフにおける道行列は、回路網の各ノードから基準ノードに至る経路に含まれる木の要素とその向きが記述できればよい。木部分グラフは第5図に示す例のようにループを作らない $(n-1)$ 個の木要素が連鎖的に、しかも基準ノードを中心として放射状に接続されている。

図にも示したように $(n-1)$ 個の木要素はそれぞれ次に示すようにノード t からで h にはいる向きに接続されているものとする。

$$i(t, h) \quad i=1, 2, \dots, n-1 \quad (12)$$

①, ②, ④, ⑥, ⑧: ブランチ番号

第 5 図 木グラフの例

Fig. 5. Example of tree graph.

あるノードから基準ノードに至る経路に含まれる要素の数をそのノードのレベルとして、ノード t ならびに h のレベルをそれぞれ $L(t)$ ならびに $L(h)$ と示す。各ノードから基準ノードに向う方向を正とすれば、 i 番目の要素のうちレベルの高いノードから低いノードへの長さが1なる道(枝) r_i は次式のように書き換えることができる。

$$r_i = j(q_i, s_i) \quad i=1, 2, \dots, n-1 \quad (13)$$

ただし、上式においてノード t のレベルが h のより高いとき、すなわち $L(t) > L(h)$ のとき

$$j=t \quad (14a)$$

$$q_i h_i = \quad (14b)$$

$$s_i = i \quad (14c)$$

また、反対にノード b のレベルが t のそれより高いとき、すなわち $L(t) < L(b)$

$$j=h \quad (15a)$$

$$q_i = t \quad (15b)$$

$$s_i = -i \quad (15c)$$

そして、 q_i が基準ノードに対応していて、零でない場合には $q_i=0$ とすると便利である。

たとえば、第5図において、(14)式については $t=1^\circ, h=2^\circ, i=3$ に対し $r_3=1(2, 3)$ 、また(15)式については $t=1^\circ, h=3^\circ, i=2$ に対し $r_2=3(1, -2)$ などのようになる。

全ての木集合要素について上述したデータ変換を施すと、それぞれの要素から選ばれた高レベルのノード番号 j は1から $n-1$ の間の重複のない数値になる。したがって、それを指数として $(n-1) \times 2$ 元の配列に

各道データを記憶させることができる。

〈4.3〉 道データ処理プログラム 上述した方法により道行列のためにリスト表示された道データ処理には木部分グラフにおける各ノードのレベル決定ならびに道データのリストアップを行なう必要がある。この両者を別々に実行してもさしつかえないが同時に実行するほうが能率が良い。以下にその処理操作を詳述する。

まず、作業領域に n 個のノードプールと $(n-1)$ 個のフラッグを用意する。そして、フラッグはあらかじめ全ての要素を零にする。ノードプールにはレベルが「0」である基準ノード n^0 または 0^0 を記録する。

木集合要素の両端のノード番号とノードプールに記録されているものと比較する。もし、一致するものがあれば、そのノードの反対側のノードはさらにレベルが1だけ高いものである。もし、 I 番目の木集合要素の一端のノード K_P がノードプールに記録されているものとする。すなわち

$$K_P = \text{LIST}(K, J) \quad (K=1 \text{ または } 2) \dots (16)$$

ただし、上式において J は次式に示すように順序ベクトル COLUMN の I 番目の数値とする。

$$J = \text{COLUMN}(I) \dots (17)$$

このとき求められた高レベルノード K_Q は次式により得られる。

$$K_Q = \text{LIST}(M, J) \dots (18)$$

ただし、

$$M = K - (-1)^K \dots (19)$$

したがって、次式に示すように道データをリストアップすればよい。

$$K_Q \{K_P, (-1)^K I\} \dots (20)$$

そして、同時にノードを K_Q ノードプールに記録し、 I 番目のフラッグに非零要素を入れて上述した道データ処理プログラムのループから除外させる。

以上の過程を全ての木集合要素が処理されるまでくり返せば、全ての道をリストに表示することができる。第6図に上述したプロセスのフロー図を示す。

第4図に示したグラフ例について、上述した方法により第1表ならびに第2表から第3表に示したリスト表示が得られる。

〈4.4〉 道行列要素の発生法 前述した道データのリスト表示から道行列要素 $P(i, j)$ の値は下記に示す操作により容易に求めることができる。

(1) まず、 $k=i$ と置き換える。

(2) k 番目の長さが1なる道(枝)は次式のようにリストされている。

第6図 道データ処理プログラムのフロー図

Fig. 6. Flow diagram of path data processor.

第3表 道のリスト表示例

Table 3. Example of path listing.

1	2	3	4	5
2	4	1	0	1
3	4	-2	5	1

$$x_k = k(q_k, s_k) \dots (21)$$

第2要素の s_k の絶対値と道行列の列番号 j を比較する。

(a) $|s_k| = j$ のとき、道行列要素 $P(i, j)$ は次式に示すように s_k の符号を付した1である。

$$P(i, j) = \text{sign}(s_k) \dots (22)$$

(b) $|s_k| \neq j$ のときには、 q_k が零であるか否かにより、次のいずれかを選ぶ。

(i) $q_k = 0$ のとき 与えられたノードから基準ノードまで調べた結果、木集合の j 番目の要素は径路に含まれていない。したがって、道行列要素は零であ

```

INTEGER FUNCTION PTPACK(FLAG,NF,II,JJ)
INTEGER FLAG,NF,II,JJ
C
ENTRY ENTRY
PATHT(II,JJ)
PATHT(JJ,II)
IP=II+1
IK=FLAG(IP)
IF (IP.EQ.1) GO TO 120
IF (IP.EQ.2) GO TO 110
IF (IP.EQ.3) GO TO 110
IF (IP.EQ.4) GO TO 110
IF (IP.EQ.5) GO TO 110
PTPACK=0
GO TO 130
110 PTPACK=0
GO TO 130
120 PTPACK=ISIGN(1,IK)
RETURN
130 END

```

第 7 図 道行列要素発生プログラム

Fig. 7. Path matrix generator.

	1	2	3	4	5
1°			1	1	1
2°				1	1
3°		-1	1	1	1
4°					1
5°	1		1	1	1
	①	②	④	⑥	⑧

1°~5°: ノード番号
①, ②, ④, ⑥, ⑧: ブランチ番号

第 8 図 道行列の例

Fig. 8. Example of path matrix.

る。すなわち

$$P(i, j) = 0 \quad (23)$$

(ii) $q_k \neq 0$ のとき 与えられたノードから基準ノードに未到達である。したがって、 $k=q_k$ として低ノードに移行して上述のプロセスをくり返す。

第 7 図に道行列要素を発生させる関数サブプログラムの例を示す。第 4 図のグラフ例において、第 3 表のデータに上述の方法を適用すると第 8 図に示す道行列が得られる。

〈4・5〉 カットセット行列要素の発生法 前節までに述べた手法ならびにプログラムにより、道行列の全ての要素をきわめて少いメモリー数で知ることができる。また、道行列はノードブランチ接続行列の木部分行列の転置逆行列に等しいことも述べた。したがって (5) 式により基本カットセット行列 Q_f を求めることができる。すなわち、行列 Q_f の要素 q_{ij} は次式により計算される。

$$q_{ij} = \sum_{k=1}^{n-1} P_{ik}^t a_{kj} \quad (24)$$

しかし、上式においてノードブランチ接続行列 A の第 j 行にはたかだか二つの非零要素しか含まれていない。また、接続のリスト表示を参照すれば非零要素の行番号ならびに要素 a_{ij} の正負は次式により容易に知ることができる。

$$k = \text{LIST}(l, m) \quad (l=1 \text{ または } 2) \quad (25a)$$

$$m = \text{COLUMN}(j) \quad (25b)$$

$$a_{kj} = -(-1)^l \quad (25c)$$

ただし、LIST ならびに COLUMN は接続のリストならびに順序ベクトルである。

さらに、 j が $(n-1)$ 以下の場合には、基本カットセット行列の木部分に対応しているため、単位行列になる。したがって、上の (25) 式に示した処理の代りに次式を採用してもよい。

$$q_{ij} = \begin{cases} 1 & (i=j \text{ のとき}) \\ 0 & (i \neq j \text{ のとき}) \end{cases} \quad (26)$$

ただし、 $j \leq n-1$

上述した方法により、回路網規模がかなり大きくなってきわめて少いメモリー数と短い計算機時間により基本カットセット行列の全ての要素を知ることができる。さらに、基本タイセット行列をはじめ基本カットセット行列 B_f から誘導される全ての行列は変換プログラムを利用すれば容易にそれらの全要素を求めることができる。第 9 図に前述した道行列要素発生法とリスト表示された接続をもとに基本カットセット行列要素を発生させる関数サブプログラムの例を示す。第 4 図のグラフ例について、上述したプログラムを適用すれば第 10 図に示すカットセット行列が得られる。

5. トポロジ処理系の性能

前節までに回路網グラフの木部分集合の選択法、道行列のデータ処理法、道行列ならびにカットセット行列の要素発生法、などについて述べた。実際に大規模な回路網の過渡特性を計算する場合には、回路網グラ

```

INTEGER FUNCTION PNCUTS(FLAG,NF,LIST,COLUMN,II,RR,II,JJ)
INTEGER FLAG,NF,LIST,COLUMN,II,RR,II,JJ
CALL PTPACK(FLAG,NF,II,JJ)
MM=MM+1
ENTRY ENTRY
PNCUT(II,JJ)
IF (II.EQ.1) GO TO 100
IF (II.EQ.2) GO TO 100
IF (II.EQ.3) GO TO 100
IF (II.EQ.4) GO TO 100
IF (II.EQ.5) GO TO 100
J=COLUMN(JJ)
DO 120 K=1,7
N=LIST(K,J)
IF (N.EQ.1) GO TO 120
IF (N.EQ.2) GO TO 120
IF (N.EQ.3) GO TO 120
IF (N.EQ.4) GO TO 120
IF (N.EQ.5) GO TO 120
PNCUTS=0
GO TO 130
120 PNCUTS=ISIGN(1,RR)
130 CONTINUE
140 PNCUTS=0
RETURN
150 END

```

[第 9 図] カットセット行列要素発生プログラム

Fig. 9. Cutset matrix generator.

	1	2	3	4	5	6	7	8	9	10
1°	1					-1		-1		-1
2°		1				-1	1			
3°			1				-1	-1	1	-1
4°				1				-1	1	-1
5°					1				1	-1
	①	②	④	⑥	⑧	③	⑤	⑦	⑨	⑩
1°~5°: ノード番号										
①~⑩: ブランチ番号										

第 10 図 カットセット行列の例

Fig. 10. Example of cutset matrix.

フのトポロジ処理に引き続いて、数学的モデルの定式化、固有値の計算、連立微分方程式の数値解析、などのためかなり多量の計算プログラムならびにデータをメモリーに収容しなければならないし、また計算時間もかなり長くなることが容易に予想される。したがってトポロジ処理系の所要メモリー数ならびに処理時間は全体の計算費用を低減させるためにもできるだけ小なることが望ましい。

本文に説明した手法によるトポロジ処理系は木選択ならびに道データ処理の準備段階と数学的モデル定式化、初期値ならびに計算結果の変数変換などのために各行列要素発生部分の二段階に分割することができる。準備段階は処理計算の終了とともにプログラムは不用となるが、第二段階のプログラムならびにデータは必要に応じていつでも使用できるように高速メモリー中に常駐させておかなければならない。行列要素発生部分プログラムの所要メモリー数は計算機の種類ならびにソフトウェアによっても異なるが、プログラム開発に使用した FACOM 230-50 CUC Fortran システムでは 714 語にすぎない。データの所要メモリー数は回路網モデルの規模に左右され、 $2(b+n-1)$ 語で

ある。準備段階の性能評価の一方法として 8 種類のモデル回路網を選び、行列表示形式のノードブランチ接続行列に消去法を適用した、カットセット行列の誘導との所要処理時間比較を行ない、その結果を第 4 表に示した。この結果、本文に述べた手法は処理時間ならびにデータの所要メモリー数は回路網規模の拡大に伴い、ますます有利になる傾向が明らかにされた。

6. あとがき

電力系統または電力用機器のように大規模な回路網の特に過渡解析計算モデルの接続に関する種々の行列の行列表示方式に対し、道行列のリスト表示によるデータ記憶方式と対応する行列要素発生関数サブプログラムから構成されるトポロジ処理システムは、高速メモリー数の節約ばかりでなく計算処理時間の大幅な短縮も伴ってきわめて有利な方式であることを示した。すなわち、小規模なモデルさえもメモリー数が 50% そして計算時間が 14% に低減させることができるが規模の拡大に伴ってこれらはそれぞれ 2.5% ならびに 4% 程度にまでしか必要とされなくなる。また、必要とされるメモリー数 $2(l+b-1)$ 個のうち、 $2l$ 個のメモリーはインプットデータとして使用されたいの場合、最後まで保存されるものであるから、データ領域として実質的に準備しなければならないのは $2(b-1)$ 個のメモリーである。しかも、モデル回路が与えられるとその大きさを事前に知ることができるのできわめて便利である。

各行列要素を実際に使用するときには、関数サブプログラム系にパラメータのプリセッティングを施すか COMMON 領域を利用することによりそれぞれ Entry name または Function name を外見的には全く行列表示方式の場合と同一に利用できる。

以上のように数多くの長所を供えた処理システム目

第 4 表 道データリスト法と行列表示形式法の性能比較

Table 4. Comparison between performances of path data list and of matrix methods.

回路網パターン		1*	2	3	4	5	6	7	8
ブランチ数 (b)		10	13	21	22	35	115	109	213
ノード数 (n)		6	10	10	21	15	47	63	132
木選択処理時間 (秒)		0.03	0.04	0.06	0.07	0.13	1.04	1.48	5.36
道データリスト法 (A)	処理時間 (秒)	0.01	0.01	0.02	0.02	0.02	0.11	0.26	0.83
	メモリー数	30	44	60	64	98	324	352	688
行列表示形式法 (B)	処理時間 (秒)	0.07	0.18	0.2	0.21	0.49	5.57	5.51	20.76
	メモリー数	54	117	189	220	490	5,290	7,303	27,903
比 較 (A/B)	処理時間比	0.14	0.056	0.1	0.095	0.041	0.02	0.047	0.040
	メモリー数比	0.56	0.37	0.32	0.29	0.2	0.061	0.043	0.025

*: 第 4 図参照

身の命令用のメモリー数は問題とするほどのこともなくしかもこれは回路網モデル規模に全く左右されない。

本文に述べたトポロジ処理システムを利用して回路網過渡解析プログラムの開発には、数学的モデルの係数行列についても、Sparse matrix programming technique を採用すればさらに大きな効果をもたらすであろう。

最後に本研究は筆者のマンチェスタ大学 (UMIST) 留学中に同大学の Wedepohl 教授, Arrillaga 博士ならびに同大学に Kharkov の Institute of Radioelectronics より留学中であった Yevedokimov 博士, などの諸氏との討論によるところが大であることに感謝いたします。また、電子技術総合研究所の小山茂夫技官にはプログラミングに際しご助力いただいたことに感謝いたします。

(昭和46年1月21日受付, 同46年8月12日再受付)

文 献

- (1) S. Seshm & M.B. Reed: Linear Graphs and Electrical Networks, (1961) Addison-Wesley
- (2) G.W. Stagg & A.H. ElAbiad: Computer Methods in Power System Analysis, (1968) McGraw Hill

- (3) N. Balabanian & T. A. Bickart: Electrical Network Theory, (1969) John Wiley
- (4) 平山・難波田: 電学誌, 88, 1179~2008 (昭 43-7~11)
- (5) 渡部: 電学誌, 89, 204 (昭 44-2)
- (6) 小野寺: グラフ理論の基礎 (昭 43) 森北出版
- (7) 小野寺・大類: 電気回路網基礎理論 (昭 42) 学献社

付 録

順序ベクトル ノード形ならびにループ形の各部分回路網は、それぞれアドミタンスならびにインピーダンス要素のみを含むように分割される。ブランチの接続ならびにその回路素子定数のリストはブランチに付けられた番号順に並べられているものとする。ブランチの優先順位は回路素子の種類に応じて次のように仮定する。

アドミタンス成分	キャパシタンス	1
	コンダクタンス	2
インピーダンス成分	抵抗	3
	インダクタンス	4

なお、コンダクタンス、および抵抗ブランチはそれぞれ他の種類のものを含まないものとする。

最初に用意される順序ベクトルは、ブランチ番号を上記の優先順位の順にそれぞれのブロックごとにとまとめたものをベクトルの要素としたものである。