（1995．7．19）

s関数を要素とする行列式高速解法の開発

益冨文男

福岡工業短期大学　電子情報システム学科
〒811－02　福岡市東区和白東3　30　1

電気回路網などの連立微積分方程式を解き時間tの関数解を求める代数学計算方法ではラプラス変換後s領域の関数を要素とするマトリックスを形成し，次いでクラーメルの行列式解法を適用し，変数ごとに有理関数解を導く演算を施す。その際，従来は項展開手法に依っていて8時間を超す場合があった。今回，誤差ゼロで小行列式に展開する方法を開発し得たので報告する。消去法の良いところを計算に採り入れたほか，列の入替えおよび行と列の一斉入替えを用いた。その結果，処理時間はs関数の次元に略々正比例する様になった。

Development to solve determinants of s-function elements algebraically and speedily

Fumio Masutomi

Fukuoka Junior College of Technology
3-30-1,Wajiro Higashi,Higashi-ku,Fukuoka shi,811-02 Japan

This paper shows the speedy CAD to minorize a given determinant of its elements including s-domain functions. The Cramer's method to get the ratinal functions in s from determinants can be easily and accurately done. Techniques with applications are shown. The calculating times are in proportion to the number of given simultaneous equations practically.

１．はじめに

ｔ関数線形連立常微分方程式にラプラス変換を施してｓ関数化し，各変数の代数解を求めるためにはｓ領域の有理関数を導く計算が必要である。ｓ関数を要素(エレメント)とする正方行列式を解く演算が必要になる。筆者は１９７８年にクラーメルの方法を用い行列要素の積の項に展開し各項の和を求める単精度計算機向け方式を完成し実用した。処理時間が(次元)の階乗倍も要したので待ち時間が多大となる場合があり２０次元までの適用を限度と考えて来た。１９９３年から倍精度演算可能なＰａｓｃａｌ言語を用い小行列式化プログラムの開発に着手し今回完成し得たので報告する。即ち，第１行内の要素を調べ低次なｓ関数を左上に置く様に列の入替えを施す。消去法適用の過程で除残ｓ次式の各係数がゼロか否かを監視し，除残が有る時には主対角線や副対角線対象要素入替え後に再計算させる。誤差ゼロになるまで再計算させる新方式である。現在は４次元以上の行列式計算には小行列式化手法を適用して，３次元以下の行列式計算は項展開法を用いている。処理時間を(次元)に比例させる時間短縮に成功し得たのが特徴である。例えばインテル３８６搭載パソコンＲＡＭＤＩＳＫ上での計算では，４０次元の場合の処理時間は５×４０＝２００秒程度である。

２．行列式の次数低減法

定数を要素する行列式解の高精度化を実現する方法に就いて今回開発した要点を記す。元の行列式よりも一つ低い次元への小行列式化を行う。その際に，元の行列式の左上コーナー(第一行第一列)の内容が０であれば，第一行内に０以外の数値を有する列を探して第一列内容との列入替えを先ず行う(符号正負の反転結果を記憶させている)。左上コーナーの内容を記憶させる。第一列第二行以降を０にする様に第二行以降各要素の数値を計算し決める(ａｂ－ｃｄ／ｅの様に除算を最後に１回だけ行う演算順序で行う)。四則演算はオーバーフローやアンダーフローに基づく有効桁落ちが生じない様にするために一列の全要素に数値ＵＡ(例えば10^{20})を乗じ，行列式計算後にその数値ＵＡで除す。計算に必要な数値ＵＡに就いては何回か試行して決める。

今回，仕様を行列式次元７０，計算結果のｓ関数の次数４０と設定し，ＭＳ－ＤＯＳ内ディレクトリファイル群を設計した。た。Ｐａｓｃａｌ言語で開発したｐｒｏｃｅｄｕｒｅ文を以下に掲げる。ディレクトリファイル群とのアクセスを行なっているのが特徴である。

ｒｎ実数値の読み出し用
ｒｃ文字列のｃｓへの読み出し用
ｗｎ実数値の書き込み用
ｗｃ文字列のｃｓへの書き込み用
ｃｒキャリッジリターン改行用
ｃｌｅａｒ（ｇｎ：ｉｎｔｅｇｅｒ）；多項式ｓの次数及び各係数記憶内容のゼロクリア用
ｔｅｎｓｏ（ｇｆ，ｇｔ：ｉｎｔｅｇｅｒ）；多項式ｇｆからｇｔへの転送(コピー)用
ｈｙｏｕｊｉ（ｇｄ：ｉｎｔｅｇｅｒ）；多項式ｇｄ内容の表示用
ｗａ（ｇａ，ｇｂ，ｇｗａ：ｉｎｔｅｇｅｒ）；多項式ｇａ＋ｇｂ＝ｇｗａの計算用
ｓａ（ｇａ０，ｇｂ０，ｇｓａ：ｉｎｔｅｇｅｒ）；多項式ｇａ０－ｇｂ０＝ｇｓａの計算用
ｓｅｋｉ（ｇｃ，ｇｄ，ｇｓ：ｉｎｔｅｇｅｒ）；多項式ｇｃ＊ｇｄ＝ｇｓの計算用
ｓｈｏｕ（ｇｅ０，ｇｅ１，ｇｓｈ，ｇｊｏ：ｉｎｔｅｇｅｒ）；多項式ｇｅ０／ｇｅ１の商ｇｓｈおよび剰余の多項式ｇｊｏの計算用

ｒｈｊ（ｋｃｐ：ｉｎｔｅｇｅｒ）；正方行列ｒエリア内容の表示用
ｍａｘｍｓ与えられた正方行列次数の読み出し用
ｉｎｉ正方行列ｑ，ｒエリア内容のゼロクリア後ｑエリアに０，１，－１などの数値やｓの多項式ｆエリア名を設定する初期値化(イニシャライズ)用
ｃｑｒ（ｋｑｒ：ｉｎｔｅｇｅｒ）；正方行列ｑエリア内容をｒエリアへコピー後，ｑエリア内容ゼロクリア用
ｇｒｃ（ｋｇｒ：ｉｎｔｅｇｅｒ）；主対角線対象(左上から右下)各行各列要素一斉入替え用
ｇｒｃ０（ｋｇｒ：ｉｎｔｅｇｅｒ）；副対角線対象(左下から右上)各行各列要素一斉入替え用
ｒｃｈ（ｋｃｐ：ｉｎｔｅｇｅｒ）；第一行内要素の最小ｓ次数調査，その列と第一列との交換処理用
ｓｓｄ（ｋｌｑ：ｉｎｔｅｇｅｒ）；正方行列右下要素の利用検討処理用
ｓｂ（ｍ，ｈ，ｋａ１，ｋａ２，ｋａ３：ｉｎｔｅｇｅｒ）；除残無し小行列式要素形成管理用
ｓｃ（ｍ，ｈ：ｉｎｔｅｇｅｒ）；小行列式要素形成計算処理用
ｓｄ（ｋｓｄ：ｉｎｔｅｇｅｒ）；小行列式計算起動用

四則演算で生じた新たなｓの多項式には新たなｆエリア名を付与して正方行列ｑ，ｒエリアの記憶内容を更新させている。小行列式化計算過程では除残無し小行列式要素形成のために自動再計算を行わせている。除残としてのｓ関数が残る場合には，計算を打ち切る方式にしている。３次行列式や２次行列式の場合は要素の項展開法を用いている。小行列式化計算プログラムの使用は４次元以上の行列式を対象として行い除残無しの結果を得ている。

キルヒホッフの第一法則(電流則)の式を表す行要素の内容は数値０，１，－１から成る。
キルヒホッフの第二法則(電圧則)の式を表す行でも数値０やｓ^{0}の係数値の要素が有る場合が多い。これらの原因は節(ノード)及び枝路(ブランチ)を前提として定式化した方式とも関係すると考える。

小行列式化計算の過程で今後必要になれば，メイン文の項展開計算プログラム箇所に４次元用などを追加して除残無し高精度高速計算の方針を維持してゆく所存である。図１に小行列式化計算の実例を掲げる。行列要素が示す整数値は¥ｍｋディレクトリｆエリア内ファイルの管理番号である。欄外右側にｆエリア内ファイルの管理番号とｆエリア内ｓの多項式次数とを掲げる。此処では，同一管理番号ならばｓの多項式内容は不変である。誤差ゼロで演算した経過を示している。プログラムメイン部の例を図２に示す。

３．おわりに

1)計算機に¥ｍｋディレクトリｆエリア内ファイルの管理番号を設定させ，ｓ関数を要素に含む行列式計算用の小行列式化計算手法を完成し得た。今回開発に成功したのは，クラーメルの方法で行列式を項展開する方式に於いて消去法を改善した手法を採用した点に在る。次元数の高い場合に小行列式化計算を誤差無しに行わせる。
2)その結果，インテル３８６ＣＰＵ２０ＭＨｚ搭載のＲＡＭＤＩＳＫ演算可能なパソコンでの計算で連立方程式数９個を解いて６次のｓの代数式を除残誤差無しに倍精度高速演算し得た。
3)所要時間は５秒ｘ９＝４５秒と少ない。
4)全ての演算を項展開法に頼っていた従来法では連立方程式数ｎの階乗倍の秒数を要していた。８時間を超す場合もあった。改善した本法では連立方程式数ｎに略々比例する様な秒数である。ラプラス変換法普及に貢献するばかりでなく，電気電子機械振動制御系を含むシステム建設設計に貢献するであろう。

	1	2	3	4	5	6	7	8	9	
1	3			4						3:1ji 4:2ji
2	5				1					5:1ji 1:0ji
3	2	1				6				6:1ji 2:0ji
4		2	1				7			7:1ji
5			8					1		8:1ji
6				1	2	2				
7						1	2		2	
8							1	2		
9		2							9	9:1ji

	1	2	3	4	5	6	7	8	
1			12	1					12:2
2	1		13		6				13:1 6:1
3	2	1				7			7:1
4		8					1		8:1
5			1	2	2				
6					1	2		2	
7						1	2		
8	2							9	9:1

	1	2	3	4	5	6	7	8	
1	1		12						12:2
2			13	1	6				13:1 6:1
3		1		2		7			7:1
4		8					1		8:1
5	2		1		2				
6					1	2		2	
7						1	2		
8				2				9	9:1

	1	2	3	4	5	6	7	
1		13	1	6				13:1 6:1
2	1		2		7			7:1
3	8					1		8:1
4		14		2				14:2
5				1	2		2	
6					1	2		
7			2				9	9:1

	1	2	3	4	5	6	7	
1	1	13		6				13:1 6:1
2	2		1		7			7:1
3			8			1		8:1
4		14		2				14:2
5				1	2		2	
6					1	2		
7	2						9	9:1

	1	2	3	4	5	6	
1	15	1	16	7			15:1 16:1 7:1
2		8			1		8:1
3	14		2				14:2
4			1	2		2	
5				1	2		
6	17		18			9	17:1 18:1 9:1

	1	2	3	4	5	6	
1	1	15	16	7			15:1 16:1 7:1
2	8				1		8:1
3		14	2				14:2
4			1	2		2	
5				1	2		
6		17	18			9	17:1 18:1 9:1

	1	2	3	4	5	
1	19	20	21	1		19:2 20:2 21:2
2	14	2				14:2
3		1	2		2	
4			1	2		
5	17	18			9	17:1 18:1 9:1

	1	2	3	4	5	
1	1	20	21	19		19:2 20:2 21:2
2		2		14		14:2
3		1	2		2	
4	2		1			
5		18		17	9	17:1 18:1 9:1

	1	2	3	4	
1	2		14		14:2
2	1	2		2	
3	22	23	24		22:2 23:2 24:2
4	18		17	9	17:1 18:1 9:1

	1	2	3	
1	2	25	2	25:2
2	23	26		23:2 26:4
3		27	9	27:3 9:1

図1　サイリスタ応用回路の共通分母s関数行列式計算経過の例

```
program fdetd(input,output);
label 8,10,12,90,99;
const
  csm = 3;
  mji = 40;      { ﾀｺｳｼｷ ji ｽｳ ﾉ ｻｲﾀﾞｲﾁ ｦ ｾｯﾃｲ }
  mko = 70;
type
  real = longreal;
  t = array[1..mji,1..mji] of 0..mko;
  t1 = array[-6..mko] of 0..mji;
  t2 = array[-6..mko, 0..mji] of longreal;
var
  aa,bb,cc,ka,kai,kaj,kak,kal,kam,max,ms,
  bn,km,kn,ko,kp,kq : integer; {ka ﾊ 0..mko ｶﾝﾘ}
  q,r : t;
  jisu : t1;     { keisu ﾉ gno ﾉ ji ｽｳ }
  keisu : t2;
  x : longreal;
  inf,ouf : string[12];
  cs : string[3];

begin
 maxms;if ms<=2 then kp:=2 else kp:=3;
 writeln('ｹｲｻﾝｺﾞ ｷｮｳﾂｳﾌﾞﾝﾎﾞbgsｦ-1 ﾍ ｶｸﾉｳｼ  ﾌﾞﾝｼ ﾊ b00..ｴﾘｱﾆ ｶｸﾉｳ ｼﾏｽ');
 inf[0]:='¥';inf[1]:='m';inf[2]:='k';inf[3]:='¥';inf[4]:='f';
 inf[5]:='k';inf[6]:='o';inf[7]:='.';inf[8]:='s';inf[9]:='u';rn;
 ko:=round(x);
 ini;
 km := ms; {ｲｽﾞﾚｶ ｷﾞｮｳﾚﾂ ﾅｲ ﾆ ﾃﾞｰﾀ ｱﾙ ﾊﾞｱｲ}
 aa := ka; ka:=aa+1; clear(ka); tenso(1,aa); {ｹｲｽｳ aa=+1 ｦ ﾏｽﾞ ｱﾀｴﾙ};
 bb := ka; ka:=bb+1; clear(ka); tenso(1,bb); {ｹｲｽｳ bb=+1 ｦ ﾏｽﾞ ｱﾀｴﾙ};
 cc := ka; ka:=cc+1; clear(ka);
 kai := 0;
 cqr(km); kq:=0;
8:
 ssd(km);
 while km > kp do
   begin
     rch(km);
     sd(km);
     cqr(km);
     rhj(km);
     kq:=0; km := km - 1;
   end;
 goto 12;
10:
 if kq=1 then grc0(km) else grc(km);
 kq:=kq+1;
 if km<=3 then begin kp:=km; goto 12; end else if kq>2 then
 begin cr;write('ｱﾀｴﾀ uaﾁ ﾍﾝｺｳ ﾖｳ ﾃﾞｽ  ftd ｶﾗ ｻｲﾄﾞ ｹｲｻﾝ'); goto 90 end
 else goto 8;

90:
end.
```

図２　開発したプログラムメイン部の例