

1. まえがき

差分法は古くから用いられてきた方法¹⁾であるが、高度な数学的知識を必要とせず、しかも微分方程式の近似解法として汎用性に富んでいる。最近とはくに電子計算機の急速な発達とあいまって多用され、構造解析法としては有限要素法と並ぶきわめて有効な近似法となっている。しかし、有限要素法をはじめ、伝達マトリックス法、剛性法、有限帯板法などいわゆるマトリックス構造解析法は、一つの小さな剛性マトリックスを考えるだけで、あとはプログラミングの仕事に移る、電子計算機の使用に適した方法である。差分法も電子計算機の使用を前提にすればこれらの解法に十分匹敵する解法であるが、微分方程式を差分表示することとマトリックスを組立てることになりの手数を要し、それだけプログラム・デバッグのために時間と労力を消費することになる。ここに報告する FORTRAN プログラムは差分式を任意の精度と形で自動的に組立てることができるもので、1971年に作製したものに改良を加えたものである。これの使用によって、微分方程式を無次元化するだけでよく、マトリックスの組立ても容易になり、プログラムを短期間のうちに完成することができる。

電子計算機プログラムによって差分式の組立てを行う方法についてはいくつかの研究報告がある。Silvester³⁾は偏微分係数の差分表示をマトリックス演算によって計算するプログラムを報告しているが、Taylor 級数法に基づいており、応用性に関してあまり簡単な手法ではない。Jensen⁴⁾、Perrone-Kao⁵⁾らは Taylor 級数法によって任意の不規則分割に対する偏微分係数の差分表示を研究しているが、2階までに限られており、分点の配置によっては特異になる不都合がある。Petravic⁶⁾は偏微分係数に対する一般的な差分係数計算法と Symbolic Algol I で書いている。Kumar⁷⁾は微係数の差分表示を作成する関数制プログラムはあるものとして、微分方程式に対する差分式を組立てるプログラムについて論じている。Csender⁸⁾は Lagrange 補間公式を用いて任意階の常微分係数の差分係数を求め、直接によって偏微分係数の差分係数を求める方法を示し、FORTRAN プログラム・リストを添えているが、マトリックス演算を必要とし、前進・後退・中央・偏に差分形を自由に作成するのに不都合である。

2. 多項式近似による差分表示の作成

任意精度の差分式を作成するに先立って、精度改良法を検討しておく必要がある。第1の方法は不等間隔法⁹⁾であるが、適切な分割法が対象とする問題とくに境界条件によって異なり、一般的に把握できない。分割法を誤るとかえって精度が低下することもあり、プログラム作製上も不利な点が多い。

第2の方法は打ち切り誤差の小さい高精度差分表示¹³⁾¹⁴⁾を用いる方法(ここでは等間隔分割とする)で、主として次の2つに分けられる。第1はより多くの分点における関数値を用いる多分点法で、他は関数値のほかに微分係数を用いる Hermitian 法¹⁴⁾とその拡張法^{15)~17)}である。不利な境界条件のもとでも Hermitian 法が有効であるという報告¹⁸⁾もあり、また2つの差分表示がほぼ一致した結果を与えるという報告¹⁹⁾もある。Hermitian 法については多項式近似による差分表示作製法とそのプログラムが示され、²⁰⁾はりの横倒し解析²¹⁾に用いられているが、常微分の場合に限られている。ここでは、簡潔で、プログラム作製上有利な多分点法を用いる。

差分表示作成法として Taylor 級数による方法と多項式近似による方法がある²²⁾が、前者はマトリックス演算を必要とするので、差分表示を陽な形で誘導できる多項式近似法を用いる。以下に1変数関数 $f(x)$ の微分係数の差分表示を任意

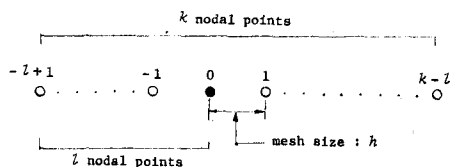


図 ~ 1

な形(前進・後退・中央・偏心)で作成する。なお、Psfanz²³⁾は任意階微分係数の差分表示を以下と同様の手法で求めているが、中央差分に限られており、境界改良法²⁴⁾を用いるためにも任意形の差分表示が必要になる。

x軸を等間隔 h に分割し、分点を設ける。 $x=x_i$ なる分点 i において差分表示を求める場合、局所座標 $\xi=(x-x_i)/h$ を用いると都合がよい。各分点に対して分点 i と 0 とする局所的番号付けを施す。図-1のように k 個の分点を用いて偏心差分表示を作る場合、 k 個の分点を通る $(k-1)$ 次多項式はLagrange補間公式より

$$f(\xi) = \sum_{j=-l+1}^{k-l} \frac{(-1)^{k-l-j} f_j}{(j+l-1)!(k-l-j)!} P_j(\xi), \quad P_j(\xi) = \prod_{p=-l+1, p \neq j}^{k-l} (\xi-p) \quad (1)$$

である。 $f(\xi)$ の $\xi=0$ における m 階微分の差分表示 $\delta^m f$ は、 $f(\xi)$ の m 階微分係数 $f^{(m)}$ を求める $\xi=0$ における

$$\delta^m f = \frac{1}{h^m} \sum_{j=-l+1}^{k-l} a_j f_j, \quad a_j = \frac{(-1)^{k-l-j}}{(j+l-1)!(k-l-j)!} \delta^m P(0) \quad (2)$$

となる。以下に $m=1 \sim 4$ に対する係数 a_j を求めた結果を示す。ただし、 $\sum$ は $p=-l+1 \sim k-l$ で $p \neq 0, j$ に対する総和を意味する。

$$(1) m=1: a_0 = -\sum 1/p$$

$$a_j = -(-1)^j \Gamma_j / j$$

$$(2) m=2: a_0 = (\sum 1/p)^2 - \sum 1/p^2$$

$$a_j = (-1)^j (2\Gamma_j / j) \sum 1/p$$

$$(3) m=3: a_0 = -(\sum 1/p)^3 + 3(\sum 1/p)(\sum 1/p^2) - 2\sum 1/p^3$$

$$a_j = -(-1)^j (3\Gamma_j / j) [(\sum 1/p)^2 - \sum 1/p^2]$$

$$(4) m=4: a_0 = (\sum 1/p)^4 - 6(\sum 1/p)^2 \sum 1/p^2 + 8(\sum 1/p) \sum 1/p^3 + 3(\sum 1/p^2)^2 - 6\sum 1/p^4$$

$$a_j = (-1)^j (4\Gamma_j / j) [(\sum 1/p)^3 - 3(\sum 1/p)(\sum 1/p^2) + 2\sum 1/p^3]$$

ただし、 $\Gamma_j = (l-1)!(k-l)! / (j+l-1)!$

$(k-l-j)!$ である。

3. 常微分係数の差分表示プログラム

1~4階常微分係数の差分表示の係数 a_j を求める関数制プログラムを図-2に示す。 M は常微分係数の階数($0 \leq M \leq 4$)； K は用いる分点数； L は対象分点を含めた原点側の分点数($1 \leq L \leq K$)； J は係数を求める分点の局所番号。

$M=1 \sim 4$ のとき $J > K-L$ または $J < -L+1$ なら0となり、 $M=0$ のとき、 $J=0$ なら1.0、 $J \neq 0$ なら0となる。

$K \leq 15$ の場合、絶対値の小さい係数では最大2桁の計算誤差が混入するが、最大係数を基準にとれば計算精度

```

FUNCTION FDEXJS(M,K,L,J)
DIMENSION S(4)
IF(M.EQ.0) GO TO 150
IF(J.LT.-L+1) GO TO 160
IF(J.GT.K-L) GO TO 160
IF(MOD(M,2).EQ.0) GO TO 300
IF(K.NE.2*L-1) GO TO 1000
IF(J.NE.0) GO TO 1000
GO TO 160
300 IF(K.NE.2*L) GO TO 310
IF(J.NE.K-L) GO TO 1000
GO TO 160
310 IF(K.NE.2*L-2) GO TO 1000
IF(J.NE.-L+1) GO TO 1000
GO TO 160
1000 MM=M
IF(J.NE.0) MM=M-1
IF(MM.EQ.0) GO TO 100
DO 2 I=1,MM
S(I)=0.
DO 1 JL=1,K
IF(JL-L.EQ.J) GO TO 1
IF(JL-L.EQ.0) GO TO -1
DJ=JL-L
S(I)=S(I)+1.0/DJ**I
1 CONTINUE
2 CONTINUE
100 JA=IABS(J)
GAM=M
IF(J) 30,10,20
10 IF(M.EQ.1) A=-S(1)
IF(M.EQ.2) A=S(1)**2-S(2)
IF(M.EQ.3) A=(-S(1)**2+3.0*S(2))*S(1)-2.0*S(3)
IF(M.EQ.4) A=((S(1)**2-6.0*S(2))*S(1)+8.0*S(3))*S(1)
+3.0*S(2)**2-6.0*S(4)
GO TO 50
20 DO 3 JL=1,JA
T1=K-L-J+JL
T2=L+JL-J
3 GAM=GAM*T1/T2
GO TO 40
30 DO 4 JL=1,JA
T1=L+J+JL-1
T2=K-L+JL
4 GAM=GAM*T1/T2
40 DJ=J
IF(M.EQ.1) A=-GAM/DJ
IF(M.EQ.2) A=GAM/DJ*S(1)
IF(M.EQ.3) A=-GAM/DJ*(S(1)**2-S(2))
IF(M.EQ.4) A=GAM/DJ*((S(1)**2-3.0*S(2))*S(1)+2.0*S(3))
IF(MOD(JA,2).EQ.1) A=-A
50 IF(ABS(A).LT.1.E-5) A=0.0
GO TO 200
150 A=0.0
IF(J.EQ.0) A=1.0
GO TO 200
160 A=0.0
200 FDEXJS=A
RETURN
END

```

図-2

有の精度が確保されている。

4. 偏微分係数の差分表示プログラム

偏微分係数の差分表示

$$\frac{\partial^{m_x+m_y} f}{\partial x^{m_x} \partial y^{m_y}} = \frac{1}{h_x^{m_x} h_y^{m_y}} \sum_{i_x=-l_x+1}^{k_x-l_x} \sum_{j_y=-l_y+1}^{k_y-l_y} a_{j_x j_y} f \quad (3)$$

の係数 $a_{j_x j_y}$ を常微分係数の差分式の直積から求める関数副プログラムを図-4に示す。解析対象領域の対称性が自動的に考慮される。対称性に応じて、図-3に示すように座標軸を設け、斜線部のみを解析すればよい。

X, Y は x, y 軸に関する量であることを示す； I は対象分点の全体番号； J は係数を求める分点の全体番号； IS は所定の軸方向の対称性を示す整数で、対称性のないとき0、対称のとき1、逆対称のとき-1； $FDEXJ$ は常微分係数の差分表示を求める関数副プログラムで、図-2のほかに Taylor 級数展開法によるものでもよく、 $FDEXJ(M, K, L, J)$ の形をしていること。

常微分係数の場合、 X 軸を用い、 $MY = KY = LY = JY = ISY = 0$ とすればよい。

5. 時間の短縮

必要な常微分係数の差分表示を前もって作成しておき、必要なときに取り出すことで、時間の短縮が可能になる。

$K=3$ で $M=1, 2$ に対する差分係数と、 $K=5$ で $M=1 \sim 4$ に対する差分係数が必要な場合を例にとると、図-5のサブルーチン副プログラム $FDCOEFF$ を呼んで一度組立ておけば、以後、図-6の関数副プログラム $FDEXJL$ を図-2の $FDEXJS$ の代りに用いればよい。 $FDCOEFF$ において $FDEXJS$ が呼び出されている。

6. プログラミングの便宜のための副プログラム

(1) 偏心数 L

境界に近い分点で微分方程式を立てる場合、用いる長個の分点を仮想点を含めて設定された分点の範囲に納めるため、偏心の度合を修正する必要がある。図-7のサブルーチン副プログラムは通常、中央差分とみなし、境界に近い分点と偏心差分とみなし、偏心数 L を計算する。

N は分割数； I は微分方程式を立てる

ISY \ ISX	1	0	-1
1			
0			
-1			

図-3

```

FUNCTION FDSYMS(FDEXJ,MX,KX,MY,KY,LX,LY,IX,IY,JX,JY,ISX,ISY)
C
  TXP=FDEXJ(MX,KX,LX,JX-IX)
  TXN=0.0
  IF (ISX.EQ.0) GO TO 10
  IF (JX.EQ.0) GO TO 10
  TXN=FDEXJ(MX,KX,LX,-JX-IX)
  IF (ISX.LT.0) TXN=-TXN
C
  10 TYP=FDEXJ(MY,KY,LY,JY-IY)
  TYN=0.0
  IF (ISY.EQ.0) GO TO 20
  IF (JY.EQ.0) GO TO 20
  TYN=FDEXJ(MY,KY,LY,-JY-IY)
  IF (ISY.LT.0) TYN=-TYN
  20 FDSYMS=(TXP+TXN)*(TYP+TYN)
C
  RETURN
END

```

図-4

```

SUBROUTINE FDCOEFF
C
  COMMON
  * /MAT/C3(2,3,3),C5(4,5,5)
  DO 1 M=1,2
  DO 1 L=1,3
  DO 1 J=1,3
  1 C3(M,L,J)=FDEXJS(M,3,L,J-L)
  DO 2 M=1,4
  DO 2 L=1,5
  DO 2 J=1,5
  2 C5(M,L,J)=FDEXJS(M,5,L,J-L)
  RETURN
END

```

図-5

```

FUNCTION FDEXJL(M,K,L,J)
C
  COMMON
  * /MAT/C3(2,3,3),C5(4,5,5)
C
  IF (M.EQ. K) GO TO 4
  IF (M.EQ. 0) GO TO 3
  IF (J.LT.-L+1) GO TO 4
  IF (J.GT. K-L) GO TO 4
  JL=J+L
  IF (K.NE.3) GO TO 1
  FDEXJL=C3(M+L,JL)
  RETURN
  1 IF (K.NE.5) GO TO 4
  FDEXJL=C5(M+L,JL)
  RETURN
  3 IF (J.NE.0) GO TO 4
  FDEXJL=1.0
  RETURN
  4 FDEXJL=0.0
  RETURN
END

```

図-6

```

SUBROUTINE FDL(K,N,I,NF1,NF2,IS,L)
C
  L=MAX0(K/2+1,K-N-NF2+1)
  IF (IS.EQ.0) L=MIN0(L,I+NF1+1)
  RETURN
END

```

図-7

分点の全体番号；NF1, NF2 はそれぞれ $I = 0$, N の境界側の
仮想分点数。

(2) 差分係数が現われる範囲

分点 I で微分方程式を立てる場合、差分係数が現われる範囲は

図-8 のサブルーチン副プログラムで求められる。

$N1, N2$ は仮想分点も含めた全分点の範囲が $N1 \sim N2$ であることを示す。たとえば $N1$ より関数値が 0 の場合、この分点を含めなければ、計算の無駄が省ける。 $J1, J2$ (出力) は差分係数が現われる範囲を示す。DO 文のパラメタにそのまま使えるように全体番号+1 とされている。

7. むすび

等分割という制限はあるが、1~4 階の常微分係数に対する任意精度で任意偏心形の差分表示を陽の形で誘導し、FORTRAN プログラムを作成した。また、これを用いて偏微分方程式の差分表示が簡単に得られるプログラムも作成した。その他の小道具的なプログラムも含めてここに報告したプログラムは筆者の研究室で数年前から使われているが、仮想分点を設けず実分点のみを用いる方法²⁴⁾と併用することによって、問題のプログラミングとデバッグの労力が大幅に軽減され、研究の能率向上に大いに役立っている。

- 1) Sokolnikoff, I. S. : Mathematical Theory of Elasticity, 2nd ed., McGraw-Hill, 1956.
- 2) 三上市蔵：曲げを受ける薄肉断面げたの座屈に関する研究，名古屋大学提出学位論文，1972.
- 3) Silvester, P. : IEEE Trans. on Microwave Theory and Techniques, 18, pp. 740-743, Oct.,
- 4) Jensen, P. S. : Computers & Structures, 2, pp. 17-29, Feb., 1972. L 1970.
- 5) Perrone, N. and Kao, R. : Computers & Structures, 5, 1, pp. 45-57, 1975.
- 6) Petracic, M., Petracic, G. K. and Roberts, K. V. : Computer Physics Communication, Holland, 4, pp. 82-88, 1972.
- 7) Kumar, A. and Owen, J. B. B. : Proc. ICE, 57, Part 2, pp. 113-127, Mar., 1974. 「1975.
- 8) Csender, Z. J. : International Jour. for Numerical Methods in Engineering, 9, 3, pp. 581-599,
- 9) Wang, P. C. : Numerical and Matrix Methods in Structural Mechanics with Applications to Computers, John Wiley & Sons, 1966.
- 10) Sugihara, K. : Trans. of Architectural Institute of Japan, 147, pp. 9-15, 1968.
- 11) 杉原健児：日本建築学会論文集，158，pp. 13-17, 1969.
- 12) 杉原健児：日本建築学会論文集，161，pp. 1-6, 1969. 「1961.
- 13) Salvadori, M. G. and Baron, M. L. : Numerical Methods in Engineering, 2nd ed., Prentice-Hall,
- 14) Collatz, L. : The Numerical Treatment of Differential Equations, 3rd ed., Springer-Verlag, 1966.
- 15) Giencke, E. : Stahlbau, 36, 9, S. 260-268, 10, S. 303-315, 1967.
- 16) 佐武正雄：土木学会論文集，165，pp. 53-58, 1969.
- 17) 岸野佑次・佐武正雄：土木学会論文報告集，177，pp. 63-70, 1970.
- 18) 成岡昌夫・他：土木学会誌，44，7, pp. 73-80, 1959.
- 19) Köppel, K. und Schönbach, W. : Stahlbau, 27, 5, S. 122-125, 1958.
- 20) Falk, S. : ZAMM, 45, Sonderheft, p. T32, 1965.
- 21) Brocks, W. : Stahlbau, 42, 6, S. 172-176, 1973.
- 22) Show, F. S. : An Introduction to Relaxation Methods, Dover Publications, 1953.
- 23) Pflanz, E. : ZAMM, 17, 5, S. 296-300, 1937.
- 24) 三上市蔵：土木学会論文報告集，265，pp. 147-150, 1977.

```

SUBROUTINE FDJ(K,L,N1,N2,I,J1,J2)
C
J1=MAXO(N1,I-L+1)
J2=MINO(N2,I+K-L)
J1=J1+1
J2=J2+1
RETURN
END

```

図-8