I－3　一次元配列のＩＣＣＧ法とスカイライン法のパソコンへの適用について

北海道大学工学部　正員　佐藤　浩一
北海道大学工学部　正員　渡辺　昇
北海道大学工学部　正員　小幡　卓司
（株）釧路製作所　正員　井上　稔康

1．まえがき

半導体技術の進歩にともない、大型コンピュータをはじめ、スーパーコンピュータ、パーソナルコンピュータ（以下、パソコンという）などコンピュータの発達には目覚ましいものがある。最近、机上でかなり高性能の超小型コンピュータが使用できるようになった。その中で、16 ビットパソコンが主流をなしており、各分野で大量に使用されている。また、32 ビットパソコンも市販されている。1969年に人類が初めて月に一歩を記したアポロ11号の宇宙船を送り込むために使われたコンピュータの演算能力が現在の PC-9801 よりも劣っていたといわれている[1]。このような報告からみても最近のパソコンの性能はかなりのものと思われる。パソコン用の言語として、BASIC，FORTRAN，C，COBOL，PASCAL などがあり、種々の分野で種々の形で利用されている。パソコンでは BASIC 言語が主流であり、使いやすさという点で爆発的に普及している。一方、科学技術計算用に開発されたプログラム言語 FORTRAN は数値計算を主体とした理工学の分野で広く使用されており、FORTRAN の利用率はますます増え続けている。FORTRAN について幾つかの批判があるが、ライブラリーの種類が圧倒的に豊富であり、世界的に通用するという意味で長い生命を持つものと思われる[2]。従来、大型コンピュータでしか処理できなかつた FORTRAN も、最近のコンピュータ技術の進歩によって、手近かにあるパソコンでも処理できるようになった。これによって、BASIC などの攻撃によって一時下火になりかけていた FORTRAN も、再び勢いを盛り返してきているように思われる。パソコン用の FORTRAN コンパイラの代表格として、マイクロソフト社の MS-FORTRAN とデジタルリサーチ社の DR-FORTRAN がある。MS-FORTRAN は、Version 3.2 までは FORTRAN 77 の基本水準に準拠してつくられたものである。DR-FORTRAN と MS-FORTRAN の最新版では、上位水準までにも対応できるものとなっているようである。他に、PC-FORTRAN も普及しているようである。また､スクリーンエジィターとして MIFES や FINAL などがあり、近い将来はパソコンでも FORTRAN が本格的に利用されるものと思われる。

コンピュータに要求されるのは計算時間、精度、容量（メモリー）などであり、大型コンピュータはその点申し分ない。FORTRAN 言語をパソコンで使用するときの大型コンピュータと比較して欠点と思われる事項としては、次のような点を挙げることができる[3]。

(1) 使用ＣＰＵのアドレス空間がたとえば１ＭＢと制限があることから、余り大きなプログラムまたはデータエリアを必要とするプログラムを実行することが困難である。従って、配列宣言などをする時などに注意が必要である。
(2) 周辺機器が限定され、入力データはキーボードから入力するのが標準である。出力装置も現在いろいろあるが、まだ不十分である。
(3) 補助記憶装置としては、フロッピーディスクまたは固定ディスクが標準であり、このためコンパイルとリンクに時間を要する。

On Application of ICCG Method and Skyline Method by One Dimensional Array to Personal Computers
by Koichi SATO, Noboru WATANABE, Takashi OBATA and Toshiyasu INOUE

しかし、これらの欠点は、次第に解消されるであろう。将来パソコンと言えどもかなりパワーを持ったものが利用できるようになるであろう。今後は、コンピュータの利用形態として、スーパーコンピュータ、特殊プロセッサを搭載した専用機、パソコンという三極分化が進行するように思われる[3]。即ち、仕事に応じて選択することになるであろうが、その大部分の仕事がパソコンレベルで処理することが可能となることが予想される。また、科学技術の教育および研究においても、パソコンは益々重要な道具として位置付けられていくように思われる[4]。

通常、構造解析には大型コンピュータが利用されている。特に有限要素解析（ＦＥＭ）においてはマトリックスがキングサイズであり、不可欠である。しかしながら、パソコンを用いる場合でも、構造解析において多少工夫すればかなりのキングサイズのマトリックス演算が出来る。例えば、ＦＥＭにおける係数マトリックスは剛性マトリックスに他ならず、剛性マトリックスには、一般に次のような性質がある。性質１）キングサイズである。性質２）スパースである。性質３）バンドマトリックスである。性質４）対称マトリックスである。性質５）正定値マトリックスである。これらの性質を上手に利用すれば、メモリーをかなり節約出来る[5]。

本論文は、パソコンでの FORTRAN による数値計算環境の向上に鑑み、FORTRAN がもともと数値計算に適した言語であることも考慮して、FORTRAN のプログラムを BASIC に書き直すことはやめて、パソコンを用いて直接 MS-FORTRAN でメモリーを節約してプログラムを組むための一つの手法を報告するのが目的である。具体的には、メモリーを有効に利用するため、剛性マトリックスをＩＣＣＧ法およびスカイライン法用に一次元配列し、何元ぐらいまでのスパース行列の解析が可能かを検討するのが目的である。

２．三角分解、コレスキー分解、修正コレスキー分解、不完全コレスキー分解について

構造解析、特に有限要素法（ＦＥＭ）においては、剛性マトリックスはキングサイズであり、非対角成分がほとんど零となり、前述のような性質がある。一般には図－１のような配列になる。各列の最初の非零成分が何行目からであるかを求め、それらを結べば、図－１のような輪郭が描ける。この輪郭の中には零成分もありうる。いま、図－１におけるような正方マトリックス Ｋ を、左下三角マトリックス Ｌ と右上三角マトリックス Ｕ との積の形、すなわち

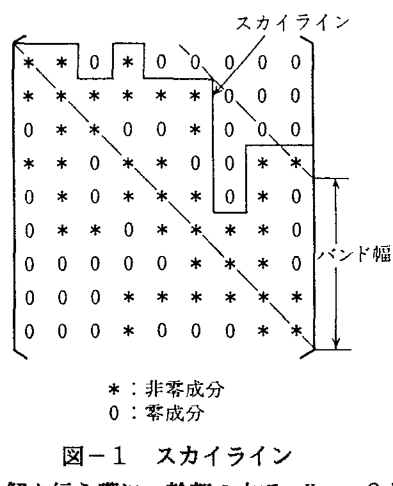
$$K = L \cdot U \qquad (1)$$

のように分解することを、三角分解という。左下三角マトリックスというのは、主対角より左下半分にだけ非零成分が存在する。逆に右上三角マトリックスというのは、主対角より右上半分にだけ非零成分が存在する正方マトリックスである。マトリックス Ｋ が対称の場合には式(1)の Ｕ を L^T（Ｌの転置行列）として

$$K = L \cdot L^T \qquad (2)$$

のように分解できる。これをコレスキー分解という。コレスキー分解では $U = L^T$ とおいたが、対角マトリックス Ｄ を導入して $U = DL^T$ のようにおくことも可能である。

$$K = L \cdot D \cdot L^T \qquad (3)$$



図－１　スカイライン

このような分解を修正コレスキー分解という。修正コレスキー分解を行う際に、輪郭の中で、$K_{ij} = 0$ ならば、$l_{ij} = 0$ とするのが、不完全コレスキー分解の”不完全”という意味である。つまりメモリー節約のために、もとのマトリックス成分が零だったところは、分解後も零とみなしてしまう[6]。有限要素法では非対角成分のほとんどが零となるため、こうした省略計算はかなりのスピードアップになるはずである。しかし、パソコンの場合は記憶容量が一番の問題だから、処理スピードよりもメモリーの節約の効果に期

待するところが大きい[5)]。

3．多元連立一次方程式の解法

多元連立一次方程式の数値解法には、大別して、1）直接法、2）反復法という2通りの方法がある。直接法において三角分解を用いる場合の解法は次のようである。多元連立一次方程式

$$Kx = b \qquad (4)$$

の係数マトリックス K を式(1)のように分解すると

$$L \cdot Ux = b \qquad (5)$$

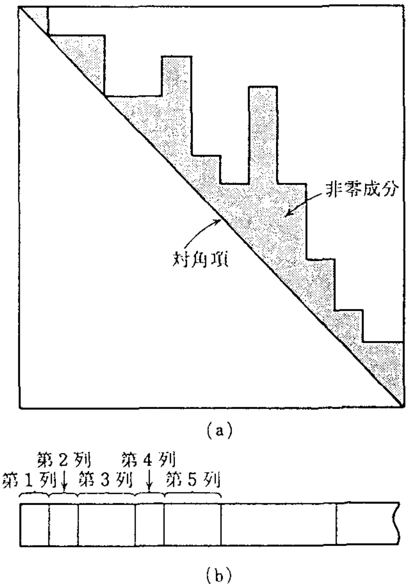
となる。ここで

$$Ux = y \qquad (6)$$

とおくならば、式(5)は

$$Ly = b \qquad (7)$$

となる。式(7) は y を未知数とする多元連立一次方程式とみなせる。この式は1行目から順に解いていけば、簡単に解ける。こうして y が得られれば、今度は式(6)を、x を未知数とする多元連立一次方程式とみなし、n行目から順に上方に向かって解いていけば、簡単に求まる。このように K を三角分解しておくと、多元連立一次方程式の解法は非常に簡単になる。



図－2　構造全体の剛性マトリックス

4．ＩＣＣＧ法とスカイライン法について

スカイライン法とは剛性マトリックスの列ごとにバンド幅を可変で設定したもので、図－1のような剛性マトリックスの非零成分の輪郭が都市のビル群のシルエットに似ていることからこの命名がある。このような剛性マトリックス（係数マトリックス）の輪郭内の成分だけを用いて前述の修正コレスキー分解し、図－2(a)のように各列の最初の非零成分と対角成分の間の領域のみを記憶し、図－2(b)のようにこれを一次元配列して解析する直接法である。この場合、輪郭内の零成分も一次元配列に入る。

一方、ＩＣＣＧ法の ＩＣ は Incomplete Cholesky(decomposition)、すなわち不完全コレスキー分解のことであり、また CG というのは Conjugate Gradient (method)、すなわち共役勾配法を意味している[5),6)]。従って、ＩＣＣＧ法を一言でいうならば、不完全コレスキー分解と共役勾配法とを組合せて、多元連立一次方程式を解く方法ということになる。この解法は直接法と反復法の長所を兼ね備えたもので、近年特に注目を集めている解法である。一般的な傾向として、直接法の方が反復法よりも有限要素法向きだと考えられてきた。しかしCGの前処理として不完全三角分解を用いる反復法の出現によって、こうした考えは見直されつつある。ＩＣＣＧ法は、そうした中でも特に有望な解法といえよう。ここでいう不完全三角分解とは、三角分解の過程で生ずる非零成分を無視して計算を進めることで、パソコンのように記憶容量がネックの場合には、メモリの節約は有効である。要するに係数マトリックスにこうした処理を施しておくと、多元連立一次方程式の解析が容易になる。しかしながら、三角分解が不完全ならば、それだけ精度が悪くなるのは仕方がない。従ってこれをCG法によって補うというのがＩＣＣＧ法という解釈も成り立つ。CG法は一般の反復法と異なり、理論上は有限回の反復によって厳密解に到達するといった特長がある。ＩＣＣＧ法における剛性マトリックス K の場合もスパース性を利用して、非零成分だけで構成される一次元配列とする（図－1の輪郭外の零成分はもちろん輪郭内にある零成分も一次元配列には入れない）。若干煩雑なプログラムになることは否めないものの、従来のバンドマトリックス法などに比べて、

メモリの大幅な削減が可能となり、その結果、キングサイズの問題も扱うことができる。要素番号や節点番号の付番の仕方に依存することなく、配列サイズは最小となるといったメリットがあるため、ＦＥＭにおいては効果的な方法である[5)]。

５．数値計算例とスカイライン法によるプログラミング

5-1. 数値計算例１　多元連立一次方程式の解法

図－３に示すような、1)キングサイズマトリックスであり、2)スパースマトリックスであり、3)バンドマトリックスであり、4)対称マトリックスであり、5)正定値マトリックスである多元連立一次方程式に関して何元ぐらいまでパソコンで解析することが出来るか検討するのが数値計算例１である。
MS-FORTRAN で PC-9801 VM2 と PC-9801 RA2 を用いて単精度で計算した。演算プロセッサーは使用していない。スクリーンエジィターとして FINAL と MIFES を用いた。計算結果として、スカイライン法で解いた場合 N = 19404 元まで計算することが出来た。ＩＣＣＧ法で解いた場合 N = 5404 元まで計算することが出来た。計算結果の印字は N1 = 100 とし、100 おきに印字している。この数値計算例の場合、単精度で十分であることがわかる。計算時間は VM2 で約４分、RA2 で約３分であった。

次のような多元連立一次対称方程式を解く。

$$\begin{Bmatrix}10\\12\\22\\22\\22\\22\\22\\22\\22\\22\\22\\22\\22\\22\\22\\22\\22\\22\\12\\10\end{Bmatrix}=\left[\begin{array}{l}
4\;3\;2\;1\\
3\;4\;3\;2\\
2\;3\;8\;6\;2\;1\\
1\;2\;6\;8\;3\;2\qquad\qquad\text{他は全部 0}\\
\quad\;\;2\;3\;8\;6\;2\;1\\
\quad\;\;1\;2\;6\;8\;3\;2\\
\qquad\quad\;2\;3\;8\;6\;2\;1\\
\qquad\quad\;1\;2\;6\;8\;3\;2\\
\qquad\qquad\;\;2\;3\;8\;6\;2\;1\\
\qquad\qquad\;\;1\;2\;6\;8\;3\;2\\
\qquad\quad\cdots\cdots\cdots\cdots\\
\qquad\quad\cdots\cdots\cdots\cdots\\
\qquad\qquad\qquad\;\;2\;3\;8\;6\;2\;1\\
\qquad\qquad\qquad\;\;1\;2\;6\;8\;3\;2\\
\qquad\qquad\qquad\quad\;\;2\;3\;8\;6\;2\;1\\
\qquad\qquad\qquad\quad\;\;1\;2\;6\;8\;3\;2\\
\quad\text{他は全部 0}\qquad\quad\;2\;3\;8\;6\;2\;1\\
\qquad\qquad\qquad\qquad\quad\;1\;2\;6\;8\;3\;2\\
\qquad\qquad\qquad\qquad\qquad\;\;2\;3\;4\;3\\
\qquad\qquad\qquad\qquad\qquad\;\;1\;2\;3\;4
\end{array}\right]\begin{Bmatrix}V(1)\\V(2)\\V(3)\\V(4)\\V(5)\\V(6)\\V(7)\\V(8)\\V(9)\\V(10)\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\V(n-2)\\V(n-1)\\V(n)\end{Bmatrix}$$

MS-FORTRAN を用いてスカイライン法で解く場合
入力データ　（ N は連立一次方程式の元数、 N1 は印字の間隔 ）
　N = 19404, N1 = 100
をファイル名 = S30DATA から読み込めば、
IND1(I),IND2(I),S7(I),DIS(I)は計算して自動的に入る (I=1,19404)。

MS-FORTRAN を用いてＩＣＣＧ法で解く場合
入力データ　（ N は連立一次方程式の元数、 N1 は印字の間隔 ）
　N = 5404, N1 = 100
をファイル名 = DATA12 から読み込む。

図－３　多元連立一次方程式

```
19404       67910
V(     1)=   .10000E+01
V(   101)=   .10000E+01
V(   201)=   .10000E+01
V(   301)=   .10000E+01
V(   401)=   .10000E+01
V(   501)=   .10000E+01
V(   601)=   .10000E+01
V(   701)=   .10000E+01
V(   801)=   .10000E+01
V(   901)=   .10000E+01
V(  1001)=   .10000E+01
V(  1101)=   .10000E+01
V(  1201)=   .10000E+01
V(  1301)=   .10000E+01
V(  1401)=   .10000E+01
V(  1501)=   .10000E+01
V(  1601)=   .10000E+01
V(  1701)=   .10000E+01
V(  1801)=   .10000E+01
V(  1901)=   .10000E+01
V(  2001)=   .10000E+01
V(  2101)=   .10000E+01
V(  2201)=   .10000E+01
V(  2301)=   .10000E+01
V(  2401)=   .10000E+01
V(  2501)=   .10000E+01
V(  2601)=   .10000E+01
V(  2701)=   .10000E+01
V(  2801)=   .10000E+01
V(  2901)=   .10000E+01

V(19001)=   .10000E+01
V(19101)=   .10000E+01
V(19201)=   .10000E+01
V(19301)=   .10000E+01
V(19401)=   .10000E+01
```

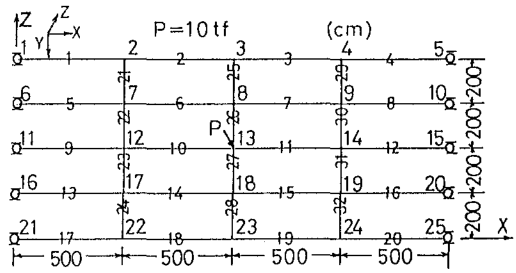
計算結果

次に、スカイライン法によるプログラムを示す（数値計算例１はこれによっている）。ＩＣＣＧ法は省略。

```
* スカイライン法による解法 *
* ファイル名 = SKY30.FOR *
* 単精度 *
*    N         = 連立方程式の元数
*    NN        = 非ゼロ要素の数
*    N1        = 印字する間隔
*    IND1(J)   = 各列(J列)の最初に非
*                ゼロ成分である行番号
*    IND2(J)   = 各列(J列)の対角成分の
*                一次元配列における番号
*    DIS(I)    = 荷重項(列ベクトル)
      PARAMETER (NX=19404,NZZX=67910)
      COMMON /A/ IND1(NX),IND2(NX)
     *           ,S7(NZZX),DIS(NX)
      OPEN(5,FILE='B:S30DATA')
      OPEN(6,FILE='B:RESULT'
     *    ,STATUS='NEW')
 9999 READ(5,101,END=1001) N,N1
      NN=(N-2)/2*3+(N-2)/2*2+1+N
      DO 11 I=6,N,2
      IND1(I-1)=I-3
   11 IND1(I)  =I-3
      IND1(1)  =1
      IND1(2)  =1
      IND1(3)  =1
      IND1(4)  =1
      N2=N/2-1
      DO 12 I=1,N2
      S7( 4+7*(I-1))=2.
      S7( 5+7*(I-1))=3.
      S7( 6+7*(I-1))=8.
      S7( 7+7*(I-1))=1.
      S7( 8+7*(I-1))=2.
      S7( 9+7*(I-1))=6.
   12 S7(10+7*(I-1))=8.
      S7(1)          =4.
      S7(2)          =3.
      S7(3)          =4.
      S7(NN-4)       =4.
      S7(NN-1)       =3.
      S7(NN)         =4.
      DO 13 I=1,N
   13 DIS(I)  =22.
      DIS(1)  =10.
      DIS(2)  =12.
      DIS(N-1)=12.
      DIS(N)  =10.
      WRITE(6,101) N,NN
  101 FORMAT(5I10)
* IND2(I)を求める *
      IND2(1)=1
      DO 1201 J=2,N
 1201 IND2(J)=IND2(J-1)+(J-IND1(J))+1
* スカイライン法による解法 (LU分解) *
      DO 8000 K=2,N
       WRITE(*,'(I5¥)') K
       IF(IND1(K).EQ.K) GOTO 8000
       IDX=IND2(K)-K
       JS=IND1(K)
       JE=K-1
       DO 5100 J=JS,JE
        IF(IND1(K).EQ.J) GOTO 5100
        INIT=IND1(K)
        IF(IND1(J).GT.INIT) INIT=IND1(J)
        T=0.0
        LNJ=IND2(J)-J
        IE=J-1
        DO 7200 I=INIT,IE
         IDXSAV=IDX+I
         SS1=S7(IDXSAV)
         LNJSAV=LNJ+I
         SS2=S7(LNJSAV)
         T=T+SS1*SS2
 7200   CONTINUE
         LJK=IND2(K)-K+J
         S7(LJK)=S7(LJK)-T
 5100   CONTINUE
        T=0.0
        IS=IND1(K)
        IE=K-1
        DO 5400 I=IS,IE
         LNSAV=IND2(I)
         IDXI=IDX+I
         SY1=S7(IDXI)
         SY2=S7(LNSAV)
         IF(ABS(SY2).LE.1.0E-30) STOP
         SYKSAV=SY1/SY2
         T=T+SY1*SYKSAV
         S7(IDXI)=SYKSAV
 5400   CONTINUE
        LNSAV=IND2(K)
        PIV=S7(LNSAV)-T
        S7(LNSAV)=PIV
 8000 CONTINUE
      DO 7000 K=2,N
       T=0.0
       IF(IND1(K).EQ.K) GOTO 7000
       IDX=IND2(K)-K
       IS=IND1(K)
       IE=K-1
       DO 7100 I=IS,IE
        IDXI=IDX+I
        SY=S7(IDXI)
        T=T+SY*DIS(I)
 7100  CONTINUE
       DIS(K)=DIS(K)-T
 7000 CONTINUE
      DO 2200 I=1,N
       LNSAV=IND2(I)
       SYS=S7(LNSAV)
       IF(ABS(SYS).LE.1.0E-30) STOP
       DIS(I)=DIS(I)/SYS
 2200 CONTINUE
      DO 2300 KK=2,N
       K=N-KK+2
       IF(IND1(K).EQ.K) GOTO 2300
       IDX=IND2(K)-K
       IS=IND1(K)
       IE=K-1
       DO 2400 I=IS,IE
        IDXI=IDX+I
        SY=S7(IDXI)
        DIS(I)=DIS(I)-SY*DIS(K)
 2400  CONTINUE
 2300 CONTINUE
      WRITE(6,1123) (I,DIS(I),I=1,
     *               N,N1)
 1123 FORMAT(5X,'V(',I5,')=',E12.5)
      GO TO 9999
 1001 STOP
      END
```

5-2. 数値計算例２　格子桁の解析例

前述のプログラムは多元連立一次方程式を解くだけのプログラムである。しかしながら、有限要素解析には、要素の剛性マトリックスおよび全体の剛性マトリックスの作成、さらに境界条件の処理などのプログラムが必要である。その一例として、図－４に示すような、曲げ剛性とねじり剛性を考慮した主桁５本、横桁３本の格子桁について解析した。節点数２５，要素数３２である。断面二次モーメントは $J=94000000\ cm^4$，$J_Q=43000000\ cm^4$，ねじり定数 $J_T=39800000\ cm^4$，$J_{QT}=15000000\ cm^4$ とした。計算時間は３０秒であった。単精度と倍精度で行った計算結果の一部を示す（たわみのみ）（表－１）。なお、格子桁に関して、曲げ剛性とねじり剛性を考慮した場合、主桁１５本、横桁４５本まで、節点数７０５、要素数１３２０まで解析出来ることを確認している。計算時間は VM2 で 約５５分、RA2 で約１８分であった。



図－４　格子桁

表－１　格子桁の解析結果（cm）

	単精度	倍精度
節点番号	Ｙ方向のたわみ	Ｙ方向のたわみ
12	0.1232230E-02	0.1232228E-02
13	0.1853370E-02	0.1853368E-02
14	0.1232230E-02	0.1232228E-02

なお、MS-FORTRAN によるパソコン用として、連続桁、平面トラス、平面ラーメン、不完全合成桁、有限要素法（二次元弾性問題）、格子桁など種々の構造解析プログラムを作成しているが、詳細については、別の機会に発表する予定である。

６．あとがき

本論文の数値計算例１で示したように、約３分程度で 19404 元のかなりキングサイズの多元連立一次方程式（但し、スパース行列とする）をパソコンで解くことが出来た。ここでは示していないが、種々の計算結果より計算精度は大型計算機と同等と思われる。本論文のパソコンを用いたＩＣＣＧ法およびスカイライン法は FORTRAN 77 に準拠しているので、メモリーがオーバーするような場合は大型コンピュータに簡単に移せるので、さらに大規模の連立一次方程式（但し、スパース行列とする）を解くことが可能である。パソコンでFORTRAN 言語を用いて実行する場合のその他の使用法としては、パソコン処理には適しない非常に時間のかかる計算で最終的に大型コンピュータで処理しなければならないが、部分的なプログラム単位ごとのプログラミングおよびディバックにパソコンを活用し、大型コンピュータに実行させる前にアルゴリズムの良し悪しを予め検証しておくことにより、大型コンピュータによる全体のプログラミングおよびディバッキングを効率的に行うことも可能である。

（参考文献）

1) テクニカルライダーズ．Ｅ編著：ＭＳ－ＤＯＳこんな時どう使う、日本文芸社、1988.
2) 島田静雄、高木録郎、和田匡夫、清水雅夫：パソコンによる橋のＣＡＤ　合成桁編、山海堂、1989.
3) 穴吹雅敏、竹本宜弘、西海英雄、松尾守之：FORTRAN 77 応用、東海大学出版会、1986.
4) 小島紀男、町田東一：FORTRAN 基礎数値計算、東海大学出版会、1989.
5) 下関正義、藤沼平一：PC-9801 有限要素法／非定常熱応力プログラミング、日刊工業新聞社、1988.
6) 森正武：FORTRAN 77 数値計算プログラミング、岩波書店、1989.
7) 三好俊郎：MS-FORTRAN による有限／境界要素解析プログラミング、サイエンス社、1985.
8) 平居孝之：有限要素法と境界要素法　パソコンによる大容量弾性解析、共立出版株式会社、1988.