解

説

溶接技術へのマイクロ・コンピュータの利用(Ⅲ)

マイクロ・コンピュータによる数値解析*

Application of Microcomputer to Welding Technology (Ⅲ)

Numerical Analysis Using Microcomputers*

戸 川 隼 人**

by Hayato Togawa**

キーワード：マイクロコンピュータ, 数値解析, 有限要素法, PCG 法, ソフトウェア

1. は じ め に

最近, どの研究室にも, どの現場にもマイクロ・コンピュータが置かれ,

データ収集 (データ・ロガー)

データ解析 (アナライザー)

計 算 (コンピュータ)

製 図 (CAD)

文書作成 (ワード・プロセッサ)

などに広く使われるようになった。この中の「計算」(特に数値解析)への応用について

どんな問題に使えるか?

どの程度の規模まで可能か?

使い易さや計算時間は?

ソフトウェアの開発状況は?

利用のテクニックは?

といった諸点を解説するのが本稿の目的である。

2. マイクロ・コンピュータの現状

数値解析をマイクロ・コンピュータで行う場合の主な問題点は,

限られた記憶容量

限られた演算速度

限られた言語

の制約の中で, いかにして実用規模の解析を行うかということである。数年前までは, この種の制約が非常に強かった。しかし, マイクロ・コンピュータの技術的進歩はきわめて速く, 現在では上記のような困難は大幅に緩和されている。

まず, 主記憶容量については表 1 のように「1年に約

表 1 記憶容量の増大傾向

機 種	発売年代	主記憶容量(標準)	同(最大)
TK-80	1977	256 バイト	1 KB
PC-8001	1980	16 KB	32 KB
PC-8801	1982	64 KB	128 KB
PC-9801	1983	128 KB	640 KB
FM-16 β	1985	512 KB	1 MB

2 倍」といった急成長が続いた結果, 今日では数百キロバイトないし 1 メガバイトが普通になっている。これはひと昔前の中～大型機の容量に匹敵し, 現在の大型機でも一般利用者のユーザ領域 (すなわち, 一人で自由に使える領域) はこの程度である。

ここでバイト (byte) というのは, コンピュータの内部で情報を表す符号の長さの単位で,

1 バイトは 8 ビット (2 進法の 8 桁)

1KB (キロバイト) は千バイト

1MB (メガバイト) は百万バイト

英数字やカナ文字は 1 字が 1 バイト

漢字は 1 字が 2 バイト

単精度の数値は 4 バイト (32 ビット)

倍精度の数値は 8 バイト (64 ビット)

であるから, たとえば記憶容量が 640 KB あれば, 単精度の数値を 16 万個 (400×400 マトリックスの 1 個分) 記憶できるわけである。

外部記憶装置としてのフロッピー・ディスクの容量も表 2 のように年々大きくなり, 最近では更に大容量のハード・ディスク (固定ディスクともいう。5～20 MB) も広く用いられるようになってきた (一部の機種では標準仕様で装置されている)。

大規模計算の例として, 有限要素法の計算におけるメモリー必要量を考えてみると, 解法としてバンド・マト

表 2 5 インチ・フロッピー・ディスクの容量

片面単密度 (初期のころ)	70 KB / 枚
両面倍密度 (最近まで)	320 KB / 枚
両面高密度 (最近発売)	1 MB / 枚

*原稿受付 昭和61年 1 月 20 日

** 日本大学理工学部数学科 Faculty of Science and Technology, Nihon University

表 3 有限要素法の計算に必要な記憶容量の例

問題の例	平面応力解析	3 次元熱伝導
分 割	20×30	10×10×10
節点数	600	1000
自由度	1200	1000
バンド幅	40	100
必要容量 (KB)	192	400

リックス法を使う場合、表 3 のようになる。前述のとおり、最近の機種には 1 MB 近くのメモリーがあるから、表 3 の程度ならば、あまり無理なく処理できる。ハード・ディスクでバック・アップしたり、後述の PCG 法を用いるなどすれば、表 3 の 10 倍ぐらいの問題まで計算が可能である。

演算速度も近年かなり向上した。これは

語長の増大 (8 ビット→16 ビット)

クロック周波数の向上 (1 MHz→10 MHz)

浮動小数点演算付加装置 (i 8087 等)

質のよいコンパイラの普及

などによるものであり、例えば 30 元の連立 1 次方程式 (非対称密行列の場合) を解くための所要時間は、表 4 に示すように、以前の約 1/70 に短縮されている。それでも、この表は浮動小数点演算付加装置 8087 を使用していない場合の値であって、もしこれを利用し、より速い CPU (たとえば V 30) と組合せれば、現在でももっと高速の処理が可能である。

大型機と比べたら、もちろん、相当に遅いが、小～中型機と比べると、さほど遜色はなく、旧式の機種よりは速いこともある。それにマイクロ・コンピュータは一人で占有して使うから、大規模の計算の際に何時間も (場合によっては数日) かかっても良いわけで、小さくて遅い割には、かなり大きな計算まで実行できる。

演算精度に関しては、中～大型機とだいたい同じであるが、詳しくいうと表 5 のような事情によりマイクロ・

表 4 処理速度の向上を、30 元の連立 1 次方程式の計算所要時間で見た例

使用した機種および言語処理系	所要時間
Level 1-3 BASIC	420 秒
FM-7 BASIC	190 秒
PC-9801 F (8 MHz)	
N 88-BASIC インタプリータ	52 秒
N 88-BASIC コンパイラ	17 秒
Lattice C コンパイラ	6 秒

表 5 実数型データの内部表現の比較

	マイクロ・コンピュータ	中～大型機
数値表現	2 進法	16 進法
仮数部の長さ	(1 ビット長い)	(標準)
丸め的方式	4 捨 5 入	切り捨て

コンピュータの方が少し有利になっている。

次に、プログラミング言語について見ると、現在は主に BASIC が使われている。BASIC といっても、以前の (中～大型計算機では今でも昔のままであるが) とは全く異なり、使い易く強力な機能をもつデラックス仕様の BASIC である。特に

グラフィック表示

対話的入出力

文字列処理

などの面では FORTRAN よりもはるかに便利であり、また、操作が簡単でデバッグも容易である。実行速度の遅いのが欠点であったが、N 88 BASIC コンパイラの出現により、インタプリタでデバッグしたプログラムをそのまま高速で実行できるようになった。

それでも、過去に中～大型機で開発されたプログラムをマイクロ・コンピュータに移植して活用するには FORTRAN の方が望ましい。初期の頃、マイクロ・コンピュータの上で FORTRAN を使えなかった主な理由は、メモリーの容量が十分でなく、ディスクも普及していなかったためであるが、近年はそのあたりの事情が全く変わり、優秀な FORTRAN コンパイラが作られ、広く使われるようになってきた。現在、比較的良好に使われているのは MS-FORTRAN である。これは、FORTRAN 77 の標準サブセット・レベル (プラス α) であり、結構速い。最近発売された RM/FORTRAN は 77 のフルセットで、複素数型 (単精度、倍精度とも) を使用でき、大型機のプログラムがそのまま走る。

ASCII ファイル (標準的な文字列形式の順編成ファイル) を仲介にすれば、FORTRAN と BASIC の間のデータの受け渡しが簡単にできるので、入出力の部分は BASIC で書き、計算の部分は FORTRAN で書く、というような方式をとることもできる。

2. 基本的な数値計算

関数近似、補間、数値積分、 $f(x)=0$ の根、連立 1 次方程式、固有値・固有ベクトル、常微分方程式…などの基本的な数値計算は、マイクロ・コンピュータでも容易にできる。記憶容量や演算速度については (極端に大規模または悪条件でない限り) ほとんど問題ないといつてよい。参考までに、簡単に解ける問題の例とその所要時間の例を表 6～表 8 に示す。

多少問題があるとすれば、数値計算ライブラリであろう。大型計算機の場合には、たいてい、メーカー提供または計算センター提供の数値計算ライブラリ (SSL, MSL などと名前はいろいろある) があって、計算の道具とし

表6 連立1次方程式を解くための所要時間
(ガウスの消去法, PC-9801 F 使用)

	BASIC コンパイラ		BASIC インタプリタ	
	単精度	倍精度	単精度	倍精度
10	1秒	1秒	2秒	5秒
20	5秒	7秒	17秒	32秒
30	17秒	21秒	52秒	2分
40	40秒	47秒	2分	4分
50	1分	1分半	4分	8分
60	2分	2分半	7分	13分
70	3分半	—	11分	20分
80	5分	—	—	—
90	7分	—	—	—
100	10分	—	—	—
110	13分	—	—	—
120	17分	—	—	—

表7 高次代数方程式を解くための
所要時間 (ベアストウ法,
PC-9801 F 使用)

次 数	所要時間
5	1秒以下
10	1秒
20	7秒
30	18秒

表8 実対称マトリックスの固有
値, 固有ベクトルを全部求
める計算の所要時間 (ヤコ
ビ法, PC-9801 F 使用)

次 数	所要時間
5	2秒
10	18秒
20	3分
30	10分
50	1時間
70	2時間半

てブラック・ボックスの感じで使うことができる。それに対してマイクロ・コンピュータの場合にはそれが無いため、一般利用者が各自で計算法を勉強してプログラムを作らなければならなかった。

しかし、先頃、IMSL という世界で最も評価の高い数値計算ライブラリのマイクロ・コンピュータ版が流通し始め、この問題は一挙に解決された。もっとも、現在のところまだ機種が限られ (PC-9801 と IBM だけ)、言語も限られ (MS-FORTRAN 用)、価格もあまり安くはないけれども (十数万円)、これで大型計算機なみの数値計算ライブラリを使えるようになったわけである。大型計算機版の IMSL よりはプログラムの本数が少ないのであるが、標準的なものは一通りそろっており、実用上は十分といってよい。

BASIC の数値計算ライブラリは、たまに広告をみかけることがあるが、「使ってみた」、「良かった」という

ような話をあまり聞かない。BASIC ではサブルーチンに引数を渡せないで FORTRAN と同じような形式の数値計算ライブラリは作れない。OS-9 の BASIC ならば引数を渡せるし分割コンパイルもできるので、これを使ってライブラリを作ろう、と開発を手がけたこともあるが、残念ながら中止になったまま今日に至っている。筆者は最近 BASIC 数値計算プログラム集 (サイエンス社) という本を書いた。数学ソフトウェアといえるような立派なものではないが、大型計算機のライブラリーの単なる焼きなおしではなく、マイクロ・コンピュータの特長を生かし、マイクロ・コンピュータの利用場面における使い易さを追求してみたもので、多少はお役に立つかもしれない。

3. 有限要素法の計算

有限要素法は、その特長であるところの著しい汎用性を実現するために、広大な記憶領域と多大な計算時間を必要とする。したがって、本質的には「大型計算機向き」の算法である。しかし同時に、有限要素法のデータ生成や結果の表示等には、対話形式の利用やグラフィック表示が重要であり、そのような面ではマイクロ・コンピュータが優れている。また、いつでも手許にあって自由に使える、ということも望ましい。将来コンピュータ通信がもっと普及して、手軽に大型機を利用できるようになれば別であるが、現状ではマイクロ・コンピュータによる有限要素解析はなかなか魅力的である。

問題は、どの程度の規模まで計算できるか、という点であるが、その限界は主にメモリーの大きさと演算速度で決まってくる。まず、メモリー・サイズであるが、話を簡単にするため、

解法はバンド・マトリックス法

語長は単精度 (4 バイト = 32 ビット)

メモリーのユーザ領域が 512 KB

とすると、剛性マトリックスに使用できる語数は、約 $512,000/4=128,000$ 語

であるから、例えば 1,000 自由度ならばバンド幅は最大 128 となる。この程度あれば、簡単な 2 次元問題の解析には十分といってよいであろう。ハード・ディスクを使えば、もう一まわり大きな問題を扱うことができる。

一方、計算時間は、自由度、バンド幅のほか、要素の種類、解法、プログラムの書き方、それから、もちろん、使用する機種やプログラム言語の処理システムなどに関係するので一概にはいえないが、後述の PCG 法を用いれば表 9、表 10 に示すような時間で結果を得ることができる。大型計算機にくらべれば遅いが、計算センター

表9 PCG 法による計算の例

使用機種: PC-9801 F (8 MHz)
 メモリー: 640 KB に増設して使用 (RAM 内だけで計算)
 使用言語: N 88-BASIC (コンパイラ)
 問題: 3次元ラプラス方程式
 解法: 差分法 (7点差分)
 分割: $20 \times 20 \times 20$
 自由度総数: 9261
 普通の意味のバンド幅: 441
 PCG 法の実質バンド幅: 4
 所要時間: 53分

表10 同様な問題でもう少し分割が粗い場合の所要時間

分割	自由度数	所要時間
$4 \times 4 \times 4$	125	約20秒
$8 \times 8 \times 8$	729	2分
$10 \times 10 \times 10$	1331	6分
$16 \times 16 \times 16$	4918	25分

に出掛けていって使うことを思えば、まあまあといったところであろう。PCG 法を使わない場合には、もう少し時間がかかるが、1例を示せば、

長方形領域上の定常熱伝導問題

三角形1次式要素

441節点 (882自由度)

バンド幅 21

所要時間	係数計算	2分
	連立1次方程式	20分
	グラフィック表示	4分

(PC-9801 F 使用) といったぐあいである。アイソパラメトリック要素の場合の例は表11にある。

固有値解析の所要時間は、ごく大雑把にいった、静的解析の3~5倍と見てよい。時間積分法による応答解析や、増分法による非線形解析は、ステップ・サイズや計算範囲によって当然大きく左右されるが、1ケースに数時間 (あるいはそれ以上) を覚悟しておく必要があり、もし大型機を使える環境にいるなら、それを使う方が賢明であろう。

4. 有限要素法のソフトの現状

マイクロ・コンピュータのメモリーが大きくなり、さらに FORTRAN コンパイラを使用できるようになったので、マイクロ・コンピュータ用の有限要素法ソフトは、今、転換期を迎えている。従来のマイクロ・コンピュータの有限要素法ソフトウェアは

メモリーが小さい

言語は BASIC しか使えない

という環境のもとで動くことに存在意義があった。マイ

クロ・コンピュータ自体が低いので、本格的な計算はもともと無理で、ごく簡単なこと (たとえば平面骨組解析) ができればよかった。

ところが最近ではマイクロ・コンピュータの性能が良くなったので、より高度な処理、より豊富なエレメント・ライブラリ、より大きな計算規模等への要求が強くなってきた。

そうになると、新規にプログラムを開発するよりも、既存の汎用プログラムを移植する方が早道である。多少旧式であっても、ユーザーは信頼性の高い、名の通った汎用プログラムを歓迎するであろう。現在、まだ大型汎用プログラムをそのまま実行できるところまではいかないが、小~中規模の汎用プログラムに関してはすでに移植が試みられている。

例えば、先頃、ANSYS の PC-9801 版が発売されて話題になっている。マイクロ・コンピュータの性能の限界のため、非線形解析の機能は省かれているが、線形解析に関しては、3次元解析が可能でソリッド要素やシェル要素を使用できるし、固有値解析もできるようになっている。また、当然、データ生成やグラフィック表示機能も完備している。

一方、SAP は IBM-PC に移植されて、順調に稼動中であり (ただし、もう少し機能強化してから発売になるらしい)、NASTRAN-PAL という、有名な NASTRAN の学生用小型版も発表されている。マーク社もマイクロ・コンピュータ用の有限要素法の汎用ソフトウェアを開発中という。

考えてみると、現在使用されている汎用プログラムには1970年前後に開発されたものが多いが、当時の大型計算機の主記憶装置の容量はあまり大きくなく、今日のマイクロ・コンピュータと大差ないのであるから、その上で開発されたソフトがマイクロ・コンピュータ上で動くのは不思議ではなく、むしろ当然といってよい。したがって、ソフトに関しても、もう間もなく大型計算機なみの状態になることが期待できる。

5. 数値計算法の最近の進歩

5.1 PCG 法とその応用

計算法に関しては近年、注目すべき進歩があった。それは、PCG (preconditioned conjugate gradient) 法である。

もともと大規模計算の手法として開発され、主としてスーパー・コンピュータで使われてきたが、マイクロ・コンピュータにもきわめて好適で、これをうまく用いると、1万円ぐらいまでの方程式を比較的短時間で解くこ

とができる。以下でその要点を説明する。

■CG (conjugate gradient, 共役勾配) 法は連立 1 次方程式の反復解法の一つで、最高度にチューニングされた SOR (successive over-relaxation, 逐次過剰緩和) 法と同じくらい優れた収束率をもつ (たとえば拙著, 「共役勾配法」, 教育出版, 参照)。

■しかし CG 法は丸め誤差に弱く, 実用解法としては問題がある, ということが以前から指摘されており, その対策として, 前処理により条件数を改良してから CG 法を適用することが「実用上の常識」とされていた。これが PCG 法 (前処理付きの CG 法) の始まりであるが, 初期の PCG 法はダイアゴナル・スケーリング (対角成分によるスケーリング) ないしはそれに近いものであったため, 効果は十分でなかった。

■1977年, Meijerink と van der Vorst は不完全三角分解を用いる前処理方法を提案した²⁾。

不完全分解というのは, 三角分解の過程で発生する非ゼロ要素 (fill in) を無視して計算を進めることで, これを用いて解を求めれば当然誤差が入るが, PCG 法ではそれを CG 法で修正しようというわけである。

具体的には, 次のように処理する。

- 1) 三角分解 (ガウスの消去法における前進消去の計算に相当) の際に, 係数行列の非ゼロ部分に関しては (たとえバンド内であっても) 計算を省略し, 当然, 書きこみもしない。こうして連立 1 次方程式

$$Ax=b$$

の係数行列 A を不完全分解した結果を L, U とする。

- 2) 適当に出発ベクトル x を選び, 残差

$$q=b-Ax$$

を求め, 方程式

$$Lz=q$$

を解いて z を求め, 方程式

$$Ur=z$$

を解いて r を求め,

$$p=r$$

とする。

- 3) 十分な精度が得られるまで以下の手続きを反復

$$q=Ap$$

方程式 $Lz=q$ を解いて z を求める

方程式 $Uy=z$ を解いて y を求める

$$\alpha=(r, p)/(p, y)$$

$$\text{新 } x=x+\alpha p$$

$$\text{新 } r=r-\alpha p$$

$$\beta=(\text{新 } r, \text{新 } r)/(\text{旧 } r, \text{旧 } r)$$

$$\text{新 } p=r+\beta p$$

注) C 言語によるプログラム例が文献 3) にある。

■上記は A が対称行列の場合のための公式である。A が非対称行列である場合のための公式としては PCR (preconditioned conjugate residual) 法がある。これは, 要するに非対称行列用の CG 法の公式と不完全分解の組合せである⁴⁾。

■Gustafson は不完全分解の際の「無視される項」を対角項にくりこむことにより不完全計算に伴う誤差を補正する方法を提案した。これはかなり有効である⁵⁾。筆者の追試結果では必要反復回数が 2~3 割少なくなっている。

■不完全分解による PCG 法を用いると, バンド幅いっぱいメモリーをとる必要がなく, 対角にごく近い少数の成分と, バンド外縁に近い 1~2 の成分 (場合によっては, それ以外のいくつかの成分) などだけを記憶させればよいので, 必要記憶容量はバンド・マトリックス法の数分の一ないし数十分の一で済むわけで, マイクロ・コンピュータの限られたメモリー・スペースの中で大きな問題を解くための手段として, 極めて有効である。但し, その代償として CG 法反復のための時間がかかる。これは問題の性質によって多少変るが, 比較的性質のよい問題の場合だと, 必要反復回数は 15~20 回程度なので, あまり負担にはならない。

筆者が試みた計算の 1 例の, 規模と所要時間の記録を表 9, 表 10 に示す。

このように, 数千円の方程式も難なく解ける。このことは実用上, 大きな意味をもつ。但し, 上の例は 1 節点 1 自由度である関係上, 実質バンド幅が狭く, PCG 法にとっては最も有利な条件になっている。弾性問題等の場合には, 節点自由度が 2 以上になるので, RAM 内で計算できる規模はもう少し小さくなり所要時間も数倍になる。アイソパラメトリック要素 (8 節点要素) の計算もしてみたが, やはりバンド内の非ゼロ要素数が多いので, 計算時間がかかった。といっても, バンド・マトリックス法よりは, はるかに記憶場所が少なくて済み, 有利である。

実行を眺めていて感じることは, ふつうの消去法の計算と違って, 三角分解に要する時間が極めて短く, 右辺の計算の方に時間がかかることで, そのため, サブスペース法などには, あまり向かないのではないかと思う。

それに対して非線形問題の増分法などの場合には, 毎回の再分解が負担にならないので好都合である。なお, PCG 法を用いる際には, なるべく規則的な分割を採用

すべきで、プログラムで規則性をできるだけ活用すると能率のよい計算ができる。

5.2 高次要素の使用

これは「新しい手法」というわけではないが、普及が遅れている傾向があるので、ちょっと言及しておこうと思う。

高次要素を用いると、比較的粗い分割で高い精度が得られるので、要素数が少なく済み、データ作成が楽になる。そのかわり、計算式が複雑で、プログラムが長くなり、要素マトリックスも大きいので、以前はマイクロ・コンピュータで処理するのが困難であった。しかし、最近ではマイクロ・コンピュータのメモリーが大きくなったので、無理なく実用規模の計算ができる。

割合に簡単に作れて効果が大きいのは、2次元問題のための4辺形8節点のアイソパラメトリック要素である。参考までに、PC-9801 F による計算例の規模と所要時間を表11に示す。問題はラプラス方程式、領域は正方形である。プログラム・リストは文献6)にある。アイソパラメトリック要素と PCG 法を組合せたプログラムも作ってみたが、この場合にはバンド内の非ゼロ要素が多いので、低次要素の場合のような大きな効果は見られなかった（それでも、メモリーが何割かで済むので、利用価値はある）。

表11 アイソパラメトリック要素による計算の実行時間の例（バンド・マトリックス法で計算、N 88 BASIC インタプリータ使用）

分 割	節 点 数	バ ン ド 幅	所 要 時 間
15×8	407	29	52分
10×10	341	35	45
8×8	225	29	18
6×6	133	23	10
4×4	65	17	3

5.3 補外法による精度改良

問題とする領域の形が単純で、規則的分割を用いた場合には、粗い分割で計算した結果と、細かい分割で計算した結果とを組みあわせて、誤差評価や精度改良を行う

ことができる。補外公式としては、Richardson の方法がよく知られているが（たとえば文献7参照）、筆者の経験ではε アルゴリズムの方がタフで、しかも効果が大きく、使い易い。ε アルゴリズムの計算式は、図式的に書くと、

$$[右] = [左] + \frac{1}{[下] - [上]}$$

で、最初、表の一番左の列に、分割の粗い順に結果を書き、それらを出発値として、順次右の列へ計算を進めてゆく。簡単な計算例を表12に示す。

表12 ε 補外法による精度改善例

分 割	計算結果		
2×2	・ 752809	— 1808. 318	
4×4	・ 752256	— 6211. 180	・ 752028875
8×8	・ 752095	— 23809. 524	・ 752038176 etc.
16×16	・ 752059		

↑
この列は
中間結果
↑
ここに第1回の
補外値がでる。

文 献

- 1) Schrefler, Lewis, Odorizzi (eds.); "Engineering Software for Microcomputers" (Proc. of 1st International Conference, Venice, 1984), Pineridge Press, 1984
- 2) J. A. Meijerink and H. A. Van der Vorst An iterative method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. of Comp., vol.31, No. 137
- 3) H. Togawa; "C loves you(3)", Information, vol. 3, no. 11, (Nov. 1984)pp. 131-133
- 4) Eisenstat; A note on generalized conjugate gradient method, SIAM, J NA, vol.20, No. 2
- 5) 村田健郎; CR 法と M 行列について。1984年9月、数理解析研究所、研究会配布資料
- 6) 戸川隼人; 「BASIC による有限要素法の基礎」, サイエンス社, 1985
- 7) Holland and Bell 著, 川井忠彦監訳; 「有限要素法, 応力解析への応用」, 朝倉書店