

電卓シリーズ（6）

地質学向きのやさしい会話型プログラミング言語ベーシック

小 出 仁（環境地質部）

1. 地質学とコンピューター

人が一生かかってもできないような計算をコンピューターはわずか数秒でやってしまう。コンピューターは人間の可能性を高めたことは確かだがそのプログラミングは慣れた人にもかなり根気のいる面倒な仕事である。

とかく数学アレルギーの地質屋が敬遠するのも無理はない。しかし最近のコンピューターの発達によって数学オンチやコンピューターぎらいでも容易にマスターできしかも地質学に特に有用なプログラミング法ができてきたことを紹介したい。

コンピューターにとっても地質学はもっとも苦手な分野の一つであろう。地質学とは地球における岩石・鉱物や資源・火山等々の空間的・時間的な分布のパターンを研究する科学であって情報科学でいうパターン認識を必要とする。比較的単純と思われそうな岩石の鑑定でも色や形や硬さ等の様々の要素を組み合わせて総合的に判断する必要がある。ただひとつの要素だけでは判別できないしまた岩石中の一点だけをどのような高度な技術を使って分析してみてもいわゆる「群盲象をなでる」のとえ通りになる。このような総合的判断となると経験を積んだ人間ならごく簡単なことでもコンピューターにとって極めて難しいことになる。

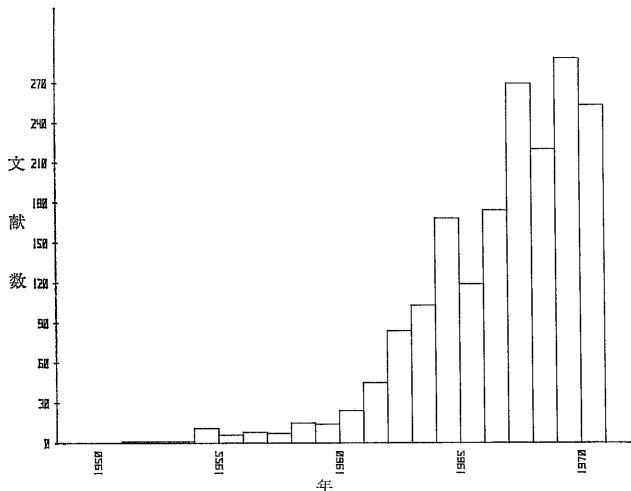
近代的なコンピューターの原理は1812年頃からのC. BABBAGE（イギリス）の研究に始まるといわれる。近

代地質学も同じイギリスでほぼ同じ頃に黎明期を迎えたがその発展期は19世紀で現在は停滞みである。他方いわゆる電子計算機は1940年代にやっと実際に作られるようになり一般に利用されるようになったのは1951年にUNIBACで商用コンピューターが作られ（日本では1958年NEAC 2201が最初）てからである。それからのコンピューターおよび情報科学の爆発的發展は御承知のとおりである。発達時期の違いやパターン認識処理のむずかしさのため地質学のコンピューター化はあまり進んでいない。それでも第1図に見られるように主にアメリカでのデータであるがコンピューターを利用した地質学関係の論文数は急速に増えている。第二世代のコンピューターが普及しはじめた1962年頃から特に増加が著しい。MERRIAM（1975）によれば1950年代は統計のためにコンピューターが利用され1960年代には分類・対比・傾向解析1970年代にはシミュレーションとしたいに利用法が高度化してきている。

地質学の中心的課題にももっとコンピューターが利用されるためにはパターン認識の処理技術が進歩する必要がある。資源産業の比重が高いアメリカでは地球科学がコンピューター・グラフィックスの進歩を促す要因になっている。物理探査（小川・津 本誌217号）やリモートセンシング（竹内他 本誌266号 山本他 本誌269号）で特にコンピューター・グラフィックスが盛んに利用されている。特にリモートセンシングはパターン認識そのものである。しかしコンピューターによるパターン認識の処理にはまだ困難な点が多い。それで人とコンピューターが対話して長所を生かし短所を補う必要がある。地質学関係のコンピューター利用はコンピューターのみによるより人とコンピューターの対話を行う方がよい結果が得られることが多い。

2. コンピューターと会話する

コンピューターはどんどん大型化・高速化されてきたが大型化されるとかえって不便になる点もある。従来のコンピューター利用方式は通常バッチ処理—計算センター方式といっ



第1図 コンピューターを利用した地質学関係文献数の増加。

て 計算センターないし中央計算室に 準備された計算プログラムを持ち込み 順番を待って コンピューターが空いた時に計算してもらう。

コンピューターが計算している間は人は関与できない。コンピューターが計算結果を出すと それを自分の研究室に持ち帰り もし満足する結果が得られていなければプログラムか入力データを直して また始めから計算し直す。バッチ処理は大計算には能率がよいが 自分の計算は数秒ですむ場合にも もし前に長い計算をコンピューターにやらせている人がいれば それが終わるまで待たなければならない。コンピューターが大型になったために 集中化するので 運搬時間も長くなる。そうすると 短い計算でも 実際は数時間 時には数日も待たないと計算結果が分らないことがある。つまり バッチ処理方式は 大計算には能率がよくても 小計算には能率が悪い。小計算をひんぱんに行なわなくてはならない プログラム開発や会話型の計算に バッチ処理は向いていない。

このような大型コンピューターの欠点を補うために考えられたのが タイム・シェアリング・システム (時分割方式 TSS) である。TSS では 中央の大型コンピューターにつながった端末装置が いくつも各所に分散配置してあり 利用者は直接コンピューターのある計算室に行かなくても 最寄の端末からコンピューターを使うことができる。しかも 実は他の端末にも他の利用者がいるのだが あたかも自分1人で使っているかのように中央のコンピューターを使うことができるという便利な方式である。実はコンピューターに交通整理の役目をするプログラムが入っていて 短い計算は優先的に処理してくれるので 短い計算ならすぐに答がでてくるわけである。だから会話型の計算に便利で 特にグラフィック・ディスプレイ端末装置は 図形を扱うことが多い地質学に有効である (津 本誌 265 号)。

もう一つの解決策は 大型化と逆行して 小型・コンパクト化することである。つまり最近はやりのオフィス・コンピューターやマイクロ・コンピューター それからこのシリーズで使用例を解説している 卓上計算機がそれである。技術進歩により 小型で安価で 素人にも容易に保守・操作ができ しかも相当の性能を備えたコンピューターを作れるようになった事が 流行の原因であろう。このような超小型コンピューターは容量が小さく 計算速度も遅いが 手近に置いて 気軽に使えるので 日常に数多く発生する小計算には かつて能率がよい。また 相当時間独占して使うこともできる場合が多いので ゆっくりとコンピューターと対話し

ながら計算することもできる。バッチ処理—計算センター方式を中央集権型とすれば TSS方式や超小型のコンピューターは地方分権型にたとえられる。あるいはコンピューター中心型と人間中心型に対比することもできる。

当り前のことだが 人間がプログラムを作って 指示しないと コンピューターは動かない。コンピューターは指示されたこと以上の事は絶対やらない。コンピューターで計算したというと 盲目的に信用する人と逆に拒絶反応を示す人がまだ多い。しかし コンピューターといえども ただの道具にすぎないので 結果が正しいか否かは 人間の作ったプログラムの責任である。コンピューターそのもの (ハードウェア) より その利用技術 (ソフトウェア) の方が大事であって コンピューターを生かすも殺すもソフトウェアしだいである。独占できるとはいっても 従来のミニ・コンによくあったように プログラムの改変も簡単にはできないようなシステムでは コチコチの石頭の人と議論しているようなもので 対話にはならない。コンピューターと対話するには 対話を可能にするような柔軟性に富むソフトウェアが必要である。

3. 会話型プログラミング言語ベーシック

日本で科学技術計算用にもっとも多く使われているプログラミング言語はフォートラン (FORTRAN) である。

しかし フォートランは元来バッチ処理用に作られた言語で会話型ではない。会話型の計算では 計算結果を見て プログラムやデータを臨機応変に変えて 徐々に目的に近づいて行く。だから プログラムの改変・選択・データの途中入力が簡単に行なえなければならない。ところが フォートランはプログラムの改変が困難で 計算途中にデータを入力したり 人が計算途中に関与することはできない。そのため会話型用に作られたプログラミング言語がベーシック (BASIC) である。

ベーシックはタイム・シェアリング・システム (TSS) でテレタイプライター等の端末装置を通して コンピューターと直接対話することを目的に作られた。そのため対話が容易にできるように 数式や英語に近い簡単な言語 (命令 コマンド) が使われている。BASIC は Beginner's All-purpose Symbolic Instruction Code の略で 名前のように簡単な 解りやすいプログラミング言語である。フォートランをマスターするには6ヶ月ぐらいいかかるが ベーシックなら2日で使えるようになるといわれる。

ベーシックはフォートランを簡易化したような言語で また最近よく使われるようになった高度なプログラミング

グ言語 ビー・エル・ワン (PL/I 科学技術計算用のフォートランやアルゴル [ALGOL] と事務計算用のコボル [COBOL] をあわせた機能をもつ) にも似ている。それで フォートランや PL/I の入門用としてベーシックを学ぶこともできる。しかし アメリカでは 入門用に限らず科学技術計算にフォートランよりベーシックの方が多く用いられている。

このシリーズでは前回まで横河ヒューレット・パッカー・カード・モデル20型 (YHP-20) という卓上計算機を用いた計算例を紹介してきた。YHP-20 も会話型で使用されるが 専用のプログラミング言語を使わなければならないので 性格としてはプログラミング可能の大型電卓である。ただし プログラミングの基本的考え方は多くの電卓やマイクロ・コンピューターと共通なので 基本的なプログラミングの流れを紹介してきた。今回は横河ヒューレット・パッカー・カード・モデル30型パーソナル・コンピューター (YHP-30) を使用する。YHP-30は YHP-20 の上級機種で 多くの周辺機器が共通に使える。しかし YHP-30 では汎用のプログラミング言語ベーシックが使える。電卓とオフィス・コンピューターの中間的性格を持っている。したがって YHP-20 と YHP-30 はかなり性格の違う計算機で 当所内における所属も使用目的もまったく異なる。今後このシリーズでも目的と計算の性格によって 両機を使い分ける。YHP-20 は既にかなり旧式にはなったが 比較的単純な電卓的な使用法に秀れた機能を持ち YHP-30 はより柔軟性が高いので やや複雑な図形処理等に秀れている。

同じベーシックといってもコンピューターにより多少の相異があるが YHP-30 を例として フォートランと比較しながら 会話型ベーシックの利点を説明する。

4. 会話型プログラミング

ベーシックのよさを理解するには 簡単なプログラムを作ってみるとよい。まず 平方根を計算し 印刷させてみよう。Xの平方根を求めるには ベーシックでは $SQR(X)$ と書く。フォートランでは $SQRT(X)$ と書くので ベーシックの方が1字だけ短くなっている。結果を印刷するには ベーシックでは：

```
10 PRINT SQR(X)
```

とタイプして キー・インする。フォートランはパッチ処理なので 命令文はパンチカードに穿孔しておいてから そのカードをコンピューターに読み込ませるが 会話型ではコンピューターのキーボードから直接キー・インするのが原則である。紙テープ等とは違い パン

チカードは順序をさし変えたり 1枚のカードだけを打ち直したりできるので プログラミングにはなくてはならぬものであった。しかし 少しプログラムが長くなると カード枚数はたちまち増え かさばって重くなる。こうなると カード1枚でもなくなると 簡単には復旧できない。パンチカードが不要になっただけでも取扱いは相当楽になる。

仮りに 先のベーシックのプログラムをしている時に誤って：

```
10 PRINT SQT(X)
```

とタイプしてしまったとする。気づかずにこのままキー・インすると

```
ERROR 8 IN LINE 10
```

とコンピューターの表示部に間違いが指摘される。スペルを間違えたのでコンピューターに理解できない命令になったためである。文法上ははっきりした間違いになっていなければ すぐには間違いが指摘されないが 実是不注意によるスペル等の間違いが 数としては多いので これだけでも大助かりである。

これだけではまだ Xの値が分っていないので コンピューターにも計算のしようがない。そこで：

```
5 LET X=3
```

とキー・インするとXの値に3が代入される。ふつうベーシックでは LET を書くが YHP-30 では略してもよいので 以後の例では略す。

フォートランの場合は 命令文はパンチ・カードの順序で実行される。ベーシックでは 文の初めに行番号を必ずつけ 行番号の若い方から順に実行する。この例では 後から入力した行番号5の命令を 前にキー・インした行番号10の命令より先に実行する。ベーシックでは 行番号を正しくつけさえすれば キー・インの順序はどうでもよい。実行順序を変えたい時は行番号を変えればよい。フォートランで実行順序を変えるには パンチ・カードの並び方を変えなければならない。パンチ・カードは枚数が少なければ便利がよいが 枚数が多くなるとカードのさし変えは困難になり さし変えたために かねて大きなエラーを起すことさえある。初めのプログラミングの時にも フォートランではコンピューターの計算順序の通りに パンチ・カードにプログラムを穿孔しなければならないが ベーシックでは人間の思考順序通りにプログラミングを行うことができる。

プログラムの最後に終りを示すために：

```
20 END
```

とキー・インする。

ここで実際にコンピューター内で どのようにプログ

ラムされているかを調べるために リストを打ち出させると：

```
5 LET X=3
10 PRINT SQR(X)
20 END
```

となっており 正しくプログラムされている。このプログラムを実行させると：

```
1.732050808
```

と結果が印刷される。

これではひとつしか計算できないので もっと多くの平方根を計算できるように プログラムを直してみよう：

```
5 READ X
```

とキー・インして 行番号5を入れ換える。そして：

```
2 DATA 6.5, 1000, 0.8, 7, 283.3675
```

と計算したいXの値をコマで区切って入れる。ベーシックでは整数と実数を区別しなくてもよいので このように様々のデータをそのまま並べて 一緒に読み込ませることができる。ただし これだけでは最初のデータしか計算しないので さらに次のようにキー・インする：

```
1 DATA 5
3 READ N
4 FOR I=1 TO N
15 NEXT I
```

これでデータの個数(N)を読み込ませて 行番号4と15の間をN回くりかえして実行させる(FOR—NEXTループと呼ばれ フォートランの DO ループに当る)。Xの数が多くなるので どのXの平方根かを知るためにXの値も一緒に印刷できるように：

```
10 PRINT X, SQR(X)
```

とXを10行目にそう入する。パンチ・カードでは1字をそう入するにも パンチ・カードを打ち直す必要があるが ベーシックでは簡単にそう入できる。ここでリストを出してみると：

```
1 DATA 5
2 DATA 6.5, 1000, 0.8, 7, 283.3675
3 READ N
4 FOR I=1 TO N
5 READ X
10 PRINT X, SQR(X)
15 NEXT I
20 END
```

となっており 5行目まではギッシリとなっているので 新しい行を追加できない。そこで REN とキー・

インすると コンピューターで自動的に10番おきに行番号をつけかえてくれるので：

```
10 DATA 5
20 DATA 6.5, 1000, 0.8, 7, 283.3675
30 READ N
40 FOR I=1 TO N
50 READ X
60 PRINT X, SQR(X)
70 NEXT I
80 END
```

そこで何の表かをはっきりさせるため ラベルをつける：

```
35 PRINT "X SQR(X)"
```

計算を実行させると：

X	SQR(X)
6.5	2.549509757
1000	31.6227766
0.8	0.894427191
7	2.645751311
283.3675	16.8335231

と平方根の表ができる。

フォートランでは データを入力したり 結果を印刷する際に書式(フォーマット)をきちんときめて 指定する必要がある。フォーマットには IBM の会計機時代の名残があるとかで 煩雑でまったく融通がきかない。ベーシックでは これまでの例でもわかるようにフォーマットを特に指定しなくても 入出力できる。フォートランの学習は入出力が完全にできるようになれば 卒業といわれるくらい 入出力がむずかしいが ベーシックはまったくの初心者でもすぐ入出力できる。それだけベーシックは早くコンピューターに親しめるわけである。特に 科学技術計算では 入出力のプログラミングに あまり時間をかけるのは得策でない。

簡単な例だが ベーシックでのプログラミングの容易さは 理解していただけたと思う。ベーシックの利点は 自然な思考過程に従って 試行錯誤を繰り返しながら プログラミングできることである。神様でもなければ 始めから誤りがまったくないプログラムを作ることはいできない。プログラムそのものを書くより その「虫取り」(プログラム中の様々の誤りを「虫」と呼ぶ)の方が手間がかかることが多い。フォートランでは間違いのある個所(つまりプログラム中の虫の居場所)を見付けること自体が困難で また見付けても修正に手間がかかる。ベーシックでは 一部の計算だけを実行させたり 途中でプリント命令を一時的にそう入して

中間結果を出させたりして 虫の居場所を見出すことが ずっと簡単にできる上に 修正やプログラムの入れ換え が 容易なので「虫取り」にかかる時間がずっと少なく てすむ。 フォートランの場合は あらかじめよく計画 を練って 完全な予定を立ててから プログラミングを 始める必要があるが ベーシックの方は 思いいたら すぐにプログラミングを始める方が むしろよい結果が 出る。

5. 会 話 型 演 算

前節の例は プログラミングの過程は会話型であった が 演算自体は会話型ではない。 簡単な基礎統計量の 計算を例にして 会話型演算の話をしたい。

基礎統計量としては まず n 個のデータ x_i の平均が ある。

$$\text{平均 (mean): } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

さらに データの散らばりを示す量として 標準偏差 が用いられる。

$$\text{分散 (Variance): } M_2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2$$

$$\text{標準偏差 (Standard deviation): } \sigma_x = \sqrt{M_2}$$

さらに データの対称性や尖りぐあいを示す量として 対称度や尖度が用いられる。

$$\begin{aligned} \text{3 次 の モーメント: } M_3 &= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3 \\ &= \frac{1}{n} \sum_{i=1}^n x_i^3 - 3\bar{x}M_2 - \bar{x}^3 \end{aligned}$$

$$\text{対称度 (Skewness): } \beta_1 = M_3^2 / M_2^3$$

$$\begin{aligned} \text{4 次 の モーメント: } M_4 &= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4 \\ &= \frac{1}{n} \sum_{i=1}^n x_i^4 - 4\bar{x}M_3 - 6\bar{x}^2M_2 - \bar{x}^4 \end{aligned}$$

$$\text{尖度 (Kurtosis): } \beta_2 = M_4 / M_2^2$$

基礎統計量を計算するプログラムを作ってみよう：

(プログラムは右上)

モデル30型パーソナル・コンピュータには関数キー が20個ついている。 この関数キーは いわばプログラム を入れておく引き出しのようなもので それぞれの関 数キーに異なったプログラムを入れておくと 関数キー を押した時に そのキーに入っているプログラムを実行 してくれる。 このプログラムを使って 基礎統計量を 計算するには まず関数キー F0を押す。 最初のREM

```
10 REM START:F0
20 DIM S(4)
30 N=0
40 MAT S=ZER
50 PRINT
60 DISP "READY"
70 END
```

```
10 REM ENTER DATA:F1
20 PRINT
30 PRINT "DATA"
40 DISP "X=";
50 INPUT X
60 PRINT X
70 N=N+1
80 T=1
90 FOR I=1 TO 4
100 T=T*X
110 S(I)=S(I)+T
120 NEXT I
130 GOTO 40
140 END
```

```
10 REM MEAN AND STANDARD DEVIATION:F2
20 PRINT
30 PRINT "BASIC STATISTICS"
40 PRINT
50 PRINT "N=";N
60 M1=S(1)/N
70 PRINT "MEAN=";M1
80 M2=S(2)/N-M1*M1
90 PRINT "STD. DEV=";SQR(M2)
100 END
```

```
10 REM SKEWNESS AND KURTOSIS:F3
20 M3=S(3)/N-3*M1*M2-M1^3
30 PRINT "SKEWNESS=";M3/M2^3
40 M4=S(4)/N-4*M1*M3-6*M1*M1*M2-M1^4
50 PRINT "KURTOSIS=";M4/M2^2
60 END
```

```
10 REM DELETE DATA:F4
20 PRINT
30 PRINT "DELETED DATA"
40 DISP "DELETE X=";
50 INPUT X
60 PRINT X
70 N=N-1
80 T=1
90 FOR I=1 TO 4
100 T=T*X
110 S(I)=S(I)-T
120 NEXT I
130 GOTO 40
140 END
```

と書いた行は注釈のための行なので コンピューターは 実行しない。 次の DIM で配列 S の要素数を指定して いる。 そして変数 N や配列 S の内容をすべて 0 とおく。 40行目の MAT が始めについた命令で 配列 S の各要素 を 1 命令で 0 にすることができる。 他にも配列や行列 の演算を容易にするような命令がベーシックには準備さ れていて 行列演算が容易なのがベーシックの長所の一 つであるが 今回は詳しく述べない。 PRINT とだけ 書くと プリンターは 1 行空送りする。 最後に DISP でコンピューターの表示部に 準備ずみを示す。

そこで関数キー F1 を押すと プリンターに DATA

という文字が印刷され コンピューターの表示部に $X = ?$ と表示が出る。そこで X の値を直接キー・ボードからタイプで入力する。50行目の INPUT 命令がキーボードからデータを直接入力するための会話型特有の命令である。INPUT 命令があると コンピューターはデータがキー・インされるまで 待っていて 入力されてから 続きを実行する。関数キー F1 では データがキー・インされると N に 1 を加え 配列 S の各要素にそれぞれ x x^2 x^3 x^4 を加える。130 番の行に達すると 40 番の行にもどり 次の x の値が入力されるのを待つ。130 番の行に達すると必ず 40 番の行にもどるから このままでは演算が繰り返されて 終りにならない。フォートランでは このような出口のないループを作ると 重大なエラーになる。つまり バッチ処理では人間が計算途中に介入しないのでいつまでも同じ計算を繰り返してしまう。ところがベーシックの場合は 途中で INPUT 命令があると コンピューターは人間が新しいデータを入力するまでそこで待っている。だから データのある間はキー・インを繰り返して行なうが データがなくなれば STOP キー等を押すことにより ループ計算をやめさせることができる。この方式はデータ数が一定していない時には便利である。

平均値を求めなくなった時は ループ計算の途中に何時でも 関数キー F2 を押すと そこまで入力された x の平均値と標準偏差を計算して印刷する。対称度や

尖度も求めたい時は続けて F3 も押す。さらに多くのデータも前のデータに加えて計算したいときはまた関数キー F1 を押して 入力が続ければよい。また 今まで入力した x の値は全部消して まったく新しいデータの平均値を求めたい場合は F0 に戻って初めから計算し直す。これで一応計算はできるわけだが 実際の計算では 間違えたデータを入力したり 不要なデータがまざったりして 一部のデータだけを 後から消したくなることもある。関数キー F4 は F1 とは逆に入力された x の値等をそれまで入力された値の和から差し引く。そこで F4 を押して 消去したい x の値を入力すればよい。その後 F1 を押せばまた入力を続けることもできる。

このプログラムは 極めて簡単だが フレキシビリティが高く データは無制限に入力したり 追加・削除もできる。ためしに ある泥岩の鉛含有量の化学分析値 (ppm単位) の平均値と標準偏差を求めてみよう。

```
DATA
31
37
31
28
35
18
70
136
18
32
25
22
14
20
32
28
12
21
```

BASIC STATISTICS

```
N= 18
MEAN= 33.88888889
STD. DEV= 27.68250326
SKEWNESS= 7.966340278
KURTOSIS= 10.56316310
```

結果を見ると 標準偏差・対称度・尖度がすべて大きく 分散して しかも歪んだデータだとい

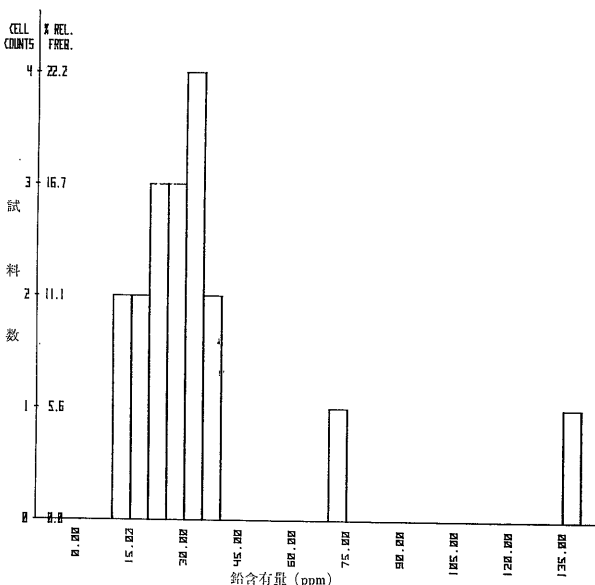
うことがわかる。ヒストグラム (第2図) を作ってみても解るように (ヒストグラムもプロッターで描いたがプロッターの説明は他の回にゆずる) 2つの値が他とかけはなれて大きい。そこで他とかけはなれた2つの値を取り除いてみる。

```
DELETED DATA
136
70
```

BASIC STATISTICS

```
N= 16
MEAN= 25.25
STD. DEV= 7.369701486
SKEWNESS= 0.034063997
KURTOSIS= 1.866319769
```

標準偏差・対称度・尖度共にはるかに小さくなり データのまとまりがよくなったことが分



第2図 ある泥岩の鉛含有量の分析値のヒストグラム。

フ ォ ー ト ラ ン と ベ ー シ ッ ク の 比 較

	フォートラン (FORTRAN) (バッチ処理)	ベーシック (BASIC) (会話型)
学習の難易度	難かしい (特に入出力).	容易 2日くらいでマスターできる.
プログラムの入力方法	予め全プログラムをパンチカードにパンチし そのカードをコンピューターに読み込ませる.	コンピューターのキー・ボードに直接タイプする.
データの入出力	繁雑 (書式を指定する).	簡単 (書式を指定しなくてもよい).
プログラム・データの保存	パンチカード・紙テープ・磁気テープ・ディスク.	左に同じだが カセット磁気テープ・フロッピーが多い.
変数名	6個以内の英字・数字の組み合わせ.	英字1字と数字1字の組み合わせ.
整数と実数	区別する.	区別しない.
エラーの探知	コンピューターに実行させないとエラーリストが出ない. リストに出ないエラーの探知はきわめて難かしい.	文法上のエラーの多くは キー・インすると直ちにエラー・メッセージが出る. 部分計算や追跡によるエラー個所の探知が可能.
プログラムの修正	1字のみの変更でもカード1枚を打ち直す. 大幅な変更はカードの入れ換えが面倒.	容易に挿入・変更・削除ができる.
人間の判断	演算途中に入れられない.	演算途中に入れられる.
副プログラム	主プログラムと独立して使いやすい.	簡単だが主プログラムと干渉しやすい.
行列演算	繁雑	簡単で高度の機能を持つ.
計算能率	よい	やや悪い

る。 2つの値がどうして他とかけはなれていたかは地質学的な解釈が必要であって 数値解析だけでは分らない。ここに 地質学者自身が自分でコンピューターを操作して解析する意義がある。このような簡単な例は データを注意深く見れば気付くが もっと複雑な問題では 地質学的な解析とコンピューター解析をそれぞれの専門家が 互いの理解もなく 別にやっているは気付かないような問題がたくさんある。だが 地質学者はコンピューターの専門家ではないので コンピューターの学習のために長い時間をさくことはできない。したがって プログラミングし易さや コンピューターの保守・操作の容易さが重要になる。

6. おわりに

会話型プログラミング言語ベーシックの長所をお話ししてきた。しかし もちろん欠点もある。一般にベーシックは計算能率は悪い。計算能率は プログラミングの能率と裏腹の関係にある。結局は 人間がプログラミングにさく時間と コンピューターの使用時間のどちらを重視するかである。計算能率のためならばアセンブラを使えばよいが 私達のようなプログラミングのアマチュアが むずかしくて汎用性のないアセンブラを学ぶ必要はないと思う。

他の欠点としては ベーシックでは変数名を英字1文字と数字1文字の組み合わせだけで表わすので 何のために使った変数名かを忘れてたり 不用意に二重の意味に使ったりしやすいことである。また 副プログラム中の変数名が 主プログラム中の変数名と独立していないの

で 気を付けないと思わぬ間違いが出る。これらの欠点は主に長いプログラムの時に問題になるもので ベーシックはやはり小計算やテスト計算に向いているといえる。しかし 最近のベーシック機には フォートランの長所を取り入れて ベーシックの欠点を取り除いたものも出てきている。

ベーシックはきわめて柔軟性に富み「虫取り」も容易なので 初心者にも学びやすい秀れたプログラミング言語である。フォートランとベーシックの比較を要約すると上の表のようになる。しかし 単に フォートランやPL/Iの入門用というだけでなく 高度な計算にも使われている。ベーシックはバッチ処理でも使えるが長所の多くは会話型で使わないと生かされない。会話型の最大の利点は 人間の判断が計算途中で加えられることである。この利点は しばしば描いてみなければ効果がよくわからない図形処理に特に有効である。図形の取り扱いについては 次回以後にお話ししたい。

ベーシックを学びたい方には それぞれのコンピューターのマニュアルの他に 入門書が多数出版されている。例えば 考案者のダートマス大学の J. C. KEMMENY と T. E. KURTZ による本がある。また (7)は統計解析とベーシック双方の入門書として使える。この小文ではベーシックのごく一部しかお話できないので 興味をお持ちの方はこれらの入門書を参照されたい。現在 APL 会話型フォートラン等他の会話型プログラミング言語も色々考案されているが おそらくベーシックは 多少拡張されながら 今後盛んに利用されると思われる。

参 考 文 献

(1) MERRIAM, D. F. (1975) : Computer perspectives in geology, in McCammon, R. B. ed., Concepts in Geostatistics, Springer Verlag, N. Y., 138~149
 (2) 小川克郎・津宏治 (1972) : 地質分野における電算機利用の現状 地質ニュース 217号 1~12
 (3) 竹内章司・上原勝徳・松野久也 (1976) : リモートセンシングデータのデジタル処理 地質ニュース 266号 1~8

(4) 山本利幸・道野敏雄・松野久也 (1977) : リモートセンシングデータの総合処理システム 地質ニュース 269号 1~11
 (5) 津宏治 (1976) : グラフィックディスプレイ装置と物理探査におけるその利用 地質ニュース 265号 20~29
 (6) KEMMENY, J. C. and KURTZ, T. E. (1971) : ベーシック入門 森口繁一監訳 尾崎義雄・神山武共訳 共立出版
 (7) 石田正次 (1977) : データ解析の基礎 現代統計数理シリーズ 森北出版

〔10頁からつづく〕

研究計画の概要:

1. 広島県北西部の中生代火成活動の研究

(四国出張所)

研究方針: 四国地方における非金属鉱床の開発調査研究及び四国内帯の火成岩類の岩石学的研究を実施するとともに 管内の地質及び地下資源に関する資料の収集・整備を行う。

研究計画の概要:

1. 香川県三豊郡下粘土鉱床調査
2. 香川県下の岩脈類の岩石学的研究
3. 高知県下の天然ガス徴候地調査

(九州出張所)

研究方針: 九州地方の地質的特性に応じた調査研究を行うとともに 管内の地質及び地下資源に関する資料の収集整備に努め もって地方の地質関係の基礎資料を提供する。

研究計画の概要:

〔63頁からつづく〕

他多くの数学者・哲学者など さらに芸術にスポーツにわれわれに強い影響力を持っていて 極めて文化的イメージが強い。これに反しラテン文化は 勿論すぐれたものがあるのだが 「ローマは一日にして成らず」のたとえとともに 強者のローマ 侵略支配のローマといったイメージが強いのは どういうことであろうか。

しかしやはりギリシア・ローマを中心とする地中海文明は きわめて香り高いもので 現代のすべての面において われわれに大きな影響を与えているのである。

しかしながらその後キリスト教護教と宗教支配のもと ガリレオなど 迫害に耐えて真理探求に生涯を捧げた人

1. 天草炭田の総括研究
2. 南部九州の金鉱床の研究
3. 佐世保層群の研究
4. 九州のけい砂鉱床の研究
5. 対州層群の水理地質学的研究

30. <資料業務>

業務方針: 当所における資料情報業務は 国内外の地球科学に関するセンター的性格が要求されている。これに対処し 特に 国際的情報交換システムの確立を積極的に図るとともに 筑波体制におけるデータベースとしての効果的運用のための整備に努める。また 編集出版業務については能率化に努める。

業務計画の概要:

1. 資料情報の調査収集
 - (1) 国内外の地球科学情報の調査を積極的に行い その網羅的収集に努める。
 - (2) 出先関係機関との連携を密にし 地方文献の収集を行う。
2. 資料情報活動の強化
 - (1) 収集資料の積極的活用のため 地学文献速報の発行体制を確立する。
 - (2) 地質文献目録 文献情報検索の機械化移行に備える。
3. 資料情報検索の機械化
4. 受入資料の整理
5. 編集出版業務の能率化
6. その他

々とのいわゆる「科学と宗教」という大命題のゆえに地中海文明はやがて西ヨーロッパに移って行くことになる。この抑圧は 生物進化にあてはめればプロトゲネーゼに当り 以後の花開く大発展をとげたチポケネーゼのエネルギーとなったといえないこともない。

それにしても ギリシア・ラテンの文化的遺産が 今もってわれわれに大きな影響力を与えている事実に感銘をうけながら 今さらながらその偉大さに 思いをいたす次第である。

参 考 文 献

松平千秋・国原吉之助共著: 新ラテン文法 南江堂刊 1977
 吳茂一・泉木吉著: ラテン語小文典 岩波書店刊 1977