

数学教育において用いられるコンピュータ言語に 関する基礎的研究

白石和夫

A Fundamental Study on Programing Languages Used for Mathematics Education

Kazuo Shiraishi

要約

数学教育において用いられるプログラミング言語の統一を図るには、ISO Full BASIC 規格に基づくのが適切である。その際、数学教育独自の拡張が望ましい部分も存在する。

1. 数学教育においてコンピュータが 果たすべき役割

数学教育は、現在、多くの困難と解決しなければならぬ課題を抱えている。教育内容を将来を担う生徒にふさわしいものに変えていくこともその一つである。教育方法の観点では、生徒自身による主体的な活動や望ましい数学観の育成が要求されている。このような現状において、数学教育のなかでコンピュータが果たすべき役割は多様である。コンピュータそのものの理解が新時代の数学教育の内容として不可欠なものである一方、コンピュータを数学の学習に役立てることが生徒の主体性を回復するために役立つものと考えられるからである。

機械的な計算はすべてコンピュータが処理してくれる時代の数学教育においては、事象

を考察するなかから規則性を見いだして機械的に処理する方法を作る出す能力を身につけられることが重要である。コンピュータによって処理できるようにすることを目指して理論展開を行うことが、望ましい数学観を育成する見地から今後重視されるようになるであろう。

論理性、創造性の育成の見地からもコンピュータを教材として用いることのよさを考える必要がある。今回の指導要領改訂では、思考力育成を意図して高等学校に平面幾何を導入されているが、平面幾何の証明よりも、はるかに具体的に明確な形で論理の訓練を行うことができるのがコンピュータのプログラミングである。平面幾何の証明では何が自明のことであるかということに関しては必ずしも明確ではないが、コンピュータの場合、その動作は明確に示され、正しいプログラムで

あるかどうかは生徒自身の手によって簡単に検証できる。さらに、近年の洗練されたプログラミング言語は、数学文よりはるかに明確に論理的に整った形で処理手順を表現することができるものである。

ところで、現在の工業技術の進歩の速さを考えるとき、10年後の数学教育においては、現在のノート型パソコン程度の能力を持ったものを生徒個人に持たせておくことが夢ではなくなるであろう。そうなると、コンピュータを教師のための道具から生徒のための道具として質的に転換させることが可能となる。生徒がコンピュータを自身の手足のように利用することができるようになったときに生徒の主体性を活かした数学教育が可能となるから、この時期においては、コンピュータ・リテラシーの修得は数学教育において必須の内容として位置づけられるものとなるであろう。

2. 数学教育におけるコンピュータ教育の現状

コンピュータに関する指導は今回の学習指導要領改訂の重点であったはずであるが、その具体的内容および方法については十分な展望がなく始められたように思われる。N88-BASIC や CEC-BASIC に代表されるマイクロソフト系 BASIC が数学教育に不向きであることはすでによく知られたことであるが、数学教育関係者にはその事実がよく知られていないようである。あるいは、知られていたとしても無視されているのが現状である。た

とえば、文部省自身の手によって今春作成された資料（文献 [1]）においても問題のある BASIC が無批判のまま用いられており、思考力を鍛え、創造性を育成しようという本来の目標が達成できるかどうか危ぶまれるのである。

コンピュータ教育は、自分の考えでプログラムを作ってそれがうまく動くという成功体験を通して、自分の頭で考えることに対する自信を持たせる数少ないチャンスを与えてくれるものである。しかしながら、一応、論理的には正しいと考えられるようなプログラムがうまく動かないとしたら、このような教育は生徒によくない心理的影響を与えるのではないだろうか。以下、具体例を示して CEC-BASIC の問題点を簡略に述べる。（文献 [2] [3] 参照）

配列の用い方を学んだばかりの生徒が「40人分の答案の得点を10点刻みで分布を調べるプログラムを作れ」という課題に答える場合、INT 関数の利用に思い至らなければ、おそらく図1のようなプログラムを書くであろう。このプログラムは CEC 仕様では文法的に正しいものであるが、このプログラムが作成された意図とは異なる意味に解釈される。その原因は、関係演算の結果が数値となることである。数学の授業のなかでそのようにある特定の処理系に特有な処理の仕方にまで言及することは適切なことではないのであるが、その事実を知らなければ安心してプログラムを作ることができないのである。もし、この事

図 1

```
10 FOR N=1 TO 40
20 INPUT X
30 FOR I=0 TO 10
40 IF I*10<=X<I*10+10 THEN LET M(I)=M(I)+1
50 NEXT I
60 NEXT N
```

実に目をつぶることにしたら、生徒にプログラムを作らせるような授業は行えないであろう。

それ以外にも、関係演算の結果が数値となることはさまざまな問題を引き起こす。図2に示す文はいずれも CEC-BASIC の正しいステートメントであるが、誤って異なる意味を持たせてプログラムを書く可能性が予測されるものである。

図 2

```
IF A=B=2 THEN GOTO 10
LET A=B=2
PRINT X^2+1=0
```

マイクロソフト系 BASIC 特有の文法であるマルチ・ステートメントの害についてもよく知られるところであるが、文献 [1] では、マルチ・ステートメントを用いずに書けるところにまでそれを用いて、その使用を勧めるかのようなものである。文献 [1] では、コロン「:」を使用することによって幾つかの文を一行に記述できる。マルチ・ステートメントを用いることによってプログラムの行を少なくしたり、まとまった処理を1行にわかりやすくまとめることができるが、多用するとプログラムの流れがたどりにくくなることがある。」

と述べているが、その程度の教え方がなされたのでは、図3のようなプログラムを書いて

図 3

```
10 IF D=0 THEN PRINT "SUNDAY" : IF D=1 THEN PRINT "MONDAY"
20 IF D=2 THEN PRINT "TUESDAY" : IF D=3 THEN PRINT "WEDNESDAY"
30 IF D=4 THEN PRINT "THURSDAY": IF D=5 THEN PRINT "FRIDAY"
40 IF D=6 THEN PRINT "SATURDAY"
```

図 4

```
10 IF A>0 THEN IF B>0 THEN LET X=1 ELSE LET X=2
```

悩む生徒が出現したとしても不思議ではない。

意味のあいまいな if 文ができてしまうこともマイクロソフト系 BASIC の特徴である。たとえば、図4は、文献 [3] の文法に従えば2通りの方法で生成することができて意味が1通りに定まらないものの例である。

CEC-BASIC を道具的に用いようとするとき最も困るのは関数のグラフや、曲線を描こうとするときである。グラフィックスを使い始めるためにお呪いのような命令を書かなければならないのもよくないけれども、実際にいちばん困ることは、数学における普通の座標系は y 軸の正の向きを上向きにとるのに、CEC 規格では明示されてはいないものの y 軸の正の向きが下向きであることが暗黙のうちに仮定されていることである。そのために、本質に関係しない部分で多大な精神的負担が要求され、道具として使用することに困難をきたしている。また、関数のグラフや曲線をつなげた折れ線で描こうとすると、line 文に何も実行しないうちから最終参照点が存在していることも煩わしいものである。さらに、マウスの利用もたいへん不便である。

3. 数学教育で用いるコンピュータ言語の統一へ向けて

数学教育において用いられるコンピュータ言語が不統一であることはさまざまな点で不便であり、統一されることが望ましい。しかし、すでに述べたように CEC-BASIC を数

学教育に用いることには多くの問題がある。

教育用の言語に対する意見のなかには基本 BASIC の使用をいうものがある。しかし、JIS 基本 BASIC の規格はプログラミングの基本を学ばせることを目的に作成されたものではなく、英語名の「minimal BASIC」からわかるように、最低限、プログラミングが可能な技術的内容のみを抽出して規格にしたものである。基本 BASIC の特徴は「IF 比較式 THEN 行番号」と「GOTO 行番号」とで制御を行うことである。ある条件が成立するときには何かを行わせたい場合には、その条件の否定が成立したときに処理したいことを飛び越すようなプログラムを書かなければならないから、実際のプログラミングには精神的に高度なものが要求される。この規格にしたがって学習するのが適すると思われるのは、大小判断の結果による分岐が可能であればすべてのアルゴリズムが表現できることを体得させることを目的としたい場合のみであろう。これはかなり技術的な内容であり、たとえば NAND 回路があればすべての論理回路が作成できるというのと同様、一般人の知識として要求される内容ではない。

Pascal は多くの人に支持され用いられている言語であるが、数学教育におけるコンピュータの利用が道具としての利用を指向していることを考えると、やはり、Pascal には弱点があることがわかる。また、プログラミング教育の面だけみても、初級レベルでは、後に述べる Full BASIC のほうがすぐれている。Pascal の強みは構造型やポインタであるが、そこまで数学教育において要求されるとは思えない。

C は現在の流行であるが、初心者が気軽に使えるものではない。たとえば、CEC-BASIC のように関係演算の結果が数値になるなど、専門家向きの文法を採用している。

JIS 基本 BASIC は、将来拡張された規格が制定されることを予告している。1991年、国

際標準化機構 (ISO) は、Full BASIC という名称で BASIC の完成された形の規格を制定した。その内容は文献 [4] で JIS 原案の形で翻訳出版されている。

Full BASIC は、現在規格化されている言語のなかでは数学教育で扱うのに最も適したものである。数値に型の別を設けず、代入文には機能語 LET を前置して等号が本来の意味とは異なる意味に用いられていることを常に意識させようとするなどの BASIC 本来のよさを守っている。さらに、構造化プログラミングの考え方を取り入れて行番号の束縛から離れてわかりやすくプログラムを書くことができるようになったことが大きな進化である。規格を詳細に検討すると、CEC-BASIC で問題になったような意味のわかりにくいプログラムができてしまわないように細心の注意を払って設計された様子をうかがうことができる。そればかりか、1行1文の原則が守られているために、プログラムの修正を繰り返してもプログラムの構造が不明確になってしまうことがない利点をも備えている。この点で、構造化プログラミング言語として先輩格の Pascal よりも優れている。

数値演算に十進浮動小数演算を取り入れ、しかも計算精度を厳格に規定したことで、計算誤差のために意図に反する動作をする心配はかなり減少している。組み込み関数も豊富に用意されているので数学自体の学習の進度に応じて適切に利用することができる。たとえば、10を底とする対数関数も追加されているから、自然対数を知らなくても対数の利用が可能である。さらに、三角関数の単位をラジアンと度のいずれかに自由に指定することができて、逆余弦、逆正弦の関数も用意されているから、弧度法や逆三角関数相互の関係を知らなくても正弦定理や余弦定理を用いて角度を求めるような問題に答えるプログラムを簡単に書くことができる。

Full BASIC の最大の利点は、使いやすい

グラフィックス機能が規格が盛り込まれたことである。たとえば、ほぼ理想に近い手順で関数のグラフや媒介変数表示の曲線を描くことができる。また、点 (x, y) の偏角を求める関数 $\text{ANGLE}(x, y)$ が追加されているから、簡単なものであれば複素数による写像を扱うことも可能である。マウスを用いて座標を入力することが簡単にできるから、曲線を描いてその性質を調べるなどの道具として使うのにも便利である。たとえば、2点からの距離の和が一定である点の軌跡を調べたければ、`get point` 文を実行するたびに2点からの距離の和を計算するようなプログラムを書き、さらに、あらかじめ定められた定数より大きい小さいかによって異なる色の点を打つようにしておけばよい。このプログラムを実行すると、求める軌跡の輪郭が次第に浮かび上がってくるものである。また、複素数による変換なども、マウスで画面をなぞると対応する点を別の色で表示するようにしておけばその性質をかなり具体的に把握することができる。一方、マウスを用いて3点を指定するとその3点を通るような円を描くプログラムを作ることなどは座標幾何の適切な練習問題になる。さらに、行列によって変換を施して図形を描く機能があり、これは図形を描くための副プログラムの一種（“絵”という）を作成しておけばそれにアフィン変換を施して描いてくれるものである。実際に一次変換を目で見て確認するプログラムが簡単に作れるばかりでなく、基本的な写像を合成して目的の変換を得る方法を考えさせることができるなど、今後の数学教育を考えるうえで示唆に富んだ内容を持っている。

現在、日本では工業技術院において JIS 制定のための作業が行われている。数学教育関係者全体の要望として意見の集約ができれば理想であるが、すでに調整の最終段階を迎えているようであり、これから準備を始めたのでは、JIS に我々の意向を反映させることは

むずかしいものと思われる。JIS が公表された時点で、JIS に反しない範囲で数学教育独自の拡張のための規格を発表するのが現実的な対応であろう。

4. 数学教育用プログラム言語整備のための指針

本稿では、文献 [4] に基づいて Full BASIC 規格の検討を行い、数学教育で用いられるコンピュータ言語として整備する際の指針を与えたいと思う。なお、Kemeny と Kurtz の「True BASIC」は、ほぼ Full BASIC に近い内容をもっている。大きな相違点は、True BASIC では数値を2進浮動小数として扱うことである ([5] [6] 参照)。そして、数値の扱いを除外して考えれば、True BASIC は数学教育用として十分な機能を持っているといつてよい。

Full BASIC 規格はいくつかの機能単位から構成されているが、そのうち数学教育で必要になるのは、中核機能単位と図形機能単位である。なお、図形機能単位には中核機能単位が含まれているので、単に図形機能単位が必要であるといつてもよい。ISO Full BASIC は米国規格と欧州規格の双方を含むように作られているのでかなり複雑な構造をもっている。欧州規格の中核は部分集合中核と呼ばれるが、これは機能語の大部分を予約語にしたもので扱いに不便である。また、欧州規格の図形機能は簡易図形出力と呼ばれるがこれも数学で用いるのに必要な機能を含んでいない。したがって、数学教育では米国系の規格である図形機能単位をもとに考えるべきだという結論に至る。なお、編集機能単位は、現在の技術水準では時代遅れのものであり、無視してよいであろう。

中核機能単位にも図形機能単位にも数学教育の目的には不要なものもかなり含まれているからこれらを整理し、逆に数学のために必要な機能を強化して数学教育独自の規格を作

成するのがよいであろう。余分なものを整理するのは、要求水準を引き下げることによって処理系の作成を容易にし処理系作成者の負担を軽減するのが目的である。

Full BASIC 規格では、計算精度はかなり厳格に規定されている。演算方式は厳密に言えば十進演算を要求していないけれども、あらかじめ定めた桁数の十進小数で数学的な結果が正しく表現できるものについては正確な値が得られることを要求しているので、事実上、十進演算とみなしてよい内容になっている。また、演算精度も10桁以上を要求しているので、整数を扱う場合にもかなりのことが実行可能である。たとえば、フェルマー数 $F_n = 2^{2^n} + 1$ の因数分解などは $n = 5$ まで規格の範囲内で扱える。

しかし、近似計算の理論を学習する場合には、理論に伴う誤差であるのか、計算機に起因する誤差であるのか明確にしておく必要があり、また、消去法による連立方程式の解法のように、わずかな誤差がプログラムの論理的な流れを変えてしまうことがあるから、十進小数だけでなく一般の有理数を正確に扱えることが要求される。規格の強化を要求する第一の点は、有理数の正確な取り扱いができるようにすることである。その場合、平方根や超越関数の演算結果も常に有理数で近似を行うことにしておくのか、それとも無理関数の結果は浮動小数にするのがよいのかという問題は、処理系自体のつくりやすさと計算速度に関係してくるから簡単には結論が得られそうもない。後者を選択した場合に組み込み関数 MAXNUM および EPS(x) の扱いに問題が生じることが予想されるので、この2つの関数については規格を充たすことを要求しないものとしておいたほうがよいであろう。

その他で規格からの拡張を要する点は、おおむね True BASIC で実現されていることである。

Full BASIC では、行番号分岐の制御文を

まったく用いずにプログラムを書くことが可能であるが、規格上はすべての行に行番号を書くことが要求されている。この条件をゆるめて行番号のないプログラムの編集と実行が可能であることを要求すべきである。この条件は、行番号にわずらわされずにプログラムの修正ができるようにするために不可欠なものである。当然のことながら、プログラムの編集機能は、True BASIC ほか最近の主要な言語ソフトの水準に達したものであることが要求される。

規格を強化すべき点の残りはグラフィックスの利用に関連して起こる問題であり、主として規格の記述が明確でない点を補強するためのものである。たとえば、 $y = \sin x$ のグラフを描くプログラムは図5のようなものになるはずである。

図5

```
10 declare external picture grid
20 set window -4,4,-4,4
30 draw grid
40 for x=-4 to 4 step 0.01
50   plot lines: x, sin(x);
60 next x
70 end

100 external picture grid
110   ask window l,r,b,t
120   set line style 3 ! 点線
130   for x=ceil(l) to int(r)
140     plot lines: x,b;x,t
150   next x
160   for y=ceil(b) to int(t)
170     plot lines: l,y;r,y
180   next y
190   set line style 1 ! 実線
200   plot lines: l,0;r,0
210   plot lines: 0,b;0,t
220 end picture
```

まず、100行以下の絵定義を書かなくても済むようにしたい。それには、システムが軸やグリッドを描くための組み込みの絵定義を持つことが望ましく、また、ユーザー自身もそれを登録できるような機構を持つことである。そして、できれば、10行の宣言文も書かなくて済むようにしたいものである。

上のプログラムでは折れ線を用いてグラフを描いているけれども、数学での利用においては実際に計算した点のみを打って描画したいことが多い。その場合、Full BASICでは、機能語 POINT を持つ幾何図形文を用いるべきであるのか、それとも、機能語 PLOT LINES をもつ幾何図形文を用いるべきであるのか判断しかねる。というのは、機能語 PLOT LINES をもつ幾何図形文の動作は描点の移動の概念で規定されているが、描点がオンになった後、移動しないでオフになったときの動作については明確な規定がなく、一方、機能語 POINT をもつ幾何図形文を用いる場合には、点の形の初期値が星印(*)になっていることと、点の形を黒点(・)に変更してもそれが画面上の1ピクセルに対応する形であるのか不明なためである。数学用の規格としてはこの点を明確にしておく必要がある。たとえば、描点がオンになったときには必ず点が描かれることを規定しておけば、50行の末尾のセミコロンを取り去るだけで目的を達成することができる。ただし、この場合、機能語 PLOT LINES を持つ幾何図形文を点を描くために用いることになり、少々違和感を生じてしまう。True BASIC 同様、PLOT LINES を単に PLOT と略記できるようにすることが望ましいであろう。

また、プログラムの実行開始時に図形表示が消去されるかどうかについても明確な規定がない。この点も明示しておく必要がある。

図形機能に関係して Full BASIC に追加したほうがよいと思われるのが True BASIC の get mouse 文である。get point 文でもマウス

を扱うことができるけれども、目的によっては True BASIC の get mouse 文を用いないと不便である。(両者の違いは、get point はボタンが押されるのを待つのに対し、get mouse は現在の座標とボタンの状態を調べることである。)

次に、その機能が用意されなくても支障のないものをリストアップしていく。中核機能単位に含まれる機能のうち必須でないものを、文献[4]の目次にしたがって示すと、

「6. 文字列」、「7.3 文字列配列」、「8.2 制御文」、「9.3 プログラム連鎖」、「11. ファイル」、「12.2 デバッグ」である。ただし、文字列については、JIS 基本 BASIC に定める程度の機能は用意しておいたほうが望ましいものと思われる。「9. プログラム分割」、および、「10. 入出力」については、文字列関数や文字列配列に関する部分は上記に対応して省かれることになる。目次に表れない細かな規定では、「5.5 数値 let 文」のうち LET 文の左辺に複数の数値変数を書いて同時的に代入する機能はなくても差し支えないものである。

図形機能単位に含まれる機能については、簡易図形出力機能単位に「13.5 絵及び描点出力」の規定を追加したものが実際に使われそうな範囲の目安である。ただし、「13.5」のうち機能語 MAT GET POINT を持つ図形入力文はなくてもよいほうの部類に属する。

図 6

```
10 SET DEVICE WINDOW 0,1,0,5/8
20 SET VIEWPORT 0,1,0,5/8
30 SET WINDOW left,right,bottom,top
```

Full BASIC での座標系設定の機構はかなり複雑であるが、描画領域が正方形であることに不満がなければ、「13.1 座標系」のなかでは機能語 WINDOW のみで十分である。SET WINDOW のみを用いて座標系を設定す

ると描画領域が正方形になるので座標範囲の設定を考えると計算が簡単であることと、どの機種で実行しても同じ形(相似形)が得られる点で True BASIC より扱いやすいであろう。ただし、画面を無駄なく利用するためには描画領域の形を変更する必要がある。その場合、機能語 VIEWPORT, および, DEVICE WINDOW が必要である。たとえば、図 6 のように座標系を設定すると画面内に縦：横が 5 : 8 の長方形の領域がとられてその領域に問題座標系が設定される。正方形のディスプレイを備えた機種がほとんど学校に導入されておらず、しかも、ドット数では進歩のない現状では、この程度のプログラムを書く面倒さは辛抱しなければならないものであろう。

なお、規格では図形の出力先を画面に限定していない。XY プロッターやページプリンタなどにも出力できるようにすべきである。

5. 数学教育が今後進むべき方向

解析分野については、どの程度までコンピュータに頼ってよいかについての議論はあるものの、進むべき方向はすでに定まっているとよい。組み合わせ論などの場合の数に関する理論も、コンピュータの利用を前提にして再帰的な考え方を教えるのに適した教材である。

しかし、Full BASIC のようにグラフィックスの扱いやすいコンピュータ言語が出現することで最も大きな影響を受けるのが幾何教育であろう。平成元年の指導要領改訂では、一次変換が高校から消え、変換幾何が初等幾何の一部として教えられるようになったけれども、これが正しい方向であるとは思えない。座標をベースにして行列を用いて議論を行う変換幾何は、実際に応用される際の用いられ方と同じであるという意味での実用性、理論構成のわかりやすさ、コンピュータを用いた実験の容易さ、一次変換の具体例であること、

行列や一次変換とともに議論することによって変換の合成の代数的な側面に目を向けさせることができるなど、多くの利点を持っている。また、マウスが気楽に使えるとも数学教育に大きな影響を与えるであろう。コンピュータとともに学ぶ幾何教育の建設が今後取り込むべき課題である。

参考文献

- [1] 文部省：高等学校数学指導資料 指導計画の作成と学習指導の工夫，学校図書，1992
- [2] 財団法人コンピュータ教育開発センター：学校で利用されるコンピュータシステムの機能に関する調査報告書 CEC仕様'90・詳細編，コンピュータ教育開発センター，1990
- [3] 財団法人コンピュータ教育開発センター：学校で利用されるコンピュータシステムの機能に関する調査報告書 CEC-BASIC'90 規格編，コンピュータ教育開発センター，1992
- [4] 西村恕彦 他編：アメリカ規格 Full BASIC, bit 別冊 共立出版，1990
- [5] J. G. Kemeny & T. E. Kurtz : Back to BASIC, 啓学出版，1990
- [6] J. G. Kemeny & T. E. Kurtz : True BASIC リファレンスマニュアル，啓学出版，1991