The Bulletin of Toyohashi Junior College
1995, No. 12, 175-185

原　著

UBASIC による線形代数学演習用 CAI

小　島　　誠

UBASIC（木田祐司，1994）というコンピュータ言語はあまり知られていないが，これを利用した数学教育用 CAI ソフトの開発について述べる．この言語の特徴は各教育者の教育方針に合わせてのソフト開発の容易さ，実行の高速性，そして最大の利点と有用性が計算誤差から解放されることである．1/3 は 0.333333 ではなく，分数の 1/3 のままで処理できる．分数が小数表示されてしまうと仮に誤差がなくても逆に分数表示にするのは大変である．誤差があると正しい分数表示は不可能である．行列の諸計算，逆行列，連立 1 次方程式，行列式の値などが通常の浮動小数点表示で計算されたのでは正解がわからない．数学教育の現場で誤差が介入しては困るのである．

1. 教育用ソフトについて

数学関連の市販ソフトとしては REDUCE，μ MATH などの数式処理システムを初めとして，Mathematica，Maple，. . . などと多数でているが，いずれも各分野の研究者向けのものである．小学校から大学までの学校教育の中ではそれぞれのレベルでの問題の把握や理解をすることから，計算練習，問題解決の能力アップ，そして応用能力のアップなどを目標としたソフトが望まれる．教育効果を上げるためにはその時点までの既習事項をよく整理把握しておくことと，これから学習させる新しい教育内容を良く吟味整理して，どのような形で学習者に提示するかが大きなポイントとなる．教育者側の教育理念，教育方針，教育方法には各個人の個性があり，学習者の学力レベル，学習態度や意欲などについても熟知していなければ本当の教育はできないので，できれば教育用ソフトは教育者個人個人で自分流のものを作ることが望ましいと考える．

開発すべきソフトはどのような形式を備えたらよいか？ ソフトの利用方法の説明やメニュー，問題の示し方などは，もちろん学習者に誤解なく正しく理解されるように工夫をしなければならない．解答についてはコンピュータが誤差のない正解を用意できることが必要である．通常のコンピュータはワードマシンで，市販のソフトはワード単位で計算処理をするため，1 つの数値が 1 ワード（単精度 32 ビットが多い）に格納される．固定小数点の整数の場合，BASIC の単精度では 4 桁までしか扱えない．小数点をもった数値，すなわち浮動小数点数値，いわゆる実数の有効桁数は単精度で 7 桁弱，倍精度でも 16 桁ほどである．FORTRAN，COBOL，C，. . . などでも状況は変わらない．したがって，ソフトメーカーのアプリケーションソフト（AP）でも同様である．浮動小数点数値は丸めの誤差を避けられないので正解を作ることができない．この問題点をどのように解決するかについては，次章で述べる UBASIC を利用することで可能である．つぎに学習者に解答をどのような形でさせるかであるが，通常の市販 CAI ソフトではいくつかの答えらしいものを用意して選択をさせるものが多い．これは現在いろいろな教育とかテストの場などで

利用されている非常に一般的な方法であるが，筆者はつねづね教育的でなく，良くない方法であると考えている．その理由は思考力を半減させる可能性を持つ方法であると考えるからである．

筋道を立てて解答にたどりつくための思考過程が大事なのであって，偶然性とか，逆思考を助長するのでは新しい知見の創造性を育てることは望めない．また，1 文字，1 数字ずつの解答枠を用意してこれに記入させる方法など（大学入試の共通テストの方法）もあるが，選択法を少し改良した方法とはいえても正解の形が見えてしまっているので決して良い方法ではないと思う．大量の採点をしなければならないというやむを得ない条件があるからであるが，教育の現場に悪い影響を与えていることに十分留意して，改善の努力を続けてほしいものである．筆者が提案したいのは解答の形式を可能なかぎり記述法とすることである．学習者は自分で解答を考えて正解に達したときに喜びを味わうものであって，この達成感が教育の原点であると考えるからである．

提示する問題は CAI ソフトの中にいろいろなタイプのものをたくさん用意しておくか，乱数で作成できるようにしておく．学習者が解答に到達したとき，コンピュータ側から正解を表示して，自分の解答が正しければつぎに進めばよいが，不正解のときはどこで間違えたかを自分なりに考えることを要求する．間違いの原因を自分で認識理解することの積み重ねが大きな進歩の原動力であるからである．

本質から少しそれるが，CAI ソフトにマルチメディアを援用することが効果的であると判断される場合は大いに利用するとよい．しかし，ソフトを自己開発するためにあれもこれも使えるようにしてからというのは負担が大きいと思うので，手近なできるところから開発を始めてみることが必要であると思う．

まとめると，既習事項と目標事項の整理をして，思考過程を大事にした解答記述方式を採用する．システム側での正解用意と，誤答の原因解明とともに問題に対する達成感を与えるように，教育の原点をにらんで自分流のソフト開発をする．

2. UBASIC の機能の中で利用する部分について

各種演算が誤差から解放されるためには多倍長演算が欠かせない．筆者自身，まだパソコンが普及する前の大型計算機の TSS 端末で FORTRAN を利用して，任意多倍長有理数演算のできるシステムの開発を手掛けた（小島，藤井（1984），小島（1985），小島，藤井（1989））が当時の高価なメモリを大量に必要とし，処理スピードも意外に遅く，大変であった．

UBASIC はパソコンを利用するためのプログラミング言語の中で一番ポピュラーな BASIC とほとんど同じ形式で手軽に利用できる多倍長演算用ソフトである．BASIC がインタプリタ言語であるのに対して，UBASIC はアセンブラで書かれた言語なので，十分高速処理ができる．また，いろいろな数学用関数が用意されているので，教育システムを作るのに便利である（森本（1992），岡野，山本（1992））．このソフト UBASIC は，システムのコピー，配布ともにフリーである．

まずはどんなことができるかを見てみる．図 1 が UBASIC の実行例である．

```
PRINT 1/2+1/6
 0.6666666666666666666
OK
PRINT 1//2+1//6
 2//3
OK
POINT 15
小数部のワード数は 15 (10進での表示桁数は 72)
OK
PRINT 1/47
 0.021276595744680851063829787234042553191489361702127659574468085106382978
OK
PRINT SQRT(2)
 1.414213562373095048801688724209698078569671875376948073176679737990732478
OK
?#PI
 3.141592653589793238462643383279502884197169399375105820974944592307816406
OK
?2^200
 1606938044258990275541962092341162602522202993782792835301376
OK
?(1+1/10000)^10000
 2.718145926825224864037664674913146536113822649220720818370865873787394594
OK
?#E
 2.718281828459045235360287471352662497757247093699959574966967627724076629
OK
?(1+1/10^50)^(10^50)
 2.718281828459045235360137467142892419432217491158397867525804160850893863
OK
```

図1　実行例

UBASICとBASICとの比較でその異同を列挙する．●印はBASICと同じ，◎印はUBASICにあってBASICと異なる機能である．

◎　プログラミング用の予約語は大文字で入力しても自動的に小文字に変換する．ユーザーが使う変数名は大文字のまま．

◎　プログラム中の余分な空白は除去してつめる．注釈行はこの限りではない．

●　関係演算子　＝，＞，＜，＜＞，＞＝，＜＝

●　算術演算子　＋，－，＊，／，￥（2数とも整数に限る），

◎　算術演算子　／／（2整数または有理数（分数）の有理数除算）

◎　算術演算子　@（2整数の除算の余り），￥の直後ならば@の計算をしなくてもシステム変数resに入っている．ただし，他の演算をすると壊れる．

◎　1変数の多項式1＋2＊_X＋3＊_X^2($3x^2+2x+1$)この式への数値の代入，因数分解，微分計算などの簡単な数式計算ができる．

◎　C言語同様A＋＝B，A＊＝B＋Cなどが使える．

◎　print＝print＋lprint＋“ABC”出力先を簡単に変更できる．

○　if文，for文は基本的に同じ．ただし，for文の初期値，終了値，増分はいずれも整数，ループからの途中脱出はcancel forを用いる．

◎　変数の型　　単変数　　16 bits　　A% のように % をつける；－32767 ～ 32767
　　長変数　　n words　word n で指定（n ≦ 542）
　　小数部分の word 数を point n で指定
　　特別変数　542 words　A# のように # をつける
　長変数, 特別変数には整数, 有理数, 実数, 複素数などの数値のほか, 文字列, 多項式も代入可.
◎　黙って普通に変数（英字で始まる英数字 16 文字）を使うと 542 words で
　整数は 10 進 2600 桁位まで.
　有理数　実数や複素数が混ざると実数化される.
　実数の小数部の桁数は point 命令で指定. 初期設定は point 4（10 進 19 桁）.
　　したがって，整数部分は 538 words.
　複素数 1.2 + 3.45#i で実部，虚部とも 270 words，10 進 1300 桁位まで.
　文字列（文字型変数はない）1 変数に 1 バイト文字で 1084 文字まで入る

図 1 の実行例とともに見ればわかるように，特別大きな桁数の整数論などの計算（木田，牧野 (1994)）をしない限り，整数と有理数の計算で誤差のない通常の数学計算ができる.

この小論で利用するもの以外の数学用関数としては普通の sin, cos, tan, atan, sinh, cosh, exp, log, abs, sgn, sqrt, int, fix, max, min, swap などのほかに各ビットの値を得るための bit, 有理数の分母，分子を得るための den, num，最大公約数，最小公倍数の gcd, lcm，順列，組み合わせの factorial, combi などなどたくさんが用意されている.

その他，便利な機能として encode, gsave, gload などもある. ここに挙げたのは UBASIC のいろいろな機能の中のごく 1 部である.

3. 開発した LINALG について

線形代数学の演習用の CAI ソフトの開発について述べる. 大学初年級での線形代数学の基礎でこれだけは，そしてこれだけ知っていればというのが行列の階数，連立 1 次方程式の解，行列式の値，逆行列を求めること，そして，ベクトルの 1 次独立，1 次従属の概念とベクトルで生成された部分空間の次元の概念である. そして，これらの計算のための，基本的で重要な行列の基本変形，いわゆる掃き出し計算があるが，その演習用の CAI ソフトがこれから述べる LINALG である.

行列の定義，加減乗の計算，正則行列の定義などを初めとして，上記の定義や概念については説明済みであるとする. 実際に，連立 1 次方程式や逆行列の計算をさせるとき，3 種の基本変形の使い方が良くわかっていなかったり，間違った使い方をしたり，数値計算を間違えたり，変形が終わったあとの処理がうまく出来なかったりというのが正しく計算できないことの原因である. この中で，数値計算の間違いは計算法を知らないということではなく，全くの不注意で，しかも結構良く間違うというものである. これは掃き出し計算を理解するうえで本質的ではないので，その部分はコンピュータに任せて学習者の負担を軽減したほうが，大事なことの理解に意識を集中できて良いと考えられる. 算術演算子「／／」を使って，正しい計算をコンピュータが行う.

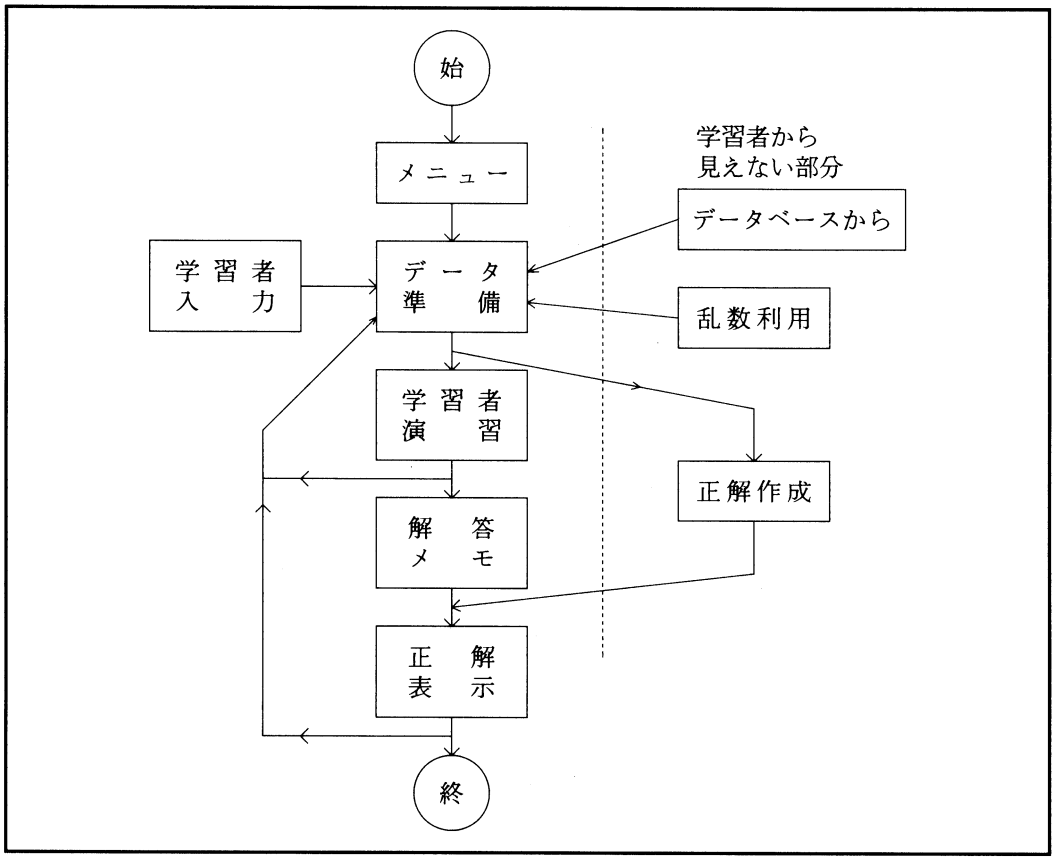

図 2　全体の流れ図

(1) 最初の画面で演習の項目を選択し，必要なデータを学習者自身が入力するか，コンピュータに任せるかして，掃き出しの計算に入る．　（プログラム 180 ～ 230）

(2) 行列を表示して，3 種の基本変形　（プログラム 240 ～ 250）

① ある行（ｉ）を C（≠ 0）で割る
② ある行（ｉ）から他の行（ｊ）の C 倍を引く
③ ２つの行（ｉとｊ）を入れ換える
④ 初めからやりなおす　⑤ 終わる

のいずれかを番号で選択させ，それぞれに応じて i，j，C を入力させる．④の「初めからやりなおす」は途中で失敗したと思ったときに，掃き出しに入る前の行列を保存確保してあるのでそれを表示して，(2) の変形を初めから繰り返すことができる．
行列の変形が終わったら⑤でつぎへ進む．

(3) 最後の行列の状態から学習者が解答をメモする．　（プログラム 260）
これが正しくできればそれぞれの問題に対しての知識が良く理解されていると判断できる．特に，連立 1 次方程式の解，行列式の値，従属なベクトルを独立なベクトルの 1 次結合として正しく表現できるかどうかなどの処理に注目する．

(4) 正解を表示する．　（プログラム 270 ～ 290）

(5) 学習の解答が正しくなかったとき,　　(プログラム 300 ～ 310)

(2) からやりなおすか, 終わるかを選択する.

以上がこのソフトの概要である. 細かいことであるが, 画面を見易くするために, 説明, 選択, 入力, 表示, 正解などのそれぞれの文字色を決めてカラー表示をしている. 図3にメインプログラムを示すが, 他に 120 ～ 160 行の注釈にあるようなサブプログラムがある. なお, 扱うデータが整数, 有理数であれば誤差はない.

```
100  ,  行列の基本変形  ;  "LINALG"  on UBASIC
110  ,        (演習用)                              '94 3/5 by M.Kojima
120  ,  Gauss-Jordan's Sweepout Method  *OWARI    400-, *RETRY   500-
130  ,  *MENU   1000-,  *TITLE   1200-, *ADIM    1300-, *DDIM   1500-
140  ,  *AINPUT 1700-,  *DINPUT  1900-, *APRI    2000-, *SWOUT  2100-
150  ,  *SWOUT0 2400-,  *SEIKAI  2700-, *SOL     2800-, *DETINV 3100-
160  ,  *A0PRI  3200-,  *LINCOMB 3300-, *DINPUT0 4000-, *FDATA  5000-
170  ,  ================================================================
180  gosub *MENU:if or {KOMOKU<1, KOMOKU>5}  then 180
190       if KOMOKU=5 then goto 310
200  gosub *TITLE:if DEHTA=1 then gosub *ADIM:else gosub *DDIM:endif
210  dim B(M.NN), A0(M, NN), IND(N), PAR(N), NPAR(N), A(M, NN) :console 1, 24, 0
220  if DEHTA=1 then gosub *AINPUT:else gosub *DINPUT:endif
230  for I=1 to M:for J=1 to NN:B(I, J)=A(I, J) :A0(I, J)=A(I, J) :next J:next I
240  locate 0, 17:color 5:print "    掃き出しの計算を始めなさい":color 7
250  gosub *APRI:gosub *SWOUT:color 5:' ************************
260  print "解答をメモして下さい. メモがすんだらどれかキーを押して下さい."
270  gosub *SWOUT0:' ****************************************
280  Q=inkey:if Q="" then 280
290  gosub *SEIKAI:print:gosub *OWARI:' ************************
300  if K=0 then gosub *RETRY:goto 240
310  color 7:console 0, 24, 1:cls 3:end:' ================================
```

図3　プログラム（メインプログラムだけ）

4. 実行例について

"LINALG" をロードして, 実行すると図4のメニュー画面が表示される.

```
                    行列, 行列式の演習を始めます
                    つぎの項目を番号で選択しなさい

                      ①  ;  行列の階数を求める
                      ②  ;  連立1次方程式を解く
                      ③  ;  逆行列を求める
                      ④  ;  行列式の値を求める
                            (基本変形で三角行列に変形をして)
                      ⑤  ;  終わりにします
番号？1

                    行列のデータについて

                      1  ;  自分の持っているデータを入力する
                      2  ;  コンピュータのデータに任せる
番号？1
```

図4　メニュー画面

データをコンピュータのデータに任せるときは，このソフト内に用意されたdata文データからデータを読み込む（＊FDATA）．ただし，③，④の正方行列のデータの一部は，そのほとんどが正則行列となるように用意された乱数データも使われるように設定されている（＊DINPUT0）．

自分のデータを使うときは図5のように行列Aの型を入力し，つづいてその成分を行方向に1つずつ入力していく．分数の入力は1//3のようにすればよい．この場合，入力が終わった後，行列が表示されて入力ミスがあればその訂正ができるようにしてある．

```
                    行列の階数を求めます
行列Aの行数，列数＝？3，4
Aの各成分を，行方向に1つずつ
？1
？3
？1
？－8

？－2
？－5
？－1
？13

？3
？8
？2
？－21


  1        3        1       －8
－2       －5       －1        13
  3        8        2       －21
      訂正しますか？　するならば，『Y』キーを
        しないならば，Y以外のどれかのキーを
```

図5　データの入力

データの入力が終わると，図6の掃き出しの計算に入る．掃き出しが終わったと判断したときは⑤で終わる．

```
                    行列の階数を求めます

        掃き出しの計算を始めなさい

    1        3        1       －8
  －2       －5       －1        13
    3        8        2       －21
                ――――――――――――――――――――――――――――――――――――
①　ある行（ｉ）を Ｃ(≠0) で割る        ②　ある行（ｉ）から他の行（ｊ）の Ｃ倍 を引く
③　２つの行（ｉとｊ）を入れ換える      ④　初めからやりなおす              ⑤　終わる
上の丸数字のどれかを選びなさい　？2
  選択　＝　2
ｉ，ｊ，Ｃ　＝　？2，1，－2
  2，　1，－2

    1        3        1       －8
    0        1        1       －3
    3        8        2       －21
                ――――――――――――――――――――――――――――――――――――
①　ある行（ｉ）を Ｃ(≠0) で割る        ②　ある行（ｉ）から他の行（ｊ）の Ｃ倍 を引く
③　２つの行（ｉとｊ）を入れ換える      ④　初めからやりなおす              ⑤　終わる
上の丸数字のどれかを選びなさい　？
```

図６　掃き出しの処理

解答をノートにメモするように指示がでて，この間にパソコン側では正解の計算を行っている．メモがすんでどれかのキーを押すと，図７のようにその正解が表示され，学習者のメモと照合して，この演習を終わりにするかやりなおすかを選択する．

```
                    行列の階数を求めます

①　ある行（ｉ）を Ｃ(≠0) で割る        ②　ある行（ｉ）から他の行（ｊ）の Ｃ倍 を引く
③　２つの行（ｉとｊ）を入れ換える      ④　初めからやりなおす              ⑤　終わる
上の丸数字のどれかを選びなさい　？2
  選択　＝　2
ｉ，ｊ，Ｃ　＝　？3，2，－1
  3　，　2　，－1

    1        0       －2        1
    0        1        1       －3
    0        0        0        0
                ――――――――――――――――――――――――――――――――――――
①　ある行（ｉ）を Ｃ(≠0) で割る        ②　ある行（ｉ）から他の行（ｊ）の Ｃ倍 を引く
③　２つの行（ｉとｊ）を入れ換える      ④　初めからやりなおす              ⑤　終わる
  選択　＝　5
解答をメモして下さい．メモがすんだらどれかキーを押して下さい．
正解はつぎの通りです．
ｒａｎｋ（Ａ）＝　2
独立な列は　　1　2
        {第 3 列} ＝　　－2・{第 1 列}　＋　　1・{第 2 列}
        {第 4 列} ＝　　 1・{第 1 列}　＋　－3・{第 2 列}
正しくできましたか？　　正解，または終わりの時は，1 を
間違えて，同じ問題を初めからやりなおすときは 0 を入力しなさい？
```

図７　正解の表示例Ｉ

各項目での正解の表示内容は次の通りである.

① 行列の階数を求めるとき,

階数のrank（A）のほかに，独立な列ベクトルの列番号（番号の小さいほうから独立なものを探す),従属なベクトルがあるときは独立なベクトルの1次結合の形を表示する(図7).

② 連立1次方程式を解くとき,

係数行列の階数rank（A）と，解がないか，解が1つだけか，そして，基本解の1次結合として表される多くの解なのかのいずれかを表示する（図8).

```
                連立１次方程式を解きます

 @@@@@     多くの解があります．たとえば     @@@@@
    特殊解，基本解，．．．と表示します

  特殊解は
      － 3
        2
        0
        0

 基本解（１番目）
        2
      － 1
        1
        0

 基本解（２番目）
      － 1
        3
        0
        1

 正しくできましたか？     正解，または終わりの時は，１ を
 間違えて，同じ問題を初めからやりなおすときは ０ を入力しなさい  ？
```

図8　正解の表示例Ⅱ

③ 逆行列を求めるとき,

①の結果のほかに，行列式の値，そして，その値が0でない（正則の）ときは逆行列を表示する（図9).

```
                逆行列を求めます
i, j, c = ?2, 3, −7／／5
  2 ,  3 , −7／／5

  1          0          0          :          1          4／／5       −2／／5
  0          1          0          :         −1         −9／／10       7／／10
  0          0          1          :         −1         −1／／2        1／／2
                ----------------------------------------------------------

①  ある行（i）を C(≠0) で割る        ②  ある行（i）から他の行（j）の C倍 を引く
③  2つの行（iとj）を入れ換える      ④  初めからやりなおす              ⑤  終わる
上の丸数字のどれかを選びなさい  ? 5
  選択 =  5
解答をメモして下さい．メモがすんだらどれかキーを押して下さい．
正解はつぎの通りです．
rank (A) =  3
det (A) = −10

Aの逆行列＝

  1          4／／5       −2／／5
 −1         −9／／10        7／／10
 −1         −1／／2         1／／2

正しくできましたか？     正解，または終わりの時は，1 を
間違えて，同じ問題を初めからやりなおすときは 0 を入力しなさい  ?
```

図9　正解の表示例Ⅲ

④　行列式の値を求めるとき，
　　③と同じ結果を表示する（図9）．

このソフトはロードの手順だけ学習者に教示しておけば，あとは学習者自身で画面の指示にしたがっていろいろな線形代数学の演習問題を実行することができる．

5. 今後の課題

学習者の解答と正解のマッチングをして，自動採点と学習記録ができるとよい．ただし，rank（A）と det（A）と逆行列については単に数値のマッチングを行えば，結果についての採点はできるが，掃き出し計算は途中の計算も，最後の結果も一通りとは限らないので，ベクトルの独立，従属に関することと，連立1次方程式の解については，ただ単に数値のマッチングをするだけではだめである．また，学習記録は教育者側だけでなく，必要に応じて学習者自身もこれを利用できる形で用意することが望ましい．

参考文献

［1］木田祐司；UBASIC86　多倍長計算用 BASIC，日本評論社（1994）.
［2］木田祐司，牧野潔夫；UBASIC によるコンピュータ整数論，日本評論社（1994）.
［3］小島　誠，藤井えみこ；任意倍長精度の有理数演算について，第 8 回 NEAC ユーザー会，pp. 169〜180（1984）.
［4］小島　誠；π の連分数展開　―任意倍長精度の有理数演算の応用例―，名古屋市立大学教養部紀要 vol. 31, pp. 1〜3（1985）.
［5］小島　誠，藤井えみこ；AMPA システムによる固有値の高精度計算，名古屋市立大学教養部紀要 vol. 35, pp. 1〜10（1989）.
［6］森本光生；UBASIC による解析入門，日本評論社（1992）.
［7］岡野節，山本浩；1992，UBASIC を用いた数学教育について，CAI 学会誌 vol. 9, no. 4, pp. 158 〜 163（1992）.