BASIC による統計プログラム入門

(9) BASIC を越えて

古川 カ*

Introduction to Statistical Programs in BASIC (9) Over the BASIC

Tsutomu FURUKAWA

National Institute of Animal Industry
Norindanchi P. O. Box 5, Ibaraki 305

IX. 終わりに

この連載では、獣医畜産分野の問題を取り上げながら、統計的解析のための実用的な BASIC のプログラム作成法を述べてきた。統計的手法についてはほんの一部しか取り上げることができなかったが、BASIC についてはグラフィックス関係を除きほぼ概略を述べることができたと思う。連載の最後に当たり、BASIC に関する話題と、データ解析の方法について私見を述べてみたい。

1. BASIC と OS

パソコン用のプログラミング言語として BASIC が広く利用されるようになった理由の一つとして、ハードウェアを購入すると BASIC が一緒についてきたことを挙げるができる。

パソコンと BASIC がセットになる経過については興味深い逸話がある¹⁾。1975年に組み立てキットとして最初のパソコン ALTAIR8800が作られた。このことを知った当時19歳のビル・ゲーツ（現マイクロソフト社会長）は、雑誌に掲載された配線図だけをたよりに ALTAIR8800用の言語作りを始めた。このとき取り上げた言語が初心者でも使える BASIC

であった。ビル・ゲーツは実物にはさわらないでたった60日間でパソコン用 BASIC を作り上げ、おもちゃと思われていた ALTAIR8800をコンピュータに変身させた。この BASIC は30万ドルで買われ、彼はそれを資金にマイクロソフト社を設立した。その後、ほとんどのパソコンはマイクロソフトの BASIC を採用するようになった。

一方、パソコンのハードウェアは急速に発達し、メモリーの内部容量や周辺装置も増えてきた。BASIC はプログラム言語であるとともに、それ自体もファイルの保存や消去、ファイル名の変更などの命令を持っている。しかし、そのファイル管理機能は低いものであり、パソコンの資源を活用するには充分ではなかった。例えば、フロッピーディスクの初期化やファイルのコピーといった基本的なファイル管理を行なうためには、それぞれ別のプログラムを利用しなければならなかった。そこで、コンピュータの資源を管理し、動かす最も基本的なソフト、つまりオペレーティングシステム (OS) の必要性が生じてきた。

1970年代の8ビットパソコンの時代、すでにアメリカにおいては CP/M という OS が使われ、機種に依存しないソフト開発が行なわれていた。現在も広く使われている WORD STAR や dBASE III の原型は、その当時に CP/M 上のソフトとして開発

*農林水産省畜産試験場
〒305 茨城県筑波農林団地内局 私書箱5号

されたものである。しかし、日本ではスタンド・アロンの BASIC が主流であり、本格的な OS を利用するまでには到らなかった。

1982年に日本でも16ビットパソコンが本格的に発売されるようになったが、ソフトは依然としてスタンド・アロンの BASIC か機械語によるものであった。16ビットパソコン用の OS である MS-DOS が本格的に意識されるようになったのは日本語ワープロの「一太郎」が発売されてからである。共通の OS を利用することにより、異なるソフト間でファイルが共用できることが分かり、現在では、アプリケーションソフトはほとんど MS-DOS 上で開発されるようになった。

さて、個人で BASIC を使ってプログラミングを作成し、問題を解決している分には OS のことはあまり意識する必要はなかった。ファイル管理の煩わしさを我慢すればよかった。しかし、アプリケーションソフトから出力したデータを自作プログラムで処理しようとすると、OS 上の BASIC が必要になってくる。また、ある程度アプリケーションを使いファイル管理の便利さになれば、どの仕事でも共通の環境、つまり OS を維持したくなってくる。MS-DOS 上では BASIC 以外にもいろいろなプログラム言語を利用することができる。

2. プログラム言語について

コンピュータに一連の命令を与えることをプログラミングといい、その際、コンピュータと会話するために用いる言葉をプログラム言語という。プログラム言語の多くは大型コンピュータとともに発展してきたが、その多くはパソコンでも利用できるようになっている。

主なプログラム言語とその特徴は次の通りである^{2,3)}。

機械語：0と1から成る、コンピュータが理解できる唯一の言語。普通はプログラム言語とはいわない。

アセンブリ言語：機械語の命令を記号に置き換え、人間が読み書きしやすいようにした言語。アセンブリ言語を機械語に変換するソフトをアセンブラというが、アセンブリ言語そのものをアセンブラと呼ぶこともある。

FORTRAN：科学技術計算用に世界で最初に開発されたプログラミング言語。大型計算機の主流言語であり、パソコン用のコンパイラも発売されている。BASICによく似ているので、

BASICが使えればFORTRANも簡単に使えるようになる。科学技術計算のプログラムが多く開発されているため、上手に利用すればプログラム開発の手間が省ける。

COBOL：事務処理用に開発された言語で、事務処理用のプログラムはほとんどCOBOLで書かれている。

ALGOL：アルゴリズム記述用の言語であり、コンピュータプログラムだけでなく、数学的アルゴリズムの記述にも用いられる。パソコンでは使われていないが、PascalやCはALGOLの流れをくむものである。

Pascal：プログラミングの教育用に開発された言語で、豊富なデータ構造と制御構造を持ち、よく体系化されている。BASICとともに入門用言語として利用されているが、システム開発用としても用いられている。

C：ミニコンのオペレーティングシステムUNIXの記述用言語として開発された。Pascalに似た構造でありながら簡潔なプログラミングが行なえ、システム開発に向いていることから、最近の人気言語である。

LISP：人工知能や数式処理などの記号処理のための言語。

Prolog：述語論理（人間の言葉についての論理）を土台にした、人工知能用言語である。

これらプログラム言語は、コンピュータの言葉に近い人間の言葉に近いということで、低級言語、高級言語という分類がなされている。アセンブリ言語は機械語と1対1に対応していることから低級言語と呼ばれ、その他の言語は人間の言葉に近いことから高級言語と呼ばれている。また、LISPは人工知能用言語として脚光を浴びているが、その誕生はFORTRANと同じくらい古く、人工知能のアセンブラとも呼ばれている。C言語はハードウェアの直接操作が容易であることから、高級アセンブラとして評価され、低級言語と高級言語の中間に位置づけられることもある。

プログラム言語で書かれたプログラムをパソコンへの命令として与え、実行させるためには、パソコンが理解できる唯一の言語である機械語に変換してやらなければならない。その作業はもちろんパソコン自身で行なうが、そのためにはインタープリタあるいはコンパイラという道具（ソフト）を利用する。インタープリタはスタンド・アロン BASIC に代表

されるように、プログラムの1行ずつを解釈しながら実行する方法で、いわば外国人と通訳を介して話しているようなものである。BASICはこの方法が多い。コンパイラはいわば、外国語の本を翻訳してもらい、日本語になってから読むようなもので、プログラム全体をまとめて機械語に変換する方法である。実行にあたっては機械語に変換されたファイル名を入力するだけでよい。ほとんどの言語はコンパイラにより機械語に変換される。BASICにはインタプリタだけでなくコンパイラもある。

BASIC インタプリタに慣れているとコンパイラ系の言語は面倒に感じてしまう。なぜなら、コンパイラ系言語では、まずエディタでソースプログラムを作成し、コンパイラでエラーチェックをして、エラーがあればまたエディタで修正する、ということを繰り返さなければならないからだ。しかし、コンパイラ系の言語では、一度完成したプログラムはMS-DOSからファイル名を入力するだけで動かすことが出来、また、処理速度も早いのが魅力である。MS-DOS上のBASICを利用する場合は、まずインタプリタによりプログラム作成とデバッグを行ない、完成したプログラムをコンパイラで機械語に変換して利用する、という使い方が一般的である。

3. BASICとC

パソコンとBASICが一体となって販売されたことにより、BASICさえ身に付ければ小さいながらもコンピュータが使えるという時期があった。そういう意味で、BASICはパソコンの普及に大きな役割を果たした。しかし、一方では、多くの人がBASICに坐折したためパソコンの普及が遅れたという意見もある。事実、パソコンがマニアからビジネスマンへ広がった背景にはソフトの充実がある。プログラム言語を知らなくても、アプリケーションソフトを使えば仕事ができるという状況こそパソコン利用の必要条件といえる。現在では、パソコン活用の道は、いかにアプリケーションソフトを使いこなすかであり、いかにプログラミングするかではなくなってきた。

ではプログラム言語の学習はまったく無意味かというと、そうではない。現実には抱えている問題をアプリケーションソフトで解決するためには、その問題をコンピュータ向きの手順に分解する必要がある。その手順はアプリケーションソフトでもプログラム言語でも同じようなものである。一つのプログラム言語を理解すればアプリケーションソフトの活用に

も道が開けるし、また別の言語も容易に理解することができる。

アプリケーションソフトのなかでも、農家あるいは事務所などでは業務用ソフトを利用することが多いが、試験研究機関や家保、指導機関では汎用的なソフトが利用されている。表計算やデータベースのソフトは、一時、簡易言語と呼ばれたこともあるように、命令手順の構築は一種のプログラミングになっている。従って、プログラム言語を学習しておけば汎用的なアプリケーションソフトの活用に役立つわけである。

BASICは入門用言語として利用されてきたが、それにとどまらず多くのアプリケーションソフトの開発にも利用されてきた。特に、試験研究機関で作成されたプログラムの約90%ではBASICが使われている(1987年調査)。しかし、BASICには次のような欠点が指摘されており⁴⁾、識者の評判はあまりよくない。

- ・実行速度が遅い。
- ・サブルーチンに引数が渡せない。
- ・変数名がプログラム全体で共通であり、サブルーチンの中だけで有効ということがない。
- ・行番号がプログラムと深く関わっている。
- ・そのため、別々に作ったプログラムをつないで使うことが困難である。
- ・ユーザ関数の定義が1行の範囲に限定される。
- ・予約語を変数名の先頭に使えないものがある。

そして最後に、構造化言語でない、従って駄目な言語である、という批判が突きつけられた。

BASICの対局にいて流行のパソコン用言語としてもはやされているのがC言語である。Cの特徴としては次のようなことがあげられる²⁾。

- ・英小文字ベースで書かれるので読みやすい。
- ・プログラムは関数の集まりとして書かれるので、効率的なプログラミングが可能である。
- ・制御構造やデータ型が豊富であり、構造化プログラミングが可能である。
- ・演算用の種類が多く、式をコンパクトに書くことができる。
- ・メモリの値を直接操作するなど、ハードウェアに近いアセンブラのような機能を持っている。
- ・言語仕様が標準化されており、異機種間の互換性が高い。

Cは、先にも述べたように、高級アセンブラとして位置づけられることもあり、システム開発などに携

わっているプロあるいはセミプロのプログラマ達が好んで使っている。パソコン雑誌の記事やプログラム言語解説書などもほとんどCで占められている。そのため、BASICしか知らないという流れに取り残されたようで、Cを使っていると格好がよいという風潮さえ見られる。プログラム言語を初めて学習するという場合にもCからスタートする例がみられるが、独習するには取っ付きにくいのではないかと心配される。

これから新しいプログラム言語を学習しようとするならば、流行にとらわれず使用目的に応じて言語を選択したほうがよいだろう。例えば、大型計算機を使いながらパソコンも利用するならばFORTRAN, パソコンのハードまで含めたシステム開発ならばC, プログラムのソースを公開してみんなで利用するならBASICが勧められる。何種類もの言語を操る必要はなく、むしろ一つの言語について理解を深めたほうがよいと思われる。

4. 新しいBASIC

さて、Cに押され気味のBASICであるが、最近、メーカー以外の処理系が発売され人気をよんでいる⁵⁾。

この連載において示してきたプログラムはすべてN88-BASIC(86)を用いて作成したものである。N88-BASIC(86)は構造化言語ではないが、それでも、プログラムする側の工夫により構造化プログラミングが可能なることを示してきた。次に紹介する2つの処理系は構造化BASICと呼ばれ、構造化のための命令を導入した言語仕様となっている。

神津システム設計事務所のBASIC/98は数少ない国産の言語処理系であり、N88-BASIC(86)に対して上位互換性を保ちながら構造化の拡張仕様を持っており、インタプリタとコンパイラがある。Quick-BASICはBASICの本来マイクロソフトが開発した処理系で、インタプリタ、コンパイラ、エディタおよびデバッガが一体となっている。

BASIC/98とQuick-BASICでは文法的に異なる仕様であるが、構造化のために拡張された仕様には共通の部分が多い。BASIC/98においてはN88-BASIC(86)に比べて次の命令が拡張されている⁶⁾。

○引数付きのサブプログラム

・サブプログラムの定義

SUB サブプログラム名 (仮引数)

:

EXIT SUB

:

END SUB

サブプログラム内部の引数以外の変数はすべてローカル変数となる。

・サブプログラムの呼び出し

CALL サブプログラム名 (実引数)

○ユーザ関数のブロック化

・ブロック型ユーザ関数の定義

DEF FN 関数名 (仮引数)

:

EXIT DEF

:

FN 関数名=式

END DEF

○制御構造におけるブロックステートメント

・IF ステートメント

IF 条件 THEN

:

ELSE

:

END IF

・SELECT ステートメント

SELECT 式

CASE 式

:

CASE ELSE

:

END SELECT

・DO ステートメント

これには大きく2つの型がある。

DO~

:

EXIT DO

:

LOOP

あるいは

DO

:

EXIT DO

:

LOOP~

~には次のいずれかを書く。

```

      WHILE  条  件
      UNTIL  条  件
      REPEAT 回  数
・FOR~NEXT ループからの抜け出し
      FOR  反復条件
      :
      EXIT FOR
      :
      NEXT

```

○行番号にとらわれないプログラム

行番号のないプログラムのロード、あるいは行番号を除いたプログラムのセーブが可能。

Quick-BASIC では、さらに、構造体までサポートされている。

これらの命令を用いれば、制御構文に関してはもはやCと大差がないプログラムを書くことができる。Cの命令が記号的であるのに対し、BASICの命令は記述的であり、プログラムはむしろ構造化BASICの方が読みやすいといえる。科学技術計算のためのプログラミングに関しては、さきに述べたBASICの欠点はほとんど解消されている。

BASIC/98やQuick-BASICの詳細はそれぞれの解説書に譲るが、今回の演習問題の解答例にはBASIC/98を用いて作成したプログラムを掲げた。

この連載が始まった時期と比較すると、パソコンを取り巻く情勢は大きく変化してきた。

まず、ハード的にはCPUの高性能化、高速化が進み、32ビットCPUを備えたパソコンがビジネスユースの主流になってきた。ハードディスクや拡張メモリのような周辺機器も充実してきた。一方ではラップトップやブックタイプといった、運搬性に優れた小型パソコンの売行きが伸びている。パソコン通信とかネットワークという言葉が日常使われるようになり、パソコンを介しての情報交換が盛んになってきた。

ソフト面では、日本語ワープロ、表計算、データベースについて主流のソフトが固まってきた。特に、表計算とデータベースという汎用的なデータ処理分野ではアメリカ製アプリケーションソフトの人氣が高い。

統計分析用ソフトでは、多変量解析を中心に多くのパッケージが販売されているが、大隅・秋山²⁾は、ほとんどの統計ソフトは厳密なチェックに対し誤った答えを出力する、と警告した。この記事は統計ソフトの購入あるいはプログラム作成の参考になると

思うが、平均値の差の検定でさえ不正確な結果を出力するソフトがあるには驚かされる。「コンピュータが計算したのだから」「市販ソフトだから」という過信してはならない、ということだろうか。

一方では、大型計算機用の統計分析パッケージがパソコンでも使えるようになってきた。社会科学用のSPSSと統合化パッケージのSASであるが、両ソフトとも世界中の大型計算機で利用されてきたソフトである。家畜育種を中心に畜産分野でよく使われていたW. R. HarveyのLSML76はLSMLMWへバージョンアップされ、パソコン用のシステムも販売されるようになった。これはIBM-PC用であるが、NECのPC-9801のMS-DOS上でも利用できる。パソコン用LSMLMW PC-1 VERSIONの申し込み先は下記の通りであり、値段は郵送料込みで150ドルである。

Dr. Walter R. Harvey
4255 Mumford Drive
Columbus, Ohio 43220
U.S.A.

様々な統計用パッケージが販売されているものの、全ての分野や手法が包含されているわけではなく、どうしても自分でプログラミングしなければならないことがある。BASICを用いてプログラミングを行うとき、最も面倒なところはデータ管理の部分である。つまり、データの入力、チェック、修正、変換、並べ替え、抽出、出力といったことにプログラミングの手間がかかる。

この中で、チェックと修正についてはデータファイルを利用することによりプログラムの負担を軽くすることができる。また、統計分析用のソフトを活用する場合も、データはファイルから入力することにより使い勝手がよくなる。このように、データを分析用プログラムとは別に管理する場合は、データファイルを管理するためのソフトが必要となる。

このための最も簡単な方法としては(5)シーケンシャルファイルの項で述べた、BASICのスクリーンエディタ機能を用いる方法がある。この方法では、新たなソフトを購入する必要はないが、ファイルの作成、チェックおよび修正程度のことしか行えない。次に、エディタソフトあるいは日本語ワープロソフトの半角モードを用いてテキストファイルを作成することもできる。データの置換や簡単なソフトは可能である。しかし、基本的にはデータ管理は表計算ソフトあるいはデータベースソフトを活用すること

が望まれる。

表計算ソフトとしては Multiplan や 1-2-3 がよく利用されているが、基本的な機能や使い方は似通っている。表計算ソフトは電子式集計表あるいは2次元電卓のようなもので、行と列に記号の数字をつけた大きな集計表と考えればよい。集計表の升目(セルと呼ばれる)にデータを書き込むだけでデータが入力されるため、入力そのものだけでなく、チェックや修正も簡単に行なえる。セル同士の計算や、列あるいは行の集計機能もある。データを並べ替えたり、条件を指定してその合致したデータを抽出すること、あるいは、簡単な統計計算も行なうことができる。プリンタへの出力は画面イメージのまま行なわれるため、桁数を揃えることに頭を悩ませる必要はない。また、データをテキストファイルへ出力させれば、別のソフトや BASIC のプログラムを用いてそのデータを解析することができる。

筆者の場合は、ほとんどのデータは表計算ソフトで管理し、統計分析は自作のソフトか LSMMLW を用いている。表計算ソフトでは、データ管理とともに、簡単な基本統計量を計算し、グラフ化してデータの性質を読み取っておく。別のソフトへデータを渡すときは、そのソフトの入力書式に合わせたファイル出力を行なう。統計ソフトの計算結果はファイルへ出力し、エディタを用いて、そのファイルの不要部分を削除してからプリンタへ出力するようにしている。

7. おわりに

パソコンはこれからますます普及し、1人に1台が自由に使えるようになってくると思われる。それにつれてデータ処理の手法も複雑化し、ブラックボックス化してくる可能性がある。データの背景に対する認識とソフトが出力する処理の中身の理解がおろそかにされると、とんでもない結果を導くことになる。したがって、技術者や研究者には対象となる家畜や実験に対する深い洞察と理解がますます求められることになる。その認識に立って、パソコンやソフトを使いこなしたいものである。

連載を終るに当たり、本稿発表の機会を与えて頂きました東京農工大学農学部小川益男教授に深謝致します。

[演習問題の解答例]

(13) 問題中のヒントにミスプリントがありました。次式のように訂正をお願いします。

$$x_2 = \rho z_1 + \sqrt{1 - \rho^2} z_2$$

BASIC/98によるプログラム例と出力結果を図60に示す。

(14) BASIC/98によるプログラム例と出力結果は図61のとおりである。混合培養の場合は実験結果とシミュレーション結果に違いがみられた。したがって、この2種類の酵母は、単にアルコールの生成だけによってお互いの生長を干渉しているのではないと考えられた⁸⁾。

```

1 'save "enshu-13.bas"
1010 REM ----> 相 関 の 有 る 正 規 乱 数
1020 DEFDBL S
1030 DEF FNNRM (R, AV, SD) = AV + R * SD
1040 DEF FNSD (N, S, SS)
1050   IF N > 1 THEN
1060     FNSD = SQR ((SS - S * S / N) / (N - 1))
1070   ELSE
1080     FNSD = 0
1090   END IF
1100 END DEF
1110 INPUT "雌 豚 の 数" = ";NP
1120 INPUT "1 日 平 均 増 体 重 の 平 均" = ";AVDG
1130 INPUT "1 日 平 均 増 体 重 の 標 準 偏 差" = ";SDDG
1140 INPUT "背 脂 肪 の 厚 さ の 平 均" = ";AVBF
1150 INPUT "背 脂 肪 の 厚 さ の 標 準 偏 差" = ";SDBF
1160 INPUT "相 関 係 数" = ";RDB
1170 DIM X(NP,2)
1180 'デ ー タ の 発 生
1190 FOR I = 1 TO NP

```

```

1200 ' -----
1210 CALL *CNRND (RDB, X1, X2)
1220 ' -----
1230 X(I,1) = FNNRM (X1, AVDG, SDDG)
1240 X(I,2) = FNNRM (X2, AVBF, SDBF)
1250 SX1 = SX1 + X(I,1)
1260 SX2 = SX2 + X(I,2)
1270 SSX1 = SSX1 + X(I,1) * X(I,1)
1280 SSX2 = SSX2 + X(I,2) * X(I,2)
1290 SS12 = SS12 + X(I,1) * X(I,2)
1300 NEXT I
1310 '基本統計量
1320 AVX1 = SX1 / NP
1330 AVX2 = SX2 / NP
1340 SDX1 = FNSD (NP, SX1, SSX1)
1350 SDX2 = FNSD (NP, SX2, SSX2)
1360 RX12 = (SS12 - SX1 * SX2 / NP) / (NP - 1) / (SDX1 * SDX2)
1370 PRINT "★★ 集団のパラメータ ★★"
1380 PRINT USING "N #####";NP
1390 PRINT "平均 標準偏差"
1400 PRINT USING "DG ####.## ####.##";AVX1,SDX1
1410 PRINT USING "BF ####.## ####.##";AVX2,SDX2
1420 PRINT "相関係数";
1430 PRINT USING " ##.###";RX12
1440 'D G に対する選抜
1450 ' -----
1460 CALL *XSORT (NP, X())
1470 ' -----
1480 NP2 = INT (NP / 2)
1490 SX1 = 0 : SX2 = 0
1500 FOR I = 1 TO NP2
1510 SX1 = SX1 + X(I,1)
1520 SX2 = SX2 + X(I,2)
1530 NEXT I
1540 SLX1 = SX1 / NP2
1550 SLX2 = SX2 / NP2
1560 PRINT "☆☆ 選抜群平均 ☆☆"
1570 PRINT USING "N #####";NP2
1580 PRINT USING "DG ####.##";SLX1
1590 PRINT USING "BF ####.##";SLX2
1600 END
1610 '
1620 ' -----
1630 SUB *CNRND (R, X1, X2)
1640 '2 変量正規乱数の発生
1650 ' -----
1660 U1 = SQR (-2 * LOG(RND))
1670 U2 = 6.28318 * RND
1680 Z1 = U1 * COS (U2)
1690 Z2 = U1 * SIN (U2)
1700 X1 = Z1
1710 X2 = R * Z1 + SQR (1 - R * R) * Z2
1720 END SUB
1730 '
1740 ' -----
1750 SUB *XSORT (N, X())
1760 '降順ソート
1770 ' -----

```

```

1780 FOR I = 1 TO N-1
1790   II = I
1800   FOR J=I+1 TO N
1810     IF X(J,1) > X(II,1)      THEN      II = J
1820   NEXT J
1830   IF II > I THEN
1840     SWAP X(I,1), X(II,1)
1850     SWAP X(I,2), X(II,2)
1860   END IF
1870 NEXT I
1880 END SUB

```

```

雌豚の数              = ? 100
1 日平均増体重の平均  = ? 700
1 日平均増体重の標準偏差 = ? 80
背脂肪の厚さの平均    = ? 1.6
背脂肪の厚さの標準偏差 = ? .2
相関係数              = ? .2
★★ 集団のパラメータ ★★
N          100
          平均      標準偏差
DG  697.33      77.30
BF   1.58       0.18
      相関係数      0.262
☆☆ 選抜群平均  ☆☆
N          50
DG  758.28
BF   1.62

```

図60 演習問題13のリストと結果

```

1 'save "enshu.14"
1010 REM ---> 混合培養における酵母菌の生長モデル
1020 *START
1030 INPUT "酵母菌の量(1,2)      = ";Y1,Y2
1040 INPUT "相対生長率(1,2)     = ";RGR1,RGR2
1050 INPUT "アルコール生成係数(1,2) = ";ALPF1,ALPF2
1060 INPUT "最大アルコール量(1,2) = ";MALC1,MALC2
1070 INPUT "単位時間数         = ";NTM
1080 INPUT "時間分割数         = ";NDT
1090 INPUT "出力間隔           = ";NPR
1100 DT = 1 / NDT
1110 ALC = 0
1120 RED1 = 0 : RED2 = 0
1130 PRINT "時間      酵母菌 1      酵母菌 2      アルコール"
1140 HF$ = "####      ###.##      ###.##      ###.##"
1150 FOR IT=1 TO NTM
1160   DO REPEAT NDT
1170     RY1 = Y1 * RGR1 * (1 - RED1) * DT
1180     RY2 = Y2 * RGR2 * (1 - RED2) * DT
1190     Y1 = Y1 + RY1
1200     Y2 = Y2 + RY2
1210     ALPD = ALPF1 * RY1 + ALPF2 * RY2
1220     ALC = ALC + ALPD
1230     RED1 = ALC / MALC1

```



```

1240      RED2 = ALC      / MALC2
1250      LOOP
1260      IF (IT MOD NPR) = 0 THEN PRINT USING HF$;IT, Y1, Y2, ALC
1270 NEXT IT
1280 END

```

```

酵母菌の量(1,2)      = ? .45, .45
相対生長率(1,2)      = ? .049, .236
アルコール生成係数(1,2) = ? .270, .122
最大アルコール量(1,2) = ? .15, .15
単位時間数          = ? 50
時間分割数          = ? 10
出力間隔            = ? 5
時間      酵母菌1    酵母菌2    アルコール
  5         0.57      1.38      0.15
 10         0.69      3.57      0.45
 15         0.80      6.91      0.88
 20         0.85      9.68      1.23
 25         0.88     11.04      1.41
 30         0.89     11.54      1.47
 35         0.89     11.70      1.49
 40         0.89     11.75      1.50
 45         0.89     11.76      1.50
 50         0.89     11.77      1.50

```

図61 演習問題14のリストと結果

参考文献

- 1) 小坂清巳：畜産におけるパソコン入門. 18-45, 中央畜産会, 東京, 1987.
- 2) 石田晴久：パソコン言語学. アスキー出版局, 東京, 1984.
- 3) 土居範久, 笈 捷彦：百科*コンピュータの基礎知識. 岩波書店, 東京, 1987.
- 4) 石田晴久：マイコン BASIC 入門. 207-216, 日本放送出版協会, 1982.
- 5) 北郷達郎：どうなる, ぼくらの BASIC. 日経バイト, 56, 122-134, 1989(3).
- 6) BASIC/98 USER'S MANUAL. 神津システム設計事務所, 東京, 1989.
- 7) 大隅 昇, 秋山直樹：問題だらけの国産統計ソフトウェア. 日経バイト, 49, 225-239, 1988(8).
- 8) Jeffers, J. N. R.: An introduction to systems analysis: with ecological applications. Edward Arnold, London, 1978; 古在豊樹, 高倉 直, 仁科弘重 訳: 生態学のためのシステム分析入門. 28-52, 朝倉書店, 東京, 1982.