

# 学生のための物理シミュレーション演習システム

真 鍋 惇\*

## Physics Simulation System for Student Practice

Atsushi MANABE

**Abstract :** A system of physics simulation practice has been introduced in the introductory physics education of a College of Technology in Japan, in which students (age 19) can make their computer program by themselves including the numerical integration of equation of motion and the graphical presentation of the results. A graphic utility (MYGRAPH) is developed to make the student programming in class unit (40 students) easy and interesting.

The practice of student is designed to investigate the several phenomena by reproducing them in computer applying the related physical law and adapting the proper model for each phenomenon. The phenomena listed for the practice include the realistic one such as free fall under air resistance and the complicated one such as the non-linear pendulum showing chaos behavior.

The example of a practice is shown for case of the falling rain drop with various radius, in which the inertial air resistance is found dominant for almost all radius from pouring to drizzling rain and the viscous drag is found effective only for misty rain.

The educational effect of this practice system is discussed together with the description of student impressions and opinions in Appendix 1. The main part of the source code of MYGRAPH is given in Appendix 2.

**Keywords :** Physics Education, Student Programming, Computer Simulation, Graphic Utility, Air Resistance

### 1. はじめに

物理教育におけるコンピュータの利用は、実験計測への利用、講義や演習形式でのシミュレーションの演示などが比較的良く研究され、また実際の教育カリキュラムにおいて利用されている。これに対して、学生、生徒がプログラムを作成し物理現象のシミュレーション実験を行う演習は、パソコン室の整備によりハード的には可能となっているが、まだまだ普及していない。そのような教材や試行の報告がいくつかあり、例えば高校における報告<sup>1-3)</sup>では、生徒に簡単なプログラムを与え、その一部を変更または改良させる方法が採られている。大学の報告例としては、対象を物

理専攻の大学初年級学生とする物理プログラミングシステム、M.U.P.P.E.T. (Maryland University Project in Physics and Educational Technology)<sup>4)</sup>が注目される。これはPascal言語を用いた本格的なシミュレーションシステムであるが、学生がプログラムの一部を変更、改良することは同じである。その場合、学生、生徒にとってブラックボックスが多くなり、彼らが創意工夫を發揮し、達成感を抱くなどの教育効果の面で問題が残ると思われる。他方、力学など分野を限定した言語システムの開発<sup>5)</sup>の例は体験的な学習には有効だが、プログラム環境の改善が優先され、物理法則とその適用方法が見えなくなっている。

本稿では、工業高等専門学校<sup>6)</sup>の4学年の専門基礎科目の一つである「応用物理」の力学分野において、約40名からなるクラスの学生全員がパソコンを使用して、何らかの物理現象を解明するためのシミュレーション実験を行うために、自分の力と創意工夫によってプログラムを作成し、実験と改良を繰り返し、主体

(1998年9月24日受理)

\* 宇部工業高等専門学校、〒755-8555 宇部市常盤台  
Ube National College of Technology,  
E-Mail : manabe@ube-k.ac.jp

的に目的を達成していくような演習を行うために組み立て、5年間にわたって実施してきた教育システムについて、学生の感想も合わせて報告する。

## 2. 物理シミュレーション演習の目的

ここでは、この演習システムにおいて目標とした教育的目的について述べる。まず最初に検討した事は、今日の研究方法に、理論、実験、コンピュータシミュレーションの三つの方法があるように、教育、特に物理教育に、講義、実験、シミュレーション演習という三つの方法があっても良いのではないかと言うことである。第三の方法を導入することによって、物理教育の方法と内容の幅を広げ、新たな可能性を開くと期待されるであろう。この演習では、何らかの現実の現象をモデル化して、コンピュータの中に再現し、現象の生起する原因やメカニズムを明らかにし、さらには新たな現象を予測するという課題に対して、学生一人一人が自分の頭と手を使って取り組むものとする。ここでは物理法則を応用する自分のアイデアを発展させつつ、何かを創造して画面に表示するという、試行の連続に伴う楽しみと喜びのある場を作り得る。すなわち、問題解決に向けての学生の自己決定能力と、創意工夫を引き出す教育方法としての物理シミュレーション演習を実現することを目的とした。

次にここで扱う力学の教育内容の面では数値計算の手法により、従来の解析的に解ける公式的な現象から離れて、より現実的な現象や、現代物理的な現象を扱える。このことは、物理教科を学生にとってより興味ある科目に変えることに繋がると期待されるが、実際の演習授業でもそのことを実感してきた。すなわち、現実的な現象や最近のトピックスなどコンピュータを利用しないと扱えない問題を取り上げ、学生の興味と、問題意識を引き出すことを物理シミュレーション演習のもう一つの目的とした。

以上のような教育的目標をもつ物理シミュレーション演習の実現のためには、プログラム作成のためのシンプルで高機能なシステムを構成し、学生達に提供することが必須である。すなわち、適切な演習テーマの選定はもちろん、数値計算のアルゴリズム、プログラム言語、グラフィック手法などの検討が必要となる。

## 3. 演習システムの条件

従来、学生、生徒にまず初期のプログラムを与え、その一部を変更または改良させる方法が採られていた

のは現在のプログラム環境のままでは、学生、生徒のプログラム作成能力を要求できないからである。しかし、プログラム環境の改善を優先するあまり、物理法則とその適用方法が見えなくなってしまうのは困る。ここでは、プログラム環境を改善しつつも、学生自身の手によるプログラミングを可能にするシステムを追求した。そこでは、プログラム文法の詳細にあまり煩わされることなく、物理の現象と法則を考えて、数値計算を計画し、計算結果を容易に高機能にグラフ化、視覚化出来る必要がある。そのような目的で、まず初心者向けの扱いやすい言語である BASIC 言語を選び、さらに統合開発環境を有し、構造化された BASIC 言語である QuickBASIC (マイクロソフト社) を選んだ。ただし、実際に演習で使用する固有の命令語の種類は最大限少なくし、グラフィックはその構造化機能を利用して作成した後述のグラフィックライブラリを使用してプログラミングするようにした。こうして、①プログラムは長文にならず、易しく簡単であり、②計算結果を、容易に高機能にグラフ化、視覚化出来るようにした。さらに、後述するように、適切な演習テーマの選定や数値計算のアルゴリズムの選定などの検討と合わせて、ブラックボックスを含まず、受講生がアイデアを容易に表現して、手作りのプログラミングを実感できる演習システムを構築してきた。クラス単位の演習を行うためには高価なソフトは使用できないが、その面でも比較的安価であり、コンパクトで扱いやすいものを実現できたと考えている。

## 4. 演習システムの要素と構成

筆者が現在実施している演習システムの概要を表 1 に示す。

受講学生は、高専4年生(19歳)で、40名編成の3クラス(機械、化学、電気または制御情報工学科)に年間約12週に渡って、2テーマの演習を実施してきた。学科によって、それまでに受講した情報処理教育はプログラム言語の種類や時間数においてかなり異なるが、物理の演習としては、そのことにあまり影響されず進行できた。これは採用した BASIC 言語の固有の命令の種類ををごく少数に限定したからである。

演習テーマは、リアリスティックなものやアドバンスドなものを組み合わせた。前者は現実の自由落下現象を空気抵抗を考慮して解明するものである。この現象はいろいろあるので、表1に示すように、学生が自由に課題を選択することとした。そのうちの雨滴の落

表 1 演習システムの概要

- A) 実施対象学生  
質点の運動方程式について受講済みの学生  
機械、電気、制御情報、物質の各科の4年生
- B) 実施期間  
年間： 90分 × 6週 × 2テーマ
- C) シミュレーションのテーマ  
1部. REALISTIC: 自由落下と空気抵抗  
以下の課題から学生が自由に選択する  
\*種々の粒径の雨滴の落下  
\*スカイダイビング  
\*打球の軌跡  
\*オイラー法の誤差と各種の修正法  
\*その他: 学生が提案する課題  
2部. ADVANCED: 複雑系への入門  
クラスにより課題を選定する  
\*非線型強制振子とカオス現象の観察  
\*生体個体数モデル (ロジスティック方程式) に見るカオスとフラクタル
- D) 数値計算のアルゴリズム  
運動方程式の数値積分  
オイラー法または修正オイラー法 (樽掛け法)
- E) プログラム言語 (MSDOS)  
構造化 BASIC と統合開発環境を使用する  
(Microsoft QuickBASIC Ver.4.5)  
使用する固有の命令語は10種以内とする
- F) グラフィックツール  
QuickBASIC 用のライブラリー: MYGRAPH  
を開発し使用する。
- G) 演習テキスト  
演習の目標と方法、オイラー法による運動方程式の解法、空気抵抗の理論、非線型振り子の運動方程式とカオスについて、QuickBASIC とマイグラフの使用法、などを記述し配布。
- H) レポートについて  
演習には到達すべき一定のゴールは定めず、レポートは各学生の進捗、進行方向に応じて、各自独自の内容で作成することとする。

を開いての地上への着地速度の典型的なデータを調べて、それをコンピュータ内に再現することにより、空気抵抗の作用を明らかにし、条件を変えたダイビングをシミュレートする。打球の軌跡問題は二次元の問題となりやや高度となるが、スポーツに関心のある学生がよく選択する。その他のテーマとしては、火山弾と火山灰の落下、いん石の落下など空気密度の高度依存性などを含む問題に発展させる。第二のアドバンスドなテーマは、最近話題の複雑系物理への入門とする。機械・電気系学科のクラスでは、振れ角を微小と近似しない非線形振子に減衰項と周期的な強制力項を含む運動方程式を扱う。実空間のアニメーションと状態空間 (位相空間) での軌道を描き、強制力の振幅を変化させると、周期が倍々と長くなる過程を通してカオス運動となることを観察し、少数自由度の決定論的運動方程式から予測不可能な不規則運動が生じることを確認させる。状態空間のポアンカレ断面図も調べる。物質工学科の学生には、生体個体数のモデルであるロジスティック差分方程式に見られるカオスとフラクタルのテーマを採用している。

運動方程式の数値積分のプログラム化は、学生にとって難解であることが毎年の演習で認識させられる。これを理解容易なアルゴリズムで行うことはこの演習のキーポイントの一つである。最も簡単なオイラー法を採用すべきで、各種のやや複雑な修正オイラー法<sup>6)</sup> やルンゲクッタ法の使用を初めから求めることは適当でない。自由落下の問題では、通常のオイラー法で間に合う。非線型振り子の場合は、加速度が位置の関数となり、長時間にわたるカオスの観測には計算の精度を要するので、次に示す最も簡単な修正オイラー法を用いる。

$$V(T+dT) = V(T) + A(T, Y, V) \times dT$$

$$Y(T+dT) = Y(T) + V(T+dT) \times dT$$

まず現在の加速度AからdT秒後の速度Vを求め、そのVを使ってdT秒後の位置Yを求める。これは、樽掛け法とも言うべき方法で、オイラー・クロマー法とも呼ばれている<sup>7)</sup>。オイラー法の修正としては最も簡単であり学生の理解も容易であり、その誤差減少の効果は大きく、学生演習には充分である。

次にプログラム言語は、やはり BASIC 言語が学生にとって最も理解し易い。なかでも統合開発環境型の BASIC はファイルの保存、呼び出し、編集、実行などがプルダウンメニュー化され、また構文チェック、オンラインヘルプ機能などもあり、これらがマウス操

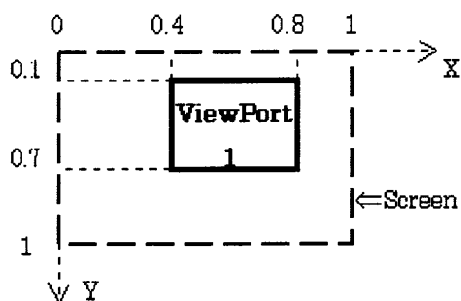
下では、種々の半径の雨滴の終端速度の測定データを与え、それをシミュレーションで再現することによって、雨の降り方のメカニズムを解明させる。霧雨、細雨と雷雨など大粒の雨では主因となる空気抵抗のメカニズムが異なることを発見する。スカイダイビングでは、ダイバーの終端速度、パラシュート

表 2 マイグラフの主なコマンド

- 1) DefineViewPort *Vnum, X1, X2, Y1, Y2*
- 2) DefineScale *Snum, Xmin, Xmax, Ymin, Ymax*
- 3) OpenViewPort *Vnum, Color, BorderColor*
- 4) SelectScale *Snum*
- 5) AutoAxis *Vnum, Snum*
- 6) PutLabels *A\$, B\$, C\$, Color*
- 7) PlotDataPoint *Form\$, Color, X, Y*
- 8) DispData *X, Y, Label\$, Data*

(注) : イタリックは引数。\$ 付きは文字列引数  
*Color* は white, red, green など色名を指定  
*num* の付いた引数は 1 ~ 8 の整数値をとる。

DefineViewPort 1, 0.4, 0.8, 0.1, 0.7



DefineScale 1, 0, 100, -5, 5

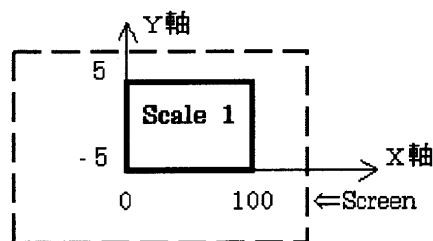


図 1 マイグラフにおけるビューポートとスケールの設定例

作で扱えるので学生演習に適している。また構造化機能とライブラリ作成機能をもつ BASIC は、適当なライブラリを作成し、それを利用させることにより学生が作成するプログラムを短く出来る。筆者はマイクロソフト社の QuickBASIC Ver.4.5 (MSDOS 用) を使っている。このソフトは主なパソコン機種に対応している。

```
' $INCLUDE : 'MYGRAPH93.BI' (ライブラリの指定)
CLS (グラフィックの仕様と作成)
DefineViewPort 1, .2, .8, .2, .7
DefineScale 1, 0, 6, 0, 120
OpenViewPort 1, blue, black
AutoAxis 1, 1
PutLabels "高さ(M)", "真空中落下", "SEC", red
G = - 9.81
Y = 100 (初期条件 高さ (m))
V = 0 (初速度 (m/s))
T = 0 (時刻 (s))
DT = .005 (時間刻み (s))
DO
V = V + G * DT (オイラー法)
Y = Y + V * DT
T = T + DT (計算値のプロット)
PlotDataPoint "DOT-L", green, T, Y
LOOP WHILE Y > 0
END
```

図 2 真空中自由落下のプログラム。最初の練習として作成するプログラムの例。  
 ( ) 内は説明文。

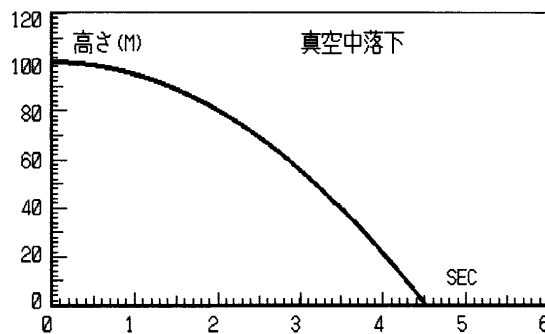


図 3 図 2 のプログラムの実行画面の白黒ハードコピー (プリンター出力)

計算結果を視覚化したり、グラフに表現するためグラフィックは必須であり、これ無くしてはシミュレーションが面白くならない。しかし、パソコンのグラフィックでは 640 × 400 などの画面のドット (ピクセル) を扱うこと、縦の軸 (Y 軸) 座標が上から下への方向を向いているなど、その命令機構は複雑で教育向きでない。計算結果をファイルに保存して、別のグラフィックソフトでグラフ化するのは、計算の進行に合わせて実時間で仕事が出来ない。

```

$INCLUDE : 'MYGRAPH93.BI'
CLS
DefineViewPort 1, .2, .75, .1, .4
DefineScale 1, 0, 5, 0, 120
OpenViewPort 1, blue, black
AutoAxis 1, 1
PutLabels "Y (m)", "真空中落下", "SEC", red

DefineViewPort 2, .2, .75, .5, .8
DefineScale 2, 0, 5, 0, 60
OpenViewPort 2, blue, black
AutoAxis 2, 2
PutLabels "- V (m/sec)", "落下速度", "SEC", red

DefineViewPort 3, .01, .1, .1, .8
DefineScale 3, 0, 1, 0, 120
OpenViewPort 3, black, yellow

G = - 9.81
Y = 100
V = 0
T = 0 : DT = .005

DO
  V = V + G * DT
  Y = Y + V * DT
  T = T + DT

  SelectViewPort 1
  SelectScale 1
  PlotDataPoint "Dot-L", green, T, Y

  SelectViewPort 2
  SelectScale 2
  PlotDataPoint "Dot", yellow, T, -V

  SelectViewPort 3
  SelectScale 3
  PlotDataPoint "Scircle-L", black, .5, Yold
  PlotDataPoint "Scircle-L", green, .5, Y
  Yold = Y
LOOP WHILE Y > 0
END
    
```

図 4 真空中自由落下 のプログラム。  
 三個のビューポートに落下物体の  
 高さ、速度、アニメーションを描く。

先述した MUPPET では PASCAL 言語を採用し  
 様々なツールを開発している。筆者はその中のグラ  
 フィック手法を参考にして、BASIC 用のグラフィッ  
 クライブラリーを作成してマイグラフ (MYGRAPH)  
 と名づけている。これは、時間、空間や座標軸を駆

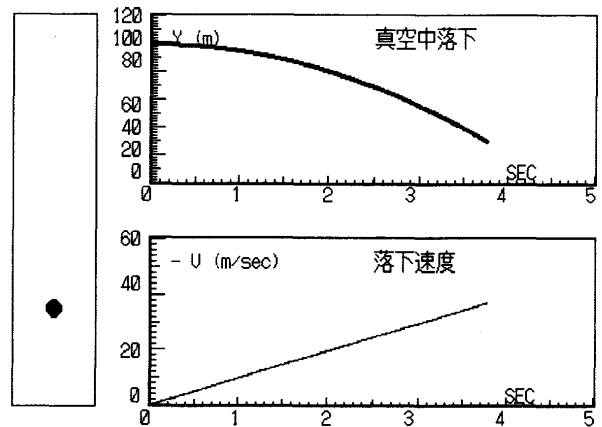


図 5 図 4 のプログラムの実行中  
 の画面のプリンター出力。

使用する物理や科学計算にマッチするように作成した  
 もので、これによってプログラムが簡単化、短文化  
 され、この演習の実施が可能になったといえる。

マイグラフは、パソコン画面を管理し、二次元空  
 間またはグラフを画面に最大 8 個まで描くための手  
 続き (プロシージャ) をライブラリー化したもので、  
 Quick BASIC と同時に起動する。マイグラフには、  
 この他にメニュー作成 (選択メニュー、データ入力  
 メニュー) など約 20 種のコマンドを含むが、学生  
 演習に必要なのは表 2 に示した 8 種の主なコマンド  
 である。

マイグラフでは、図 1 に示すように、パソコン画  
 面を縦も横も幅 1 に規格化し、その中に最大 8 個ま  
 での窓 (ビューポート) を設定し、各々に番号を付け  
 る。一方、各々の窓の縦と横方向の座標 (スケール) を通  
 常の形式で最大 8 種まで設定でき、各々に番号を付け  
 る。すべてのグラフィック命令は、番号で指定された  
 窓に対し、番号で指定された座標系を採用して実行さ  
 れる。AutoAxis により座標軸と目盛値は自動的に描  
 くことが出来る。データ点を様々な印で描くために  
 PlotDataPoint の引数 (表 2 の Form\$) を使う。マ  
 イグラフのソースコードの主要部を付録 2 に示す。

実際の演習では、まず Quick BASIC 環境の使用法  
 とマイグラフの使用法の説明を兼ねた簡単な練習を行  
 う。次に学生になじみのある真空中の自由落下を取り  
 上げ、プログラムを作製する。そのプログラムは図 2  
 に示すように簡単で短いものである。その実行結果は  
 図 3 であるが、学生たちの満足を得られる程度に出  
 来上がっている。ここでは、プリンターに出力した画  
 面のハードコピーを示し、カラー表示を白黒に変換し  
 て示している。次にこのプログラムを拡張して、三個  
 のビューポートを使用し、速度、アニメーション画面

表 3 雨滴半径と終端速度の測定値<sup>b)</sup>と  
シミュレーションで求められる抵抗係数など

雨滴半径 $r$ mm	3.25	3.0	2.4	1.4	1.0	0.5	0.2	0.1
終端速度 $V_t$ m/s	8.9	8.8	8.3	6.8	6.3	4.0	1.6	0.7
抵抗係数 $Cd^a)$	0.83	0.79	0.71	0.61	0.51	0.63	1.6	4.0
抵抗係数 $Cd^b)$	0.83	0.78	0.70	0.60	0.48	0.55	1.1	1.8
$K_1 V_t / K_2 V_t^2$	0.00065	0.0076	0.0011	0.0027	0.051	0.14	0.45	1.2
レイノルズ数 $R$	2200	2000	1500	730	490	150	25	5.5

(注) 二つの  $Cd$  値は、各々 a)  $Fd=K_2V^2$  b)  $Fd=K_2V^2-K_1V$  の場合に計算値を測定値に一致させる値。

$0^\circ\text{C}$  1 気圧の空気密度  $\rho=1.29\text{ Kg/m}^3$ , 粘性率  $\eta=1.68 \times 10^{-5}\text{ Kg/m}\cdot\text{s}$  を使用。

レイノルズ数は、 $R=rV_t \rho/\eta$  より計算。

を追加したプログラムを作成させる。そのプログラム例を図 4 に、その実行画面を図 5 に示す。プログラムは図 2 の約 2 倍の長さとなる。学生たちは、これらのプログラムを教官のガイドを受けながら作成していく中で、その方法を理解し、以後のテーマに取り組む中で使っていくことになる。

指導上の問題として、図 2、図 4 のプログラムの作製指導、ガイドの方法がある。初めからサンプルとして与えると、それをキーボードから打ち込む作業になりがちで、その場合には後の進行のための力がつかないことがある。配布テキストを参照しつつ、はじめから自分で作ることにすると、一部の学生しか出来ないことが多い。そこで、不定部分を含むプログラムの骨組みを示し、それを学生が完全なものにして行く過程で適当なガイドを加える方法が良いようである。しかし、そのあたりの設定や配分は微妙であり、前年度のやり方が今年度にもそのまま通用するとは限らない。その理由としては、年度を経るに従って変化する学生気質や感覚の違い、および教官側の慣れや意気込みの違いなどが作用していると感じられ、問題解決に自信を持って立ち向かう流れを、演習の初期に作る配慮が肝要であると実感している。

#### 4. 雨滴の落下の演習例

自由落下と空気抵抗の演習の例として、雨滴の落下のテーマによる演習の内容とその進行過程について述べる。このテーマを選んだ学生は、種々の雨滴半径(雨滴を球体として、その質量から求めた値)とその終端速度の測定値のデータを与えられ、それらを再現するシミュレーションを試みることによ

て、雨がどのように降るかを考えていくことになる。そのデータは表 3 に示す。このデータについては、文献<sup>5)</sup>が少なく、得られる値はややばらついている。雨滴の落下運動と終端速度を再現するために、半径  $r$ 、質量  $m$  の球体が、密度  $\rho$ 、粘性率  $\eta$  の空気の抵抗  $F_d(V)$  を受けて落下するときの運動方程式を数値積分する。後述のように、慣性と粘性の両方が効く領域があり、数値積分は避けられない。

$$dY/dT = V,$$

$$m dV/dT = -mg + F_d(V).$$

低速で落下する場合の空気抵抗、すなわち速度に比例する粘性抵抗は、球体の場合、ストークスの法則より

$$F_d(V) = -K_1 V, \quad K_1 = 6\pi\eta r.$$

高速では、速度の二乗に比例する慣性抵抗となり、

$$F_d(V) = K_2 V^2, \quad K_2 = 0.5\pi C_d \rho r^2.$$

ここで  $C_d$  は抵抗係数で、滑らかな表面の球体では、約 0.5 の値をとる。プログラムは、高所で生成されたある半径の球状の雨滴の速度と位置の時間変化と、落下の様子アニメーションを画面に表示するように作成することになる。図 6 はその実行画面、図 7 はプログラムの一例で、落下する雨滴のアニメーションと雨滴の位置と速度の時間変化グラフを描いている。プログラムを実行すると、粘性抵抗のモデルでは、現象を説明できないことがすぐに明らかになる。多くの物理教科書では、雨滴の落下は粘性抵抗に支配されるとしているが、これは誤りで現実の雨滴で支配的なのは慣性抵抗である。慣性抵抗を採用して、終端速度の測定値を再現するような抵抗係数を求めると表 3 の  $Cd^a)$  のようになる。

ここから先は学生の考察課題へつながるが、まず半

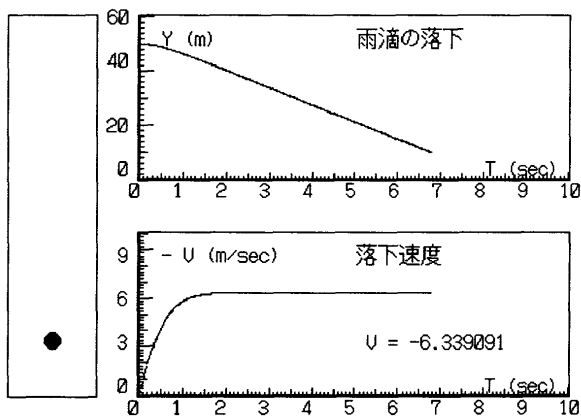


図 6 雨滴の落下プログラムの実行中の画面。  
雨滴半径 1 mm Cd=0.48 の場合。

径 1 mm 程度の雨滴は、Cd 値が約 0.5 で球に近いと言える。半径がそれより大きくなると共に Cd 値が増大しているのは、球形からの変形によると推定できる。また半径が 200 μm 以下の霧雨の抵抗係数は異常に大きい。これは落下速度が遅くなり、慣性抵抗に加えて粘性抵抗も効いてくると考えられ、両抵抗を加えて数値積分すると Cd<sup>b)</sup> となり、Cd 値の異常はかなり緩和され、粘性抵抗が効いていることが分かる。このとき粘性抵抗と慣性抵抗の抵抗力の比は 0.45 (半径 200 μm)、1.2 (半径 100 μm) となる。レイノルズ数も雨滴半径と共に大きく変化している。しかし、粘性抵抗を考慮しても、霧雨の Cd 値はまだ不自然に大きく、ストークス項の補正、空気定数の気温依存性、終端速度の測定値の誤差の考察なども可能であるが、原因を確定するには至らず、オープンな問題として残している。

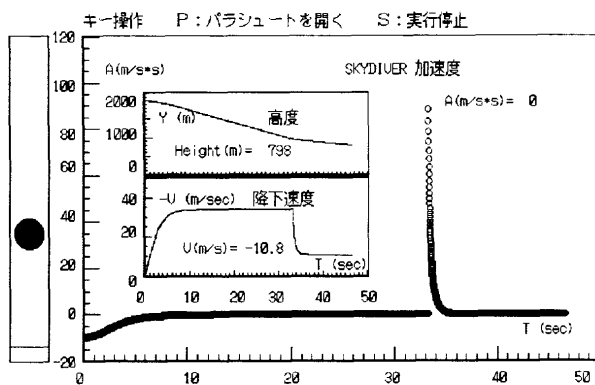


図 8 スカイダイビングのプログラムの実行例。  
高度 2000 m、体重 60 Kg  
パラシュートによって空気抵抗を  
10 倍にしている。

```

$INCLUDE : 'MYGRAPH93.BI'
DefineViewPort 1, .2, .7, .1, .4
DefineScale 1, 0, 10, 0, 60
OpenViewPort 1, blue, black
AutoAxis 1, 1
PutLabels "Y (m)", "雨滴の落下", "T (sec)", yellow

DefineViewPort 2, .2, .7, .5, .8
DefineScale 2, 0, 10, 0, 10
OpenViewPort 2, blue, black
AutoAxis 2, 2
PutLabels "- V (m/sec)", "落下速度", "T (sec)", red

DefineViewPort 3, .05, .15, .1, .8
DefineScale 3, 0, 1, 0, 70
OpenViewPort 3, black, yellow

PI = 3.1416 : G = - 9.81
WaterDensity = 1000 : AirDensity = 1.29
AirViscosity = .0000168
Radius = .001      ' 雨滴半径 (m)
Cd = .48          ' 抵抗係数
Mass = WaterDensity * Radius ^ 3 * PI * 4 / 3
K1 = 6 * PI * AirViscosity * Radius / Mass
K2 = .5 * PI * Cd * AirDensity * Radius ^ 2 / Mass
T = 0 : DT = .01 : Y = 50 : V = 0
DO
    A = G - K1 * V + K2 * V * V
    V = V + A * DT
    Y = Y + V * DT
    T = T + DT

SelectViewPort 1 : SelectScale 1
PlotDataPoint "Dot", green, T, Y

SelectViewPort 2 : SelectScale 2
PlotDataPoint "Dot", yellow, T, - V
DispData 7, 5, "V = ", V

SelectViewPort 3 : SelectScale 3
PlotDataPoint "Scircle-L", black, .5, Yold
PlotDataPoint "Scircle-L", green, .5, Y
Yold = Y

LOOP WHILE Y > 0
END
    
```

図 7 雨滴の落下プログラムの例

スカイダイビングのテーマでは、体表面積、抵抗係数などを調整して実際に近い状況を再現する。図 8 の例では、実行中にキーボード操作によってパラシュートを開き、加速度の変化を図示している。

## 5. 演習システムの評価と教育効果

この演習の実施においては、キーワードとして、“Do It Yourself” を学生に周知させた。指示された通りに進めるものであってはならず、マニュアルやサンプルを模倣するものでも困る。それらを期待する安易な気持ちとの葛藤を期待し、授業を活性化する演習にしたい。このような観点で、はじめは、おそるおそる導入した。結果は、多くの学生達から良い反応を得、毎年続くことになった。付録 1 に示すように、アンケートに寄せられた学生達の感想、意見や要望には、積極的な評価と達成感が読みとられる。もちろん、否定的な感想や意見も少数あるが、指導法の更なる改善で克服すべきものと考えている。

また、この演習は、物理概念や方法の教育と共に様々な能力の育成に有効であると考えている。すなわち、①注意力、②自ら誤りを見つけ、それを正す能力、③物事を自分で判断する力、④課題に取り組み解決していく姿勢、⑤自分にとってオリジナルな物を何か作り上げる創造性などである。これは、「物理シミュレーション演習」の特長として、原因(INPUT)と結果(OUTPUT)の因果関係を学生一人一人が明確にできること、物理の内容を柔軟に変化で

き幅が広いこと(従来型実験は内容が RIGID)、すなわち、実物ではないがそれ故に自在であり、自由度が高く個人的に行える試行錯誤型の創作学習であることが挙げられよう。このような演習は、特に大学の理工系の初年級や工業高専学生に適した一つの有効な教育方法と考える。

今後の課題として、実験とシミュレーション演習を結合して実施することや、演習室に設備されるパソコンの高機能化にともなって、例えば WINDOW 対応のプログラム言語を使用した演習の方法を検討している。

## 参 考 文 献

- 1)御須 利：物理教育 41-3 (1993) 316.
- 2)佐々木淳：物理教育 42-2 (1994) 182.
- 3)大塚信之「物理シミュレーション実験」：私信
- 4)E.F.Redish and J.M.Wilson：Am. J. Phys. 61-3 (1993) 222.
- 5)風間裕、金岩由美子：応用物理教育 15-1 (1991) 30.
- 6)例えば、a) 平田邦男著「新BAS I Cによる物理」共立出版 b) 吉沢純夫著「パソコンで見る物理シミュレーション」森北出版
- 7)H.Gould and J.Tobochnik“Computer Simulation Method” Part 1 p38 Addison-Wesley 1987.
- 8)例えば、a) A.A.Bartlett:Physics Teacher 27-5 (1989) 364. およびその引用文献. b)ブリタニカ国際大百科事典 1 (1972) 343.

## 付録 1. 学生の感想、意見、要望

(1997 年度 演習第一部実施後のアンケートより)

### ① この演習に参加して面白く感じたこと

この演習は面白いし、問題に対して全員が取り組んでいるように思えて、大変よいと思う。

物理法則について理解しやすい上、何と言っても自分のプログラムで思った通りの結果が出せたときの充実感がいい。

演習はとても面白いです。教室でする授業は聞いていることが多いので、時にはいねむりをしてしまうことがありますが、パソコンを使うことによって自主性があり、とても意欲的に授業を受けています。

数値だけでなく、視覚的に物体の動きをとらることができるので面白い。

自分のプログラムが出来たとき、特にそれが図で表されたときはうれしかった。

普通の授業に比べ、開放感があり、学生同士で意見を交換できる。

プログラムをうまく作れば、シミュレーションがちゃんと出来て、それを見るのがすごく面白いと感じた。

プログラムの意味がある程度理解できるようになると、自分の意志で落下の速度やシミュレーションの範囲を変更できるようになっていくところが面白かった。

打球のシミュレーションを行って、各成分の速度を同時に見ることが出来た。それで各方向の速度はまったく違うことが分かって面白かった。

自分でプログラムを組んでいくので、完成するのが楽しかった。

自分の考える結果がすぐその場で見れるので良かった。

各自が自分のペースで作業が出来ることと、出来たときの喜びが大きいこと

自分で考えたプログラムでうまくシミュレーションが出来たときが一番面白い。



野球部員の僕にとって、ボールの軌道のシミュレーションは非常に興味ある課題だった。

コンピュータによるシミュレーションというものを初めて行ったが、自分の立てた式の計算結果がグラフィック関数を使って実際に表示されるのでとても面白い。色も豊富であるし、自分なりにアレンジしたものになるのでとても面白い。空気抵抗など、細かい事まで考えなければならぬという事が面白い。

動的に知ることが出来、条件を変えることにより運動がどう変わるかが、簡単に調べられること。

講義での物理は、数式、理論を扱うことがほとんどだが、シミュレーションではその数式や理論が実際にはどのような現象を起こすかが感覚的によく分かるので面白い。

自由課題で自分の好きなことを追求できるところが面白い。自分で考えた運動方程式が、すぐに、どのような動きがするかわかるのが、うれしかったし面白かった。

自分の思ったとうりになるかと言う期待感と、失敗するんじゃないかと言う気持ちがまざる瞬間がどきどきして面白いし、思った通りに実行できるとさらにうれしくて、つい隣の人に「見て見て」と言ってしまう。

実験をくり返したり、条件を変えたりして、たどりつく過程が楽しい。

今までの物理の授業は理論ばかり述べられて、たいくつなところがあつたが、各自が自分の課題を見つけ、パソコンで運動の様子を見ることで、より理解が深まっていると思う。だから今やっている授業はとてもいいんじゃないかなと思う。

「シミュレーションのプログラムを自分も作れるんだ!」と思えたことがとてもうれしかった。「この数値を変えるとどうなるか」と、ワクワクしながら行った。

グラフを作成したことはあつたけど、シミュレーションは始めて、やって見て面白く感じた。

実際に起こっている現象が、シミュレーションになって現れるようになったらこの現象が起こるのかと言うことが分かりやすいと思う。

まちがった入力をしたとき、なぜかと考えることが面白い。アニメーションができて楽しいし、分かりやすく面白。プログラムを作っていく、だんだんうまく行く過程が面白かった。

先生から生徒への一方的な授業方法では受け身で授業を受けるが、ここでは自分から先生へという流れの授業なので、そこが面白かった。

## ② この演習においてつまらなかったこと

プリンタが白黒のため、作成したグラフィックをそのまま出力できなかった。

プログラムが出来ないとき、人に聞いても分からないときはものすごく腹が立つ。

分からなくなったら、そこから抜け出すことがすごく困難であつた。

プログラムを作るのがちょっと難しいのと、その説明が分かりづらく、とっつきにくい点が悪くなかつたと思う。

物理なのにプログラムを考えなければいけないところ。

プログラム作成がむずかしく苦労したこと。

最初の頃は何をやっているかわからなかつた。

プログラムの組みかたが分からず、前に進めず、時間がむだになつたこと。

むずかしすぎる場所がある。(雨粒の径など)

計算結果がオーバーフローしたり、なかなか結果にたどりつけなかつた。

自由演習の内容がむずかしくてなかなか進まないこと。

自分自身の知識不足という面があり、演習を進展させていく事ができなかつたこと。

ある程度のところに行くまで、自分が何をしているか、どこを間違えているかが分かりにくい。

エラー等の原因がわからず、先に進めないとき。

やっていることがそれほど興味のわく内容でないこと。

作られたプログラムを単に打ち込むだけになつたところがあつたこと。

いきなりプログラムを作れと言われても、計算の仕方とか、命令の名前とか良く分からない。

オイラー法が良く分からなかつた。どういう事をしているのか理解できなければ、つまらないと思う。

式をあこれこれ入れてみるけど、その式がいまいち良く理解できていないため、ただただ入力すること。

## ③ このような教育法のメリットと思うこと

自由演習では、どこまで研究を進めるかは個人の自由となり、学生の自発性を重視する点がメリットである。

むずかしい法則の使い方など、普通の授業より楽しんで出来ること。

普通の授業では答えがはじめから用意されているようなものだが、この教育法は自分で答えを求めて存在できるものとし、またクラスメイトと話し合うことでいろんな見方が出来る。やる気のある学生と、やる気のある先生同志の授業である。

分からないところを自分で調べようという気になる。

式だけでなく、自分の目で見ることによってはじめて理解できることも多いと思う。

物理に興味がない人でも、コンピュータを扱うということで楽しく出来る。

スカイダイビングを選択したが、ダイバーがどのように落ちていくかが、話して聞かされるより良く理解できた。

教室で先生の書いたことをノートに写すような授業よりは、自分の考えをしっかりと持て、積極性を持てるんじゃないかと思う。

生徒の自主性を育てると思います。

計算ばかりの授業では理解しにくいのが、シミュレーションを使うと目で見て体で理解できる。

数値を変えることで、その現象の起こり方や、変化の違いを

試行錯誤できること。

自分が行った事に対して、具体的な結果が得られて内容を理解しやすい。

いろいろなプログラムを理解し、応用をきかせたりしてとても実験できない事をシミュレーションする事が出来る。

キーワードにも表れているように、誰かのまねをしても仕方のない方法であり、人まねばかりの高専生の自主性を育てるものだと思う。

一人一人の自主性がつくと思う。分からなければまず自分で考えるという習慣がつく。

自主的に物事を考え、実行することになること。

自由テーマでは、自分で考え、学習を進めていくことから、自ら考える力、創造力などの物理ではない面も学習できた。自分で自分のプログラムを改良することが出来、個性を出せる。

コンピュータを使うことで、物理だけでなくことが学べてよかった。

授業形式が自由で、自分のペースで作業でき、友達と相談も出来る点。

物体の動きを見ることで、物理における想像力がつく。

今までのありっただけの知識を引っ張り出してこなくてはいけないので、教科書を開き直して調べ、命令の意味を理解し、それを今やろうとする事柄に応用していかなければいけない。これらは、物事を進めていけるようにする基本だと思う。

分からないところを教え合えたりするところ。理解できればすごく楽しくなるし、興味も出てくる。自分からやろうと言う気になる。

自分のペースで出来ること。授業だと理解していなくてもどんどん進んでしまうが、演習は自分で理解し、納得するまで同じ問題に取り組める。

生徒の側が、必ず考えるということが最大のメリットであると思う。普通の授業は「聞く・書く」といったことはするが、「考える」生徒はあまりいないと思う。また、先生から生徒へといった方針は、生徒に均一の力がつくが、能力は平均以上にはつかない。この方法は、個人の持つ力を最大まで伸ばせるのではないかと。

#### ④ このような教育法のデメリットとすること

せっかくの自由演習を自由時間とかんちがいして、やらない人が出てくる。

分からないことが多すぎる。

他人の作成したプログラムをそのまま利用する恐れがある。興味のない人間にとっては、普通の授業の方がいいかもしれない。

分からない人がどんどん置いていかれるところ。

プログラムを作ることに時間を取られすぎる。

実験をコンピュータで済ましてしまうところ。

プログラム作成の勉強からしないと分からないので、時間がかかる点。

一つの事に時間をかけるので、他の分野の知識がとぼしくなるところ。

物理の学力が不足しているのに、こういった応用の分野に入ると、さらに理解できなくて、やる気を失う人が出ると思う。

プログラムが苦手な人は、BASIC の演習となってしまう恐れがある。

自分では考えず、人のプログラムをそのままコピーするような感じの人もある。

#### ⑤ 演習に対する要望

自分で考えることが重要だとは思いますが、難しい式やプログラムの方法をもっと説明すべきだと思う。

簡単なもの、難しいもの、さまざまな種類の実験を用意してみる。人にはそれぞれの能力があるから、自分に合ったものを選び、興味を持って演習できるのではないかと。

もっと色々なことをゆっくりやってほしい。

練習のときにも、もっと詳しく内容を説明すべき。

興味のあるものに対する人間の集中力は計り知れないので、学生の関心のわくような魅力ある課題を多く用意する事だと思う。

最初の説明のときにつまずいた人が多いようなので、関数の説明などはもっと詳しくやった方がよい。

家にパソコンを持っている人と持っていない人の差は大きいので、後期すべてを演習にするくらいの気持ちで、一つの自由課題をもっと“こったもの”にするようにしてほしい。

もう少し、プログラムに対する講義をすれば、何も理解できないことは免れる。

ある程度くわしく講義した上でやったほうがよい。

自由課題について、公式の適用例などを増やして、納得しやすいテキストを。

自由演習の前に、もう少し練習タイムが欲しかった。

不得意な人に強制的にやらせるともったいないになる。分からない人には、もっと分かりやすく、才能ある人と同じ教え方では、そこでもう差がついてしまう。

プログラムの作り方をもう少し教えてほしい。

課題をする前に、もう少しむずかしい練習をしたい。

プログラムする前に、どのような現象、運動かを、実際にやってみたり、見たりしてからパソコンを用いていく。

式の意味をもっと理解できるような工夫が必要だ。

もう少しプログラムの説明を詳しくしてほしい。落下とか昔習った事でも少し忘れていたので、説明してほしい。

## 付録 2. マイグラフのソースコードの主要部分 (MYGRAPH93)

### 2-a) Include File, MYGRAPH93.BI -----

```

DECLARE SUB DefineViewPort (Vnum AS INTEGER, VPX1!, VPX2!, VPY1!, VPY2!)
DECLARE SUB OpenViewPort (Vnum AS INTEGER, Col AS INTEGER, Border AS INTEGER)
DECLARE SUB SelectViewPort (Vnum AS INTEGER)
DECLARE SUB DefineScale (Snum AS INTEGER, sXmin!, sXmax!, sYmin!, sYmax!)
DECLARE SUB SelectScale (Snum AS INTEGER)
DECLARE SUB Axis (Xint!, Yint!, TicX!, TicY!, LabelSign AS INTEGER)
DECLARE SUB AutoAxis (Vnum AS INTEGER, Snum AS INTEGER)
DECLARE SUB PlotDataPoint (Form$, Col AS INTEGER, X AS SINGLE, Y AS SINGLE)
DECLARE SUB LabelY (X AS SINGLE, Y AS SINGLE, Label$)
DECLARE SUB LabelX (X AS SINGLE, Y AS SINGLE, Label$)
DECLARE SUB PutLabel (Place$, Title$, Col AS INTEGER)
DECLARE SUB PutLabels (LeftTop$, Top$, RightBottom$, Col AS INTEGER)
DECLARE SUB DispData (X!, Y!, Label$, Dat!)
DECLARE SUB WindowMessage (X%, Y%, ItemNum%, Title$, Message$(), FLAG%)
DECLARE SUB InputMenu (X%, Y%, ItemNum%, Title$, Message$(), Anstype$(), Ans$(), FLAG%)
DECLARE FUNCTION SelectMenu% (X%, Y%, ItemNum%, Title$, Message$())
DECLARE FUNCTION ROUND! (X AS SINGLE, Ord AS SINGLE)
DECLARE FUNCTION Order! (X AS SINGLE)
CONST REVERSE = 8 : CONST BLINK = 16
CONST black = 0 : CONST blue = 1 : CONST green = 2 : CONST cyan = 3
CONST red = 4 : CONST magenta = 5 : CONST yellow = 6 : CONST white = 7
CONST xpix = 639 ' x = 0 - 639 for PC9800,FMR, IBM
CONST ypix = 399 ' y = 0 - 399 IBM -- 349(EGA) 479(VGA)
CONST xtext = 80 ' column = 1 - 80
CONST ytext = 25 ' row = 1 - 25 IBM - 25(EGA) 30(VGA)
CONST aspect = 1 ' for CIRCLE command
    
```

### 2-b) MYGRAPH93.BAS -----

```

'*****
' Utility MYGRAPH ----- Source Program -----
' by Dr. Atsushi Manabe (Ube College of Technology, Japan)
' 1993 For PC9800,FMR, IBM compatible
'*****
'$INCLUDE: 'MYGRAPH93.BI'
'----- COMMON Variables, shared in all subs in MYGRAPH -----
DIM Vx1(8) AS INTEGER, Vx2(8) AS INTEGER, Vy1(8) AS INTEGER, Vy2(8) AS INTEGER
DIM Xmin(8), Xmax(8), Ymin(8), Ymax(8) '--- Scale Range
COMMON SHARED /MGP1/ Vx1() AS INTEGER, Vx2() AS INTEGER, Vy1() AS INTEGER, Vy2() AS INTEGER
COMMON SHARED /MGP2/ Xmin(), Xmax(), Ymin(), Ymax()
COMMON SHARED /MGP3/ ViewPortNumber AS INTEGER, ScaleNumber AS INTEGER
'----- shared in user program and MYGRAPH -- for MENU sub --
DIM SHARED mess$(1 TO 22), Anstype$(1 TO 20, -1 TO 10), Ans$(1 TO 22)

SUB DefineViewPort (Vnum AS INTEGER, VPX1, VPX2, VPY1, VPY2) STATIC
'Define the View Port Region.
'Vnum = 1 to 8 :View Port number
    
```

```

'VPX1 etc. = 0 to 1  VPX1<VPX2  VPY1<VPY2
  Vx1(Vnum) = CINT(VPX1 * xpix)
  Vx2(Vnum) = CINT(VPX2 * xpix)
  Vy1(Vnum) = CINT(VPY1 * ypix)
  Vy2(Vnum) = CINT(VPY2 * ypix)
END SUB

SUB DefineScale (Snum AS INTEGER, sXmin, sXmax, sYmin, sYmax) STATIC
'Define the Scale of number Snum : Max and min of Xaxis and Yaxis
  Xmin(Snum) = sXmin
  Xmax(Snum) = sXmax
  Ymin(Snum) = sYmin
  Ymax(Snum) = sYmax
END SUB

SUB OpenViewPort (Vnum AS INTEGER, Col AS INTEGER, Border AS INTEGER) STATIC
'Open the renewed View Port of Number Vnum
'Col is background color and Border is border color of the View Port
  ViewPortNumber = Vnum
  VIEW (Vx1(Vnum), Vy1(Vnum))-(Vx2(Vnum), Vy2(Vnum)), Col, Border
END SUB

SUB SelectViewPort (Vnum AS INTEGER) STATIC
'Select the specific View Port as active View Port
  ViewPortNumber = Vnum
  VIEW (Vx1(Vnum), Vy1(Vnum))-(Vx2(Vnum), Vy2(Vnum))
END SUB

SUB SelectScale (Snum AS INTEGER) STATIC
'Select the specific Scale as active scale in the active View Port.
  ScaleNumber = Snum
  WINDOW (Xmin(Snum), Ymin(Snum))-(Xmax(Snum), Ymax(Snum))
END SUB

SUB PlotDataPoint (Form$, Col AS INTEGER, X AS SINGLE, Y AS SINGLE) STATIC
'Plot data (x,y) with color Col on active viewport with active scale definition
'Plotting Symbols are given by Form$ as,
'  Dot Dot-L Dot-LL          (Ex: Dot, dot, DOT, d and D are all allowed)
'  OCircle OCircle-L OCircle-LL and SCircle SCircle-L SCircle-LL
'  OTriangle STriangle and Line BoxLine PaintLine
DIM R AS SINGLE
I = ScaleNumber
rp = 3          'radius in physical coord.
xp = PMAP(X, 0)
R = PMAP(xp + rp, 2) - X  'radius in screen coord.
SELECT CASE Form$
  CASE "dot", "Dot", "DOT", "D", "d"
    PSET (X, Y), Col
  CASE "dot-l", "Dot-L", "DOT-L", "D-L", "d-l"
    CIRCLE (X, Y), R / 2.5, Col, , , aspect
    PAINT (X, Y), Col
  CASE "ocircle", "Ocircle", "OCIRCLE", "OC", "oc"
    CIRCLE (X, Y), R, Col, , , aspect
  CASE "ocircle-l", "Ocircle-L", "OCIRCLE-L", "OC-L", "oc-l"
    CIRCLE (X, Y), R * 2, Col, , , aspect

```

```

CASE "scircle", "Scircle", "SCIRCLE", "SC", "sc"
    CIRCLE (X, Y), R, Col, , , aspect
    PAINT (X, Y), Col
    ----- (中 略 an omission of a middle part ) -----
CASE ELSE
END SELECT
END SUB

SUB Axis (Xint, Yint, TicX, TicY, LabelSign AS INTEGER) STATIC
' Xint is position of Y axis, Yint is that of X axis
' if Xint < Xmin or Xint > Xmax , Y axis is not drawn, same for X axis
' X-Tics are put on from Xmin to Xmax, Y also
' ABS(TicX) is tic interval
' TicX > 0 -- up tic    TicX < 0 -- down tic , TicY also
' LabelSign 0 -- small tic without label
'           1 -- large tic with label
'           2 -- medium tic without label
'           3 -- axis without tic and label
    ----- (中 略 an omission of a middle part ) -----
END SUB

SUB AutoAxis (Vnum AS INTEGER, Snum AS INTEGER)
' Automatic standard axis processor.
' This sub is constructed using Axis sub.
    ----- (中 略 an omission of a middle part ) -----
END SUB

SUB LabelX (X AS SINGLE, Y AS SINGLE, Label$) STATIC
' To display Label$ into active ViewPort with coordinate (x,y) of active scale.
' Label$ is put just under the point (x,y).
    ----- (中 略 an omission of a middle part ) -----
END SUB

SUB DispData (X!, Y!, Label$, Dat!)
' Use LabelX, COLOR is to be set outside
  Label$ = SPACE$(2) + Label$ + SPACE$(1) + STR$(Dat!) + SPACE$(2)
  LabelX X, Y, Label$
END SUB

----- (後 略 an omission of the rest ) -----

```