

KETpic の工学教育への利用と Maxima への移植

木更津工業高等専門学校・電子制御工学科 泉 源 (Izumi Hajime)

Department of Control Engineering,
Kisarazu National College of Technology

呉工業高等専門学校・機械工学科

深澤 謙次 (Fukazawa Kenji)

Department of Mechanical Engineering,
Kure National College of Technology

1 はじめに

本稿では、配布教材を作成するための方法として、数式処理ソフト (CAS:Computer Algebra System)+KETpic [1], [2] が有効であることを述べる。

本稿は、1) 教材を作る理由、2) 現状の教材作成と CAS+KETpic との比較、4) KETpic の Maxima への移植、5) まとめ、の構成とする。

教材を作成する理由の中に次の2点ある。1つ目は教員にとっての補助道具である。教材を作成することで、どのようにして教材を活かしながら講義を進めようか、というロジカルな整理ができる点があげられる。主役を引き立てるためには、どのような事を念頭に置けばよいのか、という考えに通ずる。

2つ目は、こちらの方が理由としては大きいと思うが、学生に興味を持ってもらう事である。教材を使った講義は、学生の中にも印象として残ることが多い。本来は、その次の段階まで学生の意識を高められるものが理想であるが、何事も興味を持つ事が大事と筆者は考える。

興味を抱かせる講義というのは、どの科目にも必要な事項の1つではあるが、ここでは理工系の科目について考えてみる。工学ではさまざまな物理現象を扱い、そこには必ずといって良いほど、数式が出てくる。この数式の物理的意味を説明するために、実際にある事象を用いて説明したり、実験装置を用意して学生の前で実演する。そして再び数式と向きあう事で、理解度の向上を促す。というのが、筆者の良く取る方法であるが、この方法には問題点がある。

実演するというものであるため、ある程度の大きな教材が必要となる。昨今は理系離れを抑制する働きが社会的にあるため、各種教材装置が販売されている。しかし、このような装置はユーザ(我々、教員)から見ると非常に高価である。仕組みが分かっているものに、お金を掛けてまで購入するというものは、モノづくりを掲げている高専に所属する者としていかなものかという、考えが出てきてしまう。そうになると、自作することになる。この自作に意外と時間を削り取られてしまう。実演するからには、さまざま

な環境を想定して、装置を作り上げる必要があるからである。このような事を年間を通じて行なうには、他の業務にも少なからず、支障が生じる。

また、効果を狙って実演したとしても当の学生達には、現象は現象のままとらえるだけにとどまり、その後の自学自習になかなかつながらないことが多い。見たときにはインパクトがあっても、その後どのように数式と向き合えば良いのか我々が伝え切れていないことも要因にある。教科書にそれを補える要素が全ページに渡って書かれていればとても助かるのだが、古典的な(悪い意味ではない)学問になればなるほど、数式が多用されていて、インパクトは残ったとしてもどれが、その事象を説明しているのか分からないという、学生から貴重な意見をアンケートからもらった事がある。

だとすれば、インパクトは実験や実演で良いとして、その後どうすれば良いのか、ということになる。自学自習をしてもらうためには、時間が経っても資料として学生の手元に残るのが良いのではないか。

そこで資料作成のためには、どうしたら良いのか、ということになる。

2 現状の教材作成と CAS+KETpic との比較

専門科目における配布教材は、実演・実験だけでは自学自習の手助けにはなりにくい、といった点から必要である。では、現状はどのように行われているのかを述べる。現状で多くの手段が取られているのは、1) 文献のコピー、2) プログラミング言語+グラフソフト+ワープロの2点があげられる。この2点の特徴について以下に述べる。

2.1 文献コピーによる配布教材



図 1: 文献コピーによる教材サンプルの一部

図1に謄写版印刷機による配布教材の一部を示す。謄写版印刷機は大量印刷を目的とした印刷機であるため、原版の性能に左右されやすい、という問題点がある。そして、パラメータを変更した場合どうなるか、という資料を配布することが難しい。これでは、配布教材の目的である、自学自習による理解度向上の一助にならない。

上記のとおり、文献コピーによる配布教材は原版に左右されるので手軽ではあるが、教材という観点からは適していないことが分かる。

2.2 プログラミング言語+グラフソフト+ワープロ

図1は数式が分かっているならば、自作が可能である。ここでは、プログラミング言語を用いて作成する方法を述べる。

プログラミング言語による数値解析から配布資料を作成する手順は、次のようになる。

1. 数式を分解する (アルゴリズム：解析手法を決定する)。
2. プログラミング
3. 数値データ取得
4. グラフソフトでグラフ表示+グラフ画像データ取得
5. ワープロで編集
6. 配布

図2：プログラミング言語+グラフソフト+ワープロによる教材作成

手順1を実際に行うと次のようなソースとなる。

これは、2階微分方程式 $M\ddot{x} + \mu\dot{x} + kx = 0$ という減衰振動の様子をグラフ化するためのプログラムで、実験実習において配布することを想定したものである。工学専門科目には、実験実習という科目があり、実験のテーマによっては講義を行った後に実験を行うテーマもあれば、実験を行った後に講義を行う、という実験テーマもある。どちらの場合においても、以下に示すような配布教材を用いることの利点があげられる。

- 1) パラメータの変更ができるので、さまざまな状態を想定した実験の準備が行える。
- 2) 1)により、実験での注意力(観察力)を促すことができる。
- 3) 実験値と理論値の比較検討の目安が立てやすい。

しかしながら、同時に以下の点が問題点としてあがる。

- 1) 誤差が解析手法によって異なるため、解析手法の特徴を把握しておく必要がある。
- 2) 適切なコメントを残しておかないとソースを読み取ることが難しい。
- 3) 図3の28, 29行目に示してあるように、プログラミング言語を使用した場合の一般的な出力方法としては、数値データの出力である。

したがって、出てきた数値結果が正しいものなのかをすぐに検証できない。

```
1 #include <stdio.h>
2 //Euler_Method
3 main(){
```

```

4     float x1,x2,x1dot,x2dot,a,b;
5     float t,dt,ft,mu,M,K,pi;
6     int i,n;
7     FILE *out;
8
9     ft=10.0;
10    dt=0.01;
11    pi=3.1415926;
12    x1=0.01;    //initial condition_1
13    x2=0.0;    //initial condition_2
14    n=ft/dt;
15    mu=0.4*pi;    //viscous resistance
16    M=1.0;    //mass
17    K=4.0*pi*pi;    //spring constant
18    a=K/M;
19    b=mu/M;
20
21    out=fopen("vib.dat","w");
22    for(i=1;i<=n;i++){
23        x1dot=x2;
24        x2dot=-a*x1-b*x2;
25        x1=x1+x1dot*dt;
26        x2=x2+x2dot*dt;
27        t=i*dt;
28        printf("%e□%e□%e\n",t,x1,x2);
29        fprintf(out,"%e□%e□%e\n",t,x1,x2);
30    }
31    fclose(out);
32    return 0;
33 }

```

図3：オイラー法による減衰振動の数値解析プログラム (C言語)

そこで、グラフソフトを用いて数値データの検証を行う。グラフソフトはPlots32, gnuplotが用いられる。

Plots32の特徴は、1) ドラッグ&ドロップで結果を得ることができる。2) GUI環境が整っているので、特別な操作方法を必要としない、3) 3次元描画はできない。

gnuplotの特徴は、1) プログラム言語との連結が可能である、2) 記号処理に富んでいて、3次元描画も可能である、3) コマンドライン形式で処理することが多いため、操作に慣れが必要である。

以上のような特徴があるが、操作性を重視(時間節約)してgnuplotよりもPlots32の方がよく用いられている。

このような流れを経て、ワープロ上で編集を行った後に、配布教材として完成する。

2.3 CAS + KETpic

MapleやMathematicaに代表されるCASは、記号処理により数値データやグラフデータを得ることが用意である。基本的には数式を入力すれば、データが得られるため、ソースの可読性については大きな改善効果が得られる。

画面提示のみであれば問題はないが、配布教材として扱うためには前節同様、グラフデータを指定のファイル形式で保存してワープロで編集するといった流れになる。

ワープロは簡単に編集ができて便利なソフトであるが、今回想定している配布教材には数式が書かれるはずである。GUI環境により、数式を簡単に書けるようにはなっているが、まだまだ不十分な点がある。

そこで、CASの持つ数式処理能力を最大限に活用しつつ、数式も十分に扱うことのできる環境が求められる。

以上のような背景から、CAS+KETpicの有効性について述べる。KETpicはCAS用のマクロパッケージであり、CASの能力を活かしつつ、 $\text{T}_\text{E}\text{X}$ 文書中に正確で美しい図を手軽に挿入するためのものである[3]。このパッケージソフトは

<http://www.kisarazu.ac.jp/~masa/math/>

より、無料でダウンロード可能である。

KETpicを用いて $\text{T}_\text{E}\text{X}$ 文書中に図を挿入するには、

1. CASの中でKETpicのコマンドを使い、描画したい図形のプロットデータを計算する。
2. 得られたプロットデータを $\text{T}_\text{E}\text{X}$ 形式の図ファイルとして出力する。

コンパイルした後の図をDVIで確認しながら、図の修正を行う。これが、KETpicの利点の1つである。また、描画用の図ファイルが $\text{T}_\text{E}\text{X}$ 形式となっているため、epsファイルなどと比較してファイル容量が小さいことも利点である。

ここで、前節の2階微分方程式の結果を、MathematicaにKETpicを用いて文書に挿入する例を図4に示す。

```

1 Get["ketpic.m"];
2 M = 1.5; mu = 0.4*Pi; k = 4*Pi*Pi; tend = 10;
3 sol=NDSolve[{M*x''[t]+mu*x'[t]+k*x[t]==0,x[0]==0.01,x'[0]==0.0},x[t],{t,0,
  tend}];
4 q1 = plotdata[x[t] /. sol, {t, 0, tend}, PlotPoints -> 1000];
5 sy = 300;
6 setwindow[{0, 10}, {-0.01*sy, 0.01*sy}];
7 q2 = scaledata[q1, 1, sy];
8
9 openfile["fig.tex"];
10 beginpicture["1cm"];
11 drwline[q2, 1.5];
12 setax["d1.5", "t", "e", "x", "nw1", "0", "sw"];
13 endpicture[1];
14 closefile[];

```

図4：CAS(Mathematica) + KETpicによる描画作成例

図4のソースをCAS上で作成する。3行目が、2階微分方程式を表している。図3のオイラー法を用いたプログラムと比較すると可読性に富んでいることが分かる。また、7行目までが図形のプロットデータを計算するルーチンとなっていて、9行目以降が $\text{T}_\text{E}\text{X}$ 形式の図ファイルとして出力するルーチンとなっている。 $\text{T}_\text{E}\text{X}$ 文書に挿入するには、

```

1 \documentclass{jarticle}
2
3 \newlength{\Width}
4 \newlength{\Height}
5 \newlength{\Depth}
6
7 \begin{document}
8 \begin{figure}[h]
9 \begin{center}
10
11 \input{fig02.tex}
12
13 \end{center}
14 \end{figure}
15
16 \end{document}

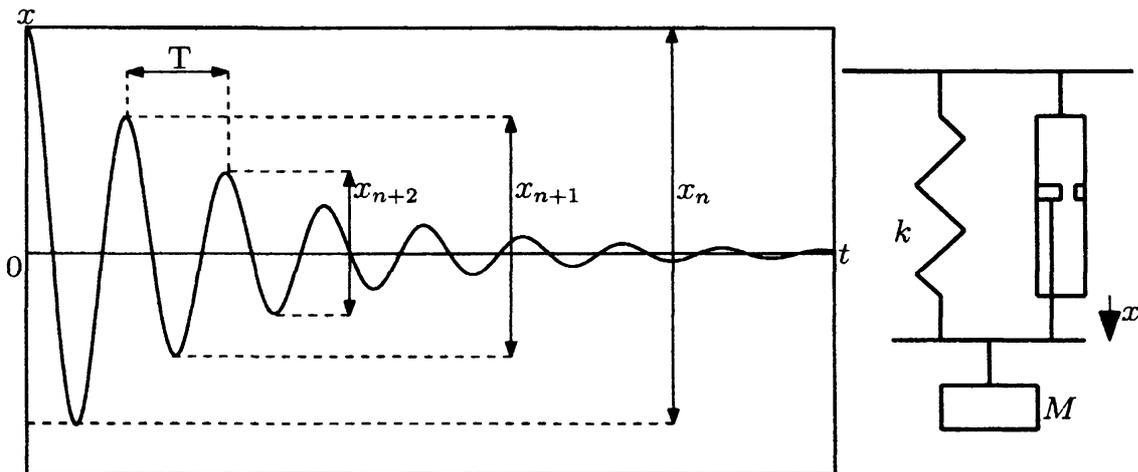
```

図5：TeX 文書への挿入方法

3～5行目は、作成した図ファイルを取込むために定義する距離変数である。

以上の操作を基本とすると、図6に示すような配布教材の大枠がCAS + KETpicで作成できる。

ワープロで編集を行うと、図に用いられているフォントと文章のフォントが異なる場合がある。しかしながら、CAS + KETpicでは、TeX 文書中に挿入することを前提としているためフォントが異なるということが起こらない。図7はKETpicを用いた描画が、正確であるという点を利用した物理教材の例である[4]。 x 、 y の長さを定規で測定して、値を式に代入すると重力加速度 g が得られるというものである。この例を応用すると、実環境での抵抗を考慮した仮想実験が、机上で行うことが可能になるという、教材例である。



$M\ddot{x} + \mu\dot{x} + kx = 0$ 図6：CAS+KETpicによる描画例1

測定: x_1 , y_1 , 0.1秒ごとの質点の x 座標 ($\rightarrow v_{0x}$)

$$\Rightarrow \tan \theta = \frac{2y_1}{x_1}, \quad v_0 = v_{0x} \sqrt{1 + \frac{4y_1^2}{x_1^2}}, \quad g = \frac{2y_1}{x_1^2} v_{0x}^2.$$

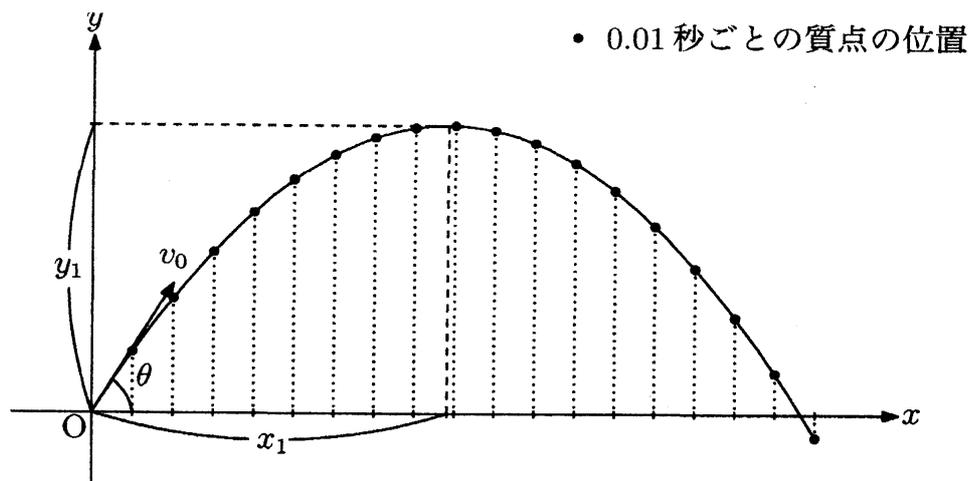


図7：CAS+KETpicによる描画例2

3 KETpicのMaximaへの移植

前章の教材作成例から，MapleやMathematicaといった市販ソフト+KETpicの組み合わせが，配布教材に適していることを示した．この章では，何故KETpicをMaximaへ移植する必要があるのか，という点について述べることにする．

前章において，配布教材を作成するために，CAS+KETpicを使用する利点をあげることができた．

1. 多くのソフトウェアを使用しない．
2. 大量印刷に耐え得る，美しいモノクロ描画である．
3. 正確な図である．

これらは，教材作成のために時間を掛けない，という効果につながる．

1章より，我々が配布教材を作成する目的として自学自習をあげた，実演・実験で興味を持たせ，配布教材で理解度を高める，という流れを掴むことができる．

しかしながら，興味を持って理解度が高くなれば，「このパラメータを変えたらどうなるんだろう？」と思うはずである．そのとき，市販のCASソフトで作成したものであれば，ソースを学生に配布しても環境がなければ，学生は自分の考えを試すことが難しい．また，市販のフリーソフトは高価である．少し試してみようという感覚では，購入意欲はなかなか沸いてこないだろう．そこで，フリーのCASソフトウェアで同様のことが可能であれば，学生の自学自習というもう1つの目的も達成することが可能である．では次に，CASにKETpicを使用する際の条件を述べる．

1. KETpicをCASにロードできる．
2. データや文字列をテキストファイルに書式付書き出しが可能である．
3. 2次元・3次元データを取り出せる．

4. 描画コードを生成するために、数値や文字列を操作できる。

今回は、Maxima への移植を対象とした、Maxima + KETpic で 2 階微分方程式 ($M\ddot{x} + \mu\dot{x} + kx = 0$) の基本描画作成例と描画例を示す。

```

1 load(dynamics)$
2 load("ketpic4maxima.mac");
3 M:1.5;k:4*%pi*%pi;mu:0.4*%pi;a:k/M;b:mu/M;
4 rkdat:rk([v,-a*x-b*v],[x,v],[1,0],[t,0,10,0.001])$
5 rkxy:makelist([rkdat[i][1],rkdat[i][2]],i,1,length(rkdat))$
6 d1:plotdata([discrete,rkxy])$
7 setwindow(0,10,-1,1);
8 ketpic_open_file("fig.tex","tpic","w");
9 openpicture("1cm");
10 drwline(d1);
11 closepicture(1);
12 ketpic_close_file("tpic");

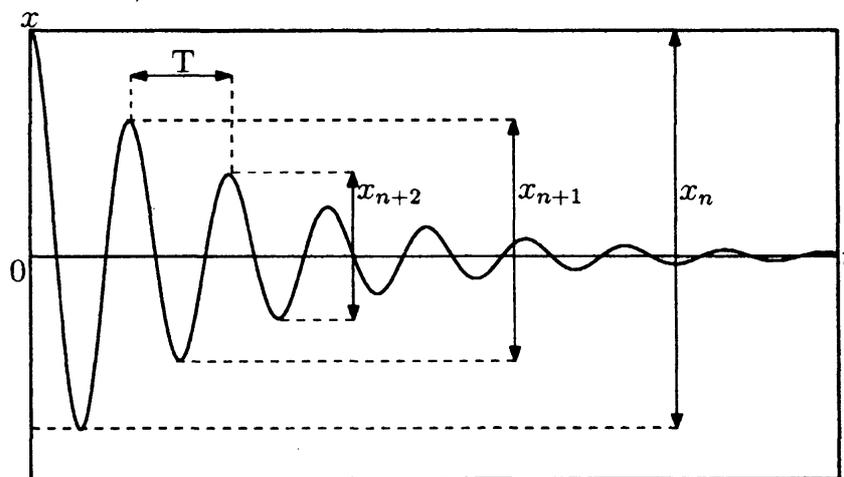
```

図 8 : Maxima + KETpic による作成例

基本コマンドは、Maxima 版よりも前に開発されている Maple, Mathematica 版を踏襲しているため、KETpic のコマンドを CAS 毎に習得する必要はない。Maxima 版は

<http://www.kisarazu.ac.jp/~masa/math/>

より、 α 版 (Ver. 2.5.7c) が無料でダウンロードできる。



$$M\ddot{x} + \mu\dot{x} + kx = 0$$

図 8 : Maxima+KETpic による描画例

4 まとめ

実演・実験だけでなく、数式から学生に現象の全体像を捉えさせるために、KETpic による描画は有効である。それは KETpic による描画が見たとおりの正確な描画ができるためである。