数式処理ソフト MAPLE による数学教育

関西学院大・理工　西谷滋人

1. はじめに

高等教育において数式処理ソフトを教える試みがいろいろなされているが，教員の個人レベルでの試験的導入でしかなく，学科のカリキュラム全体に組み込まれていることはまれである．これは，数式処理ソフトの導入による学生への教育効果，教官の負担を読み切ることができないからであろう．電卓や計算尺と同じ問題をはらんでいる．しかし，わからないからと言って一歩も踏み出せずにいては，日本の大学が抱える数学教育の問題を克服することはできない．来年度より，本学部数理科学科において数式処理ソフトを用いた演習が必修化されることは，単に学生に数式処理ソフトの使用を奨励するという段階を超えて，全教官が数式処理ソフトの使用を前提に講義を進めることが可能となる画期的な試みである．著者個人は 10 年以上にわたって高等教育の現場で数式処理ソフトの導入をいろいろと試みたが，学科の教官・学生大多数への数式処理ソフトの日常的な使用を説得することができず，成功したとは言いがたい．しかし，いくつかの試行錯誤のなかで，学習者の多様性を体感してきた．その経験を通して得られた，学習者レベルに合った効率的な数式処理ソフト習得法を紹介する．

2. シングルステップの式変形：数式処理ソフトで簡単にできること

数式処理ソフトを習得するのは，プログラミング言語を習得するのとは少し違ったセンスが必要である．しかしそのセンスはなにもむずかしい構文を新たに覚えるのではなく，それまで積み上げてきた数学の知識にほんの少し修正を加えるだけでいい．この修正は中学の 1，2 年生あたりで導入される代数計算，あるいは式変形の約束と似たところがある．まずは，数式処理ソフト Maple で簡単にできることを観ながら，等号の意味を再確認しよう．

中学あたりでつまずく学生の多くが，式の変形，代入，方程式を混同していることがある．例えば，小島寛之の著書に，よくある中学生の間違いとして，

$$3x - x = 3$$

という例が載っている [1]．これは $3x$ が $3 * x$ の略記であることを知らずに「3 と x から x を引いたら 3 が残る」と思っているらしい．たしかに小学校の「$3\frac{1}{2} - \frac{1}{2} = 3$」なら正しい計算となる．期待している正しい「変形」は

$$3x - x = 2x$$

である．ところが，元の式も方程式と見なせば正しく，その辺りの混乱を引きずって知識を積み上げているようである．もうひとつ，プログラミングを大学で教えていて，なかなか納得しない学生のなかに，

$$x = x + 1$$

は明らかに「式の変形，および方程式」として間違っているからというのがいる．刷り込まれた知識を否定するのはことのほか困難なようだ．しかし，いざ自分が教えるときには

「プログラミング言語の仕様上そうなっているから自分で適当に判断してね.」となる．これらすべては等号「＝」があまりにも無節操に利用されているからと見なすことができよう．
　コンピュータが解釈しなければいけない数式処理ソフトである Maple ではこのような曖昧さが許されない．たとえば,

　　a=3,b=2 のとき，3a+4b の式の値はいくらか.

という問題を Maple に教えてやる場合には,

```
> a:=3;
  b:=2;
  3*a+4*b;
```

$$a := 3$$
$$b := 2$$
$$17$$

となる．等号の代わりに使われている記号":="は，代入を意味している．これに，等号を間違って使ったとしても

```
> restart;
  a=3;
  b=2;
  3*a+4*b;
```

$$a = 3$$
$$b = 2$$
$$3a + 4b$$

となりなにも意味のある答えを返してこない．$a = 3$ や $b = 2$ は単にこういう式があると解釈され，a, b に数値が代入されることはない．ちなみに，restart は上の方で a, b に代入した値を忘れさせるための初期化 (restart) 操作であり，行末の";"は Maple が解釈すべきひとかたまりの入力の区切りを意味している．こういう解釈で問題を読み直すと,

　　a に 3 を,b に 2 を代入したとき，$3 * a + 4 * b$ の式の値はいくらか.

となる．かけ算も含めて省略することは許されないし，等号の意味を明示的に教える必要がある.
　同様に,

　　$3x - x = 3$ の方程式を満たす x はいくらか.

は

　　$3 * x - x = 3$ を x の方程式として解け (solve).

と読み直して,

```
> solve(3*x-x=3,x);
```

$$\frac{3}{2}$$

という Maple のコマンドになる．また，展開や因数分解といった式変形も明示的に

　　$(x - 2)^2$ を展開 (expand) せよ.

および

　　$x^2 - 3x + 2$ を因数分解 (factor) せよ.

と読み直して,

```
> expand((x-2)^2);
```

$$x^2 - 4x + 4$$

```
> factor(x^2-3*x+2);
```

$$(x-1)(x-2)$$

としなければならない．また通分 (normal) は

```
> normal(1/a+1/b);
```

$$\frac{b+a}{ab}$$

となる．

さらに，

$ax + bx - c$ を同類項でまとめよ．

というちょっと曖昧な問題は

$ax + bx - c$ を x の同類項でまとめよ (collect)．

と読み直して，

```
> collect(a*x+b*x-c,x);
```

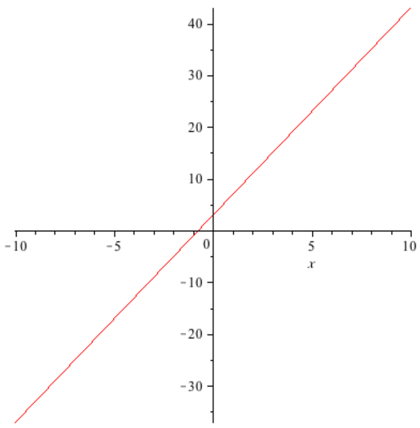
$$(a+b)x - c$$

となる．また関数のプロット

$y = 4x + 3$ をプロットせよ．

は，

```
> plot(4*x+3,x);
```



となる．

またこれだけでなく，微分 (diff) や積分 (int) も

```
> diff(4*x+3,x);
```

$$4$$

```
> int(4*x+3,x);
```

$$2x^2 + 3x$$

として求めることができる．これらを新たな式 (eq2) として代入すれば，

```
> eq2:=int(4*x+3,x);
```

$$eq2 := 2x^2 + 3x$$

再利用が可能となり，

```
> solve(eq2=0,x);
```

$$0, -\frac{3}{2}$$

などとできる．

数式処理ソフトではもっと複雑な計算もコマンド一発ででき，2次方程式の解法や部分積分，線形代数などの途中の複雑な計算・手順にわずらわされることはない．ただし，数学の用語に該当する英単語を覚えなければならない．理系に来ている学生の多くは，英語ができないという消去法で理系を選択していることが多いので，横文字を見ただけで凍り付く傾向がある．しかし，専門用語を覚え，英語の論文を読む際に不可欠の単語に早いうちからなじむという観点から，この程度の暗記は避けるべきではなかろう．逆に，ある程度の経験のある研究者は，ほとんど自分の知識だけで自然にMapleが習得・操作できる．

3. マルチステップの式変形：数式処理ソフトでも簡単にできないこと

先ほど紹介した例がコマンド一発で解が得られるのに対して，だいぶ込み入った例を次に紹介する．まず，次に示す式変形の課題は，普通の試験問題としては，成立していないことに注意していただきたい．学生さんたちは，紙面の上で導出されてしまっている式を再度込み入ったソフトを使って導出することに問題としての意味を感じない，あるいは何をすればいいか分からないそうである．ところが，数式処理ソフトの熟達者にとっては，こういう類いの問題は，使っている数式処理ソフトで何が簡単にできて何がやりにくいかという評価や，どういったときにどういった手法を使うのかが学習できる典型的な問題である．また，実際の研究においても，式の導出を確認し，さらに値を変えるという作業はもっとも遭遇頻度が高い．数式処理ソフトの熟達者が備えているべき行動，つまり数式処理ソフト学習のターゲット行動はこのような「式の導出をコマンドでおこなう」あるいは「マルチステップの式変形」である．

3.1. **課題.** プランクの法則は，黒体の表面から放出される光の強度は以下のようになることを示している．

$$I(\nu, T) = \frac{2h\nu^3}{c^2}\frac{1}{\exp\left(\frac{h\nu}{kT}\right) - 1} \tag{1}$$

ここで，hはプランク定数，cは光速度，kはボルツマン定数，νは振動数，Tは温度を表している．つまり，面積Aの黒体の表面からは，立体角$\mathrm{d}\Omega$に向かって，$(\nu + \mathrm{d}\nu)$の振動数帯から$I(\nu, T)A\mathrm{d}\nu\mathrm{d}\Omega$の強度を持った光が放出されていることを示している．これをすべての立体角および振動数で積分すれば，黒体の放出する単位面積あたりの全強度を示すステファンーボルツマンの関係

$$j = \sigma T^4 \tag{2}$$

が導かれる．この法則を以下のヒントを参考にしてMapleで導出せよ．

3.2. **ヒント：導出．** 立体角の積分公式から，全積分は

$$j = \frac{2h}{c^2}\int_0^{\infty}\int_0^{2\pi}\int_0^{\frac{\pi}{2}}\frac{\nu^3}{\exp\left(\frac{h\nu}{kT}\right)-1}\cos(\phi)\sin(\phi)\mathrm{d}\nu\mathrm{d}\theta\mathrm{d}\phi \tag{3}$$

で与えられる．θ と ϕ についてまず積分すると，被積分関数の中の定数を A として，

$$\int_0^{2\pi}\int_0^{\frac{\pi}{2}}\mathrm{A}\cos(\phi)\sin(\phi)\mathrm{d}\theta\mathrm{d}\phi = \mathrm{A}\pi \tag{4}$$

が得られる．これより

$$j = \frac{2\pi h}{c^2}\int_0^{\infty}\frac{\nu^3}{\exp\left(\frac{h\nu}{kT}\right)-1}\mathrm{d}\nu \tag{5}$$

となる．ここで $\nu = \frac{xkT}{h}$ で置換すると，$\mathrm{d}\nu = \frac{kT}{h}\mathrm{d}x$ を忘れず，

$$j = \frac{2\pi k^4T^4}{c^2h^3}\int_0^{\infty}\frac{x^3}{\exp(x)-1}\mathrm{d}x \tag{6}$$

となる．この積分は

$$\int_0^{\infty}\frac{x^3}{\exp(x)-1}\mathrm{d}x = \frac{\pi^4}{15} \tag{7}$$

であるので，

$$j = \sigma T^4 \tag{8}$$

$$\sigma = \frac{2}{15}\frac{\pi^5k^4}{h^3c^2} \tag{9}$$

が得られる [2]．

3.3. **ヒント：Maple のテクニック．** どこから手をつけてもよいが，最終的な Maple スクリプトとしては，プランクの法則の式から σ の値までがすべて式の変形，代入であらわされていることが理想である．特に難しいのは係数の取り出しで，例えば，

```
> restart;
  nu:=x*k*T/h;
  cdx:=diff(nu,x);
```

$$\nu := \frac{xkT}{h}$$

$$cdx := \frac{kT}{h}$$

とすれば，置換積分に伴う係数を定義することができる．また，$\frac{2\pi k^4T^4}{c^2h^3}$ などは coeff 関数で x の 3 乗の係数を取り出すなどとすればよい．ただし，

```
> c1:=A2*x^3/(exp(x)-1);
  coeff(c1,x^3);
```

$$c1 := \frac{\mathrm{A2}x^3}{\exp(x)-1}$$

```
Error, unable to compute coeff
```

となるが，以下のようにすれば係数 A2 が取り出せる．

```
> coeff(c1*(exp(x)-1),x^3);
```

$$A2$$

3.4. **解答例.** 解答例を示しておく．まず立体角に対する積分の確認は

```
> restart;
  eq1:=int(int(A*sin(phi)*cos(phi),phi=0..Pi/2),theta=0..2*Pi);
```

$$eq1 := A\pi$$

となる．次に，この結果を受けて Planck の公式を書き下すと

```
> eq2:=2*h*nu^3/c^2/(exp(h*nu/k/T)-1)*Pi;
```

$$eq2 := \frac{2\,h\,\nu^3\pi}{c^2\left(\exp\left(\frac{h\nu}{kT}\right)-1\right)}$$

である．ここで変数変換

```
> nu:=x*k*T/h;
  dx:=diff(nu,x);
```

$$\nu := \frac{xkT}{h}$$

$$dx := \frac{kT}{h}$$

をおこなう．すると eq2 は

```
> eq2*dx;
```

$$\frac{2x^3k^4T^4\pi}{h^3c^2(\exp(x)-1)}$$

となる．このままとりあつかっても変形できるが，ややこしい係数を切り離しておく．上記の Maple のテクニックのヒントに従って

```
> eq2*dx*(exp(x)-1);
```

$$\frac{2x^3k^4T^4\pi}{h^3c^2}$$

とした後で，係数を取り出す.

```
> c1:=coeff(eq2*dx*(exp(x)-1),x^3);
```

$$c1 := \frac{2k^4T^4\pi}{h^3c^2}$$

残された被積分関数を取り出す.

```
> eq3:=eq2*dx/c1;
```

$$eq3 := \frac{x^3}{\exp(x)-1}$$

積分を実行して，

```
> eq4:=int(eq3,x=0..infinity)*c1;
```

$$eq4 := \frac{2}{15}\frac{\pi^5 k^4 T^4}{h^3 c^2}$$

ここでは係数をつけなおしている．最終的に σ の値は，それぞれの物理定数を代入して

```
> h:=6.6261*10^(-34);
  k:=1.3807*10^(-23);
  c:=2.9979*10^8;
  evalf(coeff(eq4,T^4));
```

$$h := 6.626100000\ 10^{-34}$$

$$k := 1.380700000\ 10^{-23}$$

$$c := 2.997900000\ 10^{8}$$

$$5.671228656\ 10^{-8}$$

となり，ステファン・ボルツマン定数が正しく導かれる．

3.5. **分析.** この問題を 2 つの演習クラスの受講生に，学習を初めて一ヶ月たった時点での試験問題として課した．かたや数学科，かたや情報科学科の 3 回生である．それまでに基本操作を教え，使いそうなコマンドの一覧表を手渡している．その結果，数学科の学生は 18 人中 6 名がこの問題を解いた．一方，情報科学科学生は 51 人中 3 名しか解けなかった．問題を観ればわかる通り，広義積分をのぞけば，これは典型的な高校の数式変形の問題である．複雑な問題をコマンド一発で答えが出るシングルステップの問題に分解し，その解を設定して問題を試行錯誤しながら解決して行く能力に差がついていることがうかがわれる．入学時の学力差はそれほど大きくないことから，1，2 回生での数学の演習においてこのような類いの問題になじんでいるかいないかが大きな差となって現れたと考えられる．

一方，プログラミングの要素が強くなる問題においてはこの傾向が逆転することを申し添えておく．いずれにしろ，数式処理ソフトを使わなくても解ける学習者は，数式処理ソフトを使っても解けるという試験結果であった．この結果は，一般のカリキュラムへ数式処理ソフトを導入しても教育効果はあまりないと判断されても仕方がないものである．しかし，もう少し学習行動を突っ込んで考えると，少し違った観点を得ることができる．

4. 数値計算と視覚化：非線形最小二乗法を例に

数式処理ソフトのスキル向上の方策を示す前に「Maple を使うご利益」を実感できる例を示そう．それは数値計算と視覚化で，非線形最小二乗法によるデータフィットが好例である．

4.1. **課題.** Figure 1 のようなデータにローレンツ関数をフィットする非線形最小二乗法の問題を考えよう．ただし，以下では結果を見やすくするためデータ数を 8 個として解説している．

フィッティング関数を

$$f(t;\mathbf{a}) = a + \frac{b}{c + (t-d)^2} \tag{10}$$

のような単純な 4 個のパラメータ $\mathbf{a} = (a, b, c, d)$ を持ち，時間 t で変動するローレンツ関数とする．パラメータの初期値を $\mathbf{g_0} + \delta\mathbf{g_1} = (a_0 + \delta a_1, b_0 + \delta b_1, c_0 + \delta c_1, d_0 + \delta d_1)$ とする．

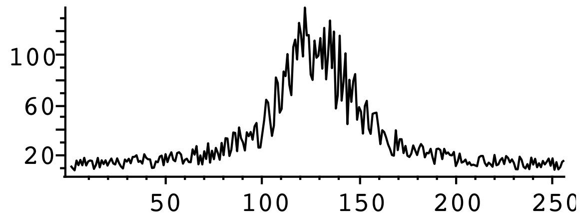


FIGURE 1. **ローレンツ関数にノイズを加えて作成したサンプルデータ.**

このとき関数 f をパラメータの真値 $\mathbf{g_0} = (a_0, b_0, c_0, d_0)$ のまわりでテイラー展開して，高次項を無視すると，

$$\begin{aligned}\delta f &= f(t;\mathbf{g_0}+\delta\mathbf{g_1}) - f(t;\mathbf{g_0}) \\ &= \left(\frac{\partial}{\partial a}f\right)_{\mathbf{g_0}}\delta a_1 + \left(\frac{\partial}{\partial b}f\right)_{\mathbf{g_0}}\delta b_1 + \left(\frac{\partial}{\partial c}f\right)_{\mathbf{g_0}}\delta c_1 + \left(\frac{\partial}{\partial d}f\right)_{\mathbf{g_0}}\delta d_1\end{aligned} \tag{11}$$

となる．ここで，偏微分はパラメータの組 $\mathbf{a_0}$ での関数であることを明示している．時刻 $t = 1, 2, \ldots, 8$ に対応したデータ値を $f_1, f_2, \ldots, f_8$ としよう．各データ点とモデル関数から予測される値との差を $\delta f_1, \delta f_2, \ldots, \delta f_8$ とすると，この差分ベクトルは

$$\begin{pmatrix}\delta f_1\\ \delta f_2\\ \vdots\\ \delta f_8\end{pmatrix} = J\begin{pmatrix}\delta a_1\\ \delta b_1\\ \delta c_1\\ \delta d_1\end{pmatrix} \tag{12}$$

となる．ここで J はヤコビ行列と呼ばれる行列で，8 行 4 列の行列

$$J = \begin{pmatrix}\left(\frac{\partial}{\partial a}f\right)_{t=1} & \left(\frac{\partial}{\partial b}f\right)_{t=1} & \left(\frac{\partial}{\partial c}f\right)_{t=1} & \left(\frac{\partial}{\partial d}f\right)_{t=1}\\ \left(\frac{\partial}{\partial a}f\right)_{t=2} & \left(\frac{\partial}{\partial b}f\right)_{t=2} & \left(\frac{\partial}{\partial c}f\right)_{t=2} & \left(\frac{\partial}{\partial d}f\right)_{t=2}\\ \vdots & \vdots & \vdots & \vdots\\ \left(\frac{\partial}{\partial a}f\right)_{t=8} & \left(\frac{\partial}{\partial b}f\right)_{t=8} & \left(\frac{\partial}{\partial c}f\right)_{t=8} & \left(\frac{\partial}{\partial d}f\right)_{t=8}\end{pmatrix} \tag{13}$$

である．これは，関数 f の各パラメータによる偏微分関数にデータ点の t 値を入れた値で構成される．このような矩形行列の逆行列は転置行列 J^{T} を用いて，

$$J^{-1} = \left(J^{\mathrm{T}}J\right)^{-1}J^{\mathrm{T}} \tag{14}$$

で求められる．従って真値からのずれは

$$\begin{pmatrix}\delta a_2\\ \delta b_2\\ \delta c_2\\ \delta d_2\end{pmatrix} = \left(J^{\mathrm{T}}J\right)^{-1}J^{\mathrm{T}}\begin{pmatrix}\delta f_1\\ \delta f_2\\ \vdots\\ \delta f_8\end{pmatrix} \tag{15}$$

となる．理想的には $(\delta a_2, \delta b_2, \delta c_2, \delta d_2)$ は $(\delta a_1, \delta b_1, \delta c_1, \delta d_1)$ に一致するはずだが，測定誤差とテイラー展開で無視した高次項のために一致しない．それでも初期値に比べ，真値により近づく．そこで，新たに得られたパラメータの組を新たな初期値に用いて，より良いパラメータに近付けていくという操作を繰り返す．新たに得られたパラメータと前のパラ

メータとの差がある誤差以下になったところで計算を打ち切り，フィッティングの終了となる．

4.2. Maple によるデータの準備. 線形代数計算のためにサブパッケージとして LinearAlgebra を呼びだしておく．

```
> restart;
  with(plots):
  with(LinearAlgebra):
```

ローレンツ型の関数を仮定し，関数として定義する．

```
> f:=t->a+b/(c+(t-d)^2);
```

$$f := t \rightarrow a + \frac{b}{c + (t-d)^2}$$

関数を一次代入 (subs) を用いて作り，データをリスト T に入れておく．

```
> ndata:=8:
f1:=t->subs({a=1,b=10,c=1,d=4},f(t));
T:=[seq(f1(i),i=1..ndata)];
```

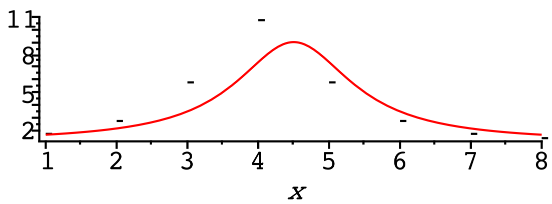
$$f1 := t \rightarrow \mathrm{subs}\left(\{a=1, b=10, c=1, d=4\}, f(t)\right)$$

$$T := \left[2, 3, 6, 11, 6, 3, 2, \frac{27}{17}\right]$$

初期値 (g1=guess1) を真値からずらして仮定して作った関数と，先ほど作ったデータをともに表示する．

```
> l1:=listplot(T,connect=false):
  g1:=Vector([1,8,1,4.5]):
  guess1:={a=g1[1],b=g1[2],c=g1[3],d=g1[4]};
  p1:=plot(subs(guess1,f(x)),x=1..ndata):
  display(l1,p1);
```

$$guess1 := \{a = 1, b = 8, c = 1, d = 4.5\}$$



4.3. **Maple による解法例.** ヤコビ行列の中の微分を新たな関数として定義しておく.

```
> dfda:=unapply(diff(f(x),a),x);
  dfdb:=unapply(diff(f(x),b),x);
  dfdc:=unapply(diff(f(x),c),x);
  dfdd:=unapply(diff(f(x),d),x);
```

$$
\begin{aligned}
dfda &:= x \to 1 \\
dfdb &:= x \to \frac{1}{c+(x-d)^2} \\
dfdc &:= x \to -\frac{b}{(c+(x-d)^2)^2} \\
dfdd &:= x \to -\frac{b(-2x+2d)}{(c+(x-d)^2)^2}
\end{aligned}
$$

関数値 $f(i)$ とデータ値 $T[i]$ との差分ベクトルを df として求める.

```
> df:=Vector([seq(subs(guess1,T[i]-f(i)),i=1..ndata)]);
```

$$
df := \begin{pmatrix} 0.39623 \\ 0.89655 \\ 2.53846 \\ 3.60000 \\ -1.40000 \\ -0.46154 \\ -0.10345 \\ -0.01554 \end{pmatrix}
$$

for-loop で，データ点でのヤコビ行列を求める.

```
> Jac:=Matrix(ndata,4):
  for i from 1 to ndata do
    Jac[i,1]:=subs(guess1,dfda(i));
    Jac[i,2]:=subs(guess1,dfdb(i));
    Jac[i,3]:=subs(guess1,dfdc(i));
    Jac[i,4]:=subs(guess1,dfdd(i));
  end do:
```

このまま表示すると見にくいので，桁数を落として出力する.

```
> interface(displayprecision = 3):
  Jac;
```

$$
\begin{pmatrix}
1.000 & 0.075 & -0.046 & -0.319 \\
1.000 & 0.138 & -0.152 & -0.761 \\
1.000 & 0.308 & -0.757 & -2.272 \\
1.000 & 0.800 & -5.120 & -5.120 \\
1.000 & 0.800 & -5.120 & 5.120 \\
1.000 & 0.308 & -0.757 & 2.272 \\
1.000 & 0.138 & -0.152 & 0.761 \\
1.000 & 0.075 & -0.046 & 0.319
\end{pmatrix}
$$

J^{-1} を求める.

```
> InvJac:=(MatrixInverse(Transpose(Jac).Jac)).Transpose(Jac);
```

$$\begin{pmatrix} 0.565 & 0.249 & -0.354 & 0.040 & 0.040 & -0.354 & 0.249 & 0.565 \\ -2.954 & -0.506 & 4.012 & -0.552 & -0.552 & 4.012 & -0.506 & -2.954 \\ -0.352 & -0.029 & 0.557 & -0.176 & -0.176 & 0.557 & -0.029 & -0.352 \\ -0.005 & -0.012 & -0.035 & -0.080 & 0.080 & 0.035 & 0.012 & 0.005 \end{pmatrix}$$

J^{-1} と差分ベクトル df との積を計算する.

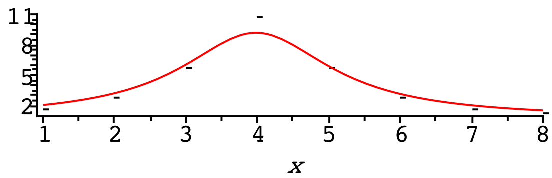
```
> dg:=InvJac.df;
```

$$dg := \begin{pmatrix} -0.235 \\ 5.592 \\ 0.613 \\ -0.520 \end{pmatrix}$$

これをまたもとのパラメータの近似値 (guess1) に入れ直して表示させる.

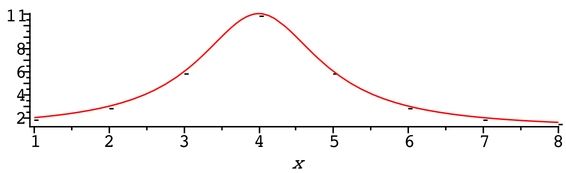
```
> g1:=g1+dg:
  guess1:={a=g1[1],b=g1[2],c=g1[3],d=g1[4]};
  p1:=plot(subs(guess1,f(x)),x=1..ndata):
  display(l1,p1);
```

$$guess1 := \{a = 0.765, b = 13.592, c = 1.613, d = 3.980\}$$



カーブがデータに近づいているのが確認できよう. 差分ベクトルを求める操作以降を繰り返すと, dg の各要素が 0 に収束していく. これらの操作を 4 回繰り返した後のパラメータの値とフィッティング関数の様子を以下に示す.

$$guess1 := \{a = 1.006, b = 9.926, c = .989, d = 4.000\}$$



非線形のデータフィットは数式による説明だけではパラメータが多く何をすればいいのかが分かりにくい. しかし Maple を使えば, データから求まる差分ベクトル, ヤコビ行列などの成分や, データと関数との適合具合をグラフで見ながら計算の進行を確かめることができる.

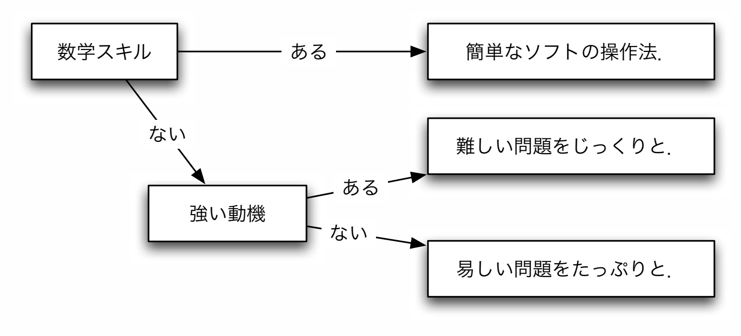


Figure 2. **数式処理ソフトスキルの学習法.**

5. 数式処理ソフトスキル向上の方策の提案

最後の例は，数式処理ソフトが備える視覚化と数値計算の利点を示している．このような作業は，C 言語や FORTRAN などのプログラミング言語と視覚化ソフトを使ってもできるが，数式処理ソフトの操作に熟達していると極めて容易である．これだけでも数式処理ソフトを学習させる動機となろう．しかし，そこで示したスクリプトはすこし凝りすぎで，高度なテクニックの習得が必要となり，初学者に学習させるような代物ではない．どちらかというとそのまま使ってもらう見本のような物であろう．

真の数式処理ソフトの熟達者となるためのターゲット行動は,「マルチステップの式変形」である．では，そのようなスキルの獲得にはどのような教授法が適当であろうか．数学のスキルがすでにある学習者が数式処理ソフトに習熟することはそれほど難しくない．「シングルステップの式変形」で示した通り英語になじむぐらいである．数学を使う動機も数学のスキルも高い学生が，無理して数式処理ソフトを使う必要はなさそうに思えるかもしれない．しかし，数式処理ソフトを使った問題解決法は，一度身に付けた戦略を捨てる必要はなく，手軽に計算ミスをなくしたり，新たな展開の方向を見つけることができる．

では，数学のスキルが低い学生にどのようにして数式処理ソフトのスキルを習得させればいいだろうか．まず,「数学」と「計算」を分けて考えよう．著名な物理学者のファインマンはその著書のなかで，物理の問題を解く際に必要となる代数や，微分，積分は，小学生時代の九九と同じように暗記するよう薦めている．ところが直後に彼は,「(物理学者のやり方を習うには,）公式の暗記だけに頼ることはやめて，自然界の様々なものごとの相互の関連性を理解すること，そのことを心がけてほしい.」としている [3]．この線引きに数学の先生方はご立腹されるだろうが，ある程度の下位技能の自動化は不可欠である.「マルチステップの式変形：数式処理ソフトでも簡単にできないこと」でお見せした通り，数式処理ソフトを使えば,「数学」にあたる式変形と，単なる「計算」とを区別することができる．そして「計算」はほとんどの場合に自動化してもよい下位技能にあたる．

まず数学のスキルは低くとも，動機は高い学習者に対する適切な方策を考えよう．この場合は難しい問題をじっくりと解くのが良さそうだ．前報において，学生が Maple を使うのに障壁を感じる原因の一つとして,「すぐに問題が難しくなり，Maple に費やす時間よりも問題とその解答を理解するのに多くの時間が割かれ，実際に Maple をいじっている時間はわずかのように思われます.」という学生の感想をあげた [4]．その当時は，この批判に対して，著者個人の経験から，簡単な問題をいくらやってもすぐにコマンドを忘れてしまい時間の無駄と反論した．それは，強い動機を持った学習者，すなわち複雑な問題を解か

なければならずかつ「計算」の能力が低い場合には数式処理ソフトが不可欠と信じている学習者，に対する方策であったのであろう．いくつかの問題を数式処理ソフトを使って自力で解けたという成功体験が，ソフトに対する強い信頼感を生み，習熟への動機を高めていく．

では，数学のスキルが低く，さらに動機も低い学生に対する適切な方策はなんであろうか．数学の問題解決は，自動車の運転や，ピアノの運指と同じで，手続き的知識と見なすことができる．このようなスキルを身につけるためには，ある程度のレベルまで単純練習を集中して繰り返すのが効果的とされている [5]．ある程度の「料理」の能力がなければ，「包丁」の使い方になじんでも，料理を作ることはなくそのスキルもすぐに忘れてしまい，身に付く物が何もない．現在の「ゆとり」あるいは「大学全入」世代に高校での相当量の演習を通じた「数学」能力の習得はあまり期待できない.「計算」を数式処理ソフトに頼りながらでも，簡単な「数学」の演習を多くこなさせることが効果的な方策ではなかろうか.そのような視点からのテキストの試みとして,「チャート式 Maple 演習」を公開している [6]．逆説的な言い方になるが，いままで何もしていないというのはこだわる解決法がないのだから，数式処理ソフトを使うという新しいやり方に抵抗感がなく，素直に習慣化するかも知れない．学生たちがその昔関数電卓を手に数学や物理の問題に挑んだように，まずは数式処理ソフトを立ち上げるようになることを願う．

References

[1] 小島寛之,『数学でつまずくのはなぜか』, 講談社,2008.
[2] 英語版 Wikipedia，項目「Stefan-Boltzmann law」より訳出, http://en.wikipedia.org/wiki/Stefan-Boltzmann_law.
[3] ファインマン，ゴットリーブ，レイトン，『ファインマン流物理がわかるコツ』, 岩波書店，2007, p.31.
[4] 西谷滋人,『Maple を利用した応用数学教育』, コンピュータ＆エデュケーション, Vol.13(2002), 33–39.
[5] B. フィールディング,『同じテーブルの 10 人の名前，簡単に覚えられます.』, 三笠書房, 2005.
[6] 西谷滋人,『チャート式 Maple 演習』, http://ist.ksc.kwansei.ac.jp/~nishitani/Lectures/Maple/BottomLine0.html.