数式処理ソフトＭａｘｉｍａの利用について

愛媛県立八幡浜高等学校 橋 本　潔

１ はじめに

ＭａｘｉｍａはＭａｔｈｅｍａｔｉｃａやＭａｐｌｅと同様、非常に本格的な数式処理システムである。　Ｍａｘｉｍａは　GNU　Public License(GPL)のもとにリリースされており、汎用数式処理システムでは　数少ない非商用(無料)・オープンソースのもので、その完成度は商用のものに劣らない。これまで、Ｍａｔｈｅｍａｔｉｃａには興味はあるが、導入コストがかかるという理由から手を出せなかった人も多いはずである。そこで、今回、無料で使用できる数式処理ソフトＭａｘｉｍａの利用法について研究することにした。

２ インストール方法

現在、ＭａｘｉｍａはWindows版とUNIX版がある。今回は、学校現場での利用場面を考え、Windows版のインストールの仕方を紹介する。導入は非常に簡単で、後述のホームページから　maxima-5.9.2.exe　をダウンロードし、ダブルクリックしていくのみである。グラフの描画にはＧｎｕｐｌｏｔを使用することになるが、Ｍａｘｉｍａのインストール時に自動的に組み込まれるようになっている。また、フロントエンドとして使用するＴｅＸｍａｃｓについても、同様にダウンロードしたwintexmacs-1.0.5.exeをダブルクリックすることでインストールすることができる。

３ 基本となる使い方

起動後に現れるプロンプトで、処理したい数式やコマンドを入力し、行末にセミコロン(;)を入力することで処理を行う。処理した結果は即時に返されるが、行末にセミコロンではなくドルマーク($)を使用することで、結果を表示せずに入力作業を継続することができる。

(1) 関数を定義する

繰り返し使用する関数を定義しておくことで、その後に続く処理を簡単に記述することができる。関数の定義は、「 関数名:= 式 」の形で行う。

例.　f(x):= x^3+2;
g(x):= sin(x)*cos(x);

(2) 式を展開する

「 expand(式); 」とすることで、式の展開が行われる。式の記述に使われる演算子は、「 +,−,*,／, ^ 」などプログラミングで使われているものと同じものが使用できる。

(3) 式を因数分解する

「 factor(式); 」とすることで、因数分解が行われる。

(4) 方程式を解く

１変数の方程式については、「 solve(方程式); 」とすることで、解を求めることができる。連立方程式については、「 solve([方程式１, 方程式２ ,…] ,[変数1, 変数2,…]); 」と方程式・変数を複数ならべることで連立方程式の解くことができる。

例.　solve([x^2-2*x-3=0], x);
　　solve([x+y=5,3*x-5*y=10] , [x,y]);

(5) 微分・積分する

微分するには、「 diff(式 , 変数); 」、積分するには、「 integrate(式 , 変数); 」とすることで微分・積分することができる。

例. diff(sin(x) , x);
　　integrate(1/x , x , 1 , 2);

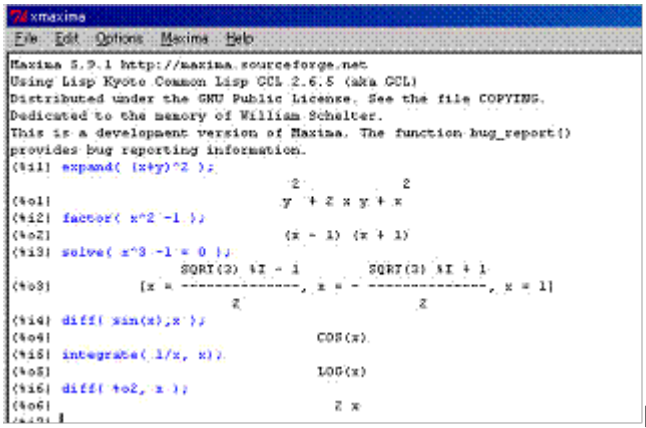

図１．(2)〜(5)の使用例

(6) 関数の極限を求める

関数の極限を求めるには、「 limit(関数, 変数, 値); 」とすることで関数の極限を求めることができる。∞を使用するときは、「 inf 」を用いて表すことになっている。

例.　limit((2*x+1)/(3*x+2), x , inf);
　　limit(sin(3*x)/x, x , 0);

(7) 行列に関する演算

行列の定義は行ごとに成分を指定し、定義する。行列の和は「 + 」 で、積は「 . 」で計算する。また、逆行列は「 invert(A); 」で求めることができる。

例. A:matrix([1,2],[3,4]); // 行列A,Bを定義し
B:matrix([1,0],[0,1]); // 和を表示する
A+B;

4 少々高度な使い方

(1) 以前の結果を使用する

Ｍａｘｉｍａの処理は、入力に対しそれをその場で解釈し実行するインタープリタ形式で行われる。直前に行った結果を使用したいときは、「 % 」を指定することで、複雑な式を記述しないで作業を行うことができる。また、行頭の「 %o1 」や「 %o2 」などのラベルを指定することで、さらに遡って結果を参照することも可能である。

(2) グラフをかく

グラフを描画するには「 plot2d(式, 範囲); 」とすることで、Ｇｎｕｐｌｏｔ が呼び出され、きれいなグラフが描画できる。下図は、「 plot2d(x*sin(x), [x,0,6]); 」と入力して得られるグラフである。

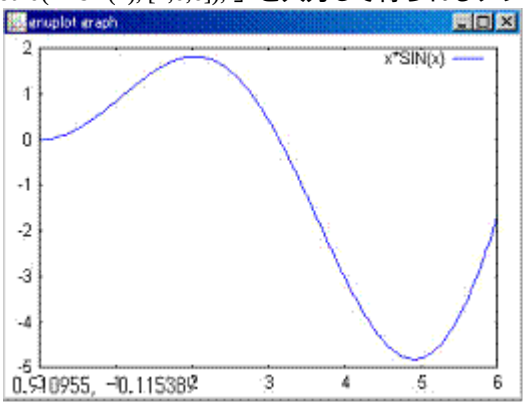

図２． ２次元グラフ出力例

(3) 結果をＴｅＸ形式で出力する。

処理した結果をＴｅＸ形式に変換して、文書の中に組み込むことができる。その際には「 Tex(式, ファイル名); 」の形で実行すれば、出力結果をファイル中に書き出すことができる。

5 プログラム

Ｍａｘｉｍａの処理は、一つずつ入力しながら結果を表示するだけではない。あらかじめ実行する処理をファイルに書いておき、それを「 batch(“ファイル名”); 」で一括処理することができる。

```
IsPrime(x):=block(
    local(i,w),
    w:1,
    for i:2 thru x-1
       do(
         if fix(x/i)*i=x then w:0,return
       ),
    return(w)
);
```

リスト１ 素数の判定（Maxima）

```
Function IsPrime(x)
    Dim i ,w As Integer
    w=1
    For i = 2 To x - 1
        If Int(x / i) * i = x Then
            w = 0
            Exit Function
        End If
    Next
    IsPrime = w
End Function
```

リスト２ 素数の判定（VBA）

```
Gojohou(x,y):=block(
   local(a,b,r,i),
   a:x,
   b:y,
   if x<y then (a:y, b:x),
      r:1,
      ldisplay([ a, b ] ),
      for i:0 while r > 0
         do(
            r:remainder(a,b),
           ldisplay( [ i,a,b,r ] ),
              a:b,
              b:r
         ),
    return(a)
);
```

リスト３ ユークリッドの互助法（Maxima）

```
Function Gojohou(x, y)
    Dim a, b, r, i As Integer
    a = x
    b = y
    If x < y Then a = y: b = x
    r = 1
    i = 0
    While r > 0
        r = a Mod b
        MsgBox "[ " & i & "," & a & "," & b
& "," & r & "]"
        a = b
        b = r
        i = i + 1
    Wend
    Gojohou = a
End Function
```

リスト４ ユークリッドの互助法(VBA)

上記のリストは、数学Ｂの「数値計算とコンピュータ」での素数の判定とユークリッドの互助法をプログラムしたものである。ExcelのVBAで同じアルゴリズムで作成したものを用意しているので、比較していただけると細かい文法は理解できるものと思う。リスト３の実行例は次のようになる。ldisplay関数によって各変数の値を表示しているので、変数の値の変化を観察することで、プログラムの流れがよく理解できると思う。

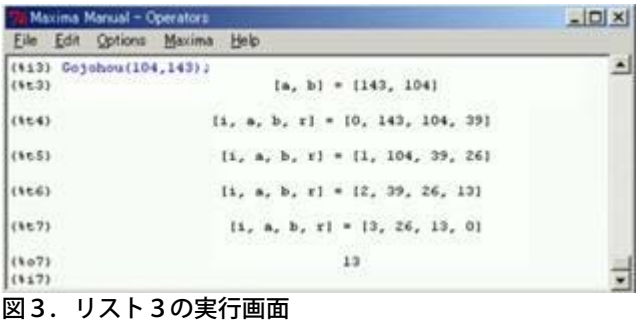

図３．リスト３の実行画面

６ フロントエンドソフトの利用

ＴｅＸｍａｃｓはＬａＴｅＸ文書を見たままに編集することができるエディタであるが、Ｍａｘｉｍａのような数式処理ソフトのフロントエンドとして使用することができる。ＴｅＸｍａｃｓからＭａｘｉｍａを呼び出すには、

「Insert → Session → Maxima」

とすればよい。次の図は、同じ表示をＭａｘｉｍａの表示とＴｅＸｍａｃｓをフロントエンドとして使用した画面である。

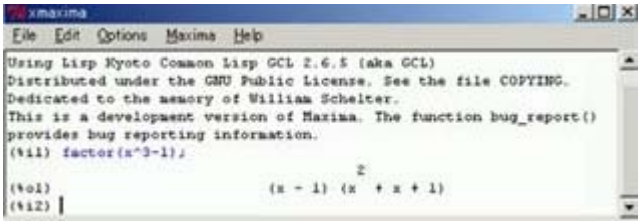

図４．Ｍａｘｉｍａによる表示

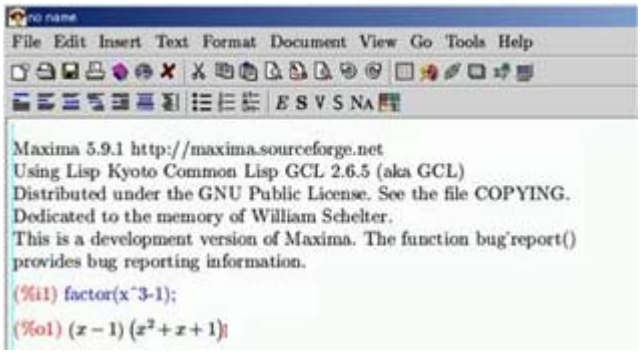

図５．ＴｅＸｍａｃｓをフロントエンドにした表示

　ＴｅＸｍａｃｓでは、数式をＴｅＸのフォントを使用して表示するため、教科書とほぼ同等の数式表示ができる。Ｍａｘｉｍａ以外にＴｅＸｍａｃｓを起動させる分、コンピュータに負担がかかり表示が少し遅くなる傾向があるが、生徒に利用させる場合には、ＴｅＸｍａｃｓをフロントエンドに利用しておきたいところである。

７ まとめと今後の課題

使いはじめてそれほど経っていないが、完成度の高いソフトウェアである。Ｇｒａｐｅｓなどのグラフ描画を目的としたソフトウェアではないため、授業で演示を目的とした利用には不向きである。しかし、数学Ｂの数値計算の分野では、アルゴリズムの説明を行った後、実際にプログラミング言語を用いて体験させる場面が出てくる。その際に、教科書通りのＢＡＳＩＣではなくこのような数式処理ソフトのプログラム機能を用いることも有効でないかと思われる。また、入試問題では解答の流れは作れるものの途中の計算でミスをしてしまい、正解にたどり着かない。特に数学Ⅲの微分・積分の問題では、１つ１つの計算が生徒たちには重い内容であり、計算途中で解答の流れを見失う生徒も少なくない。このような体験を繰り返すことで、数学に対する興味を失う生徒も出てくる。そんな生徒には、途中をＭａｘｉｍａのような数式処理ソフトで計算させ、自分が考えた解答の流れが正しいことを確認させることで、数学に対する興味を失わずに学習を継続することができると思われる。計算力は、他の演習を行うことで養成することになるが、今の理系離れを少なくするための学習法として、一つの方策になるかもしれない。

　まだ、授業での実践が十分とは言えず、学習効果や生徒の反応は確認できないが、今後も利用方法の研究を行い、Ｍａｘｉｍａをはじめとする数式処理ソフトの有効な利用法を研究していきたい。

８ 参考

(1) Maxima Homepage
　　http://www.ma.utexas.edu/users/wfs/maxima.html
(2) Gnuplot Homepage
　http://www.gnuplot.info/
(3) TeXmacs Homepage
　http://www.texmacs.org/