

本稿は [Linux Japan 誌](#) 2001 年 11 月号に掲載された記事に補筆修正したものです。

## Maxima で数学を

今回は数式処理システム Maxima の紹介をします [1][W<sup>3</sup>]. 日本ではあまり知られていませんが、山内さんの紹介ページ [2][W<sup>3</sup>] にもありますように、市販の Maple, Mathematica に負けない高い機能が備わっています。ネットワークにも対応しており、Maxima サーバーにアクセスして電卓代りに使うこともできます。また、最新版では xmaxima という GUI も加わって見栄えも充実してきました (図 1)。

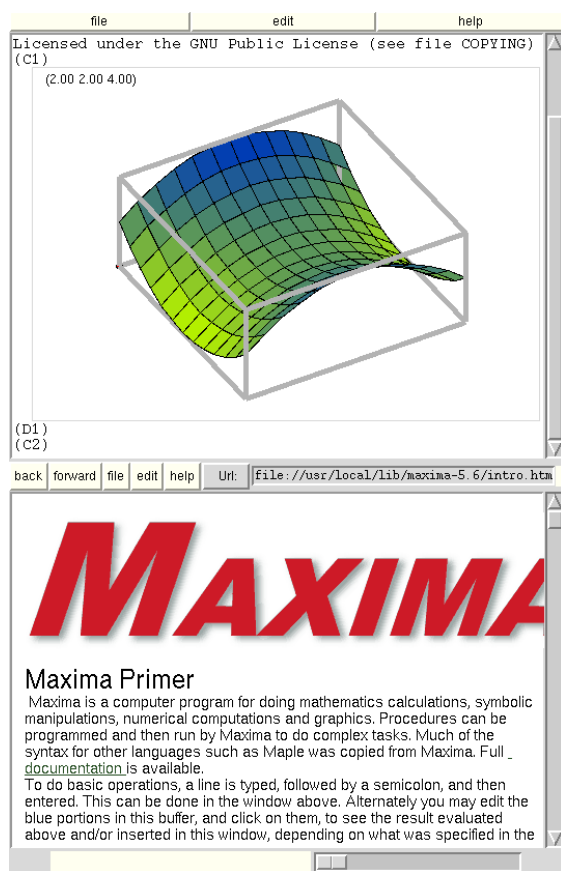


図 1 xmaxima の実行画面

ネットワークを通じて入手可能な数式処理システムは他に MuPAD [3][W<sup>3</sup>] が有名ですし、かなり完成度が高いのですが、無料のライセンスは学生・教育者に限定されています。また、ソースが公開されていないので、ここでは取り上げません。商用であれば、例えば Mathematica や Maple の完成度は非常に高く、特にインターフェース部分において MuPAD は及ばないところがあります。しかし、Maxima のようにソースが公開されているという事は、誰でも開発に参加できるとい

う意味において絶対的な価値を持ちます。数式処理システム全般については、インターネット時代の数学という書籍が出版されています [4]。

### 多機能ぶり

maxima には “Xmaxima Primer” という入門があります。まず

```
$ xmaxima
```

として、xmaxima を起動しましょう。すると図 1 のように xmaxima の下半分の窓に現れます。止めたくなったら quit(); と入力します。

微積分: integrate(), diff()

Primer の文字色が青となっている部分は、maxima で実行可能な命令です。その部分をダブルクリックすると、上部にその実行結果が表示されます。最初は、不定積分です。

```
(C1) integrate(1/(1+x^3), x);
```

不定積分も  $1/(1+x^3)$  ともなると理工系の学生でも記憶してないと思います。筆者ももちろん覚えてません。数学公式集をひっぱり出すしかないのですが (もちろん腕試し、あるいは定期試験対策のため自分で解いてみることは大変良い勉強になります)、該当ページを探すのも一苦勞です。ところが、数式処理システムのシンボリックな処理能力はこの程度の不定積分はものともしません。うだうだいわずにダブルクリックしましょう。図 2 のように上半分の窓 (maxima の実行窓) に、最初の結果 (D1) として表示されます。積分の命令は

```
integrate(f(x), x);
```

です。説明の必要がないくらい簡単明瞭です。次の例題  $\cos x$  の微分も実行しましょう。これは簡単すぎますね、理工系ならば誰でも Maxima に頼らずともできるはずのものです。微分の命令も簡単明瞭で

```
diff(f(x), x);
```

となっています。

定積分は積分範囲指定を付け加えるだけです。

```
(C4) integrate(1/(1+x^2), x, 0, 1);
```

Octave や Scilab でも数値積分は可能です (というよりそれしかできないのですが) が、当然ながら結果は数値でしか表示されません。数式処理システムでは、結

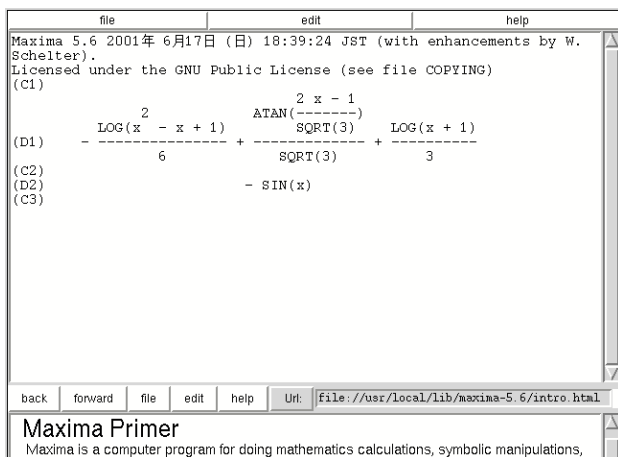


図 2 微積分の実行例

果がある記号定数 ( $\pi$  や  $e$ ) になる場合にはそのように表示されるところが違います。この例では、結果が

$$(D4) \quad \frac{\%PI}{4}$$

となります ( $\%PI$  は もちろん  $\pi$ )。すなわち、

$$\int_0^1 \frac{dx}{1+x^2} \left( = \left[ \arctan x \right]_0^1 \right) = \frac{\pi}{4}$$

がきちんと示されました。

### グラフィック : plot2d(), plot3d()

続いて内部モジュールを使ったグラフィック機能が紹介されます。2次元プロットの例は、 $\sin x$  を 0 から  $2\pi$  の範囲で描くものです。

```
(C5) plot2d(sin(x), [x,0,2*%PI]);
```

グラフ窓の左上隅にカーソルを置くとグラフメニューが現れます。上から 4 番目の Save をクリックすると、ホームディレクトリに sdfplot.ps という名前の PostScript 形式のファイルが作成されます。3次元プロットの例は、

```
(C6) plot3d(x^2-y^2, [x,-2,2], [y,-2,2], [grid,12,12]);
```

$z(x,y) = x^2 - y^2$  を  $x, y$  とともに  $-2$  から  $2$  までの範囲、格子  $12 \times 12$  で描くものです。 $z$  値に応じて彩色された美しい図は左マウスボタンで掴んでグルグル回転させることができます。格子数が  $12 \times 12$  と少ないうちは筆者の Celeron 500MHz マシンでも滑らかに動きます。 $30 \times 30$  にすると (入門にある命令分は編集可能ですから、`[grid,30,30]` と書き換えて再度ダブルクリックしてみてください)、さすがにつらいらしく、操作している間は面素の大部分が消えて枠 (直方体) が回転

します。回転を停止するとやや間を置いて全面素が再び表示されます。この辺りの感覚は GNUPLOT に良く似ています。

### 関数の極限 : limit()

シンボリックといえば無限大 ( $\infty$ ) が扱えなければ話になりません。解析学における関数の極限の重要性は言うを待ちません。

```
(C7) limit((2*x+1)/(3*x+2), x, inf);
```

$\infty$  だけでなく、 $\frac{0}{0}$  も正しく扱います。

```
(C8) limit(sin(3*x)/x, x, 0);
```

### 任意精度計算 : FPPREC, bfloat()

この連載で gp(/PARI) や Calc などの任意精度計算ツールを取り上げましたが、数式処理システムでは当然のように実装されています。maxima でシンボリックな結果を倍精度実数に数値化して表示するには、

```
float(%pi);
%pi,number;
```

としますが、任意精度実数に数値化する場合は bfloat() を用います。精度桁数は FPPREC (大文字小文字を区別しないので fpprec も可) に与えます。

```
(C9) block([FPPREC:100], bfloat(%pi));
```

この例では、FPPREC の変更指定が外に及ばないように block で囲んでいます。

また、FPPREC に関係なく整数演算は桁が全て保持されます。例えば

```
100!;
2^1000;
1/50!;
```

と実行してみてください。任意精度計算ツールでも説明しましたが、任意あるいは多倍長精度と倍精度の変換がいつ行われるかを見極めないと失敗します。例えば、次の後ろの 3 つのコマンドの結果の違いを比べてください。

```
FPPREC:50;
bfloat(sin(1.0));
bfloat(sin(bfloat(1.0)));
bfloat(sin(1));
```

方程式の解 : solve()

次には、筆者にも解ける簡単な代数方程式を解いて入門の最初の節が終わります。

```
(C11) solve([x+y+z=5, 3*x-5*y=10,y+2*z=3],[x,y,z]);
(C12) solve(x^2-5*x+6=0,x);
```

複素係数 (虚数単位  $i$  は %I) も受け付けます。

```
(C?) solve(x^2+2*x+%i=0,x);
(D?) [x = - SQR(1 - %I) - 1, x = SQR(1 - %I) - 1]
```

もちろん非線形あるいは超越方程式まで含めて任意の方程式の解が solve() で与えられるわけではなく、自分でプログラムを組まなければならない場合もあります。

例えば超越方程式

$$e^x + x - 2 = 0$$

は、solve では

```
(C1) solve(exp(x)+x-2=0,x);
(D1) [x = 2 - %E]
```

となって、ちょっと唖然としてしまいます。数値解ならば、Newton 法を用いることができ、モジュールとなっています。それを load して使ってみましょう。初期値と終了誤差を与える必要があります。

```
(C2) load("newton");
(D2) /usr/local/lib/maxima-5.6/share/newton.mc
(C3) newton(exp(x)+x-2,x,0,10^(-6));
(D7) 0.44285470382975
```

入門の次の節は線形代数 (行列) に関するもので始まっています。しかし、この紹介は数値を与えてしまっており、数式処理システムの機能を十分に表現していません。そこで、少しだけ、内容を変更して実行させましょう。青色の命令文分を編集してダブルクリックします。

### 行列演算

まず、行列を生成する matrix() コマンドです。デフォルトでは数値ですが、一部を変数にしてしましましょう。

```
(C13) A:matrix([1,2],[3,a]);
(C14) B:matrix([1,b],[1,1]);
```

以降はそのまま実行させます。

```
(C15) A+B;
(C16) A.B;
(C17) A^-1
```

最後の行列 A の逆行列を求めるコマンドを実行すると、

次のように要素に変数が入った形で表示され、まさにシンボリックな処理がなされていることが判ります。

```
(D17) [ [ a 2 ]
        [ ---- - ---- ]
        [ a - 6 a - 6 ]
        [ ]
        [ 3 1 ]
        [ - ---- - ---- ]
        [ a - 6 a - 6 ] ]
```

もちろん固有値問題  $Ax = \lambda x$  もシンボリックに求められます。

```
(C93) A:matrix([1,a],[a,1]);
(D93) [ [ 1 a ]
        [ ]
        [ a 1 ]
(C94) eigenvectors(A);
(D94) [[ [1 - a, a + 1], [1, 1]], [1, - 1], [1, 1]]
(C95) x:transpose([1,-1]);
(D95) [ [ 1 ]
        [ ]
        [ - 1 ]
(C96) (1-a)*x; A.x;
(D96) [ [ 1 - a ]
        [ ]
        [ a - 1 ]
(C97) [ [ 1 - a ]
        [ ]
        [ a - 1 ]
(D97) [ [ 1 - a ]
        [ ]
        [ a - 1 ] ]
```

となり、確かに固有値  $(1-a)$  と固有ベクトル  $([1, -1]^t)$  が求められています。

### 常微分方程式

簡単な常微分方程式を解くことができるのも、数式処理システムの特長でしょう。Maxima では 2 階まで扱っています。

```
(C23) ode2('diff(y,x)+3*x*y=sin(x)/x, y,x);
```

を実行させると (これは  $y' + 3xy = \sin x/x$  という常微分方程式です)

```
(D23) y = %E [ [ 2
                2 3 x
                ---- / ----
                3 x 2
                2 [ %E SIN(x)
                (I ----- dx + %C)
                ] x
                / ]
```

を得ます。何となく判ると思いますが、今の結果を、TeX の数式ソースを出力するコマンド tex() を使って出力させると、図 3 のようになります。

```
(C24) \tex(ode2('diff(y,x)+3*x*y=sin(x)/x, y,x));
$$$y=e^{-\frac{3}{2}x^2}\over{2}}\left(\int e^{\frac{3}{2}x^2}(\sin x)\over{2}}\right)\left(\frac{\sin x}{x}\right)dx+C\right)$$$
```

図 3 L<sup>A</sup>T<sub>E</sub>X の数式ソースを出力するコマンド tex() による出力文字列 .

これを TeX のソースに取り込んで処理すると (%C の % は TeX では、それ以降をコメントしてしまう記号なので、除かなければなりません) .

$$y = e^{-\frac{3}{2}x^2} \left( \int \frac{e^{\frac{3}{2}x^2} (\sin x)}{x} dx + C \right)$$

となって、解の関数形がはっきりします .

入門の最後は、関数の定義と block() による、局所変数を用いた複雑な関数あるいは手続きの宣言についてです .

関数の定義 : block

関数の定義も簡便です . 最初は単文定義する場合です .

```
(C27) f(x):=x+2;
(C28) f(3);
```

続いて、複数の文を並べる場合には block() で囲む必要があることを示しています .

```
(C31) g(x):=block([u:x+3,w], u:u^2, w:(y+2)^2, u+w);
(C32) g(2);
(D32) (y + 2)^2 + 25
```

block 内部には条件文も含めることができます . 入門の例は、if 文の条件がちょっとおかしいと思えますから、つぎのように書き換えて実行させましょう .

```
(C33) h(x):=block([u:x+3,w],
if(u<6) then w:(T+2)^2 else w:(F+2)^2+1,u^2+w);
(C34) [h(3),h(2)];
(D34) [(F + 2)^2 + 37, (T + 2)^2 + 25]
```

for ループは C と少し用語が変わっていて

```
for i:1 thru 10 do (.....)
```

です . 最後にヘルプです . xmaxima のどちらのウィンドウメニューにも help がありますが、項目を直接指定して調べることができます .

```
(C38) describe("log");all;
```

## さらに馴染むには

入門が終わったら、サンプルスクリプトやデモを動かしてみましょう . /usr/local/lib/maxima5.\*/ 以下に share,share1,share2 等があって、そこにモジュール \*.mc やデモ \*.dem があります . サンプルスクリプトを実行させるには、

```
example(ODE2)
```

とします . 何も指定しないと、example 可能な項目が全て表示されます . これは一期に最後まで実行してしまうので、スクロールバックしないと理解できません . そこで、1 つのコマンド実行毎にキー入力を求めてくる、demo("...") を実行させてみましょう . ただし example と共通ではありません .

```
demo("cf.dem");
```

最初にして最後は、お決まりの manual です . /usr/local/lib/maxima 5.\*/doc ではなく /usr/local/lib/maxima5.\*/info に TeX のソースがありますから、make します . エラー終了しますが、250 ページを超える maxima.dvi がほぼできあがります . 前回紹介した dvi2pdf で PDF 化すると 1MB 強の大きさが 760K になります . TeX はどうもというならば、HTML 版 maxima.html も同じディレクトリに展開されてあります .

## Geomview

maxima の内部グラフィックは筆者は結構気に入っていますが、細かい制御が良くわかりません . 辛い使い慣れた外部ツールに描画させることができます . いわずもがな gnuplot です .

```
plot3d(sin(x)*sin(y)/x/y, [x,-6,6],
[y,-3,9], [PLOT_FORMAT,GNUPLLOT]);
```

しかも嬉しいことに、図 4 のような GUI フロントエンド (Tcl/Tk) が一緒に立上ります . 正式公開バージョン 3.7.1 のユーザーにとっては、回転も自由ですし、1 行コマンドエディタもあり、親切設計といえましょう .

レンダリング画像 (しかも動く) が欲しいという場合には、Geomview[5][W3] に描画させることも可能です . Geomview は OpenGL ライブラリを用いた 3 次元描画ツールです . OpenGL 互換の MesaGL の他に Motif(Lesstif) も必要で、動くようにするにはやや腕力が必要かもしれません . しかし MesaGL は結構ポピュラーとなってきており、どの配布系でも探せばありま

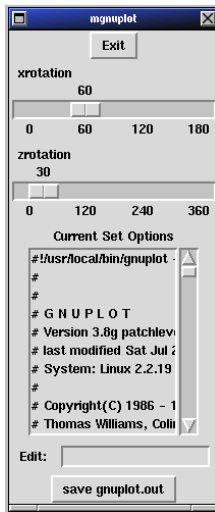


図 4 xmaxima から gnuplot を呼出した際に現れる，GUI フロントエンド。

すし，Motif は Open になったので，Geomview のダウンロードページからも持ってこれます（ただし，\*.tgz）。これらのライブラリをヘッダーも含めてインストールしておき，ソースファイル geomview-1.8.1.tar.gz をコンパイル作業用のディレクトリで展開して，configure 以下コンパイルします。筆者が Plamo2.2 で行った作業は，

1. MesaGL という名前では認識してくれないので GL ライブラリとしてリンクを作成した。

```

ln -fs libMesaGL.so libGL.so
ln -fs libMesaGLU.so libGLU.so
ldconfig
  
```

2. configure で Motif のパスを指定した（geomview のバイナリパッケージを使用）

```
./configure --with-motif=/openmotif
```

これで，make; make install が滞りなく終了しました。後は，geomview への実行パス /usr/local/Geomview/bin を通しておきます。

さて maxima から geomview をグラフィックビューアに指定するには，gnuplot と他は同じで，次のように PLOT\_FORMAT に GEOMVIEW を指定します。

```
..., [PLOT_FORMAT, GEOMVIEW];
```

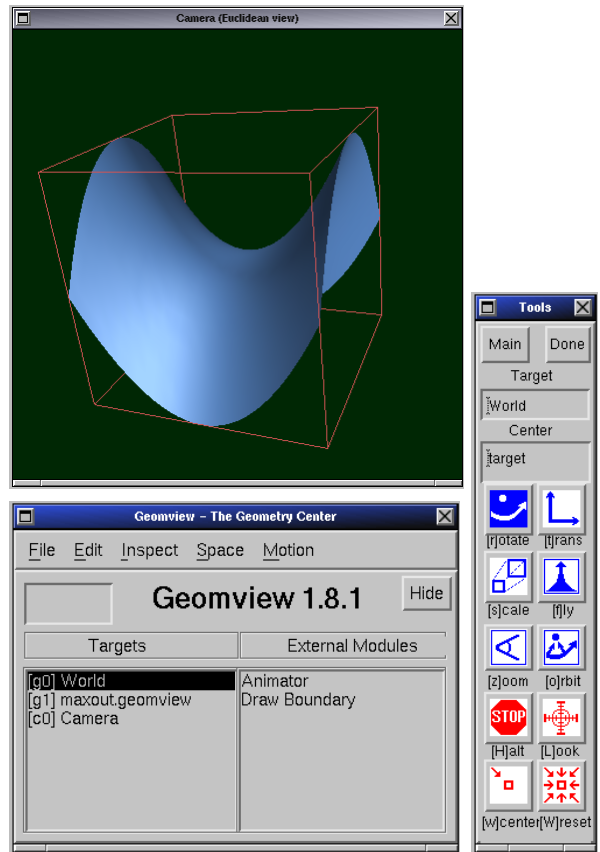


図 5 Geomview の起動時には 3 つの窓が現れる：メイン（下左），Tools（下右），Camera（上）。メインの Inspect から Appearance を選択するとオブジェクトの描画属性が変更できる。

## 参考文献

- [1] Maxima 本家 <http://www.ma.utexas.edu/users/wfs/maxima.html>
- [2] 日本語化パッチを作成している山内さんのページ <http://phe.phyas.aichi-edu.ac.jp/~cyamauch/maxima/>
- [3] Mupad の販売元 <http://www.sciface.com/>
- [4] 野寺隆志 他：インターネット時代の数学シリーズ 1 「これだけは知っておきたい数学ツール」，共立出版（1999）
- [5] Geomview の公式サイト <http://www.geomview.org/>