Software Review
ソフトウェアレビュー

数式処理ソフトウェア Maple 9
―シミュレーション教材作成ツールとしての利用―

Maple 9

名古屋大学大学院情報科学研究科　**中村泰之**

◎**Key Words**　数式処理，シミュレーション，アニメーション

1　はじめに

Maple はカナダの Waterloo 大学[1]での研究をもとに，Waterloo Maplesoft 社[2]が開発を行っている数式処理ソフトウェアであり，日本ではサイバネットシステム[3]が販売している。このたび，新バージョンである Maple 9 がリリースされ，Mac OS X にも正式に対応したことから，利用を検討しているユーザーも多いのではないだろうか。同種のソフトウェアとしては Mathematica[4]や MuPad[5]などがあるが，それらとの詳細な比較や，Maple の機能に関する網羅的な紹介は他所にゆずり，ここでは Maple 9 の新機能の一部を紹介するにとどめ，筆者が主にシミュレーション教材作成ツールとして Maple 9 を利用した場合のレビューを中心に行いたい。

2　2 つのインタフェースと新しいヘルプシステム

Maple 9 では，Standard と Classic の 2 つのインタフェースが用意されており，Classic インタフェースは Maple 8 のインタフェースと同様であるが，Standard インタフェースは一新されている（Fig. 1）。Maple 8 からバージョンアップしたユーザーや，メモリの少ないコンピュータ上で使用する場合には Classic インタフェースのほうが馴染みやすく，利用しやすいであろう。起動時間も短く軽快である。それに対して，さまざまな便利な機能を積極的に活用するためには，Standard インタフェースを利用すべきであろう。その便利な機能の 1 つとして，新しいヘルプシステムのインタフェースがある。Maple のヘルプは以前から非常に充実しており，各種パッケージ，関数の利用方法が例とともに詳しく掲載されている。そして今回は，検索機能，履歴表示機能が加わり，表示も別ウィンドウで見やすくなっており，大変便利な新ヘルプシステムとなっている。さらに完成度が高まったと思われる。

Fig. 1　Standard インタフェース（上）と Classic インタフェース（下）

Contact to：ynakamura@is.nagoya-u.ac.jp

3　微分方程式の数値解法

シミュレーション教材を作成するにあたっては，微分方程式を数値的に解くことが必要になる場合が多い。微分方程式の数値解を求めるのであれば，C 言語や Fortran などで計算することのほうが速度的な面で有利であり，一般的であるが，数式処理ソフトウェアを用いることの利点もある。たとえば，次式で表されるローレンツ方程式を解くことを例として取り上げてみよう。

$$\frac{dx}{dt}=-\sigma x+\sigma y,\quad \frac{dy}{dt}=-xz+rx-y,\quad \frac{dz}{dt}=xy-bz.$$

パラメータ $(\sigma, r, b)=(10, 28, 8/3)$，初期条件 $x(0)=1.0$，$y(0)=z(0)=0.0$ のとき，Maple では数値解の計算から $(x(t), y(t), z(t))$ の軌道の描画まで，一連の作業を Fig. 2 のように非常に簡単に実行することができるのである。

微分方程式を解くための dsolve コマンドでは数値解を求めるためのさまざまなアルゴリズムを指定することもでき，教育的効果も期待できる。そして，Maple 9 では，上記のようなコマンド形式ではなく，ODE アナライザというグラフィカル・ユーザー・インタフェース（GUI）上で対話形式により微分方程式を解く機能が追加されており，初心者にも取り組みやすい工夫がなされている（Fig. 3, 4）。ODE アナライザでは Maple のコマンドを表示する機能もあり（Fig. 4 内右下部分），コマンドの利用方法を学ぶこともできる。

ここで，微分方程式の数値解析と解軌道の描画に関し

```
> ode:= diff(x(t),t) = -sigma*x(t) + sigma*y(t),
        diff(y(t),t) = - x(t)*z(t) + r*x(t) - y(t),
        diff(z(t),t) = x(t)*y(t) - b*z(t):
  sigma := 10: r := 28: b := 8/3:
  ini:=x(0)=1.0, y(0)=0.0, z(0)=0.0:
  sol:=dsolve({ode, ini}, numeric):
  plots[odeplot](sol, [x(t), y(t), z(t)], t=0..50,
                 numpoints=4000, axes=frame);
```

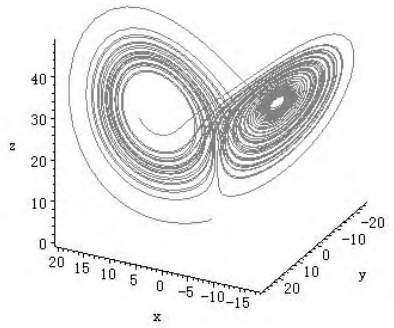

Fig.2　ローレンツ方程式の解析

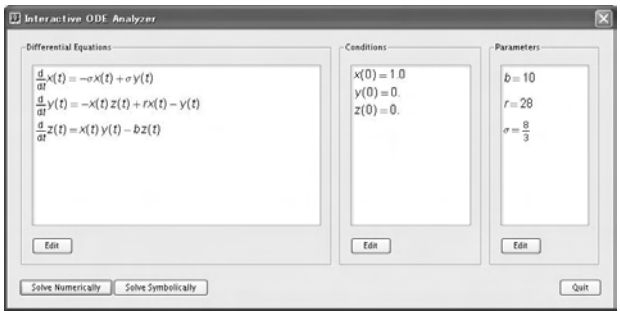

Fig.3　ODE アナライザによる微分方程式の定義

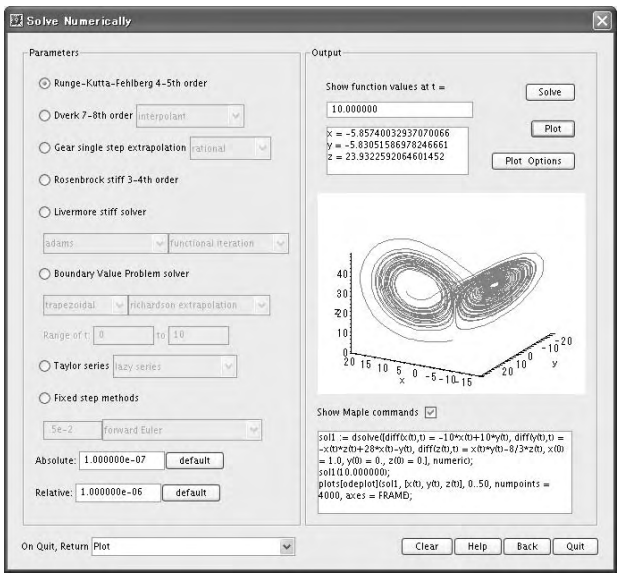

Fig.4　ODE アナライザによる微分方程式の数値解法の設定

て Mathematica との比較を簡単にコメントしておく。どちらも数値解析に関しては，さまざまなアルゴリズムを選択でき，コマンド書式の複雑さに関する違いもなく，優劣について言及することは難しいが，数値解の描画に関しては，三次元グラフ（画像）のマウスによる回転操作が可能であることなどから，直感的な操作性などの面で Maple のほうが若干優れていると言える。Mathematica でも同様の操作は可能であるが，バージョン 4.0 以降で RealTime 3 D パッケージを利用する必要があり，またその場合には座標軸が表示されなくなる。そして Mathematica では，Maple の ODE アナライザに相当する微分方程式の解析用 GUI インタフェースも現在のところ用意されておらず，Maple のほうがより初心者が取り組み安い工夫がなされていると言える。ただし，以上のことは Maple と Mathematica のごく一部の機能に関する比較を行ったにすぎず，数式処理ソフトウェアとしての総合的な優劣は，好みや利用目的などから利用者ごとに異なってくるはずである。

4　Maplets パッケージを用いたシミュレーション教材の作成

Maplets とはボタン，テキストエリア，プルダウンメニュー，スライダーなどのツールをウィンドウ上に配置し，それによって Maple の機能を利用することのできるアプリケーション（Maplet アプリケーション）を作成するためのパッケージである。Maple 7 でアドオン・パッケージとして初めて提供され，Maple 8 では標準の機能となり，今回の Maple 9 ではさらなる改良がなされている。GUI の各要素は Java の Swing API[6]により実現されているが，Java の知識は必要とはしておらず，GUI アプリケーションを作成するための有効な手段の 1 つと考えられる。先に紹介した ODE アナライザも Maplet アプリケーションの 1 つである。

筆者は，比較的容易に GUI ソフトウェアが作成できる手軽さと，豊富な数値計算アルゴリズムを利用できる便利さから，シミュレーション教材を作成するために Maplets を利用することはよい手段の 1 つではないかと考えている[7]。Fig. 5 に Maplets を用いたシミュレーション教材の例を示す。単振り子，波のシミュレーションであるが，それぞれパラメータ，初期条件を GUI コンポーネントから設定でき，解析解，数値解をプロットエリアでアニメーションとして表示できるようにしている。どれも，シミュレーションの開始，停止，コマ送りが可能で，波のシミュレーションでは三次元（3 D）表示も実現している。もし Java でこれと同等の教材を作ろうとすれば，膨大なコード数を必要とし，ある程度のプログラミング能力も要求されるであろうし，3 D 表示

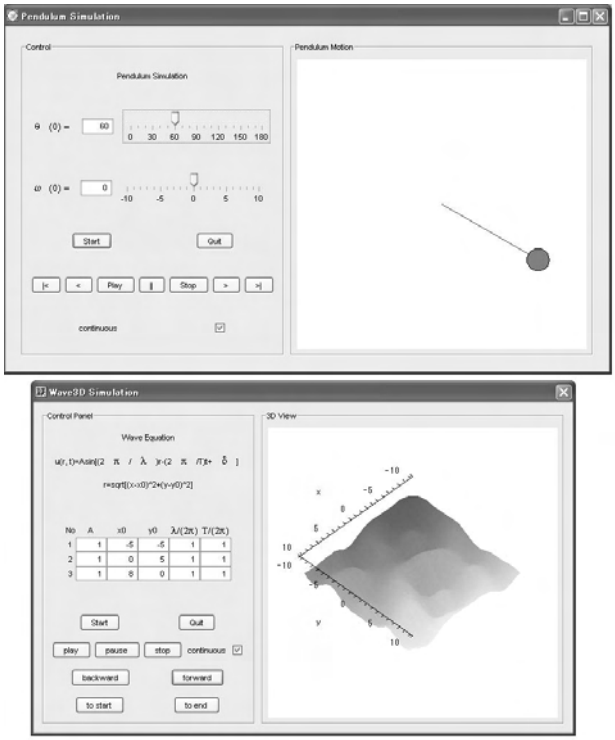

Fig.5　Maplets による教材例

には Java 3 D[8]などの技術を導入しなければならない。それに比べて Maplets を利用すれば，はるかに少ないコード数で，比較的容易に教材作成が可能である。Java プログラミングの経験がある方は，同様の教材を Maplets で作成し，その容易さを実感してみていただきたい。

Maplets も Maple 9 で歓迎すべき改良がなされた。プロットエリアでのアニメーションが可能になったという意味で，シミュレーション教材のために Maplets が実用的になったのは Maple 8 からだと思うが，Maple 9 では，Maplet アプリケーション内で 3 D 画像のマウスによるリアルタイムな回転操作が可能になっている。シミュレーション結果を視覚的に理解するためには，重要な改良であると言える。もちろん，Maple 8 でも回転操作は可能であったのだが，それを実現するためにプロットエリアの表示角度を，スライダー操作や角度の入力によって変更できるようなプログラミングが必要であった。

一方，作成の容易さとひきかえに，欠点もいくつか認めなければならない。まず，Java によるプログラミングに比べて，細かな設定が不可能であるということがあげられる。たとえば，スライダーの値をマウス操作により連続的に変更できない点などである。スライダーの値の変更結果は，マウス操作が終了した（マウスから指が離れた)時点でのスライダーの値が返されるだけであり，これでは Fig. 5 の振り子のシミュレーションにおいても，初期の振り子の位置をスライダーの動作と連動して連続的に動かしながら設定することができないのである。また，シミュレーション教材としてはある意味で致命的であるのだが，リアルタイムのシミュレーション（アニメーション）が困難であるという点があげられる。これは，Maplets というより Maple が行う計算方法に関わることであるのだが，通常，ある時刻から指定された時刻までの計算をすべて行ってから，その計算結果を使ってアニメーションを行うのである。したがって，長い時間のシミュレーションであれば，スタートボタンを押してから，実際にアニメーションが開始されるまでタイムラグがあるのである。短い時間ステップごとに計算し，その結果を描画しながらアニメーションを行うということも不可能ではないが，描画速度の面で劣ることは否めない。

5　その他の改良点

その他にも，Maple 9 で追加・改良されたさまざまな機能があるが，その一部を紹介したい。

5.1　OpenMaple による Maple と他言語との連携

他言語との連携に関しては，従来の C 言語や Java などに加え，MATLAB,Visual Basic のコード生成機能が加わった。また，Maple から C 言語でコンパイルされたネイティブコードや Java の class ファイルを呼び出すことはこれまでも可能であったが，Maple 9 では OpenMaple と呼ばれる新しい API が導入され，C 言語から Maple の計算エンジンにアクセスすることが可能となった。これらの新機能により，Maple と他のプログラミング言語との連携の自由度がさらに増えてきたと言える。また Java の JNI を利用すれば，Maple エンジンへのアクセスが記述された C 言語によるネイティブコードを Java から呼び出すことにより，間接的に Java から Maple へのアクセスも実現できる。なお，Java から Maple の計算エンジンにアクセスするには，MapleNet[9]を利用することのほうが，より直接的であり有効な手段である。

5.2　高速フーリエ変換の改良

時系列解析などでは離散データのフーリエ変換を行う場面がよくでてくるが，計算時間を短縮するために，高速フーリエ変換が一般的に利用される。Maple 8 でも FFT という関数により実行可能であったが，データ数が 2^m 個の場合にしか適用できなかった。しかし，Maple

```
> restart;
> m:=13:
> Z:=Vector(2^m,i->evalf(exp(I*i/3)));
                    Z := [ 8192 Element Column Vector
                           Data Type: anything
                           Storage: rectangular
                           Order: Fortran_order ]
> x := array(1..2^m):
  y := array(1..2^m):
  for i to 2^m do x[i]:=Re(Z[i]): y[i]:=Im(Z[i]): end do:
> st:=time():
  FFT(m, x, y):
  time()-st;
                              14.231
> with(DiscreteTransforms):
> st:=time():
  Z1:=FourierTransform(Z, 2^m):
  time()-st;
                               0.230
```

Fig. 6　高速フーリエ変換の速度比較

9 では新しく DircreteTransform パッケージが提供され，FourierTransform 関数により，任意の個数のデータの変換も可能になったばかりでなく，より高速なフーリエ変換が可能となった。Pentium Ⅲ 750 MHz，メモリ 768 MB，Windows XP Home Edition のコンピュータ上で，約 8,000 個のデータをフーリエ変換することにより，計算速度を比較したのが Fig. 6 である。FourierTransform 関数を利用することによる，劇的な計算速度の向上を伺うことができる。時系列予測のシミュレーション教材作成などには有効であろう。なお，ここでは従来の FFT 関数を使って比較するために，データ数は 2^{13} 個としている。

5.3　グラフィック機能の改良

Maple 9 ではグラフィックス・オブジェクトの透明度を設定できるようになった。これは OepnViz[10] という技術を導入することにより実現されたものである。特に，3 D グラフィックスに関して視覚的な面で表現の幅が広がったと言える。なお，この表現はワークシート上では Standard インタフェースのみで可能であるが，Maplet アプリケーション上であれば，それを実行するインタフェースは問わない。

6　今後改良を望む点

Maple 9 でさまざまな機能が追加・改良され，筆者が主に利用しているシミュレーション教材の作成にとっても，大変便利になったが，いくつか今後改良を望む点もある。まず，今回のバージョンアップにより Maplet アプリケーション内で 3 D オブジェクトのマウスによる回転操作が可能になったが，今後はマウスによるズーミング操作（拡大・縮小），あるいは視点の移動が可能になればありがたい。これにより，さらに表現の幅が広がると考えられる。また，3 節で紹介した ODE アナライザで微分方程式を設定する場合（Fig. 3），初期値やパラメータの設定に π（Maple では Pi）などの定数を用いることができず，evalf コマンドで数値的に評価しておかなければならないようである。数値よりも π などを使ったほうがわかりやすい場合もあるので，改良できないものであろうか。そして，Maple 9 の日本語版であれば，新ヘルプシステムにおける検索機能で日本語による検索も可能にすれば初心者にも優しいものになると思う。

7　おわりに

Maple 9 について，やや偏った観点からではあるがレビューを行い，あらためてシミュレーション教材作成ツールとしては大変便利なソフトウェアであるとの実感を深めた次第である。MapleNet などと併せて利用すれば，e–Learning システムの構築も利用できると考えられ，大変魅力的である。ユーザーによりそれぞれ利用方法，求める機能が異なるであろうから，ぜひ体験版などを通して一度実際に利用してみることをお勧めする。

8　謝辞

今回のレビューを行うにあたり，サイバネットシステム株式会社より，日本での Maple 9 の販売に先がけて Maple 9 英語版を利用する機会を与えていただきました。感謝いたします。

参考文献

[1] http://www.scg.uwaterloo.ca/SCG/
[2] http://www.maplesoft.com/
[3] http://www.cybernet.co.jp/maple/
[4] http://www.wolfram.com/
[5] http://www.mupad.de/
[6] http://java.sun.com/products/jfc/
[7] 中村泰之，「数式処理ソフトウェアを用いた物理シミュレーション教材」，大学の物理教育，2003–2（2003）pp. 61–64
[8] http://java.sun.com/products/java-media/3D/index.jsp
[9] http://www.maplesoft.com/maplenet/
[10] http://www.openviz.com/