WEB WATCHER

数値計算用フリーソフトを使ってみた

皆さんは，コンピューターを使って光学関係の数値計算シミュレーションをする場合，どのようなプログラム言語をお使いですか．私は，大型コンピューターの Fortran から始め，その後エンジニアリングワークステーションに移ってからは C 言語，最近では Matlab も一部利用しています．C 言語は， 数値計算だけであれば高い汎用性を有しており，たとえコンピューターの CPU が異なっている場合にも再度コンパイルするだけで実行できます．したがって，同じソースコードを使い，パラメーターだけを変更したものを複数のコンピューターで実行することにより計算時間の短縮が可能です．一方，Matlab は計算速度こそ速くありませんが，多くの数値計算ライブラリーを備え，柔軟なプログラムの記述を可能にし，短期間に科学技術計算プログラムを開発できます． Matlab を一度使うと，C 言語よりさらに使いやすいことを実感します．おもな特徴としては，次のようなものがあります．

- 複素数の変数が用意されている
- 計算結果をグラフィックとして簡単に表示できる
- 信号処理などの関数があらかじめ用意されている
- 対話形式とプログラムによる実行が可能である
- 行列演算を基礎とした簡素なプログラミングが可能である

Matlab は C 言語を理解できていれば，習得は容易です．加えて，C 言語より少ない記述で同じ計算を実行できるため，非常に優れた言語といえます．ただし，Matlab は有料で，残念ながら価格もパソコンソフトを買うようなわけにはいきません．

ところで，Scilab という言語をご存じですか．Scilab は 1990 年から INRIA[1] により開発されている，Matlab と非常によく似た言語です[2]．うれしいことに無料で使用でき，Windows, Linux, Solaris, HP-UX, MacOS などの実行プラットホームがあります．以下では，実際に Scilab を使った事例を紹介しながら，Matlab や Scilab の特徴を説明します．なお Scilab のバージョンは 3.1.1，実行プラットホームには MacOS X (10.4.3) を使用しています．

光の干渉やホログラムを計算する場合，光の波動性の記述には，複素数による変数の取り扱いを必要とします．先に述べたように C 言語では複素数の変数がないため，実数部と虚数部にそれぞれ変数を割り当てて，複素数計算のプログラムを記述する必要がありました．しかし，Matlab や Scilab では，複素数の変数が用意されているため，簡素でわかりやすいプログラム記述が可能です．まずは簡単な例を紹介しましょう．

```
-->a=0.1+0.2*%i
 a  =
    0.1 + 0.2i
-->b=0.3+0.4*%i
 b  =
   0.3 + 0.4i
-->a*b
 ans  =
  - 0.05 + 0.1i
```

これは複素変数 a, b の乗算を対話式で行ったものです．Scilab では虚数を %i で表します．虚数を含む数値を入力した時点で，変数は複素変数になるため，あらかじめ定義しておく必要はありません．また，言語自体が複素数をサポートしているため，例えば振幅を計算する場合などは，abs 関数により，簡単に求めることができます．

```
-->abs(a)
 ans  =
   0.2236068
```

このような対話式の実行はプログラム終了後にも可能で，実行後，変数の値の確認や，さらに対話的に次の処理を実行することもできます．

次に，以下の行列変数 m を考えます．

$$m=\begin{bmatrix}1 & 4\\ 3 & 6\end{bmatrix}$$

これは以下のように記述します．

```
-->m=[1 4;3 6]
 m  =
!   1.    4. !
!   3.    6. !
```

このように，行列を取り扱う場合にあらかじめ行列に対応した配列を確保しておく必要はなく，テータが与えられると，メモリーにその領域を確保します．これは，実験データの処理などでデータ数が変化した場合でも，プログラムの変更や配列の大きさを与えることなく処理が可能です．また，複素数と同様に行列の四則演算なども簡単な記述で可能ですが，これについては専門書に譲ることにしましょう．

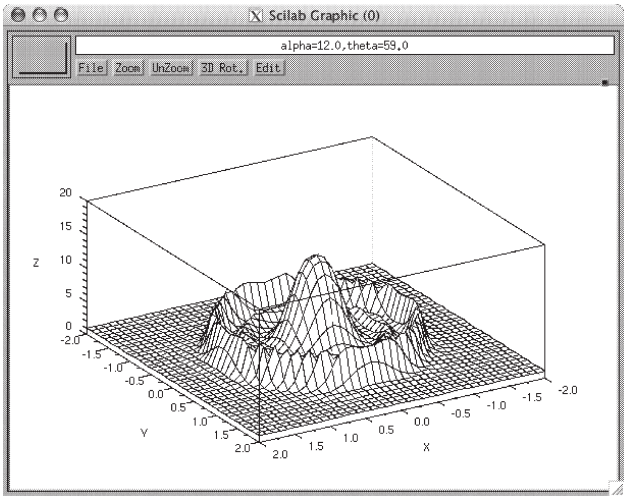

図1　三次元グラフ．

シミュレーションによっては，計算結果をグラフとして表示したい場合があります．C言語ではデータ出力はテキストが基本となっているため，計算結果の三次元データなどを確認するには，計算結果をファイルに出力し，さらにそれを読み込み処理するための別のソフトウェアーが必要です．Scilabでは，二次元や三次元データをグラフ化し可視化するための関数があります．あらかじめ用意されているmilk_drop関数を使い，三次元グラフを表示した例を紹介します．

```
x=-2:0.1:2; y=x;
z=eval3d(milk_drop,x,y);
plot3d(x,y,z)
```

図1が表示された三次元のグラフです．このように，三次元グラフの関数にx，y，zの値を与えるだけで，簡単にグラフ表示が可能です．加えて，“3D Rot.”のボタンを押せば，グラフの回転，傾き，Z軸の長さの変更ができるため，計算結果を調べるには非常に便利です．グラフはPostScriptやgifなどの形式で保存もできるため，論文への利用も可能でしょう．また，このプログラムはScilabのエディターから実行したものです．このエディターはデバッガーなどの機能を備えているため，プログラムを記述しながら，プログラムの動作を確認できます．

C言語では，数値計算用の関数を自前で用意する必要がありました．私も一度，参考文献に従って光学的な高速フーリエ変換[3]の長いプログラムをC言語で作った経験があります．しかし，Scilabでは数値計算の関数が充実しています．例えば，フーリエ変換に関していえば一次元，二次元のフーリエ変換に加えて多次元のフーリエ変換までも用意されています．この他にも，共分散や相関などの信号処理でよく使う関数もあります．

また，参考文献の中に面白い記述がありましたので，それを私のコンピューターで実行してみました．

```
tic;
for j=1:1000
  for k=1:1000
    a(j,k)= j+k;
   end
end
toc
```

tocはticからの時間を計測します．C言語に慣れた人はついこのようなループで計算したくなると思います．このプログラムを実行しますと，私のコンピューターでは58秒かかります．このプログラムを次のように変更します．

```
tic;
a=zeros(1000,1000);
k=1:1000;
for j=1:1000
    a(j,k)= j+k;
end
toc
```

計算時間は大幅に短縮され，実行時間は0.16秒になりました．この種の言語では，変数のメモリーサイズを固定しておき，ベクトル演算を利用することで，計算時間を短縮できるようです． 今後，MatlabやScilabで数値計算のプログラムを作られるときは，頭の片隅にでも留めておいていただければと思います．

ScilabはMatlabと若干記述が異なる部分があります．また，Matlabにはニューラルネットワークや画像処理などの追加できるライブラリーがあります．しかし，追加ライブラリーなどが必要な場面はそう多くはなく，Scilabも十分，研究用のシミュレーターとして利用できるものと思います．なお，ここでは紹介できませんでしたが，インターネット上には日本語によるScilabの利用方法や関数などの解説も数多くありますので，参考にされることをお勧めします．

この記事に関するお問い合わせは，kadono@mech.saitama-u.ac.jp，もしくはhayasaki@opt.tokushima-u.ac.jpまでお寄せください．

（広島県立東部工業技術センター　広川勝久）

文　　献

1）http://www.inria.fr/
2）http://www.scilab.org/
3）辻内順平，村田和美：光学情報処理（朝倉書店，1974）．
4）櫻井　鉄：Matlab/Scilabで理解する数値計算（共立出版，2003）．