

数値計算ツール Octave, FreeMat & Scilab を使ってみよう

総合情報基盤センター 准教授 布村 紀男

1. はじめに

商用ソフトウェアの MATLAB を意識して作られた行列演算, グラフ機能パッケージとして Octave, FreeMat, Scilab などが Web を通じて入手し, 手軽に試してみることができる.

Wikipedia^[1,2]によれば Octave とは, MATLAB と互換性をもったフリーの数値解析ソフトウェアであり, GNU によって開発されているとある. また, FreeMat も無料で利用できるソフトウェアである. 数値計算環境, プログラミング言語を提供し, 多くの MATLAB 関数のサポートに加えて, いくつかの GUI 機能も備わっている.

一方, Scilab は, フランスの INRIA と ENPC で開発された数値計算システムで, 数値計算機能以外に信号処理, 行列や多項式の数式処理, 関数のグラフィックなども充実しており, コマンドは MATLAB に似ているがこちらは前の2つと違って, 互換性がない. これもフリーウェアであり, 現在, 英語版, フランス語版がある.

2. 入手からインストール・起動

各サイト^[6,9,10]より主要 OS 対応のバイナリとソースコードが公開されており, 自分の環境にあったものをダウンロードして, インストールすることができる. ソースコードからコンパイルしてインストールする場合は, 数値計算に必要なライブラリを入れておく必要がある. 随時バグ修正等の更新が行われているので, 各自, Web サイトをチェックしていただきたい. Linux, Mac の場合, コマンドラインから起動できる. 一方, Windows の場合, 起動用ショートカットアイコンをクリックすることにより起動する.

3. 対話モードによる利用

MATLAB と同様に対話的にコマンドを入力し, 随時, 結果が表示される電卓的な使い方もできる. Octave と FreeMat はプロンプトが違っただけ同じように扱える. 以下に Octave での対話的な使用例を示す.

```
> (123+345) * (7-6) * 8^2
ans = 29952
> sin(0.25*pi) * log(1+i)
ans = 0.24506 + 0.55536i
> exp(1)
ans = 2.7183
> sin(0.25*pi) + cos(0.25*pi)
ans = 1.4142
> quit
```

Scilab でも同様にプロンプトに続けて操作できる.

```
-->(123+345) * (7-6) * 8^2
ans =
29952.
-->sin(0.25*pi) * log(1+i)
ans =
0.2450645 + 0.5553604i
-->exp(1)
ans =
2.7182818
-->sin(0.25*pi) + cos(0.25*pi)
ans =
1.4142136
-->quit
```

quit または **exit** は終了するコマンドである.

4. 機能概説

MATLAB を利用した経験のある読者の方は不要な説明と感じるでしょうが, ここでは主な機能を簡単に述べる. なお, より詳細な機能については参考文献^[3,4,5]と各マニュアルが参照になる.

4.1 ベクトル, 行列演算

MATLAB と同じくこれらのソフトウェアの特徴は, ベクトルや行列を簡単に取り扱えることである. ベクトルと行列は, 次のように四角括弧 [] で, 囲み, 行方向 (横方向) は空白またはカンマで区切り, 列方向 (縦方向) はセミコロン ; で区切る.

```
> A=[1 2;3 4]
```

```
A =
```

```
1 2
```

```
3 4
```

```
> a=[5;6]
```

```
a =
```

```
5
```

```
6
```

ベクトルと行列の演算はスカラーの場合と同様に行うことができる.

```
> A*a
```

```
ans =
```

```
17
```

```
39
```

行末に;をつけると値の表示はなくなる.

```
> B=A*a;
```

単に変数名を入力すると変数の値が表示される.

```
> B
```

```
B =
```

```
17
```

```
39
```

ベクトルと行列の転置は'をつける.

```
> B'
```

```
ans =
```

```
17 39
```

行列要素を取り出す場合は、丸かっこ()内に要素を指定する.

```
> A(1,2)
```

```
ans = 2
```

行列要素に直接、値を代入することもできる.
さらに一行または一列の要素を全て取り出したい場合は:で指定する.

```
> A(2,2)=10
```

```
A =
```

```
1 2
```

```
3 10
```

```
> A(1,:)
```

```
ans =
```

```
1 2
```

```
> A(:,1)
```

```
ans =
```

```
1
```

```
3
```

ベクトルや行列のスカラー倍は、気にせずにそのまま記述すれば良いが、行列の要素同士の掛け算、割り算は、次のように記述するため、注意が必要である.

```
> C=2*A
```

```
C =
```

```
2 4
```

```
6 20
```

```
> A.*C
```

```
ans =
```

```
2 8
```

```
18 200
```

```
> A./C
```

```
ans =
```

```
0.50000 0.50000
```

```
0.50000 0.50000
```

良く使う数列ベクトルの生成を以下に示す.

始点:差分:終点 (0から10まで2刻のデータ生成)

a は行ベクトルとして設定される.

```
> a=0:2:10
```

```
a =
```

```
0 2 4 6 8 10
```

差分を省略すると差分は1に設定される.

```
> b=0:10
```

```
b =
```

```
0 1 2 3 4 5 6 7
```

```
8 9 10
```

同じ表現を **linspace(始点, 終点, 個数)** を使うと次のように指定できる.

```
> c=linspace(0,10,6)
```

```
c =
```

```
0 2 4 6 8 10
```

範囲は差分を指定するため最後の要素が終点の値にならないこともあり, **linspace** および **logspace** は、指定した要素数の行ベクトルを生成する. 刻みは範囲と個数から自動的に決まる.

4.2 組み込み数学関数

Octave, FreeMat, Scilab には多くの数値演算用の関数が組み込まれている. ただし、現状では、FreeMat は他に比べると組み込み関数が少ないのが残念である. 以下にいくつか例を見る. 各関数の説明は **help** コマンドで参照できる.

> **help cos** のようにコマンドを入力する.

・三角関数

```
> x=1:4
x =
    1    2    3    4
> cos(x)
ans =
    0.54030 -0.41615 -0.98999 -0.65364
```

・統計関数

ベクトル要素の最小, 最大値, 和, 積は,

min(), max(), sum(), prod(),

平均, 標準偏差, 分散は

mean(), std(), var()で組み込まれている.

0 から 1 までの間の一様乱数は **rand()**, 平均 0, 分散 1 である標準正規分布の乱数は **randn()** の関数が利用できる. 引数を指定すると乱数行列が作成できる.

```
> randn(3,3)
ans =
    0.23756 -1.41678    0.98165
   -0.74434 -1.14407   -0.14617
   -0.16691    0.67919   -0.15782
```

並べ替え操作は **sort()** 関数が組み込まれている.

・行列操作

逆行列は, **inv()** で求めることができ, 行列の固有値, 固有ベクトルは **eig()** で計算できる.

[U, D]=eig(A)

Scilab は **eig()** ではなく **spec()** を指定する.

U は固有ベクトルを列ベクトルとして含む行列,

D は固有値を対角成分にもつ対角行列である.

4.3 グラフィックス機能

多くのグラフ作成の関数が組み込まれている. 一般的な折れ線グラフは **plot** 関数で描画する.

3D のグラフは **surf(), mesh(),** 等高線は **contour()** 等で描画できる. グラフの保存は **print** コマンドで行うことができ, ファイル保存形式は **png, jpg, ps, eps** 等が指定できる.

4.4 プログラミング

対話モードにより, コマンドを直接入力して計算するのは直感的にわかりやすいが, 条件分岐などの複雑な処理を行う場合はプログラムを組んだり, 関数を定義したりすることができる. プログ

ラムはスクリプトと呼ばれ, テキストエディタでプログラムを記述し, テキストファイルで保存する. Octave, FreeMat では **ファイル名.m** という拡張子 **m** の MATLAB 形式の M スクリプト, 一方 Scilab では **ファイル名.sci**(または **sce**) の拡張子 **sci, sce** の独自形式である. 例えば行列とベクトルの積を作成するスクリプトの場合, **script01.m** というファイル名でエディタで記述し, 保存を行う. その後にそのスクリプトを実行する手順となる.

```
A=[3 4 ; 5 6];
```

```
c=[1 ; 2];
```

```
B=A*c
```

実行はコンソール上で拡張子なしのファイル名を

```
> script01
```

入力することによりスクリプトが実行される.

・分岐処理

通常のプログラミング言語と同じように **if** 文を用いて分岐処理を行う.

```
if a<0
    b=-2*a
else
    b=2*a
end
```

・繰り返し

for 文により繰り返し処理を行うことができる. 1 から 10000 までの整数の和を計算する処理は, 次のように記述できる.

```
N=10000; s=0;
for i=1:N
    s=s+i;
end
```

ただし, **for** 文の実行は, この種のソフトでは非常に時間がかかることがどの解説書にも記述されており, なるべく **for** 文を使わない記述方法を心がけることが望ましい. 上の例では **sum()** 関数を使って, 記述すれば非常に高速になる.

```
s=sum(1:10000);
```

4.5 ユーザ定義関数

スクリプトの先頭行で **function** を記述することで関数を定義することができる. また, スクリプトとは別に関数ファイルとして保存することもできる. 書式を以下に示す.

function [出力リスト] = 関数名(入力リスト)
定義文

end [**function**]

以下は、円の半径 r を引数に円周を求める関数について記述である。

```
function l=circle_length(r)
    l=2*pi*r;
end
```

4.6 その他

説明などのコメントには次の記号が使える。

%(FreeMat), %#(Octave), //(Scilab)

次の定数は、Scilab では%をつける必要がある。
虚数 i, j (FreeMat), i, j, I, J (Octave), $\%i$ (Scilab),
円周率 π $\%pi$ (FreeMat, Octave), $\%pi$ (Scilab)

5. バッチモードによる利用

Octave の場合、コンソールから M スクリプトを実行するには、M スクリプトの保存場所のディレクトリに移動し、先に示したようにスクリプトを実行するか、スクリプトを呼び出し実行する。

```
source "/mfile/example01.m" (Linux)
source "D:\Ymfile\Yexample01.m" (win)
```

6. 使用例紹介

6.1 複素数演算: ナイキスト線図

複素数演算の例として、電気化学インピーダンスのナイキスト線図を以下に示す。

・M ファイル(Octave, FreeMat 共通)

```
clear all
%Parameter
Rsol=10;Rct=10;RL=10;L=10^-3;Cdl=10^-6;
w=logspace(0,6,100);
Inv_ZF=(1/Rct)+(RL+(i*L*w)).^(-1);
Z=Rsol+(Inv_ZF+(i*Cdl*w)).^(-1);
Zre=real(Z); Zim=imag(Z);
%Nyquist Plot
plot(Zre,-Zim,'k-o');
grid on
title('Nyquist Plot');
xlabel('ReZ');
ylabel('-ImZ');
```

リスト-1

グリッドやマーカーは同一に指定したにも関わらず、同じようにならず、それぞれ個性が出ている。

・Sci ファイルの場合

```
clear all
// parameter
Rsol=10;Rct=10;RL=10;L=10^(-3);
Cdl=10^(-6);
w=logspace(0,6,100);
Inv_ZF=(1/Rct)+(RL+(%i*L*w)).^(-1);
Z=Rsol+(Inv_ZF+(%i*Cdl*w)).^(-1);
Zre=real(Z); Zim=imag(Z);
//Nyquist Plot
plot(Zre,-Zim,'k-o');
xgrid();
title('Nyquist Plot');
xlabel('ReZ');
ylabel('-ImZ');
```

リスト-2

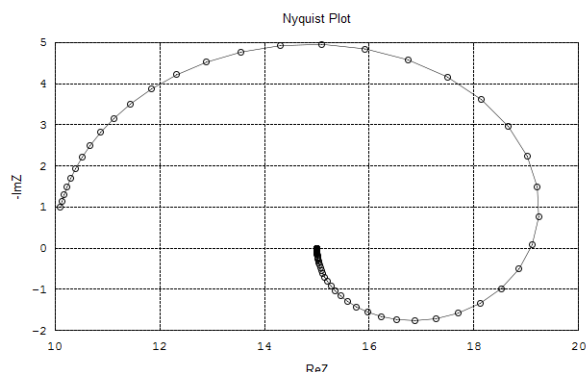


図 1.1 ナイキスト線図(Octave)

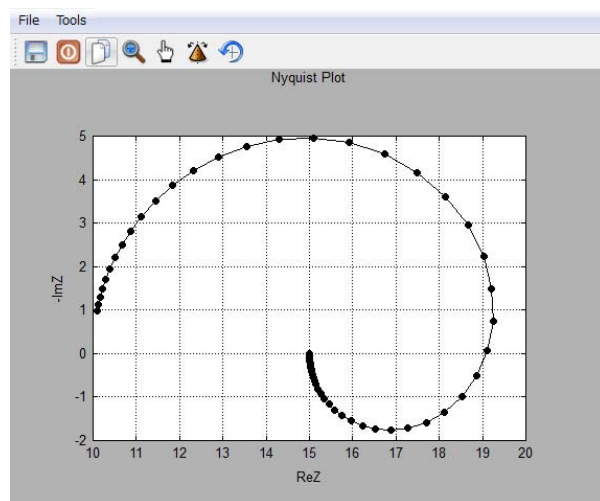


図 1.2 ナイキスト線図(FreeMat)

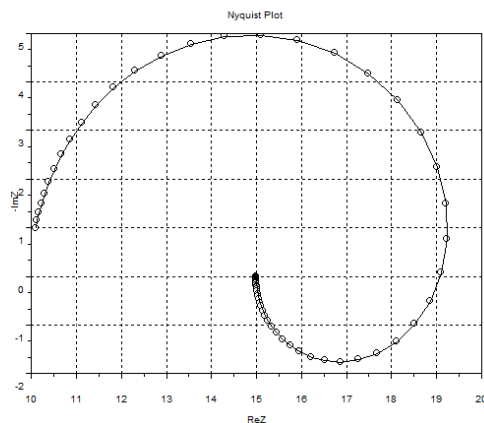


図 1.3 ナイキスト線図(Scilab)

8.2 行列演算：1次元結晶エネルギーバンド

単位格子に、同種2原子を含み、基底原子軌道が1つの場合の1次元結晶のバンド構造は固有値問題を解くことで得られる^[13]。この場合は解析的に解くことが可能だが、ここでは数値解で求めて、結果をグラフ表示する例を紹介する。

M ファイル(Octave,FreeMat 共通)

```
clear all
%parameter
E0=0;t1=2;t2=1;N=31;kmin=-1;kmax=1;
for kN=1:N
    k=kmin+(kmax-kmin)*(kN-1)/(N-1);
    H=[E0 t1+t2*exp(-i*pi*k);
        t1+t2*exp(i*pi*k) E0];
    [V,D]=eig(H);
    a=sum(D);
    E(kN,:)=sort(a);
    X(kN)=kmin+2*(kN-1)/(N-1);
end
hold on
h=plot(X,E,'r*');
axis([-1 1 -3 3])
title('Energy band of 1-D dimerized crystal')
xlabel(' k(pi/a) ')
ylabel(' Energy (eV) ')
grid on
hold off
```

リスト-3

Scilab には MATLAB の M ファイル形式を Sci ファイルに変換するツールとして、m2sci と

いうコマンドが用意されている。これを用いて、スクリプトの変換をしたのが、リスト-4である。グラフの出力結果を図 2.1～図 2.3 に示す。これも先の例と同じく細部は異なっている。

```
clear
//parameter
E0=0;t1=2;t2=1;N=31;kmin=-1;kmax=1;
for kN = 1:N
    k = kmin+(kmax-kmin)*(kN-1)/(N-1);
    H = [E0,t1+t2*exp(-(i*pi)*k);
        t1+t2*exp((i*pi)*k),E0];
    [V,D] = mtlb_eig(H);
    a = sum(D,"m");
    E(kN,1:length(gsort(a,"g","i"))) =
    gsort(a,"g","i");
    X(1,kN) = -1+(2*(kN-1))/(N-1);
end;
set(gca(),"auto_clear","off")
plot(X,E,"r*");
h = gce();
set(gca(),"data_bounds",matrix([-1,1,-3,3],2,-1))
title("Energy band of 1-D dimerized crystal")
xlabel(" k(pi/a) ")
ylabel(" Energy (eV) ")
set(gca(),"grid",[1,1])
set(gca(),"auto_clear","on")
```

リスト-4

9. 雑感

私は MATLAB とのコマンド互換性という点で、Octave,FreeMat が良いと感じる。しかし、グラフを簡単に描画する MATLAB 組み込み関数 ezplot, ezplot3 の未実装や高品質のグラフィックスの細部は、ほぼ互換性が無く残念に思う。

Octave は、2系から3系の変更に伴い、グラフファイル出力は MATLAB と同じ print 文を使うことができるようになったことが大きい。FreeMat は、現時点で日本語での情報はあまり多くない状況である。さらにグラフィックスでは MATLAB 標準の組み込み関数の少なさが目立つ。今後、利用例なども含めて発展途上であるが、並

列処理の実装など新しい取組みも検討されているので、今後の動向が気になるところである。

一方、**MATLAB** とのコマンド互換性が無い **Scilab** にはグラフ機能の充実性やここでは紹介できなかったが、**Scicos** の利用により **MATLAB** の **Simulink** 的な使い方もできるので、他の2つとは異なり、これは利用価値があると思う。

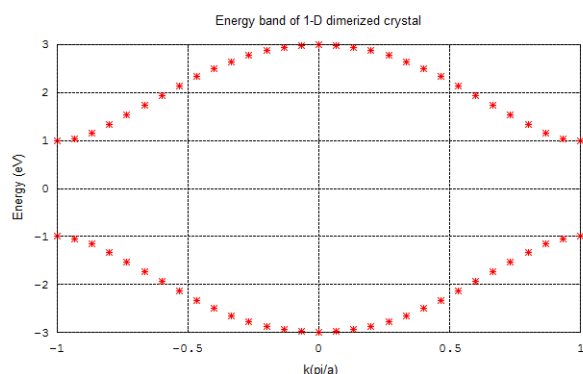


図 2.1 1次元結晶のバンド図 (Octave)

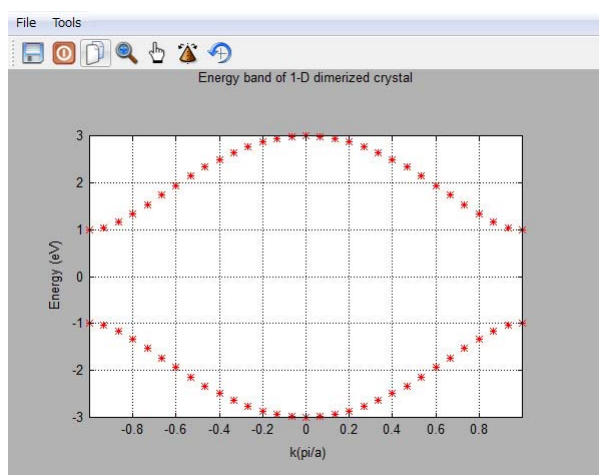


図 2.2 1次元結晶のバンド図 (FreeMat)

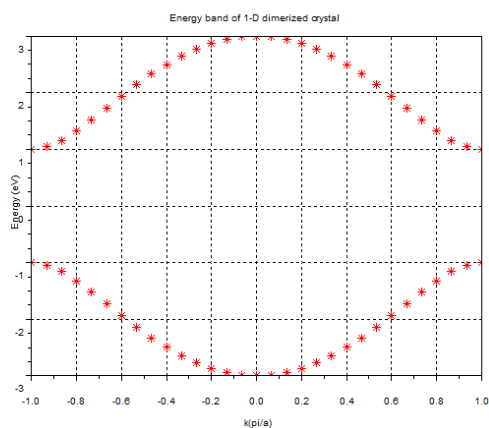


図 2.3 1次元結晶のバンド図 (Scilab)

日本語対応という点では、現在はどれも未対応である。しかし、オープンソースプロジェクトであるので、関心のある方はプロジェクトに参加して日本語版への貢献をされてもよいのではないのでしょうか。今回紹介したフリーソフトは、商用ソフトと違って多少バグがあろうが、うまく使いこなすことがユーザに求められていると私は感じる。

10. おわりに

誌面の都合により、詳しい解説ができなかったが、付属の英文マニュアルは、どれも詳しく書かれているので、そちらに譲る。さらに **Web** での解説記事や書籍の情報も豊富なので是非、参考文献をご覧くださいとよい。

経験に優るものはないので、興味を持たれた方は、実際に **Web** から入手してインストールを行い、まずは自分で操作してみることをお勧めする。

参考文献

- [1] http://ja.wikipedia.org/wiki/GNU_Octave
- [2] <http://ja.wikipedia.org/wiki/Scilab>
- [3] 上坂吉則:「**MATLAB+Scilab** プログラミング事典」ソフトバンククリエイティブ
- [4] 櫻井鉄也:「**MATLAB/Scilab** で理解する数値計算」東京大学出版会
- [5] 赤間世紀:「**Octave** 教科書」工学社
- [6] GNU Octave
<http://www.gnu.org/software/octave/>
- [7] GNU Octave 日本語版マニュアル Ver.2.1.73
<http://www.obihiro.ac.jp/~suzukim/masuda/octave/html/>
- [8] 数値演算言語 Octave
<http://adlib.rsch.tuis.ac.jp/~akira/unix/octave/index-j.html>
- [9] FreeMat Ver.3.6
<http://freemat.sourceforge.net/>
- [10] Scilab ホームページ
<http://www.scilab.org/>
- [11] Scilab 入門
<http://www.ecl.sys.hiroshima-u.ac.jp/scilab/introscilab/introscilab.html>
- [12] Scilab つかいませんか
<http://user.numazu-ct.ac.jp/~mochizuki-k/scilab/>
- [13] S.Datta「Quantum Transport Atom to Transistor」CAMBRIDGE UNIVERSITY PRESS