

科学技術計算用シミュレーション・ソフトウェアをマスターしよう

第1章

フィルタ設計のための Scilab入門

まず、本書で利用する“^{サイラブ}Scilab”というフリーの科学技術計算用シミュレーション・ソフトウェアの活用法について、実行例を示しながら解説する。

とくに、フィルタを解析し、入出力信号波形をグラフ表示するための基本コマンドを紹介して、読者のみなさんが独力で有用な信号処理プログラムを作成できるように、Scilabプログラミングの基礎を固めます。

1.1 Scilab とは

本書では、フィルタ設計理論の習得をめざして、“Scilab”を活用することを前提にしています。Scilab は、INRIA (フランス国立情報学自動制御研究所) が開発・提供する大変便利なシミュレーション・ソフトウェアです。グラフ表示機能も多彩で、しかもプログラム作成もとても簡単であるという特徴があります。なお、Scilab 全般の情報や使用法に関する詳しいマニュアルなどは、INRIA の Web サイト (<http://www.scilab.org/fr/>) を参照してください(図 1.1)。

Scilab を利用すれば、パソコン画面上に信号波形や周波数特性などをパッと表示して、実際に“やってみる、体験する、目で確かめてみる”ことができます(図 1.2)。つまり、ビジュアルに信号処理プロセスを把握することができるように、フィルタを解析、グラフ表示するための有用なコマンド



図 1.1 Scilab の Web サイト (2008 年 11 月現在)

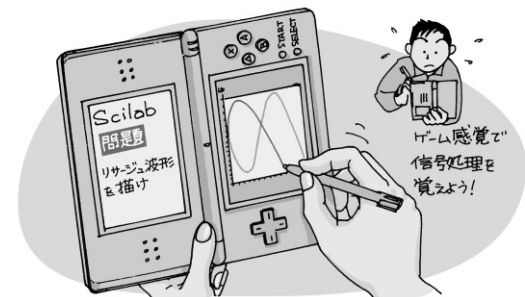


図 1.2 Scilab は信号処理用ゲーム・ソフトのようなもの

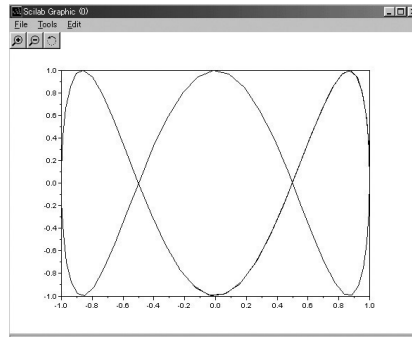


図 1.3 リサージュ曲線のグラフ表示

(命令)が多数用意されているのです。

たとえば,

```
--> x = 0:0.1:2*pi;
--> plot(cos(x), sin(3*x));
```

と入力するだけで、図 1.3 に示すリサージュ曲線を描くことができるのです。あとで実際に体験してもらえれば、「なんとすばらしいソフトなんだ」とひとしきり感心されるでしょう。

なお、Scilab のインストールについては、付録 (pp.203-207) を参考にしてください。これ以降は、みなさんが Scilab を使用できることを前提に解説を進めるとしましょう。

1.2 Scilab の基本コマンドによる処理

まずは、Scilab を利用するために、基本コマンドの処理内容や利用法について説明します。

1.2.1 入力, 出力表示

たとえば、変数に数値 123.4 を代入するときの表記は、`a=123.4` のようにします。なお、キーボードから入力した文字や記号などは `このアミかけ` でマークします。

実行例 1.1

```
--> a = 123.4
```

改行キー `Enter` を押すと、代入の結果が表示され、

実行例 1.2

```
--> a = 123.4 Enter
a =
123.4
```

となります。また、 1.234×10^{-5} のように表現される数値の入力は、指数部を `e` で表し、

実行例 1.3

```
--> a = 1.234e-5
a =
    0.0000123
```

と入力します。このとき、4が表示されませんが、format 命令を用いて表示形式を変更できます。format 命令の一つ目の引き数が 'v' か 'e' によって指数部を付けるかどうかを指定し、二つ目の引き数で出力するときの表示文字数の上限を指定します。文字数には、符号(±)、小数点(.)、指数部記号(D)を含めてカウントします(図 1.4)。なお、デフォルト(既定値)は、format('v',10)であり、表示文字数の上限を超えた部分は四捨五入された値として出力されます。

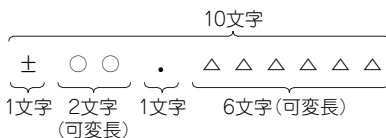
実行例 1.4

```
--> format('v',5)
--> a = 1.234e-5
a =
    0.00
--> format('v',12)
--> a = 1.234e-5
a =
    0.00001234
--> format('e',12)
--> a = 1.234e-5
a =
    1.23400D-05
--> format('v',10) ; // デフォルトに戻す
```

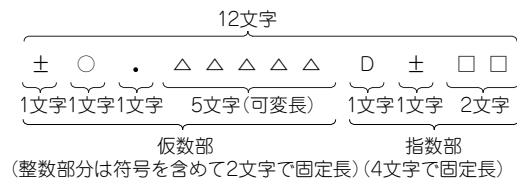
Scilab では、2 個の連続するスラッシュ記号(//)から後ろに記述した文字列はコメントと見なされるので、そこに書いた内容は実行結果に影響しません。ただ、日本語文字などの2バイト・コードは文字化けするので、英数字と記号の入力が基本になります。

実行例 1.5

```
--> a = 9 // real number
a =
    9.
```



(a) format('v',10)の場合



(b) format('e',12)の場合

図 1.4 format 命令の各パラメータの意味

対話的に実行しているときに処理結果が次々と画面表示されるのは便利なのですが、プログラムとして実行したり、大量のデータを取り扱うときは表示画面がスクロールして見にくくなります。このようなときには、行末にセミコロン(;)を付けると、処理結果は表示されず、処理命令のみが表示されます。

Scilab では、変数名の大文字と小文字を区別するので、次のように変数名として a と A を混在させることも可能です。

実行例 1.6

```
--> format('v',10)
--> a = 1.2;
--> A = 8.9;
--> a + A
ans =
    10.1
--> b = a - A
b =
   -7.7
--> disp(a * A)
10.68
--> disp(a * A);
10.68
```

変数 b に値や計算結果を代入したときは、結果の出力は `b =` のように変数名が表示されますが、代入していないときには `ans =` と表示されます。

また、結果だけを表示させたいときは、`disp` 命令を用います。ただし、`disp` 命令を用いたときは、行末にセミコロンを付けても処理結果はかならず表示されます。

さらに、1行に複数の命令をまとめて書くときには、カンマ(,)かセミコロンで区切ります。このとき、カンマのときは結果が出力されますが、セミコロンのときは出力されません。

コラム 1

入力時に知っておくと便利な機能

もしも入力ミスをしたときは、最初にカーソル移動キーの `↑` あるいは `↓` を押すことにより、以前に入力した文字列を再表示できます。次に、`→` あるいは `←` を押して、変更箇所へ移動して修正します。このような方法で、すべての文字

列を再入力することなく、以前に入力した文字列を再利用して、ミスしたところだけを効率よく修正することができます。また、`Home` キーを押すと、入力行の先頭にカーソルを移動できます。