JAPLA 研究会資料　2008/3/22

Ｊの数学関数グラフィックス(nplot)－その１

西川　利男

１．Ｊのいろいろなグラフィックス

Ｊは計算処理だけにとどまらず、グラフィックスにおいても極めて強力である。
ところで、一口にグラフィックスといっても

- 数学の計算結果を表示するグラフ
- 経済、統計などの折線グラフ、ヒストグラム、円グラフなど
- アートとしての画像表示

などいろいろある。そして、それぞれ内容はかなり異なっている。
ここでは、数学の関数表示を目的としたグラフィックスをとりあげる。

Ｊには、簡便だが非常に多機能なグラフィックス・ツールとしてplotルーチンがあるが、以下のような理由で筆者にとっては使い勝手がよくない。

- 計算値に無限大や複素数が現れたときは、当然そのときはエラーになる。
- x，ｙの座標値を「かってに、自動的に」正規化しグラフ図示するので真の値の大きさが比較できなくなってしまう。
- 複数のグラフの重ね描きができない。

先月、筆者の個人的な必要性から、レムニスケート曲線を表示するグラフィックスを報告した[1]。これをもとに、もっと汎用的にいろいろな数学関数に対しても、手軽に使えるグラフィックス・ツールを、nplotとして構築した。

２．数学関数グラフィックスとは

数学関数グラフィックスに望まれる仕様とは、次のようになると思われる。

① 関数の計算値＝グラフの大きさを保持して表示する。
② 座標軸目盛と目盛数値の表示
③ 無限大や複素数の値の適切な扱い＝その点を飛ばして表示しない。
④ 複数のグラフの重ね描き＝無理関数の±の値のグラフにも対応する。
⑤ グラフの色表示もほしい。とくに重ね描きの場合に必要である。
⑥ グラフの拡大、縮小（ズーミング）
⑦ グラフの移動

今回はこれらの項目のうち、実現できたところだけを報告する。 ユーザインターフェース(使い勝手)をも含めて、次回にわたっても引き続き作成、改良をつづけたい。

文献

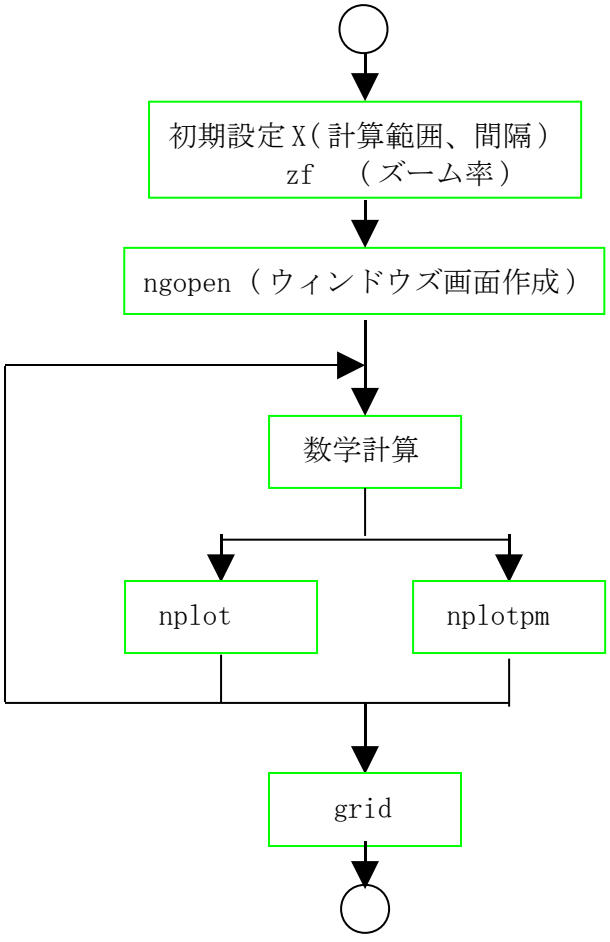
[1] 西川利男「Ｊグラフィックスによる一般のレムニスケート図形」
JAPLA 研究会資料 2008/2/23

３．プログラムの構成と主なポイント

全体のプログラム構成は次のようになっている。

フォーム作成	ngopen ……	gopen(isigraph にある)を一部修正したもの
グラフプロット	nplot ……	グラフを描く
	nplotpm……	無理関数などの±のグラフを描く
座標目盛	grid ……	座標軸、目盛、目盛文字を描く
補助関数	draw ……	X,Y の配列の値をつなぐ
	im2z ……	虚数値の値を排除する(0 にする)
	val2pixel …	計算値からピクセル値に変換する
	wrttxt ……	目盛文字をまとめて書く
	zoom ……	グラフの拡大、縮小をおこなう
グローバル値	range ……	X の範囲(デフォルト_10, 10)
	interval …	X の間隔(デフォルト 200)
	COLOR ……	重ね合わせグラフの色(赤、緑、青、茶)
サンプル関数	多項式, sin,..など	
	直角双曲線…	無限大の不連続値の例
	平方根……	虚数値と±の計算値の例
	楕円……	色を替えた重ね合わせグラフの例
	レムニスケート…色を替えた重ね合わせグラフの例	
実行の書式	nplot X,.f X … X と f X(関数値)とのタテ連結の配列をデータ	

また、プログラムの流れは次のようになる。



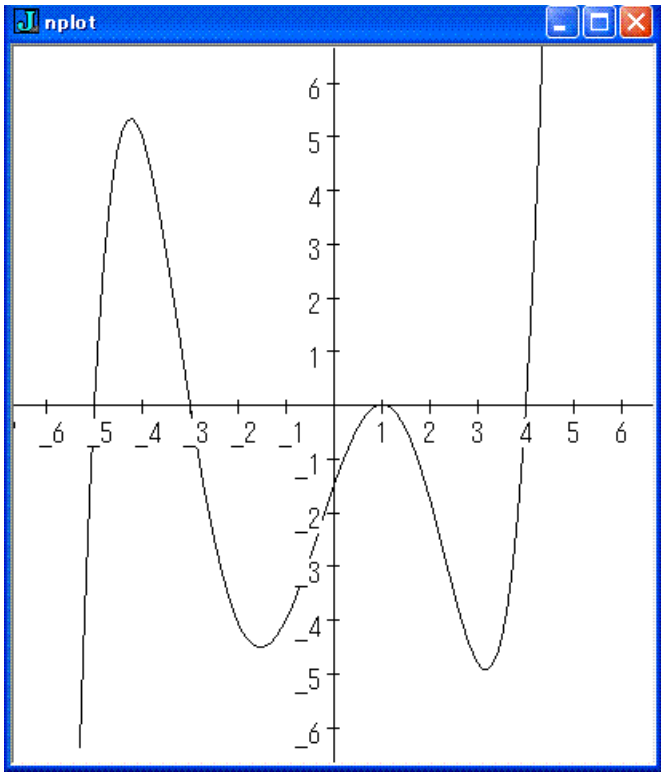
なお、個々のプログラムの処理の詳細はプログラム・リストを参照されたい。

４．プログラム nplot の実行例

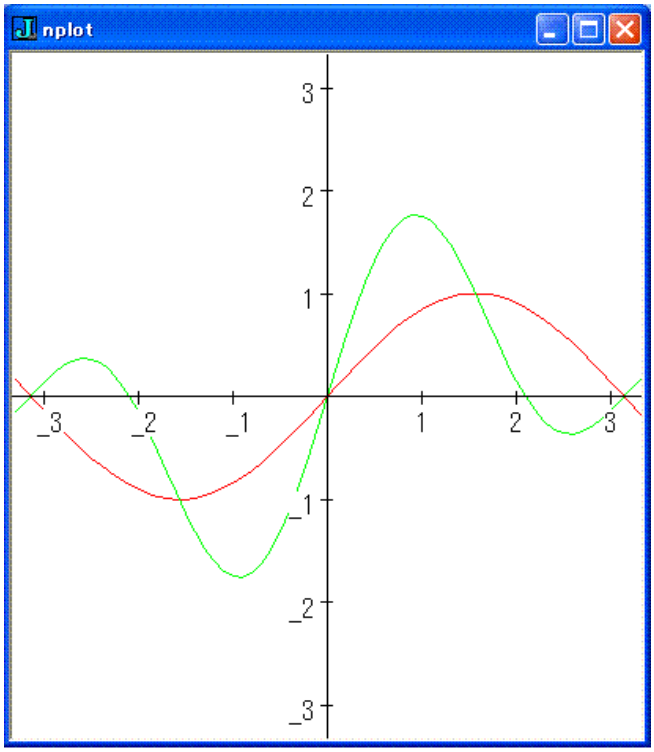
まずは、以下の書からつぎの高次多項式の関数を nplot により表示してみよう。

VNR Concise Encyclopedia of Mathematics, p.125, Fig.5.2-20

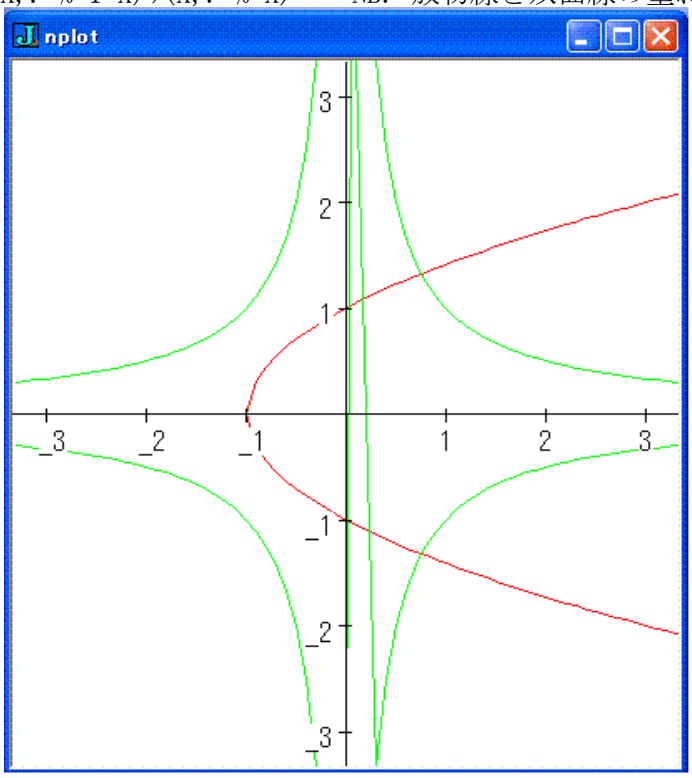
```
nplot X,.(0.025*X^5)+(0.05*X^4)+(_0.6*X^3)+(_0.55*X^2)+(2.575*X)+(_1.5)
```



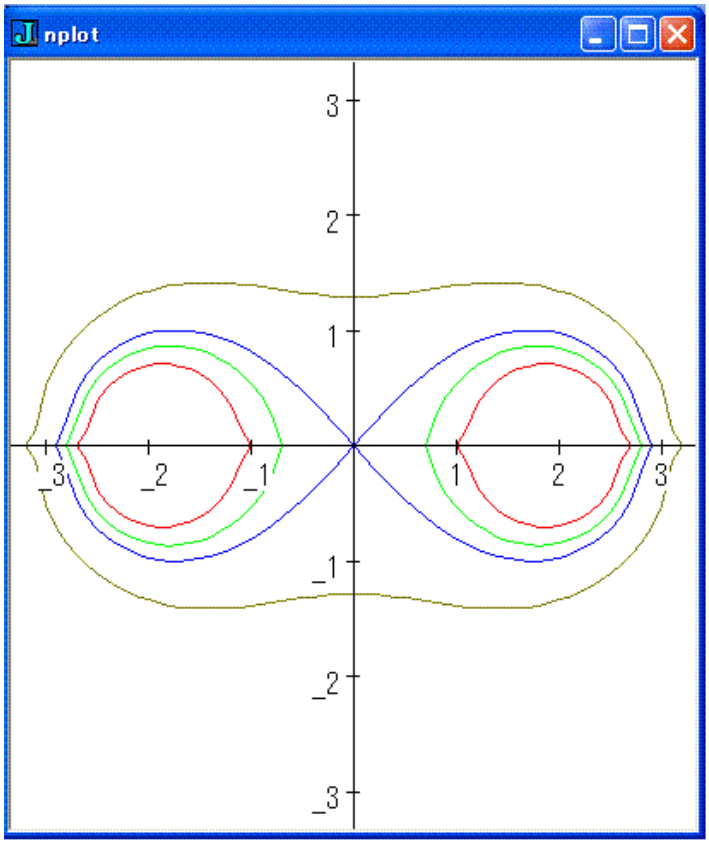

```
COLOR nplot(X,.(sin X));(X,.(sin X)+(sin 2*X)) NB. サイン曲線の重ね合わせ
```



COLOR nplotpm (X,. %:1+X);(X,. % X)　　NB. 放物線と双曲線の重ね合わせ



COLOR nplotpm (X,.YLEM8);(X,.YLEM12);(X,.YLEM16);(X,.YLEM32)
NB. いろいろなレムニスケートの重ね合わせ



プログラムリスト

```
NB. nplot (Nishikawa's plot tool) by T.Nishikawa 2008/2/22, 2/29
NB.   multi graphs and coloring 2008/3/1
NB.   npotpm = plus-minus plotting  2008/3/3
NB.   zooming graph and scale       2008/3/5
NB. plot math function values, enable scaling and multi graphs draw

NB. Usage:
================================================================================
NB. Set Scanning Range & Interval on X:
NB.              X =: steps _5 5 100
NB. Zooming Factor:
NB.              zf =: 150
NB. Exaple:      nplot
X,.(0.025*X^5)+(0.05*X^4)+(_0.6*X^3)+(_0.55*X^2)+(2.575*X)+(_1.5)
NB. y=x^3-x+1:   nplot X,. (X^3) + (-X) + (1)
NB.              (255 0 0) nplot X, (X^3) + (-X) + (1)
NB. Sin_Curve:   nplot X,.sin X
NB. Rectangular Hyperbolic:   nplot X,. 1%X
NB. y=sqrt(x+2): nplot X,. %: X + 2
NB.              YSQ =: %: X + 2
NB.              nplot (X,.YSQ);(X,.-YSQ)
NB.  plus-minus plot: nplotpm X,.YSQ
NB.  red_line plot:   (255 0 0)  nplotpm X,.YSQ
NB. ellipse:     nplot (X,.YELLIP4);(X,.-YELLIP4)
NB.              COLOR nplotpm (X,.YELLIP3);(X,.YELLIP4);(X,.YELLIP5)
NB. lemniscate:  nplot (X,.YLEM16);(X,.-YLEM16)
NB.              nplotpm (X,.YLEM8);(X,.YLEM16);(X,.YLEM32)
NB.              COLOR nplotpm (X,.YLEM8);(X,.YLEM12);(X,.YLEM16);(X,.YLEM32)
NB.
================================================================================

NB. set range and interval
require 'numeric trig'
range =: _10 10
interval =: 200
X =: steps range, interval      NB. _4 to 4 divided in 200 intervals

NB. set zoom factor(zf)
zf =: 150                       NB. defaul: zf = 150

NB. change zoom factor
NB. e.g. zoom 75(zoom_out), zoom 200(zoom_in)
zoom =: 3 : 0
zf =: y.
)
```

```
NB. adjust math values to pixels
val2pixel =: 3 : 0
500 + zf * y.
)

require 'gl2'

NB. not require 'isigraph'
NB. ngopen: renamed from gopen of isimain.ijs
NB.   open graphics window
NB. y. = controlname;parentname
NB.    if either empty, default to 'isigraph'
NB. e.g   ngopen ''
NB.       ngopen '';'J Graphics'
ngopen=: 3 : 0
y=: 2{.}.0;y.
'c n'=: (<'isigraph') ((# i.@#)y=<'') }y
if. (<c) e. <;._2 wd 'qp;' do.
  wd 'psel ',c,';ptop'
  glclear''
else.
  wd 'pc ',c,' closeok;pn *',n
  wd 'xywh 0 0 200 200;cc g0 isigraph rightmove bottommove'
  wd 'pas 0 0;pcenter;pmove 150 30 -1 -1'
  wd 'ptop;pshow;'
end.
)



NB.  im2z 2j1 2 _3j5 3 4j_5 => 0 2 0 3 0
NB. imaginary to zero
im2z =: 3 : 0"0
imf =. 0 ~:  {:"(1) +.  y.
imx =. imf # i.# y.
0 imx } y.
)

NB. Examples
================================================================
NB. Square Root Function Curve
YSQ =: %: X + 2
NB. nplot X,.YSQ
NB. nplot (X,. YSQ);(X,. -YSQ)

NB. Lemniscate Curve
NB. Lemniscate Solution
```

```
lemni =: 3 : 0"0
:
x =. y.
m =. x.
r =. (-x^2) + _4 + %: (16 * x^2) + m
if. r >: 0 do. %: r else. 0 end.
)


YLEM8 =: 8 lemni X
YLEM12 =: 12 lemni X
YLEM16 =: 16 lemni X
YLEM32 =: 32 lemni X
NB. nplot (X,.YLEM16);(X,.-YLEM16)

NB. Ellipse of Focii(_1, 0) and (1, 0)
ellip =: 3 : 0"0
:
x =. y.
m =. x.
r =. (-x^2) + _4 + 2*m
if. r >: 0 do. -: %: r else. 0 end.
)

YELLIP3 =: 3 ellip X
YELLIP4 =: 4 ellip X
YELLIP5 =: 5 ellip X

NB. Color Data
COLOR =: (255 0 0);(0 255 0);(0 0 255);(125 125 0)
BLACK =: (0 0 0);(0 0 0);(0 0 0)

NB. plot
================================================================================
nplot =: 3 : 0   NB. required draw subroutine
if. 2 = $$y. do. 0 0 0 nplot y. else. BLACK nplot y. end.
:
ngopen 'isigraph';'nplot'
if. 2 = $$y.
  do.
    x. draw im2z y.
  else.
    i =. 0
    while. i < #y.
      do.
        yi =. im2z > i{y.
        xi =. > i{x.
```

```
        xi draw yi
        i =. i + 1
    end.
  end.
grid ''
glshow ''
)


NB. plus-minus plot
============================================================
nplotpm =: 3 : 0   NB. required draw subroutine
if. 2 = $$y. do. 0 0 0 nplotpm y. else. BLACK nplotpm y. end.
:
ngopen 'isigraph';'nplot'
if. 2 = $$y.
  do.
   X =. {."1 y.
   Y =. im2z {:"1 y.
   x. draw X,.Y
   x. draw X,.-Y
  else.
   i =. 0
   while. i < #y.
     do.
       yi =. > i{y.
       xi =. > i{x.
       X =. {."(1) yi
       Y =. im2z {:"(1) yi
       xi draw X,.Y
       xi draw X,.-Y
       i =. i + 1
   end.
end.
grid ''
glshow ''
)

draw =: 3 : 0
:
    DD =. val2pixel y.
    DA =. , (*/"(1) 0 <: DD) # DD
    glrgb x.
    glpen 1 0
    gllines  | DA
)
```

```
NB. axis, grid and numbering ================================================
NB.   required wrtxt subroutine
grid =: 3 : 0
glrgb 0 0 0
glpen 1 0
NB. draw axes ---------------------------------------------------------------
gllines 0, 500, 1000, 500 NB. x-axis
gllines 500, 0, 500, 1000 NB. y-axis
NB. draw scale grid ---------------------------------------------------------
ns =: 9
nsp =: >: i. ns
nsm =: - >: i. ns
gllines L:0 <"(1)((val2pixel nsp),.490) ,. ((val2pixel nsp),.510) NB. xp-grid
gllines L:0 <"(1)((val2pixel nsm),.490) ,. ((val2pixel nsm),.510) NB. xm-grid
gllines L:0 <"(1)(490,.(val2pixel nsp)) ,. (510,.(val2pixel nsp)) NB. yp-grid
gllines L:0 <"(1)(490,.(val2pixel nsm)) ,. (510,.(val2pixel nsm)) NB. ym-grid
NB. numbering scale grid --2008/3/5------------------------------------
XPG =: _2<¥"(1) (_10 + (val2pixel nsp),."(0) 490),. nsp   NB. xp-number
XMG =: _2<¥"(1) (_10 + (val2pixel nsm),."(0) 490),. nsm   NB. xm-number
YPG =: _2<¥"(1) (460,. (10 + val2pixel nsp)),. nsp        NB. yp-number
YMG =: _2<¥"(1) (440,. (10 + val2pixel nsm)),. nsm        NB. ym-number
XYG =: XPG, XMG, YPG, YMG
i =. 0
while. i < #XYG
 do. wrtxt i{XYG
   i =. i + 1
 end.
glshow ''
)

wrtxt =: 3 : 0
gltextxy (L:0) 0{"(1) y.
gltext (L:0) ":(L:0) 1{"(1) y.
)
```