JAPLA 研究会資料 2008/4/26

Ｊの数学関数グラフィックス(nplot2)－その２
オブジェクト指向(OOP)による改良版

西川　利男

１．数学関数グラフィックス OOP 改良版 nplot2

Ｊには簡便で強力なグラフィックス・ツール plot があるが、数学関数の表示を目的としてもっと融通性のある多機能なグラフィックスを目指して、nplot を先に発表した[1]。しかし、プログラムの長大、複雑化とともに見通しも悪くなり、それを改善し、かつユーザ・インターフェースをより便利にするため、オブジェクト指向(OOP)手法により、改良版 nplot2 を作った。

仕様としては、前回の版になおいくつかの機能を追加した。

・複数の数学関数のグラフを重ねて表示する。
・複素数根、分数関数など、不連続点が現れるグラフにも対応する。
・表示グラフの拡大、縮小、ズーミングを可能にする。
・マウスクリックで任意の位置にグラフを移動させる。
・典型的な例題については、リストメニューから選択して実行できる。

簡単な数学関数の実行であれば、例えば次のようにキーボードからコマンドとして入力して、そのまま実行できる。

```
nplot X ;sin X
nplot X ;((X^3) + (_2*X^2) + (3*X) - 0.5)
```

しかし、いくつかの関数の組み合わせで表す場合には、Ｊで簡単なプログラムを作った方がよい。

２．プログラムの構成

先に幾何グラフィックスの OOP 化を示したが[2]、そのときと同様プログラムをベースプログラムとクラスプログラムとに分けて分担させた。

・ベースプログラム nplot2.ijs
　・組み合わせ関数の定義の記述
　・対話型選択リストのウィンドウの記述
・クラスプログラム pnplot.ijs
　・数式関数のグラフィック表示処理の本体
　・目盛、座標軸表示処理
　・ズーム機能（メニューバーからポップアップボタンとして）
　・原点の移動（マウスの左ボタン）

プログラムの大要はベースプログラムとクラスプログラムとに分割したことを除けば、前回報告したものとほとんど同じである。

[1] 西川利男「Ｊの数学関数グラフィックス(nplot)－その１」
JAPLA 研究会資料 2008/4/26
[2] 西川利男「Ｊによるディジタル幾何学－その２　オブジェクト指向(OOP)による汎用・多機能化　三角形の外心、内心、シムソン線、オイラー線などへの適用」JAPLA 研究会資料 2007/10/27

３．いくつかのプログラムの実際

プログラム全体はかなり長大なものになっているが、先に述べたようにグラフィックス処理の大部分はクラスプログラムとし、通常ユーザは例えばレムニスケート関数など、組み合わせ数学関数の場合に、ベースプログラムにその定義を書くだけでよいようにした。

ここでは新しく追加したプログラムのいくつかを説明する。

・リストボックスによる関数例の選択プログラム（ベースプログラム）

まず、リストボックスのウィンドウを表示する動詞を以下のように作る。このとき選択するリストデータは別に定義された名詞Examplesを用いる。

```
NB. Popup Select Example using Listbox ======= Refer J Users Manual p.134-5
NB.   imported from j3¥salute_multi.js
listbox =: 3 : 0
wd 'pc listbox;xywh 5 5 60 60;'
wd 'cc lb0 listbox;'
wd 'set lb0 ', ;Examples,&.>LF
wd 'pas 5 5;'
wd 'pcenter;'
wd 'pshow;'
wd 'wait;pclose;'
wd 'q;'
)
```

次の動詞selectは上で定義されたlistboxを呼び出し、選択操作により返されたインデックスの値indxを元に、別に定義された名詞RunCommandの中から文字列として、取り出し、これを実行する。

```
NB. Select Math_plot Example in Listbox
select =: 3 : 0
INF =: listbox ''
indx =: {:>13{INF
wr {:>12{INF
wr runcom =. >(". indx){RunCommand
". runcom
)
```

以下は名詞ExamplesとRunCommandの定義である。

```
Ex0 =: (<'sin_curve'), <'sin_X%X'
Examples =: ;:'hyperbolic square_root ellipse lemniscate'
Examples =: Ex0, Examples

RunCommand =: 'nplot X,.sin X';'nplot X,. YsinX_X';'nplot X,. 1%X';'(255 0 0)
     nplotpm X,.YSQ'
RunCommand =: RunCommand,<'COLOR nplotpm
     (X,.YELLIP3);(X,.YELLIP4);(X,.YELLIP5)'
RunCommand =: RunCommand,<'COLOR nplotpm
     (X,.YLEM8);(X,.YLEM12);(X,.YLEM16);(X,.YLEM32)'
```

・グラフのズームイン・ズームアウトのプログラム（クラスプログラム）

メニューのポップダウンボタン zin, zout を操作して拡大、縮小をおこなうが、処理の実際はズームファクター zf を√２倍したり、√２で割ったりしている。このとき zf は数学値からピクセル値への変換関数 val2pixelx, val2pixely で以下のように倍率係数として用いられているので、これにより拡大、縮小が行われるのである。

```
pnplot_zin_button=: 3 : 0
zf =: zf * %:2
glclear ''
zdisplay ''
)

pnplot_zout_button=: 3 : 0
zf =: zf % %:2
glclear ''
zdisplay ''
)
zf =: 150
origx =: 500              NB. x-origin
val2pixelx =: 3 : 0       NB. adjust math values to pixels
origx + zf * y.
)
origy =: 500              NB. y-origin
val2pixely =: 3 : 0       NB. adjust math values to pixels
origy + zf * y.
)
```

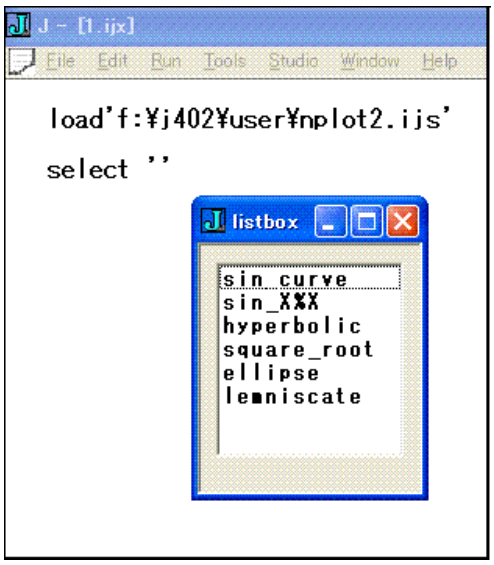
・マウスによるグラフ移動のプログラム（クラスプログラム）

マウス左ボタンアップの操作で、得られたピクセル座標値 xx, yy を原点にセットして、zdisplay によりグラフの再表示を行う。zdisplay ではグラフだけでなく、座標軸、目盛も表示し直す。

```
pnplot_ngraph_mblup=: 3 : 0
d=. ". sysdata
xx=:(0{d) * 1000 % (2{d)
yy=:(1{d) * 1000 % (3{d)
glclear ''
origx =: <. xx
origy =: <. yy
zdisplay ''
NB. gltextalign TA_BOTTOM
NB. gltext ": xx, yy
glshow ''
)
```

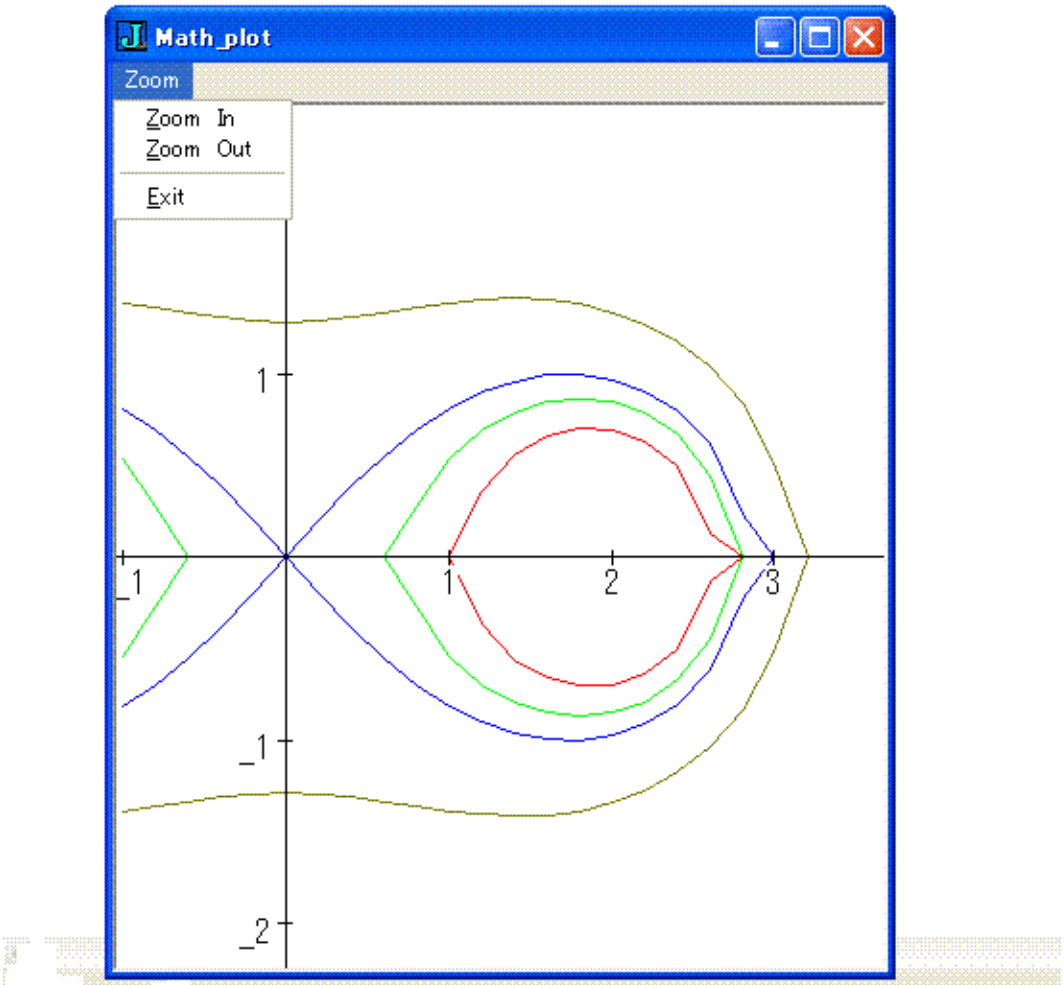
４．実行例

リストボックスからサンプル関数を選んで実行させる。



レムニスケートをクリックして選択すると即実行、ズーム、移動を行った。

```
   select ''
lemniscate
COLOR nplotpm (X,.YLEM8);(X,.YLEM12);(X,.YLEM16);(X,.YLEM32)
```



プログラムリストーベースプログラム

```
NB. nplot2.ijs OOP version
NB.  base program  2008/3/16, 3/17
NB.
NB. Usage: nplot X,. sin X
NB.        nplot (X,. sin X);(X,. (sin X) + (sin 2*X))
NB.        ((255 0 0);(0 0 0)) nplot (X,. sin X);(X,. 2*sin 1-X)
NB.        COLOR nplot (X,. sin X);(X,. 2*sin 1-X)
NB.        nplotpm (X,.YSQ)
NB.        (255 0 0) nplotpm (X,.YSQ)
NB.        COLOR nplotpm (X,.YELLIP3);(X,.YELLIP4);(X,.YELLIP5)
NB.        COLOR nplotpm (X,.YLEM8);(X,.YLEM12);(X,.YLEM16);(X,.YLEM32)
NB. Popup Select Example using Listbox: select ''  2008/3/17

wr =: 1!:2&2

corequire 'user¥Classes¥pnplot.ijs'

X =: steps _10 10 200
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);(0 0 0)


nplot =: 3 : 0
if. 2 = $$ y.
  do. (0 0 0) nplot y.
  else. BLACK nplot y.
end.
:
PARAM =: (<x.), (<y.), (<0)
ins =. PARAM  conew 'pnplot'
empty ''
)


nplotpm =: 3 : 0
if. 2 = $$y.
  do. (0 0 0) nplotpm y.
  else. BLACK nplotpm y.
end.
:
PARAM =: (<x.), (<y.), (<1)
ins =. PARAM  conew 'pnplot'
empty ''
```

```
)
NB. Examples
NB. ========================================================================
NB. (sin X) % X => zoom out for good curve
X =: steps _20 20 200
YsinX_X =: 1 (X i.0) } (sin X) % X
YsinX_X =: 8 * YsinX_X

NB. Square Root
YSQ =: %: X + 2

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. 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. Popup Select Example using Listbox ======= Refer J Users Manual p.134-5
NB.   imported from j3¥salute_multi.js
listbox =: 3 : 0
wd 'pc listbox;xywh 5 5 60 60;'
wd 'cc lb0 listbox;'
wd 'set lb0 ', ;Examples,&.>LF
```

```
wd 'pas 5 5;'
wd 'pcenter;'
wd 'pshow;'
wd 'wait;pclose;'
wd 'q;'
)

NB. Select Math_plot Example in Listbox

select =: 3 : 0
INF =: listbox ''
indx =: {:>13{INF
wr {:>12{INF
wr runcom =. >(". indx){RunCommand
". runcom
)

Ex0 =: (<'sin_curve'), <'sin_X%X'
Examples =: ;:'hyperbolic square_root ellipse lemniscate'
Examples =: Ex0, Examples

RunCommand =: 'nplot X,.sin X';'nplot X,. YsinX_X';'nplot X,. 1%X';'(255 0 0)
      nplotpm X,.YSQ'
RunCommand =: RunCommand,<'COLOR nplotpm
     (X,.YELLIP3);(X,.YELLIP4);(X,.YELLIP5)'
RunCommand =: RunCommand,<'COLOR nplotpm
     (X,.YLEM8);(X,.YLEM12);(X,.YLEM16);(X,.YLEM32)'
```

プログラムリストークラスプログラム

```
NB. nplot class file
coclass 'pnplot'

PNPLOT=: 0 : 0
pc pnplot;pn "Math_plot";
menupop "Zoom";
menu zin "&Zoom" "In" "" "";
menu zout "&Zoom" "Out" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 0 0 200 200;cc ngraph isigraph;
pas 0 0;pcenter;pmove 150 30 -1 -1;
rem form end;
)

wr =: 1!:2&2

require 'numeric trig'

X =: steps _10 10 100

create=: 3 : 0
wd PNPLOT
formhwnd=: wd'qhwndp'
NB. initialize form here
wd 'pshow;'
NB. (255 0 0) nplot X,. sin X
grid ''
x =. >0{y.   NB. y. is left argument of 'conew' in instance
y =. >1{y.
z =. >2{y.
if. z = 0
  do.   x nplot y
  else. x nplotpm y
end.
zf =: 150
)

destroy=: 3 : 0
wd'pclose'
codestroy''
)
```

```
pnplot_cancel=:pnplot_cancel_button=:pnplot_close=:destroy

formselect=: 3 : 'wd''psel '',formhwnd'

pnplot_exit_button=:destroy

pnplot_zin_button=: 3 : 0
zf =: zf * %:2
glclear ''
zdisplay ''
)

pnplot_zout_button=: 3 : 0
zf =: zf % %:2
glclear ''
zdisplay ''
)

zdisplay =: 3 : 0  NB. required display and pmdisplay
if. 0 = $$CODA
 do. (>CODA) display XYDA   NB. for nplot
 else. CODA pmdisplay XYDA        NB. for nplotpm
 end.
)


NB.
================================================================================
zf =: 150

origx =: 500              NB. x-origin
val2pixelx =: 3 : 0       NB. adjust math values to pixels
origx + zf * y.
)

origy =: 500              NB. y-origin
val2pixely =: 3 : 0       NB. adjust math values to pixels
origy + zf * y.
)


NB. axis, grid and numbering  ============================================
NB.   required wrtxt subroutine
NB. origin shift 2008/3/10
grid =: 3 : 0
```

```
glrgb 0 0 0
glpen 1 0
NB. draw axes ------------------------------------------------------------
gllines 0, origy, 1000, origy   NB. x-axis
gllines origx, 0, origx, 1000   NB. y-axis
NB. draw scale grid ------------------------------------------------------
ns =: 9
nsp =: >: i. ns
nsm =: - >: i. ns
oripx =. origy + 10
orimx =. origy - 10
ORIMX =: orimx
oripy =. origx + 10
orimy =. origx - 10
orimya =. origx - 40
orimyb =. origx - 60
gllines L:0 <"(1) ((val2pixelx nsp),.orimx) ,. ((val2pixelx nsp),.oripx)
NB. xp-grid
gllines L:0 <"(1) ((val2pixelx nsm),.orimx) ,. ((val2pixelx nsm),.oripx)
NB. xm-grid
gllines L:0 <"(1) (orimy,.(val2pixely nsp)) ,. (oripy,.(val2pixely nsp))
NB. yp-grid
gllines L:0 <"(1) (orimy,.(val2pixely nsm)) ,. (oripy,.(val2pixely nsm))
NB. ym-grid
NB. numbering scale grid --2008/3/5---------------------------------------
XPG =. _2<¥"(1) ((_10 + (val2pixelx nsp)),."(0) orimx),. nsp
NB. xp-number
XMG =. _2<¥"(1) ((_10 + (val2pixelx nsm)),."(0) orimx),. nsm
NB. xm-number
YPG =. _2<¥"(1) (orimya,. (10 + val2pixely nsp)),. nsp
NB. yp-number
YMG =. _2<¥"(1) (orimyb,. (10 + val2pixely 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.
```

```
)


require 'gl2'

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. 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);(0 0 0)

NB. plot
NB. ========================================================================
nplot =: 3 : 0   NB. required display and draw subroutines
if. 2 = $$y. do. 0 0 0 nplot y. else. BLACK nplot y. end.
:
NB. nplotopen ''
CODA =: < x.
XYDA =: y.
glclear ''
x. display XYDA
)


NB. plus-minus plot
NB. ============================================================
nplotpm =: 3 : 0   NB. required pmdisplay, display and draw subroutines
if. 2 = $$y. do. 0 0 0 nplotpm y. else. BLACK nplotpm y. end.
:
NB. nplotopen ''
CODA =: x.
XYDA =: y.
CODA pmdisplay XYDA
)

pmdisplay =: 3 : 0
:
CODA =: x.
XYDA =: y.
```

```
if. 2 = $$y.
  do.
   X =. {."1 y.
   Y =. {:"1 y.
   CODA display L:0 (X,.Y);(X,.-Y)
  else.
   i =. 0
   while. i < #y.
     do.
       xi =. > i{x.
       yi =. > i{y.
       X =. {."(1) yi
       Y =. {:"(1) yi
       xi display L:0 (X,.Y);(X,.-Y)
       i =. i + 1
   end.
end.
grid ''
glshow ''
)


display =: 3 : 0
:
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 ''
)

draw =: 3 : 0
:
XY =: y.
X =. {."1 XY
```

```
Y =. {:"1 XY
DX =: val2pixelx X
DY =: val2pixely Y
NB.    DD =. val2pixelx y.
DD =. DX,.DY
    DA =. , (*/"(1) 0 <: DD) # DD
    glrgb x.
    glpen 1 0
    gllines  | DA
)

NB. Mouse Operation ================================================
pnplot_ngraph_mbldown=: 3 : 0
d=. ". sysdata
xx=:(0{d) * 1000 % (2{d)
yy=:(1{d) * 1000 % (3{d)
NB. gltextalign TA_BOTTOM
NB. gltext ": xx, yy
)

pnplot_ngraph_mmove=: 3 : 0
d=.". sysdata
if. -.4{d do. return. end.
xx=:(0{d) * 1000 % (2{d)
yy=:(1{d) * 1000 % (3{d)
NB. gltextalign TA_BOTTOM
NB. gltext ": xx, yy
)

pnplot_ngraph_mblup=: 3 : 0
d=. ". sysdata
xx=:(0{d) * 1000 % (2{d)
yy=:(1{d) * 1000 % (3{d)
glclear ''
origx =: <. xx
origy =: <. yy
zdisplay ''
NB. gltextalign TA_BOTTOM
NB. gltext ": xx, yy
glshow ''
)

NB. inform X, Y position 2008/3/15
pnplot_ngraph_mbrdown=: 3 : 0
d=. ". sysdata
```

```
xx=:(0{d) * 1000 % (2{d)
yy=:(1{d) * 1000 % (3{d)
XX =. (xx - origx) % zf
YY =. (yy - origy) % zf
NB. gltextalign TA_BOTTOM
NB. gltext (7.2) ": XX, YY
glshow ''
posit XX, YY
)

NB. popup window for X, Y position values
posit =: 3 : 0
wd 'pc mywin;pn "X,Y";'
wd 'xywh 10 10 60 10;'
wd 'cc e0 edit;'
text =. (6.2) ": y.
wd 'set e0 *', text
wd 'pas 5 5;pcenter;pmove 160 180 -1 -1'
wd 'pshow;wait;'
wd 'pclose;qp;'
)
```