双方向Ｊグラフィックスによるカオス・ダイナミックス

西川　利男

１．はじめに

Ｊ言語はコンパクトなコーディングでありながら、配列処理をはじめとして高度の数値計算を得意としている．同時にその結果を表示するＪグラフィックスはVisual Basic以上に手軽でかつ強力である．

科学のツールとしてのコンピュータは複雑な数値計算を行うことはもちろんだが，その結果を目に見える形で示すという点でグラフィックスの役割は非常に大きい．さらに現在では，単に計算結果をグラフで示すだけにとどまらず，マウスなどの使用により座標の位置，あるいは図形そのものを入力して，それについて処理し結果をグラフ化するという，いうなれば双方向のグラフィックスが重要であり，最近ではこれが手軽に行えるようになった．Ｊはそれを行うのに最も優れたプログラム環境であるといってよい．

ここ，数回にわたりＪグラフィックスのこのような利用として，関数グラフィックス，
フラクタル，天体運動の軌跡などを報告してきた．今回はカオス・ダイナミックスのグラフ解析への利用を紹介する．

２．ロジスティック関数の反復によるダイナミックスとカオス現象

人口増加や食料の問題，あるいはハエなど昆虫の繁殖など生態系システムの数学モデルを示す式として次のロジスティック関数が有名である．

$$f(x) = c\,x(1 - x)$$

あるいはこれは次の漸化式として書いてもよい．

$$x_{n+1} = c\,x_n(1 - x_n)$$

上の式でパラメータ c と初期値 x_0 を与えて反復計算すれば次々と値 x_n が求められる．したがって時間の経過とともにいろいろに変化する時系列量とみることもできる．

また，すぐ後に示すが，このロジスティック関数は，反復計算させると与えたパラメータ c と初期値 x_0 によって最終的に一つの値に収束するときもあるが，パラメータや初期値のちょっとした値のちがいによりとんでもない挙動をしめす関数としても有名である．つまり確定した式でありながら，ごくわずかの初期条件のちがいで予測不可能となる．つきつめれば数値計算の信頼性を脅

かすような懐疑をもたらす重大な問題となる．この現象は一般にはカオス（ Chaos ）と呼ばれている．

Ｊではこれらの実験を簡単に行うことができる．
まず，ロジスティック関数をつぎのようにＪの tacit （暗黙）関数としてを定義しよう．

```
   f =: [ * ] * 1: - ]
```

そして実行はつぎのように行われる．パラメータ $c = 3.2$ ，初期値 $x_0 = 0.5$ として関数ｆを作用させると，計算結果として値 0.8 と求められる．

```
   3.2 f 0.5
0.8
```

得られた結果にｆを繰り返し作用させると次々と値が得られる．いうまでもなくＪでは計算処理は右から行われる．

```
   3.2 f 3.2 f 0.5
0.512
   3.2 f 3.2 f 3.2 f 0.5
0.799539
```

Ｊでは関数の反復実行に便利な　^:(power)　という副詞がありこれを使えばずっと能率的である．さらに各項をまとめて得ることができる．

```
   3.2 f^:(3) 0.5
0.799539
   3.2 f^:(i.3) 0.5
0.5 0.8 0.512
   3.2 f^:(i.10) 0.5
0.5 0.8 0.512 0.799539 0.512884 0.799469 0.513019 0.799458 0.51304 0.799456
```

初期値 x_0 を 0.5 とし，パラメータ c の値を 1.5 ， 3.2 ， 3.5 と変化させたときの最初の10項はつぎのようになる．

```
   1.5 3.2  3.5 f^:(i.10) 0.5
      0.5        0        0
    0.375      0.8    0.875
0.351563    0.512 0.382813
0.341949 0.799539 0.826935
 0.33753 0.512884 0.500898
0.335405 0.799469 0.874997
0.334363 0.513019  0.38282
0.333847 0.799458 0.826941
 0.33359  0.51304  0.500884
0.333461 0.799456 0.874997
```

これから，次ぎの傾向が見てとれるだろう．

c ＝ 1.5 では１つの値 0.333 に収束する．

c ＝ 3.2 では 0.513 と 0.799 の２つの値に収束するらしい．

c ＝ 3.5 では 0.383 ， 0.501 ， 0.827, 0.875と４つの値に収束するらしい．

このように収束値がいくつかに分かれることは分岐（ Bifurcation ）といわれる．

また，いろいろ実験してみればわかるが，初期値 x_0 を別の値にしても収束値におけるこの結果は変わらない．

つぎには c ＝ 4 ときめて，初期値を x_0 ＝ 0.1 ， 0.25， 0.3 ，…といろいろ変えてみる．

```
      4 f^:(i.10) 0.1 0.25 0.3 0.5 0.51 0.749 0.8
          0.1 0.25      0.3 0.5      0.51    0.749      0.8
        0.36 0.75      0.84   1     0.9996 0.751996     0.64
    0.9216 0.75    0.5376   0 0.00159936 0.745992   0.9216
0.289014 0.75  0.994345   0 0.00638721 0.757952 0.289014
0.821939 0.75 0.0224922   0  0.0253856 0.733844 0.821939
0.585421 0.75 0.0879454   0  0.0989649 0.781268 0.585421
0.970813 0.75  0.320844   0   0.356683 0.683553 0.970813
0.113339 0.75  0.871612   0   0.917841 0.865234 0.113339
0.401974 0.75  0.447617   0   0.301635 0.466418 0.401974
0.961563 0.75  0.989024   0   0.842605 0.995489 0.961563
```

このように c ＝ 4 の場合には初期値の取り方によって収束するときもあれば，でたらめな値を振動するときもあり，一概に定められない．とくに 0.5 と 0.51 のようにほんのわずかな初期値の違いがまったくかけ離れた傾向をしめす．

パラメータ c を連続的に変化させたときの収束のようすを示す図は分岐ダイアグラムと呼ばれフラクタルとも関連して興味深い．これについてはあらためて４章に述べる．

なお，この現象は正確には決定論的カオスというべきものである．しかし，一般にはカオス（ Chaos ）＝混沌と解され，自然現象のすべてが混沌としている，などとセンセーショナルにいわれているのはいかがなものであろうか．

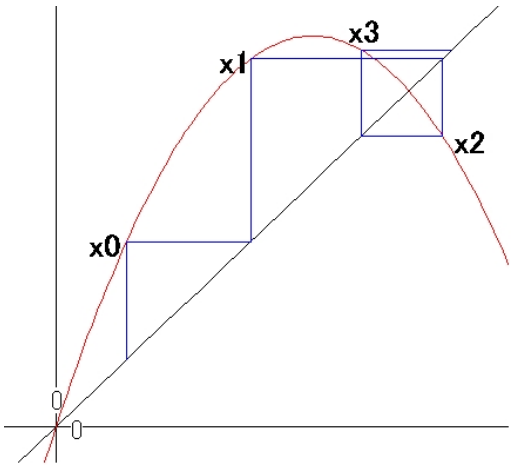
３．双方向Ｊグラフィックスによる反復ロジスティック関数の図による収束の追跡

今回の報告の主な目的は反復ロジスティック関数の収束の問題を例として，数値データではなく座標上の点データとして目に見える形で図式に追跡することにある．

それには図のように関数と４５°傾斜直線との交点を順次結んでできるジグザグ直線を追ってゆく．このようにして収束のようすを目に見える形で示すことができる．

あるロジスティック関数を決めた上で，初期値の位置をマウスにより

任意に指定してそのダイナミックスを追跡していこう.



実行のようすは以下のようになる．

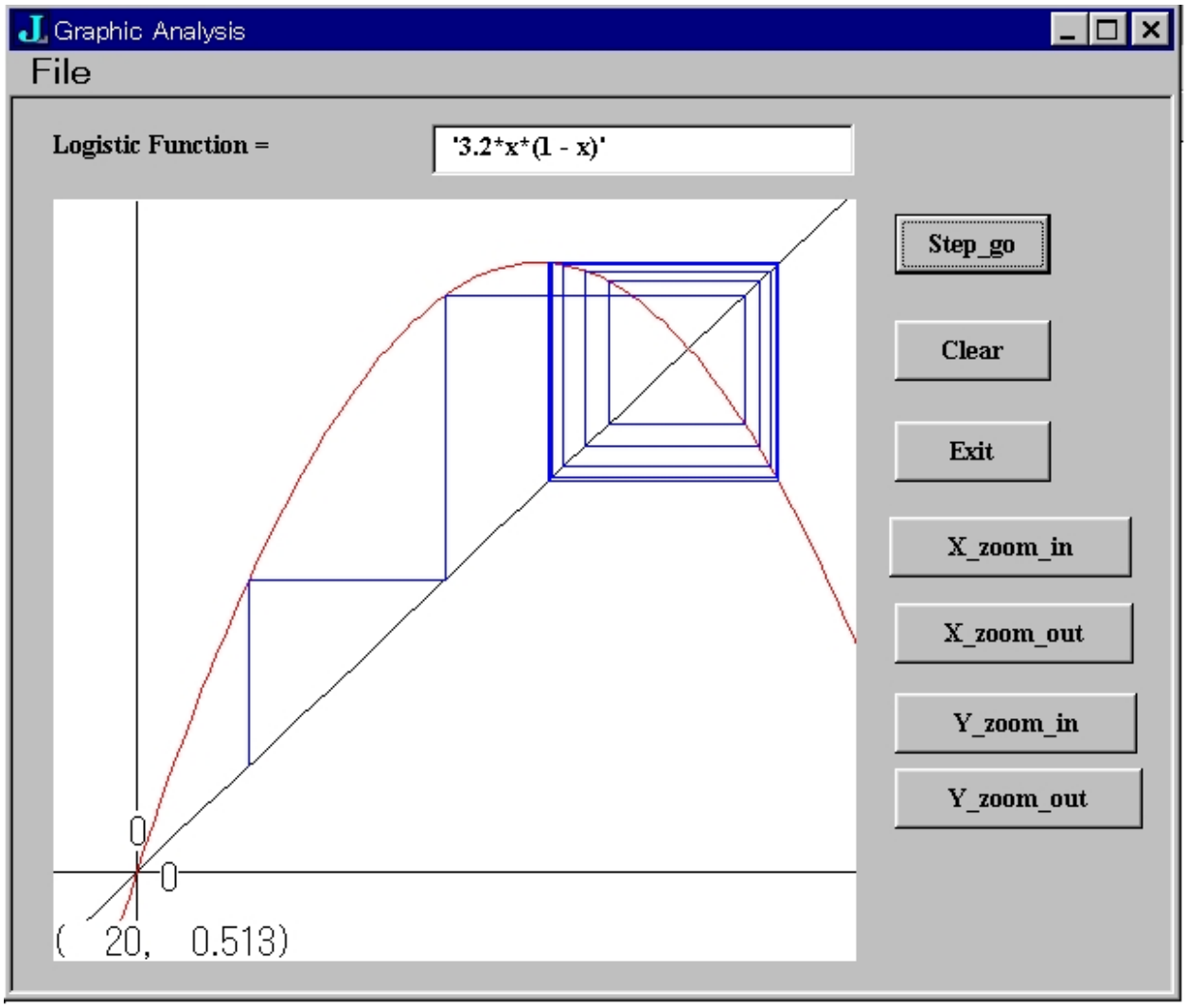
(1) ロジスティック関数を入力する．デフォルトではf(x) = 3.2x(1-x) となっている．

(2) 画面の適当な位置にマウスを置き右ボタンをクリックすると，その位置を座標軸の原点としてグラフが描かれる．座標値は画面の左下に数値で表示される．

(3) マウスの左ボタンで初期値の位置をきめる．

(4) Step_go ボタンをクリックする毎に現在のステップの値にロジスティック関数を作用させ，その値と４５°線との間でジグザグ直線で収束線を描く．

(5) Zoom_In, Zoom_Out ボタンによりＸ軸，Ｙ軸の座標軸を拡大，縮小できるので，ちょうど良い大きさにして観察する．



数値をキーボードから入力し，計算結果を検討する方法でもよいが，マウスから図上で位置として入力し，途中経過が図により示される方法の方がずっと迫力があることはだれでもわかる．筆者はこれを双方向グラフィックスと名づけたが，パソコン利用の方法として教育などにもっと使われてほしいと思う．

４．Ｊの plot パッケージ・グラフィックスによる分岐ダイアグラム

先に述べた分岐ダイアグラムはＪの plot パッケージを用いれば，Ｊのごく簡単なコーディングで実現することができる．詳細は志村正人氏の報告を参照されたい．

ここでは，これを対話的に示してみよう．

まず以下のように３つのパッケージを取り込む．

```
load 'plot trig numeric'
```

Ｘ座標として関数　steps を用いて，例えば，2.5 から　4 以下の 1000 点の値をつくる．

```
X =. steps 2.5 3.999 1000
```

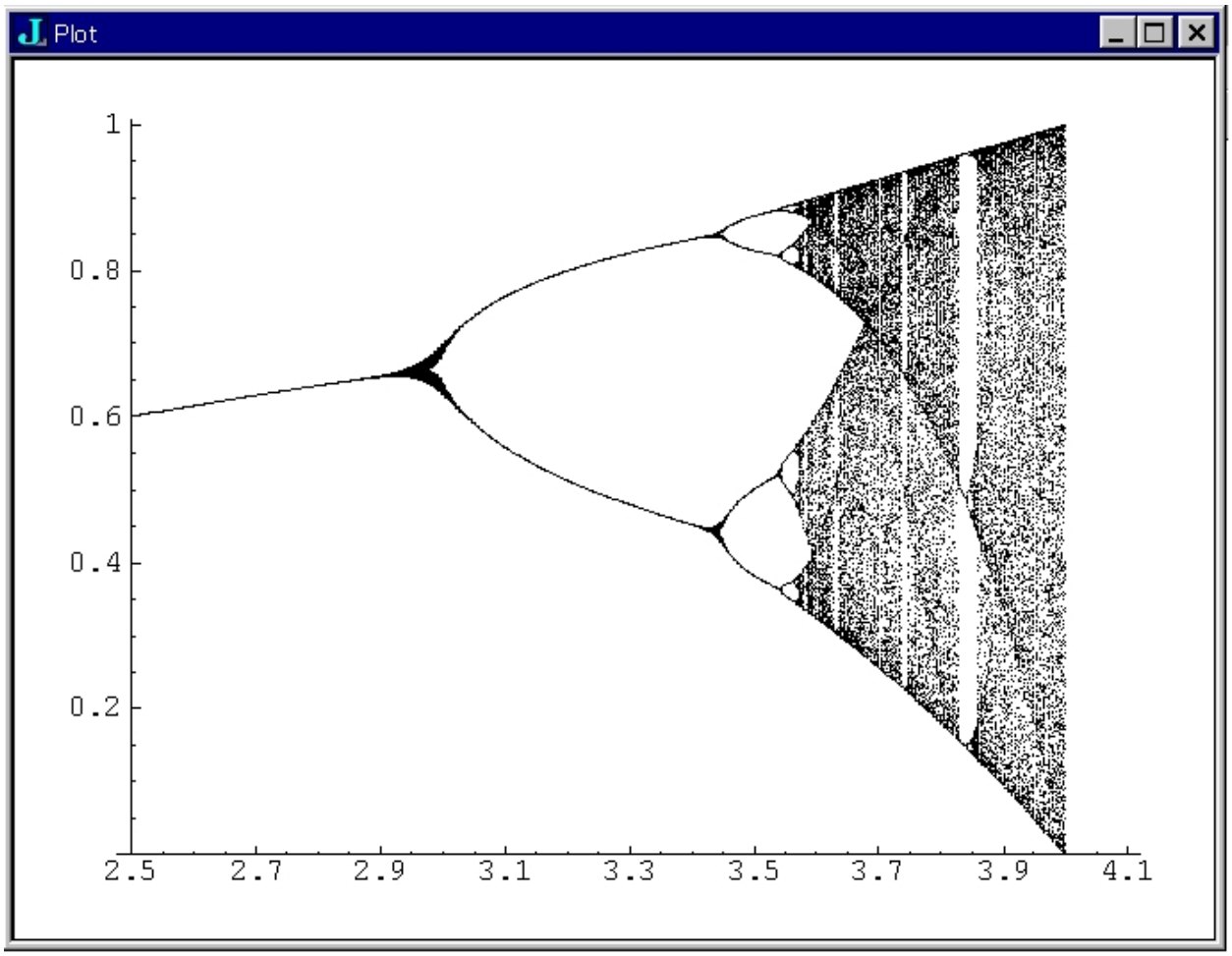
つぎに，パラメータ ***c*** としてこのＸ座標の値を，初期値は　0.5 とし，ここでは40項から 100 の項について収束値を求めて，これをＹ座標の値とする．

```
Y =. X f^:(40+i.100) 0.5
```

線の種類としては　dot を指定してプロットする．

```
'dot' plot X;Y
```

たったこれだけの操作で，つぎのように分岐ダイアグラムがあざやかに得られた．



参考文献

[1] ラウヴェリエール, 西川利男訳「初めてのフラクタル」 p.107-113, 丸善(1996).

[2] R. Devaney,"Chaos, Fractals, and Dynamics", p.26-74, Addison-Wesley(1990).

[3] デイビス, 好田順治訳「美しい数学」 p.376-394, 青土社(1997).

付録　Ｊのプログラム・リスト

```
NB. graph_ana.js
NB. Chaos Dynamics Graphical Analysis
NB.    referred from R. Devaney, "Chaos, Fractals and Dynamics", p.39
NB.    programmed by T. Nishikawa,  2003-7-30, 2003-12-1

require 'gl2'
require 'trig'

GRAPHANA=: 0 : 0
pc graphana closeok;pn "Graphic Analysis";
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 192 23 34 12;cc ok button;cn "Step_go";
xywh 192 64 34 12;cc cancel button;cn "Exit";
xywh 9 20 175 151;cc graf isigraph;
xywh 9 6 127 10;cc label static;cn "Logistic Function = ";
xywh 91 5 93 11;cc func edit ws_border es_autohscroll;
xywh 192 44 34 12;cc clear button;cn "Clear";
xywh 191 83 53 12;cc xz1 button;cn "X_zoom_in";
xywh 192 100 52 12;cc xz2 button;cn "X_zoom_out";
xywh 192 118 53 12;cc yz1 button;cn "Y_zoom_in";
xywh 192 133 54 12;cc yz2 button;cn "Y_zoom_out";
pas 6 6;pcenter;
rem form end;
)

run =: graphana_run
graphana_run=: 3 : 0
wd GRAPHANA
NB. initialize form here
wd 'pshow;'
xa =: 500
ya =: 500
F =: 1     NB. x-axis zoom factor default
```

```
H =: 1     NB. y-axis zoom factor default
 NB. fn =: 3 : '_1 + y.^2'
fn =: 3 : '3.2*y.*(1 - y.)'
 wd 'set func ""3.2*x*(1 - x)""'
'*** Dynamics Graphical Analysis End ***'
 )

 graphana_cancel_button=: 3 : 0
wd 'pclose;'
 )

 pm_i =: (}:@(|.@(-@i.))) , i.    NB. generate plus-minus integers
pm_2i =: (2&*@pm_i) % (<:)
 pm_4i =: (4&*@pm_i) % (<:)
pm_ni =: 3 : 0
 :
(x. * pm_i y.) % (<: y.)
 )



 NB. generate axis graduation data, for X = _4 to 4
S10 =: 10
GR0 =: 2#250 * >: i. S10
 GR0 =: (|.-GR0), GR0
 GR1 =: , >(+:S10)#<20 _20
 GR =: <"1 GR0 ,. GR1
 GRX =: ,"(2) > ((+:S10), 2)$GR
 GRY =: ,"(2) > ((+:S10), 2)$ |.each GR


adj =: + 250&*
 jad =: (%&250) @ (-~)

 NB. graphic disply for function and 45 degree line
NB.    by mouse right button down
graphana_graf_mbrdown=: 3 : 0
d=. ". sysdata
 xa=: (0{d) * 1000 % (2{d)
 ya=: (1{d) * 1000 % (3{d)
```

```
 NB. xa =: 500
 NB. ya =: 500
 glclear ''
glrgb 0 0 0
glpen 1 0
glmove xa,ya
NB. axes and graduations
gllines 0, ya, 1000, ya                              NB. x-axis
 gllines"(1) (xa, ya, xa, ya) +"(1) F * GRX    NB. x-graduations
gllines xa, 0, xa, 1000                       NB. y-axis
 gllines"(1) (xa, ya, xa, ya) +"(1) H * GRY    NB. y-graduations
 NB. Number Scaling
 i =. - S10
while. i < S10 do.
     gltextxy (_10 + xa + 250*F*i), (80 + ya)
     gltext ":  i
      i =. i + 1
     end.
 i =. - S10
while. i < S10 do.
      gltextxy (30 + xa), (20 + ya + 250*H*i)
     gltext ":  i
      i =. i + 1
     end.

NB. 45 degree (y = x) line
X =. 4 pm_ni 100
gllines  , |: (xa adj F * X) ,: (ya adj H * X)
NB. dyanamic function
 Y =. H * fn F %~ X
 glrgb 255 0 0
glpen 1 0
 gllines  , |: (xa adj X) ,: (ya adj Y)
 NB. gltextalign TA_BOTTOM
NB. gltext (":F),', ',(":H)
glshow ''
)

NB. initial X positioning
NB. by mouse left button down
```

```
graphana_graf_mbldown=: 3 : 0
 d=. ". sysdata
 x=. (0{d) * 1000 % (2{d)
 y=. (1{d) * 1000 % (3{d)
 X=: xa jad (0{d) * 1000 % (2{d)
 Y=: ya jad (1{d) * 1000 % (3{d)
 glrgb 0 0 255
 glpen 1 0
 gltextalign TA_BOTTOM
 gltext '(', (5.2": F %~ X),', ', (5.2": H %~ Y), ')      '
 glshow ''
 )

graphana_graf_mmove=: 3 : 0
 d=."". sysdata
 if. -.4{d do. return. end.
 x=.(0{d) * 1000 % (2{d)
 y=.(1{d) * 1000 % (3{d)
 X=: xa jad (0{d) * 1000 % (2{d)
 Y=: ya jad (1{d) * 1000 % (3{d)
 gltextalign TA_BOTTOM
 gltext '(', (5.2": F %~ X),', ', (5.2": H %~  Y), ')      '
 glshow''
 )

 NB. dynamic iterations stepwise
 J =: 1
 graphana_ok_button=: 3 : 0
 x0 =. xa adj X
 y0 =. ya adj H * F %~ X
 x1 =. xa adj X
 y1 =. ya adj H * fn F %~ X
 x2 =. xa adj F * fn F %~ X
 y2 =. ya adj H * fn F %~ X
 X =:  F * fn F %~ X
 XA =. F %~ X
  YA =. fn F %~ X
  J =: J + 1
  glrgb 0 0 255
 glpen 1 0
```

```
gllines x0, y0, x1, y1, x2, y2
gltextalign TA_BOTTOM
gltext '(', (5":J),', ', (7.3":XA), ')                    '
glshow ''
)

NB. enter function
graphana_func_button=: 3 : 0
func =. 'y.' amdstr 'x' of func            NB. convert 'x' to 'y.'
". 'fn =: 3 : ', func
J =: 1
)

graphana_clear_button=: 3 : 0
J =: 1
glclear ''
glrgb 0 0 0
glpen 1 0
glmove xa,ya
gllines 0, ya, 1000, ya                          NB. x-axis
gllines"(1) (xa, ya, xa, ya) +"(1) F * GRX   NB. x-graduations
gllines xa, 0, xa, 1000                            NB. y-axis
gllines"(1) (xa, ya, xa, ya) +"(1) H * GRY   NB. y-graduations
i =. - S10
while. i < S10 do.
     gltextxy (_10 + xa + 250*F*i), (80 + ya)
     gltext ":  i
      i =. i + 1
     end.
i =. - S10
while. i < S10 do.
      gltextxy (30 + xa), (20 + ya + 250*H*i)
     gltext ":  i
      i =. i + 1
     end.
x =. 4 pm_ni 100
gllines  , |: (xa adj F * x) ,: (ya adj H * x)
y =. H * fn F %~ x
glrgb 255 0 0
glpen 1 0
```

```
 gllines  , |: (xa adj x) ,: (ya adj y)
 gltextalign TA_BOTTOM
 gltext (":F),', ',(":H)
glshow ''
 )

graphana_xz1_button=: 3 : 0
 F =: F * 2
 gltextalign TA_BOTTOM
 gltext (":F),', ',(":H), '                '
 )

graphana_xz2_button=: 3 : 0
 F =: F % 2
 gltextalign TA_BOTTOM
 gltext (":F),', ',(":H), '                '
 )

graphana_yz1_button=: 3 : 0
 H =: H * 2
 gltextalign TA_BOTTOM
 gltext (":F),', ',(":H), '                '
 )

graphana_yz2_button=: 3 : 0
 H =: H % 2
 gltextalign TA_BOTTOM
 gltext (":F),', ',(":H), '                '
 )

NB. Function Entry using String Amend from "html.js"
NB.     'y.' amdstr 'x' of '1+x*(1-x)'
 NB. 1+y.*(1-y.)
 str2box=. 3 : 0
 :
 Z1=. x. E. y.
 Z2=. (|. x.) E. (|. y.)
 Z3=. 1, }: |. Z2
Z4=. Z1 + Z3
 Z5=. Z4 <;.1 y.
```

```
AM=: ((<,x.) = Z5) # i. #Z5
NB. AM: global variable for string amend, revised 2003-7-31
Z5
)
 of=. str2box  NB. alias for string amend
 amdstr=. 3 : 0
 :
; (<x.) AM } y.           NB. AM: global variable
)

subs=. [. & ((((e.&) ((# i.@#)@)) (@))) ]]
```