Ｊ研究会資料　2006/11/25

Ｊによる微分方程式のグラフィック・アプローチ－その１・続き

Ｊのバージョンとウィンドウズ・グラフィックス

西川 利男、中野 嘉弘

１．はじめに

先月、「微分方程式の数値解と方向場表示」[1]なる発表を行ったが、そのプログラムの実行に際して、何人かの方から私（西川）の元にクレームが寄せられた。

私自身つい手慣れていることから、Ｊ３上でプログラム作成を行ったが、多くの方々からＪ４，Ｊ５の上ではエラーが出て実行できないという、ことであった。最初は、簡単な手直しで良いと思っていたが、やってみるとＪのグラフィックス命令がＪのバージョンによりかなり差異があることが分かった。

ちなみに私がＪ３にこだわるのは単なる保守主義ではなく、次の理由からである。

・それ以前のDOS-JPC, JFW, J2などと原始関数レベルで互換がとれている

・日本語表記が可能である

・Ｊのシステム自身がコンパクトであり、とくに私が愛用するJ305では、バージョンは古くてもリリースは５と非常に安定しており、いわゆる良く枯れている

しかしながら、新しいＪのユーザのためにも、やはりＪ４、Ｊ５上でも動くようにする必要を感じた。

実は、北海道、札幌市の中野嘉弘氏から、早速にプログラムを走らせようとしたら、エラーが続出で動かない。そして、私の対応よりも早く、ご自身で直され改定版を送って下さった。　従って、この報告はＪのバージョンの違いによるグラフィックス命令の差異をこの機会に私がまとめたものである。

２．Ｊ３とＪ４，Ｊ５とのグラフィックス仕様の差異

一口にグラフィックスというが、いろいろなレベルのものが考えられ、私の独断的分類では次のようになる。

・低レベルのグラフィックス（汎用）　isigraph, gl2, gl3

・高レベルのグラフィックス（簡便）　plot, gdgraph

・グラフィックス・アプリケーション　Excel, Maple, Mathematica

Ｊの低レベル、高レベルのグラフィックスとは、それぞれちょうどアセンブリ言語とBASIC, FORTRANなど高級プログラミング言語に相当するものと思う。

手軽に計算結果をグラフで表示したいといった目的には、plotやgdgraphは非常に良

く出来ていて便利である。かつ志村正人氏の親切な解説がある。

しかしながら、まとまったウィンドウズのシステム構築の中でグラフィックスを自分の思うように使いたいというときには、多少不便でも低レベルのグラフィックスを使うことになる。今回の私の例のように、方向場のグラフ表示で矢印を思いの場所に描くという目的には、gl2 の命令によらざるを得ない。

Ｊの低レベルのグラフィックスの仕様でＪ３以前とＪ４以降との大きな違いは次のとおりである。

Ｊ３以前　…　ウィンドウズ・ドライバ命令 wd の引数として、g で始まる文字列パラメータでグラフィックス操作を行う。　例えば

```
wd 'glines x0 y0 x1 y1'
```

Ｊ４以降　…　グラフィックス操作で上の仕様は取り除かれ、その代わりに平面グラフィックスには gl2(11!:2x)、 立体グラフィックスには gl3(11!:3x)の命令群が設けられた。上と同じ操作は次のようになる。

```
load 'gl2'
gllines x0 y0 x1 y1
```

なお、ウィンドウズそのものの操作命令はそのままである。このためグラフィックスの画面表示は次のようになる。

```
glshow ''      NB.  グラフィック・チャイドの表示
wd 'pshow'     NB.  親ウィンドウの表示
```

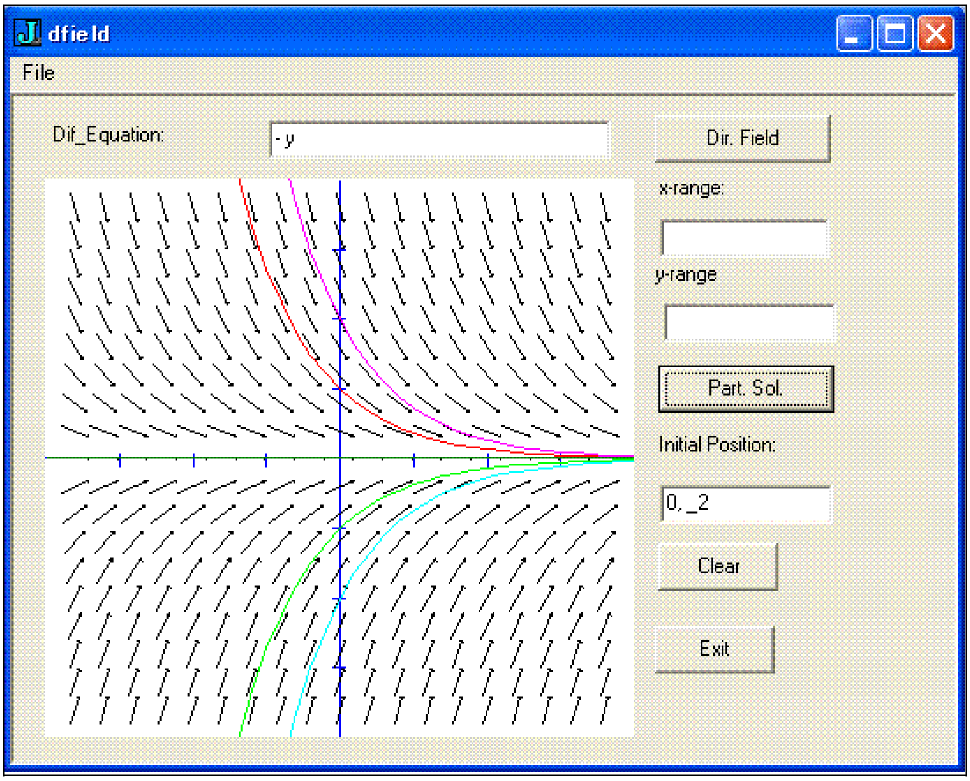
微分方程式グラフィックスのＪ４以降への改訂は、従って以上の変更を行えばよい。プログラムではかなりの行数の変更が必要になるが、Ｊのスクリプト編集画面上で[Edit]-[Find]-[Replace]で行えば、置換は容易にできる。

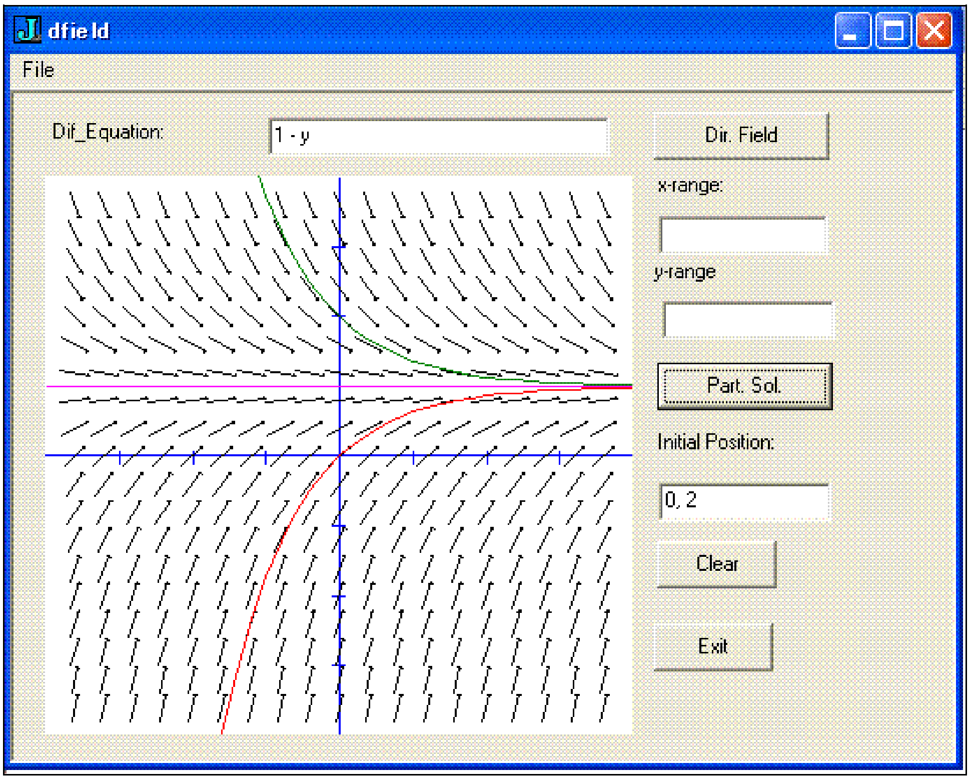
３．終わりに

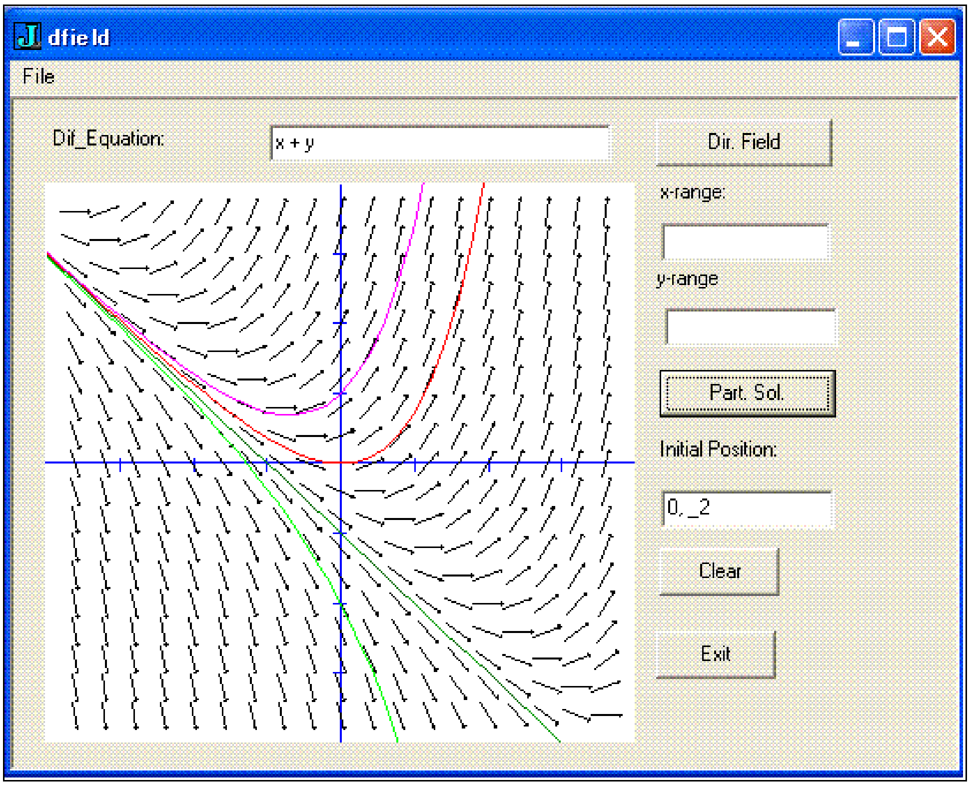
私の意図は前報にも述べたが微分方程式をもっと親しみ易く、道具として使ってもらいたいためであり、その意味であらためて種々の実行例を以下あげた。また、念のため中野氏作成によるＪ４，Ｊ５で実行可能な改定プログラムリストもあげた。

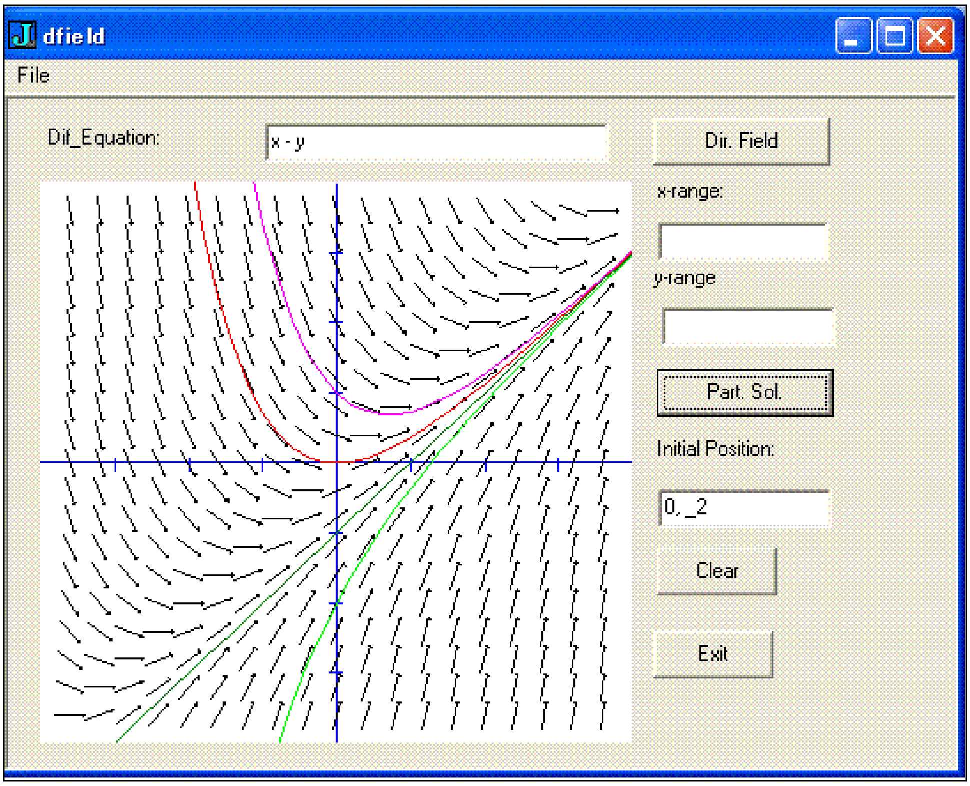
最後に改定プログラムをいただいた中野嘉弘氏に厚く御礼申し上げるとともに、本報告の著者として連名になっていただいた。いうまでもなく文責は西川にある。

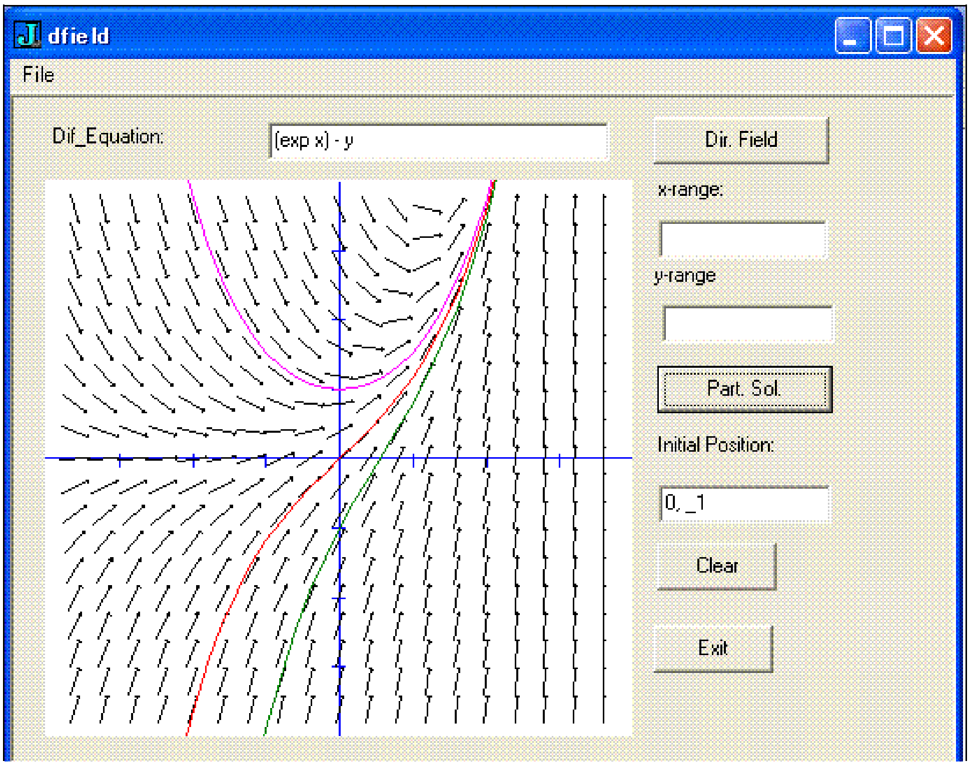
[1] 西川利男「Ｊによる微分方程式のグラフィック・アプローチ－その１
１階常微分方程式－数値解と方向場表示の活用」Ｊ研究会資料 2006/10/28

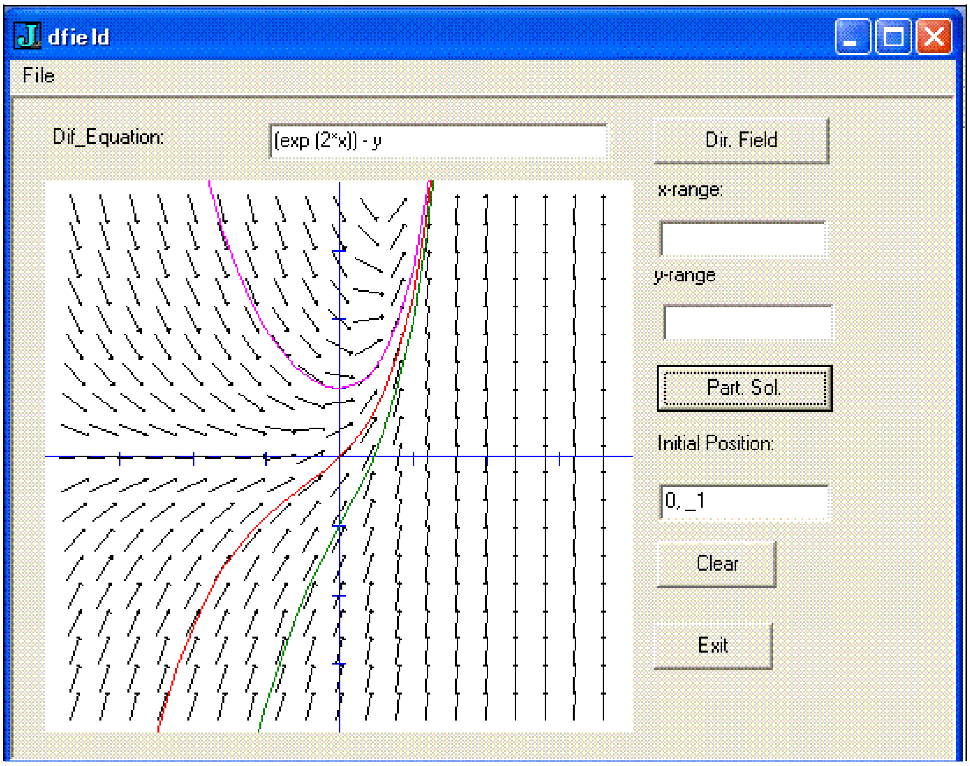

dfield
File
Dif_Equation:
- y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _2
Clear
Exit

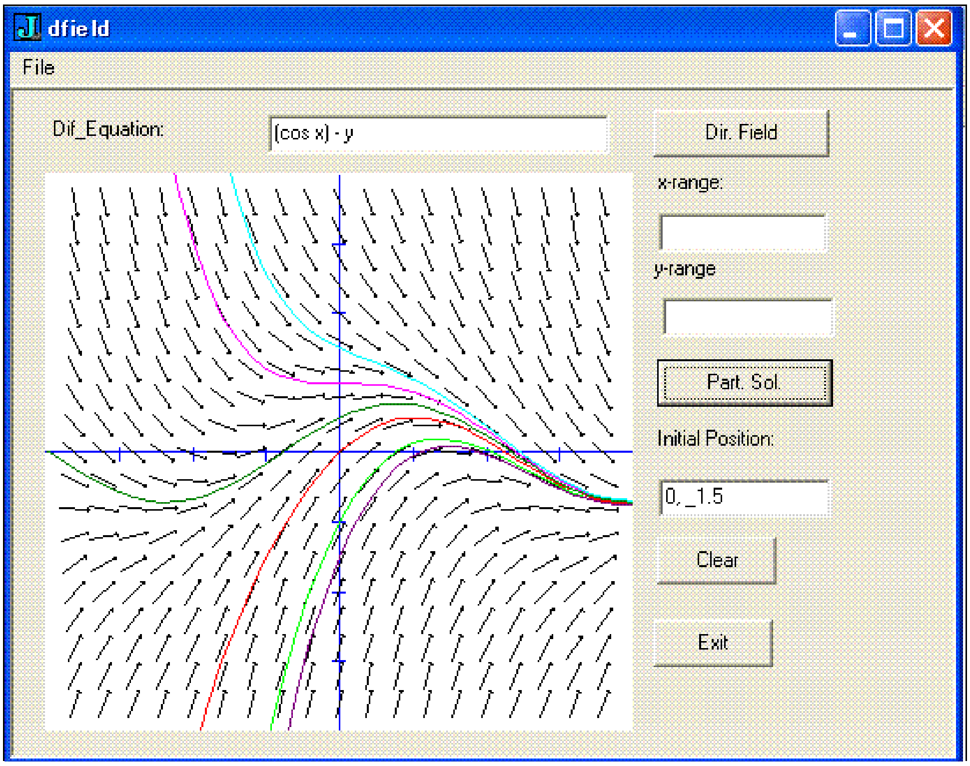

dfield
File
Dif_Equation:
1 - y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, 2
Clear
Exit

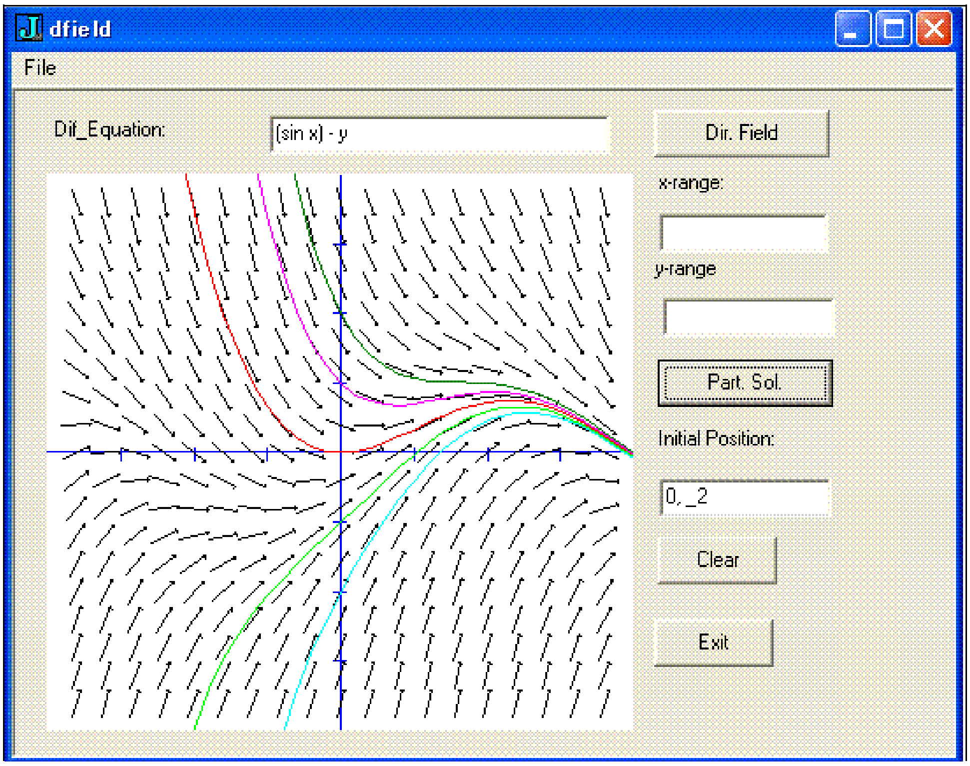

dfield
File
Dif_Equation:
x + y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _2
Clear
Exit

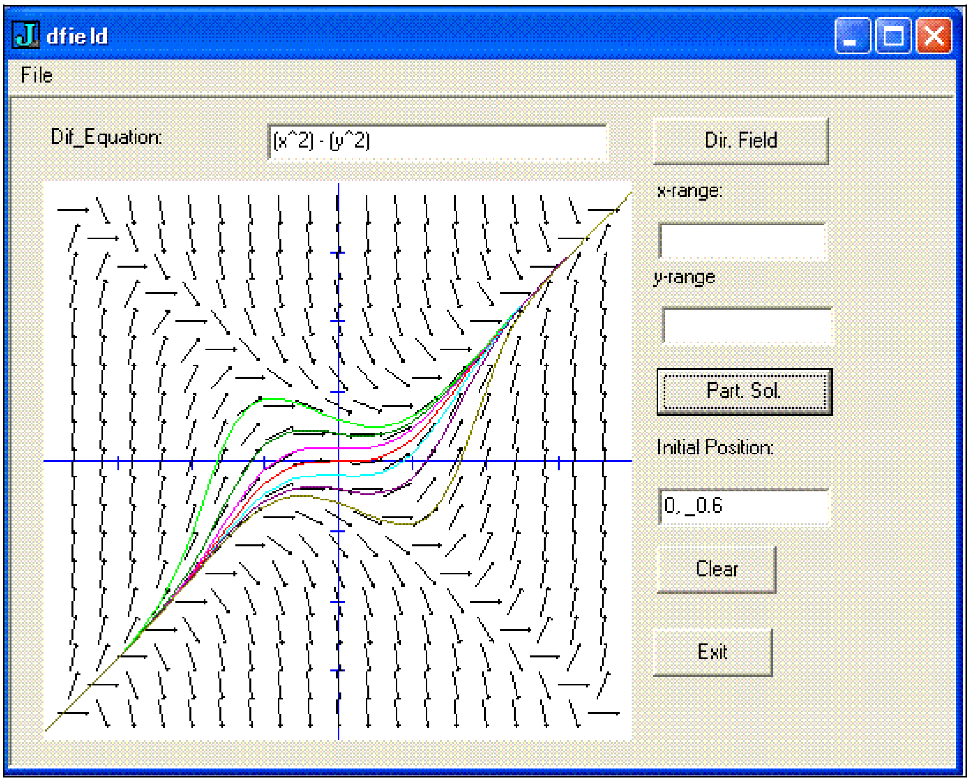

dfield
File
Dif_Equation:
x - y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _2
Clear
Exit

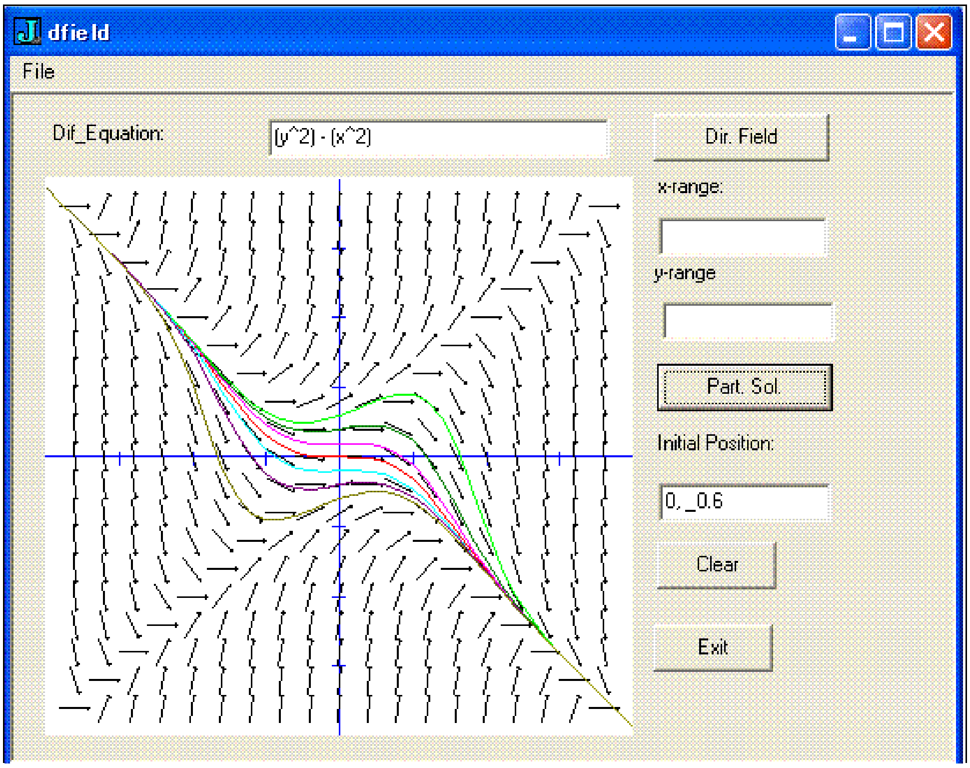

dfie ld
File
Dif_Equation:
(exp x) - y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _1
Clear
Exit


dfie ld
File
Dif_Equation:
(exp (2*x)) - y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _1
Clear
Exit


dfield
File
Dif_Equation:
(cos x) - y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _1.5
Clear
Exit


dfield
File
Dif_Equation:
(sin x) - y
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _2
Clear
Exit


dfie ld
File
Dif_Equation:
(x^2) - (y^2)
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _0.6
Clear
Exit


dfie ld
File
Dif_Equation:
(y^2) - (x^2)
Dir. Field
x-range:
y-range
Part. Sol.
Initial Position:
0, _0.6
Clear
Exit

プログラムリスト

```
NB. dfield J4,J5 version 2006/11/2 by T. Nishikawa
NB.     revised from suggestion by Y. Nakano
NB. ODE Direction Field and Numerical Calculation

require 'gl2'
require 'isigraph'
coinsert 'jgl2' NB. required only for J5

DFIELD=: 0 : 0
pc dfield;
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep;
menu exit "&Exit" "" "" "";
menupopz;
xywh 183 113 34 12;cc clear button;cn "Clear";
xywh 182 134 34 12;cc cancel button;cn "Exit";
xywh 9 21 167 141;cc graph isigraph;
xywh 182 5 50 12;cc go button;cn "Dir. Field";
xywh 11 7 50 10;cc label static;cn "Dif_Equation:";
xywh 68 6 108 11;cc DE edit ws_border es_autohscroll;
xywh 183 68 50 12;cc sol button;cn "Part. Sol.";
xywh 183 85 78 12;cc label static;cn "Initial Position:";
xywh 183 98 50 11;cc POS edit ws_border es_autohscroll;
xywh 183 20 42 10;cc label static;cn "x-range:";
xywh 183 31 49 11;cc xrange edit ws_border es_autohscroll;
xywh 182 42 50 10;cc label static;cn "y-range:";
xywh 184 52 50 11;cc yrange edit ws_border es_autohscroll;
pas 6 6;pcenter;
rem form end;
)

run =: dfield_run
```

```
dfield_run=: 3 : 0
wd DFIELD
NB. initialize form here
XR =: _4, 4
'XRA XRB' =: XR
YR =: _4, 4
'YRA YRB' =: YR
range_set ''
icol =: 0
wd 'pshow;'
)

dfield_clear_button=: 3 : 0
glclear ''
icol =: 0
range_set ''
glshow ''
)

dfield_close=: 3 : 0
wd'pclose'
)


NB. Imported from dfield.js J3 version ================================
NB. Ordinary Differential Equation(ODE)
NB. Direction Field Graphic Display 2006/9/27
NB. and DE Numerical Solution       2006/9/28
NB. revised for any x, y ranges     2006/10/6

NB. Conversion 'y / x' => '(1&{) % (0&{)'
df =: 3 : 0
require 'regex'
y =. y.
if. '-' = {. y do. y =. '0 ', y end.
y =. '\([[:space:]]*-' ('(0'&,@(}.)) rxapply y  NB. '(-' => '(0-'
```

```
y =. '[[:digit:]]+\.*[[:digit:]]*' (,&'"_') rxapply y
y =. ('\/';'%') rxrplc y
y =. ('sqrt';'%:@') rxrplc y
y =. ('sin';'1: o. ') rxrplc y
y =. ('cos';'2: o. ') rxrplc y
y =. ('tan';'3: o. ') rxrplc y
y =. ('cot';'4: o. ') rxrplc y
y =. ('exp';'^@') rxrplc y
y =. ('log';'^.@') rxrplc y
y =. ('x';'(0&{)') rxrplc y
y =. ('y';'(1&{)') rxrplc y
'(', y, ')'
)
```

```
NB. (d '*:@y - *:@x') dfunc 2 3 => 5
dfunc =: 1 : 0
'x y' =. y.
z =. ". u. ,' ','"(1) ', (":x),' ',(":y)
x, y, z
)
```

```
range_set =: 3 : 0
glclear ''
glrgb 0 0 255
glpen 1 0
draw"(1) > (XRA, 0, XRB, 0);(0, YRA, 0, YRB)       NB. draw x-axis and y-axis
draw"(1)  (((>:XRA)+i.(>:  XRB-XRA)),._0.1),"(1)  (((>:XRA)+i.(>:  XRB-XRA)),.0.1)
        NB. draw x-graduations
draw"(1)  (_0.1,.((>:YRA)+i.(>:  YRB-YRA))),"(1)  (0.1,.((>:YRA)+i.(>:  YRB-YRA)))
        NB. draw y-graduations
glshow ''
)
```

```
NB. draw collection of lines, given as x, y in array
draw =: 3 : 0
'x0 y0 x1 y1' =. y.
gllines"(1) (XR Adj x0), (YR Adj y0), (XR Adj x1), (YR Adj y1)
NB. wd 'glines ', ": (XR Adj x0), (YR Adj y0), (XR Adj x1), (YR Adj y1)
)

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

NB. Input Equation in Edit Box
dfield_DE_button=: 3 : 0
Eq =: DE
icol =: 0
)

NB. Color Data
COL =: 255 0 0;255 0 255;0 128 0;0 255 0;0 255 255;128 0 128;128 128 0;0 128 128

NB. DE numerical solution
dfield_sol_button=: 3 : 0
glrgb"(1) > (8|icol){COL
icol =: icol + 1
glpen 1 0
DA =. 0.2 (df Eq) rk^:(i. >: >. 5*XRB) PX, PY
'DXA DYA' =. |: DA
DXYA =. , (XR Adj DXA),.(YR Adj DYA)
DB =. _0.2 (df Eq) rk^:(i. >: >. 5*(-XRA)) PX, PY
'DXB DYB' =. |: DB
DXYB =. , (XR Adj DXB),.(YR Adj DYB)
gllines"(1) DXYB
gllines"(1) DXYA
glshow ''
)
```

```
NB. Input Initial Position(X, Y) in Edit Box
dfield_POS_button=: 3 : 0
'PX PY' =: ". POS
)



dfield_xrange_button=: 3 : 0
XR =: ". xrange
'XRA XRB' =: XR
dfield_clear_button ''
)

dfield_yrange_button=: 3 : 0
YR =: ". yrange
'YRA YRB' =: YR
dfield_clear_button ''
)

NB. adj =: %&4.0@(500&*@(4.0&+))
Adj =: 3 : 0
:
'a b' =. x.
(y. - a) * 1000 % (b-a)
)

NB. Direction Field Display
dfield_go_button=: 3 : 0
NB. I =: 0.4 * (-@(i.@-) , }.@i.) 10
IX =: XRA + 0.4*(>:i.19)
IY =: YRA + 0.4*(>:i.19)
IXY =. >,{ IX;IY
glrgb 0 0 0
glpen 1 0
0.2 draw_arrow"(1) (df Eq) dfunc"(1) IXY
)
```

```
draw_arrow =: 3 : 0
:
'ax ay' =. x. arrow y.
NB. AR =: , (_4, 4) Adj ax,.ay
AR =: , (XR Adj ax),.(YR Adj ay)
gllines"(1) AR
glshow ''
)

NB. Arrow Symbol
NB.  length arrow center slope
NB.   eg. 1 arrow _2 1 0.5
arrow =: 3 : 0
:
require 'plot'
d =. x.
'x0 y0 a' =. y.
NB. arrow figure
'p0x p0y' =: (-d), 0
'p1x p1y' =. d, 0
'p2x p2y' =. (0.9*d), (0.1*d)
'p3x p3y' =. (0.9*d), (_0.1*d)
'p4x p4y' =. d, 0
NB. plot (_4, 4, 4, _4, _4, p0x, p1x, p2x);(_4, _4, 4, 4, _4, p0y, p1y, p2y)
px =. p0x, p1x, p2x, p3x, p4x
py =. p0y, p1y, p2y, p3y, p4y
NB. rotate arrow
Cos =. 1 % %: 1 + *: a
Sin =. a % %: 1 + *: a
'xx yy' =. (2 2$Cos, (-Sin), Sin, Cos) (+/ . *) (px ,: py)
xx =. (x0) + xx
yy =. (y0) + yy
NB. plot (_4, 4, 4, _4, _4, xx);(_4, _4, 4, 4, _4, yy)
xx;yy
)
```

```
NB. Differential Equation ====================================
NB. Runge-Kutta argumented Dif_Equation /2006/9/5
NB.   inc (dif_eq)  rk^:(cycles) initial x, y
NB. Using d-verb, Enable Ordinary Notation as d 'y-x'
NB.   0.1 (df 'y-x') rk^:(i.11) 0 0
rk =: 1 : 0
:
h =. x.
x =: 0{ y.
Y =: 1{ y.
F =: u.
k1 =: ". F, (":x),',',(":Y)
k2 =. ". F, (":x + h%2),',',(":Y + h*k1%2)
k3 =. ". F, (":x + h%2),',',(":Y + h*k2%2)
k4 =. ". F, (":x + h),',',(":Y + h*k3)
(x+h), (Y + h*(k1 + (2*k2) + (2*k3) + k4)%6)
)

NB. Euler Adverb Defiition
NB. 0.1 (df'y-x') euler^:(i.11) 0 0
euler =: 1 : 0
:
h =. x.
'X Y' =. y.
k =. ". u. , (":X), ',', (":Y)
(X+h), (Y + h*k)
)

NB. 0.1 ((1&{)-(0&{)) euler0^:(i.11) 0 0
euler0 =: 1 : 0
:
h =. x.
'X Y' =. y.
k =. u. X, Y
(X+h), (Y + h*k)
)
```