

J / WIN の使用法

ビジネス・サイエンスグラフ [改訂版]

Masato SHIMURA
JCD02773@nifty.ne.jp
Last update 2005 04 23

目次

1	Jのグラフ	2
1.1	準備	2
2	2 D グラフ	2
2.1	numeric trig	6
3	データの型とグラフ	6
3.1	X;Y の値	6
3.2	関数のグラフ	8
3.3	散布図	8
3.4	マトリクスデータ	11
4	グラフの利用	12
4.1	ワープロへなどへの張り付け	13
4.2	TEX への張り付け	13
5	plot driver	14
5.1	PD 多彩なグラフィックコマンド	16
6	OOP と Multi	17

6.1	OOP	17
6.2	OOPの準備	17
6.3	Multi と Sub	18
6.4	2 axis	21
6.5	yrange	22
7	3D グラフ	23
7.1	3Dのデータ	24
8	ベクトル 等高線	27
8.1	ベクトル	27
8.2	contour	28
9	File の連続出力	29

プロフィール

- A ちゃん: PC は入門段階。時々難解な質問や無理な要望を出す。item W 君: 標準的なウィンドウズユーザー
- P さん: OS マニア。J は使い込んでいる。
- D 先生: Net に出没しているらしい。

1 J のグラフ

A) データのグラフが書きたい。折角 J で計算したのだから J で書きたい。

W) J のグラフィックスのレベルがグンと上がって、実用的になったようですね。例えば、関数解析をしながら、グラフを書いてみる、書いたグラフをプリントする、 \LaTeX や HTML に張り付けると言ったことが、簡単に出来るようになった。

P) 噂ではドイツ産の巨大会計ソフトのグラフィックに J が活躍しているようだ。このときにぐっとレベルが上がりグラフ専用ソフトと比べても遜色がない。

アプリケーションの感覚で簡単に使える。更に、J504 で plot 周りが格段に強化されたようだ。

1.1 準備

P) 最初にクラシックスタイルでやってみよう。まず、`plot` をロードしよう

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

`load 'plot'` だけでもいいが、`trig` は円関数、`numeric` はグラフィックスの大変便利なユーティリティー群なので、同時に load しておこう。

三つのファイルの順番は任意だ。

2 2 D グラフ

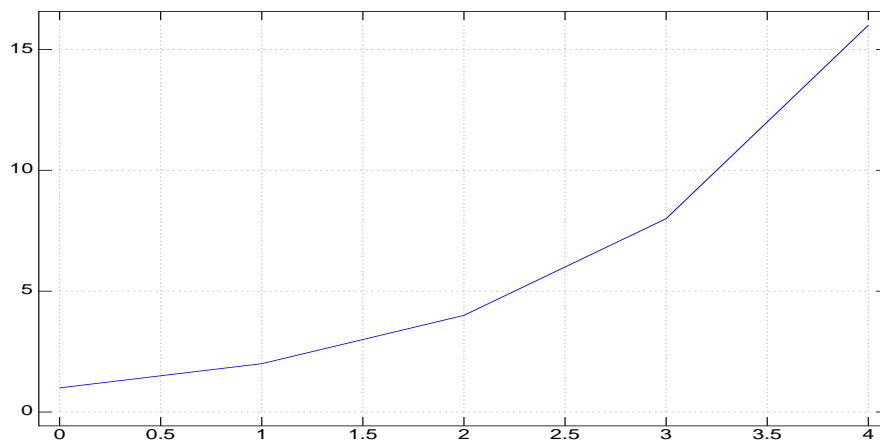
W) 最初に紹介するグラフィックの関数は `plot` と `pd` だ。plot は単独で使える手軽で便利な関数だ。pd は論文に張り付けたり、本格的なプレゼンテーション用に出力する本格的な方法だ。pd とは plot driver のことで、開発者はローレベルの関数とも言っている。

W) **J Primer** にサンプルがあるので、やってみよう。

```
plot 5 10 23 45 8
plot 1+i.100
'bar' plot i.5
plot sin i.16
```

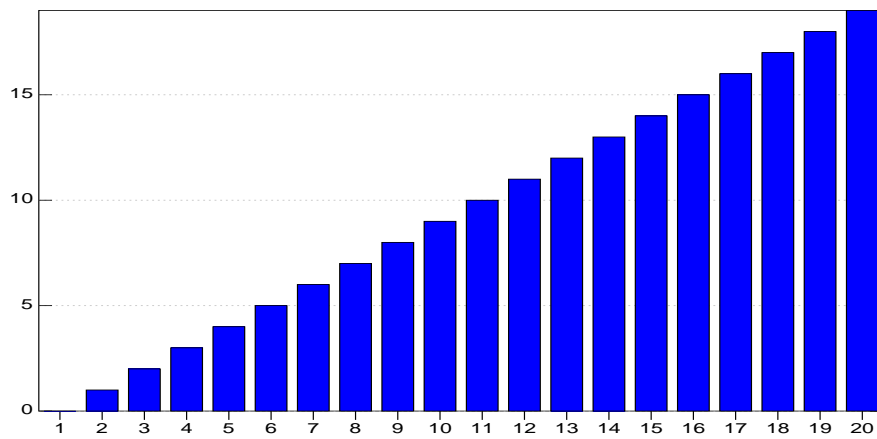
```
'bar' plot 2^i.5
pd 'eps \doc\ws\j\jpap53_02.eps'
```

```
'bar' plot i.20
pd 'eps \doc\ws\j\jpap53_02.eps'
```



W) なかなか良くできていますね。すばらしい。デフォルトの折れ線とバーチャートは簡単に出来ますね。

P) plot の解説は **Studio |Lab** の Graphics の中に **plotPackage** が入っている。Primer も Lab も英文だがさっと目を通すと良い。Lab は **Ctrl + J** でグラフが次々に出てくるので、そのコマンドの記述だけ、先ず目を通せばよい。3D まで簡単に紹介されている。更に本格的なサンプルは **Studio Demos** に plot が入っている。ここまで出来るという見



本だ。少し進んだらソースコードをなぞるのが良い勉強法だ。

W) plot パッケージの基本命令は **pd plot** の 2 つの様ですね。

それにしても 2 D は機能が良く揃っている。

A) どの様な種類のグラフがあるのですか。

<p>2D グラフの種類 (左パラメータ)</p>	<ul style="list-style-type: none"> • line デフォルト • point • dot 微小な点 • marker 大きな点 (J504 以降) • bar • errorbar • fbar NB.Floating barchart • sbar NB.Stacked barchart • stick NB.Stick bar • hilo NB.High-Low 2 3 • pie • radar NB. レーダーチャート • density 濃淡 • contour 等高線 • area 面で描画 • poly 回帰直線とデータで囲まれたの範囲を塗る。 	<pre>'point' plot 1 2 3 4 5;3 4 5 6 7 'point' plot (20?100);(20?100) 'sbar' plot >:?>: i.3 5 'fbar' plot >:?>: i.3 5 'radar' plot 4 6 \$ 24 ? 100 'pie' plot 1 2 3 4 5</pre>
<p>右パラメータ</p>	<p>x;y plot に渡すデータ (X 軸の値;Y 軸の値) X Y を一個ずつ組み合わせて plot する。 x の値の省略時 Y の値のみ指定されたときは、X は自動割り付け</p> <p>*1</p>	<pre>data=. 1 2 3 4;2 4 7 9 plot 1 2 3 4;2 4 7 9 'point' plot data plot 1 2 3 4 5 x=. steps 0 10 100 plot x;sin x plot steps 0 20 100</pre>

2.1 numeric trig

P) **numeric trig** の 2 つのファイルの簡単な解説は、**system/main** の中の **numeric.ijs, trig.ijs** のスクリプトをプリントして、コメントを読まなければならない。
trig は単なる三角関数のようだ。

util numeric.ijs の util	steps Xの値を細かく取って、滑らかな曲線を描く。 steps 0 10 100 (from 0 to 10 in step 100) Xの値を 0 から 10 まで、0.1 刻みの 100 個の数	plot (sin;sin*cos) steps 0 10 100
	range range 2 10 range 2 10 0.1 Xの値を 2 から 10 まで、0.1 刻みの数	plot (];sin) steps 0 10 100
いろいろ 試してみよう。		plot (sin;]) range 0 10 0.01

3 データの型とグラフ

3.1 X;Y の値

A) さらさらと流されても、Jのデータ構造とグラフの種類がわからない。

W) Jの **plot** のデータの基本は次のとおり。

X軸とY軸を(;)でリンクして、XとYをボックスの状態で連結する。これがJのグラフィックスの基本形だ。

2 4 3 5 ; 7 4 6 2

2 4 3 5 9	7 4 6 2 2
-----------	-----------

- ・ X軸は一行のリストのデータが基本。(X の値を省略し、Y の値のみをパラメーターとして与えた場合は、X の値は (i.# Y) で Y の個数が自動的にカウントされ、割り付けられる。)
 - ・ Y軸はリストやマトリクスをとる。(ボックスでX と結合する。)
- ラミネート (:;) でリストを積み重ね、マトリクスにすることも出来る。
- ・ X軸のデータ個数とYのデータ長は同じでなければならない。

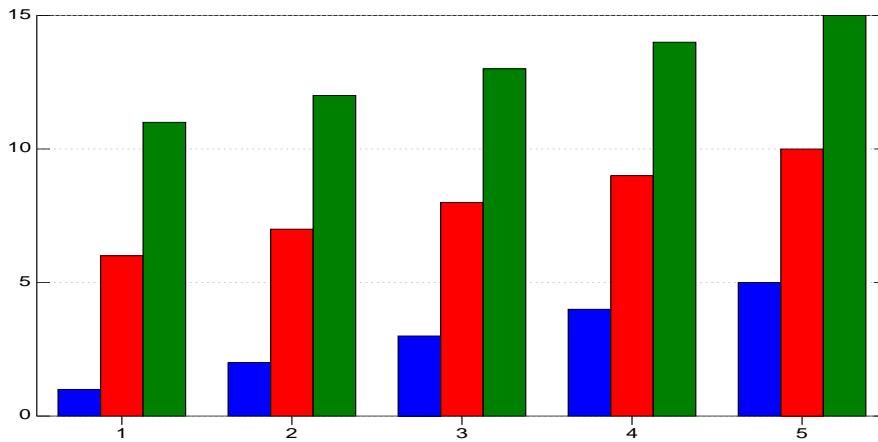
A) 何本かのラインを描きたいときは？

W) マトリクスを使えば良いでしょう。x軸の値は省略できる。色も適当に付けてくれる。好みに指定したいときは、本格的な pd を用いる。

```
>: i. 3 5
```

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
```

これは、行毎に三本のラインを書いてくれる。'bar' plot >:i.3 5 では、1 6 11 と言うように、列を組み合わせた5組のバーチャートが表示される。少し重ね合わせると三次元になりそうだ。



3.2 関数のグラフ

A) 関数をダイレクトにグラフにしたいときはどうしたら良いのですか。

W) plot の右パラメータに関数を持ってこることも出来るようですね。関数の後ろに区間を指定すれば、その計算結果を plot に渡し、グラフが作成されます。

P) これは、関数が計算しているのであって、plot の機能ではない。だから、どんなに難しい関数を持ってきても計算してくれるが、一度、計算結果を TMP などを受けておく方がよい。

$y = 3x$	plot *&3 i.10
$y=x^2$	plot ^&3 i.10

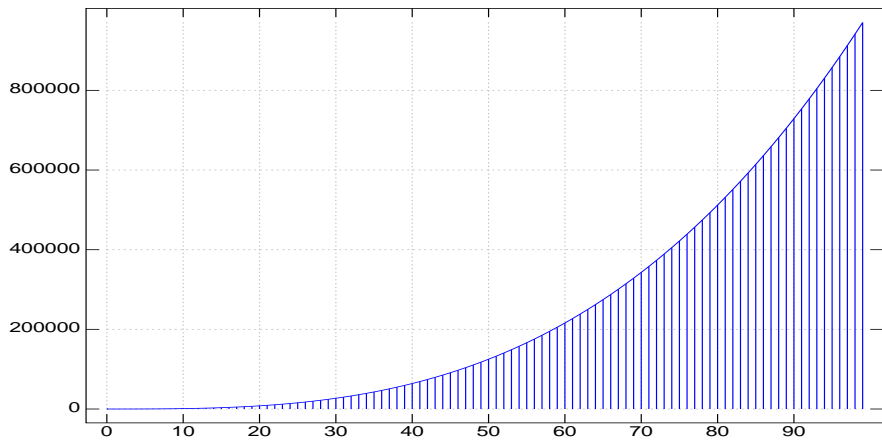
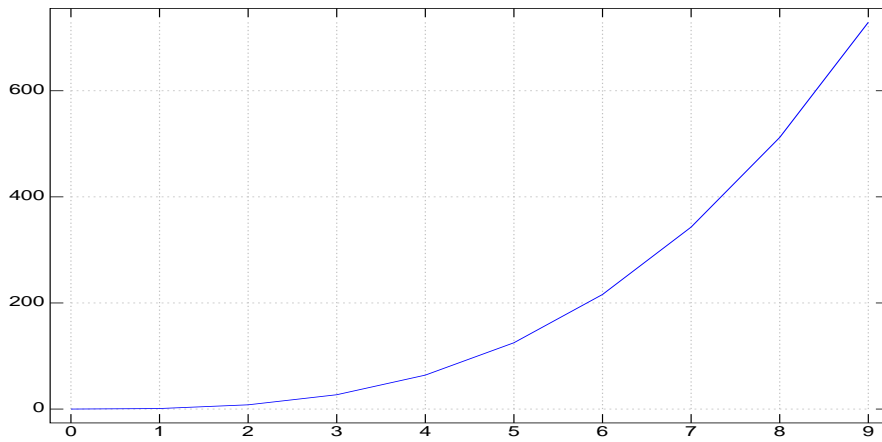
これでやってくれる。次はちょっと高等テクニックです。

```
'line,stick' plot ^&3 i.100
pd 'eps \doc\ws\j\jpap53_041.eps'
```

3.3 散布図

A) 散布図と回帰直線を同時に書きたい。

P) 散布図は



'point' plot x;y

で描ける。x;y で x と y の値を相互に組み合わせしてくれる。point dot は点が小さすぎるようだ。

J504 で marker がサポートされた。点のはっきりと出る。mark の種類も指定できるようだ。

マニュアルには載っていないが poly というタイプもあり、直線回帰をした上で、データとの間を塗ってくれる。

数値を line で描いてくれる。

P) a1 の型から回帰して散布図を描くスクリプトを pd の方で作ってみよう。

```
|: a1=. ?. 20 2 $ 100
46 79 54 60 60 46 13 51 78 90 31 60 64 13 76 25 68 42 99 99
```

```
55 52 39 57 94 78 18 92 60 62 16 64 71 3 26 77 48 91 97 9
```

```
a2=. {|: a1
```

```
{: a2 reg L:0 {. a2
```

```
+-----+
```

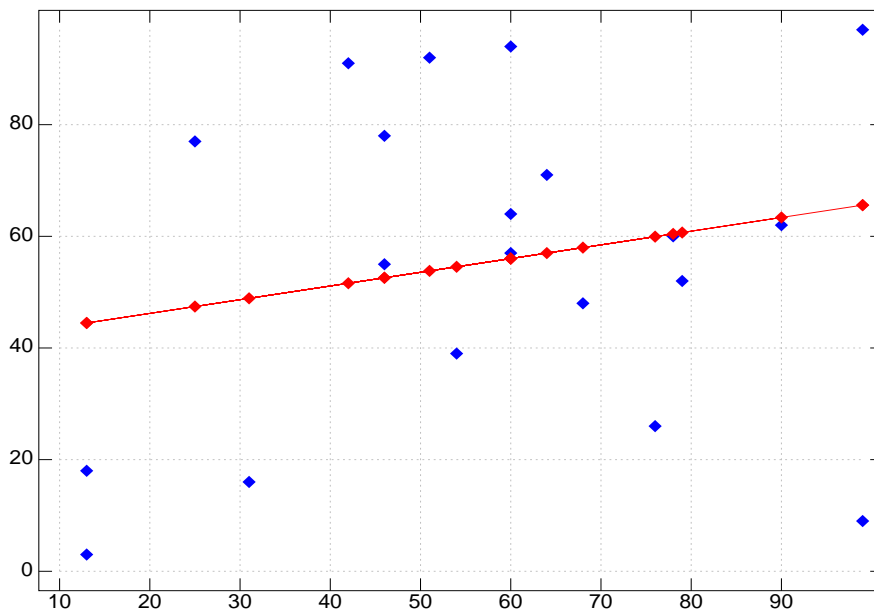
```
|41.285 0.245495|
```

```
+-----+
```

```
f=. (> {: a2 reg L:0 {. a2)&p.
```

```
sanpu a2
```

```
pd 'eps \doc\ws\j\jpap53_05.eps'
```



P) Aちゃん、メープルシロップ入りの紅茶がいいね。

A) Wさんにも入れて差し上げましょう。

```
sanpu=: 3 : 0
```

```
pd 'reset'
```

```
pd 'type marker'
```

```

pd y.
pd 'color red'
pd 'type line marker'
pd ({. y.},< f y.
pd 'show'
)

reg2=:3 : '>({: y.) %. (L:0) 1 ,. (L:0) {. y.'
f=: 3 : '(reg2 y.)&p. >{. y.'

```

3.4 マトリクスデータ

W) 先ほど、何本かのラインを描くときに、マトリクスを使うことを試みましたがマトリクスのデータを plot はどう取り扱っているのが興味がありますね。

P) 2 D系はラインポイント系、バー系、レーダー・パイ系に分けられるようだ。

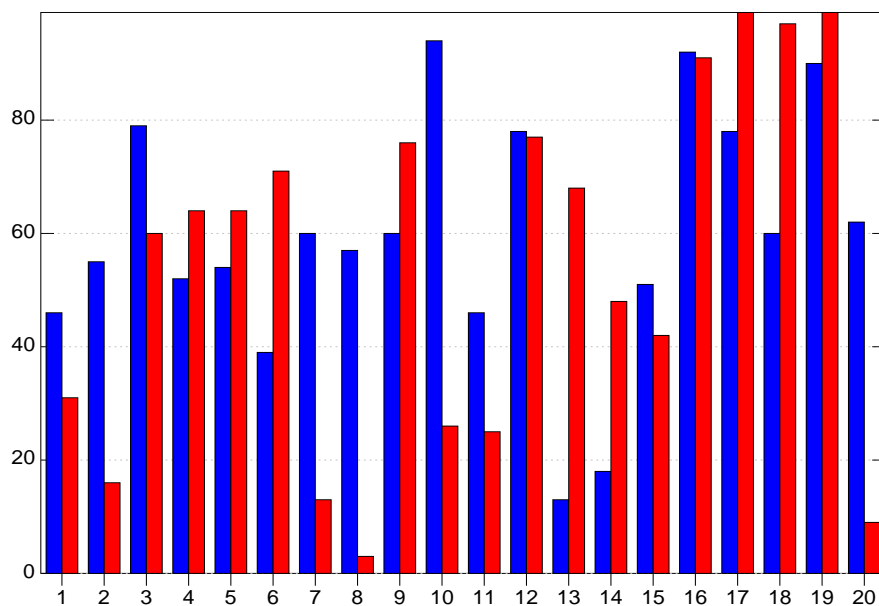
```

a1=. ?. 2 20 $ 100
a=. 20?.100
+-----+
|46 99 23 62 42 44 12 5 68 63 76 84 53 83 69 97 74 82 10 79|
|
+-----+
+-----+
|46 55 79 52 54 39 60 57 60 94 46 78 13 18 51 92 78 60 90 62|
|31 16 60 64 64 71 13 3 76 26 25 77 68 48 42 91 99 97 99 9|
+-----+

```

Y のマトリクスの値は複数のデータとして取り扱われる。色は自動で付けてくれる自分で指定したいときは、 pd 'color green,blue,red,... と指定する。

左が X 軸、右のマトリクスが y の値となる。



ラインポイント系は、xの値をしっかり使っている。

バー系は stick 以外はxに関しては、個数が重要で、値に興味は無く、yのマトリクスデータ行をデータ単位として plot している。

レーダー系もバー系と類似している。

4 グラフの利用

W) pd のグラフの利用に関するコマンドが整備されて使いやすくなった。 プリント、クリップボードへのコピー、ビットマップファイルの作成が、簡単に出来る。

print	プリンターでの印刷	pd 'print'
クリップボードへのコピー	pd 'clip ' pd 'clip w h' w 幅 h 高さ 単位 0.01mm. デフォルトは 10000 7500	pd 'clip' pd 'clip 7500 10000' wmf 型式でクリップボードに格納され、簡単にワープロなどに張り付けられる。

eps pdf 形式 でのセーブ	J504 でサポート。 pd 'eps temp.eps' pd 'pdf temp.pdf'	
wmf 形式で セーブ	pd 'save temp.wmf w h'	pd 'save temp.wmf' pd 'save temp.wmf 100000 75000' wmf Windows Metafile
bmp 形式で セーブ	pd 'savebmp temp.bmp w h '	pd 'savebmp temp.bmp 10000 10000' pd 'savebmp temp.bmp1 ' ビットマップ形式 24bit color

4.1 ワープロへなどへの張り付け

P) セーブしたファイルは、クリップボード以外は、temp\foo.eps と指定しないと J のルートに出てくるようだ。

クリップボード経由は何の問題もなく MS-WORD などのワープロに張り付けられる。ワープロの編集・張り付けを用いれば良いし、なかなか綺麗だ。プレゼンテーションソフトでの使用も問題無い。

```
pd 'clip'
```

これがおすすだ。また、今までにも何回か pd 'epstemp.eps' でセーブした。

4.2 T_EX への張り付け

P) T_EX は、DVI ドライバによって、扱える画像ソフトに多少差があるようだが、ポストスクリプトの eps が一番簡単だ。

最近/mediabb/のパッケージがサポートされ、バウンディングボックスの煩雑さからも解放された。

```
\usepackage{mediabb}
```

eps での取り込みは次のように簡単に出来る。

```
'surface' plot i. 10 10 10
```

```
pd 'eps \doc\ws\j\jpap532_05.eps'
```

```
pd 'pdf \doc\ws\j\jpap532_03.pdf'
```

HTML へは PDF の方が便利で、Acrobat Reader で画像を読むことになる。

5 plot driver

A) plot driver ってなにですか。

W) 今までにも、

```
pd 'eps temp\foo.eps'
```

```
sanpu=: 3 : 0
```

```
pd 'reset'
```

```
pd 'type marker'
```

```
pd y.
```

```
pd 'color red'
```

```
pd 'type line marker'
```

```
pd ({. y.},< f y.
```

```
pd 'show'
```

```
)
```

のように登場していた。詳細は、UserManual の plot に書いてあるが、やプレゼンテーション用に本格的なグラフを描くことですね。

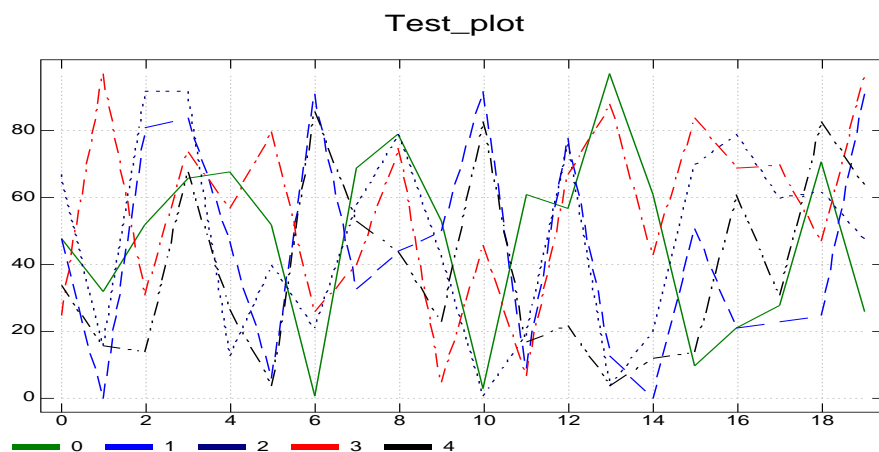
P) 上の sanpu でかなりのことはやっている。

5.0.1 type

W) type は前に整理してあります。pd では、途中で、代えられるので、色と組み合わせると多彩な表現が可能です。UserManual の Package/plot の中の plotTypes に種類が列挙してあります。

5.0.2 Key

P) このあたりはこちゃごちゃ書くより、サンプルを見た方がいいのではないか。あまり綺麗ではないが、必要な物は詰まっている。自分が何を加えたいかを決めて、マニュアルやデモを見ると答えはある。



```
plot_test_0=: 3 : 0
pd 'reset'
pd 'title Test_plot'
pd 'penstyle 0 1 2 3 4'
pd 'pensize 2'
pd 'color green,blue,navy,red,black'
pd 'keypos bottom outside left'
pd 'keystyle horizontal'
pd 'key 0 1 2 3 4'
pd ? 5 20 $ 100

pd 'show'
)
```


5.1 PD 多彩なグラフィックコマンド

W) プレゼンテーションには見栄えも大事ですね。まずは、サンプルを見てみましょう。実データ以外は、クォテーションで囲んでいる。

A) PD って何だったかな。

W) マニュアルには次のように書いてある。

pd is the low-level verb that handles all calls to Plot.

plot is a cover verb for pd that will handle most simple uses of Plot.

```
pd 'new'
```

```
pd 'type bar'
```

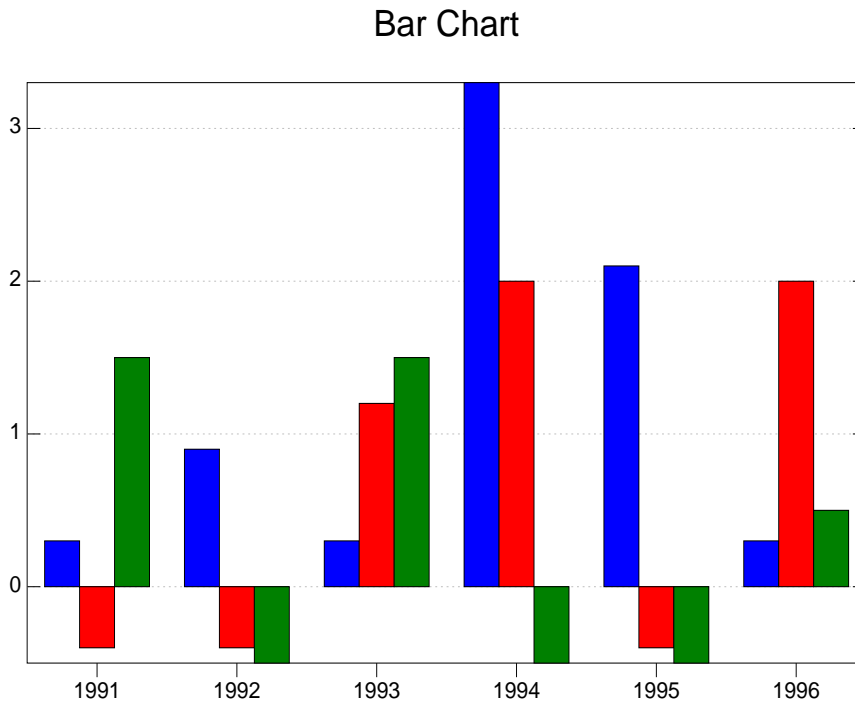
```
pd 'border 0'
```

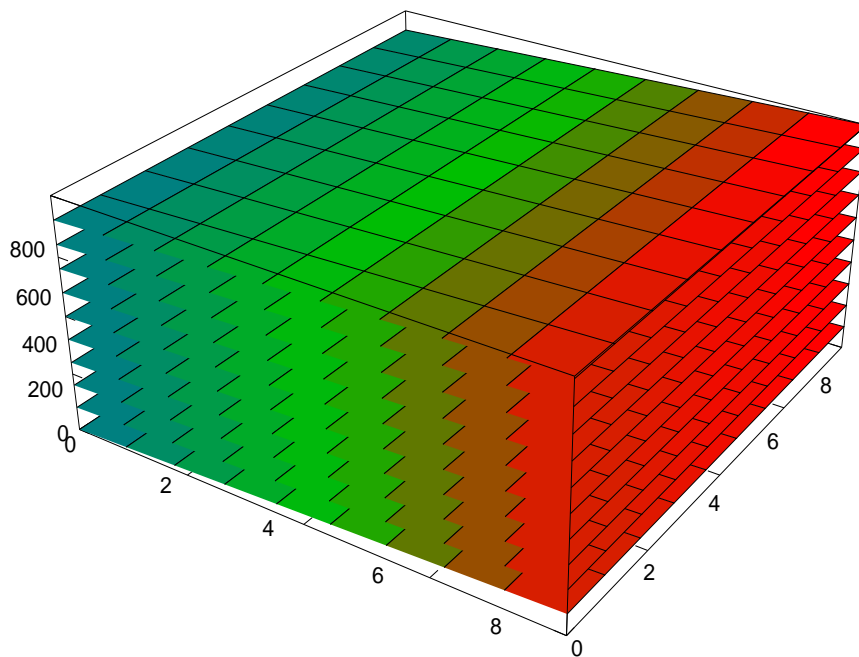
```
pd 'xlabel "1991" "1992" "1993" "1994" "1995" "1996"'
```

```
pd 'title Bar Chart'
```

```
pd 0.6 0.8 1* _0.5+?3#,:3 3 4 7 7 4
```

```
pd 'show'
```





6 OOP と Multi

6.1 OOP

A) オブジェクトって何ですか。

P) OOP は簡単に言えば、同じ関数を複数の環境で同時に使えるように機能アップしたものだ。

A) OOP を使うと、どう便利になるのですか。

W) `conew` で指定した枚数のグラフィックウィンドウを開くことが出来るので、何枚かのグラフを見比べることができますね。

6.2 OOP の準備

OOP では `plot` の代わりに `JZPLOT` をロードする。

```
load 'jzplot trig numeric'
```

```
a=.conew 'jzplot'
```

```
b=.conew 'jzplot'
```

P) OOP にすると、plot の画面が a,b・・・と conew で定義した数まで使える。
OOP を使うときは、plot の代わりに plot__a plot__b としなければならない。面倒ならば

```
pla=.plot__a  
plb=.plot__b
```

のように簡略化した再定義を行えば良い。

OOP を使わないのであれば、単に plot だけで良い。

OOP では__とアンダーバーを2個連続して用いる。

```
plot__a 5 10 23 45 8
```

```
plot__b 5 10 23 45 8
```

```
'sbar'plot__a i.4 7
```

6.3 Multi と Sub

A)OOP ってカッコいい割に実用性が今一でないの。

W) 何か面倒くさい感じがしますね。

A) 次の例の方が実用的でないの。

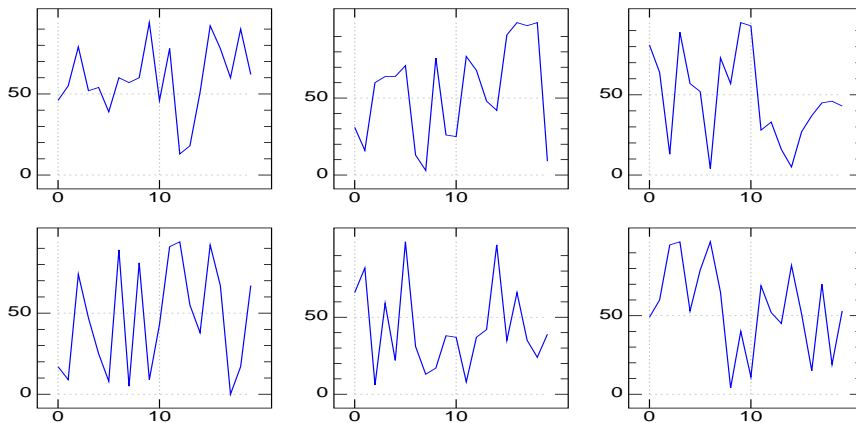
P)J504 からサポートされたマルチグラフだ。デモは複雑なので、簡単に書いてみた。

```
plot_multi0 ?. 6 20 $ 100  
pd 'eps \doc\ws\j\jpap53_11.eps'
```

/pd 'multi 1 1 , 1 1 1/' と pd ' ' ;"1 y. ' のところがコツの様だ。サンプルは、画像の大きさを変えているが、同じサイズの方が実用性が大だ。

```
plot_multi0=: 3 : 0  
pd 'reset'  
pd 'multi 1 1 , 1 1 1'  
pd ' ' ;"1 y.
```

```
pd 'show'
)
```

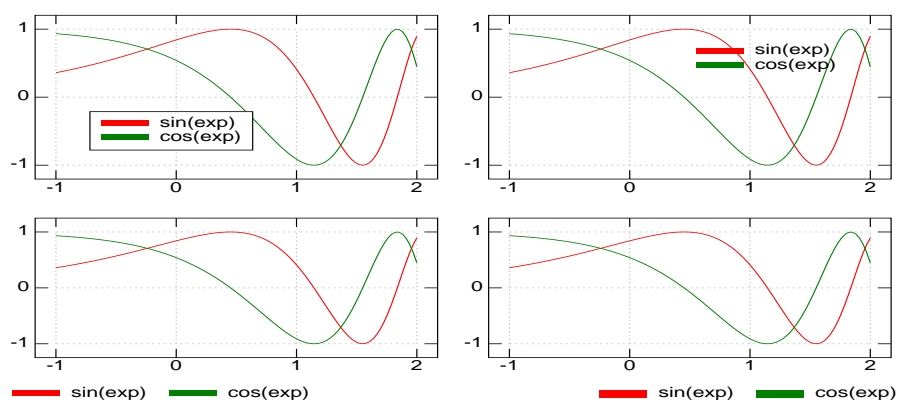


W) Sub も J504 でサポートされましたがこちらの方も本格的なグラフの様ですね。

```
DKEYS ''
```

```
pd 'eps \doc\ws\j\jpap53_12.eps'
```

Key Styles



D) Demo に少し手を加えて、動くようにしてみよう。これをサンプルとして、作り込めばよい。長そうだが、一つ作って COPY して修正する。

OOP はほとんど出番がないようだ。

```
x=: steps _1 2 100
d1=: x;sin ^x
d2=: x;cos ^x
```

```
NB. =====
DKEYS=: 3 : 0
```

```
pd 'textfont arial 25 bold italic'
pd 'textcolor darkslateblue'
pd 'textc 500 _10x Key Styles'
```

```
pd 'sub 0 0 1000 _40x'
pd 'sub 2 2'
```

```
pd 'new'
pd 'color red,green'
pd 'keypos b'
pd 'key sin(exp),cos(exp)'
pd d1
pd d2
```

```
pd 'new'
pd 'color red,green'
pd 'keystyle o'
pd 'keypos r'
pd 'key sin(exp),cos(exp)'
pd d1
pd d2
```

```
pd 'new'
pd 'color red,green'
pd 'keystyle ho'
pd 'keypos bo'
```

```

pd 'key sin(exp),cos(exp)'
x=. steps _1 2 100
pd d1
pd d2

pd 'new'
pd 'color red,green'
pd 'keystyle hof'
pd 'keypos bro'
pd 'key sin(exp),cos(exp)'
pd d1
pd d2

pd 'show'
)

```

6.4 2 axis

A) 経済誌などに形が似ているが、よく単位の違うグラフをのせているが、やってみたい。
W) J504 で y2axis がサポートされて、簡単に出来るようになった。y2axis と書き込むだけです。

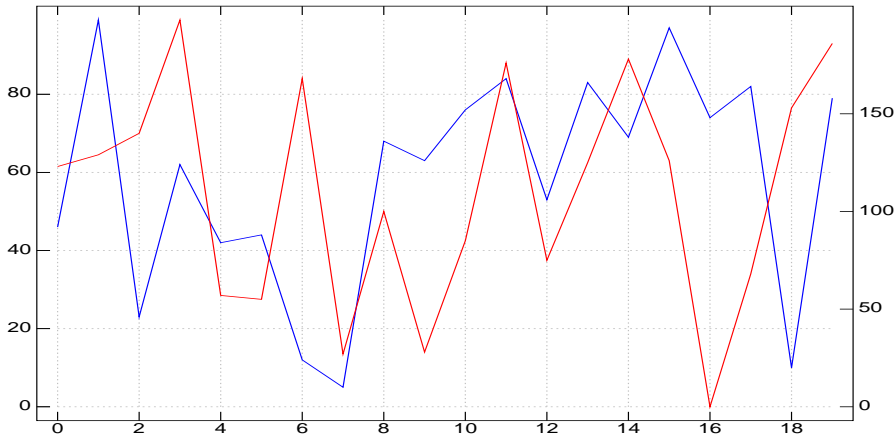
```

plot_multi1 ''
pd 'eps \doc\ws\j\jpap53_13.eps'

plot_multi1=: 3 : 0
pd 'reset'
pd 20 ? . 100
pd 'color red'
pd 'y2axis'
pd 20 ? 200
pd 'show'

```

)



6.5 yrange

y 軸の座標はデフォルトではオートになっているが、yrange で明示的に与えることも出来る。書式は pd 'yrange 最小値 最大値' である。これを自動的に与える関数を作ってみた。

yrange との接続が多少複雑なので、書き留めておく。

yrange の後にスペースが必要。

MIN,MAX は文字化して接続する。

```
maxmin=: 3 : '(>./>./ y.),(<./<./y.)'
```

```
plot_y2axis=: 3 : 0
```

```
MIN=:|. maxmin {. y.
```

```
MAX=:|. maxmin {: y.
```

```
pd 'reset'
```

```
pd 'yrange ',":MAX NB. 'yrange+space' and need ":
```

```
pd {: y.
```

```
pd 'y2axis' NB. change y2axis
```

```
pd 'yrange ',":MIN
```

```
pd {. y.
```

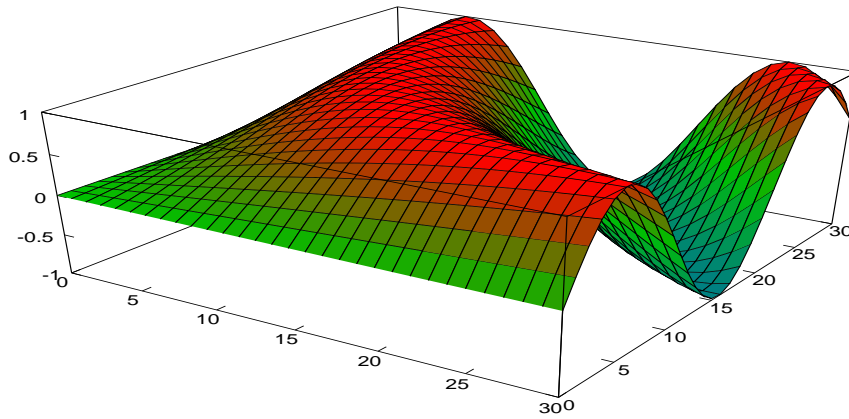
pd 'show'
)

7 3D グラフ

W) 3 D グラフもよさそうですね。

P) 2 D と同様だが、SURFACE VIEWPOINT などの 3D コマンドがある。当然、データ構造は x;y;z となる。

<p>m のパラメータ 3 D グラフの種類</p>	<p>density NB.2D expression of 3D data line stick NB. vertical bars surface wire NB.Wire Surface</p>	<pre>data=.sin */~ steps 0 3 30 'surface' plot data (sombrero) 'stick' plot data x=.y=.steps _3 3 50 z=.(cos % 3&+) x (+/&:*:) y 'wire' plot x;y;z</pre>
<p>3D の機能</p>	<ul style="list-style-type: none"> • viewpoint • viewsize 	<pre>'surface;viewpoint 1 0 0.7;viewsize 1 1 0.2'plot_a sin */~ steps 0 3 30</pre>



W) EXCEL のような簡易立体棒グラフはないようですね。stick は、完全な三次元チャートですね。

7.1 3Dのデータ

7.1.1 Line

まず、オードソックスなところから。x;y;z の形のデータ

```
plot (i. 10);(i. 10);i.10
```

```
d2
```

```
+-----+-----+-----+
|0 1 2 3 4 5 6 7 8 9|0 1 2 3 4 5 6 7 8 9|0 1 2 3 4 5 6 7 8 9|
+-----+-----+-----+
```

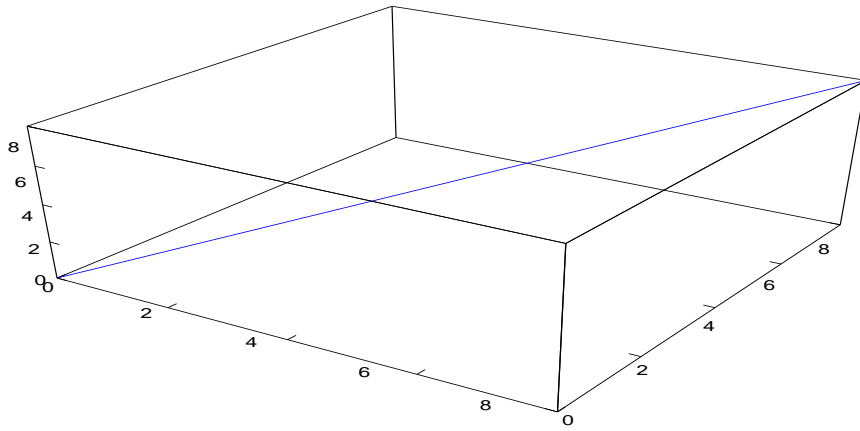
この形に、'line' と型指定をすると何故かエラーになる。

```
'line' plot (i.10);(i.10);i.10
```

3面ともマトリクスにすると

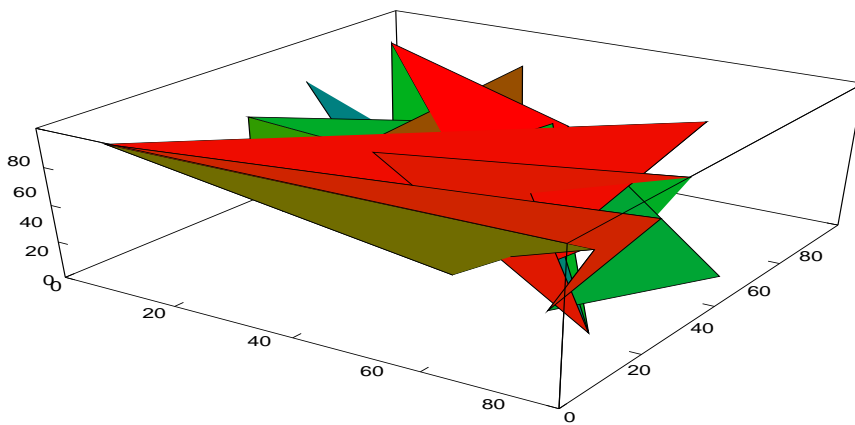
```
] d4=. (? 5 5 $ 100);(?5 5 $ 100);(?5 5 $ 100)
```

```
+-----+-----+-----+
```



```
|39 44 47 38 27|31 69 11 76 52|19 98 24 81 13|
| 7 36 25 67 67|55 58 32 70 14|39 60 17 82 15|
|51 72 72 94 46|16 48 86 81 55|98 25 23 10 21|
|94 32 46 51 66|73 31 13 52 31|63 69 79 69 75|
|40 60 98 15 67|58 51 43 25 37|66 63  5 59 22|
+-----+-----+-----+-----+
```

plot d4

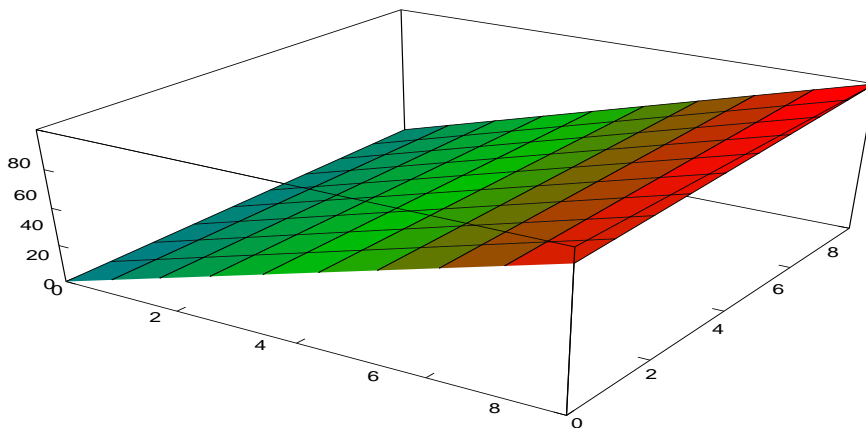


7.1.2 Wire Surface

'surface' plot i.10 10

P) この系はX Y軸は単に個数(#)を、Z軸が個々のマトリクスの実数をとって高さを描いているようだ。

```
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
```



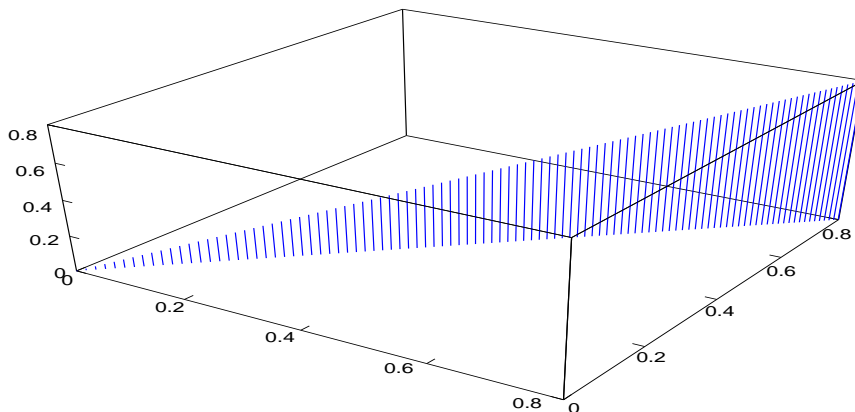
サーフェス系は、オードソックスなボックスに入ったリスト形式の、x;y;z 型のデータを受けつけないようだ。

7.1.3 Stick

P) デモの `plot` の 3D に `stick` が入っている。J は `bar` より高機能な `stick` を導入したようだ。

`stick` は 2D では、バリバリと棒グラフを描いてくれるが、3D は相当繊細なようで、大きな実数には向かないようだ。

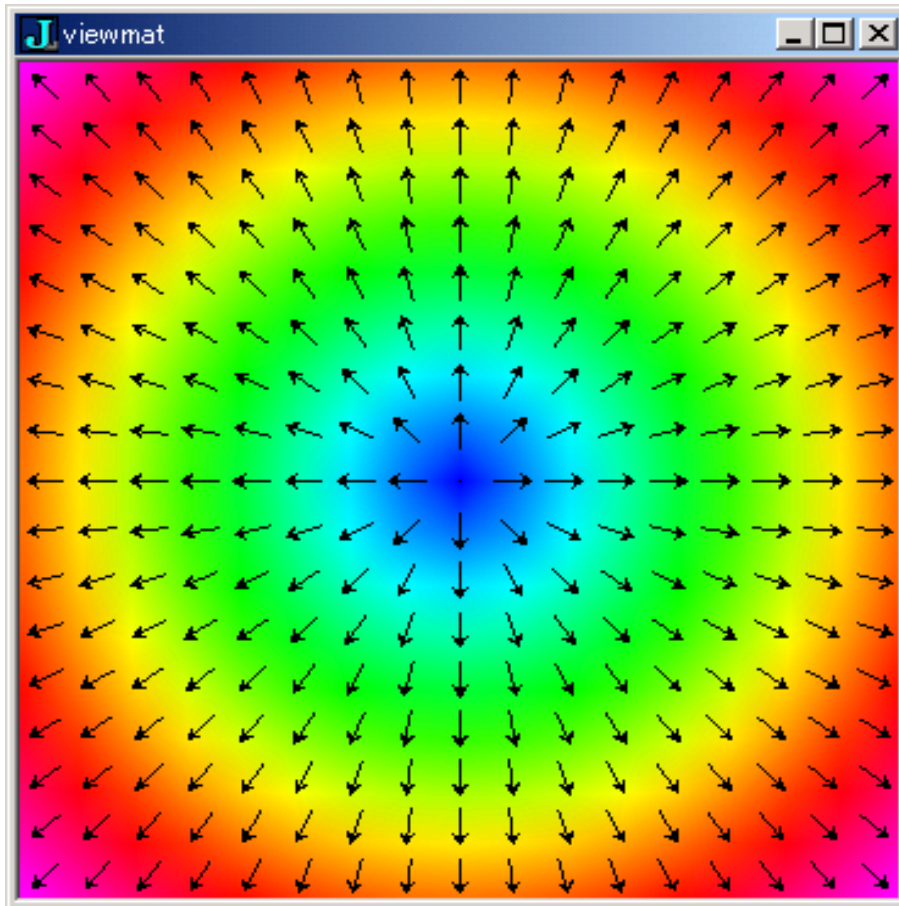
```
dat=. sin steps 0 1 100
pd 'new'
pd 'type stick'
pd dat;dat;dat
pd 'show'
```



8 ベクトル 等高線

8.1 ベクトル

P)J504 でサポートされた機能に、`viewmat` の強化がある。Lab の `viewmat` を見てほしいが、リカーシブプロットが簡単に出来る。更に、複素数ベクトルの矢印が表示できるようになっている。線形代数の教科書に出ていることもあるが、力学や電気工学など、応用範囲は広いようだ。



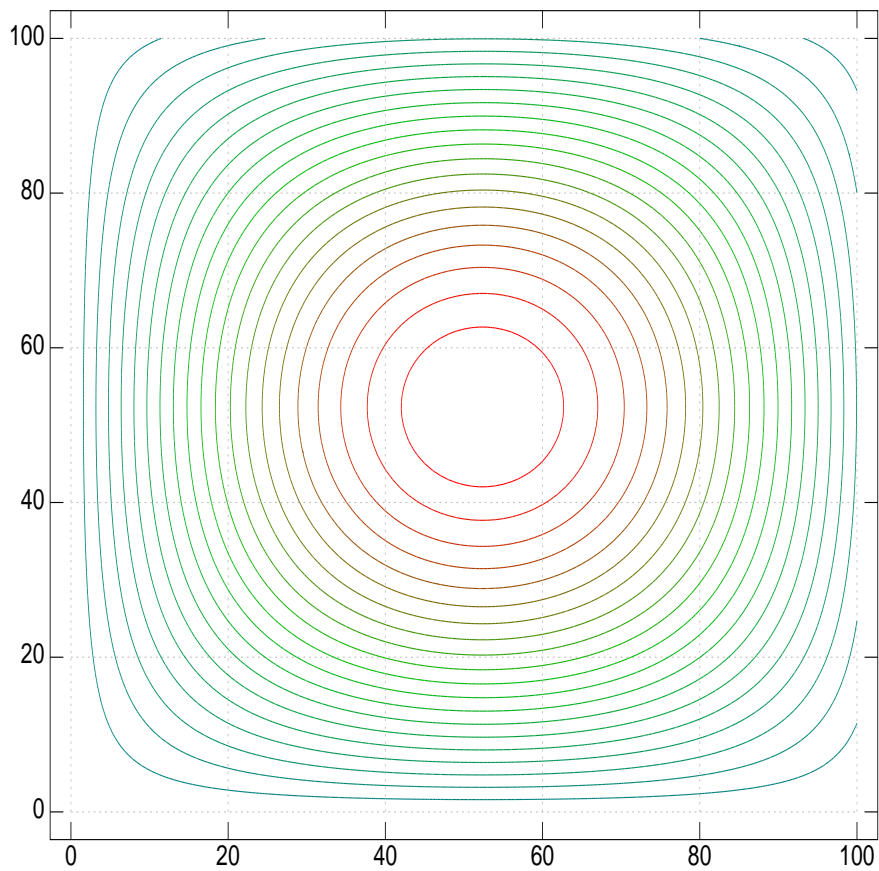
8.2 contour

A) 等高線って何ですか。

W) 文字通り等高線です。J504 からのサポートで、以前は色の濃淡でやっていた。

```
'contour' plot (sin */ sin) steps 0 3 100
  pd 'eps \doc\ws\j\jpap53_15.eps'
```

```
plot_contour ''
  pd 'eps \doc\ws\j\jpap53_14.eps'
```



9 File の連続出力

P) pd の中でループを組めばファイルの連続出力は出来る。J504 では pdf eps の出力の時、セーブを一々効いてくる

次のようなファイルを作り、edit|configure|extended programs に登録しておく
と連続出力できる。

ファイル名	do_pdf.bat
記述する一行	Echo off