JAPLA 研究会資料 2020/10/17

J-gl2 グラフィックスによるベクトルの和と内積の理解

西川　利男

線形代数の基本となるベクトルとその内積の理解には、数値を使った表現よりも、具体的な幾何学的図形を利用した方法がもっととられて良いのではないだろうか。

前回の JAPLA の例会で、志村氏の発表[1]と共に、日経ソフトウェアの「Procssing を用いたプログラミングで理解する数学」[2] という話題が紹介された。

[1] 志村正人「プログラミングで理解する数学－ベクトル」JAPLA, 2020/9/10.
[2] 吉岡直人「Procssing を用いたプログラミングで理解する数学」、
日経ソフトウェア、2020 年 9 月号、p. 94-の 100, 日経 BP (2002).

これらの発表に刺激されて、同様なことをＪのグラフィックスでやってみた。

１．ベクトルをＪの考え方で理解する

ふつう数学ではベクトルを空間内における矢印などと表現しているが、これはあまりに抽象的すぎて、頭に描くのは困難であると私は思う。

ベクトルをＪの名詞、動詞の考え方でつぎのように理解する。

・ベクトルとは数値の集合であり名詞である。決して動きを伴う動詞ではない。
・ベクトルの和（＝合成）を求める演算操作は動詞である。
・行列とは、ベクトルの和や積などを使って、もっと複雑に行う動詞である。
・ベクトルの内積とは２つのベクトルの関係（＝属性）を示す１つの値＝名詞である。

ベクトルのイメージを形づくるには、数値の集合よりも、グラフィックスの図形の方がずっと容易である。そのためには、Ｊのグラフィックスは極めて有用である。

２．J-gl2 を用いたグラフィックス

ふつうはＪのプログラミング操作は ijs で行い、その結果は ijx 画面でみる。
しかし、Ｊグラフィックスでは、グラフィックス画面で結果をみることになる。それを行うにはＪの gl2 グラフィックスが必要である。また、Ｊの plot ルーチンは便利であるが、関数値を表示するのでないので、やはり不十分であると思う。

ところでＪの gl2 グラフィックスプログラムについて、私が常用する J4 版と最新の J9 版とはかなり違っている。従って、それぞれの版について、プログラミングを行うことになった。

３．線形代数の基本－ベクトルの和と内積

まず、線形代数の教科書をひもといてみよう。[3]

[3] 田河生長ら「線形代数」大日本図書(1993).

そこには、平面上のベクトルとして、有向線分、単位ベクトルなどの定義のあと、ベクトルの演算として

$$a+b=\overline{A}B+\overline{C}D$$

のような式が延々と述べられている。

しばらくして、ベクトルの成分表示として

$$a=(a_x,a_y)$$

として表せるという式が出てくる。

つづいて、ベクトルの内積として、次の式が現われる。

$$\vec{a}\cdot\vec{b}=|\vec{a}||\vec{b}|\cos\theta$$

その間、もっぱら文字だけで、数値は出てこない。格調は高いが非常に抽象的である。

もっと具体的に数値をあげて、図により説明してみよう。ベクトルをグラフ用紙の上で表すには、大きさに応じた位置をきめて、原点と結んだ矢印として表す。これを位置ベクトルと呼んでいる教科書もあるが、この語が良いと思う。位置ベクトルとは図形であり、数値の並びとしてのベクトルと関連はあっても違うので、混同しないこと。

$$VA=(2,4)$$

もうひとつ別なベクトルをつくる。それぞれのベクトルはＪでは名詞である。

$$VB=(5,-1)$$

このベクトルを使って、ベクトルの和という演算 + を行う。これは動詞である。

$$VC=VA+VB=(2+5,4+(-1))=(7,3)$$

この操作はベクトルの合成とも呼ばれる。

さらに、ベクトルの内積とはそれぞれの要素を掛けたものの和である。

$$\text{内積}=(2\times5)+(4\times(-1))=10-4=6$$

これは、ひとつの値であり名詞である。

内積は何のために使われるのか？　２つのベクトルの間の関係、属性を表す。例えばこの値が０なら、直交するという属性を示す。

ここまでの流れを以下、Ｊのグラフィックスで見てみよう。

J4 と J9 とで異なるので、それぞれ別々に示す。

３．J4 におけるベクトル代数のグラフィックスによる理解の実際

３．１　入力ボタンからのベクトル値入力

まずは実行のようすを示す。J のプログラムは最後に示した。

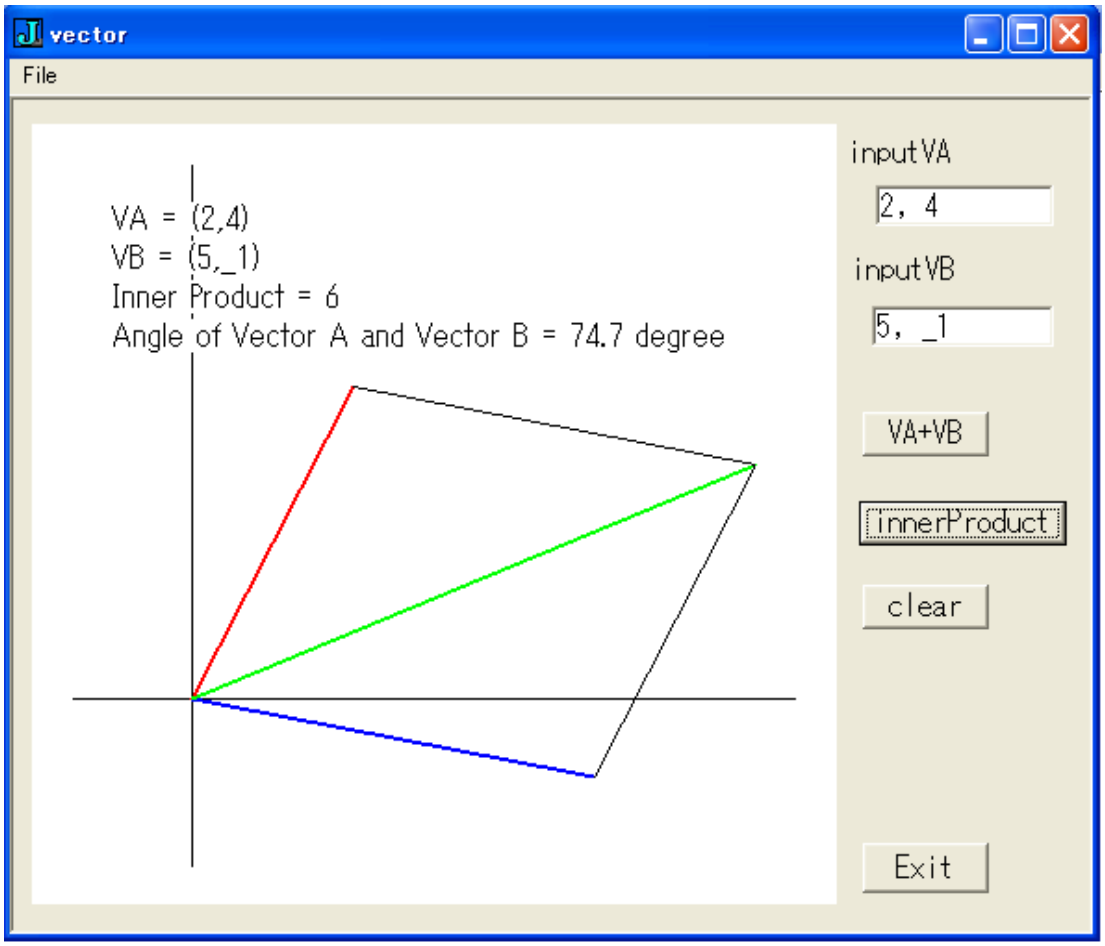
run ''

により、グラフィックス画面があらわれたら、マウスの右ボタンを押すとその点がグラフの原点になる。つまり任意の位置で、ベクトル表示を見ることができる。

inputVA の入力ボックスに 2, 4 と打ち込むと、赤でベクトル VA が表示される。同様にして inputVB の入力ボックスに 5, _1 と打ち込むと、青でベクトル VB が表示される。ここでベクトルとは位置ベクトルの意味である。

続いて、ボタン VA+VB を押すと、緑でベクトルの和が表示される。
ボタン innerProduct を押すと、ベクトルの内積が計算され表示される。
同時に２つのベクトルの間の角度が示される。

違う値で行うときには、clear ボタンで画面表示を消去する。その後、
あらためて別の値で上の操作を行えばよい。

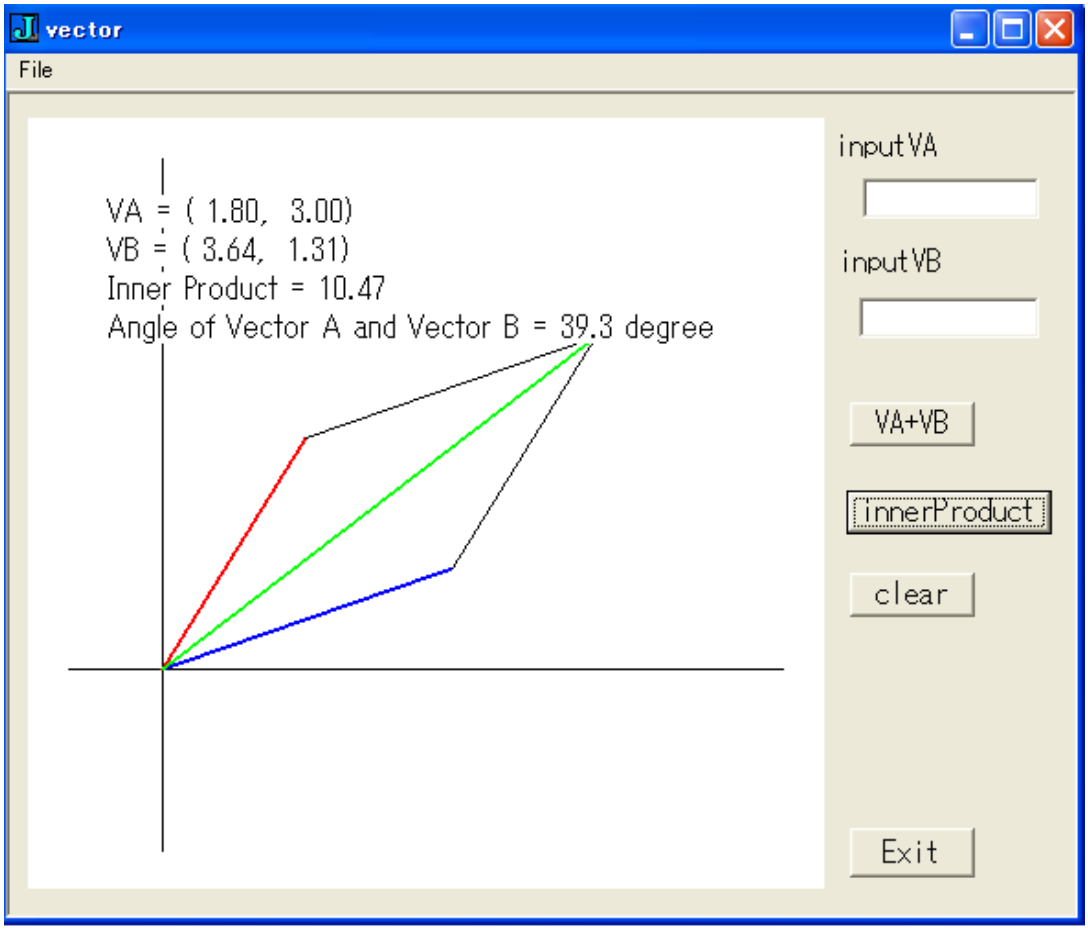

vector
File
VA = (2,4)
VB = (5,_1)
Inner Product = 6
Angle of Vector A and Vector B = 74.7 degree
inputVA
2, 4
inputVB
5, _1
VA+VB
innerProduct
clear
Exit

３．２　マウスボタンによるベクトルの入力

さらに、便利な方法として、マウスの左ボタンをクリックすると、画面の任意の位置でベクトルを入力できる。

赤と青の２つのベクトルを入力してから、次にボタン VA+VB を押すと、前と同様に緑でベクトルの和が表示される。

さらに、ボタン innerProduct を押すと、ベクトルの内積が計算され表示される。



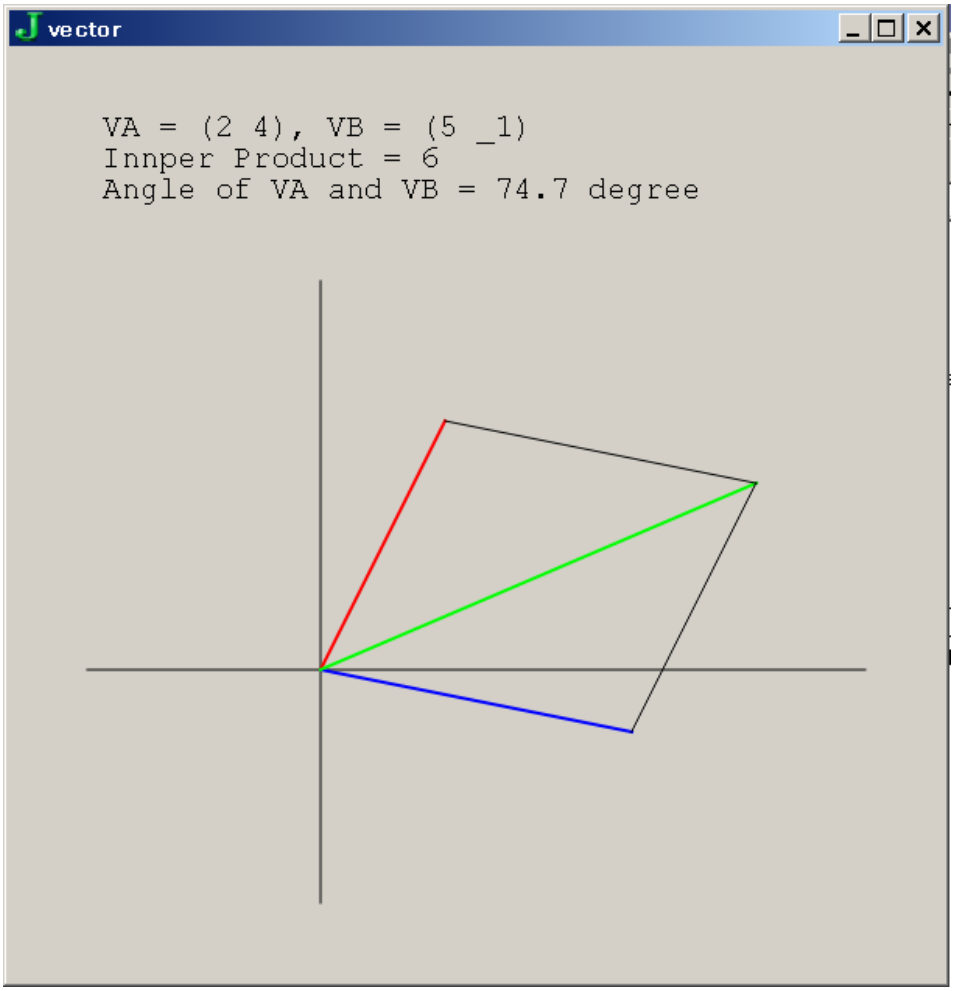
４．J9 におけるベクトル代数グラフィックスの実際

J9 でのグラフィックスでは、画面上で値の入力や操作の実行ができないので、プログラムの実行時に run の引数として、次のように行う。

```
run (2, 4);(5, _1)
```

すると次のように実行画面が一度で現れる。

J9 版プログラムは最後に示した。



Jプログラム　－　J4-gl2 グラフィックス

NB.

Vector

Arithmetic

```
load 'trig numeric'
load 'gl2'

VECTOR=: 0 : 0
pc vector;
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 229 178 34 12;cc cancel button;cn "Exit";
xywh 5 6 217 187;cc vectorarith isigraph;
```

```
xywh 229 116 34 11;cc clear button;
xywh 228 96 56 11;cc innerProduct button;
xywh 225 9 37 10;cc label static;cn "inputVA";
xywh 232 20 49 11;cc inVA edit ws_border es_autohscroll;
xywh 226 37 50 10;cc label static;cn "inputVB";
xywh 231 49 50 11;cc inVB edit ws_border es_autohscroll;
xywh 229 75 34 11;cc vectorC button;cn "VA+VB";
pas 6 6;pcenter;
rem form end;
)

MSG =: 'Set XY-Origin by Mouse Right Buton'

run =: vector_run
vector_run=: 3 : 0
wd VECTOR
NB. initialize form here
wd 'pshow;'

glfont "courier new bold"
gltextxy 100 900
gltext MSG
glshow ''
)

vector_close=: 3 : 0
wd'pclose'
)

vector_cancel_button=: 3 : 0
vector_close''
)

vector_clear_button=: 3 : 0
glclear ''
glshow ''
```

```
)



NB. draw x-axis and y-axis by mouse_right_down =======
vector_vectorarith_mbrdown=: 3 : 0
d=. ". sysdata
x0=:(0{d) * 1000 % (2{d)  NB. global
y0=:(1{d) * 1000 % (3{d)  NB. global
glclear ''
glrgb 0 0 0
glpen 1 0
glmove x,y
gllines 50, y0, 950, y0
gllines x0, 50, x0, 950
glshow ''
)

length =: %:@(+/@:(*:))

vector_inVA_button=: 3 : 0
VA =: ". inVA
glrgb 255, 0, 0
glpen 4, 0
gllines x0, y0, (x0 + {. 100 * VA) , (y0 + {: 100 * VA)
glshow ''
)

vector_inVB_button=: 3 : 0
VB =: ". inVB
glrgb 0, 0, 255
glpen 4, 0
gllines x0, y0, (x0 + {. 100 * VB) , (y0 + {: 100 * VB)
glshow ''
)
```

```
vector_vectorC_button=: 3 : 0
VC =. VA + VB
glrgb 0, 255, 0
glpen 4, 0
gllines x0, y0, (x0 + {. 100 * VC) , (y0 + {: 100 * VC)

glrgb 0, 0, 0
glpen 1, 0
gllines (x0 + {. 100 * VA) , (y0 + {: 100 * VA), (x0 + {. 100 * VC), (y0 + {:
100 * VC)
gllines (x0 + {. 100 * VB) , (y0 + {: 100 * VB), (x0 + {. 100 * VC), (y0 + {:
100 * VC)
glshow ''
)
```

```
NB. inner product  ==========================================
inprod =: (+/@:(*"(0)))

vector_innerProduct_button=: 3 : 0
gltextxy 100 900
gltext 'VA = ', '(', ("": {.VA), ',', ("": {:VA), ')'
gltextxy 100 850
gltext 'VB = ', '(', ("": {.VB), ',', ("": {:VB), ')'

inPAB =: VA inprod VB
gltextxy 100 800
gltext 'Inner Product = ', "": inPAB

angAB =. (180%1p1) * (arctan ({: % {.) VA) - (arctan ({: % {.) VB)
gltextxy 100 750
gltext 'Angle of Vector A and Vector B = ', (4j1"": angAB), ' degree'
glshow ''
)
```

Jプログラム－J9版

```
NB. Vector Arithmetic
NB. vector_arith_J9.ijs
NB. revised for J901

load 'trig numeric'

NB. load 'gl2'
load 'graph'
coinsert 'jgl2'


VECTOR=: 0 : 0
pc vector closeok;
minwh 600 600;cc g isigraph flush;
pas 0 0;
)

run =: vector_run
vector_run =: 3 : 0
wd VECTOR
'VA VB' =: y
wd 'pshow'
)

vector_g_paint=: 3 : 0
gllines 50, 400, 550, 400  NB. x-axis
gllines 200, 150, 200, 550  NB. y-axis

NB. VA =: 2, 4
glrgb 255 0 0
glpen 2
gllines 200, 400, (200+40*{.VA), (400-40*{:VA)

NB. VB =: 5, _1
glrgb 0 0 255
```

```
glpen 2
gllines 200, 400, (200+40*{.VB), (400-40*{:VB)

VC =: VA + VB
glrgb 0 255 0
glpen 2
gllines 200, 400, (200+40*{.VC), (400-40*{:VC)

glrgb 0 0 0
glpen 1
gllines (200+40*{.VA), (400-40*{:VA), (200+40*{.VC), (400-40*{:VC)
gllines (200+40*{.VB), (400-40*{:VB), (200+40*{.VC), (400-40*{:VC)

inPAB =: VA inprod VB
gltextcolor ''
glfont '"courier new" 12'
gltextxy 60 40
gltext 'VA = ', '(', ("):VA), '), VB = (', ("):VB), ')'
gltextxy 60 60
gltext 'Innper Product = ', ": inPAB

angAB =: (180%1p1) * (arctan ({: % {.) VA) - (arctan ({: % {.) VB)
gltextxy 60 80
gltext 'Angle of VA and VB = ', (4j1": angAB), ' degree'
)

inprod =: (+/@:(*"(0)))
```