JAPLA 2010/Jul/24　　外積.txt　2010.Jul 15-21

J言語に於ける 内積 と 外積 の演算

APL との 競争

中野 嘉弘（札幌市 87 歳）

FAX 専 011-588-3354

E-mail:　yoshihiro@river.ocn.ne.jp

は し が き

Yahoo 知恵袋・数学カテゴリーは、年寄りのホビーとして最適である。

そこから計算の話題を拾う機会が多い。 最近も、ベクトルの外積の質問があり、

ロートルの回答が BA (Best Answer) を得た。 それをきっかけに、ちょっと

調べたことを報告する。 昔なじみの APL 言語では、ベクトルや行列の「内積」

inner products (普通の行列の積)と「外積」outer products の解説が、

APL の得意分野として丁寧に解説されて居る。 では、我がJ言語では、どうで

あろうか？ 有名な Norman Thomson 氏の名著 " J: The Natural Language for

Analytic Computing " (文献 1) の索引には Inner Products はあるが、Outer

Products の名前は見えない。

我らが J Quick Reference (Ver. 2.0) (文献 2) ではどうだったかな？

兎に角、私の愛用して居る諸例で議論しよう。

それに刺激されて、多少の気付いた事を、報告したい。

1. 内積関数 idot 、多項式の積 polymult 、行列式 det など

● idot =: ip =: +/ .*

```
   ] xlip=. 1 + 0j1  (複素数、j は虚数単位)
1j1
   xlip idot xlip  -> 0j2

   ip ^:(1) 〜 xlip -> 0j2
```

```
  ip ^:(i.5) 〜 xlip ->  1j1 0j2 _2j2 _4 _4j_4

   ] xlim=. 1 - 0j1  （複素数、j は虚数単位）-> 1j_1

  ip ^:(i.5) 〜 xlim -> 1j_1 0j_2 _2j_2 _4 _4j4
```

● polymult=: pm=: +//.@(*/)

```
   ] x3i =. (%: 3) - 0j1 -> 1.73205j_1

    x3i polymult x3i -> 2j_3.4641

    x3i idot x3i -> 2j_3.4641（同じ結果）

    pm ^: (i.9) 〜 x3i ->

   1.73205j_1 ->  1.7j_1 （概略値）
    2j_3.4641 ->    2j_3
_8.88178e_16j_8 ->    0j_8
   _8j_13.8564 ->   _8j_14
  _27.7128j_16 ->  _28j_16
_64j1.42109e_14 ->  _64
   _110.851j64 -> _111j64
  _128j221.703 -> _128j222
1.42109e_13j512 ->    0j512
```

● det =: - / . *   NB. determinant

● diag =: (<0 1)&|: NB. by N. Thomson

● un=: 3 : '=@i.y' NB. unit matrix

● invm=: %.   NB. inverse matrix

● absv=: 3 : '%: +/ y^2' NB. √(a^2 +b^2)

● vcos=: (a idot b)%(absv a)*(absv b) NB. a･b･cos θ

☆ 問 1) 2 次正方行列の A2 の n乗を求めよ。

```
   A2 =. 2 2 $  1 2 3 0
```

普通はケイレイ・ハミルトンの定理を使うので、その先に固有値問題を

解き、対角化し、n 乗し、最後に逆行列で割るなど、面倒くさいな！

結果だけを書けば、

```
 an=. (2*(-2)^n) + 3^n,  bn=. ((_2)^(n+1)) +2*3^n,
```

cn =. (_3*(_2)^n)+3^(n+1), dn=. (3*(_2)^n)+2*3^n として

解1 C =. 2 2 $ (an, bn, cn, dn) % 5 。

J言語では、その手間は無い。n =. 3 の場合には

解1J =. ip ^:(n-1) 〜 A2 -> 2 2 $ 13 14 21 6 である。

☆ 問2） 3次正方行列の A の n乗を求めよ。

A =. 3 3 $ 7 1 2 3 1 6 3 6 1

★ 解1 ）

・n= 2 で、 A ip A -> 58 20 22
42 40 18
42 15 43

・n = 3 で、 ip ^:(2) 〜 A -> 532 210 258
468 190 342
468 315 217

「註」 執筆中に、志村氏の論文「Jの簡易反復法による関数のプログラミング」

(2009/7/30) の pp.8-10 に丁寧な、同巧の解説を発見した。(文献 3)

早速、トライしたが、演算中に、未定義記号 power_matrix や mp 等のエラーで

残念ながら、処理不可能となった。 そこで、

中野の手持ちの関数 ip 等をを集めて、似た処理をした例を示す。

与行列 (Yahoo 知恵袋・数学カテ 2010/7/2 質問、7/5 解決。)

```
 unt3
7 1 2
3 1 6
3 6 1
 det unt3
_200
```

固有値

```
 LF0 unt3
10 _5 4
```

```
 (<=/~i.3),<"2 ip ^:(i.3)~unt3
┌─────┬──────────────────────┐
│1 0 0│7 1 2│58 20 22│532 210 258│
│0 1 0│3 1 6│42 40 18│468 190 342│
│0 0 1│3 6 1│42 15 43│468 315 217│
└─────┴──────────────────────┘
 LF0 each <"2 ip ^:(i.3) ~ unt3
┌──────── ─────────┬────────────┐
```

```
|10 _5 4|100 25 16|1000 _125 64|
└──────┘└──────────────────────┘
```

固有値 10, 5, 4 の n 乗が(答)です。

上記は n= 0, 1, 2, 3 乗の例示です。

内積は主題では無いから、これ位で。

2. 外 積 odot

▲ 記号 A x V : 物理学で多用。

大きさ |A|*|V|* sinθ (両ベクトルの交角)

方向 両ベクトルの張る平面に垂直(右ねじ方向)

行列式表示 | i j k | ・・・ 基本ベクトル

| a b c | ・・・ ベクトル A の 3 成分

| p q r | ・・・ ベクトル V の 3 成分

展開形 i*(b*r - c*q) - j*(a*r - c*p) + k*(a*q - b*p)

◎ sumodot =: + / NB. direct sum

```
 A          B
1 2 3       2  3
4 5 6

 A + / 2  ->  3 4 5
        6 7 8


 A + / B
3 4
4 5
5 6

6 7
7 8
8 9
```

● mulodot=:op =: * / NB. outer product

```
  A * / B
 2  3
 4  6
```

```
 6  9

 8 12
10 15
12 18
```

◆ ベクトル の 外積

```
 A =. 1 2 3 ,   B=.4 5 6 7

 A * / B
4  5  6  7
8 10 12 14
12 15 18 21
```

◎ 文字の外積

```
 A =. 'alpha,beta,gamma,delta'

 B =. ',abc'

 B = / A
0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

   (B = / A),"1 0|: +/"1 B = / A
0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 3
1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 6
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

△ 外積 と グラフの作成 (1) 三角波

北海道情報大学の最初の実習機(IBM)内の APL プログラムでの例を、最近、再発見したので、紹介する。(文献 4) 簡単な話だ。

```
  w =. 4 5 6 1 4 5 6 2 4 5 6 3 4 5 6

  ((4 5 6) = / w) { ' *'

  *   *   *   *
   *   *   *   *
    *   *   *   *
```

▽ 外積 と グラフの作成 (2) 放物線

前例より、少し工夫が要る。(文献 4)

```
  ] r =. |. i. 11
  10 9 8 7 6 5 4 3 2 1 0

  ] x=. 1 + i.11
```

```
  1 2 3 4 5 6 7 8 9 10 11

   ] v =. (x - 5) * (x - 7)
    24 15 8 3 0 _1 3 8 15 24


   (r > / v)

0 0 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0

  (r > / v) { '* 0'
**        **
**        **
***      ***
***      ***
***      ***
***      ***
***      ***
****    ****
****    ****
****    ****
***** *****

  (r > / v) { ' *'
 *******
 *******
  *****
  *****
  *****
  *****
  *****
   ***
   ***
   ***
    *
```

多少の工夫が欲しい問題だ。

３．　む　す　び

兄弟言語 APL と J言語　の接点を探る面白い問題になった。

さらに、発展出来そうな分野である。

文　献

1) Norman Thomson : " J: The Natural Language for Analytic Computing " (2001)

2) Japan APL Assoc.: " J Quick Reference (Ver. 2.0)" 1998/2010

3) 志村正人:「Jの簡易反復法による関数のプログラミング」 JAPLA 2009/7/30 pp.8-10

4) 中野嘉弘:「電算実習システム IBM-MUSIC/SP における コンピュータ言語 APL の運用の研究」 北海道情報大学紀要　第 2 巻第 2 号 平成 3 年 3 月 (1991) pp. 81-114