

行列演算の書式と J プログラム — J からの発想 —

西川 利男

先月例会のとき、志村正人氏から筆者の既発表の報告「カオス・フラクタルと J グラフィックス」の記載事項について質問を受けた。私の説明不足から、誤解を招いたようでもう少し丁寧に説明し、その考え方の由来を述べたい。

1 組の値 (x_0, y_0) からある演算操作を行なって得た次の値 (x_1, y_1) への変化を、ふつう次のような行列表記で表す。

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

これを J の考え方から解釈してみよう。

値 = 演算 値
(名詞) (動詞) (名詞)

ここで行列の各要素部分を演算として強調した筆者独自の書式で書いてみる。

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} a^* & b^* \\ c^* & d^* \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

この書式で、エノンのカオスの式(次ページ参照)を書いてみたのが、以下である。

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} -a^* \text{ square} & 1^* \\ b^* & 0^* \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

ここで、*square* としたのは 2 乗の意味である。

次に J のプログラムだが、この演算操作をそのまま書けるコーディングはないので、多少の工夫として、gerund-evoked gerund の構文を利用する。

```
f `g `:(0) A B
```

例えば、次の簡単な例でやってみる。

```
-: `%: `:(0) 2 3
```

```
1 1.5
```

```
1.41421 1.73205
```

とすると、名詞である 1 組の値に対して、半分(-:)、平方根(%:)のそれぞれの結果を返す。さらに、動詞 dia、により対角要素を取り出し、+ / dia を行う。

```
dia =: i. @#}
```

```
dia -: `%: `:(0) 2 3
```

```
1 1.73205
```

```
+ / dia -: `%: `:(0) 2 3
```

```
2.73205
```

この演算を 1 組の値 (x_0, y_0) に対して、2 回行なえば、行列演算をしたことになる。

なお筆者は J の考え方を使って、「行列(matrix)なる演算操作は動詞であり、一方、値ベクトル、つまり配列(array)は名詞である」ことを強調したい。

以下、「カオス・フラクタルと J グラフィックス」 JAPLA 研究会資料 2007/8/4 から、該当のページを再掲載した。

3. エノンのカオス・フラクタル

エノン(Henon) のカオス・フラクタルでは、IFS はもっと複雑な反復関数系である。

$$\begin{aligned} x_1 &= 1 - a x_0^2 + y_0 \\ y_1 &= b x_0 \end{aligned}$$

ここでは関数を要素とする一種の拡張されたマトリックスの形で表現してみる。

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} -a \text{ square} & 1 \\ b & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

J では、次のようにしてこれに合わせたコーディングが可能である。

```
HA =: 1.5 ] HB =: 0.25
dia =: i.@# NB. extract diagonal terms
henon =: 3 : 0
'x0 y0' =. y.
x1 =. +/ dia (1: - (HA"_ * *:))`]:(0) x0, y0
y1 =. +/ dia (HB"_ * ])`0: `(0) x0, y0
x1, y1
)
```

```
display henon^(i.10000) 1 1
```

