

なぜ、MATLABは使われるのか。 演算子の面から考える

小林 一行

MATLABは、行列演算と数値解析機能をベースとしたプログラミング言語と対話型環境を提供するソフトウェアです。筆者がMATLABに最初に触れたのは、約15年前のVersion4のときでした。その後もMATLABのバージョンアップは続き、現在ではVersion7.13 (R2011b)まで進化しています。

MATLABは、なぜ研究分野から製品開発に渡る広い分野で、長きにわたり使用され続けているのでしょうか？

この特集では、主としてMATLABおよびMATLAB互換ソフトウェアの使い方を中心に解説します。具体的な使い方の解説を始める前に、少し基本に立ち戻り、MATLABおよびMATLAB互換ソフトウェアがなぜ使われるのか、演算子の面から見てどこが便利なのか、どのようにして使うと効果的なのかを、筆者の主観を基に紹介します。

(図1). 行列表記をサポートしているため、行列の四則演算も数式をそのまま書くだけで計算できます(図2)。

行末のセミコロンの有無は、エコーバックをするかしないかの違いです。セミicolonがない場合は結果が表示され、ある場合は結果は表示されず計算だけが行われます。図2の例では、行列Aの末尾には';'があるため内容は表示されず、行列Bだけが表示されています。両方の行列を表示したい場合は、図3のようにセミコロンの代わりにカンマを使います。

これらは行列計算用ソフトウェアとしては標準的な機能であり、何が便利なのかと思うかもしれません。確かにこの程度の機能ならば、MATLABにこだわる必要はないかもしれませんが、MATLABには、ほかの言語にはない便利かつ特有の演算子が幾つか用意されています。

1. 行列を簡単に記述できる ～行列演算子～

MATLABの演算子は、MATLABの名称から由来するように、行列をサポートしていることが特徴の一つでしょう。C言語などのプログラミング言語で行列計算をさせる場合は数十行のプログラムを書く必要がありますが、MATLABでは角カッコで行列を囲み、列の区切りにカンマまたはスペースを、行の区切りにセミicolonを入れるだけで簡単に行列を表現できます

```
>> A=[1,2,3;3,4,5;7,8,9]
A =
     1     2     3
     3     4     5
     7     8     9
```

カンマは行の区切りを、
セミicolonは列の区切りを表す

図1 行列の表現方法

2. 行列を配列として計算できる ～配列演算子～

配列演算子とは、通常の演算子の前にドットが付いた演算子のことです。加算や減算は、行列の場合でも要素ごとの引き算や足し算をするため、.+, .-という演算子は定義されていません。問題は乗算のときです(図4)。MATLABの*は乗算演算子ですが、普通に使った場合は行列の乗算を計算します。行列ではなく配列として扱い、加減算と同じように要素ごとの計算をしたい場合に、配列乗算演算子.*を使います(図5)。

同じ演算を.*演算子を使わないで書くと、図6のようになります。当然ですが、行列演算の乗算とは異なり行列A、Bともに同じ大きさでなければ計算ができません。

行列の大きさは、size関数により取得できます。C言語などではあらかじめ行列の大きさを宣言する必要がありますが、MATLABでは行列Cは行列の大きさを宣言しなくても動的に

```
>> A=[1,2,3;3,4,5;7,8,9];B=[4,2,4;5,6,7;7,8,9]
B =
     4     2     4
     5     6     7
     7     8     9
>>A*B
ans =
    35    38    45
    67    70    85
   131   134   165
```

行の区切りはスペースでもOK

行末にセミicolonがない場合のみ
計算結果を表示する

数式を書くだけで
行列演算ができる

図2 数式を書くだけで行列演算ができる

```
>> A=[1,2,3;3,4,5;7,8,9],B=[4,2,4;5,6,7;7,8,9]
A =
     1     2     3
     3     4     5
     7     8     9
B =
     4     2     4
     5     6     7
     7     8     9
```

AとBの両方の変数の値を
表示したい場合は、カンマで区切る

図3 文の区切りには、セミicolonだけでなくカンマも使える