Excel で行列計算

竹縄　知之

本記事の内容は Matrix2021.xlsm
http://www2.kaiyodai.ac.jp/~takenawa/app-analysis/Matrix2021.xlsm で実行できます.

1. ワークシートで行列計算

Excel には基本的な行列計算の関数が用意されており，ワークシート上で簡単に使うことができる．以下のワークシートは行列 $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ と $B = \begin{pmatrix} 3 & 4 \\ 5 & 6 \end{pmatrix}$ に対して演算を行ったものである．

	A	B	C	D	E	F	G	H	I	J
1										
2	配列数式(行列)の入力									
3	単に数字を入力する．(B4:C5を選択して"={1,2;3,4}"と入力し、Ctrl+Shift+Enterにより決定する方法もある)									
4		A			B					
5		1	2		3	4				
6		3	4		5	6				
7										
8		A+B	B9:C10を選択して"={B5:C6+E5:F6}"と入力し、Ctrl+Shift+Enterにより決定する							
9		4	6							
10		8	10							
11										
12		AB=MMULT			A*B					
13		13	16		3	8				
14		29	36		15	24				
15										
16		A^{-1}=MINVERSE			b	x=A^(-1)b				
17		-2	1		5	-8				
18		1.5	-0.5		2	6.5				
19										

適当な数値を入力する

Excel のシートに行列を入力するのは，基本的には入力したいセルをすべて選択した上で，数式を書き，Ctrl+Shift+Enter で決定する．関数としては例えば，

行列の入力：　={1,2;3,4}　（各セルに数値を直接入力しても可）
行列の和:　　=B5:C6+E5:F6　B5:C6, E5:F6 によって行列 A, B が指定されている．
行列の積：　　=MMULT(B5:C6,E5:F6)
行列の成分ごとの積：　=B5:C6*E5:F6
行列 A の逆行列：　=MINVERSE(B5:C6)
行列 A の転置：　=TRANSPOSE(B5:C6)

などがある．これらを組み合わせて，例えば連立方程式

$$A\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

を

=MMULT(MINVERSE(B5:C6),E17:E18)

により，解くことができる．（ワークシートの F17，F18 のセル）

2. VBA で行列計算

繰り返しを伴うような複雑な計算では，Excel VBA （Visual Basic for Application）上で計算をしてその結果をワークシートに返したくなる場面がよく生じる．しかし，VBA で予め用意されている関数はそれ程多くなく，ユーザーが新しく関数を定義するか，ワークシート関数を使う必要がある. 後者について, 例えばアークサイン Asin を VBA で使うには,

WorksheetFunction.Asin(x)

と，関数名の前に WorksheetFunction.（ピリオドでつなぐ） をつければよい．ワークシート関数には行列に関する関数以外にも統計計算など様々な関数が多数用意されているので，これらが VBA でも使えることは便利である．

また，VBA で行列を扱うには，行列のデータをワークシートと同じ形式の変数として持つ必要があるが, それには, Variant という形式を使えばよい. 実はワークシート関数には, 行列の加法やスカラー倍といった最も基本的な関数がないが（ワークシート上ではそれぞれ“＋”,“＊”を使えばよいだけだが，VBA 上ではこのようには書けない），Variant 形式で変数を持っておけば，これらをユーザーが定義しておくことにより，既存のワークシート関数とユーザー定義関数を混ぜて使うことができる．

以下はサンプルファイル（Matrix2016.xls）の VBA の Microsoft Excel Objects --- Sheet 2 に書いたプログラムである．

- ここで Sub から End Sub までがメインプログラムであり，Function から End Function は行列の和を計算する関数である．
- 変数の型指定は，基本的には

  Dim 変数名 As 形式名

  とするが，関数の引数や出力は宣言部で As を用いて行う．
- このプログラムではシート関数，例えば WorksheetFunction.Transpose(A) とユーザー定義関数 MAdd(A, B) が同時に使われている．

```
Sub Matrix()

    Dim A As Variant, B As Variant, bb As Variant
    Dim At As Variant, Am As Variant
    Dim Det As Double, AplusB As Variant, AB As Variant, AmB As Variant

    'データの読み込み　データ型は Variant の配列
    A = Range("C5:D6").Value    'A は 2×2 の配列
    B = Range("F5:G6").Value    'B は 2×2 の配列
    bb = Range("I5:I6").Value    'bb は 2×1 の配列（ベクトルの演算は 2 次元配列に対するもの）


    '行列の計算
    At = WorksheetFunction.Transpose(A) '行列 A の転置
    Am = WorksheetFunction.MInverse(A)    'Am = A^{-1}
    Det = WorksheetFunction.MDeterm(A)    '行列 A の行列式
    AplusB = MAdd(A, B)                    'A+B
    AB = WorksheetFunction.MMult(A, B)    'AB
    AmB = WorksheetFunction.MMult(Am, B)    'Am B = A^(-1)B
    Ambb = WorksheetFunction.MMult(WorksheetFunction.MInverse(A), B) 'A^(-1)bb

    '計算した行列をシートに出力
    Range("C9:D10").Value = At
    Range("F9:G10").Value = Am
    Range("I9:I9").Value = Det
    Range("C13:D14").Value = AplusB
    Range("F13:G14").Value = AB
    Range("I13:J14").Value = AmB
    Range("L13:L14").Value = Ambb

End Sub

Function MAdd(X As Variant, Y As Variant) As Variant
```

```
    Dim Z As Variant
    m = UBound(X, 1) '行列 X の行の数
    n = UBound(X, 2) '行列 X の列の数
    ReDim Z(1 To m, 1 To n) '配列 Z の大きさを指定
    For i = 1 To m
        For j = 1 To n
            Z(i, j) = X(i, j) + Y(i, j)
        Next j
    Next i
    MAdd = Z
End Function
```

出力結果