الفصل الثالث

Mathematical Operations with Arrays

يوضح هذا الفصل اساسيات استخدام برنامج Matlab لاجراء العمليات الحسابية من جمع ،طرح ،ضرب وقسمة على المصفوفات وبصورة عامة. تعتبر عملية الجمع والطرح من ابسط العمليات الحسابية التي تطبق على المصفوفات بصورها المختلفة. وكذلك تطبق عملية الضرب ،القسمة والاس على المصفوفات بطريقتين مختلفتين: الطريقة الاولى هي استخدام الرموز *، / و ^ في عملية الضرب ، القسمة و الاس على التوالي للمصفوفات وفق قواعد الجبر الخطي. اما الطريقة الثانية فتستخدم الرموز .* ، ./ و .^ التي تسبقها النقطة (.) لضرب، قسمة والرفع للاس على التوالي للمصفوفات عنصر بعنصر والتي تدعى *Element By Element*. كما ويمكن استخدام رمز القسمة (\) في كلا الطريقتين.

٣.١- الجمع والطرح: *Addition and Subtraction*

نستطيع استخدام عملية الجمع (+) والطرح (–) لجمع وطرح المصفوفات بشرط ان تمتلك المصفوفات نفس العدد من الصفوف والاعمدة. وكذلك جمع وطرح المصفوفات مع اي قيمة عددية. بصورة عامة اذا كان لدينا مصفوفتان A و B حجمها (2×3) :

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}$$

ان جمع المصفوفة A مع المصفوفة B ياخذ الشكل الاتي:

$$\begin{bmatrix} (A_{11}+B_{11}) & (A_{12}+B_{12}) & (A_{13}+B_{13}) \\ (A_{21}+B_{21}) & (A_{22}+B_{22}) & (A_{23}+B_{23}) \end{bmatrix}$$

ادناه بعض الامثلة لتوضيح عملية الجمع والطرح:

```
>> VectA=[8 5 4]; VectB=[10 2 7];
```

Define two vectors.

```
>> VectC=VectA+VectB
VectC =
    18     7    11
```

Define a vector VectC that is equal to VectA + VectB.

```
>> A=[5 -3 8; 9 2 10]
A =
     5    -3     8
     9     2    10
>> B=[10 7 4; -11 15 1]
B =
    10     7     4
   -11    15     1
```

Define two 2 × 3 matrices A and B.

```
>> A-B
ans =
    -5   -10     4
    20   -13     9
```

Subtracting matrix B from matrix A.

```
>> C=A+B
C =
    15     4    12
    -2    17    11
```

Define a matrix C that is equal to A + B.

```
>> VectA+A
??? Error using ==> plus
Matrix dimensions must agree.

>>
```

Trying to add arrays of different size.

An error message is displayed.

```
>> VectA=[1 5 8 -10 2]
VectA =
     1     5     8   -10     2
>> VectA+4
ans =
     5     9    12    -6
>> A=[6 21 -15; 0 -4 8]
A =
     6    21   -15
     0    -4     8
>> A-5
ans =
     1    16   -20
    -5    -9     3
```

Define a vector named VectA.

Add the scalar 4 to VectA.

4 is added to each element of VectA.

Define a 2 × 3 matrix A.

Subtract the scalar 5 from A.

5 is subtracted from each element of A.

٣.٢ـ ضرب المصفوفات: *Array Multiplication*

يتم تنفيذ عملية الضرب (*) في برنامج الـMatlab بموجب قواعد الجبر الخطي. فعملية ضرب المصفوفة A مع المصفرفة B يعتمد بالاساس على تساوي عدد الاعمدة في المصفوفة A مع عدد الصفوف في المصفوفة B ويكون ناتج عملية الضرب مصفوفة جديدة عدد صفوفها يساوي عدد صفوف المصفوفة A وعدد اعمدتها يساوي عدد اعمدة المصفوفة B. وكما مبين في المثال الاتي:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix} \text{ and } B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{bmatrix}$$

$$\begin{bmatrix} (A_{11}B_{11}+A_{12}B_{21}+A_{13}B_{31}) & (A_{11}B_{12}+A_{12}B_{22}+A_{13}B_{32}) \\ (A_{21}B_{11}+A_{22}B_{21}+A_{23}B_{31}) & (A_{21}B_{12}+A_{22}B_{22}+A_{23}B_{32}) \\ (A_{31}B_{11}+A_{32}B_{21}+A_{33}B_{31}) & (A_{31}B_{12}+A_{32}B_{22}+A_{33}B_{32}) \\ (A_{41}B_{11}+A_{42}B_{21}+A_{43}B_{31}) & (A_{41}B_{12}+A_{42}B_{22}+A_{43}B_{32}) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 & 3 \\ 2 & 6 & 1 \\ 5 & 2 & 8 \end{bmatrix} \begin{bmatrix} 5 & 4 \\ 1 & 3 \\ 2 & 6 \end{bmatrix} = \begin{bmatrix} (1\cdot5+4\cdot1+3\cdot2) & (1\cdot4+4\cdot3+3\cdot6) \\ (2\cdot5+6\cdot1+1\cdot2) & (2\cdot4+6\cdot3+1\cdot6) \\ (5\cdot5+2\cdot1+8\cdot2) & (5\cdot4+2\cdot3+8\cdot6) \end{bmatrix} = \begin{bmatrix} 15 & 34 \\ 18 & 32 \\ 43 & 74 \end{bmatrix}$$

اما اذا كانت عملية الضرب لمصفوفتين مربعة الحجم فان الناتج ياخذ نفس الحجم. ان عملية الضرب لا تخضع الى عملية التبادل اي ان $\boxed{\boldsymbol{A * B \neq B * A}}$.

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

Define a 4 × 3 matrix A.

Define a 3 × 2 matrix B.

```
>> C=A*B
C =
    28    27
    65    49
    98    32
    84    38
>> D=B*A
??? Error using ==> *
Inner matrix dimensions must agree.

>> F=[1 3; 5 7]
F =
     1     3
     5     7
>> G=[4 2; 1 6]
G =
     4     2
     1     6
>> F*G
ans =
     7    20
    27    52
>> G*F
ans =
    14    26
    31    45
```

Multiply matrix A by matrix B and assign the result to variable C.

Trying to multiply B by A, B*A, gives an error since the number of columns in B is 2 and the number of rows in A is 4.

Define two 2×2 matrices F and G.

Multiply F*G

Multiply G*F

Note that the answer for G*F is not the same as the answer for F*G.

تستطيع ضرب المتجهات بشرط عدد العناصر في المتجه الاول الذي يكون على شكل صف يساوي عدد العناصر في المتجه الثاني والذي يكون على شكل عمود والناتج هو كمية عددية (1×1). اما عند ضرب متجه على شكل عمود بمتجه على شكل صف فالناتج هو مصفوفة من عدد صفوفها صفوف المتجه الاول وعدد اعمدتها اعمدة المتجه الثاني. ادناه امثلة على ضرب المتجهات:

```
>> AV=[2 5 1]
AV =
     2     5     1
>> BV=[3; 1; 4]
BV =
     3
     1
     4
>> AV*BV
ans =
    15
>> BV*AV
ans =
     6    15     3
     2     5     1
     8    20     4
>>
```

Define a three-element row vector AV.

Define a three-element column vector BV.

Multiply AV by BV. The answer is a scalar. (Dot product of two vectors.)

Multiply BV by AV. The answer is a 3×3 matrix.

كذلك يمكن ضرب الكمية العددية بالمصفوفة بحيث يضرب هذا المقدار بكل عنصر من عناصر المصفوفة وكما موضح ادناه:

```
>> A=[2 5 7 0; 10 1 3 4; 6 2 11 5]        Define a 3 × 4 matrix A.
A =
     2     5     7     0
    10     1     3     4
     6     2    11     5
>> b=3                                     Assign the number 3 to the variable b.
b =
     3
```

```
>> b*A                                     Multiply the matrix A by b. This can be
                                           done by either typing b*A or A*b.
ans =
     6    15    21     0
    30     3     9    12
    18     6    33    15
>> C=A*5
C =                                        Multiply the matrix A by 5 and assign
    10    25    35     0                   the result to a new variable C. (Typ-
    50     5    15    20                   ing C = 5*A gives the same result.)
    30    10    55    25
```

٣.٣ قسمة المصفوفة: *Array Division*

تخضع عملية قسمة المصفوفات ايضا لقواعد الجبر الخطي.

المصفوفة الاحادية : *Identity Matrix*

وهي مصفوفة مربعة التي يكون فيها عناصر القطر الرئيسي تساوي 1 وبقية عناصر المصفوفة تساوي 0 والتي يمكن تعريفها بواسطة الامر eye.
عند ضرب المصفوفة أو المتجه بالمصفوفة الاحادية (وفق قواعد الجبر الخطي) فان الناتج المصفوفة نفسها والمتجه نفسه وادناه امثلة على ذلك:

$$AI = IA = A$$

$$\begin{bmatrix} 7 & 3 & 8 \\ 4 & 11 & 5 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 3 & 8 \\ 4 & 11 & 5 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 8 \\ 2 \\ 15 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ 15 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 6 & 2 & 9 \\ 1 & 8 & 3 \\ 7 & 4 & 5 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 2 & 9 \\ 1 & 8 & 3 \\ 7 & 4 & 5 \end{bmatrix}$$

مقلوب المصفوفة: *Inverse of Matrix*

اذا كانت المصفوفة B هي مقلوب المصفوفة A فان ناتج ضربهما يساوي المصفوفة الواحدية ويجب ان تكون المصفوفة مربعة:

$$BA = AB = I$$

مثال على ذلك:

$$\begin{bmatrix} 2 & 1 & 4 \\ 4 & 1 & 8 \\ 2 & -1 & 3 \end{bmatrix}\begin{bmatrix} 5.5 & -3.5 & 2 \\ 2 & -1 & 0 \\ -3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 5.5 & -3.5 & 2 \\ 2 & -1 & 0 \\ -3 & 2 & 1 \end{bmatrix}\begin{bmatrix} 2 & 1 & 4 \\ 4 & 1 & 8 \\ 2 & -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

نستخدم الصيغة A^{-1} أو الامر $inv(A)$ لايجاد مقلوب المصفوفة وادناه امثلة على مقلوب المصفوفة:

```
>> A=[2 1 4; 4 1 8; 2 -1 3]        Creating the matrix A.
A =
     2     1     4
     4     1     8
     2    -1     3
>> B=inv(A)                        Use the inv function to find the
                                   inverse of A and assign it to B.
B =
    5.5000   -3.5000    2.0000
    2.0000   -1.0000         0
   -3.0000    2.0000   -1.0000
>> A*B                             Multiplication of A and B gives the identity matrix.
ans =
     1     0     0
     0     1     0
     0     0     1
```

```
>> A*A^-1                          Use the power -1 to find the inverse of A.
                                   Multiplying it by A gives the identity matrix.
ans =
     1     0     0
     0     1     0
     0     0     1
```

ملاحظة: المصفوفة التي لها مقلوب هي المصفوفة المربعة والتي يكون محددها لا يساوي صفر.

المحدد: *Determinants*

وهي دالة للمصفوفات المربعة، ويمكن ايجاد محدد المصفوفة ذات الحجم (2×2) بهذه الطريقة:

$$|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}, \quad \text{for example,} \quad \begin{vmatrix} 6 & 5 \\ 3 & 9 \end{vmatrix} = 6 \cdot 9 - 5 \cdot 3 = 39$$

ولايجاد محدد اي مصفوفة مربعة نستخدم الامر $det(A)$.

القسمة باتجاه اليسار(\): *Left Division*

يمكن الاستفادة من عملية القسمة باتجاه اليسار Left Division لحل مجموعة من المعادلات الخطية وكما موضح ادناه:

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + A_{13}x_3 &= B_1 \\ A_{21}x_1 + A_{22}x_2 + A_{23}x_3 &= B_2 \\ A_{31}x_1 + A_{32}x_2 + A_{33}x_3 &= B_3 \end{aligned}$$

فيمكن كتابة هذه المعادلات على شكل مصفوفة وكما ياتي:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

أن العلاقة النهائية تكون بالشكل الاتي:

$$AX = B \quad \text{where } A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \text{and } B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}.$$

حيث A مصفوفة ذات بعدين والتي تمثل الطرف الايسر من المعادلات الخطية وهي معاملات جذور، أما B متجه على شكل عمود والذي يمثل الطرف الايمن من المعادلات. يمكن حل هذه العلاقة بضرب طرفيها بمقلوب المصفوفة A من جهة اليسار لكلا الطرفين:

$$A^{-1}AX = A^{-1}B$$

وتكون نتيجة الطرف الايسر كما ياتي:

$$A^{-1}AX = IX = X$$

فالحل النهائي هو:

$$X = A^{-1}B$$

يمكن كتابة العلاقة الاخيرة باستخدام عملية القسمة من جهة اليسار وكما ياتي:

$$X = A \backslash B$$

القسمة باتجاه اليمين (/): *Right Division*

أن عملية القسمة من جهة اليمين تستخدم ايضا لحل المعادلات الخطية التي تاخذ الصيغة الاتية:

$$XC = D.$$

حيث C مصفوفة ذات بعدين والتي تمثل الطرف الايسر من المعادلات الخطية وهي معاملات جذور، أما D متجه على شكل صف والذي يمثل الطرف الايمن من المعادلات. يمكن حل هذه العلاقة بضرب طرفيها بمقلوب المصفوفة C من جهة اليمين لكلا الطرفين:

$$X \cdot CC^{-1} = D \cdot C^{-1}$$

$$X = D \cdot C^{-1}$$

يمكن كتابة العلاقة الاخيرة باستخدام عملية القسمة من جهة اليمين وكما ياتي:

$$X = D/C$$

مثال: استخدم احد العمليات الحسابية للمصفوفات لحل نظام المعادلات الخطية الاتية:

$$\begin{aligned} 4x - 2y + 6z &= 8 \\ 2x + 8y + 2z &= 4 \\ 6x + 10y + 3z &= 0 \end{aligned}$$

الحل:

$$\begin{bmatrix} 4 & -2 & 6 \\ 2 & 8 & 2 \\ 6 & 10 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} 4 & 2 & 6 \\ -2 & 8 & 10 \\ 6 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 8 & 4 & 0 \end{bmatrix}$$

```
>> A=[4 -2 6; 2 8 2; 6 10 3];        Solving the form AX=B.
>> B=[8; 4; 0];
>> X=A\B                             Solving by using left division: X=A\B.
X =
   -1.8049
    0.2927
    2.6341
>> Xb=inv(A)*B                       Solving by using the inverse of A: X = A^-1 B.
Xb =
   -1.8049
    0.2927
    2.6341
```

```
>> C=[4 2 6; -2 8 10; 6 2 3];        Solving the form XC=D.
>> D=[8 4 0];
>> Xc=D/C                            Solving by using right division: X=D/C.
Xc =
   -1.8049    0.2927    2.6341
>> Xd=D*inv(C)                       Solving by using the inverse of C: X = D·C^-1.
Xd =
   -1.8049    0.2927    2.6341
```

٣.٤ عمليات عنصر بعنصر: *Element By Element Operations*

استخدمت الرموز * و / لعمليات ضرب وقسمة المصفوفات التي تخضع لقواعد الجبر الخطي. في كثير من الحالات بعد تمثيل البيانات على شكل مصفوفة نحتاج الى ضرب وقسمة عناصر المصفوفة عنصر بالعنصر المناظر له في المصفوفة الاخرى. لذا فان برنامج الـ Matlab استخدم علامة النقطة التي يمكن اضافتها على يمين عمليات الضرب والقسمة والرفع للاس ايضا (.*، ./، .^) وذلك لضرب وقسمة وللرفع للاس عناصر المصفوفة بالعنصر المناظر له في المصفوفة الاخرى. وتتم هذه العملية للمتجهات والمصفوفات ذات نفس العدد من العناصر. وكما موضح في الجدول ادناه:

Symbol	Description	Symbol	Description
.*	Multiplication	./	Right division
.^	Exponentiation	.\	Left Division

اذا كان لدينا متجهين a و b فان عمليات الضرب والقسمة والرفع للاس عنصر بعنصر يمكن تمثيلها بالشكل الاتي:

$$a\ .*\ b = \begin{bmatrix} a_1 b_1 & a_2 b_2 & a_3 b_3 & a_4 b_4 \end{bmatrix}$$

$$a\ ./\ b = \begin{bmatrix} a_1/b_1 & a_2/b_2 & a_3/b_3 & a_4/b_4 \end{bmatrix}$$

$$a\ .\hat{}\ b = \begin{bmatrix} (a_1)^{b_1} & (a_2)^{b_2} & (a_3)^{b_3} & (a_4)^{b_4} \end{bmatrix}$$

اما اذا كان لدينا مصفوفتين A و B فان عمليات الضرب والقسمة والرفع للاس عنصر بعنصر توضح بالامثلة الاتية:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

$$A\,.\!*\,B = \begin{bmatrix} A_{11}B_{11} & A_{12}B_{12} & A_{13}B_{13} \\ A_{21}B_{21} & A_{22}B_{22} & A_{23}B_{23} \\ A_{31}B_{31} & A_{32}B_{32} & A_{33}B_{33} \end{bmatrix} \qquad A\,./\,B = \begin{bmatrix} A_{11}/B_{11} & A_{12}/B_{12} & A_{13}/B_{13} \\ A_{21}/B_{21} & A_{22}/B_{22} & A_{23}/B_{23} \\ A_{31}/B_{31} & A_{32}/B_{32} & A_{33}/B_{33} \end{bmatrix}$$

$$\mathrm{A}\,.\hat{}\,n = \begin{bmatrix} (A_{11})^n & (A_{12})^n & (A_{13})^n \\ (A_{21})^n & (A_{22})^n & (A_{23})^n \\ (A_{31})^n & (A_{32})^n & (A_{33})^n \end{bmatrix}$$

والامثلة ادناه توضح ذلك:

```
>> A=[2 6 3; 5 8 4]
A =
     2     6     3
     5     8     4
```

Define a 2 × 3 array A.

```
>> B=[1 4 10; 3 2 7]
B =
     1     4    10
     3     2     7
```

Define a 2 × 3 array B.

```
>> A.*B
ans =
     2    24    30
    15    16    28
```

Element-by-element multiplication of array A by B.

```
>> C=A./B
C =
    2.0000    1.5000    0.3000
    1.6667    4.0000    0.5714
```

Element-by-element division of array A by B. The result is assigned to variable C.

```
>> B.^3

ans =
     1    64  1000
    27     8   343
```

Element-by-element exponentiation of array B. The result is an array in which each term is the corresponding term in B raised to the power of 3.

```
>> A*B

??? Error using ==> *
Inner matrix dimensions must agree.
```

Trying to multiply A*B gives an error since A and B cannot be multiplied according to linear algebra rules. (The number of columns in A is not equal to the number of rows in B.)

كما ويمكن استخدام هذه الطريقة لحساب قيمة الدالة لكل قيمة من قيم المتغير الذي يكون على شكل متجه او مصفوفة وكما موضح ادناه:

```
>> x=[1:8]
x =
    1    2    3    4    5    6    7    8
>> y=x.^2-4*x
y =
   -3   -4   -3    0    5   12   21   32
>>
```

Create a vector x with eight elements.

Vector x is used in element-by-element calculations of the elements of vector y.

```
>> z=[1:2:11]
z =
     1     3     5     7     9    11
>> y=(z.^3 + 5*z)./(4*z.^2 - 10)

 y =
   -1.0000    1.6154    1.6667    2.0323    2.4650    2.9241
```

Create a vector z with six elements.

Vector z is used in element-by-element calculations of the elements of vector y.

٣.٥ الدوال المبنية لتحليل المصفوفات: *Built in function For Analyzing Arrays*

يمتلك برنامج الـ Matlab العديد من الوال المبنية لتحليل المصفوفات وكما مبين في الجدول الاتي:

Function	Description	Example
mean(A)	If A is a vector, returns the mean value of the elements of the vector.	>> A=[5 9 2 4]; >> mean(A) ans = 5
C=max(A)	If A is a vector, C is the largest element in A. If A is a matrix, C is a row vector containing the largest element of each column of A.	>> A=[5 9 2 4 11 6 11 1]; >> C=max(A) C = 11
[d,n]=max(A)	If A is a vector, d is the largest element in A, and n is the position of the element (the first if several have the max value).	>> [d,n]=max(A) d = 11 n = 5
min(A)	The same as max(A), but for the smallest element.	>> A=[5 9 2 4]; >> min(A) ans = 2
[d,n]=min(A)	The same as [d,n]= max(A), but for the smallest element.	
sum(A)	If A is a vector, returns the sum of the elements of the vector.	>> A=[5 9 2 4]; >> sum(A) ans = 20

`sort(A)`	If A is a vector, arranges the elements of the vector in ascending order.	`>> A=[5 9 2 4];`
		`>> sort(A)`
		`ans =`
		`2 4 5 9`

`det(A)`	Returns the determinant of a square matrix A.	`>> A=[2 4; 3 5];`
		`>> det(A)`
		`ans =`
		`-2`

`dot(a,b)`	Calculates the scalar (dot) product of two vectors a and b. The vectors can each be row or column vectors.	`>> a=[1 2 3];`
		`>> b=[3 4 5];`
		`>> dot(a,b)`
		`ans =`
		`26`

`cross(a,b)`	Calculates the cross product of two vectors a and b, (axb). The two vectors must have each three elements.	`>> a=[1 3 2];`
		`>> b=[2 4 1];`
		`>> cross(a,b)`
		`ans =`
		`-5 3 -2`

`inv(A)`	Returns the inverse of a square matrix A.	`>> A=[2 -2 1; 3 2 -1; 2 -3 2];`
		`>> inv(A)`
		`ans =`
		`0.2000 0.2000 0`
		`-1.6000 0.4000 1.0000`
		`-2.6000 0.4000 2.0000`

Function	Purpose
length	Length of vector or largest array dimension
ndims	Number of array dimensions
numel	Number of array elements
size	Array dimensions
circshift	Shifts array circularly
diag	Diagonal matrices and diagonals of matrix
fliplr	Flips matrix from left to right
flipud	Flips matrix up to down
reshape	Reshapes array
sort	Sorts array elements in ascending or descending order
transpose	Transpose
max	Largest elements in array
min	Smallest elements in array
sum	Sum of array elements
prod	Product of array elements

امثلة:

دوال الطول، الاعداد العشوائية، الابعاد، عدد العناصر.

```
>> x = [7.1, 3.4, 7.2, 28/4, 3.6, 17, 9.4, 8.9];
>> z = size (x);  % size of x vector
                                    /*تمثل ابعاد المصفوفة*/
>> y = length(x);  % length of x vector
                      /*size تمثل البعد الاكبر من اخراج*/
>> w = ndims(x);  % no of dimensions in array x
                            /*size عدد عناصر اخراج*/
>> v = numel(x);    % no of elements in x
                          /*size ضرب عناصر اخراج*/
>> x

  x =

     7.1000    3.4000    7.2000    7.0000    3.6000
  17.0000    9.4000    8.9000

  >> y

  y =

       8

  >> w

  w =

       2

  >> z

  z =

     1   8
  v =
     8
```

```
>> sum (x)
ans =
   63.6000

>> max (x)
ans =
    17

>> min (x)
ans =
    3.4000

>> sort (x)
ans =
    3.4000    3.6000    7.0000    7.1000    7.2000
8.9000    9.4000   17.0000
```

سؤال واجب/ احسب size، length، numel، ndims للتالي:

```
1. m=[8];
2. w=[5,17];
3. v=[5;9;15];
4. s=[5,7,13;15,8,22];
5. n=ones(3,5);
```

مثال:

```
>> v = [ 23 45 12 9 5 0 19 17];  % horizontal vector
>> w=sort(v)                        % sorting v
>> m = [2 6 4; 5 3 9; 2 0 1]; % two dimensional
array
>> s=sort(m, 1);            % sorting m along the row
>> t=sort(m, 2);          % sorting m along the column
>> v

v =
    23    45    12     9     5     0    19    17

w =
     0     5     9    12    17    19    23    45
>> m

m =
     2     6     4
     5     3     9
     2     0     1
```

```
>> s

s =
     2     0     1
     2     3     4
     5     6     9

>> t

t =
     2     4     6
     3     5     9
     0     1     2
>> k=fliplr(a);

>> l=flipud(a);

>> k

k =

     3     2     1
     6     5     4
     9     8     7

>> l
l =

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

امثلة:

دوال العدد الاكبر والاصغر والمجموع.

```
>> A = rand (4, 6)
A =
 0.1509  0.8537  0.8216  0.3420  0.7271  0.3704
 0.6979  0.5936  0.6449  0.2897  0.3093  0.7027
 0.3784  0.4966  0.8180  0.3412  0.8385  0.5466
 0.8600  0.8998  0.6602  0.5341  0.5681  0.4449

>> [mx, r] = max (A)
mx =
  0.8600 0.8998  0.8216  0.5341  0.8385  0.7027
r =
    4     4     1     4     3     2
```

ملاحظة:

```
>> max (A');
```
(اكبر عنصر لكل سطر)

```
>> [mn, r] = min (A)
mn =
   0.1509  0.4966 0.6449  0.2897  0.3093  0.3704
r =
      1    3    2    2    2    1
```

ملاحظة:

```
>> min (A');
```
(اصغر عنصر لكل سطر)

ملاحظة: اكبر عنصر في مصفوفة ثنائية البعد.

```
>> mmx = max (mx)
mmx =
           0.8998

>> [mmx, i] = max (A (:))
mmx =
            0.8998
i =
     8
```

ملاحظة: توجد طريقة أخرى:

```
>> z = max (max (A));
>> z = min (min (A));
```

ملاحظة: نفس الشيء لحساب المجموع sum.

```
>> z = sum (sum (A));
```

٣.٦ توليد اعداد عشوائية: *Generation of Random Numbers*

يمكن توليد مصفوفة من اعداد عشوائية باستخدام الامر rand والذي يولد اعداد عشوائية محصورة بين الرقم 0 و 1. وكما موضح في الجدول ادناه:

Command	Description	Example

`rand`	Generates a single random number between 0 and 1.	`>> rand`
		`ans =`
		`0.2311`

`rand(1,n)`	Generates an n-element row vector of random numbers between 0 and 1.	`>> a=rand(1,4)`
		`a =`
		`0.6068 0.4860 0.8913 0.7621`

`rand(n)`	Generates an $n \times n$ matrix with random numbers between 0 and 1.	`>> b=rand(3)`
		`b =`
		`0.4565 0.4447 0.9218`
		`0.0185 0.6154 0.7382`
		`0.8214 0.7919 0.1763`

`rand(m,n)`	Generates an $m \times n$ matrix with random numbers between 0 and 1.	`>> c=rand(2,4)`
		`c =`
		`0.4057 0.9169 0.8936 0.3529`
		`0.9355 0.4103 0.0579 0.8132`

`randperm(n)`	Generates a row vector with n elements that are random permutation of integers 1 through n.	`>> randperm(8)`
		`ans =`
		`8 2 7 4 3 6 5 1`