MULTIPROD TOOLBOX
[MATLAB ® evolves becoming ARRAYLAB ☺]

Multiplying two N-D arrays of matrices, vectors or scalars

Paolo de Leva
University Institute of Movement sciences, Rome, IT

SUMMARY

Any kind of multiple products between two block arrays of matrices, vectors or scalars (fig. 1).

DESCRIPTION

MULTIPROD is a powerful generalization for N-D arrays of the MATLAB function MTIMES and the matrix multiplication operator (*). While MTIMES works only with 2-D arrays, MULTIPROD works also with multidimensional arrays.

Besides the element-wise multiplication operator (.*), MATLAB 7 includes only two functions which can perform products between N-D arrays: DOT and CROSS. However, these functions can only perform two kinds of product: the dot product and the cross product, respectively. MULTIPROD can perform any kind of linear algebraic and vectorial products, except for cross products:

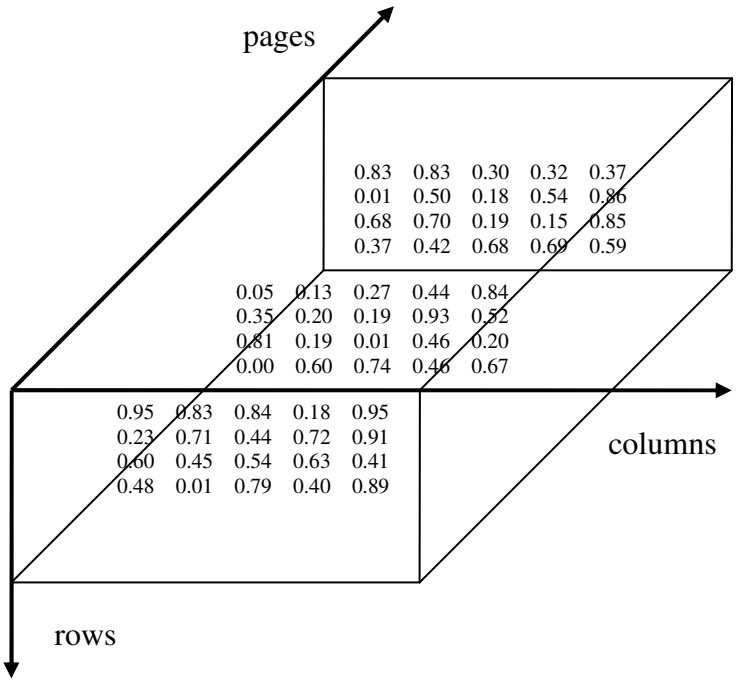

***Figure 1**. A 3-D array, with size **4×5×3**, may be described as a "block array" containing three **4×5** matrices (one per page), or also four **5×3** matrices (one per row). With MULTIPROD, these matrices can be automatically multiplied by matrices, vectors or scalars contained in another array.*

1) Both outer and inner products between arrays of vectors.
2) Arrays of scalars by arrays of scalars, vectors or matrices.
3) Arrays of vectors by arrays of scalars, vectors or matrices.
4) Arrays of matrices by arrays of scalars, vectors or matrices.

Synthetically, it can be stated that, with MULTIPROD, the "*matrix laboratory*" **MATLAB** ® evolves becoming **ARRAYLAB** ☺, an "array laboratory".

Multidimensional arrays may contain matrices or vectors or even scalars along one or two of their dimensions. For instance, a 4-by-5-by-3 array **A** contains three 4-by-5 matrices along its first and second dimension (fig. 1). Thus, array **A** can be described as a *block array* of matrices, and its size can be indicated as follows: (4-by-5)-by-3 or, more simply, (4x5)x3.

Let's say that:

- **A** is (2x5)x6.
- **B** is (5x3)x6.
- **C** is (2x3)x6.

With MULTIPROD, the six matrices in **A** can be multiplied by the six matrices in **B** in a single step:

C = MULTIPROD(**A**, **B**).

Any multidimensional array can be described as a *block array*. Here are a few examples:

- **A** is a 2x5x(6x3) array containing matrices along dimensions 3 and 4.
- **B** is a 2x5x(3x4) array containing matrices along dimensions 3 and 4.
- **C** is a 2x5x(1x1) array containing scalars along dimensions 3 and 4.
- **D** is a 2x5x(6) array containing vectors along dimension 3.
- **E** is a 2x5x(3) array containing vectors along dimension 3.
- **F** is a 2x5x(1) array containing scalars along dimension 3.

For instance, MULTIPROD can:

- Multiply the 10 matrices in **A** by the 10 matrices, vectors or scalars occupying the same position in **B**, **C**, **E**, or **F**.
- Multi-multiply **D** by **A.**
- Multi-multiply the arrays of scalars **C** and **F** by any of the other arrays.
- Perform multiple outer products between **D** and **E**.

MULTIPROD has a broad field of potential applications. By calling MULTIPROD, multiple geometrical transformations such as rotations or roto-translations can be performed on large arrays of vectors in a single step and with no loops. Multiple operations such as normalizing an array of vectors, or finding their projection along the axes indicated by another array of vectors can be performed easily, with no loops and with two or three rows of code.

Sample functions performing some of these tasks by calling MULTIPROD or by using parts of its code are included in the separate toolbox "Matlab Central > File Exchange > **Vector algebra for multidimensional arrays of vectors**".

A sample function (LOC2LOC) performing a multiple roto-translation by calling MULTIPROD is included in this package. This function uses 3-element translation vectors and 3x3 rotation matrices. If you prefer to work with homogeneous coordinates and 4x4 roto-translation matrices, you can obtain the same result just by calling MULTIPROD twice.

If you have a single matrix **R**, and you want to multiply it by an array of vectors **MultiV**, you need to use the function MATEXP to replicate **R** and create an array of matrices **MultiR** which can be multi-multiplied by **MultiV** with MULTIPROD:

MultiV2 = MULTIPROD(**MultiR**, **MultiV, ...**).

MATEXP is included in this package.

MULTIPROD calls, in some cases, the function MULTITRANSP, performing matrix transpositions in arrays of matrices. I wrote it as a separate function because I discovered it can be useful for several different purposes. MULTITRANSP is included in this package.

A document written with MS Word and explaining in detail and schematically the meaning of the implemented syntaxes is included, for those who want to understand the code. The help text in the function was written with extreme care, and should be enough for those who just want to use the function.

Since I wanted to be of service to as many people as possible, MULTIPROD itself was written, debugged, refined and in several parts even optimized for speed with extreme care. I hope it or a future version of it, modified according to your suggestions, will be included in the next version of MATLAB. The code ("testMULTIPROD.m") I used to test the function output is included in this package.

Examples:

```
1) If  A is ..................... a  5x(6x3)x2  array,
   and B is ..................... a  5x(3x4)x2  array,
   C = MULTIPROD(A, B, [2 3]) is a  5x(6x4)x2  array.

2) If  A is ..................... a  5x(1x3)x2  array,
   and B is ..................... a  5x(3x1)x2  array,
   C = MULTIPROD(A, B, [2 3]) is a  5x(1x1)x2  array, while
   C = MULTIPROD(B, A, [2 3]) is a  5x(3x3)x2  array.

3) If  A is ..................... a  5x(6x3)x2  array,
   and B is ..................... a  5x(1x1)x2  array,
   C = MULTIPROD(A, B, [2 3]) is a  5x(6x3)x2  array.

4) If  A is ........................... a  5x(6x3)x2  4-D array,
   and B is ........................... a  5x(3)x2    3-D array,
   C = MULTIPROD(A, B, [2 3], [2]) is a  5x(6)x2    3-D array.

5) If  A is ........................... a  5x(6x3)x2  4-D array,
   and B is ........................... a  5x(1)x2    3-D array,
   C = MULTIPROD(A, B, [2 3], [2]) is a  5x(6x3)x2  4-D array.

6) If both A and B are ..................  5x(6)x2    3-D arrays,
   C = MULTIPROD(A, B, [2 0], [0 2]) is a  5x(6x6)x2  4-D array, while
   C = MULTIPROD(A, B, 2) is .......... a  5x(1)x2    3-D array.
```

MULTIPROD

Multiplying 1-D or 2-D subarrays contained in two N-D arrays

HELP TEXT

C = MULTIPROD(**A**,**B**) is equivalent to **C** = MULTIPROD(**A**,**B**,[1 2],[1 2])
C = MULTIPROD(**A**,**B**,[D1 D2]) is eq. to **C** = MULTIPROD(**A**,**B**,[D1 D2],[D1 D2])
C = MULTIPROD(**A**,**B**,D1) is equival. to **C** = MULTIPROD(**A**,**B**,D1,D1)

1) MATRICES BY MATRICES (2-D by 2-D arrays) (*).

If SIZE(**A**) == 2 and SIZE(**B**) == 2,
C = MULTIPROD(**A**, **B**, [1 2], [1 2]) is equivalent to **C** = **A** * **B**.
When both **A** and **B** have N dimensions ("), with N >= 2,
C = MULTIPROD(**A**, **B**, [D1 D2], [D1 D2]) has N dimensions and contains the products of the matrices in **A** by those in **B**:
A (N-D) contains P-by-Q matrices along dimensions D1 and D2,
B (N-D) contains R-by-S matrices along dimensions D1 and D2,
C (N-D) contains P-by-S matrices along dimensions D1 and D2.
Vector inner (1b) or outer (1c) products are performed if P=S=1 or Q=R=1, and products involving scalars (1d) if P=Q=1 or R=S=1 or P=Q=R=S=1. In the latter case, **C** = **A** .* **B**. (°)

2) MATRICES BY 1-D ARRAYS (2-D by 1-D arrays) (*).

When **A** has N + 1 ("), and **B** has N dimensions ('), with N >= 1,
C = MULTIPROD(**A**, **B**, [D1 D2], D1) contains the products of the matrices contained in **A** by the 1-D arrays contained in **B** (°):

Array	Dimensions	Size of subarray		Contained along
A	N + 1	P-by-Q	(2-D)	dimensions D1 & D2
B	N	R	(1-D)	dimension D1
C (a)	N	P	(1-D)	dimension D1
C (b)	N + 1	P-by-Q	(2-D)	dimensions D1 & D2

(a) If the subarrays in **B** are not scalars (size: R > 1), **C** has N dimensions. In this case, the 1-D subarrays in **B** and **C** are meant to represent column matrices, and Q = R is required.
(b) If the subarrays in **B** are scalars (size: R = 1), **C** has N + 1 dimensions, and Q = R is not required.

3) 1-D ARRAYS BY MATRICES (1-D by 2-D arrays). (*)

When **A** has N ('), and **B** has N + 1 dimensions ("), with N >= 1:
C = MULTIPROD(**A**, **B**, D1, [D1 D2]) contains the products of the 1-D arrays contained in **A** by the matrices contained in **B** (°):

Array	Dimensions	Size of subarray		Contained along
A	N	Q	(1-D)	dimension D1
B	N + 1	R-by-S	(2-D)	dimensions D1 & D2
C (a)	N	S	(1-D)	dimension D1
C (b)	N + 1	R-by-S	(2-D)	dimensions D1 & D2

(a) If the subarrays in **A** are not scalars (size: Q > 1), **C** has N dimensions. In this case, the 1-D subarrays in **A** and **C** are

meant to represent row matrices, and Q = R is required.
(b) If the subarrays in **A** are scalars (size: Q = 1), **C** has N + 1 dimensions, and Q = R is not required.

4) 1-D ARRAYS BY 1-D ARRAYS. (*)

Without limitations on the length of **A** and **B** along dimension D1:
a) **C** = MULTIPROD(**A**, **B**, [D1 0], [0 D1]) returns outer products along dimension D1 of **A** and **B**, and is equivalent to **C** = OUTER(**A**, **B**, D1). If **A** and **B** have N dimensions ('), **C** has N + 1 dimensions ("). See function OUTER for details. (°)

When **A** and **B** have the same size and any length along dimension D1:
b) **C** = MULTIPROD(**A**, **B**, [0 D1], [D1 0]) returns inner (dot) products along dimension D1 of **A** and **B**, and is equivalent to **C** = DOT(**A**, **B**, D1). See function DOT for details. (°)
c) **C** = MULTIPROD(**A**, **B**, D1, D1) is equivalent to the above syntax (4b).

When either **A** or **B** has (or both have) length 1 along dimension D1:
d) **C** = MULTIPROD(**A**, **B**, D1, D1) returns products of scalars by N-element vectors (N > 1), or N-element vectors by scalars or scalars by scalars along dimension D1 of **A** and **B**. (°)

When **A** and **B** have the same size, and length 3 along dimension D1:
e) **C** = CROSS(**A**, **B**, D1) can be used to perform cross products along dimension D1 of **A** and **B**. See function CROSS for details.

MULTIPROD is a generalization for N-D arrays of MTIMES and the matrix multiplication operator ("").*

Notes:

(*) N-D arrays can be called SCALARS if all their dimensions have length 1 (1-by-1-by-1), or VECTORS if they have at most one non-singleton dimension (1-by-1-by-P), or MATRICES if N > 1 and they have at most two non-singleton dimensions (1-by-P-by-Q). Some matrices are also vectors and/or scalars, and some vectors are scalars. E.g., 1-by-1 arrays are scalars, vectors and matrices.

(") Including trailing singletons up to dimension D2, if they exist.

(') Including trailing singletons up to dimension D1, if they exist, and excluding the second dimension in P-by-1 arrays, if D1 = 1. For instance, with D1 = 1, P-by-1 arrays (column matrices) are treated as 1-D, with D1 = 2 as 2-D, ...

(°) COMMON CONSTRAINTS FOR ALL SYNTAXES (EXCEPT FOR 4e)
Products involving 2-D and/or 1-D scalars (1-by-1 matrices or 1-element 1-D arrays) are allowed. Scalars may multiply 1-D or 2-D arrays of any size, except for syntax 4b (in this case, **A** and **B** must have the same size). When scalars are not involved, the "inner dimensions" must have the same length (Q = R for syntaxes 1, 2, 3; for 4a, 4b, 4c see functions DOT and OUTER). D1 is allowed to be larger than NDIMS(**A**) and/or NDIMS(**B**). **A** and **B** must have the same lengths along all dimensions except for those containing the specified matrices (D1 and D2) or vectors (D1), unless otherwise specified (syntax 4b). D2 = D1 + 1 is required.

Examples:

A 5-by-6-by-3-by-2 array may be considered to be a block array containing ten 6-by-3 matrices along dimensions 2 and 3. In this case, its size is so indicated: 5-by-(6-by-3)-by-2 or 5x(6x3)x2.

1a) If **A** is ..... a 5x(6x3)x2 array,
and **B** is ..... a 5x(3x4)x2 array,
C = MULTIPROD(**A**, **B**, [2 3]) is a 5x(6x4)x2 array.

1b) If **A** is ..... a 5x(1x3)x2 array,
and **B** is ..... a 5x(3x1)x2 array,
C = MULTIPROD(**A**, **B**, [2 3]) is a 5x(1x1)x2 array, while
1c) **C** = MULTIPROD(**B**, **A**, [2 3]) is a 5x(3x3)x2 array.

1d) If **A** is ..... a 5x(6x3)x2 array,
and **B** is ..... a 5x(1x1)x2 array,
C = MULTIPROD(**A**, **B**, [2 3]) is a 5x(6x3)x2 array.

2a) If **A** is ..... a 5x(6x3)x2 4-D array,
and **B** is ..... a 5x(3)x2 3-D array,
C = MULTIPROD(**A**, **B**, [2 3], [2]) is a 5x(6)x2 3-D array.

2b) If **A** is ..... a 5x(6x3)x2 4-D array,
and **B** is ..... a 5x(1)x2 3-D array,
C = MULTIPROD(**A**, **B**, [2 3], [2]) is a 5x(6x3)x2 4-D array.

4a) If both **A** and **B** are ..... 5x(6)x2 3-D arrays,
C = MULTIPROD(**A**, **B**, [2 0], [0 2]) is a 5x(6x6)x2 4-D array, while
4c) **C** = MULTIPROD(**A**, **B**, 2) is ..... a 5x(1)x2 3-D array.

See also **DOT, OUTER, CROSS, CROSSDIV, MULTITRANSP**.

$ Version: 1.1 $

CODE	by:	Paolo de Leva (IUSM, Rome, IT)	2005 Sep 15
	speed-optimized by:	Code author	2005 Sep 27
COMMENTS	by:	Code author	2005 Sep 15
OUTPUT	tested by:	Code author	2005 Sep 27