

Multiplicaciones básicas tensoriales implementadas en Matlab

Lucio G. Cruz

Octubre, 2008

Índice

1. Introducción	1
2. Convenciones utilizadas	2
3. Lista de Funciones	3

1. Introducción

Este breve documento describe varias funciones definidas en el lenguaje de Matlab, creadas para obtener multiplicaciones básicas tensoriales.

La lista de funciones tensoriales creadas y anexas al fin del documento, son:

Tipo de Producto	N. Tensorial	N. Indicial	Resultado (Ord. tensor)
Producto Diádico	$\mathbf{A} \otimes \mathbf{B}$	$A_{ij}B_{kl}$	$\mathbf{C} = C_{ijkl}$ (4^{to} orden)
Contracción simple	$\mathbf{A} \cdot \mathbf{B}$	$A_{ik}B_{kj}$	$\mathbf{C} = C_{ij}$ (2^{do} orden)
Contracción doble	$\mathbf{A} : \mathbf{B}$	$A_{ij}B_{ij}$	C (Orden cero, escalar)
Contracción doble	$\mathbf{A} : \mathbf{B}$	$A_{ijkl}B_{kl}$	$\mathbf{C} = C_{ij}$ (2^{do} orden)
Contracción doble	$\mathbf{A} : \mathbf{B}$	$A_{ijmn}B_{mnkl}$	$\mathbf{C} = C_{ijkl}$ (4^{to} orden)
Contracción cuádruple	$\mathbf{A} :: \mathbf{B}$	$A_{ijkl}B_{ijkl}$	C (Orden cero, escalar)

La intención es que las funciones creadas sirvan en implementaciones numéricas hechas en Matlab, y así mismo sean mejoradas en el proceso de ser utilizadas.

Es importante notar que estas funciones no tienen transformación de bases, ni simplificaciones, por lo cual un tensor de segundo orden es representado como una matriz de 3x3 con nueve componentes, donde cada posición esta representada por 2 sub-índices. Un tensor de cuarto orden tendrá 81 componentes y cada posición estará representada por cuatro sub-índices.

Cualquier mejora o otra función puede ser enviada a lucruz@unicauca.edu.co, con lo cual contribuiría a la mejora de la biblioteca de funciones.

2. Convenciones utilizadas

Las siguientes son las convenciones utilizadas en los nombres de las funciones y en su creación:

EC : Tensor C de orden cero (escalar), (Valor de retorno en las funciones)
 TA2O: Tensor A de segundo orden, (Argumento en las funciones)
 TB2O: Tensor B de segundo orden, (Argumento en las funciones)
 TA4O: Tensor A de cuarto orden, (Argumento en las funciones)
 TB4O: Tensor B de cuarto orden, (Argumento en las funciones)
 TC2O: Tensor C de segundo orden, (Valor de retorno en las funciones)
 TC4O: Tensor C de cuarto orden, (Valor de retorno en las funciones)

Como es bien conocido la estructura de una función es:

$$[Valor\ de\ Retorno]=nombre\ de\ la\ función(Argumentos)$$

De acuerdo a esto la estructura de las funciones sera la siguiente:

$$[TIPRETORNO]=descripción_TIPARG1_TIPARG2(ARG1,ARG2)$$

donde:

[TIPRETORNO]: Es el tipo de retorno de acuerdo a las convenciones establecidas.

descripción_TIPARG1_TIPARG2: Nombre de la función que describe el producto que realiza, indicando los dos tipos de argumentos que se le deberán

pasar a esta, de acuerdo a las convenciones establecidas.

(*ARG1, ARG2*): Argumentos que deberán ser del orden y tipo que la función requiere.

En la siguiente sección se crea la lista de funciones, de acuerdo a las convenciones establecidas.

3. Lista de Funciones

La siguiente es la lista de funciones y su estructura:

Tipo de Producto	N. Indicial	Estructura de la función
Producto Diádico	$A_{ij}B_{kl}$	[TC4O]=prodiadico_TA2O_TB2O(A,B)
Contracción simple	$A_{ik}B_{kj}$	[TC2O]=contracsimple_TA2O_TB2O(A,B)
Contracción doble	$A_{ij}B_{ij}$	[EC]=contracdoble_TA2O_TB2O(A,B)
Contracción doble	$A_{ijkl}B_{kl}$	[TC2O]=contracdoble_TA4O_TB2O(A,B)
Contracción doble	$A_{ijmn}B_{mnkl}$	[TC4O]=contracdoble_TA4O_TB4O(A,B)
Contracción cuádruple	$A_{ijkl}B_{ijkl}$	[EC]=contracquadруп_TA4O_TB4O(A,B)

Para llamar la funciones desde un programa de Matlab, y que el resultado se almacene en una variable C, utilizaremos la siguiente lista:

Tipo de Producto	N. Indicial	Forma de llamar la función
Producto Diádico	$A_{ij}B_{kl}$	C=prodiadico_TA2O_TB2O(A,B)
Contracción simple	$A_{ik}B_{kj}$	C=contracsimple_TA2O_TB2O(A,B)
Contracción doble	$A_{ij}B_{ij}$	C=contracdoble_TA2O_TB2O(A,B)
Contracción doble	$A_{ijkl}B_{kl}$	C=contracdoble_TA4O_TB2O(A,B)
Contracción doble	$A_{ijmn}B_{mnkl}$	C=contracdoble_TA4O_TB4O(A,B)
Contracción cuádruple	$A_{ijkl}B_{ijkl}$	C=contracquadруп_TA4O_TB4O(A,B)

Al final de este documento se encuentran anexas las funciones descritas.

Referencias

- [1] García J., Rodríguez J y Vidal J., *Aprenda Matlab 7.0 como si estuviera en primero*, Univ. Politécnica de Madrid, www.tayuda.com, Madrid, 2005.

```
function [TC40]=prodiadico_TA20_TB20(A,B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Filtro de Argumentos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m,n]=size(A);
[o,p]=size(B);
if m~=n || o~=p
    error('Alguno de los tensores de orden 2 no es cuadrado')
    return
end
if m~=3 || o~=3
    error('Alguno de los tensores argumento no es de segundo orden')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:3
    for j=1:3
        for k=1:3
            for l=1:3
                TC40(i,j,k,l)=A(i,j)*B(k,l);
            end
        end
    end
end
end
```

```
function [TC2O]=contracsimple_TA2O_TB2O(A,B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Filtro de Argumentos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m,n]=size(A);
[o,p]=size(B);
if m~=n || o~=p
    error('Alguno de los tensores de orden 2 no es cuadrado')
    return
end
if m~=3 || o~=3
    error('Alguno de los tensores argumento no es de segundo orden')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TC2O=zeros(3); %Inicializamos tensor de segundo orden
for i=1:3
    for j=1:3
        for k=1:3
            TC2O(i,j)=TC2O(i,j)+A(i,k)*B(k,j);
        end
    end
end
end
```

```
function [EC]=contracdoble_TA2O_TB2O(A,B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Filtro de Argumentos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m,n]=size(A);
[o,p]=size(B);
if m~=n || o~=p
    error('Alguno de los tensores de orden 2 no es cuadrado')
    return
end
if m~=3 || o~=3
    error('Alguno de los tensores argumento no es de segundo orden')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
EC=0; %Inicializamos el escalar C
for i=1:3
    for j=1:3
        EC=EC+A(i,j)*B(i,j);
    end
end
end
```

```
function [TC20]=contracdoble_TA40_TB20(A,B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Filtro de Argumentos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m,n,o,p]=size(A);
[q,r]=size(B);
if m~=n || m~=o || m~=p
    error('El tensor de orden 4 no es cuadrado')
    return
end
if q~=r
    error('El tensor de orden 2 no es cuadrado')
    return
end
if m~=3
    error('El tensor argumento A no es de cuarto orden')
    return
end
if q~=3
    error('El tensor argumento B no es de segundo orden')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TC20=zeros(3); %Inicializamos el tensor C de segundo orden
for i=1:3
    for j=1:3
        for k=1:3
            for l=1:3
                TC20(i,j)=TC20(i,j)+A(i,j,k,l)+B(k,l);
            end
        end
    end
end
end
```

```
function [TC40]=contracdoble_TA40_TB40(A,B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Filtro de Argumentos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m,n,o,p]=size(A);
[q,r,s,t]=size(B);
if m~=n || m~=o || m~=p
    error('El tensor A de orden 4 no es cuadrado')
    return
end
if q~=r || q~=s || q~=t
    error('El tensor B de orden 4 no es cuadrado')
    return
end
if m~=3
    error('El tensor argumento A no es de cuarto orden')
    return
end
if q~=3
    error('El tensor argumento B no es de cuarto orden')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Inicializamos el tensor C de cuarto orden
TC40(:,:,1,1)=zeros(3);
TC40(:,:,1,2)=zeros(3);
TC40(:,:,1,3)=zeros(3);
TC40(:,:,2,1)=zeros(3);
TC40(:,:,2,2)=zeros(3);
TC40(:,:,2,3)=zeros(3);
TC40(:,:,3,1)=zeros(3);
TC40(:,:,3,2)=zeros(3);
TC40(:,:,3,3)=zeros(3);
for i=1:3
    for j=1:3
        for k=1:3
            for l=1:3
                for m=1:3
                    for n=1:3
                        TC40(i,j,k,l)=TC40(i,j,k,l)+A(i,j,m,n)+B(m,n,k,l);
                    end
                end
            end
        end
    end
end
end
end
```



```
function [EC]=contracquadrup_TA40_TB40(A,B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Filtro de Argumentos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m,n,o,p]=size(A);
[q,r,s,t]=size(B);
if m~=n || m~=o || m~=p
    error('El tensor A de orden 4 no es cuadrado')
    return
end
if q~=r || q~=s || q~=t
    error('El tensor B de orden 4 no es cuadrado')
    return
end
if m~=3
    error('El tensor argumento A no es de cuarto orden')
    return
end
if q~=3
    error('El tensor argumento B no es de cuarto orden')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
EC=0; %Inicializamos el escalar C
for i=1:3
    for j=1:3
        for k=1:3
            for l=1:3
                EC=A(i,j,k,l)+B(i,j,k,l);
            end
        end
    end
end
end
```