Subject: general matrix multiplication
Posted by David Schmidt on Mon, 23 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

All,

I'm looking for a routine to perform generalized matrix multipication over particular indexes within arrays. For example, let A be a (10,3,20) array and B be a (20,5,10,30) array. I want to be able to multiply and add (i.e. matrix multiply) the elemens within index 1,3 of A and 3,1 of B and produce a result of dimension (3,5,10). While this can be done simply by using FOR loops, I'm looking for a routine that does this efficiently, using built-in IDL routines. Does anyone know of such a routine? ...how I could construct such a routine?

Thanks,

David
--

// /
/ David Schmidt /
/ P-21, MS-D454 Voice: (505) 665-3584 /
/ Biophysics Group Fax: (505) 665-4507 /
/ Los Alamos National Laboratory Email: dschmidt@lanl.gov /
/ Los Alamos, NM 87545 /
// /

Subject: Re: general matrix multiplication
Posted by David Schmidt on Fri, 06 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
>
>
> hmmm - I am not sure I understand how you will get a result with the
> dimensions you indicate. Anyway, here some hints that will hopefully
> help you a little:
> While matrix multiplication is indeed provided by IDL with the
> ## operator (see online help:
> "The ## operator does what is commonly referred to as matrix
> multiplication. It computes array elements by multiplying the rows of
> the first array by the columns of the second array. The second array
> must have the same number of rows as the first array has columns. The
> resulting array has the same number of rows as the first array and the
> same number of columns as the second array."
>

```
>  I don't think you will have a chance to get around the loops over
>  3, 5, and 30. Your piece of code should be something like
>      result = fltarr(10,10,3,5,30)
>      for i=0,2 do begin
>          for j=0,4 do begin
>              for k=0,29 do begin
>                  tmp = A(*,i,*) ## B(*,j,*,k)
>                  result(*,*,i,j,k) = tmp
>              endfor
>          endfor
>      endfor
>
>  Regards,
>  Martin.
>
```

Martin,

Thanks for your note.

I think there is a way of getting around the FOR loops.
If we can rearrange the indicies so that A(10,3,20) becomes A1(3,10,20)
and B(20,5,10,30) becomes B1(10,20,5,30), then we can use the REFORM
function to make 2D matrices:
A2=REFORM(A1,3,200), B2=REFORM(REFORM(B1,200,5,30),200,150).
Then we can use the # operater: C=A2#B2 which will produce a (3,150)
array and then the REFORM function again to produce a (3,5,30) array.
This algorithm will be MUCH faster than the FOR loop version, especially
as the number of indices grows.

The key, though, is being able to rearrange the order of indicies, or to
have a more general form of REFORM that allows one to specify which
indices are to be combined and in what order. I don't know of a way to
do this using built-in IDL functions. Does anyone?
Again, I'd like to use built-in functions to save on CPU speed.

David
--


```
  ////////////////////////////////////////////////////////// /
/ David Schmidt                                            /
/ P-21, MS-D454                  Voice: (505) 665-3584     /
/ Biophysics Group                 Fax: (505) 665-4507     /
/ Los Alamos National Laboratory   Email: dschmidt@lanl.gov /
/ Los Alamos, NM 87545                                     /
 ////////////////////////////////////////////////////////// /
```