Subject: Re: HDF SDS array access in IDL
Posted by thompson on Thu, 29 Oct 1998 08:00:00 GMT
View Forum Message <> Reply to Message

"Dr. G. Scott Lett" <slett@holisticmath.com> writes:

> Ok, ok.  Let me try this a different way.

> Standard linear algebra packages written in FORTRAN, such as Linpack,
> followed the _convention_ that general matrices were stored as two
> dimensional arrays, accessed as (row, column).  Because there were no matrix
> operations built into old FORTRAN, this was just a convention.

> Fortran now has a number of matrix operations built into the standard
> language.  An example is the MATMUL intrinsic function, which follows the
> (row, column) convention.  So, now the convention is part of the standard.
> You can multiply an array A of shape (3,4) by an array B of shape (4,3).
> MATMUL(A,B) returns an array of shape (3,3).

> IDL has a number of matrix operations and linear algebra functions.  They
> follow the convention of storing matrices as two dimensional arrays in
> (column, row) order.  The matrix multiplication operator in IDL can multiply
> an array A of size
> [3,4] and an array B of size [4,3].  A ## B returns an array of size [4,4].

Ah, but there are two different kinds of matrix operations in IDL.  You can also use A # B to return an array of size [3,3].  If you use the # operator, then IDL behaves as having matrices in (row,column) order, and the rows and columns are stored as in Fortran.  If you use the ## operator, then it behaves as having matrices in (column,row) order, and the rows and columns are stored as in C.

I don't believe the ## operator was even introduced until IDL/v4.  An old IDL/v3 manual that I have laying around states that # is the matrix multiplication operator, and there is no mention of ## as an operator.  I suspect that ## was added to IDL to make it more C-like.  Originally, IDL was written to more closely emulate Fortran.

All the IDL procedures in use here that I'm aware of uses the # operator, and thus follow a (row,column) convention.  However, since those matrices are used internally, there's no confusion to anyone who prefers to use the ## operator instead.

Actually, even if one does use ## instead of #, isn't there still a difference in the way IDL indexes elements of a matrix?  If you follow the ## convention, then MATRIX(3,5) would be the third column, fifth row.  However, in C, wouldn't you write this MATRIX[5][3]?

> This whole question can be academic, or at most cosmetic, unless one does
> things such as linking Fortran linear algebra codes into IDL.  ...

It's also vitally important if one is passing data arrays back and forth between IDL and Fortran or C routines.  If one has a 1000x300 array in IDL, it's also a 1000x300 array in Fortran, but one had better treat it as a 300x1000 array in C.

William Thompson