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

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].

This whole question can be academic, or at most cosmetic, unless one does things such as linking Fortran linear algebra codes into IDL. The difference in storage/access conventions can also have a profound impact on the efficiency of certain algorithms, as David Fanning pointed out. Is an LU decomposition of a matrix faster when the matrix is stored by rows or by columns? Try it and see.