Yes, Fortran does have array operations, but the concepts of 'row' and 'column' don't really apply. More to the point, Fortran has a _matrix_ operation, MATMUL, that assumes that an array is represented by a two dimensional array in column-major order. For Fortran, this means that a matrix is accessed as (row, column). Most mathematics texts and papers index matrices with subscripts in the same order.

The old standard FORTRAN's didn't have array or matrix operations, so it was up to the people designing FORTRAN matrix/linear algebra software to decide if they preferred row-major or column-major. According to the LINPACK Users' Guide, the designers of LINPACK chose column major order (they called it 'column oriented') to improve performance on 'operating systems with paging and cache memory.' The authors point out that the BLAS, upon which LINPACK was based, allow matrices to be accessed by rows instead of columns, so that a matrix could be represented by a two dimensional array, accessed as (column, row), but LINPACK works only one way. EISPACK followed the same convention as LINPACK, and most similar libraries followed this convention.

IDL is not quite Fortran, not quite C.

Flame on, brothers and sisters.

Axel vom Endt <endt@bu.edu> wrote in message
news:36DE990B.5136E2EF@bu.edu...
> Steve Scheele wrote:
>
>> Fortran is column-major, and IDL was originally written in Fortran. I assume
>> that this is the reason that IDL is column-major. However, since Fortran has
>> no array arithmetic, it seems somewhat irrelevant at this point.
>
> Now that's a statement I cannot resist to comment on. Fortran *does*
> have array arithmetic, along with a number of nice features C is
> lacking. It does have it for nine years now, time to make the news
> heard.
>
> You are not talking about this old thing from 1977 called FORTRAN (all
> caps), are you? :-)
>

> Axel