Subject: Re: Interesting Rant
Posted by Earl F. Glynn on Wed, 15 Nov 2006 17:19:58 GMT
View Forum Message <> Reply to Message

"Richard Edgar" <rge21@pas.rochester.edu> wrote in message
news:ejd4g4$416$1@mail.rochester.edu...
> Paul van Delst wrote:
>
>>> Someone sent me a link to this interesting IDL rant this morning:
>>>
>>>   http://www.sccs.swarthmore.edu/users/07/sstvinc2/research/st upid.html

> Row-major vs column major is a silly point... you just need to know
> which way the language does it, and that's the end of the matter. And
> I'd dispute the bit about 'every other language in the history of
> mankind' too ;-)

Perhaps a bit exaggerated above, but I don't think this is a silly point at
all, especially when one must switch between programming languages to
maintain software. Perhaps, I'm wrong, but IDL's arrays seem to be unique,
and unlike C or FORTRAN.

C: row-major format, [nrows, ncolumns], 0-origin
"Elements are stored in rows, that is, the rightmost subscript varies
fastest as elements are accessed in storage order." Kernighan & Ritchie,
The C Programming Language.

```
X[0][0]  X[0][1]  X[0][2]
X[1][0]  X[1][1]  X[1][2]
...
```

The data are stored in memory from left-to-right across a row, with rows
ordered from top to bottom.

FORTRAN or R: column-major format [nrows, ncolumns], 1-origin

```
X[1,1]  X[1,2]  X[1,3]
X[2,1]  X[2,2]  X[2,3]
...
```

The data are stored in memory from top-to-bottom across a column, with
columns ordered from left to right.

But IDL doesn't follow EITHER of these models completely and is strangely
unique:
IDL: column-major format, [ncolumns, nrows], 0-origin

IDL Array Storage and Indexing

http://www.dfanning.com/misc_tips/colrow_major.html

"Arrays in IDL are stored in row order which means the first index varies the fastest. Therefore, whenever you write any kind of loop that accesses an array try to vary the first element the fastest." P 1-11, Ronn Kling, "Application Development in IDL"

X[0,0] X[1,0] X[2,0]
X[0,1] X[1,1] X[2,1]
...

The data are stored in memory from left-to-right across a row, with rows ordered from top to bottom. SO, IDL is just like C if you reverse the order of the subscripts. Forget about that reversal if you're thinking in C, and there's a bug in your code.

But when dealing with images, IDL arrays really go from bottom to top?

...
X[0,1] X[1,1] X[2,1]
X[0,0] X[1,0] X[2,0]

Or did I miss something about how IDL stores arrays in memory?

The lack of a standard convention here could easily contribute to programming errors, especially if one uses IDL and other programming languages. Converting code from other languages could also be problematic because of silly mistakes getting subscripts right.

This may not be a problem if IDL is the only language one uses, but if one works with a variety of programming languages this uniqueness is not desirable (at least to me).

efg

Earl F. Glynn
Scientific Programmer
Stowers Institute for Medical Research