Subject: Re: IDL: are dynamic arrays supported?
Posted by davidf on Wed, 11 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Darran Edmundson (dEdmundson@Bigfoot.com) writes:

```
>  I am a new user of IDL.  I am trying to input data for a 2d surface
>  created by a Fortran program.  The data has the following form:
>
>   xnum
>   x1 x2 x3 x3 x5
>   x6 x7 x8 ...
>   y1
>   z1 z2 z3 z4 z5
>   z6 z7 z8 ...
>   y2
>   z1 z2 z3 ...
>
>  Here, xnum is the number of x gridpoints, x1..xn form the x(xnum) vector,
>  y1, y2, ... form the y(ynum) vector and the z1..zn are the height data
>  at the current y value.  The problem is that ynum isn't explicitly
>  listed at the beginning of the file.
>
>  The obvious solution is to read the file twice, first to glean ynum,
>  and second to actually grab the data.  However, I wonder is it possible
>  (and desirable) to import the data in a single pass?  My IDL manual
>  suggests creating a 2D array bigger than I think I will need.  This is
>  inelegant and I wonder about some form of dynamic array.
```

Humm. I am notoriously bad at this kind of analysis, but I'm
sitting here with my leg up in the air popping pain pills
for my bum knee, so I figure this time I have a built-in
excuse: my brain is addled.

I don't believe this file can be read in the manner described
in the IDL manuals. They refer to a 2D array of values, when
what you really have here, it seems to me, are X and Y vectors
and the 2D array that corresponds to values at the
locations specified by the vectors, with the file arranged
in a slightly obscure way.

I presume that although the specific number of Y elements is
not written in the file, it is a specific known number. If
this is not the case, then I can't think of a way to avoid
reading the data in some kind of loop. Perhaps like this:

```
  ReadF, lun, newY, ZVector
  y = [Temporary(y), newY]
  z = [Temporary(z), ZVector]
```

Where you then reform Z into the proper sized 2D array at the
end of the loop. This is the sort of "dynamic array" concept
that you mention.

But suppose the number of Y elements were known. Say it is 30.
Then I might read this dataset with code like this:

```
     ; Read the X vector

   ReadF, lun, xnum
   x = FltArr(xnum)
   ReadF, lun, x

      ; Read the rest of the data in one gulp. Assume ynum=30.

   data = FltArr(30 * xnum)
   ReadF, lun, data

      ; Reform the data so that the y vector is the first column.

   data = Reform(data, xnum+1, 30)

      ; Separate the y vector from the z data.

   y = Reform(data[0,*])
   z = Temporary(data[1:*,*])
```

That should do it. This would be considerably faster than a loop.
And at least in my addled state, it *seems* elegant. :-)

Cheers,

David