Subject: Re: Collection of different size arrays?
Posted by cedricl on Wed, 14 Jul 2004 02:11:21 GMT
View Forum Message <> Reply to Message

Jonathan Greenberg <greenberg@ucdavis.edu> wrote in message
news:<BD19AC1F.1052D%greenberg@ucdavis.edu>...

> I'm unfamiliar with the use of structures, but I'm guessing this is the
> answer. Lets say I have some program generate arrays using the following
> algorithm:

Pointers are probably the cleanest solution. But personally I don't like having to care about possible memory leaks, so here are two alternatives:

1. Use a sparse matrix. Look up the SPRS* functions in the help. I'm not sure how easy it is to access individual elements, since they were designed for matrix operations, and not really storage. Perhaps you could build a simple function that takes care of the indexing.

2. Use structures. This is an ugly and inefficient solution, but it can be done, and the uglyness could be contained in a few key functions. The fields of a structure can be accessed by an index: "print, some_structure.(5)", and of course, each field of a structure can be an array of a different size. These fields can be built automatically using create_structure. I can flesh out the details tomorrow if you want.

Anyway, unless you have some profound aversion to pointers, like I do, you should use them in this case.

BTW, is there an online version of the IDL docs?

Cedric