Subject: Big arrays, reducing data
Posted by Eric Hudson on Wed, 21 Mar 2007 17:15:51 GMT
View Forum Message <> Reply to Message

Hi,

I have what I hope is an easy question and then probably a hard one.

1) I need to make some big arrays (ideally 16000^2 elements or more) but find I often get "unable to allocate memory" errors. Is there some way of determining (at run time) the largest array that I can make? In C, for example, I'd try to allocate the memory and check for whether it was allocated, then cut the array size if it wasn't. Is there an equivalent technique in IDL?

2) The reason I want to make these big arrays is that I have sets of on the order of 20,000 data curves (50-200 pts each). I'd like to reduce these to a set of "common curves" -- around 100 averaged/ extracted/smoothed curves which are representative of the larger set. The curves are complex -- I don't have anything to fit to them -- and they are noisy. But I get the feeling that if I handed someone a stack of 20,000 of them and said "sort these into groups which are similar" that they'd be able to do it. The question is, is there a good way to do this programmatically?

My original thought was to calculate an rms difference between each pair of curves and then reduce by partnering up the curves which are good matches to each other. Repeat. To do this easily seems to require big arrays and a pretty large amount of computation time. I'd appreciate any thoughts.

Thanks,
Eric