Subject: Generally accessing the rest of the elements in an array
Posted by T Bowers on Tue, 20 Feb 2001 22:58:47 GMT
View Forum Message <> Reply to Message

How do I access the 2nd + dimensions of an array generally, without knowing the
number of higher dims this array has. E.g. say a is a 3 column by n-dimensional
aray, and n is unknown. Here, I'll define it as:

a = indgen(3,2,4)

I want the equivalent of (in this case):
b = (a[0,*,*])^2 + (a[1,*,*])^2 + (a[2,*,*])^2

but this requires *'ing the correct dimensions ([0,*,*] for 3 dims,
[0,*,*,*] for 4 dims
etc). What I need is a general way to access the "rest" of the data, as Paul Harvey
would say.

Actually, what I *really* want is to access it all generally so if a is 3 columns, it'll be
as above

b = (a[0,*,*])^2 + (a[1,*,*])^2 + (a[2,*,*])^2
but if it's 4 columns, it'll be
b = (a[0,*,*])^2 + (a[1,*,*])^2 + (a[2,*,*])^2 + (a[3,*,*])^2
5 columns...
b = (a[0,*,*])^2 + (a[1,*,*])^2 + (a[2,*,*])^2 + (a[3,*,*])^2 + (a[4,*,*])^2
n columns...
b = (a[0,*,*])^2 + (a[1,*,*])^2 + (a[2,*,*])^2 + ... + (a[n-1,*,*])^2

but I don't think this is possible without a for loop.

Anyway, back to the ranch. I do know that i could ask the data and select in a case
or if then else block like

if (nDims eq 2) then b = b = (a[0,*])^2 + (a[1,*])^2 + (a[2,*])^2
else if (nDims eq 3) then b = (a[0,*,*])^2 + (a[1,*,*])^2 + (a[2,*,*])^2
else if (nDims eq 4) then b = (a[0,*,*,*])^2 + (a[1,*,*,*])^2 +
(a[2,*,*,*])^2

but I'm really doing other stuff in here too, and each of these cases would end up
having about 10 cases *within* those cases. Uuuuugly.

I tried variants of
a[0, *]
but I can't figure it out.

tia,
todd

Subject: Re: Generally accessing the rest of the elements in an array
Posted by T Bowers on Thu, 22 Feb 2001 19:47:41 GMT
View Forum Message <> Reply to Message

Thanks to all.

William: Thanks for a solution.

Jaco: Total() is out. I'm actually doing more. The sums of squares was just for example.

Paul: Yes, the thought of this was what made me have the idea of asking if there was a quick n easy way first. I hadn't thought about the a[0,*,*,*,...] way. I agree, ugh.

Again, thanks.
