Subject: Speedy way to get compare array elements. . .
Posted by Sean Raffuse on Wed, 13 Nov 2002 23:06:05 GMT
View Forum Message <> Reply to Message

Exalted newsgroup,

What is the fastest way to compare two adjacent values in an array?

Something like:

whoa_huge_jump = where(A[i+1]-A[i] GT 500)

Since this is IDL, I'm assuming I don't have to make a loop.

Thanks,

Sean

Subject: Re: Speedy way to get compare array elements. . .
Posted by marc schellens[1] on Mon, 25 Nov 2002 11:36:51 GMT
View Forum Message <> Reply to Message

Sean Raffuse wrote:
> Exalted newsgroup,
>
> What is the fastest way to compare two adjacent values in an array?
>
> Something like:
>
> whoa_huge_jump = where(A[i+1]-A[i] GT 500)
>
> Since this is IDL, I'm assuming I don't have to make a loop.
>
> Thanks,
>
> Sean

You might do it that way with
i=lindgen(n_elements(A)-1)
or have a look a the shift function which might be faster.
Even the convol function with a kernel of [-1,1] might be used here I
think.
Hope that helps,
marc

Subject: Re: Speedy way to get compare array elements. . .
Posted by Jaco van Gorkom on Thu, 28 Nov 2002 16:13:58 GMT
View Forum Message <> Reply to Message

"trouble" <the_cacc@hotmail.com> wrote in message
news:5f9f0a23.0211280658.71bff9f4@posting.google.com...
> Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:<on4raklqpk.fsf@cow.physics.wisc.edu>...
>> ...The cool thing about the trick is that IDL *automatically* truncates
>> the vector A so that it matches the length of A[1:*].
>
> Ah, but is there any way to not make it do that? Say you have 2
> vectors of different lengths:
>
> x = findgen(50)
> y = findgen(100)
>
> and you want to form z = x * y, but have z the same length as y
> putting zeros where x has no value. The (sorry) way I am doing it is
>
> z = y * 0
> z[0:49] = x * y
>
> which clearly is too much programmer work since I have to get the
> lengths... Any ideas?

Well, inserting an array into a subrange of another array can also be
done by specifying just the start index. So your
z[0:49] = x * y
is equivalent to
z[0] = x * y
In fact, the second option executes quite a bit faster.

cheers,
Jaco