Subject: How can I integrate? (easy question?)
Posted by bdb112 on Wed, 20 Mar 1991 12:37:28 GMT
View Forum Message <> Reply to Message

How can I efficiently do a running sum of a vector (integrate)? The explicitly coded version (IDL v2)
for i=1,n-1 x(i)=x(i) + x(i-1) achieves the desired effect, but takes (VS3100) 1 second for a 2000 element array compared to .01 sec for z=x+x, a similar number of operations. This is such an obvious thing to do that I must be missing something obvious - I can't find it in the userlib either (just deriv).

Subject: Re: How can I integrate? (easy question?)
Posted by bruce on Fri, 22 Mar 1991 00:22:39 GMT
View Forum Message <> Reply to Message

maybe you just want TOTAL(x).
--
-Bruce
"Just half a turn and there's your worm!"

-Bob Bet Bait Box

Subject: Re: How can I integrate? (easy question?)
Posted by ramesh on Mon, 25 Mar 1991 17:33:17 GMT
View Forum Message <> Reply to Message

In article <1991Mar20.233728.1@csc.anu.edu.au> bdb112@csc.anu.edu.au writes:
> How can I efficiently do a running sum of a vector (integrate)? The explicitly
> coded version (IDL v2)
> for i=1,n-1 x(i)=x(i) + x(i-1) achieves the desired effect, but takes (VS3100)
> 1 second for a 2000 element array compared to .01 sec for z=x+x, a similar
> number of operations. This is such an obvious thing to do that I must be
> missing something obvious - I can't find it in the userlib either (just
> deriv).

Assuming "i" represents the current array index value upto which you want to sum the elements of the array "x", try:

```
sum_x = total(x(0:i))
```

or a variation thereof. The dimension of array "sum_x" will be automatically dimensioned to the value contained in "i"

Note: Function "total" is a PV Wave system routine.

R.V.

Subject: Re: How can I integrate? (not so easy?)
Posted by bdb112 on Mon, 25 Mar 1991 20:18:41 GMT
View Forum Message <> Reply to Message

In article <1991Mar20.233728.1@csc.anu.edu.au>, bdb112@csc.anu.edu.au writes:
> How can I efficiently do a running sum of a vector (integrate)? The explicitly
> coded version (IDL v2)
> for i=1,n-1 x(i)=x(i) + x(i-1) achieves the desired effect, but takes (VS3100)
> 1 second for a 2000 element array compared to .01 sec for z=x+x, a similar
> number of operations. This is such an obvious thing to do that I must be
> missing something obvious - I can't find it in the userlib either (just
> deriv).

Two respondents have suggested
sum_x = total(x(0:i)) ; where i is the maximum idex of the array
When I try this, I get a scalar result, rather like a definite integral, when
what I want is a vector which is a running total, like an indefinite integral.

e.g. x = [1,1,2] sum_x = [1,2,4]
Of course, the next step would be to allow an optional vector which contained
the abscissae, if not equally spaced, or the spacing if equally spaced, but I
would be happy with the simple result.
