Subject: Re: Integrator taking vectors as input?
Posted by elias on Thu, 27 May 2010 18:07:10 GMT
View Forum Message <> Reply to Message

On May 27, 4:35 pm, Craig Markwardt <craig.markwa...@gmail.com> wrote:
> On May 27, 2:31 am, Elias <elias.rous...@gmail.com> wrote:
>
>
>
>
>
>> On May 26, 5:18 pm, "jsch...@gmail.com" <jsch...@gmail.com> wrote:
>
>>>> I was wondering if an IDL integrator exists where it can accept
>>>> vectors instead of scalars as inputs for upper and lower limits of the
>>>> integral. I want to apply it to big datasets and I want to avoid using
>>>> loops, which tend to be much slower.
>
>>> The IDL routines like QROMB accept vector inputs as the limits.
>
>>> See the documenation (e.g.http://star.pst.qub.ac.uk/idl/QROMB.html)
>>> for details.
>
>>> Josiah
>
>> Thanks a lot,
>
>> I tried QROMB and QSIMP, the problem is that they use internally loops
>> when vectors are provided for the limits. In that case, since my
>> integrals have constants that they are dependent from the values of
>> the limits, it doesn't work, since the constants are also vectors
>> (that I pass in the function I integrate through a COMMON block).
>
>> Eg. at a single step of the internal QROMB loop, the limits are
>> scalars while the constants are vectors. Therefore the code crashes...
>
> It's really up to you. You are really demanding a lot of an
> integrator: *no* loops and also presumably you want the result to be
> accurate. I suspect you will need to write your own if it's that
> important to you. Since most integrators need to subdivide the
> interval in some way - and hence use a loop - you are presumably
> limiting your accuracy that can be achieved with a single step of the
> trapezoidal rule.
>
> Craig

Craig, thanks for the answer

I understand that what I am asking is not easy, I was just wondering if something ready existed. I was not requiring a code without loops. I was simply considering a routine that when I give limits of eg. lower=[a1, b1], upper =[a2, b2] and constants=[c1, c2], that these, the discrete steps in between and the constants are passed to the function that is to be integrated as a vector. QROMB has an internal loop that passes first a1,a2 and then b1, b2 separately. Apart from being more time consuming (I have to apply this to datasets of 100-200 million points (or more since the dataset grows continuously), it also creates problems with my common block - so time is not the only issue.

Anyway, I did manage in the end to include a vectorized integration scheme in my code that uses Simpson's rule and gives an almost identical result as QSIMP (less than 1% difference in the worst case). I still havent applied it to the big dataset to see how much time I gain, but I am optimistic.

Elias
