Subject: Re: Chunk Array Decimation
Posted by JD Smith on Thu, 10 Oct 2002 18:51:39 GMT
View Forum Message <> Reply to Message

On Thu, 10 Oct 2002 07:26:46 -0700, Jaco van Gorkom wrote:

> "JD Smith" <jdsmith@as.arizona.edu> wrote in message
> news:pan.2002.10.10.00.18.45.172062.5209@as.arizona.edu...
>>
>> Aha, a very interesting and compact submission. I think there's one
>> error there. Should it not be data[ri[nh+1:*]] without the inds? I
>> translated as:
>>
>> h = histogram(inds, REVERSE_INDICES=ri) nh = n_elements(h) tdata =
>> [0.,total(data[ri[nh+1:*]],/CUMULATIVE)] vec9 =
>> tdata[ri[1:nh]-nh-1]-tdata[ri[0:nh-1]-nh-1]
>
> Ok, this is how I had coded it originally. I tested it on some
> ten-element sample vectors, got confused, and convinced myself that the
> extra inds had to be in. Reading the original problem again, I suppose
> you are right.
>
>> The biggest problem with this method, though, is roundoff error, which
>> accumulates like mad in a cumulative total of any length. If you do it
>> in floating point, as you've written, I find unacceptably large
>> roundoff errors for as few as 500 individual indices.
>
> Oops. Ok, so we should use /DOUBLE, indeed. But a quick test of
> TOTAL(REPLICATE(1., 30000000), /CUMULATIVE) reveals roundoff errors only
> after roughly 17,000,000 elements on my machine. If you tested the
> routine with a data vector data = FINDGEN(100000) then the average value
> for each data element would be 50,000 , so after 500 indices the
> cumulative total is already 25,000,000. Realistic data might well have a
> lower average value, or even zero average for data spread around zero,
> so that the cumulative total of 25,000,000 is only reached after many
> more elements or never.
>
> And what exactly do you mean by 'accumulates like mad in a cumulative
> total'? Suffers a cumulative total more from roundoff error than a
> single total? That should not be, should it? The way I see it, not
> having any computer science background, the error in a cumulative total
> would be the (very small) error in each sum (accumulating many, many
> times), plus the (much larger) roundoff errors that occur when we get to
> very high values. So as long as we do not get to high values we should
> be ok, especially since we take differences out of subelements of the
> total, which will not be very far apart, probably.

I guess I mean that to get the answer for a single element of the result

vector, say the 100th element, you must do *many* more additions with this technique. You are throwing away most of the additions in the accumulation. Each addition has associated with it an intrinsic roundoff error. Compounding these errors many times is what leads to the "mad accumulation" of roundoff in the cumulative sum. This is more or less the same whether performing a regular or cumulative total. However, with all the other techniques in this thread, only the data in that bin are summed (a sum which still has its own roundoff error, but it's much smaller).

Essentially the issue is, your way performs the calculation of v+x+y+z as:

a+b+c+d+e+f+g+h+j+k+l+m+n+o+p+q+r+s+t+u+v+x+y+z -
a+b+c+d+e+f+g+h+j+k+l+m+n+o+p+q+r+s+t+u

An example:

The sum of the first n natural integers is n*(n+1)/2 (a result Gauss discovered as a young schoolboy, much to the consternation of his lazy teacher, who had assigned her students a full morning to summing the first 100 integers). E.g. for n=1 million:

```
IDL> n=1000000LL & print,n*(n+1)/2
         500000500000

IDL> print,total(1.+findgen(n)),FORMAT='(D18.2)'
   499898744832.00
```

a difference of 101744640, or about .02%. Interestingly, the result is slightly different with accumulation:

```
IDL>  print,(total(1.+findgen(n),/CUMULATIVE))[n-1],FORMAT='(D18.2 )'
   499941376000.00
```

but it's roughly the same magnitude. Try it on your machine and see. Roundoff error is always roughly the same magnitude in terms of the mantissa. This can be a large number or small number, depending on the exponent. The relative roundoff is therefore a more interesting measure.

It's still a neat technique though.

JD