Subject: Re: Center of mass???
Posted by J.D. Smith on Tue, 09 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Paul Hick (pphick@ucsd.edu) writes:
>
>> Reasoning by analogy to the 2D case, this should work, I think:
>>
>> xcm = Total(Total(Total(array,3),2) * Indgen(s[0])) / totalMass
>> ycm = Total(Total(Total(array,3),1) * Indgen(s[1])) / totalMass
>> zcm = Total(Total(Total(array,2),1) * Indgen(s[2])) / totalMass
>
> Well, it seems to work in the simple-minded cases I
> tried it on. :-)
>
> I'll write it up in an article if no one else has
> serious objections.
>

Why stop at 3 dimensions? How about any dimension:

```
function com, arr,DOUBLE=dbl
  s=size(arr,/DIMENSIONS)
  n=n_elements(s)
  tot=total(arr,DOUBLE=dbl)
  if keyword_set(dbl) then ret=dblarr(n,/NOZERO) else
ret=fltarr(n,/NOZERO)
  for i=0,n-1 do begin
     tmp=arr
     for j=0,i-1   do tmp=total(tmp,1,DOUBLE=dbl)
     for j=i+1,n-1 do tmp=total(tmp,2,DOUBLE=dbl)
     ret[i]=total(findgen(s[i])*tmp,DOUBLE=dbl)/tot
  endfor
  return,ret
end
```

I would have posted this yesterday, but I couldn't help but feel there
must be a better way to do it.

Here's why:

In the 3D example above, Paul calculates total(array,3) twice, which
means once too many. It could have been saved between steps.

The number of times total() is called on the array in this

straightforward, brute-force method is n*(n-1) (for each of n dimensions
total the array over the other n-1), not counting the final index
total. And many of those are repeats to the exact same total() call
with the exact same array! Since we needed to do all directional
sub-totals, there must be a more efficient way, analogous to the 3D case
of saving total(array,3).

After a bit of scratching on paper, I found that I could do it with only
(n-1)(n+2)/2 calls to total(), by working simultaneously from the last
dimension backward and the first dimension forward, saving sub-totals
which can be shared by subsequent calls in both cases. For 10
dimensions that becomes 54 vs. 90 calls to total() (though only 5 vs. 6
for 3 dimens -- 1 saved call as described above). The problem is how to
code this model. Two reciprocating mutually recursive functions should
do the trick, but I haven't had time to explore. Any takers? Anybody
think they can save more calls to total()?

This may be folly, since few will use it for more than 3D, but it's an
interesting problem.

JD

P.S. Another wrinkle: The total()'s you'd prefer to save are the ones
that do the most work... those which total the "largest" dimensions. So
sorting by dimension size first of all might speed things up even more.

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|