Subject: Re: smooth function and rounding error
Posted by jeanh on Fri, 18 Dec 2009 00:13:01 GMT
View Forum Message <> Reply to Message

> y=DBLARR(n)
> y=A

y will be of the same type as A, so int, float etc...

> y[i]=total(A[i-width/2:i+width/2-1])/width

I believe you want to divide by width+1, because you are considering
A[i] in your total. A;so, be sure that width is a float!

Jean

> many thanks,
> simona

Subject: Re: smooth function and rounding error
Posted by simona bellavista on Fri, 18 Dec 2009 11:19:30 GMT
View Forum Message <> Reply to Message

thank you for your reply

> y will be of the same type as A, so int, float etc...

yes, anyway I am using double

>> y[i]=total(A[i-width/2:i+width/2-1])/width
>
> I believe you want to divide by width+1, because you are considering

ok, I try also this, but according to the previous link it should be
just width, I am trying to follow closely the prescription they give
for smooth function. can you see any discrepancy between their recipe
and my implementation?

> A[i] in your total. A;so, be sure that width is a float!

actually it was a long(it is an even long),because I thought idl would
cast it automatically to double in division.

now I am casting it explicitly:

y[i]=total(A[i-width/2:i+width/2-1])/double(width)

or

y[i]=total(A[i-width/2:i+width/2-1])/double(width+1)

in both cases I cannot see any difference, relative errors always are
~10^-1

any suggestions/hints/comments?

I forgot to say I am on linux and using idl 7.0.3

Subject: Re: smooth function and rounding error
Posted by lbusett@yahoo.it on Fri, 18 Dec 2009 13:05:32 GMT
View Forum Message <> Reply to Message

Hi Simona,

in the Help of the smooth function it is said that "If a Width
value is even, then Width+1 will be used instead."
so I think that if you are using an even number as width you should
modify both the line:

y[i]=total(A[i-width/2:i+width/2-1])/width

with

y[i]=total(A[i-width+1/2:i+width+1/2-1])/float(width+1)

, and the line:

for i=(width)/2,n-(width)/2-1,1 do begin

with

for i=(width+1)/2,n-(width+1)/2-1,1 do begin

Otherwise, you can leave the program as in the original post and just
add the command "width = width + 1" before the for loop.

Hope this helps,

Lorenzo

Subject: Re: smooth function and rounding error
Posted by simona bellavista on Fri, 18 Dec 2009 15:02:32 GMT

thanks Lorenzo, I missed that

> in the Help of the smooth function it is said that "If a Width
> value is even, then Width+1 will be used instead."

anyway it looks even worse, that is unexpected to me.

I tried those routines for width=[10,100,100]; I see the relative
error gets bigger as the width gets smaller.
I include a subset of my data
0.11032583 0.094536981 0.075814606 0.056046214
0.037181503 0.021005635 0.0089338077 0.0018573446
6.1061458e-05
0.0032205319 0.010475696 0.020567541 0.032017375
0.043320873 0.053128306 0.060387275 0.064433200
0.065024946
0.062328237 0.056852040

I tried the following on the above data:

```
width=4
smoothed=SMOOTH(x,width)
smoothed_eq=smooth_equivalent(x,width)
diff=abs(smoothed-smoothed_eq)/double(smoothed)
```

and I get a diff ~0.5

I also found an old post
http://groups.google.it/group/comp.lang.idl-pvwave/browse_th
read/thread/cfc1b4e88de957a1/e695c8d11af5446c?hl=it&lnk= gst&q=smooth#e695c8d11af5446c
where they talk about roundoff error in smooth, but it was 12 years
ago!

recap of implementation:

```
function smooth_equivalent, A, width
 n=n_elements(A)
 y=DBLARR(n)
 y=A
 width=width+1
 for i=(width)/2,n-(width)/2-1,1 do begin
  y[i]=total(A[i-width/2:i+width/2-1])/double(width)    ; or double
(width+1)
 endfor
 RETURN, y
end
```

cheers,
simona

Subject: Re: smooth function and rounding error
Posted by jeanh on Fri, 18 Dec 2009 17:18:34 GMT
View Forum Message <> Reply to Message

Simona,

for an even width, the following is working:
for i=(width)/2,n-(width)/2-1,1 do &
 y[i]=total(A[i-width/2:i+width/2])/(width+1)

I have removed the -1 in your upper boundary in A, and added width+1
only at the division, because you are also considering A[i].
With test data, it gives the same results as smooth.

For uneven width, it does not work... probably because the width must be
"more to the left" or "to the right", or simply because the way of
dividing width/2 returns some decimals, which are truncated by default.

Jean

simona bellavista wrote:
> thanks Lorenzo, I missed that
>
>> in the Help of the smooth function it is said that "If a Width
>> value is even, then Width+1 will be used instead."
>
> anyway it looks even worse, that is unexpected to me.
>
> I tried those routines for width=[10,100,100]; I see the relative
> error gets bigger as the width gets smaller.
> I include a subset of my data
> 0.11032583 0.094536981 0.075814606 0.056046214
> 0.037181503 0.021005635 0.0089338077 0.0018573446
> 6.1061458e-05
> 0.0032205319 0.010475696 0.020567541 0.032017375
> 0.043320873 0.053128306 0.060387275 0.064433200
> 0.065024946
> 0.062328237 0.056852040
>
> I tried the following on the above data:
>
> width=4
> smoothed=SMOOTH(x,width)
> smoothed_eq=smooth_equivalent(x,width)

>   diff=abs(smoothed-smoothed_eq)/double(smoothed)
>
> and I get a diff ~0.5
>
> I also found an old post
>   http://groups.google.it/group/comp.lang.idl-pvwave/browse_th
read/thread/cfc1b4e88de957a1/e695c8d11af5446c?hl=it&lnk= gst&q=smooth#e695c8d11af5446c
> where they talk about roundoff error in smooth, but it was 12 years
> ago!
>
> recap of implementation:
>
> function smooth_equivalent, A, width
>   n=n_elements(A)
>   y=DBLARR(n)
>   y=A
>   width=width+1
>   for i=(width)/2,n-(width)/2-1,1 do begin
>   y[i]=total(A[i-width/2:i+width/2-1])/double(width)   ; or double
> (width+1)
>   endfor
>   RETURN, y
> end
>
> cheers,
> simona
