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