
Subject: Re: why the variable become -NaN?

Posted by [jameskuyper](#) on Thu, 30 Oct 2008 11:03:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

xiao wrote:

```
> HI~ every one. i got another question here. I have long program and
> there is a part of it. The strange thing is , when I print at the
> first place, 'target' is normal float number. but after the small
> part, i print it again, it become -NaN, Why is that? any clues?
>                                     Thank you~~~
>
>   for jj= jjstr, jjend do begin
>   for ii= iistr, iient do begin
```

Where are the corresponding ending statements for those 'for' statements?

```
>   disx=abs(glon-plon(ii,jj))
>   disy=abs(glat-plat(ii,jj))
>   dist=sqrt(disx*disx+disy*disy)
```

The only use you are making of disx and disy in the code you've given us is to square them. If that's also true in the full program, there is no need to use 'abs', you'll get the same result whether or not you use it. Using it wastes a small amount of CPU time and adds a small but unnecessary bit of complexity to your program.

```
> ; must avoid missing data ("missing")
>   target=pdatt(ii,jj)
> ;
>   if (mmselec eq 1) then begin
>     print,target
>   endif
> ;target still right here
>
>   if (dist LT Range and $
>     target NE missing) then begin
>     if(dist eq 0.0) then begin
```

Please note that you've performed the wrong test here. Even if dist is not exactly 0.0, it can be very small. If it's too small, wgt will end up being very big. I'm not sure whether wgt can overflow; but other calculations involving wgt certainly could overflow. The check you should be performing is "if(abs(dist) lt epsilon)", where epsilon is a small number, greater than 0.0, whose precise value should be carefully chosen to prevent overflows.

```
>     wgt(ii,jj)= 1.
>   endif else begin
```

```
> ; wgt(ii,jj)= 1./(dist*dist)
>   wgt(ii,jj)= 1./dist
>   endelse
```

I find that code extremely suspicious. the value for wgt at a distance of 0 is 1.0, while for very small distances it's huge. That might make sense, if you're trying to do some kind of self-avoidance thing. However, in that case, the weight at a distance of 0.0 should be 0.0, not 1.0. Keep in mind that at a distance of 1.0, your weighting function has exactly the same value it has at the center. Is that really what you want to do?

```
> if (mmselec eq 1) then begin
>   print,target
> endif
> ; it is not right any more :(
```

That doesn't seem possible to me, since the value of "target" is read, but never written, between the two print commands. If this were C, I'd recommend looking into the possibility that you're writing past the ends of the "wgt" array, and thereby accidentally overwriting 'target'. This being IDL, I'm not sure what think.
