
Subject: Re: transforming an array where some values can't
Posted by [Evilio del Rio](#) on Mon, 09 Feb 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 6 Feb 1998, Joel D. Offenbergs wrote:

```
> bowler@eisner.decus.org writes:
>
>> I'm trying to write a function that will transform an arbitrary array by taking
>> the natural log of every element in the array.
>
>> What I have is
>
>> function xform, input
>> temp = alog(input) ; I realize I could combine these 2 statements
>> return, temp      ; but it makes debugging easier
>> end
>
>> Unfortunately, I can't guarantee that there won't be some elements that are
>> zero and thus invalid arguments to alog.  what's the "most efficient" way to
>> take the alog of any element that's greater than 0 and set the value of any
>> that are less than or equal to 0 to some small value (1e-7 for example)?
>
IDL softly handles any math exception (IEEE standard) so you don't need to
worry about the small/negative values for alog() :
```

```
IDL> print,alog(0.0)
-inf
IDL> print,alog(-1.0)
nan0x2000000
```

You just need to handle special case with the function `FINITE()`. However,
many of the standard plot/calculation routines can treat NaN values as
missing data, try for example:

```
IDL> plot,alog(randomn(seed,100))
```

Cheers,

Evilio Jose del Rio Silvan Institut de Ciencies del Mar
E-mail: edelrio@icm.csic.es URL: <http://www.ieec.fcr.es/~evilio/>
"Anywhere you choose,/ Anyway, you're gonna lose"- Mike Oldfield
