
Subject: "ALOG2" ? (Ugly code follows)

Posted by [Amara.Graps](#) on Sat, 03 Apr 1999 08:00:00 GMT

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

Hi Folks,

With regards to my previous message,

Since I needed an "ALOG2" function today, I went ahead and wrote it. I would still be interested in a better ALOG2 function (to compute the logarithm base 2 of the elements of an array) if any of you have a good one.

What I wrote is very ugly, inelegant, and the loops could bog down my application. Do you have any suggestions for getting rid of the loops?

Also, I am really surprised that RSI doesn't have such a simple function as this, since it is a standard function in C, and IEEE math functions, (and Matlab has this too). Maybe RSI can add this function capability to their "Perhaps To Do" list for future IDL upgrades ??

Thanks for any suggestions for any improvement on the code,

Amara

```
*****  
;  
FUNCTION WALOG2, x  
;  
;+  
; NAME:  
; WALOG2  
;  
;  
; PURPOSE:  
; This function the logarithm base 2 of elements of an array.  
;  
;  
; CATEGORY:  
; Simple Math  
;  
;  
; CALLING SEQUENCE:  
;   nlog2x = WALOG2(x)  
;  
;  
; INPUTS:  
;   x: array  
;  
;  
; OUTPUTS:  
;   nlog2x: 2^J  
;  
;
```

```

; MODIFICATION HISTORY:
; Written by: Amara Graps, Multiplex Answers, Germany
;   April 1999
;-

;Find epsilon precision (only caring about single precision for now)
eps = 1.0
WHILE ( (1 + eps) GT 1) DO BEGIN
    eps = eps/2.
END
eps = 2.0*eps

n = N_ELEMENTS(x)
nalog2 = DBLARR(n)

FOR index = 0, n-1 DO BEGIN

    num = x(index) ;We want to find the value q, such that 2^q = this number

    ;Find the integer power of 2 first
    k = 1L
    J = 0L
    WHILE ( k LT num ) DO BEGIN
        k=2*k
        J = 1+J
    END

    ;Use bisection to find value between 2^J and 2^(J+1)

    CASE 1 OF

    ( k NE num ): BEGIN

        maxj = J
        J = J - 1
        minj = J
        maxit = 30 ;maximum iterations
        dx = maxj-minj ;increment
        rtbis = minj ;left bisection point
        finish = 0 & q = 1 ;to enter while loop

        WHILE NOT(finish) AND (q LE maxit) DO BEGIN
            dx = dx * 0.5
            xmid = rtbis +dx
            fmid = 2L^xmid

            IF fmid LE num THEN rtbis = xmid ELSE finish = 0

```

```
;If number found is within epsilon then quit
IF ABS(fmid-num) LE eps THEN finish=1
```

```
    q = q + 1
END   ;while q
value = xmid
END   ;Case k NE num
```

```
ELSE: BEGIN
    ;Our number is exactly a power of 2
    value = J
END
```

```
ENDCASE
```

```
    nalog2(index) = value ;assign out array index the alog2 value
```

```
END ;FOR index
```

```
RETURN, nalog2
End   ;of function WALOG2
```

```
.*****
,
```

```
--
```

```
*****
```

```
Amara Graps          | Max-Planck-Institut fuer Kernphysik
Interplanetary Dust Group | Saupfercheckweg 1
+49-6221-516-543      | 69117 Heidelberg, GERMANY
Amara.Graps@mpi-hd.mpg.de * http://galileo.mpi-hd.mpg.de/~graps
*****
```

```
"Never fight an inanimate object." - P. J. O'Rourke
```