

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

>  
> What I would need is something like the MATLAB command `max(i)` where `i` is an  
> array(`n,m`) and `max(i)` returns a vector containing the maximum value of each  
> column. The IDL `max` command only returns a scalar containing the absolute  
> maximum of the array.  
>  
> I even tried a big loop running through each column to find the max of each,  
> but IDL doesn't accept loops with 64000 iterations  
>  
> What alternatives do I have?  
>  
> Thomas

—

|||||||\\\\\\\\\\\\-----//\\\\\\\\\\\\ //|||||||

Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,  
Cambridge, MA 02138      phone (617) 496 8318 fax (617) 495 4551  
e-mail mgs@io.harvard.edu web <http://www-as/people/staff/mgs/>

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

In fact IDL \*does\* accept loops with 64000 iterations; you just need to

make sure your loop variable is of LONG type, e.g.

```
for i = 0L, 64000L do begin
  if (i mod 1000L) eq 0L then print, i
endfor
```

Cheers,  
Liam.

--

Liam E. Gumley  
Space Science and Engineering Center, UW-Madison  
<http://cimss.ssec.wisc.edu/~gumley>

---

---

Subject: Re: Max value vector  
Posted by [Justin Ashmall](#) on Fri, 21 May 1999 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Sorry for replying to my own message but I was being a bit foolish.  
Why not use a LONG as an array counter:

```
for i=0L,640000L do .....
```

Justin

Justin Ashmall <[Justin\\_Ashmall@hotmail.com](mailto:Justin_Ashmall@hotmail.com)> wrote in message  
news:7i3mt2\$67e\$1@jura.cc.ic.ac.uk...

> It wouldn't be very efficient but you could use a while loop using a LONG  
as

> the counter

>

> e.g.

> i=0L ;type long

> WHILE (i LT 64000) do begin

> maxcol[i]=MAX(data[i,\*])

> i=i+1L

> end

>

> Justin

>

> Thomas C. Stubbings <[stubbing@fbch.tuwien.ac.at](mailto:stubbing@fbch.tuwien.ac.at)> wrote in message

> news:7i3mfr\$c9k\$1@news.tuwien.ac.at...

>> What I would need is something like the MATLAB command max(i) where i is

> an

>> array(n,m) and max(i) returns a vector containing the maximum value of

> each  
>> column. The IDL max command only returns a scalar containing the  
absolute  
>> maximum of the array.  
>>  
>> I even tried a big loop running through each column to find the max of  
> each,  
>> but IDL doesnt accept loops with 64000 iterations  
>>  
>>  
>> What alternatives do I have?  
>  
>  
>

---

---

Subject: Re: Max value vector  
Posted by [Justin Ashmall](#) on Fri, 21 May 1999 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

It wouldn't be very efficient but you could use a while loop using a LONG as  
the counter

e.g.  
i=0L ;type long  
WHILE (i LT 64000) do begin  
    maxcol[i]=MAX(data[i,\*])  
    i=i+1L  
end

Justin

Thomas C. Stubbings <[stubbing@fbch.tuwien.ac.at](mailto:stubbing@fbch.tuwien.ac.at)> wrote in message  
news:7i3mfr\$c9k\$1@news.tuwien.ac.at...  
> What I would need is something like the MATLAB command max(i) where i is  
an  
> array(n,m) and max(i) returns a vector containing the maximum value of  
each  
> column. The IDL max command only returns a scalar containing the absolute  
> maximum of the array.  
>  
> I even tried a big loop running through each column to find the max of  
each,  
> but IDL doesnt accept loops with 64000 iterations  
>  
>  
> What alternatives do I have?

---

---

Subject: Re: Max value vector

Posted by [Evan Fishbein](#) on Sun, 23 May 1999 07:00:00 GMT

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

---

Tom,

I have a routine that finds the max and min over the first index of a multidimension array using a for loop as you described. You need to make sure that the for loop variable is a long by initializing it to long

The code below takes `x1(n1,n2...nn)` and returns `lu(n2..nn,2)` where `lu(...,0)` are the lowerbounds and `(lu(...,1)` are the upperbounds

```
function lubound, x1, ignore=ignore, all=all
;
; calculates the lubounds of elements of x1 have the same 2nd, 3rd...nth
; indices. i.e. bounds by contracting over the first index.
;
if n_elements(all) ne 1 then all=0
if all eq 0 then begin
    x=reform(x1)
end else begin
    x=reform(x1, n_elements(x1))
end
sx=size(x)

if n_elements(ignore) eq 0 then begin
    if sx(0) le 1 then begin
        lub= [min(x), max(x)]
    end else begin
        nx=sx(2+sx(0))/sx(1)
        lub=make_array(nx, 2, type=sx(sx(0)+1) )
        x=reform(x, sx(1), nx)
        for i=0!, nx-1 do begin
            lub(i,*)=[min(x(*,i)), max(x(*,i)) ]
        end
        lub=reform(lub, [sx(2:sx(0)),2] )
    end
end else begin
    if sx(0) le 1 then begin
        ips=where(x ne ignore)
        if ips(0) ge 0 then begin
            lub= [min(x(ips)), max(x(ips))]
        end else begin
            lub=replicate(ignore,2)
        end
    end else begin
        nx=sx(2+sx(0))/sx(1)
```

```

lub=make_array(nx, 2, type=sx(sx(0)+1) )
x=reform(x, sx(1), nx)
for i=0!, nx-1 do begin
    ips=where(x(*,i) ne ignore)
    if ips(0) ge 0 then begin
        lub(i,*)= [min(x(ips,i)), max(x(ips,i))]
    end else begin
        lub(i,*)=replicate(ignore,2)
    end
end
lub=reform(lub, [sx(2:sx(0)), 2] )
return, lub
end

```

"Thomas C. Stubbings" wrote:

> What I would need is something like the MATLAB command max(i) where i is an  
> array(n,m) and max(i) returns a vector containing the maximum value of each  
> column. The IDL max command only returns a scalar containing the absolute  
> maximum of the array.  
>  
> I even tried a big loop running through each column to find the max of each,  
> but IDL doesnt accept loops with 64000 iterations  
>  
> What alternatives do I have?  
>  
> Thomas

---

Subject: Re: Max value vector

Posted by [Amara Graps](#) on Wed, 26 May 1999 07:00:00 GMT

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

---

"Thomas C. Stubbings" wrote:

>  
> What I would need is something like the MATLAB command max(i) where i is an  
> array(n,m) and max(i) returns a vector containing the maximum value of each  
> column. The IDL max command only returns a scalar containing the absolute  
> maximum of the array.  
>  
> I even tried a big loop running through each column to find the max of each,  
> but IDL doesnt accept loops with 64000 iterations  
>  
> What alternatives do I have?  
>

> Thomas

Thomas,

Here is another alternative among your choices. I wrote this simple function to mimic Matlab's min function. Just change the occurrences of "min" with "max" and you'll have what you are looking for.

Amara

```
-----

FUNCTION WMMIN, array, ind
;+
;NAME:
;    WMMIN
;
;PURPOSE:
;    To find the minimum elements of an array or matrix. This
;    function works the same way as Matlab's MIN: [y,ind]=min(array)
;
;CATEGORY:
;    Math.

;CALLING SEQUENCE:
;    Result = WMMIN(array, ind)
;
;INPUTS:
;    Array: The data array.
;
;OUTPUTS:
;    Result =
;        WMMIN returns the lowest value of an array, or if the
array
;        is a matrix, it returns the lowest value of each
*column* in
;        the matrix.
;    ind = the index of the lowest value(s).
;
;NOTES: IDL's array indices are one less than Matlab's.
;
;EXAMPLE:
;    >array=[[1,3,2],[6,3,4],[9,1,0]]
;    >print, array
;    1    3    2
;    6    3    4
;    9    1    0
```

```

;   y= WMMIN(array, ind)
;   >print, y
;   1   1   0
;   >print, ind
;   0   2   2
;
;
;MODIFICATION HISTORY:
;   Amara Graps, BAER, San Francisco, December 1994.
;   Amara Graps, Multiplex Answers, Heidelberg, April 1999.
;   Modified to handle "degenerate" matrices (0 length in one
dimen)
;-

```

```

array = REFORM(array) ;eliminates the "degenerate" dimension(s)

```

```

t = SIZE(array)

```

```

IF t(0) eq 1 THEN BEGIN

```

```

    ;1D array
    minval = MIN(array)
    ind = !c

```

```

ENDIF ELSE BEGIN

```

```

    ;Matrix
    numcol = t(1)      ;number of columns
    numrow=t(2)        ;number of rows

```

```

    minval = FLTARR(numcol)
    ind = FLTARR(numcol)

```

```

    ;Step through the cols, and find the mins of the columns
    ;(the way Matlab does it)

```

```

    FOR i = 0L, numcol-1 DO BEGIN
        tt = array(i,*)
        minval(i) = MIN(tt)
        ind(i) = !c
    
```

```

    END    ;i

```

```

ENDELSE

```

```

!c = 0
RETURN, minval
END

```

```

-*****
,
--

```

```

*****

```

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

---