
Subject: Re: where help

Posted by [pit](#) on Tue, 28 Jan 1997 08:00:00 GMT

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

In article <32ED447F.1CFB@irc.chmcc.org>,

Phil Williams <williams@irc.chmcc.org> writes:

> How do I get the x,y coords from the result of where?

Maybe you find the following routine usefull. It should do what you want for arrays of any dimension. Read the header how to use and interpret the result.

Peter

FUNCTION Prod, data

on_error, 2

produkt = 1.*data(0)

FOR i=1, n_elements(data)-1 DO \$

 produkt = produkt*data(i)

return, produkt

END

FUNCTION Where_n, data, cond, count

;+

; NAME:

; WHERE_N

; PURPOSE:

; Find the n-dim indices where the array DATA fullfills a given
; condition.

; CALLING SEQUENCE:

; RESULT= WHERE_N (Array condition, [,COUNT])

; INPUTS:

; DATA : n-dim array expression as explained in the Manpage for
; where. All Constructs (like "eq 0", "GT limit" etc)
; are allowed.

; OPTIONAL INPUTS:

; COUNT : (Output) used for passing back the number of matches

; OUTPUTS:

; Result is a long array of dimension (COUNT, N) where COUNT is
; the number of zero elements in DATA and N is the dimension of
; the input array. The optional parameter COUNT holds the
; number of matches

; PROCEDURE:

; An Array expression is a byta array that is unity where the

```
; condition is fulfilled and zero elsewhere. The where-function
; returns a 1-d array. Reformat this and return the reformatted
; array.
; EXAMPLE:
; Be A an 4-dim array. The call
; Res = where_n(abs(A) LE 10)
; returns an array of the size (n_matches, 4). For example the
; 4th match element in A can be addressed as
; A(res(3,0), res(3,1), res(3,2), res(3,3)).
; MODIFICATION HISTORY:
; 20-Okt-1992 P.Suetterlin, KIS 2-d version Where2
; 29-Aug-1995 PS extended to n-dim, changed syntax to match the
; use of the IDL where-function.
;-
```

```
on_error, 2
```

```
IF n_params() EQ 0 THEN BEGIN
  print, 'Use: result=where_n(data[,count]) data is n-dim array'
  return, undefined
ENDIF
```

```
s = size(data)
dim = s(0)
```

```
ix = where(data EQ 1, count)
```

```
IF count EQ 0 OR s(0) EQ 1 THEN return, ix
```

```
res = intarr(count, dim)
res(*, 0) = ix MOD s(1)
FOR i=1, dim-1 DO $
  res(*, i) = (ix MOD long(prod(s(1:i+1))))/prod(s(1:i))
```

```
return, res
```

```
END
```

```
--
```

```
~~~~~
~~~~~
```

```
Peter "Pit" Suetterlin http://www.uni-sw.gwdg.de/~pit
```

```
Universitaets-Sternwarte Goettingen
```

```
Tel.: +49 551 39-5048 pit@uni-sw.gwdg.de
```

```
-- * -- * ...-- * -- * ...-- * -- * ...-- * -- * ...-- * --
```

```
Come and see the stars! http://www.kis.uni-freiburg.de/~ps/SFB
```

```
Sternfreunde Breisgau e.V. Tel.: +49 7641 3492
```
