Subject: Padding arrays - vector subscripts not working
Posted by James[2] on Fri, 02 Jul 2010 18:45:55 GMT
View Forum Message <> Reply to Message

Hello Everyone,

I'm writing a phase correlation routine for image registration. A preliminary step is padding the images with zeroes. Here's a simplified example of what I'd like to do:

```
X = bytarr(5,5,5)
Y = bindgen(2,2,3)
diff = (size(X, /dimensions) - size(Y, /dimensions))/2
X[diff] = Y
```

Unfortunately, this gives me an error message:

% Array subscript for X must have same size as source expression.

The strange part is, if I type in:

```
X[1,1,1] = Y
```

it works fine - but using the array DIFF (whose contents are [1,1,1]) it fails. What's happening? Is there any way I can do this in IDL?

By the way, I would like my final program to work on arrays with any number of dimensions, so I'd rather avoid a kludge like X[diff[0], diff[1], diff[2]] = Y.

Subject: Re: Padding arrays - vector subscripts not working
Posted by penteado on Tue, 06 Jul 2010 19:15:32 GMT
View Forum Message <> Reply to Message

On Jul 6, 3:38 pm, James <donje...@gmail.com> wrote:
> out[start] = array

Sorry, that was meant to be

```
out[start]=array[*]
```

However, this lead to me noticing that the part of out you want to set is not contiguous. So start will have to be an array containing the proper indexes of out that correspond to the locations to be changed, instead of just the starting index. Which is a bit tricky for an arbitrary number of dimensions.

Subject: Re: Padding arrays - vector subscripts not working
Posted by penteado on Tue, 06 Jul 2010 20:48:25 GMT
View Forum Message <> Reply to Message

I think this does it, but you should check. I am assuming that both out and array have the same number of dimensions.

```
function pad, array, dims

  ;error checking, etc. removed for more compact newsgroup post

  type = size(array, /type)
  out = make_array(dims, type=type)

  ;calculate where to put the original array in the new
  sz = size(array, /dimensions)
  dimdiff=dims-sz
  diff = dimdiff/2

  out_strides = [1L, product(dims, /integer, /cumulative)]
  out_ind = total(out_strides*diff, /integer) ;starting index where
out is to be set

  array_ind=l64indgen(n_elements(array)) ;indexes of array to use in
out
  out_ind+=array_ind ;make the index list a contiguous list starting
at out_ind
  ;if more than 1D, the section of out to be set may not be
contiguous, so make the jumps in the indexes at every dimension
  array_strides = [1L, product(sz, /integer, /cumulative)]
  for i=1,n_elements(dims)-1 do out_ind+=(array_ind/
array_strides[i])*dimdiff[i-1]*out_strides[i-1]

  out[out_ind] = array[array_ind]
  return, out
end
```

Subject: Re: Padding arrays - vector subscripts not working
Posted by James[2] on Wed, 07 Jul 2010 18:20:26 GMT
View Forum Message <> Reply to Message

The following case-statement method is pretty ugly, but it is good for speed and memory use. I tested your program and it works, but it is slower than the program below because of its L64INDGEN call.

```
type = size(array, /type)
out = make_array(dims, type=type, value=filler)
```

```
x = (dims - sz)/2

case numdims of
  1: out[x[0]] = array
  2: out[x[0], x[1]] = array
  3: out[x[0], x[1], x[2]] = array
  4: out[x[0], x[1], x[2], x[3]] = array
  5: out[x[0], x[1], x[2], x[3], x[4]] = array
  6: out[x[0], x[1], x[2], x[3], x[4], x[5]] = array
  7: out[x[0], x[1], x[2], x[3], x[4], x[5], x[6]] = array
  8: out[x[0], x[1], x[2], x[3], x[4], x[5], x[6], x[7]] = array
endcase

return, out
```