Subject: Generating a grid in the 3D,4D,5D...N space - Advice/Combinatory/Matrices
Posted by clement.feller@obspm. on Mon, 13 Nov 2017 14:50:11 GMT
View Forum Message <> Reply to Message

Hello everyone,

I coming back to you for some advice on how to properly generate a grid in an N-D space. I hope that this expression is the proper one in english, but in any case, let me illustrate this by the following exemple:

```
>  a = indgen(3,3) & print, a
 0 1 2
 3 4 5
 6 7 8
```

What I am looking for would be to find the clean and proper IDL way to generate the following sets of combinations:

```
 0,1,2
 0,1,5
 0,1,8
 0,4,2
 0,4,5
 .....
 .....

 6,7,2
 6,7,5
 6,7,8
```

Now, I have found ways to do this for a 2D,3D,4D,5D space with either nested loops (yuck! I know), or with combinations of rebin, reform and transpose.
I've been successfully using those solutions for several weeks, yet I wonder on how to expand this to a general case and in the proper IDL way.

Why and how this is useful to me ?
I am actually trying to evaluate a function with several parameters. Let's call it f(x, p_0, ..., p_n).
Given x, I want to evaluate f with multiple sets of parameters.
E.g. to generate a regular grid I can use INTERPOLATE, e.g. for a 2D space with 10 evaluations for each dimension:

```
>  p = [[0.,1.], [0,100]]
>  p = transpose(p)
>  interpolate(p,1./9.*findgen(10))
       0.0000000       0.0000000
      0.11111111       11.111112
      0.22222222       22.222223
      0.33333334       33.333336
      0.44444445       44.444447
```

```
   0.55555558      55.555557
   0.66666669      66.666672
   0.77777779      77.777779
   0.88888890      88.888893
    1.0000000      100.00000
```

If one wants an irregular grid (with 10 evaluations in the first dimension and 5 in the second one), one can use a nested loop and play with indices.

So, once you have this table, how can one generate the proper sets of combinations of indices ? Another way to look at it is that you just want to "multiply" or chunk index your table, i.e. to generate the n vector used in the histogram's i-vector example (http://www.idlcoyote.com/tips/histogram_tutorial.html).

I've playing around with nested indgen, looking for a repetitive motive from the 2D to the 5D space when using rebin, reform, transpose to assemble a generic command. But nothing much so far....

Does anybody out there already had a go with such problem before or any advice ?

I thank you all in advance for your replies.
Cheers,
/C