Subject: Re: Generating a grid in the 3D,4D,5D...N space - Advice/Combinatory/Matrices
Posted by clement.feller@obspm. on Tue, 14 Nov 2017 13:16:45 GMT
View Forum Message <> Reply to Message

Hello again,

Thanks both of you for your replies.

@Mike: I had looked into this before (I think Jeremy Bailin has published a code similar to yours called combigen.pro), but I then meet difficulties in selecting part of the generated combinations.

@Markus: I say, your code is sleek and nifty. I like your solution.

In the meantime, I had given this problem some more thoughts and I had come up with another slow ugly one that doesn't work for all cases:

```
function gen_indices_comb, m, n
;d I/O:
;d m -> long integer corresponds to the number of row in original table
;d n -> long integer corresponds to the number of columns in original table
;d
;d vals -> long array listing the vectors of indices to extract the
;d         different possible combinations from the values of the original
;d         table
;d
;d NOTES: SLOW CODE, a mitigation of the values of m and n is REQUIRED
;d        Cases, where m & n are greater than 9, are not to considered
;d        with this code

 nmax = m^n

;c Assemble command generating vector of indices
 cmd = 'tmp = ['
 for ijk=(m-1L),1L,-1L do $
   cmd += ' (lmn/n^'+string(ijk,format='(I03)')+') mod n,'
 cmd += 'lmn mod n ]'

;c initialiase memory
 ini = indgen(m,n)
 tmp = lonarr(m)
 val = lonarr(m, m^n)

;c execute command for each type of combination
 for lmn=0L,(n^m-1L) do begin
   void = execute(cmd)
   if void ne 1 then message, ' > Error generating indices.'
   val[*,ijk] = ini+tmp*m
```

```
 endfor

 return, val
end
```

Afn I'm considering this post solved, I'll update it with a definitive version of my solution.

Again thanks your replies,
/C
