On Wednesday, March 18, 2015 at 12:38:53 PM UTC-7, Paul van Delst wrote:
> Hello,
>
> On 03/17/15 22:49, Matthew Argall wrote:
>>> t=FINDGEN(5,200)
>>> x = t(1,*)
>>> y = t(2,*)
>>> z = t(3,*)
>>>
>>> p = PLOT3D(x, y, z,'o')
>>
>>
>> I have found that function graphics are particularly picky about
>> whether you pass them a column vector or row vector. Problems arise when
>> you pass data in as a row vector (a 1xN array, as you are doing above).
>> The solution is to reform your data into a column vector (Nx1 array),
>> like so
>>
>> x = reform(t[1,*])
>> y = reform(t[2,*])
>> z = reform(t[3,*])
>>
>> p = plot3d(x, y, z, 'o')
>
> Matthew's suggestion is a good one, but being the memory-layout-worrier
> that I am (preferring t[*,0] over t[0,*]), I would simply transpose the
> "t" array directly after reading it in from wherever it was created,
>
> The original data...
>
> IDL> t = findgen(5,200)
> IDL> help, t
> T   FLOAT   = Array[5, 200]
> IDL> print, t[0,*]
>   0.00000
>   5.00000
>   10.0000
>   15.0000
>   20.0000
>   25.0000
>   30.0000
>   35.0000
>   40.0000
>   45.0000

>   .....
>
> Transpose it for all subsequent use:
>
> IDL> t = transpose(t)
> IDL> help, t
> T   FLOAT   = Array[200, 5]
> IDL> print, t[*,0]
>   0.00000   5.00000   10.0000   15.0000   20.0000
>   25.0000   30.0000   35.0000   40.0000   45.0000
>   ....
>
> When the trailing dimension is the degenerate one, IDL happily ignores
> it....
>
> IDL> x = COS(t) * (1 + t / 10)
> IDL> help, x
> X   FLOAT   = Array[200, 5]
> IDL> p = PLOT3D(x[*,0], x[*,1], x[*,2],'o')
>
>
> cheers,
>
> paulv

Thank you
Now I can plot it.
Sometimes I don't understand how this formatting works...

-LMM