Subject: Re: transforming a row vector into a column vector (continued)
Posted by peter.albert@gmx.de on Tue, 25 Oct 2005 14:48:18 GMT
View Forum Message <> Reply to Message

Hi Francois,

well, the answer is already in your post :-)

What does the output of size(b) actuall mean?
The first "2" tells us that b is 2-dimensional.
The following "1" tells us that the number of columns (the x-dimension) is 1.
The following "4" tells us that the number of rows (the y-dimension) is 4.

Voila, that's the definition of a column vector, isn't it? It has one column and n rows.

IDL does not know different types of vectors, a vector in its 1-dimensional form always is a row vector.

As for your second question, again the answer is in your post: The "possibility for knowing what type an array is" is the SIZE() command:

```
case 1 of
  (size(array))[0] eq 1:print, "This array is a row vector"
  (size(array))[0] eq 2: print, (size(array))[1] eq 1 $
    ? "This array is a column vector" $
    : "This array really is a 2D array"
  else: print, "This array has even more dimensions"
endcase
```

Cheers,

Peter

Subject: Re: transforming a row vector into a column vector (continued)
Posted by Mark Hadfield on Tue, 25 Oct 2005 21:39:33 GMT
View Forum Message <> Reply to Message

Peter Albert wrote:
> IDL does not know different types of vectors, a vector in its
> 1-dimensional form always is a row vector.

I would go further than that. IDL is not a matrix-oriented language (unlike Matlab, which definitely is) and its fundamental data structures are the scalar and the array, the latter with 1-7 dimensions. With

arrays it uses a Fortran-like relationship between indexing and storage (ie inner dimension varies fastest in memory) and it also supports a handy 1-D indexing scheme that makes use of this relationship.

That's about it. Basically, IDL doesn't know about rows or columns, vectors or matrices, tensors or quaternions. You *can* simulate these mathematical structures with arrays, and some built-in IDL functions and operators will help you, but make sure you check the conventions about the correspondence between the IDL concepts (arrays, dimensions, indices) and the mathematical concepts (matrix, row, column).

And read:

http://www.dfanning.com/misc_tips/colrow_major.html

PS: I have used IDL successfully to solve SVD problems, but I had to read the documentation carefully and check out my understanding with toy examples.

PS2: To my mind, IDL's agnosticism about matrices is vastly preferable to Matlab's "everything is a double-precision matrix" stance.

--
Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)