
Subject: Re: Bug/feature in matrix multiply
Posted by [Martin Schultz](#) on Mon, 15 Mar 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Fardal wrote:

> [...]

> So my question becomes, why does this happen? If IDL is going to
> treat an 3x1 array differently than a 3-element vector, it shouldn't
> just cavalierly remove the trailing dimension in my opinion. The
> behavior is probably documented somewhere but I couldn't find it in
> the hyperhelp. There is this one sentence in "Combining Array
> Subscripts with Others": "As with other subscript operations, trailing
> degenerate dimensions (those with a size of 1) are eliminated."
>
> I also notice that the behavior is somewhat inconsistent, in that
> converting an expression to one of the same type does not remove
> the trailing dimension:
> [...]

I agree (somewhat). Generally, I found that IDL is quite "smart" in removing trailing dimensions so that one doesn't have to worry too much about them. But when you do encounter a case where you need to (and this is frequent for any type of matrix manipulation), IDL is just too smart and you have to think twice to outsmart it ;-)

Anyway: you can always make sure you get what you want with

```
a = transpose(reform(a))
```

These statements are not very costly in terms of execution time, because it's only messing around with the array descriptor (at least I believe so).

Regards,
Martin.

--

Dr. Martin Schultz
Department for Engineering&Applied Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu

Internet-homepage: <http://www-as.harvard.edu/people/staff/mgs/>
