Subject: How does REFORM work in PV-Wave
Posted by jeyadev on Tue, 30 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

I have a question about the way reform works on 3d arrays.

First, the specific problem. I have a data file that has 8 columns and 56 rows. Each column corresponds to an independent variable. The The 56 rows correspond to an 8 x 7 array of points on a plane. What I would like to do is to read the data file and then make a new 3d array so that each plane of this 3d array corresponds to one of the independent variables (so that there are 8 separate 8 x 7 planes).

After some experimentation I found that the solution was

newdata = reform(data, 8, 8, 7)

But, I do not understand why this works. In fact, I find it quite counter intuitive and I do not understand why the 'plane' index should be the first one (more below). The fact that there is a degeneracy here made my experimentation easier! There are only 3 possibilities, instead of 6. But, that raises the more interesting question. Also, I understand that the first index of newdata identifies the *plane* for each variable and I can do, for example,

surface, reform(newdata(1,*,*))

to see how the second variable varies over the plane. Now,

if I had N variables on a grid with C columns and R rows and the data file was in the format

r1 c1 v1 v2 v3 ... vN
r2 c1 v1 v2 v3 ... vN
.....
.....
R c1 v1 v2 v3 ... vN
r1 c2   ....
r2 c2   ....

etc. In the above, you can ignore the first two columns -- they are there just to show the order. The actual data and just C x R rows of N elements each.

what would be the reform command be? (With that I should be able to figure out the answer for what it should be in the row and column orders were reversed in the original data file!)

I guess what I am missing is the *order* in which elements are stored.
I think that PV-Wave stores 2d arrays in the 'row' format (first index
varies fastest), but what about higher dimensional arrays?

thanks

--

Surendar Jeyadev jeyadev@wrc.xerox.com
