Subject: Re: 3d matrices and LUSOL
Posted by James Kuyper on Fri, 25 Jan 2002 17:18:07 GMT
View Forum Message <> Reply to Message

M wrote:

> Hi all,
>
> I am in desperate need of help!!
>
> I have a set of linear equations represented as matrix arrays which need to
> be solved using the LU decomposition technique. The two arrays consist of a
> 14 x 14 array, and a 14 x 1 array, but each element in the matrices itself
> is an array of 221 elements (ie the matrices are 3dimensional...?)
>
> So i need to solve the system using LUDC and LUSOL, but i have to do it 221
> times (ie a solution for each 'layer' of the matrices)
>
> The first question is, is there a way to declare the two input arrays as
> 3-d? I tried defining the matrix using matrix=[[a,b,..],[...,...,...] etc]
> where a,b,... = arrays, but this isn't recognised as a 14 x14 square matrix
> which is 221 elements 'deep'. Instead, it expands each array across the
> row, making it a 3094 x 14 matrix. (it needs to be square to run LUDC)
>
> Is there a way i can force IDL to see it as a 'layered' 3-d matrix?

Yes. If 'a', 'b', 'c' etc are 221-element arrays, then

```
    [ [[a],[b]], [[c],[d]] ]
```

defines a (221,2,2) array. If you need a different ordering of indices
(I suspect that you want (2,2,221)), then you need to use transpose()
and reform(). In APL, which is part of IDL's ancestry, the equivalent of
transpose() could hanndle arrays of any dimension. However, in IDL
transpose() won't work on arrays of rank higher than 2. Therefore, do
the following:

```
big = reform(transpose([[a],[b],[c],[d]]),2,2,221)
```

You'll need to call LUDC and LUSOL seperately for each of the 221
layers. Therefore, if there's any advantage to forming them all into one
big array, that advantage will have to lie in some other part of the code.