Subject: Matrix Vector Multiply in IDL
Posted by K Banerjee on Tue, 26 Nov 2002 16:25:58 GMT
View Forum Message <> Reply to Message

Folks,

I am trying to port some IDL code to C/C++ and I am stuck on a matrix multiplication problem. As I understand things, IDL stores an M by N matrix as an N by M array. Here are the dimensions of the relevant arrays:

KX: 148 by 1, a vector in the linear algebra sense.
Fac1: 148 by 3 array, a 3 by 148 matrix.
fac1tr: transpose of Fac1

```
; Explictily carrying out the matrix-vector multiply:
  fac1tr = transpose(Fac1)
  tempBetas = fltarr(3)
  for i = 0, 2 do begin
    for j = 0, 147 do begin
      tempBetas(i) = tempBetas(i) + fac1tr(i, j) * KX(i)
    endfor
  endfor
```

Here are the values in tempBetas:

```
tempBetas[       0] = [  9.24656e-06]
tempBetas[       1] = [ -5.54790e-06]
tempBetas[       2] = [      745.434]
```

The corresponding values from my C/C++ code are:

```
element[0] = [-4.01067e-07]
element[1] = [2.75595e-08]
element[2] = [745.434]
```

As you can see, the above values are "close" to the values computed in IDL.

```
; Now carrying out IDL's matrix vector multiply:

  Betas=Fac1##transpose(KX)
```

The values of Betas end up being:

```
Betas:[       0] = [      4.35200]
```

```
Betas:[       1] = [     -2.40834]
Betas:[       2] = [      795.034]
```

Why don't tempBetas and Betas have the same values? How is using
"##" different from computing the matrix-vector multiply with
2 nested for loops? (I am probably missing something obvious!)

Thanks.

K. Banerjee