Subject: Re: function of two variables
Posted by andy on Fri, 18 Mar 1994 18:48:05 GMT
View Forum Message <> Reply to Message

In article <CMtu54.5o@ireq.hydro.qc.ca>, brooker@toka.ireq-ccfm.hydro.qc.ca writes:

```
>  Hello from Canada!!
>
>  I have the following IDL question:
>
>  Assume I have the functions Y=2.*X + 3. and I want to plot Y vs X over the
>  range [0,10]
>
>  I could code this as
>
>   X=findgen(101)/100.*10.
>   Y=2.*X + 3.
>   plot,x,y
>
>  Very nice!! Only array operations. NO LOOPING!!!!!! Very FAST!!!!!
>
>  But what if now I had Z=2.*X + 3.*Y and I want to plot Z as X and Y both
>  range from 0 to 10. A way to code this is
>
>   X=findgen(101)/100.*10.
>   Y=X
>   num_y=n_elements(Y)
>   num_x=n_elements(X)
>   z=fltarr(num_x,num_y)
>   for j=0,num_y-1 do begin
>        z(*,j)=2.*X + 3.*Y(j)
>        endfor
>   surface,Z,X,Y
>
>  Very inefficient because of the loop!! Very slow!!
>
>  Is there anyway to do this more efficiently?
>
>  Thanks,
>         Peter Brooker
>         brooker@toka.ireq-ccfm.hydro.qc.ca
```

Peter,

I would recommend trying...

```
  z =  fltarr(num_x,num_y)+replicate(1,num_y)#(3.*y)+(2.*X)#replica te(1,num_x)
```

I didn't find your example slow on a SUN Sparc 2, but hopefully this
suggestion will be faster on your system.

Andy

--

```
   ,__o    Andrew F. Loughe   (Code 971)
  -\_<,    NASA Goddard Space Flight Center    phone: (301) 286-5899
(*)/'(*)   Greenbelt, MD 20771                 email: andy@toto.gsfc.nasa.gov
```

Subject: Re: function of two variables
Posted by black on Mon, 21 Mar 1994 14:01:57 GMT

View Forum Message <> Reply to Message

In article <CMtu54.5o@ireq.hydro.qc.ca>,
 <brooker@toka.ireq-ccfm.hydro.qc.ca> wrote:

[Stuff deleted]

```
> But what if now I had Z=2.*X + 3.*Y and I want to plot Z as X and Y both
> range from 0 to 10. A way to code this is
>
>  X=findgen(101)/100.*10.
>  Y=X
>  num_y=n_elements(Y)
>  num_x=n_elements(X)
>  z=fltarr(num_x,num_y)
>  for j=0,numy-1 do begin
>      z(*,j)=2.*X + 3.*Y(j)
>      endfor
>  surface,Z,X,Y
>
> Very inefficient because of the loop!! Very slow!!
>
> Is there anyway to do this more efficiently?
>
> Thanks,
>       Peter Brooker
>       brooker@toka.ireq-ccfm.hydro.qc.ca
```

What you need are two two dimensional arrays that contain X and Y values. The size of number of X co-ords by number of Y co-ords. These arrays essentaially stor the X & Y values at each point on the grid. So the value in the elements in the X array change in one direction say along the rows, but stays consatnt in the other direction. The Y array does the opposite. Given that the Y=X in your code you only need to come up with one array, since Y is the transpose of X.

So the next trick is to come up with the X array. This is simply done by taking your existing X array and using the matrix multiply in the following way

1) create a 1d array that contains 1 of the size required - the number of Y indices call this UNITY

2) do X#UNITY.

This does what you want.

John Black.