Subject: implementing pre pritten routines
Posted by aontman1 on Sat, 19 Jun 2004 21:34:34 GMT
View Forum Message <> Reply to Message

Hi,
I was wondering if anyone could help me. I was given pre written routine
this is what is given to me (by the name of the author i believe it came off one the messages here)
I have 3 dimensional array of data (100x40x10)
I was wondering if someone could help me write a code that would interpolate the data (I hope that makes some sense). Any other suggestions are welcome
Thank you so much for taking your time to answer my silly question
Lafoz

```
; PURPOSE:
;
;    Perform n-dimensional linear interpolation on arrays of
;    arbitrary dimensionality (ala the bilinear and trilinear
;    interpolation of INTERPOLATE).
;
; CALLING SEQUENCE:
;
;    value=ninterpolate(data,point)
;
; INPUTS:
;
;    DATA: The n-dimensional array to interpolate.
;
;    POINT: A vector of length n specifying the (single) point for
;       which the interpolated value is desired.
;
; OUTPUTS:
;    The interpolated value at the specified point.
```

this is the code:

```
function ninterpolate,data,point
  n=n_elements(point) & two_n=2L^n & reb=[n,two_n]
  if n ne size(data,/N_DIMENSIONS) then $
     message,'Point must specify 1 coordinate for each dimension'
  if n eq 1 then return, interpolate(data,point[0])
  inds=(rebin(indgen(1,two_n),reb) AND
rebin([ishft(1,indgen(n))],reb)) gt 0
  base=floor(point) & f=float(point)-base
  f=[(1.-f),f]
   multi_to_single=[1,product((size(data,/DIMENSIONS))[0:n-2],/ CUMULATIVE)]
  data_ind=rebin(base,reb)+inds
  data_ind=long(total(data_ind*rebin(multi_to_single,reb),1))
```

```
  return,  total(product(f[rebin(indgen(n),reb)+inds*n],1)*data[data_in d])
end
```