
Subject: Finding Backward trajectory of wind data
Posted by [shambhu](#) on Fri, 16 Dec 2011 04:50:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all

I am a newbi to IDL. Am trying to find wind trajectory using nearest grid point. Am using NCEP data file air.2000.nc which consists of lon, lat, level, and time. But am getting error which is not reachable to me. Please help me on this. Code & error iare as follows:

```
pro ngp,fid,posx,nx,posy,ny,posz,nz, $  
AVERAGE=average,WRAPAROUND=wraparound,NO_MESSAGE=no_message
```

```
filepath='/home2/shambhu/'  
filename=dialog_pickfile(path=filepath)  
fid=ncdf_open(filename)  
;print,'file id is: ',fid
```

```
fileinq_struct=ncdf_inquire(fid)  
nvars=fileinq_struct.nvars  
;print,'variables in file: ',nvars
```

```
for varndx=0,nvars-1 do begin $  
  varstruct=ncdf_varinq(fid,varndx)  
  for attndx=0,varstruct.natts-1 do begin $  
    attname=ncdf_attname(fid,varndx,attndx)  
    ncdf_attget,fid,varndx,attname,fid  
    ;print,attname,string(fid)  
    posx=string(fid)  
    ;print,'posx: ' , posx  
    ;print,'Var Name:',  
    varstruct.name,' ',varstruct.datatype,' ',varstruct.dim
```

```
  endfor  
endfor
```

```
nrsamples=n_elements(fid)  
nparams=n_params()  
print,nparams  
print,'nparams: ',nparams  
dim=(nparams-1)/2
```

```
print,'dim: ',dim
```

```
IF dim LE 2 THEN BEGIN  
  nz=1  
  print,'nz: ',nz
```

```

    IF dim EQ 1 THEN ny=1
ENDIF

read,'Enter value for nx',nx
read,'Enter value for ny',ny

nxny = long(nx)*long(ny)

print,'nxny: ',nxny
; Some error handling.

on_error,2 ; Return to caller if an error occurs.

print,'IF loop'

IF NOT (nparams EQ 3 OR nparams EQ 5 OR nparams EQ 7) THEN BEGIN
    message,'Incorrect number of arguments!','continue
    message,'Syntax: NGP, VALUE, POSX, NX[, POSY, NY, POSZ, NZ,' + $
        '/AVERAGE, /WRAPAROUND, /NO_MESSAGE]'
ENDIF

IF (nrsamples NE n_elements(posx)) OR $
    (dim GE 2 AND nrsamples NE n_elements(posy)) OR $
    (dim EQ 3 AND nrsamples NE n_elements(posz)) THEN $
    message,'Input arrays must have the same dimensions!'

IF NOT keyword_set(no_message) THEN $
    print,'Interpolating ' + strtrim(string(nrsamples,format='(i10)'),1)
$
    + ' samples to ' + strtrim(string(nxny*nz,format='(i10)'),1) + $
    ' grid points using NGP...'

; Compute nearest grid points.

IF keyword_set(wraparound) THEN BEGIN
    ; Coordinates of nearest grid point (ngp).
    ngx=fix(posx+0.5)
    ; Periodic boundary conditions.
    bad=where(ngx EQ nx,count)
    IF count NE 0 THEN ngx[bad]=0
    IF dim GE 2 THEN BEGIN
        ngy=fix(posy+0.5)
        bad=where(ngy EQ ny,count)
        IF count NE 0 THEN ngy[bad]=0
        IF dim EQ 3 THEN BEGIN
            ngz=fix(posz+0.5)
            bad=where(ngz EQ nz,count)

```

```

        IF count NE 0 THEN ngz[bad]=0
    ENDIF
ENDIF
bad=0 ; Free memory.
ENDIF ELSE BEGIN
    ; Coordinates of nearest grid point (ngp).
    ngx=fix(posx)
    IF dim GE 2 THEN BEGIN
        ngy=fix(posy)
        IF dim EQ 3 THEN ngz=fix(posz)
    ENDIF
ENDELSE

; Indices of grid points to which samples are assigned.
CASE dim OF
    1: index=temporary(ngx)
    2: index=temporary(ngx)+temporary(ngy)*nx
    3: index=temporary(ngx)+temporary(ngy)*nx+temporary(ngz)*nxny
ENDCASE

; Interpolate samples to grid.

field=fltarr(nx,ny,nz)

FOR i=0I,nrsamples-1I DO field[index[i]]=field[index[i]]+value[i]

; Compute weighted average.

IF keyword_set(average) THEN BEGIN
    ; Number of samples per grid point.
    frequency=histogram(temporary(index),min=0,max=nxny*nz-1I)

    ; Normalize.
    good=where(frequency NE 0,nrgood)
    field[good]=temporary(field[good])/temporary(frequency[good] )
ENDIF

print,field

close,value

END ; End of function ngp.

```

Error: NCDF_ATTNAME: Expression must be a scalar in this context: FID.
 % Execution halted at: NGP 17 /home2/shambhu/ngp.pro
