Subject: Re: Array extraction, multiple slices from paired values
Posted by Andrew[2] on Sat, 05 Feb 2005 04:12:58 GMT
View Forum Message <> Reply to Message

Hopefully my reply is what you meant,

you can extract the data at the points you want by doing the following

new_data=subdata[loc[0,*],loc[1,*],*]

This would give you a 4x4xt array for new_data that is sampled at the
locations specified in LOC

Cheers
Andrew

Subject: Re: Array extraction, multiple slices from paired values
Posted by MarioIncandenza on Sat, 05 Feb 2005 23:01:31 GMT
View Forum Message <> Reply to Message

Not what I wanted. See, each row of LOC is an XY pair corresponding to
a location. So the 4x4xt output from the command in your post gives me
the 4 time series I want, plus 12 others I don't want, and doesn't get
me any closer to extracting them. I need to get 2-dimensional output.
In loop form, this would work like this:
WHAT_I_WANT = fltarr(NLOC,NT)
for I=0,NLOC-1 do WHAT_I_WANT[I,*]=SUBDATA[LOC[0,I],LOC[1,I],*]

Which works great, but that step is currently 75% of the total
processor time for my script, which seems excessive, since it's not a
processing step at all but a simple subset.

Subject: Re: Array extraction, multiple slices from paired values
Posted by Chris Lee on Sun, 06 Feb 2005 12:43:48 GMT
View Forum Message <> Reply to Message

In article <1107644491.108963.151620@f14g2000cwb.googlegroups.com>,
"MarioIncandenza" <ejhyer@gmail.com> wrote:

> Not what I wanted. See, each row of LOC is an XY pair corresponding to a
> location. So the 4x4xt output from the command in your post gives me the
> 4 time series I want, plus 12 others I don't want, and doesn't get me
> any closer to extracting them. I need to get 2-dimensional output. In
> loop form, this would work like this: WHAT_I_WANT = fltarr(NLOC,NT)

> for I=0,NLOC-1 do WHAT_I_WANT[I,*]=SUBDATA[LOC[0,I],LOC[1,I],*] Which
> works great, but that step is currently 75% of the total processor time
> for my script, which seems excessive, since it's not a processing step
> at all but a simple subset.

Hi,

If you want to access individual elements you have to explicitly index
all of the dimensions, to get the element pairs you want you need to
index the array as...

```
; data(nx, ny, nz)
; x(nr) & y(nr)
; res = data[x#replicate(1,nz), y#replicate(1,nz), replicate(1,nr)#lindgen(nz)]
```

The following shows it working with data(100,100,100) and x(100), y(100).
For these numbers you get a factor of 2 improvement. If you try and
subset data from an array with more than three dimensions, REPLICATE
won't work and you need REFORM & REBIN. (The 'i' loops are simply for
timing purposes)

```
nx=100 & ny=100 & nz=100 & nr=1000
data=randomu(100,nx,ny,nz)
x=long(randomu(200,nr)*nx)
y=long(randomu(300,nr)*ny)

res=fltarr(nr,nz)

t1=systime(1)
for i=0, 1000 do for j=0, nr-1 do res[j,*]=data[x[j],y[j],*]
t2=systime(1)



rz=replicate(1L,nz)
rr=replicate(1L,nr)
lz=lindgen(nz)
t3=systime(1)
for i=0, 1000 do res2=data[x#rz,y#rz,rr#lz]
t4=systime(1)

print, "Loop method: ",t2-t1
print, "Explicit index: ",t4-t3
```

;Chris