Subject: improve calculation time
Posted by g.nacarts on Tue, 08 Jul 2014 18:29:57 GMT
View Forum Message <> Reply to Message

Hello

I am trying to improve my code in order to make it faster. A part of my code is the following

```
DIM = make_array(301,301,/DOUBLE)
pval = 0

for i=0, 300 do begin
  for j=0, 300 do begin
    x = round(i + Bx[i, j])
    y =    round(j + By[i, j])
         if (x lt 301) && (y lt 301) && (x ge 0) && (y ge 0) then begin
             DIM[i,j] = SIM[x, y]
         endif
    pval= pval + (DIM[i,j] - TIM[i, j])^2.
   endfor
endfor
```

Because I repeat the above procedure a lot of times it takes time to run. I noticed that the DIM array it's creating in every single iteration and I supposed that it takes time. I was wondering if I can write it in a different way in order to overwrite it in each iteration instead of delete it and create it all the time. To overwrite it I need to add it into the loop?

Regards,
Gina

Subject: Re: improve calculation time
Posted by Matthew Argall on Tue, 08 Jul 2014 20:10:08 GMT
View Forum Message <> Reply to Message

On Tuesday, July 8, 2014 2:29:57 PM UTC-4, g.na...@gmail.com wrote:

```
>  Hello
>
>
>
>  I am trying to improve my code in order to make it faster. A part of my code is the following
>
>
>
>  DIM = make_array(301,301,/DOUBLE)
>
>  pval = 0
>
```

```
>
>
>  for i=0, 300 do begin
>
>    for j=0, 300 do begin
>
>      x = round(i + Bx[i, j])
>
>      y =    round(j + By[i, j])
>
>           if (x lt 301) && (y lt 301) && (x ge 0) && (y ge 0) then begin
>
>               DIM[i,j] = SIM[x, y]
>
>           endif
>
>      pval= pval + (DIM[i,j] - TIM[i, j])^2.
>
>     endfor
>
>  endfor
>
>
>
```

> Because I repeat the above procedure a lot of times it takes time to run. I noticed that the DIM array it's creating in every single iteration and I supposed that it takes time. I was wondering if I can write it in a different way in order to overwrite it in each iteration instead of delete it and create it all the time. To overwrite it I need to add it into the loop?

>
>
>
> Regards,
>
> Gina

I *think* this is what you want. Make sure to double check...

```
SIM = randomu(1, 301, 301)
TIM = randomu(8, 301, 301)
DIM = make_array(301,301,/DOUBLE)
pval = 0
Bx = randomu(5, 301, 301)
By = randomu(2, 301, 301)
x = round(rebin(lindgen(1,301), 301, 301) + Bx)
y = round(transpose(rebin(lindgen(1,301), 301, 301)) + Bx)
```

```
i_sim = where( (x lt 301) and (y lt 301) and (x ge 0) and (y ge 0), n_sim )
if n_sim gt 0 then DIM[i_sim] = SIM[i_sim]

pval = total( (DIM - TIM)^2 )
```

Subject: Re: improve calculation time
Posted by g.nacarts on Wed, 09 Jul 2014 15:38:48 GMT
View Forum Message <> Reply to Message

This part of the code it doesn't make sense to me.

```
i_sim = where( (x lt 301) and (y lt 301) and (x ge 0) and (y ge 0), n_sim )
if n_sim gt 0 then DIM[i_sim] = SIM[i_sim]
```

Does the i_sim corresponds to SIM that I wrote or it's something different?

Subject: Re: improve calculation time
Posted by Matthew Argall on Wed, 09 Jul 2014 15:57:53 GMT
View Forum Message <> Reply to Message

> Does the i_sim corresponds to SIM that I wrote or it's something different?

i_sim is the index values that correspond to 0 <= x <= 300 and 0 <= y <= 300. Assuming x and y form a grid that define the locations of the points in SIM, i_sim is the indices within SIM that fall within your grid.

In other words SIM[i_sim] are the values of SIM that fall within the grid points 0 <= x <= 300 and 0 <= y <= 300.

Subject: Re: improve calculation time
Posted by g.nacarts on Thu, 10 Jul 2014 09:36:14 GMT
View Forum Message <> Reply to Message

I tried to run the code you suggest me and I always get error when I run it. I notice something but I am not sure if this is the problem.

```
In my code the x and y were: X LONG  =   383
                             Y LONG  =   383
In your code the x and y are:X LONG  = Array[384, 384]
                             Y LONG  = Array[384, 384]
```

The error I got is the following:
Expression must be a scalar or 1 element array in this context: <BYTE   Array[384, 384]>.

Subject: Re: improve calculation time
Posted by Matthew Argall on Thu, 10 Jul 2014 12:45:59 GMT
View Forum Message <> Reply to Message

When I copy and paste what I wrote into IDL, I do not get any errors, so you will have to provide more details.

Also, in my example, I defined X and Y to have 301 elements, not 383. I did this because in your example DIM is 301x301 and both loops go from 0 to 300.

Subject: Re: improve calculation time
Posted by g.nacarts on Thu, 10 Jul 2014 13:33:43 GMT
View Forum Message <> Reply to Message

I do not get any errors either when I copy and paste what you wrote into IDL. Once I wrote them into my program I got the error. What I wrote is a function that I call afterwards in my procedure and there the error came up. The think that I noticed it make sense to be part of the problem?

X LONG =   300 and   X LONG = Array[301, 301] are the same?

I just change the dimensions of my array before. The above are the original one

Subject: Re: improve calculation time
Posted by Matthew Argall on Thu, 10 Jul 2014 13:46:03 GMT
View Forum Message <> Reply to Message

> I just change the dimensions of my array before. The above are the original one

Yes, you will have to alter my example to fit your data.

In my example, X is originally a 301 element array:

```
x = indgen(1,301)
```

In order to add X to Bx and to access SIM and DIM (which are 301x301 arrays), I converted X to a 301x301 array. This allowed me to eliminate FOR loops.

```
x = lindgen(1,301)
x = round(rebin(x, 301, 301) + Bx)
```

The error that you mention usually is a result of comparing an array to a scalar

```
IDL> if bytarr(348,348) eq 0 then print, 'No Error'
% Expression must be a scalar or 1 element array in this context: <BYTE      Array[348, 348]>.
```