Subject: Re: Optimization Question: Sum at each element of array
Posted by Brian Daniel on Tue, 15 Feb 2011 13:09:56 GMT
View Forum Message <> Reply to Message

You're using the for loop to count over regression iterations.
Without knowing more about myfunc(), it's not clear to me that you'll
be able to do this with only array operations.

Regards,
Brian

On Feb 15, 12:58 am, Charles Steinhardt <charles.steinha...@ipmu.jp>
wrote:
> Hello,
>
> I'm trying to optimize the following:
>
> for i=0, 100 do begin
> y = y + myfunc(x, x + sigma * (i-50)/10.0, P[2]*myfunc2(x + sigma
> * (i-50)/10.0), sigma)
> endfor
>
>
>
> -Charles

Subject: Re: Optimization Question: Sum at each element of array
Posted by Jeremy Bailin on Tue, 15 Feb 2011 14:31:29 GMT
View Forum Message <> Reply to Message

On Tuesday, February 15, 2011 12:58:02 AM UTC-5, Charles Steinhardt wrote:
> Hello,
>
> I'm trying to optimize the following:
>
> for i=0, 100 do begin
> y = y + myfunc(x, x + sigma * (i-50)/10.0, P[2]*myfunc2(x + sigma
> * (i-50)/10.0), sigma)
> endfor
>
> Here, x, y, and sigma are arrays of the same cardinality. I know the
> for loop is slow in IDL compared to array operations, but I'm having
> problems finding a faster way to do this. Is it really faster to make
> an array of findgen(101) and then do some sort of summation over
> that?
>
> I'm hoping somebody has run into this before - I'd appreciate any

> advice you have!
>
> Thank you,
>
> -Charles

It depends on myfunc and myfunc2. It looks like myfunc currently takes 3 arguments of the same cardinality, and myfunc2 takes one argument? If they can be written in such a way that they can take an extra dimension (of length 101, in this case), then yes, it would be faster to just do a total. So, for example:

```
xsize = size(x, /dimen)
ndimen = n_elements(xsize)
fullsize = [xsize,101]
xfull = rebin(x, fullsize, /sample)
sigmafull = rebin(sigma, fullsize, /sample)
ifull = rebin( (findgen(101)-50)/10., fullsize, /sample)

y = total( myfunc(xfull, xfull + sigmafull * ifull, $
  P[2]*myfunc2(xfull + sigmafull * ifull), sigmafull), ndimen )
```

-Jeremy.

Subject: Re: Optimization Question: Sum at each element of array
Posted by James[2] on Tue, 15 Feb 2011 20:15:30 GMT
View Forum Message <> Reply to Message

On Feb 14, 9:58 pm, Charles Steinhardt <charles.steinha...@ipmu.jp>
wrote:
> Hello,
>
> I'm trying to optimize the following:
>
> for i=0, 100 do begin
> y = y + myfunc(x, x + sigma * (i-50)/10.0, P[2]*myfunc2(x + sigma
> * (i-50)/10.0), sigma)
> endfor
>
> Here, x, y, and sigma are arrays of the same cardinality. I know the
> for loop is slow in IDL compared to array operations, but I'm having
> problems finding a faster way to do this. Is it really faster to make
> an array of findgen(101) and then do some sort of summation over
> that?
>
> I'm hoping somebody has run into this before - I'd appreciate any
> advice you have!
>

> Thank you,
>
> -Charles

One simple optimization step is to replace:

```
y = y + (blahblahblah...)
```

with

```
y += (blahblahblah...)
```

This avoids making IDL acquire memory each iteration to store the result of y + (blahblahblah...).
