

---

Subject: Re: Efficient Programming  
Posted by [Leonard Kramer](#) on Sun, 02 Jul 1995 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

dean@phobos.cira.colostate.edu wrote:

```
>
>   What would be the most efficient way to get this to work in IDL. This is
> obvious slow with the FOR DO BEGIN.....
>
>
> Kelly Dean
> CSU/CIRA
>
> =====
=====
>
> nx = 1024
> ny = 768
> North_bound = 3429
> West_bound = 9249
> line_res = 4
> elem_res = 8
>   LINarr = LONarr(nx*ny)
>   PIXarr = LONarr(nx*ny)
>   k = 0L
> *   FOR y = 0, ny-1 DO BEGIN
> *     FOR x = 0, nx-1 DO BEGIN
> *       LINarr(k) = North_bound + ( y * line_res)
> *       PIXarr(k) = West_bound + ( x * elem_res)
> *       k = k + 1L
> *     ENDFOR
> *   ENDFOR
>
```

You see those lines I marked with an \* above. Replace them with:

---

Subject: Re: Efficient Programming  
Posted by [Leonard Kramer](#) on Sun, 02 Jul 1995 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

dean@phobos.cira.colostate.edu wrote:

```
>
>   What would be the most efficient way to get this to work in IDL. This is
> obvious slow with the FOR DO BEGIN.....
>
>
```

```

> Kelly Dean
> CSU/CIRA
>
> =====
=====
>
> nx = 1024
> ny = 768
> North_bound = 3429
> West_bound = 9249
> line_res = 4
> elem_res = 8
>   LINarr = LONarr(nx*ny)
>   PIXarr = LONarr(nx*ny)
>   k = 0L
> *   FOR y = 0, ny-1 DO BEGIN
> *     FOR x = 0, nx-1 DO BEGIN
> *       LINarr(k) = North_bound + ( y * line_res)
> *       PIXarr(k) = West_bound + ( x * elem_res)
> *       k = k + 1L
> *     ENDFOR
> *   ENDFOR
>

```

You see those lines I marked with an \* above. Replace them with:

Subject: Re: Efficient Programming  
 Posted by [hebeling](#) on Sun, 02 Jul 1995 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

In article <3t5b5h\$2ts4@yuma.ACNS.ColoState.EDU>, dean@phobos.cira.colostate.edu writes:

```

>
>   What would be the most efficient way to get this to work in IDL. This is
>   obvious slow with the FOR DO BEGIN.....
>
>
>   Kelly Dean
>   CSU/CIRA
>
>   =====
=====
>
> nx = 1024
> ny = 768
> North_bound = 3429
> West_bound = 9249

```

```

> line_res = 4
> elem_res = 8
>   LINarr = LONarr(nx*ny)
>   PIXarr = LONarr(nx*ny)
>   k = 0L
>   FOR y = 0, ny-1 DO BEGIN
>     FOR x = 0, nx-1 DO BEGIN
>       LINarr(k) = North_bound + ( y * line_res)
>       PIXarr(k) = West_bound + ( x * elem_res)
>       k = k + 1L
>     ENDFOR
>   ENDFOR
>

```

Try this:

```

nx = 1024L           ;ought to be LONGWORD
ny = 768L           ;ought to be LONGWORD
North_bound = 3429
West_bound = 9249
line_res = 4
elem_res = 8

```

```

linarr = replicate(1,nx) # (lindgen(ny)*line_res) + north_bound
pixarr = (lindgen(nx)*elem_res) # replicate(1,ny) + west_bound

```

```

linarr = reform(linarr,nx*ny) ;force array to be 1-dim
pixarr = reform(pixarr,nx*ny) ;force array to be 1-dim

```

That's at least an order of magnitude faster than the double FOR loop.

Harald Ebeling

```

=====
University of Cambridge
Institute of Astronomy
Madingley Road
Cambridge CB3 0HA
United Kingdom

```

ebeling@ast.cam.ac.uk (Ignore the address in the header - I'm only visiting...)

---