
Subject: Arrays with complex index ?

Posted by [Henrik E. Nilsen](#) on Sat, 04 Dec 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello all,

Anyone know if there's a way to reference a floating point array using a complex number?

what I would like to do is something like:

```
cnum=complex(5,5) & cnum2=complex(2,3) & fnum=10.7
arr=fltarr(cnum)
arr[cnum2]=fnum
```

and not:

```
arr=fltarr(5,5)
arr[float(cnum),imaginary(cnum)]=fnum
```

I suspect that this is not possible. Does anyone know how time consuming the 'float' and 'imaginary' commands are? (I'm using these statements at the core of a loop structure, and so any savings in time are important)

Thanks.

- Henrik

Henrik E. Nilsen Chairman, The Maidanak Foundation
Graduate student, astrophysics <http://www.maidanak.org>
University of Bergen, Norway Welhavens gate 61
E-mail: Henrik.Nilsen@fi.uib.no N-5006 Bergen, Norway

Subject: Re: Arrays with complex index ?

Posted by [Liam Gumley](#) on Sat, 04 Dec 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Henrik E. Nilsen wrote:

```
> Anyone know if there's a way to reference a floating point array using a
> complex number?
>
> what I would like to do is something like:
>
> cnum=complex(5,5) & cnum2=complex(2,3) & fnum=10.7
> arr=fltarr(cnum)
> arr[cnum2]=fnum
```

>
> and not:
>
> arr=fltarr(5,5)
> arr[float(cnum),imaginary(cnum)]=fnum
>
> I suspect that this is not possible. Does anyone know how time
> consuming the 'float' and 'imaginary' commands are? (I'm using these
> statements at the core of a loop structure, and so any savings in time are
> important)

Regarding the use of a complex variable as an array subscript, here's what the documentation says (Building IDL applications; Subscript Examples):

"Subscripts can be any type of vector or scalar expression. If a subscript expression is not integer, a longword integer copy is made and used to evaluate the subscript."

Thus the following statements are legal, but only the real part of the complex number is used to form the subscript:

```
IDL> arr = indgen(10)
IDL> index = complex(1.0, 2.0)
IDL> help, long(index)
<Expression> LONG = 1
IDL> help, arr[index]
<Expression> INT = 1
```

The second example you gave looks fine to me, e.g.

```
arr = fltarr(5,5)
arr[float(cnum),imaginary(cnum)] = fnum
```

I don't think there's any great time penalty incurred by using float() and imaginary(). You'll gain much more speed by finding a way to express your algorithm in array operations rather than loop operations.

Cheers,
Liam.
<http://cimss.ssec.wisc.edu/~gumley>

Subject: Re: Arrays with complex index ?
Posted by [wbiagiot](#) on Mon, 06 Dec 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <82bor4\$036\$1@news.doit.wisc.edu>,

Henrik,

If you are interesting in looking at the timing of IDL routines (or yours for that matter), IDL5.2 and upward has a pretty good execution profiler (run->profile). This nifty feature gives you the exact execution time and count for every one of your selected routines. This will allow you to see where your program execution time is spent. Also, IDL routines like 'COMPLEX' or 'FIX' are what I call "mini-routines" and are so quick as to be inconsequential when looking at cumulative execution time. Liam was correct in stating that if you can perform your operations as array vectors that that is the best way to go by far.

Regards,

Bill B.

--

"They don't think it be like it is, but it do."

Oscar Gamble, NY Yankees

Sent via Deja.com <http://www.deja.com/>
Before you buy.
