Subject: Re: Reading 32-bit complex numbers in IDL (16-bit real / 16-bit imaginary)
Posted by Carsten Lechte on Sun, 07 Aug 2011 15:52:42 GMT
View Forum Message <> Reply to Message

On 08/07/2011 02:01 PM, Waqas A. Qazi wrote:
> 32-bit imaginary. How can I read the 16-bit real - 16-bit imaginary
> complex number in IDL?

This looks like half precision floats
< https://secure.wikimedia.org/wikipedia/en/wiki/Half_precisio n_floating-point_format>.
However, there are several variants. Do you have any description of your data file that mentions IEEE 754?

I am not aware of any programming language that supports this format natively. Maybe this link in the wikipedia article can help you:
<http://www.mathworks.com/matlabcentral/fileexchange/23173>

Else you would have to do the bit fiddling yourself to promote them to 32bit.

chl

Subject: Re: Reading 32-bit complex numbers in IDL (16-bit real / 16-bit imaginary)
Posted by Wout De Nolf on Sun, 07 Aug 2011 23:12:24 GMT
View Forum Message <> Reply to Message

On Sun, 7 Aug 2011 05:01:34 -0700 (PDT), "Waqas A. Qazi"
<waqastro@yahoo.com> wrote:

> Hi,
>
> I couldn't find a discussion of this specific problem on the group, so
> I am posting it here in hope of a solution.
>
> I have complex data in a binary file, with each value comprising of 4
> bytes (32 bits), such that the 32-bit complex number consists of 16-
> bit real and 16-bit imaginary. I could use define a complex array in
> IDL and then read the data using the readu command, however the
> "complex" format in IDL is a 2*32-bit definition, i.e. 32-bit real and
> 32-bit imaginary. How can I read the 16-bit real - 16-bit imaginary
> complex number in IDL?
>
>
> Thanks,
> Waqas.

Have a look at these routines:
http://tinyurl.com/43ca8sk

I made them once to understand floating point numbers. They are not optimized for speed. I added half precision for you and it is not hard to add any other precision if you like (like quadruple). An example for half precision:

integer='3555'x
float=BINARYTOFLOAT(integer,precision=0)

So you can use READU to read 16bit integers from your file, convert them to 32bit float with BINARYTOFLOAT and then pair them up to 32bit complex with COMPLEX.

If you have many numbers, this will be slow and BINARYTOFLOAT should be vectorized.

Hope it helps,
Wox

---

Subject: Re: Reading 32-bit complex numbers in IDL (16-bit real / 16-bit imaginary)
Posted by Waqas A. Qazi on Fri, 12 Aug 2011 13:15:35 GMT
View Forum Message <> Reply to Message

Hi chl and wox,

Thanks, I took up from the info both of you noted here, and did some reading on my own.

It seems the numbers are indeed half-precision floating point (never knew this was a data-type!), but in the data storage here, they are specified as integers and should be read as such. The data specification document states clearly that "samples are stored as 16 bit / 16 bit complex integer (4 bytes).

Wox, I understand the logic you have presented, and I will follow that, but have one question. After defining an intarr (16 bit) and reading 2-byte integers and later pairing them up for intput into the complex function, why do I have to convert from integer to float first? I think that the complex function should automatically convert them to single-precision float complex??

Thanks,
Waqas.

---

Subject: Re: Reading 32-bit complex numbers in IDL (16-bit real / 16-bit imaginary)
Posted by Wout De Nolf on Sat, 13 Aug 2011 12:26:58 GMT
View Forum Message <> Reply to Message

On Fri, 12 Aug 2011 06:15:35 -0700 (PDT), "Waqas A. Qazi"
<waqastro@yahoo.com> wrote:

> Hi chl and wox,
>
> Thanks, I took up from the info both of you noted here, and did some
> reading on my own.
>
> It seems the numbers are indeed half-precision floating point (never
> knew this was a data-type!), but in the data storage here, they are
> specified as integers and should be read as such. The data
> specification document states clearly that "samples are stored as 16
> bit / 16 bit complex integer (4 bytes).
>
> Wox, I understand the logic you have presented, and I will follow
> that, but have one question. After defining an intarr (16 bit) and
> reading 2-byte integers and later pairing them up for intput into the
> complex function, why do I have to convert from integer to float
> first? I think that the complex function should automatically convert
> them to single-precision float complex??
>
>
> Thanks,
> Waqas.

When I say "convert integer to float" I don't mean that you would
convert 10 to 10.0 for example. The value of the integer is not
concidered at all, it's the "bit-content" that is used. The integer is
just a bag of bits that represent a foating point number according to
the IEEE754 standard. That's why the function is call BINARYTOFLOAT
and not INTEGERTOFLOAT or something. In the example I gave:

```
IDL> integer='3555'x
IDL> f=binarytofloat(integer,precision=0)
IDL> print,integer
  13653
IDL> print,f
    0.333252
```

You can see that the integer value 13653 has nothing to do with
0.33325... However both numbers have the same binary representation,
namely

```
IDL> print,integer,format='(b016)'
```

0011010101010101

Note: Since IDL can't handle 16bit floats, the binary representation of f (32bit float) isn't the same anymore as that of 13653. The value of f is correct however: the value represented by 13653 under the half-precision IEEE 754 convention.

---