Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by thompson on Thu, 23 Jan 2003 23:38:50 GMT
View Forum Message <> Reply to Message

Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:

> Craig Markwardt wrote:
>> Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:
>>
>>
>>> How can I add the NEEDED d to get this:
>>>
>>> 2452662.305203d
>>>
>>> out of this:
>>> maxtime = jul2cal((data(0,maxgust_time)), /TO_STRING, /MDY)
>>>
>>> In my program (data(0,maxgust_time)) fetches 2452662.305203 out of the
>>> array, but if I don't add the d to the julian date it calculates the
>>> wrong time in the above statement.
>>
>>
>> You can use
>> double(data(0,maxgust_time)),
>> but the variable DATA should already be in double precision. At least
>> it should be if you expect 13 decimal digits of precision to be
>> maintained. When you type the number directly on the command line,
>> you probably do have to use the "D" to indicate double precision, but
>> you should not have to if the variable DATA is already double.
>>
>> Craig
>>

> I've checked my array a bit more and it seems as if there's something
> wrong with it. From the file I'm reading in with read-ascii I should
> have this:
> 2452662.499876 2.719500 6.216000 343.494000
> 955.793400 93.911600 -5.444307

> but the print, data in IDL shows:
> 2.45266e+06 2.71950 6.21600 343.494 955.793
> 93.9116 -5.44431

> If I replace the read-acsii with Reimar Bauers read_data_file I get:
> 2452662.5 2.7195000 6.2160000 343.49400
> 955.79340 93.911600 -5.4443070

> but as you see the julian date in the first element is wrong in both

> arrays. How can I do ensure that I get all digits inserted ?

I tried the following

```
IDL> a = 2452662.499876  ;Single precision
IDL> print,a
  2.45266e+06
IDL> a = 2452662.499876d ;Double precision
IDL> print,a
       2452662.5
```

It looks like read_data_file is reading the data correctly as double precision, while read_ascii is apparently reading everything into as single precision. I know that it looks like A is being rounded off in the second case, but that's only because of the default format being used for printing. If you use an explicit format, you can see more of the digits.

```
IDL> print,a,format='(F20.6)'
      2452662.499876
```

If, on the other hand, the data was read in as single precision, it really will be truncated.

```
IDL> a = 2452662.499876  ;Single precision
IDL> print,a,format='(F20.6)'
      2452662.500000
```

Bill Thompson
