Subject: number of decimals?
Posted by Y.T. on Thu, 09 Dec 2004 04:49:24 GMT
View Forum Message <> Reply to Message

I've almost decided that the following is impossible in IDL, but I figured I'd give it one more try...

I have a large dblarray of numbers -- say ten by ten million or so:

IDL> f = dblarr(10,1e7)
IDL> f = 1000.*randomn(seed,1e8)

(In reality these are real data values, not random numbers, of course).

I'd like to write these to a file (in the sense of a printf) with an accuracy of two digits behind the decimal point. I COULD do something like the following:

IDL> printf,unit,f,format='(10f9.2)'

but that'll introduce additional spaces wherever a number is smaller than 10000. So a typical line might look like this:

123.45 678.23 1.23 12345.67 ...
etc

But what I want is

123.45 678.23 1.23 12345.67 ...

What I'm trying to do would be written in C somewhat like this: "%.2f %.2f %.2f", i.e. a floating point number with two decimals.

The help tells me that I can specify the total width as zero to obtain a "natural width" but if I do a "f0.2" in the above example, I get things like

123.450000000 678.23000000 1.23000000 12345.6700000000 ...

I have looked into the %"... form of the 'format' keyword but that only supports "%w.n" forms, not "%.n" forms.

For now, I am doing something unspeakably ugly like this:

printf,strcompress(string(f,format='(10f9.2)'))

which is reasonably workable as long as f is less than about 1/4 of my

total memory -- otherwise the various conversion start swapping and all
hell breaks loose.

I understand that I could do this simply line by line with strcompress,
but that takes approximately a metric forever.

Is there some trick or something that allows me to write a number with
two decimals and "just the right number of digits" before the decimal
point?

Thanks in advance...
cordially

Y.T.

--
Remove YourClothes before you email me.

Subject: Re: number of decimals?
Posted by Chris Lee on Thu, 09 Dec 2004 08:56:50 GMT
View Forum Message <> Reply to Message

In article <1102567764.693437.32940@c13g2000cwb.googlegroups.com>,
"Unknown" <ytyourclothes@p.zapto.org> wrote:

> ...
> IDL> printf,unit,f,format='(10f9.2)'
> but that'll introduce additional spaces wherever a number is smaller
> than 10000. So a typical line might look like this: 123.45 678.23
> 1.23 12345.67 ... etc
> But what I want is
> 123.45 678.23 1.23 12345.67 ...
> What I'm trying to do would be written in C somewhat like this: "%.2f
> %.2f %.2f", i.e. a floating point number with two decimals.
> ...
> Y. T.

No tricks, but you can use the C format codes in IDL

f=randomn(seed, 10,10)
print, format='(10(%"%0.5f "))',f[0,*]

That will write out floating point numbers with 5 decimal places and a
single space between each number. Your emacs loving
friends will disown you for breaking their copy-rectangle-to-register,

but I think it's what you want.

Chris.

Subject: Re: number of decimals?
Posted by JD Smith on Thu, 09 Dec 2004 15:23:27 GMT
View Forum Message <> Reply to Message

On Thu, 2004-12-09 at 08:56 +0000, Christopher Lee wrote:
> In article <1102567764.693437.32940@c13g2000cwb.googlegroups.com>,
> "Unknown" <ytyourclothes@p.zapto.org> wrote:
>
>> ...
>> IDL> printf,unit,f,format='(10f9.2)'
>> but that'll introduce additional spaces wherever a number is smaller
>> than 10000. So a typical line might look like this: 123.45 678.23
>> 1.23 12345.67 ... etc
>> But what I want is
>> 123.45 678.23 1.23 12345.67 ...
>> What I'm trying to do would be written in C somewhat like this: "%.2f
>> %.2f %.2f", i.e. a floating point number with two decimals.
>> ...
>> Y. T.
>
> No tricks, but you can use the C format codes in IDL
>
> f=randomn(seed, 10,10)
> print, format='(10(%"%0.5f "))',f[0,*]
>
> That will write out floating point numbers with 5 decimal places and a
> single space between each number. Your emacs loving
> friends will disown you for breaking their copy-rectangle-to-register,
> but I think it's what you want.

For most versions of IDL, your example doesn't work works because 6 is
the default number of decimal places for "natural length" floats, which
is all the "%0.5" requests (i.e. you could have written %0.100). It's
the same with normal format codes, in IDL 6.0:

IDL> print,FORMAT='(2(F0.2,:," "))',!PI,!PI^4
3.141593 97.409103

Starting with IDL6.1, IDL finally respects width "0" formats:

IDL 6.1:

```
IDL> print,FORMAT='(2(F0.2,:," "))',!PI,!PI^4
3.14 97.41
```

I hadn't appreciated that you could mix FORTRAN style and C-style format codes, which could be very useful.

JD
