

---

Subject: what am I doing wrong?

Posted by [babaka](#) on Fri, 17 Feb 1995 00:09:36 GMT

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

---

Dear Netters,

I'm trying to plot the function:

$$f(x)=x*\log(x/w)-x \quad (x>0,w>0)$$

using PV~WAVE.

I know that this function has a minimum at  $x=w$ .

To see this, note that

$$f'(x)=\log(x/w)$$

Setting  $f'(x)=0$  gives  $x=w$ . On the other hand:

$$f''(x)=1/x > 0 \text{ for all } x$$

Therefore  $x=w$  is the minimum.

THE PROBLEM IS that when I plot this function in PV~WAVE the minimum appears at another point.

I'm using the following set of commands:

```
WAVE> w=5.0
WAVE> x=INDGEN(400)
WAVE> x=x+0.1 + 0.01 ; x goes from 0.01 to 40.01
WAVE> fx=x*ALOG10(x/w)-x
WAVE> PLOT,x,fx
```

I would appreciate if you would tell me what I'm doing wrong?

Regards,  
Babak

--

Babak A. Ardekani [babaka@atom.ansto.gov.au](mailto:babaka@atom.ansto.gov.au)  
Dept. of Nuclear Medicine phone: +61-2-516-8011  
Royal Prince Alfred Hospital fax: +61-2-550-5172

---

Subject: Re: what am I doing wrong?

Posted by [thompson](#) on Mon, 05 Oct 1998 07:00:00 GMT

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

---

Bruce Bowler <[bbowler@bigelow.org](mailto:bbowler@bigelow.org)> writes:

> a couple of things... It is a top level script but there are no  
> continuation lines that I can see... (true, my parser is \*quite\*  
> different than IDL's :-)

> second, what happens depends on how I run it, which I find quite strange  
> but then again, I'm a relative newbie at IDL...

> I've made some changes, while trying to figure things out. Here is the  
> new script in it's entirety... (note I realize some of this [especially  
> the case statement] aren't the best way to do things but like I said,  
> I'm a newbie and I'm trying some things out and this is stripped down  
> procedure anyway...)

(stuff deleted)

> if I run it as follows...

> # idl  
> idl> .run 6953

> it works fine...

> if I run it like

> # idl 6953

> the results are

(stuff deleted)

> unfortunately, because of the way I need to interact with some other  
> stuff, I really need to run it the second way. What can I do to make it  
> work like the first way? I don't think precompiling will work either as  
> I generate the file "on the fly".

If you're using Unix, then what you need to do is to create a file like the following:

```
#!/bin/csh
```

```
setenv MY_IDL_SCRIPT $argv
idl << done
.run $MY_IDL_SCRIPT
done
```

Use "chmod +x" to make the file executable, and then you can use this to execute whatever IDL script you want.

William Thompson

---

---

Subject: Re: what am I doing wrong?  
Posted by [Bruce Bowler](#) on Mon, 05 Oct 1998 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

a couple of things... It is a top level script but there are no continuation lines that I can see... (true, my parser is \*quite\* different than IDL's :-)

second, what happens depends on how I run it, which I find quite strange but then again, I'm a relative newbie at IDL...

I've made some changes, while trying to figure things out. Here is the new script in it's entirety... (note I realize some of this [especially the case statement] aren't the best way to do things but like I said, I'm a newbie and I'm trying some things out and this is stripped down procedure anyway...)

```
!path = !path + ":" + getenv("SEADAS") + "/idl_lib" +
"/disk2/seadas_data"
```

```
inputFile = "S1998228192646.L1A_HNSG"
```

```
baseDir = strmid(inputFile,0,rstrpos(inputFile,"/")+1)
baseFile = strmid(inputFile,rstrpos(inputFile,"/")+1,999)
passId = strmid(baseFile,0,rstrpos(baseFile,"."))
datePart = strmid(passId,0,8)
ozoneFile = datePart + '12_EPTOMS.OZONE'
temp = size(findfile(ozonefile))
help, temp
print, temp
defOzone = 0
CASE 1 OF
  (temp[0] EQ 1): BEGIN
    ozoneFile = '$SDSDATA/CLIMATOLOGY.OZONE'
    defOzone = 1
    print, 'using default ozone data'
  END
```

```

(temp[0] NE 1): BEGIN
  defOzone = 0
  print, 'ozone data found'
  END
ELSE: print , 'things are really wacked'
ENDCASE
print, "after case"
end

```

if I run it as follows...

```

# idl
idl> .run 6953

```

it works fine...

if I run it like

```

# idl 6953

```

the results are

```

> For basic information, enter "IDLInfo" at the IDL> prompt.
>
> % Compiled module: RSTRPOS.
> % Compiled module: REVERSE.
> TEMP      LONG  = Array[4]
>   1      1      7      1
> % Case statement found no matches.
> % Execution halted at: $MAIN$
>
>   (temp[0] EQ 1): BEGIN
>     ^
> % Syntax error.
> At: 6953, Line 15
> using default ozone data
>
>   END
>   ^
> % Syntax error.
> At: 6953, Line 19
>
>   (temp[0] NE 1): BEGIN
>     ^
> % Syntax error.
> At: 6953, Line 20
> ozone data found
>

```

```

>     END
>     ^
> % Syntax error.
> At: 6953, Line 23
>
>     ELSE: print ,'things are really wacked'
>     ^
> % Syntax error.
> At: 6953, Line 24
>
>     ENDCASE
>     ^
> % Syntax error.
> At: 6953, Line 25
> after case
>
> end
> ^
> % Syntax error.
> At: 6953, Line 27

```

unfortunately, because of the way I need to interact with some other stuff, I really need to run it the second way. What can I do to make it work like the first way? I don't think precompiling will work either as I generate the file "on the fly".

Jeffrey Newmark wrote:

```

>
> Hi Bruce,
>
> Is this a section of a procedure or a top level script? If it is top
> level then you need
> $ to continue lines, otherwise, IDL compiles each line by itself.
>
> cheers,
> Jeff Newmark
>
> Bruce Bowler wrote:
>
>> I have a script that does...
>>
>>> anydata, subtemp, 45.0, 40.0, -71.5, -65.0, inbounds
>>>
>>> print, subtemp
>>> print, inbounds
>>>
>>> if (inbounds eq 1) then begin
>>>   openw, tarunit, 'tarfile.include', /GET_LUN

```

```
>>> printf, tarunit, l1a
>>> printf, tarunit, ozonefile
>>> printf, tarunit, metFile1
>>> printf, tarunit, metFile2
>>> printf, tarunit, 'SEAWIFS_SENSOR_CAL.TBL'
>>> close, tarunit
>>> endif
>>
>> When I run it, I get
>>
>>> S1998228192646.SST
>>>    0
>>
>> and tarfile.include is created, which since inbounds is 0, shouldn't
>> happen... can anyone explain why it's not working or some additional
>> diagnostics that might point to the cause?
>>
>> TIA,
>> Bruce
>
> -----
> -----
>
> Dr. Jeffrey Newmark
> SOHO - EIT Team Scientist
>
> voice: (301) 286-3163           Code 682.3
> email: newmark@eitv2.nascom.nasa.gov   GSFC/NASA
>     newmark@gsfc.nasa.gov           Greenbelt, MD 20771
```

---

---

Subject: Re: what am I doing wrong?  
Posted by [Jeffrey Newmark](#) on Mon, 05 Oct 1998 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Bruce,

Is this a section of a procedure or a top level script? If it is top level then you need \$ to continue lines, otherwise, IDL compiles each line by itself.

cheers,  
Jeff Newmark

Bruce Bowler wrote:

> I have a script that does...

```
>
>> anydata, subtemp, 45.0, 40.0, -71.5, -65.0, inbounds
>>
>> print, subtemp
>> print, inbounds
>>
>> if (inbounds eq 1) then begin
>>   openw, tarunit, 'tarfile.include', /GET_LUN
>>   printf, tarunit, l1a
>>   printf, tarunit, ozonefile
>>   printf, tarunit, metFile1
>>   printf, tarunit, metFile2
>>   printf, tarunit, 'SEAWIFS_SENSOR_CAL.TBL'
>>   close, tarunit
>> endif
>
> When I run it, I get
>
>> S1998228192646.SST
>>   0
>
> and tarfile.include is created, which since inbounds is 0, shouldn't
> happen... can anyone explain why it's not working or some additional
> diagnostics that might point to the cause?
>
> TIA,
> Bruce
```

-----

Dr. Jeffrey Newmark  
SOHO - EIT Team Scientist

voice: (301) 286-3163                      Code 682.3  
email: newmark@eitv2.nascom.nasa.gov      GSFC/NASA  
       newmark@gsfc.nasa.gov                Greenbelt, MD 20771

### File Attachments

1) [mail.inc](#), downloaded 116 times

---

---

Subject: Re: what am I doing wrong?  
Posted by [thompson](#) on Tue, 06 Oct 1998 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Bruce Bowler <[bbowler@bigelow.org](mailto:bbowler@bigelow.org)> writes:

> William Thompson wrote:

>>

>> Bruce Bowler <[bbowler@bigelow.org](mailto:bbowler@bigelow.org)> writes:

>>

> (stuff deleted)

>>> if I run it as follows...

>>

>>> # idl

>>> idl> .run 6953

>>

>>> it works fine...

>>

>>> if I run it like

>>

>>> # idl 6953

>>

>>> the results are

>>

>> (stuff deleted)

>>

>>> unfortunately, because of the way I need to interact with some other

>>> stuff, I really need to run it the second way. What can I do to make it

>>> work like the first way? I don't think precompiling will work either as

>>> I generate the file "on the fly".

>>

>> If you're using Unix, then what you need to do is to create a file like the

>> following:

>>

>> #!/bin/csh

>> setenv MY\_IDL\_SCRIPT \$argv

>> idl << done

>> .run \$MY\_IDL\_SCRIPT

>> done

>>

>> Use "chmod +x" to make the file executable, and then you can use this to

>> execute whatever IDL script you want.

> Unfortunately, as I said before, that won't work in my case. I am

> passing this to a third party package that invokes it as

> idl xxx

> I have no control over that.

There are a couple of ways you can accomplish this. One way, as already

pointed out by Stein Vidar, is to end each line within a loop with the characters "& \$", so that IDL treats it all as one line. As I recall, this works unless the loop is very large.

Another way is to use two procedure files. One file will contain the main-level program that you wish to run. For example, let's say you save it in a file called myprogram.idl. The second file will then contain the single line

```
.run myprogram
```

You can then pass the name of this second file to IDL. For example, if the second file is named run\_myprogram.pro, then the statement

```
idl run_myprogram
```

will execute the main-level procedure myprogram. It would be just the same as if you had typed ".run myprogram" at the command prompt.

William Thompson

---

---

Subject: Re: what am I doing wrong?  
Posted by [steinhh](#) on Tue, 06 Oct 1998 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <361A08FE.D4EEB7A1@bigelow.org> Bruce Bowler <bbowler@bigelow.org> writes:

> Unfortunately, as I said before, that won't work in my case. I am  
> passing this to a third party package that invokes it as  
>  
> idl xxx  
>  
> I have no control over that.

Then you have to put "&\$" on the end of each line in the IF .. THEN BEGIN ... END block. This causes the lines to be interpreted as if they were entered as a single line, with "&" inbetween each statement.

Regards,

Stein Vidar

---

---

Subject: Re: what am I doing wrong?  
Posted by [Bruce Bowler](#) on Tue, 06 Oct 1998 07:00:00 GMT

---

William Thompson wrote:

>  
> Bruce Bowler <[bbowler@bigelow.org](mailto:bbowler@bigelow.org)> writes:  
>

(stuff deleted)

>> if I run it as follows...

>  
>> # idl  
>> idl> .run 6953

>> it works fine...

>> if I run it like

>> # idl 6953

>> the results are

> (stuff deleted)

>> unfortunately, because of the way I need to interact with some other  
>> stuff, I really need to run it the second way. What can I do to make it  
>> work like the first way? I don't think precompiling will work either as  
>> I generate the file "on the fly".

> If you're using Unix, then what you need to do is to create a file like the  
> following:

```
> #!/bin/csh  
> setenv MY_IDL_SCRIPT $argv  
> idl << done  
> .run $MY_IDL_SCRIPT  
> done
```

> Use "chmod +x" to make the file executable, and then you can use this to  
> execute whatever IDL script you want.

Unfortunately, as I said before, that won't work in my case. I am  
passing this to a third party package that invokes it as

```
idl xxx
```

I have no control over that.

---

---

Subject: Re: what am I doing wrong?

Posted by [Bruce Bowler](#) on Tue, 06 Oct 1998 07:00:00 GMT

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

---

William Thompson wrote:

> Another way is to use two procedure files. One file will contain the  
> main-level program that you wish to run. For example, let's say you save it in  
> a file called myprogram.idl. The second file will then contain the single line  
>  
> .run myprogram  
>  
> You can then pass the name of this second file to IDL. For example, if the  
> second file is named run\_myprogram.pro, then the statement  
>  
> idl run\_myprogram  
>  
> will execute the main-level procedure myprogram. It would be just the same as  
> if you had typed ".run myprogram" at the command prompt.

Hmmmmm... I like it...

---