Subject: Re: Obtaining exponent from a scientific format number
Posted by Michael Baca on Wed, 17 Apr 2002 16:50:22 GMT
View Forum Message <> Reply to Message

Well, part of this will depend on the format of the input number. The easy way to get the exponent of the number is:

value=6.8977653d-18 ; the d will assign the value to a double
exp_num = floor(alog10(value)) ; works for both positive and negative exponents

Then you just need to decide you starting position. In your message, you moved the decimal place 7 positions to the right so the quick method would be:

exp_num = exp_num - 7 ; moves the decimal 7 places to the right
base_val = floor(value/10.0d^exp_num)

Not fancy, but should do the job. The good news is that this will also work on an array. So if you have an array of value, you can use the exact same code and takes minimal effort. Much more efficient than looping over values and counting positions in a string.

The biggest issue is handling the initial value. You may have problems keeping track of the 7+ decimal places if not defined properly. My inital attempts used 6.8977653e-18 and would consistently lose the 3 in the base number unless I defined the number as above (use d not e). Besides, recalling my days as a physics TA, do you really know the number that well. :)

Mike

"Juan I. Cicuendez" <jicicuendez@gmv.es> wrote in message
news:4b04c513.0204170601.64440c58@posting.google.com...
> Hi all,
>
> I have a problem and I am looking for a faster solution:
>
> I have a scientific formatted number (e.g. 6.8977653e-18) and I have
> to split the exponent and number into two parts like this:
> long:68977653
> exp:-25
> where the first factor has to be a long number and the second the
> exponent.
> The exponential factors can algo change.
>
> The solution I came up is to turn the number into strings and then
> byte(mystring), obtaining the position of '.' and 'e' and then back to

> numbers. This seems to be quite slow and since I have a large number
> of data I don't think is very efficient.
>
> Thank you very much in advance, I would appreciate any hints.
>
> Juan

Subject: Re: Obtaining exponent from a scientific format number
Posted by Dick Jackson on Wed, 17 Apr 2002 16:56:20 GMT

Hi Juan,

"Juan I. Cicuendez" <jicicuendez@gmv.es> wrote...

> I have a scientific formatted number (e.g. 6.8977653e-18) and I have
> to split the exponent and number into two parts like this:
> long:68977653
> exp:-25
> where the first factor has to be a long number and the second the
> exponent.
> The exponential factors can algo change.
>
> The solution I came up is to turn the number into strings and then
> byte(mystring), obtaining the position of '.' and 'e' and then back to
> numbers. This seems to be quite slow and since I have a large number
> of data I don't think is very efficient.

Here's a way that you might like, but I know some others won't, as it uses
the mysterious and cryptic "regular expressions" feature. I make a few
assumptions:

- you have an array of strings ready to process, for example:
  str = ['6.8977653e-18', '-46.7654e-19']

- all of them have a '.' and an 'e'

If so, these 3 lines of IDL code should work for you:

```
;   Match each string with a pattern like "*.*e*"
;               " *  .  * e  *"
strPieces = StRegEx(str, '(.*)(\.)(.*)(e)(.*)', /Extract, /SubExpr)
;   strPieces now contains StrArr(6, n), columns 1-5 hold the pieces

;   Put columns together to make desired 'long' values
longs = Reform(Long(strPieces[1,*] + strPieces[3,*]))
```

```
;    Adjust values in last column to get correct exponents
exps = Reform(Long(strPieces[5,*]) + StrLen(strPieces[3,*]))
```

```
IDL> print, longs
   68977653    -467654
IDL> print, exps
      -11       -15
```

There may be methods that execute faster, on my system this takes 0.0003 s
per string.

Cheers,
--
-Dick

Dick Jackson   /   dick@d-jackson.com
D-Jackson Software Consulting /   http://www.d-jackson.com
Calgary, Alberta, Canada   / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Obtaining exponent from a scientific format number
Posted by David Fanning on Wed, 17 Apr 2002 17:07:23 GMT
View Forum Message <> Reply to Message

Dick Jackson (dick@d-jackson.com) writes:

> Here's a way that you might like, but I know some others won't, as it uses
> the mysterious and cryptic "regular expressions" feature.
> ;   " * . * e *"
> strPieces = StRegEx(str, '(.*)(\.)(.*)(e)(.*)', /Extract, /SubExpr)
> ;   strPieces now contains StrArr(6, n), columns 1-5 hold the pieces

I just think anything that has a bunch of &%#@s in it
is a real programmer's solution, using real programmer
language! :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Obtaining exponent from a scientific format number
Posted by Dick Jackson on Wed, 17 Apr 2002 22:21:17 GMT

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.172759ba2f12aa6198988f@news.frii.com...
> Dick Jackson (dick@d-jackson.com) writes:
>
>> Here's a way that you might like, but I know some others won't, as it uses
>> the mysterious and cryptic "regular expressions" feature.
>> ;   " *   .   * e   *"
>> strPieces = StRegEx(str, '(.*)(\.)(.*)(e)(.*)', /Extract, /SubExpr)
>> ;   strPieces now contains StrArr(6, n), columns 1-5 hold the pieces
>
> I just think anything that has a bunch of &%#@s in it
> is a real programmer's solution, using real programmer
> language! :-)

Just to clarify, I think David would be paying a Real Compliment if he
capitalized Real Programmer, but it seems he's been working on his backhand!
:-)

Hmm, upon seeing cool IDL stuff David has done, I've sometimes said
"Unreal!". Does that make him an Unreal Programmer? :-)

Cheers,
--
-Dick

Dick Jackson   /   dick@d-jackson.com
D-Jackson Software Consulting /   http://www.d-jackson.com
Calgary, Alberta, Canada   / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Obtaining exponent from a scientific format number
Posted by Nigel Wade on Thu, 18 Apr 2002 08:41:26 GMT

Juan I. Cicuendez wrote:

> Hi all,
>
> I have a problem and I am looking for a faster solution:
>
> I have a scientific formatted number (e.g. 6.8977653e-18) and I have
> to split the exponent and number into two parts like this:
> long:68977653

> exp:-25
> where the first factor has to be a long number and the second the
> exponent.
> The exponential factors can algo change.
>
> The solution I came up is to turn the number into strings and then
> byte(mystring), obtaining the position of '.' and 'e' and then back to
> numbers. This seems to be quite slow and since I have a large number
> of data I don't think is very efficient.
>
> Thank you very much in advance, I would appreciate any hints.
>
> Juan

How about a mathematical solution?

IDL> number=6.8977653d-18

IDL> digits=8
IDL> exponent=floor(alog10(number))-(digits-1)
IDL> mantissa=round(number/10.0d0^exponent)
IDL> print,exponent
      -25
IDL> print,mantissa
   68977653
IDL>

It won't be entirely accurate due to the use of logs and exponentiation,
but it might be close enough (I've used double everywhere to increase
accuracy).

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
        University of Leicester, Leicester, LE1 7RH, UK
E-mail :   nmw@ion.le.ac.uk
Phone :   +44 (0)116 2523568, Fax : +44 (0)116 2523555