

---

Subject: Re: Need help with Wavelet Workbench  
Posted by [jab7981](#) on Wed, 07 Apr 1999 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On 7 Apr 1999 14:11:34 GMT, steinh@ulrik.uio.no (Stein Vidar Hagfors Haugan) wrote:

>  
> In article <370b52f7.335334@news.frontiernet.net>  
> jkbishop@frontiernet.net (Jonathan Bishop) writes:  
>  
  
>> When I put the 32768 point set in,  
>> the data set gets truncated to 16384 points because  
>> fix(alog(n\_elements(x\_work))/alog(2))) evaluates to 14 instead of  
>> 15. alog(n\_elements(x\_work))/alog(2) is given as 15.0000. Can  
>> someone explain this so even a mechanical engineer can  
>> understand?  
>  
> IDL> print,alog(32768)/alog(2),form='(g15.10)'  
> 14.99999905  
>  
> It would be wiser in this case to use round() instead of  
> fix() -- or use the logb() function I posted recently!

I thought of that, but naive use of round will make it think a dataset

of 32767 points is  $2^{15}$  points, then an array subscript will be out of range. A kludge like `round(100.*alog(...)/alog(2))/100` would at least work here. I'll pass on using logb, though something along those lines might be needed if WWB is overhauled to use long data sets; I'll just comment out the check and be careful, now that I know the cause.

The result is that WWB only works for datasets < 32768 in length rather than <= 32768 in its original form.

>  
> Regards,  
>  
> Stein Vidar

--  
Jonathan Bishop

---

---

Subject: Re: Need help with Wavelet Workbench

---

Posted by [steinhh](#) on Wed, 07 Apr 1999 07:00:00 GMT

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

---

In article <370b52f7.335334@news.frontiernet.net>  
jkbishop@frontiernet.net (Jonathan Bishop) writes:

> I'm trying to use Wavelet Workbench on a long (48000 pt) signal. I  
> think that two separate problems are occurring.  
>  
> I upsampled the data set to 65536 points (by zero padding in frequency  
> space). I hacked wreaddat, wdyadlng, wdyad, and wfwtpo to use long  
> integers in some places. The result is that I can now plot the  
> scalogram for my data set (wreaddat, wintwave, wdoscog are the  
> programs I'm calling). However, the plot of the scalogram looks like  
> only the first half of the data set is being used. The coarser scales  
> have some variation just beyond the half-way point (bleed-over from  
> the convolution process?), but the more detailed scales show a solid  
> color in the upper half of the time axis. Anyone have any ideas what  
> is going on?

My initial guess would be that there's still some problem  
with the use of integer vs long... Other than that, I  
haven't a clue.

(I should mention that I've no insight into the programs  
that are discussed here, I'm only guessing)

> So far, I have tried upsampling again to  $2 \times 65536$  points (whatever  
> that is). The result is that the convolution-by-FFT process in  
> wmfilt takes forever (I didn't wait for it to finish; it was taking at  
> least 10 times as long as the 65536 point set, as verified by  
> printed status statements). I don't understand why, but would  
> the FFT process be the problem with the 65536 data set?

What's the algorithmic complexity of a full wavelet  
decomposition? I'm sure it's not simply  $\log_2(n)$ ... An  
individual FFT is of order  $\log_2(n)$ , however, but doesn't  
the wmfilt procedure do a lot (order  $n$  at least?) of them?  
Since it's printing out it's status as it churns away, I  
assume it's not just one gargantuan fft operation that's  
taking so long...

Other than that, execution time may not be as expected  
when a problem grows, owing to a problem that's larger  
than the processor cache size, or due to swapping.

> When I put the 32768 point set in,  
> the data set gets truncated to 16384 points because  
>  $\text{fix}(\text{alog}(n\_elements(x\_work))/\text{alog}(2)))$  evaluates to 14 instead of

> 15. `alog(n_elements(x_work))/alog(2)` is given as 15.0000. Can  
> someone explain this so even a mechanical engineer can  
> understand?

```
IDL> print,alog(32768)/alog(2),form='(g15.10)'  
14.99999905
```

It would be wiser in this case to use `round()` instead of  
`fix()` -- or use the `logb()` function I posted recently!

Regards,

Stein Vidar

---