## Subject: Re: strange behaviour of bytscl by large arrays Posted by Lajos Foldy on Tue, 24 Apr 2012 17:30:10 GMT

View Forum Message <> Reply to Message

On Monday, April 23, 2012 10:22:08 PM UTC+2, Chris Torrence wrote:

- > Well, wrong is perhaps too strong of a word. The real word is "fast". I just did a test where I changed the internal implementation of FINDGEN to use an integer counter. The "float" counter is 4 times faster than using an integer counter and converting it to floats.
- > However, perhaps we could look at the size of the input array, and switch to using the slower integer counter if it was absolutely necessary. I'll give it a thought.
- > Thanks for reporting this!
- > - Obsa

>

>

- > Cheers,
- > Chris
- > Exelis VIS

I could not reproduce this 4x slowdown. The integer counter + conversion method is only 30% slower in the following C test program (Intel Core i5-2500, 64 bit Linux):

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
double timediff(struct timeval* tv1, struct timeval* tv2)
 return tv2->tv sec-tv1->tv sec+(tv2->tv usec-tv1->tv usec)*1e-6;
}
int main()
 int n=1000000000, j;
 float* x=malloc(n*sizeof(float));
 float f;
 struct timeval tv1, tv2;
 gettimeofday(&tv1, NULL);
 for (j=0; j< n; j++) x[j]=j;
 gettimeofday(&tv2, NULL);
 printf("integer counter: %lf %f\n", timediff(&tv1, &tv2), x[n-1]);
 gettimeofday(&tv1, NULL);
 f=0.0:
 for (j=0; j< n; j++) x[j]=f++;
 gettimeofday(&tv2, NULL);
 printf("float counter: %lf %f\n", timediff(&tv1, &tv2), x[n-1]);
```

```
}
```

Also, IDL help says:

The FINDGEN function creates a floating-point array of the specified dimensions. Each element of the array is set to the value of its one-dimensional subscript.

So it should be equivalent to float(lindgen()), as one-dimensional subscript is an integer.

But I don't want to convince you, I can accept that it is a feature :-)

regards, Lajos