
Subject: Comparing tabulated functions

Posted by [jameskuyper](#) on Thu, 03 Jan 2008 15:47:18 GMT

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

I've got two functions of time, each of which is tabulated at thousands of different monotonically increasing time values; in the worst case, there may be as many as 14,400 time values. I want to estimate the difference between these functions. Unfortunately, the two functions will not, in general, be tabulated at the same time values. A simple approach that works fairly well is to use spline interpolation to interpolate both of them to a common grid, and then difference the interpolated values. However, occasionally one or the other of the two functions have data gaps. For any time value that is more than 1 second away from the nearest tabulated time for either function, I want to fill in the difference array with !VALUES.D_NAN. I believe this will cause the PLOT command to skip that point - if not, I need to find some other approach that will have this effect.

I can write simple, highly efficient C code that does exactly what I want; translating that code into IDL would make heavy use of loops, and therefore wouldn't be very efficient. I can write simple IDL code that takes no advantage of the fact that the arrays are monotonically increasing, calculating the difference of every point on the output grid from every point on either of the two input tables, and finding the minimum, but that seems extremely inefficient (and a memory hog!). Is there a simpler way to do this?

Subject: Re: Comparing tabulated functions

Posted by [lasse](#) on Fri, 04 Jan 2008 17:35:21 GMT

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

Hi,

On 3 Jan, 15:47, jameskuy...@verizon.net wrote:

> I can write simple, highly efficient C code that does exactly what I
> want;

excuse my heretic question: why would you want to use IDL, then?

Regards
Lasse Clausen

Subject: Re: Comparing tabulated functions

Posted by [jameskuyper](#) on Sat, 05 Jan 2008 23:42:49 GMT

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

Lasse Clausen wrote:

```
> Hi,  
>  
> On 3 Jan, 15:47, jameskuy...@verizon.net wrote:  
>> I can write simple, highly efficient C code that does exactly what I  
>> want;  
>  
> excuse my heretic question: why would you want to use IDL, then?
```

Convenience. The code to bring the data into an IDL context already exists, and I wanted to use the results as data for IDL plotting commands. Therefore, to bring C into this, I would have had to interface between C and IDL twice, which is moderately complicated, no matter how you do it.

I was pretty sure that there was an IDL feature whose name I couldn't quite remember that would allow me to deal with this with relative efficiency and simplicity in pure IDL. I was right - `VALUE_LOCATE` is what I was looking for.
