Subject: Re: Finding the closest value in an array...
Posted by Wayne Landsman on Tue, 30 Mar 2004 16:54:47 GMT
View Forum Message <> Reply to Message

Tim Robishaw wrote:

> Seems like every few minutes I'm taking a scalar and trying to locate
> which value in an array it's closest to. VALUE_LOCATE() finds the
> interval of a monotonic vector that the value lives in, so it's not
> quite what I'm looking for, but it's awfully close! I end up just
> doing this:
>
> IDL> useless = min(abs(vector-value),minindx)
> IDL> closest = vector[minindx]

Presuming your vector is not monotonic, then your two lines perform the minimum necessary number of calculations (and are completely vectorized) -- you have to check every value in vector.   Though you don't need to save the "useless" scalar value returned by MIN, this value did need to be calculated, and the extra overhead is very MINimal ;-)

If you have a monotonic vector then -- like VALUE_LOCATE() -- you can use bisection to minimize the number of values in vector that you need to check.

> I'd bet
> VALUE_LOCATE() is doing a lot more stuff behind the scenes than the
> simple two lines above (judging from the old Goddard library routine).

You probably saw Craig Markwardt's implementation of the VALUE_LOCATE algorithm for users of IDL V5.2 and earlier, with its bizarre call SPL_INTERP.   The program is not really performing cubic interpolation, but rather taking advantage of the bisection algorithm within the intrinsic SPL_INTERP function.   Even with the unnecessary spline calculations this method is still faster than coding the bisection algorithm within IDL.   Since V5.3 the bisection algorithm is available directly in the VALUE_LOCATE function.

--Wayne Landsman