Re: Highest precision of finite difference, given machine's epsilon

Source: http://sci.tech−archive.net/Archive/sci.math.num−analysis/2006−07/msg00206.html

- *From*: bv <bvoh@xxxxxxxxxxxxx>
- *Date*: Fri, 14 Jul 2006 21:53:01 GMT

hmobahi@xxxxxxxxx wrote:

> Computing derivative using finite difference is
> df(x)/dx=(f(x+h)−f(x))/h . Of course, the smaller the h is the higher
> precision is achieved, but if h gets too small, there is a risk of
> losing precision due to machine's limitation in representing very small
> numbers.

You're to be commended for the astute observation that even n e t l i b
ode codes apparently *missed*. e.g. look at their clumsy attempts to
estimate initial steps involving finite difference eqns.

> Somewhere I read a rule of thumb for choosing h is as
> h=|x|*sqrt(epsilon) , where epsilon is the smallest number that the
> machine can represent.

Correct, although you have to be mindful of the variations applied
to higher order difference schemes. The formula is derived from a
general error model first postulated by Henrici in his classic textbook.

> 1. The first question: is this rule of thumb correct? I just read it in
> a web page and I cannot trust it so easily.

You can trust, but you must verify, as you've done. btw, it's not a rule
of thumb, it's numerical analysis equivalent of "mc^2".

> 2. What if I am interested in the second derivative, i.e. (
> f(x−h)−2*f(x)+f(x+h)) / (h^2) ... now how the rule works? Should it be
> h=|x|*sqrt(epsilon) or (h^2)=|x|*sqrt(epsilon) ?

Actually, it's neither. If it was, then the former squared would be correct, that is, one formula, one h. For actual h in this case take a look at, http://www.uc.edu/sashtml/ormp/chap5/sect28.htm

--
sdx
http://www.sdynamix.com

.