Subject: Re: Comparing 2 arrays
Posted by James Kuyper on Mon, 27 Aug 2007 17:40:40 GMT
View Forum Message <> Reply to Message

Conor wrote:
> On Aug 26, 12:43 pm, David Fanning <n...@dfanning.com> wrote:
>> Jean H. writes:
>>> to get back to a previous discussion we had a few month ago about being
>>> "sufficiently close to zero", shouldn't it be (data1.A - data2.B) LT
>>> epsilon * data1.A , with epsilon=(machar()).eps?
>>
>> Humm, I don't recall that discussion. But I can see how
>> this number might meet the criteria of "sufficiently close".
>> On the other hand, I can also envision situations where
>> the number could be orders of magnitude larger and still
>> work for a particular application. I'm probably mistaken,
>> but it seems to me "sufficiently close" is an arbitrary
>> value that must be picked empirically to match the data
>> and what you are trying to do with it.
>>
>> Cheers,
>>
>> David
>>
>> P.S. I'm just thinking that "sufficiently close" to a
>> black hole, for example, might be a completely different
>> number than "sufficiently close" to my house.
>>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming:http://www.dfanning.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
> Hmm... I think Jean might be on to something. After all, the error in
> question hear is the rounding error of the computer, and that rounding
> error is always an error on the last 'bit' of a floating point
> number. So for instance if you had two floating point numbers:
>
> 1.1123453e15
> and
> 1.1123454e15
>
> These might be the same number (to within the rounding error) but the
> difference between them is about 6.7e07. That's assuming of course
> that I'm properly understanding floating point representation (I'm an
> astronomer, not a computer engineer).

The direct effect of a single roundoff error shouldn't be more than 1 bit in the last position. However, it is often the case that two different numbers that should mathematically be the same, have been brought together through a long series of operations. A roundoff error in the first operation could be magnified or reduced by the next operation, in addition to that operation creating round-off errors of its own. In general, you must either analyze the propogation of error through the calculations, or at least measure the typical error sizes empirically, as David suggested.
