
Subject: Re: Convolving speed issue

Posted by [Paul Van Delst\[1\]](#) on Mon, 21 Apr 2008 19:21:30 GMT

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

pgrigis@gmail.com wrote:

>

> rog...@googlemail.com wrote:

>> Dear Paolo,

>> yes, it isnt so much slower. I could also use fftw, which is available

>> for matlab and also for IDL. But nevertheless, I had to convolve it

>> directly and not in frequency domain.

>

> OK, here's where I cannot follow you: why do you care how the

> convolution is preformed? If you get the same answer by the

> direct (slow) and FFT (fast) method, why would you not want to use

> the latter one?

I've been lurking about in this thread.

I have to agree with Paolo. If you do convolutions "explicitly" [my terminology] in the original data domain, it's nearly always going to be much slower than doing multiplications in the frequency domain.

Mathematically, the operators should be the same (with due respect paid to numerical precision issues, i.e. always FFT in double precision in IDL! :o)

>> The problem is that the matrices

>> could be very large (their size is changing dynamically) and there are

>> also some iterations (nearly 200) for each convolve. So I always try

>> to use as often as possible reform, replicate and rebin, because those

>> are very fast for manipulating or computing arrays.

Probably more details are required about the iterations for each convolve. What is changing? The data itself? The convolution function? Both?

Without knowing some more of the nitty gritty details of your problem it's hard to comment usefully in a generic manner.

In my experience, however, I've found that doing something like

$$\text{FFT}^{-1}(\text{FFT}(\text{data}) * \text{FFT}(\text{convfn}))$$

is nearly always much much speedier[*] than doing

$$\text{Data}(x) \text{ convfn}$$

where (x) is my ASCII Art representing the convolution operator. :o)

cheers,

paulv

[*] Where the data and convfn have been specified (or zeropadded) to an efficient number of points, FFT-wise.
