
Subject: Re: Correction: 2D FFT
Posted by [steinhh](#) on Tue, 29 Oct 1996 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <326DCEDB.41C6@phys.strath.ac.uk>, Graeme K Harkness
<graeme@phys.strath.ac.uk> writes:

|> Walid,
|>
|> I'm pretty sure that this re-arrangement of the frequency space is
|> related
|> to the methods used to do Fast Fourier Transforms in general (but it's
|> been
|> a long time since I studied this stuff :-)
|>

Well, sort of, but it also makes sense to have e.g., the zero frequency
at (0,0) as others have pointed out..

|> I have a couple of routines (FFT1D and FFT2D) which take the Fourier
|> transforms and return you the re-ordered data (with zero frequency at
|> the
|> centre). They do use the standard IDL routines and then re-order
|> afterwards
|> so they aren't the most efficient things ever but they work very well.
|> (I suppose if you wanted very efficient code you wouldn't be using the
|> FFTs
|> in IDL anyway since I'd bet you could call an external function in
|> FORTRAN
|> (or something) to do it much faster!)
|>

I wouldn't recommend spending time on trying to beat IDL's array
operations, especially stuff like the FFT functions! They're quite
well optimized.

Once when I had to compute a lot of auto-correlation functions,
I tried to use Numerical Recipes to beat IDL, taking advantage
of the fact that my data points were real, not complex, etc.,
and I was ending up with real data points as well.

Even using every trick in the book, I ended up not saving more
than about 5% of the execution time. The effort would have been
a waste of time if it hadn't been for the fact that I had to
also apply a filter in frequency space, which could be done
a lot more efficient when they were done in the program
containing the ffts.

Stein Vidar
