Subject: Re: Fast Cosine Transform Routine
Posted by steinhh on Mon, 25 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

> I use IDL a lot and currently have the need to perform 2D cosine transforms.
> I have implemented this using the FFT command in IDL. (How to do a cosine
> transform using the FFT is described in "Numerical Recipes in C"). I am not
> happy with this approach and would prefer to use a more direct method. I can
> code the cosine transform into IDL but first I would like to see if anyone
> knows of an IDL routine that already exists for 2D cosine transforms.

The question is, why would you like to do a strict cosine transform
rather than the full FFT and extracting the cosine part?

If your problem is speed, my guess is that you won't get anywhere
near a decent return on the invested time. If you're out to
save space, you could get away with half (or less, depending
on what kind of processing you do).

Once when I was doing a lot of signal processing in IDL, and needed
to do a lot of autocorrelation calculations, I tried to "beat"
IDL's FFT when transforming back from the frequency domain to
the time domain. Even though I took the advantage of using a
cosine transform instead of a full FFT (and at the same time
locking myself into using datasets with 2^N points), the only
reason I saved some time was because I needed to do a lot of
calculations, like filtering etc, before transforming back.
Thus I could write the whole sequence of FFT->various filtering->
(reverse) cosine transform etc in one block of C code, saving
a lot of IDL's command interpreting overhead.

Even though this was a 1D transform, not 2D, I think in general
that IDL's built in numerical array processing like FFT is
so highly optimized that it's difficult to save time rewriting
stuff.

Regards,

Stein Vidar