## Subject: Re: n-point FFT
Posted by tandp on Tue, 21 Nov 2000 08:00:00 GMT
View Forum Message <> Reply to Message

In article <onzoit8fai.fsf@cow.physics.wisc.edu>,
craigmnet@cow.physics.wisc.edu wrote:


> Jean Marc Delvit <delvit@onecert.fr> writes:
>>
>> I wants to know how to do a n-point FFT with IDL
>> the same of FFT(X,n) in matlab (if the length of X is less than n, X is
>> padded with trailing zeros to length n)
>
> ... and if you read the documentation, you will find that the IDL
> routine FFT will work on an array of any size.  It's very easy to
> do any needed zero-padding yourself (ie, fft([x, fltarr(nzeros)]) ).
>
> Craig


It should also be noted that IDL 5.4 implements a mixed-radix FFT. Time
series of length equal to a power of 2 are no longer a prerequisite for
performance. This eases the need for zero-padding and improves the
accuracy of time series that can conform to the radices (radiches?) used.

Mike


## Subject: Re: n-point FFT
Posted by Paul van Delst on Tue, 21 Nov 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Paul Woodford wrote:
>
> In article <onzoit8fai.fsf@cow.physics.wisc.edu>,
> craigmnet@cow.physics.wisc.edu wrote:
>
>> It's very easy to
>> do any needed zero-padding yourself (ie, fft([x, fltarr(nzeros)]) ).
>
> Easy, but inconvenient.  I also missed the Matlab notation when I
> switched to IDL, and finally hacked a small "efft" function to recreate
> it.

Different strokes, but I would never recommend this sort of thing mostly because too many
times I've seen (both others and myself) get in trouble with FFT's because something I or
someone else assumed about the particular FFT implementation (fortran, idl, matlab, C,
various flavours - take your pick) was wrong (or non-portable). Particularly when it comes
to specifying exactly where the Nyquist point is.

Thus, I would always hope that my colleagues are painstakingly clear about how many points they expect to have in, and with how many zeros they padded, their spectra(um).

> I keep forgetting to send this feature request to RSI...

I wouldn't consider it a feature, but a user bug waiting to happen...... +/- 1 point errors in FFTs can be capriciously subtle  :o(

BTW, I'm not bitter...no, not at all (snuckin' fessin' rotten rasterdly FFTs.....)

:o)

paulv

--
Paul van Delst        Ph:  (301) 763-8000 x7274
CIMSS @ NOAA/NCEP       Fax: (301) 763-8545
Rm.207, 5200 Auth Rd.   Email: pvandelst@ncep.noaa.gov
Camp Springs MD 20746

---

## Subject: Re: n-point FFT
Posted by davidf on Tue, 21 Nov 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Paul Woodford (woodford@us.net) writes:

> Easy, but inconvenient.  I also missed the Matlab notation when I
> switched to IDL, and finally hacked a small "efft" function to recreate
> it.
>
> I keep forgetting to send this feature request to RSI...

I've noticed as a rule of thumb that if something takes
less than 30 seconds to code up in IDL that it usually
gets a low score on the "must have in the next version
of IDL" sweepstakes. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/

## Subject: Re: n-point FFT
Posted by Paul Woodford on Tue, 21 Nov 2000 08:00:00 GMT
View Forum Message <> Reply to Message

In article <onzoit8fai.fsf@cow.physics.wisc.edu>,
craigmnet@cow.physics.wisc.edu wrote:

> It's very easy to
> do any needed zero-padding yourself (ie, fft([x, fltarr(nzeros)]) ).

Easy, but inconvenient.  I also missed the Matlab notation when I
switched to IDL, and finally hacked a small "efft" function to recreate
it.

I keep forgetting to send this feature request to RSI...

--
Paul Woodford, Ph.D.
Essex Corporation

For faster email response, replace us dot net with essexcorp dot com

## Subject: Re: n-point FFT
Posted by Craig Markwardt on Tue, 21 Nov 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Jean Marc Delvit <delvit@onecert.fr> writes:
>
> I wants to know how to do a n-point FFT with IDL
> the same of FFT(X,n) in matlab (if the length of X is less than n, X is
> padded with trailing zeros to length n)

... and if you read the documentation, you will find that the IDL
routine FFT will work on an array of any size.  It's very easy to
do any needed zero-padding yourself (ie, fft([x, fltarr(nzeros)]) ).

Craig

--
 --------------------------------------------------------------- --------------
Craig B. Markwardt, Ph.D.        EMAIL:   craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 --------------------------------------------------------------- --------------