Subject: dissapointing fftw
Posted by R.G. Stockwell on Fri, 06 Feb 2004 18:51:58 GMT
View Forum Message <> Reply to Message

Hi all,
there has been discussions about using fftw in idl through
external calls recently. Our wonderful SA set it up here,
and unfortunately the results are a bit dissapointing.

The results:
1) there is a step where the fftw algorithm creates a "wisdom" file
to determine which algorithm is ideal for the given situations
(depending on length, dimension, variable type, processor, etc.).
This _can_ be very time consuming, and since it depends on the length
of the data, it is not very general at all. (perhaps minutes to determine
wisdom when using the exhaustive search).
This only needs to be done once (but has to be redone if the length
of the data changes). There is also a small delay in loading the dlm and
reading the file, but this is only done once when you start idl.

2) fftw is slightly slower than IDL fft for _some_complex 1D time series, slightly
faster for _some_ complex data. I initially found fftw to be slower, but later tests
showed if faster, see below.
3) fftw is slightly faster than IDL fft for real 1D time series (fftw
only calculates the positive 1/2 of the spectrum)
4) fftw is much (~8) times faster for 2D ffts of real data
(again fftw calcs only 1/2 the spectrum).

So, imho, use fftw when ffting 2D real-valued images (especially if
they are the same size). I.E. it is ideal for ffting data from a CCD for instance.

For general fft-ing of general time series (various length), might as
well stick with IDL fft.

This is dissapointing that the fftw is so slow in idl (my guess because of
the overhead of the external call). When I compare IDL fft to matlab fft
(which internally uses fftw), matlab smokes idl, almost an order of magnitude
faster.

More detailed results follow.

Cheers,
bob

float:

Elapsed time for /exhaustive =   4832.5616
SPEW   COMPLEX   = Array[524289]
SPEIDL   COMPLEX   = Array[1048576]
FFTW: 0.36619304
IDL fft: 0.53575690
float,nthreads=2:
Elapsed time for /exhaustive =   22502.406
SPEW   COMPLEX   = Array[524289]
SPEIDL   COMPLEX   = Array[1048576]
FFTW: 0.48127429
IDL fft: 0.68620352
/destroy:
Elapsed time for /exhaustive =   5364.2469
SPEW   COMPLEX   = Array[524289]
SPEIDL   COMPLEX   = Array[1048576]
FFTW: 0.36528679
IDL fft: 0.54512086
float 2d:
Elapsed time for /exhaustive =   195.64079
SPEW   COMPLEX   = Array[513, 1024]
SPEIDL   COMPLEX   = Array[1024, 1024]
FFTW: 0.060319290
IDL fft: 0.50715707
float 2d,nthreads=2:
Elapsed time for /exhaustive =   616.68032
SPEW   COMPLEX   = Array[513, 1024]
SPEIDL   COMPLEX   = Array[1024, 1024]
FFTW: 0.069700079
IDL fft: 0.50730018
float 2d,/destroy:
Elapsed time for /exhaustive =   196.30516
SPEW   COMPLEX   = Array[513, 1024]
SPEIDL   COMPLEX   = Array[1024, 1024]
FFTW: 0.073251941
IDL fft: 0.50794953
double:
Elapsed time for /exhaustive =   7275.4535
SPEW   DCOMPLEX = Array[524289]
SPEIDL   DCOMPLEX = Array[1048576]
FFTW: 0.28960719
IDL fft: 0.73290416

 /estimate:
float:
% Compiled module: COMPARE_FFT.
% Loaded DLM: FFTW.
% FFTW: Imported wisdom from file.
Elapsed time for wisdom =   0.48689103

SPEW COMPLEX = Array[524289]
SPEIDL COMPLEX = Array[1048576]
FFTW: 0.48639197
IDL fft: 0.53754315
/destroy:
Elapsed time for wisdom = 0.49383688
SPEW COMPLEX = Array[524289]
SPEIDL COMPLEX = Array[1048576]
FFTW: 0.48593453
IDL fft: 0.53892898
% Compiled module: DIST.
float 2d:
Elapsed time for wisdom = 1.1336241
SPEW COMPLEX = Array[513, 1024]
SPEIDL COMPLEX = Array[1024, 1024]
FFTW: 0.10067668
IDL fft: 0.49136082
float 2d,/destroy:
Elapsed time for wisdom = 1.0966880
SPEW COMPLEX = Array[513, 1024]
SPEIDL COMPLEX = Array[1024, 1024]
FFTW: 0.086122580
IDL fft: 0.49067524
double:
% FFTW: Can't read wisdom file.
Elapsed time for wisdom = 162.97562
SPEW DCOMPLEX = Array[524289]
SPEIDL DCOMPLEX = Array[1048576]
FFTW: 0.57721099
IDL fft: 0.66078199
/destroy:
Elapsed time for wisdom = 163.17508
SPEW DCOMPLEX = Array[524289]
SPEIDL DCOMPLEX = Array[1048576]
FFTW: 0.57687245
IDL fft: 0.66050471
double 2d:
Elapsed time for wisdom = 1.3941269
SPEW DCOMPLEX = Array[513, 1024]
SPEIDL DCOMPLEX = Array[1024, 1024]
FFTW: 0.12435352
IDL fft: 0.61359194
doubble 2d,/destroy:
Elapsed time for wisdom = 1.3879058
SPEW DCOMPLEX = Array[513, 1024]
SPEIDL DCOMPLEX = Array[1024, 1024]
FFTW: 0.11186049
IDL fft: 0.61340666

complex:
Elapsed time for wisdom =   164.32227
SPEW   DCOMPLEX = Array[1048576]
SPEIDL   COMPLEX   = Array[1048576]
FFTW: 0.70433186
IDL fft: 1.0169743
/destroy:
Elapsed time for wisdom =   163.18478
SPEW   DCOMPLEX = Array[1048576]
SPEIDL   COMPLEX   = Array[1048576]
FFTW: 0.70836432
IDL fft: 1.0172801
complex 2d:
Elapsed time for wisdom =   1.9998610
SPEW   DCOMPLEX = Array[1024, 1024]
SPEIDL   COMPLEX   = Array[1024, 1024]
FFTW: 0.22408933
IDL fft: 0.94814202
complex 2d,/destroy:
Elapsed time for wisdom =   1.9922080
SPEW   DCOMPLEX = Array[1024, 1024]
SPEIDL   COMPLEX   = Array[1024, 1024]
FFTW: 0.21766154
IDL fft: 0.94665491
dcomplex:
Elapsed time for wisdom =   278.01961
SPEW   DCOMPLEX = Array[1048576]
SPEIDL   DCOMPLEX = Array[1048576]
FFTW: 0.86303161
IDL fft: 1.2512911
/destroy:
Elapsed time for wisdom =   278.62218
SPEW   DCOMPLEX = Array[1048576]
SPEIDL   DCOMPLEX = Array[1048576]
FFTW: 0.85589477
IDL fft: 1.2513304
dcomplex 2d:
Elapsed time for wisdom =   2.5013170
SPEW   DCOMPLEX = Array[1024, 1024]
SPEIDL   DCOMPLEX = Array[1024, 1024]
FFTW: 0.29264280
IDL fft: 1.2062091
dcomplex 2d,/destroy:
Elapsed time for wisdom =   2.5212729
SPEW   DCOMPLEX = Array[1024, 1024]
SPEIDL   DCOMPLEX = Array[1024, 1024]
FFTW: 0.29645642
IDL fft: 1.2053123

Subject: Re: dissapointing fftw
Posted by b_gom on Tue, 10 Feb 2004 18:56:36 GMT
View Forum Message <> Reply to Message

Bob,

I share some of your disapointment, but I don't have Matlab speeds to compare to. The speed of FFTW does rely heavily on the plan, and for some jobs it is just as well to stick with the IDL FT. I suspect the speed advantage will vary quite a bit with the data and hardware at hand.

Here's a plot of a quick test on my machine, using the fftw_one function and IDL's FT function on a complex array of various lengths:
http://people.uleth.ca/~brad.gom/fftw/new-3.png

Here's the actual times as a function of length. Red is the FFTW.
http://people.uleth.ca/~brad.gom/fftw/new-1.png

Arrays less than 2^16 are faster in IDL.
These data sets were all powers of 2 in length, and the trends will be different for non power of two lengths. For example, here are the results for arrays of (2^n)+1 in length:

http://people.uleth.ca/~brad.gom/fftw/new-4.png
http://people.uleth.ca/~brad.gom/fftw/new-5.png

and for all lengths between 10 and 110:

http://people.uleth.ca/~brad.gom/fftw/new-6.png
http://people.uleth.ca/~brad.gom/fftw/new-6.png

I haven't dug into my DLM to see where time is being wasted, but it seems as though you still have to carefully consider the size of data going into your FT routine if you want the best performance, no matter which routine you use..

Brad

"R.G. Stockwell" <noemail@please.com> wrote in message news:<snRUb.22$an2.31659@news.uswest.net>...
> Hi all,
> there has been discussions about using fftw in idl through
> external calls recently. Our wonderful SA set it up here,
> and unfortunately the results are a bit dissapointing.
>
> The results:
> 1) there is a step where the fftw algorithm creates a "wisdom" file
> to determine which algorithm is ideal for the given situations

> (depending on length, dimension, variable type, processor, etc.).
> This _can_ be very time consuming, and since it depends on the length
> of the data, it is not very general at all. (perhaps minutes to determine
> wisdom when using the exhaustive search).
> This only needs to be done once (but has to be redone if the length
> of the data changes). There is also a small delay in loading the dlm and
> reading the file, but this is only done once when you start idl.
>
> 2) fftw is slightly slower than IDL fft for _some_complex 1D time series, slightly
> faster for _some_ complex data. I initially found fftw to be slower, but later tests
> showed if faster, see below.
> 3) fftw is slightly faster than IDL fft for real 1D time series (fftw
> only calculates the positive 1/2 of the spectrum)
> 4) fftw is much (~8) times faster for 2D ffts of real data
> (again fftw calcs only 1/2 the spectrum).
>
> So, imho, use fftw when ffting 2D real-valued images (especially if
> they are the same size). I.E. it is ideal for ffting data from a CCD for instance.
>
> For general fft-ing of general time series (various length), might as
> well stick with IDL fft.
>
> This is dissapointing that the fftw is so slow in idl (my guess because of
> the overhead of the external call). When I compare IDL fft to matlab fft
> (which internally uses fftw), matlab smokes idl, almost an order of magnitude
> faster.
>
>
> More detailed results follow.
>
> Cheers,
> bob
>
>
>
> float:
> Elapsed time for /exhaustive = 4832.5616
> SPEW COMPLEX = Array[524289]
> SPEIDL COMPLEX = Array[1048576]
> FFTW: 0.36619304
> IDL fft: 0.53575690
> float,nthreads=2:
> Elapsed time for /exhaustive = 22502.406
> SPEW COMPLEX = Array[524289]
> SPEIDL COMPLEX = Array[1048576]
> FFTW: 0.48127429
> IDL fft: 0.68620352
> /destroy:

> Elapsed time for /exhaustive = 5364.2469
> SPEW COMPLEX = Array[524289]
> SPEIDL COMPLEX = Array[1048576]
> FFTW: 0.36528679
> IDL fft: 0.54512086
> float 2d:
> Elapsed time for /exhaustive = 195.64079
> SPEW COMPLEX = Array[513, 1024]
> SPEIDL COMPLEX = Array[1024, 1024]
> FFTW: 0.060319290
> IDL fft: 0.50715707
> float 2d,nthreads=2:
> Elapsed time for /exhaustive = 616.68032
> SPEW COMPLEX = Array[513, 1024]
> SPEIDL COMPLEX = Array[1024, 1024]
> FFTW: 0.069700079
> IDL fft: 0.50730018
> float 2d,/destroy:
> Elapsed time for /exhaustive = 196.30516
> SPEW COMPLEX = Array[513, 1024]
> SPEIDL COMPLEX = Array[1024, 1024]
> FFTW: 0.073251941
> IDL fft: 0.50794953
> double:
> Elapsed time for /exhaustive = 7275.4535
> SPEW DCOMPLEX = Array[524289]
> SPEIDL DCOMPLEX = Array[1048576]
> FFTW: 0.28960719
> IDL fft: 0.73290416
>
> /estimate:
> float:
> % Compiled module: COMPARE_FFT.
> % Loaded DLM: FFTW.
> % FFTW: Imported wisdom from file.
> Elapsed time for wisdom = 0.48689103
> SPEW COMPLEX = Array[524289]
> SPEIDL COMPLEX = Array[1048576]
> FFTW: 0.48639197
> IDL fft: 0.53754315
> /destroy:
> Elapsed time for wisdom = 0.49383688
> SPEW COMPLEX = Array[524289]
> SPEIDL COMPLEX = Array[1048576]
> FFTW: 0.48593453
> IDL fft: 0.53892898
> % Compiled module: DIST.
> float 2d:

> Elapsed time for wisdom = 1.1336241
> SPEW COMPLEX = Array[513, 1024]
> SPEIDL COMPLEX = Array[1024, 1024]
> FFTW: 0.10067668
> IDL fft: 0.49136082
> float 2d,/destroy:
> Elapsed time for wisdom = 1.0966880
> SPEW COMPLEX = Array[513, 1024]
> SPEIDL COMPLEX = Array[1024, 1024]
> FFTW: 0.086122580
> IDL fft: 0.49067524
> double:
> % FFTW: Can't read wisdom file.
> Elapsed time for wisdom = 162.97562
> SPEW DCOMPLEX = Array[524289]
> SPEIDL DCOMPLEX = Array[1048576]
> FFTW: 0.57721099
> IDL fft: 0.66078199
> /destroy:
> Elapsed time for wisdom = 163.17508
> SPEW DCOMPLEX = Array[524289]
> SPEIDL DCOMPLEX = Array[1048576]
> FFTW: 0.57687245
> IDL fft: 0.66050471
> double 2d:
> Elapsed time for wisdom = 1.3941269
> SPEW DCOMPLEX = Array[513, 1024]
> SPEIDL DCOMPLEX = Array[1024, 1024]
> FFTW: 0.12435352
> IDL fft: 0.61359194
> doubble 2d,/destroy:
> Elapsed time for wisdom = 1.3879058
> SPEW DCOMPLEX = Array[513, 1024]
> SPEIDL DCOMPLEX = Array[1024, 1024]
> FFTW: 0.11186049
> IDL fft: 0.61340666
> complex:
> Elapsed time for wisdom = 164.32227
> SPEW DCOMPLEX = Array[1048576]
> SPEIDL COMPLEX = Array[1048576]
> FFTW: 0.70433186
> IDL fft: 1.0169743
> /destroy:
> Elapsed time for wisdom = 163.18478
> SPEW DCOMPLEX = Array[1048576]
> SPEIDL COMPLEX = Array[1048576]
> FFTW: 0.70836432
> IDL fft: 1.0172801

> complex 2d:
> Elapsed time for wisdom = 1.9998610
> SPEW DCOMPLEX = Array[1024, 1024]
> SPEIDL COMPLEX = Array[1024, 1024]
> FFTW: 0.22408933
> IDL fft: 0.94814202
> complex 2d,/destroy:
> Elapsed time for wisdom = 1.9922080
> SPEW DCOMPLEX = Array[1024, 1024]
> SPEIDL COMPLEX = Array[1024, 1024]
> FFTW: 0.21766154
> IDL fft: 0.94665491
> dcomplex:
> Elapsed time for wisdom = 278.01961
> SPEW DCOMPLEX = Array[1048576]
> SPEIDL DCOMPLEX = Array[1048576]
> FFTW: 0.86303161
> IDL fft: 1.2512911
> /destroy:
> Elapsed time for wisdom = 278.62218
> SPEW DCOMPLEX = Array[1048576]
> SPEIDL DCOMPLEX = Array[1048576]
> FFTW: 0.85589477
> IDL fft: 1.2513304
> dcomplex 2d:
> Elapsed time for wisdom = 2.5013170
> SPEW DCOMPLEX = Array[1024, 1024]
> SPEIDL DCOMPLEX = Array[1024, 1024]
> FFTW: 0.29264280
> IDL fft: 1.2062091
> dcomplex 2d,/destroy:
> Elapsed time for wisdom = 2.5212729
> SPEW DCOMPLEX = Array[1024, 1024]
> SPEIDL DCOMPLEX = Array[1024, 1024]
> FFTW: 0.29645642
> IDL fft: 1.2053123

Subject: Re: dissapointing fftw
Posted by R.G. Stockwell on Tue, 10 Feb 2004 20:41:43 GMT
View Forum Message <> Reply to Message

"Brad Gom" <b_gom@hotmail.com> wrote in message
news:bde24eff.0402101056.19207d7e@posting.google.com...
> Bob,
>
> I share some of your disapointment, but I don't have Matlab speeds to
> compare to. The speed of FFTW does rely heavily on the plan, and for

> some jobs it is just as well to stick with the IDL FT. I suspect the
> speed advantage will vary quite a bit with the data and hardware at
> hand.
>
> Here's a plot of a quick test on my machine, using the fftw_one
> function and IDL's FT function on a complex array of various lengths:
> http://people.uleth.ca/~brad.gom/fftw/new-3.png
>
> Here's the actual times as a function of length. Red is the FFTW.
> http://people.uleth.ca/~brad.gom/fftw/new-1.png
>
> Arrays less than 2^16 are faster in IDL.
> These data sets were all powers of 2 in length, and the trends will be
> different for non power of two lengths. For example, here are the
> results for arrays of (2^n)+1 in length:
>
> http://people.uleth.ca/~brad.gom/fftw/new-4.png
> http://people.uleth.ca/~brad.gom/fftw/new-5.png
>
> and for all lengths between 10 and 110:
>
> http://people.uleth.ca/~brad.gom/fftw/new-6.png
> http://people.uleth.ca/~brad.gom/fftw/new-6.png
>
> I haven't dug into my DLM to see where time is being wasted, but it
> seems as though you still have to carefully consider the size of data
> going into your FT routine if you want the best performance, no matter
> which routine you use..
>
> Brad

Hi Brad,
good info, it is interesting to see those plots.
In my work I use a lot of S-Transforms to calculate the
local spectrum (as a function of time), and it employs a
lot of FFT calls. But I also use it for arbitrary lengths of time
and the requirement of the fftw to calculate a plan for each different
length is a real killer. Plus, the creation of the plan can be very very
slow (i.e. minutes). I also use complex data and complex spectra, so
I can't take advantage of the fftw ability to return only the positive
frequency half of the spectrum.

Oddly enough the matlab invokation of fftw does not seem to have that step
(of calculating the plan).
Hopefully IDL will incorporate it into a future version, because the order
of magnitude speed increase that the matlab example suggests would be quite
nice to have.

Cheers,
bob

Subject: Re: dissapointing fftw
Posted by stevenj on Wed, 11 Feb 2004 05:08:37 GMT
View Forum Message <> Reply to Message

"R.G. Stockwell" wrote...
> [...] But I also use it for arbitrary lengths of time
> and the requirement of the fftw to calculate a plan for each different
> length is a real killer. Plus, the creation of the plan can be very very
> slow (i.e. minutes). [...]
> Oddly enough the matlab invokation of fftw does not seem to have that step
> (of calculating the plan).

You can create plans with the FFTW_ESTIMATE option to skip the
plan-optimization step and just pick one based on heuristics. I think
that's what Matlab does, probably with precomputed plans ('wisdom')
for a few sizes like powers of two.

Subject: Re: dissapointing fftw
Posted by R.G. Stockwell on Wed, 11 Feb 2004 17:39:11 GMT
View Forum Message <> Reply to Message

"Steven G. Johnson" <stevenj@alum.mit.edu> wrote in message
news:27cfb406.0402102108.3e4a96ae@posting.google.com...
> "R.G. Stockwell" wrote...
>> [...] But I also use it for arbitrary lengths of time
>> and the requirement of the fftw to calculate a plan for each different
>> length is a real killer. Plus, the creation of the plan can be very very
>> slow (i.e. minutes). [...]
>> Oddly enough the matlab invokation of fftw does not seem to have that step
>> (of calculating the plan).
>
> You can create plans with the FFTW_ESTIMATE option to skip the
> plan-optimization step and just pick one based on heuristics. I think
> that's what Matlab does, probably with precomputed plans ('wisdom')
> for a few sizes like powers of two.

Yes, matlab probably does something like that. In my time tests I ran the
/exhaustive plan do find the optimal fftw algorithm, then I looped through
repeated calculations on the same size random time series. This gave the
best possible times for fftw which are dissapointing when compared to idl's fft.
Presumably running the code with /patient or /estimate would be even slower
(at least not faster) than the results I got.

You bring up a good point though. For my application of arbitrary sized ffts,
I should compare the /estimate plan, and calculate the time required for those
fftw computations as compared to IDLs.

Cheers,
bob

PS I have always thought that IDL's fft was really quick, up until I saw
Matlab's fft a month or so ago.
