Subject: Re: FFT accuracy
Posted by ali on Mon, 20 Apr 1992 17:24:39 GMT
View Forum Message <> Reply to Message

In article <9204171736.AA03199@dip.eecs.umich.edu>, pan@ZIP.EECS.UMICH.EDU writes...
> I had two examples that show the problem of numerical accuracy in
> using FFT of either IDL or PVWAVES. Is the FFT function
> provided by either IDL or PVWAVES is suitable for accuracy-demanding
> calculation? Should I go look for a double precision FFT?
> Any comment is appreciated? -tinsu pan
>

For the examples shown, IDL computes the FFTs to the best accuracy attainable with single precision floating point. The differences between IDL and WAVE are simply due to differences in the floating point implementation on the IBM RS-6000 and the DECStation.

The examples would more properly be written:

```
  x = findgen(256,256)
  y = float(fft(fft(x,-1),1))
  err = abs(x-y)
  print, 'Maximum relative error: ', max(err,i) / x(i)
Maximum relative error:   1.83841e-07
  print, 'Total relative error: ', total(err)/total(x)
Total relative error:   4.21436e-08
```

For 512 x 512 the errors are:
Maximum relative error: 1.34636e-06
Total relative error: 8.42841e-08

These examples, run on a SPARC show the worst case error to be well within the expected IEEE single precision significance of between 6 to 9 decimal digits.

A double-precision FFT would, of course, provide better accuracy, but is not provided because there is no double precision complex data type.

There was, however, a much worse error in the FFT that occured with arrays whose dimensions were larger prime numbers. This error has been fixed in IDL and the revised code will appear in IDL Version 2.3, which will be released shortly.

Here is an excerpt from the release notes of IDL Version 2.3:

The fast Fourier (FFT) function produced unnacceptable errors when working on arrays with dimensions that had large PRIME factors. There was no problem with dimensions that are a power of two, three, etc., or whose largest prime factor is less than those discussed below .

For example, a forward/inverse transform on an array with 1181 elements produced unrecognizable results. The problem diminished with smaller prime factors. The problem has been fixed, at the expense of longer execution time. The inaccuracies became apparent when the largest prime factor of a dimension was over approximately 227.

Arrays with dimensions whose largest prime factor is less than approximately 100 presented no problem. Execution time remains unchanged for this type of arrays.

Examples: Array with a dimension of:
260 (2x2x5x13) no problem
261 (3x3x29) no problem
262 (2x131) moderate problem
263 (1x263) problem
264 (2x2x2x3x11) no problem
265 (5x53) no problem

- Research Systems, Inc.
