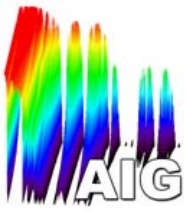

Performance Test Report
for the
SPIRE Interactive Analysis - WP Fourier Transformation

SPIRE-UOL-REP-002220

Prepared by:

Peter Davis (University of Lethbridge)
peter.davis@uleth.ca

Andres Rebolledo (University of Lethbridge)
andres.rebolledo@uleth.ca

Trevor Fulton (University of Lethbridge)
trevor.fulton@uleth.ca

Approved by:

David Naylor (University of Lethbridge)
naylor@uleth.ca

Document History:

Issue	**Date**
Version 1.0	November 19, 2004

Acronyms

HK	HouseKeeping
IA	Interactive Analysis
iNFFT	iterative Non-uniform Fast Fourier Transformation
MPD	Mechanical Path Difference
OPD	Optical Path Difference
RMS	Root Mean Square
SMECT	Spectrometer Mechanism Timeline
SDI	Spectrometer Detector Interferogram
SDS	Spectrometer Detector Spectrum
SDT	Spectrometer Detector Timeline
ZPD	Zero Path Difference

Purpose of this document:
This document details the test procedure and results to show that the task Fourier Transformation was efficiently implemented in Java giving accurate results within the framework of the SPIRE Interactive Analysis (IA). Please refer to "*SPIRE ICC Consolidated WorkPlan*", prepared by Ken King, for the definition of the functionality to be provided by the WP Fourier Transformation. The "*Technical Note on the ICC Work Package Fourier Transformation*", SPIRE-UOL-NOT-002204, details the approach to implementing the Fourier Transformation for the SPIRE imaging FTS. Please refer to the flow chart in the appendix A for a structural overview of all involved processing steps.

Table of Contents:

Reference data for testing

Data and code used for the SPIRE Science Team meeting in September 2004 at RAL were taken as reference to test the Java implementation. The base spectrum is given in Figure 1. Its inverse Fourier Transform is a finely and regularly sampled interferogram. To simulate the performance of SPIRE, jitter was applied to the speed of the simulated mirror mechanism. The form of this jitter was based on that observed in the SMEC test data provided by LAM. The amplitude of the stage jitter was either set to 0.3% or 3% of the average stage speed. A finely sampled jittered stage position timeline was computed from the simulated stage speed timeline and a finely sampled master timeline. The jittered stage position timeline was then sampled at 80Hz and 250Hz to create the simulated detector and stage position timelines, I(t´) and z(t) respectively, that are to be expected from SPIRE.

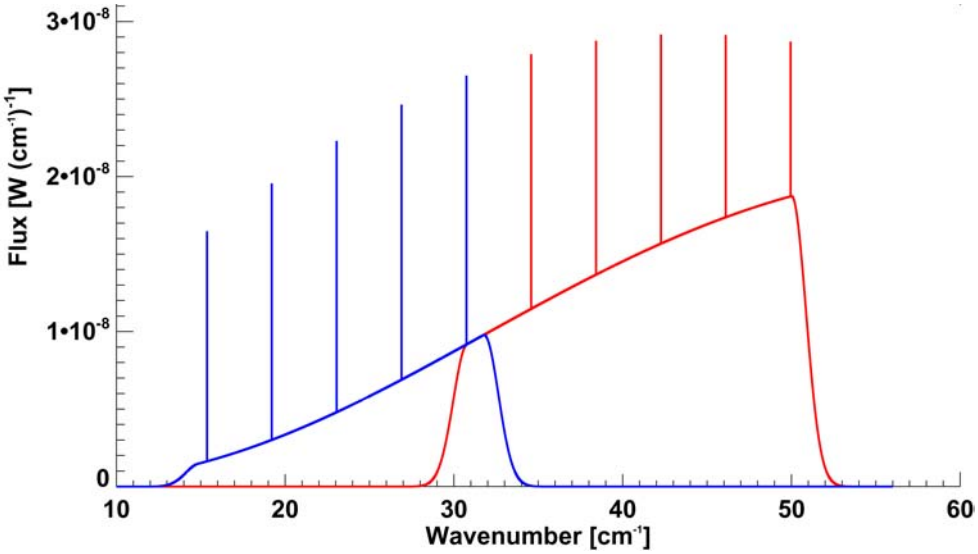

Figure 1: Simulated test data for the SPIRE FTS data reduction for the two SPIRE FTS bands: 20K black body continuum; emissivity of β = 1.5; 300K CO lines. The spectra for the short and long wavelength bands are given in red and blue respectively.

CreateRegSampledInterferogram

Description: Takes bolometer signals I(t') from SDT Product and time-sampled stage positions z(t) from SMECT and interpolates z(t) -> x(t'')[1] and I(t') -> I(t´´) to create a interferogram I(x) regularly sampled in optical path difference. Uses cubic spline interpolation. Inherits execute() method from CreateInterferogram, defines interpolate() method.
Extends: CreateInterferogram
Task Parameters: SDT Product, SMECT Product, HK Product, Interpolation Type
Result: SDI Product
Task Execution Steps:

1. Create regularly gridded mpd axis, x (includes zpd)
2. Truncate x so it does not exceed range of stage travel.
3. Truncate x such that it contains an even number of points: (n/2) left of zpd, zpd, and (n/2)-1 right of zpd.
4. Convert to OPD (x2 TFTS, x4 SPIRE, subtract ZPD)
5. For each scan:
    a. If this is a backwards scan reverse z(t) (required by CubicSplineInterpolator)
    b. Interpolate the irregularly sampled position timeline z(t) to create a regularly sampled position timeline x(t´´)
    c. Interpolate the signal timeline I(t') onto the timeline t´´ to create I(x)
    d. Append to interferogram product

Test Procedure:
Java and IDL were fed with identical timelines, bolometer signal as a function of time I(t') and irregular stage position as a function of time z(t). Two data sets are used for testing, one with 0.3% stage jitter and one with 3% stage jitter. The timelines t and t´ were expressed in double precision as units of clock ticks (1/312500 s) and z in μm.

Stage position interpolation:

The stage position interpolation uses a cubic spline to interpolate z(t) onto regularly sampled position grid x(t´´). The regularly sampled position grid x is calculated using the sampling frequency of the bolometers, 80Hz, and the starting and end position of the stage. ZPD is known and included as a point in the x grid.

Figure 2 and Figure 3 show the spectra, created from interferograms with stage position timelines with 0.3% jitter and 3% jitter respectively:

[1] The convention in this report is to use z for an irregularly spaced position grid and x for a regularly spaced position grid.

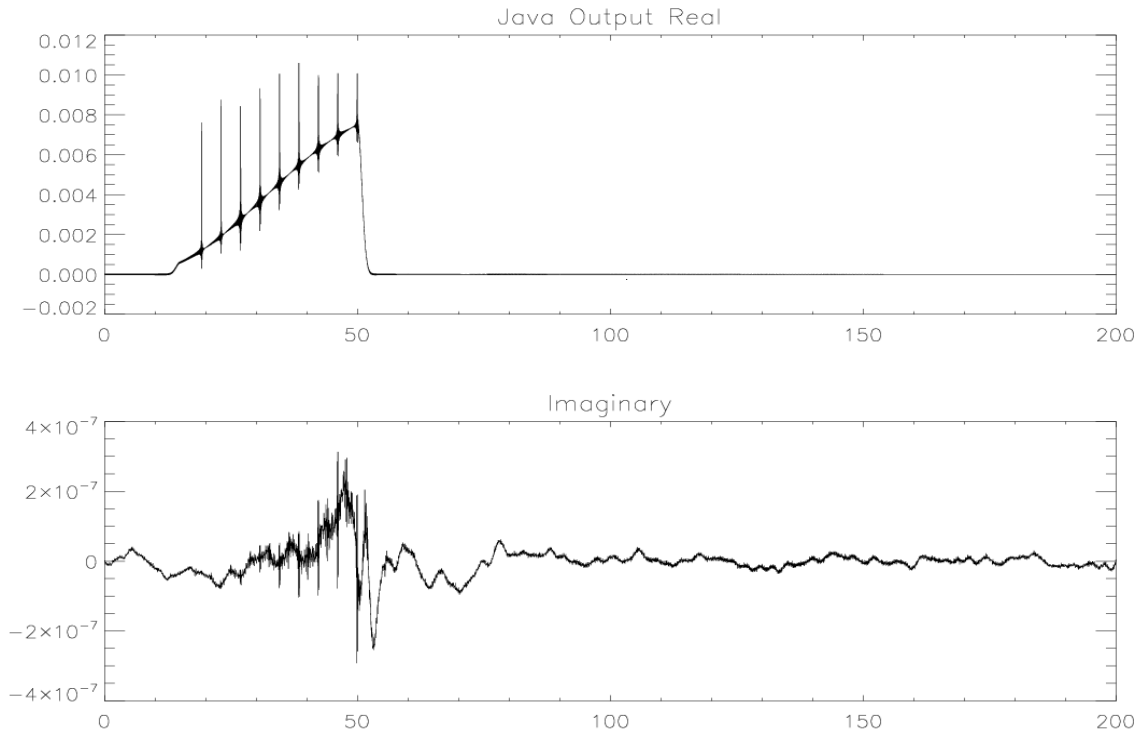

Figure 2: Resulting spectrum, 0.3% stage jitter; total power in the imaginary: 1.2E-4

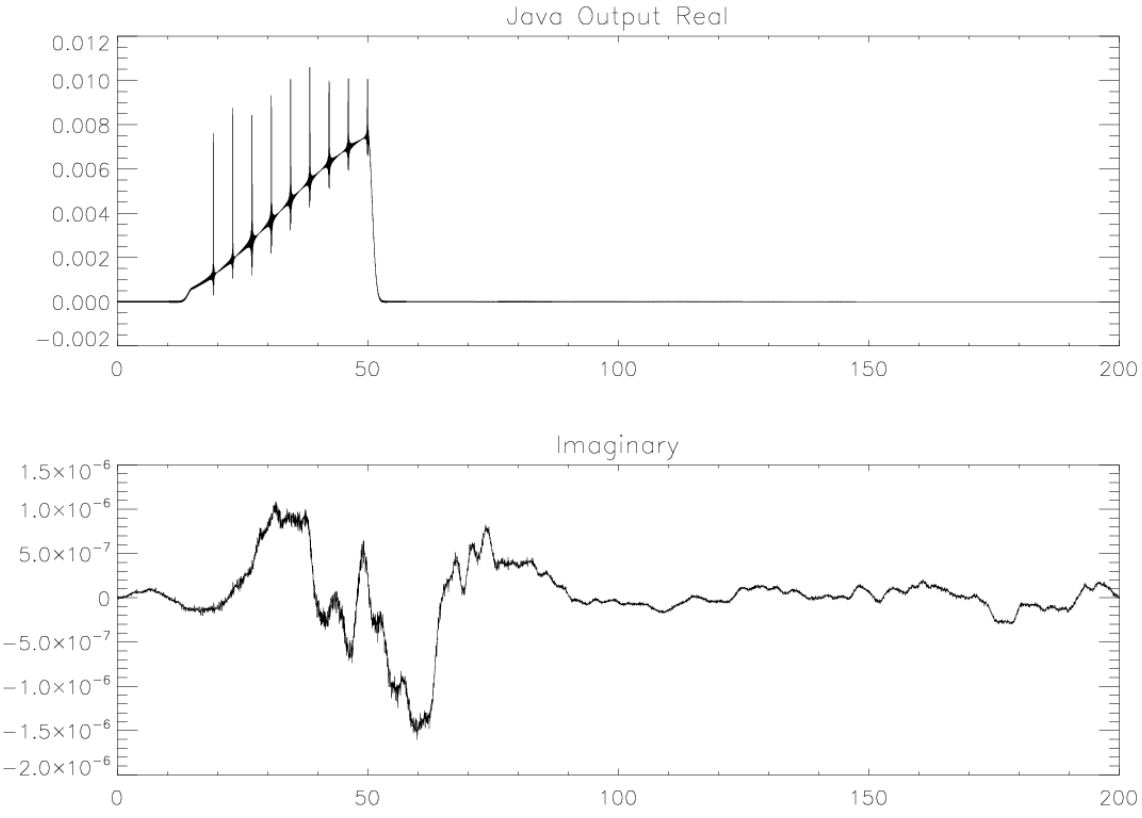

Figure 3: Resulting spectrum, 3% stage jitter; total power in the imaginary: 1.1E-3

Comparison of Figures 2 and 3 show that a ten-fold increase in stage jitter leads to an increased amount of power in the imaginary part of the resulting spectrum. Also, a considerable amount of that power extends further out in the spectral domain beyond the detection band. The overall power in the imaginary part scales approximately with the amplitude of the introduced jitter.

Signal interpolation:

The signal interpolation step computes the signal I(t) onto the new t´´ grid with a cubic spline fit to create the regularly sampled interferogram.

The test data are derived under the assumption that the stage is subject to considerable deviations from the perfect speed. It is therefore not possible to specify at exactly what moments in time the stage passed the points in the regular position grid x. Considerable oversampling of the initial test data could soften this limitation. This option was not pursued due to time restrictions. An absolute measure of the quality of interpolation is therefore not available.

In lieu of an absolute comparison, a relative comparison of two different methods of interpolation was undertaken. Notwithstanding that the CubicSplineInterpolator in IA and IDL SPLINE interpolator are implemented slightly differently, it is still instructive to compare the impact of these two spline interpolations on the resulting, regularly sampled interferograms. Table 1 shows the difference between the regular-position timestamps as calculated by the two interpolation methods.

(A) position - signal interpolation	(B) position - signal interpolation	RMS(A-B) / central maximum
Java - IDL	IDL - IDL	8.73e-010
IDL - Java	Java - Java	8.82e-010

Table 1: RMS of the differences of the timestamps between interferogram samples with different implementations of the spline interpolation for the stage position interpolation, (stage jitter at 0.3%)

The order of magnitude of the effect of the stage position interpolation scheme on the RMS can be compared to the effect of swapping the signal interpolation scheme in Table 2.

(A) position - signal interpolation	(B) position - signal interpolation	RMS(A-B) / central maximum
Java – IDL	Java - Java	1.31e-005
IDL – Java	IDL - IDL	1.31e-005

Table 2: RMS of the differences of the timestamps between interferogram samples with different implementations of the spline interpolation for the signal interpolation, (stage jitter at 0.3%)

Table 1 shows the effect of the different spline implementations for the stage position interpolation (z(t´) → z(t´´)). The RMS of the differences is in both cases on the order of 7e-13. Table 2 shows the effect of the different spline implementations for the signal interpolation (I(t) → I(t´´)). The RMS of the differences is in both cases on the order of

1e-8. The error introduced by the second interpolation is larger by a factor of 10 000 in comparison to the first interpolation. Clearly, and as expected, the critical interpolation is the second interpolation of the signal.

Open points:

- It may be possible to significantly improve on the signal interpolation and the stage interpolation. A windowed-sinc interpolation scheme would be preferable. However, problems due to the irregular sampling of the data make the windowed-sinc interpolation much more difficult to implement. Trevor will report his work in this area in a respective presentation.
- The time offset for the signal and position timeline due to the AD conversion has not been taken into account so far.
- The dependence of OPD on the off-axis angle is not taken into account as of now. The best way of doing this is TBD. What are the relevant angles per pixel? Is it preferable to stretch the spectra or the interferograms?
- Neither the ZPD values nor the MPD -> OPD conversion calibration tables have been implemented yet.
- Can a useful signal error be specified from the interpolation?
- Is the accuracy of the position measurements a function of the stage position?
- The stage interpolation could be improved by changing the interpolation parameters depending on whether the interpolation takes place around the central peak or not.

RegSampledFT

Description: FFT on regular sampled interferograms. Performs either doublesided or singlesided FFT. Inherits execute() method from CreateInterferogram, defines ifgmMethod().
Extends: InterferogramTask
Task Parameters: SDI Product, FT type
Result: SDS Product
Task Execution Steps:

1) For each scan:
    a. If double-sided FT and interferogram is single-sided, use the double-sided portion.
    b. If single-sided FT and interferogram is single-sided, use the side of the interferogram containing more points.
    c. Butterfly
    d. Perform FFT
    e. If this is the 1st scan, calculate wavenumber grid based on stage travel
    f. Store resulting spectrum in SDS

A regularly sampled interferogram, with no stage jitter at all, should be perfectly symmetric and a Fourier Transformation should put all the power into the real part and no power at all into the imaginary part. Figure 4 shows the actual results:

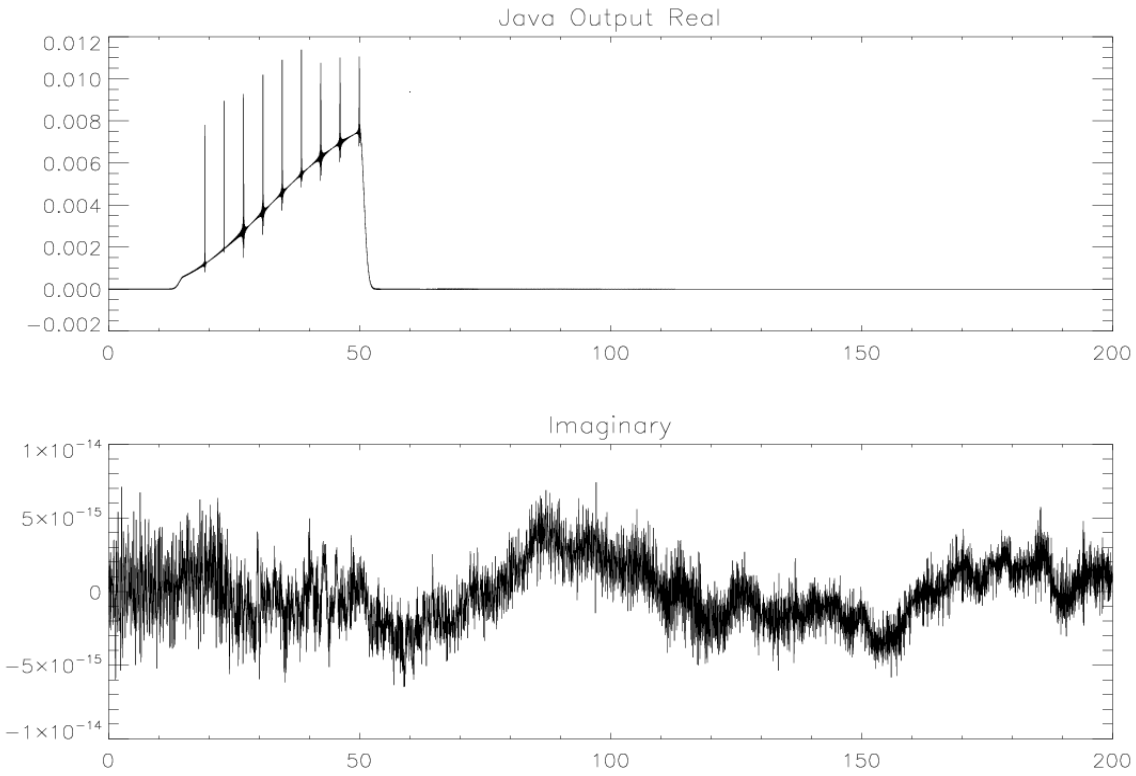

Figure 4: The real and imaginary parts of the spectrum, no stage jitter applied; total power in the imaginary part: 9.03290e-012

The power in the imaginary part is smaller than the power in the real part by 18 orders of magnitude, which suggests that the implementation of the Fourier Transformation is limited by the double precision floating-point error of the computer system.

Spectra of Java and IDL Interferograms:

The overall accuracy of the processing pipeline was determined by comparing the calculated spectra with the expected spectrum. A fitting routine, designed to extract the relevant scientific parameters (blackbody temperature, optical density factor (β), line centres and line amplitudes), was applied to each of the calculated. Table 3 shows a comparison between the extracted scientific parameters for the spectra calculated using the CreateInterferogram task (Interpolation & FFT) and those from the spectra derived using an iterative Fourier Transform method (iNFFT).

Processing Scheme	Jitter	BB Temp (K)	Stdev BB Temp (K)	β	Stdev β
Theoretical	n/a	20	n/a	1.5	n/a
Fitted	0	19.99923	n/a	1.500047	n/a
Interpolation & FFT	0.3	19.99624	0.00294	1.500748	0.000635
Interpolation & FFT	3	19.99810	0.00561	1.500504	0.000988
iNFFT	0.3	19.99950	0.00007	1.500047	0.000006
iNFFT	3	19.99576	0.00358	1.500260	0.000660

Processing Scheme	Jitter	Avg Line Centre Difference (per Resolution Element)	Stdev Line Centre Difference (per Resolution Element)	Avg Amplitude Difference (%)	Stdev Amplitude Difference (%)
Theoretical	n/a	n/a	n/a	n/a	n/a
Fitted	0	0.00318	n/a	0.340	n/a
Interpolation & FFT	0.3	0.00255	0.00271	0.327	0.240
Interpolation & FFT	3	0.00194	0.00201	0.393	0.231
iNFFT	0.3	0.00285	0.00312	0.301	0.251
iNFFT	3	0.00632	0.01015	1.051	1.038

Table 3: Comparison of spectral features derived from differently processed spectra, both bands

The spline interpolation yields differences for T and β in the range of less than 0.1%. The line features are within the precision of the fitting routine. The iterative FT scheme is better when compared to the spline interpolation by about a factor of 10 for the continuum parameters.

PhaseCorrectRegSampled

Description: Corrects phase shift caused by dispersive elements (e.g. optics, electronics). Inherits execute() method from InterferogramTask, defines ifgmMethod().
Extends: InterferogramTask
Task Parameters: SDI Product, SDS Product (created from double-sided RegSampledFT)
Result: SDI Product
Task Execution Steps:

1) Calculate phase of double-sided spectrum.
2) Calculate weighted fit to phase, $\varphi(\sigma)$, within the observation band to a 1st (or higher) degree polynomial.
3) Calculate phase function $e^{-i\varphi(\sigma)}$
4) If interferogram is double-sided:
    a. Multiply spectrum with phase function to correct phase.
    b. Compute FFT^{-1} of phase corrected spectra to create interferogram.
    c. Replace old interferogram in SDI with phase corrected SDI.
5) Else interferogram is single-sided:
    a. Butterfly phase function, making imaginary portion anti-symmetric.
    b. Compute FFT^{-1} of phase function.
    c. Truncate phase function to create PCF.
    d. If PCF apodization is requested, apodize the PCF.
    e. Convolve the PCF with the original interferogram to correct the phase.
    f. Replace old interferogram in SDI with phase corrected SDI.

Double-sided Phase Correction

The reference spectrum from the SLW band with 3% jitter was used as our test case. To introduce a linear phase shift, we created our interferogram using an x grid that missed ZPD by 0.5μm in MPD (2μm OPD). A straightforward double-sided FT produced the following:

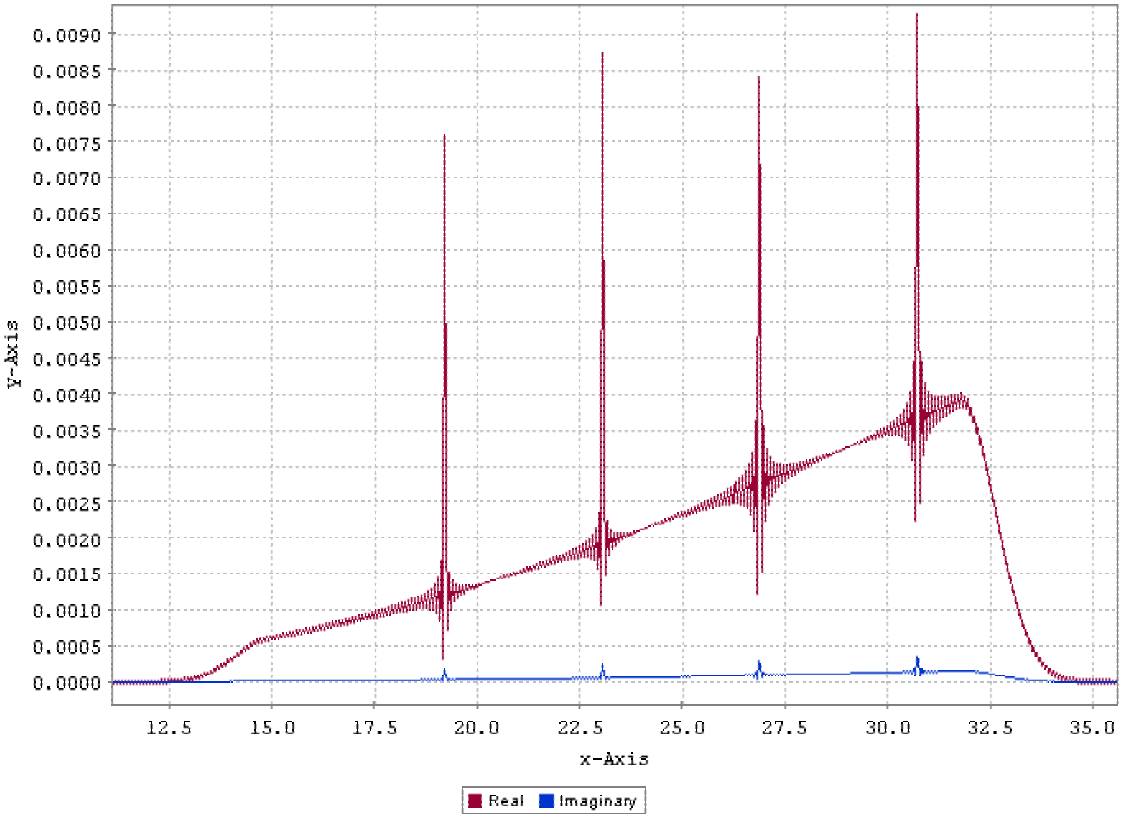

Figure 5: FT of interferogram with a linear phase shift

The phase of this spectrum was calculated. Within the passband, from 15 to 32 cm^{-1}, the expected linear phase shift is noticeable (see Figure 6). The phase within the band was used to calculate a weighted fit to a 1st degree polynomial. The weights were calculated using ABS(spectrum in band)*0.01. This yields the following phase $\varphi(\sigma)$:

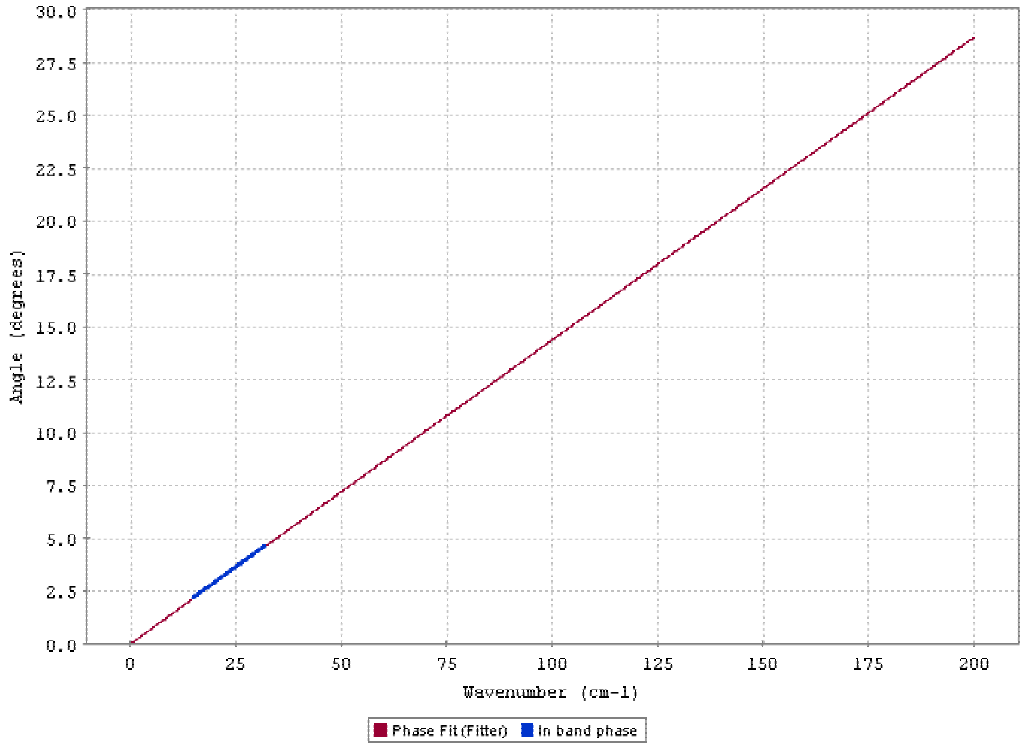

Figure 6: In band phase and fit phase expressed by polynomial 2.66e-5 + 0.00126x

The phase correction function is defined as PCF = $e^{-i\varphi(\sigma)}$. Multiplying the spectrum by the phase correction function produces the phase corrected spectrum. The following is the in-band phase of the corrected spectra:

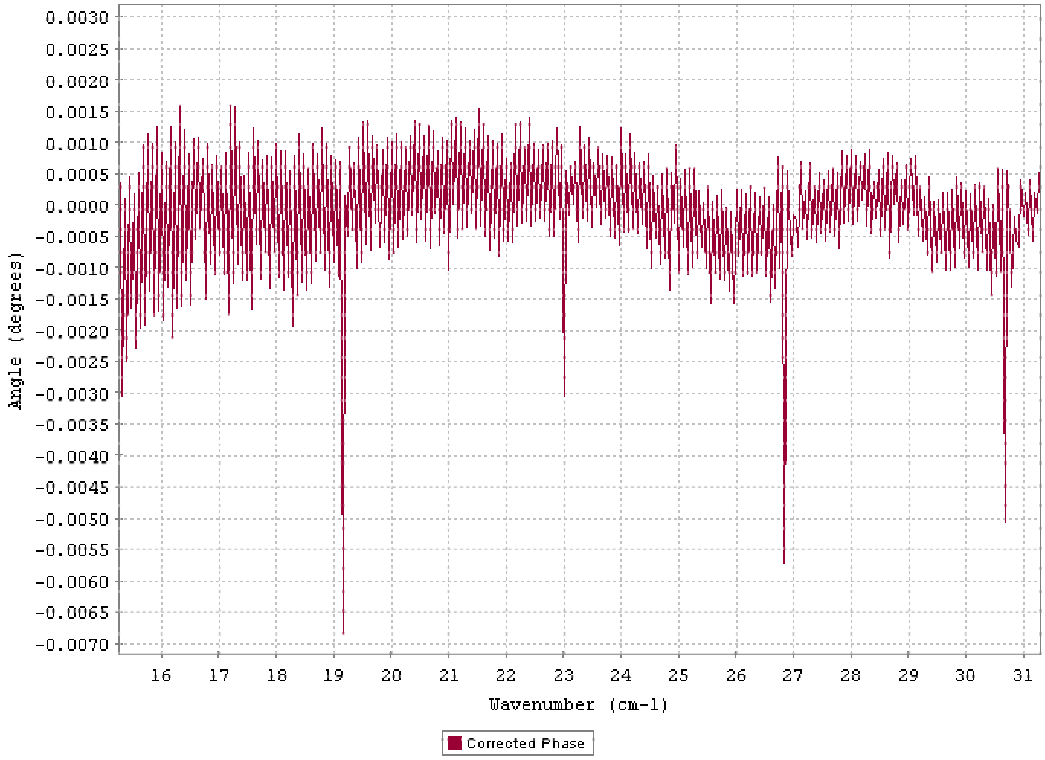

Figure 7: The corrected phase within the detection band

The corrected phase is less than 1/10000 of a radian. The spikes in phase correspond to the lines in the spectrum. The shape of the in-band phase after removing the linear phase can be compared to the in-band phase of the double-sided FT without introduced phase error (see Figure 8):

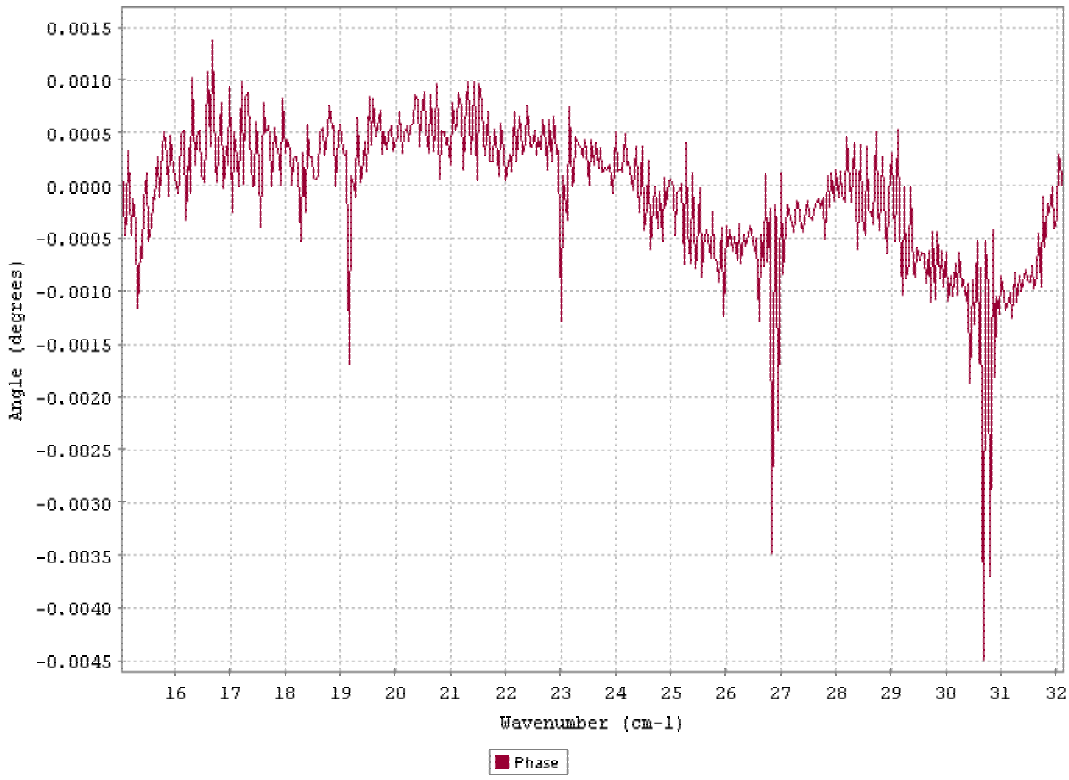

Figure 8: Phase of double-sided FT, no phase error introduced

The artificially created phase can be removed to a large degree with the performed procedure. The overall amplitude of the phase is slightly increased. A comparison between Figures 7 and 8 shows that no additional non-linear structure appears to have been added to the phase.

Single-sided Phase Correction

Phase correction for single-sided interferograms (Figure 9), while mathematically equivalent to that for double-sided interferograms, differs from a procedural point of view. Since, in the single-sided case, the uncorrected spectral content is known only for the short portion of the interferogram about ZPD, direct correction by multiplication in the spectral domain cannot be performed. Owing to the fact that multiplication in one domain is equivalent to convolution in the other domain, single-sided phase correction takes place in the spatial domain by convolution with a phase correction function. The phase correction function for single-sided interferograms is derived by first finding the phase of the short double-sided portion of the interferogram about ZPD. As with double-sided phase correction, a weighted function is fit to the measured phase. The single-sided PCF is simply the inverse Fourier Transform the fitted phase function ($PCF=FT^{-1}(e^{-i\varphi(\sigma)})$, see Figure 11). Convolution of the original single-sided interferogram with this PCF

results in a phase-corrected interferogram. The spectrum is then calculated by transformation of the phase corrected single-sided interferogram.

Note that the phase corrected interferogram may then be apodized as in Figure 12 prior to transformation in order to smooth any rippling which may occur about the spectral lines (Figure 13)

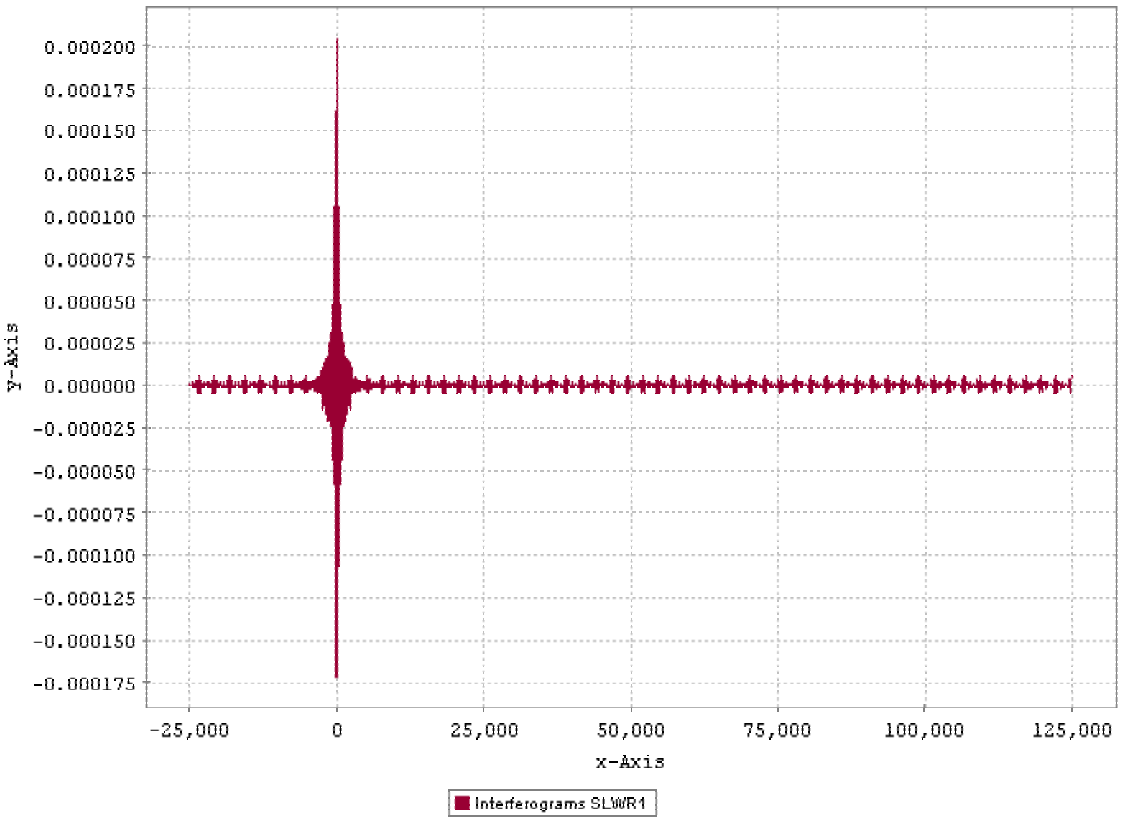

Figure 9 Single-sided interferogram. Note the short double-sided portion about ZPD

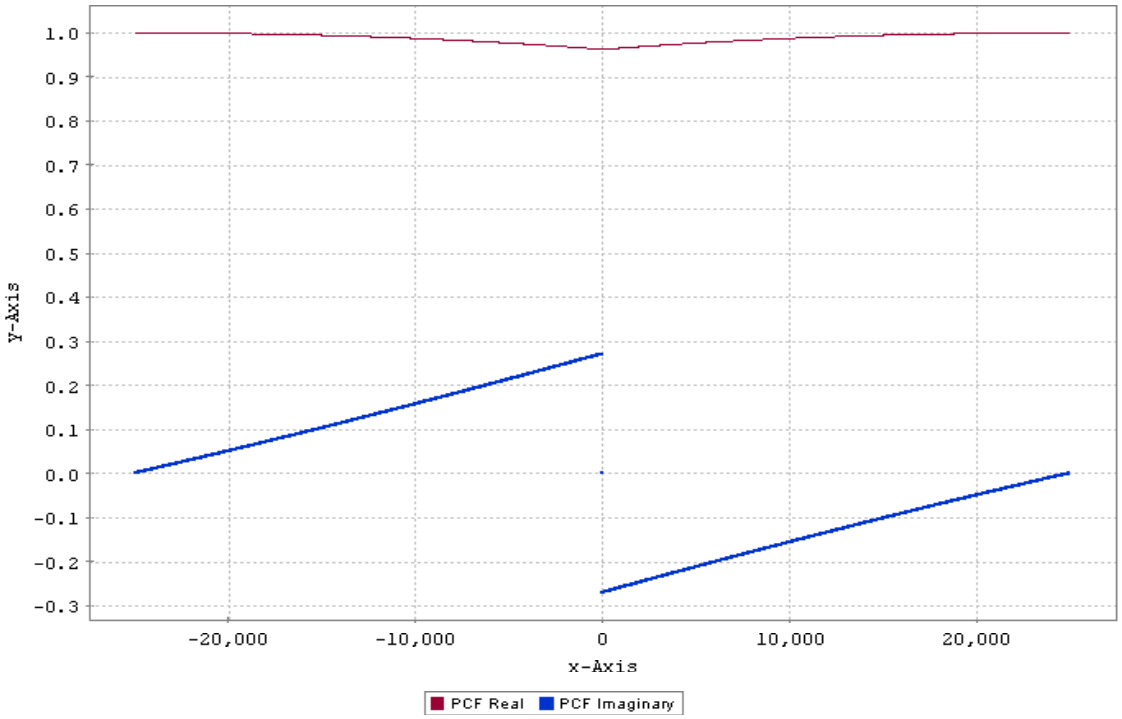

Figure 10: Butterflied $e^{-i\varphi(\sigma)}$

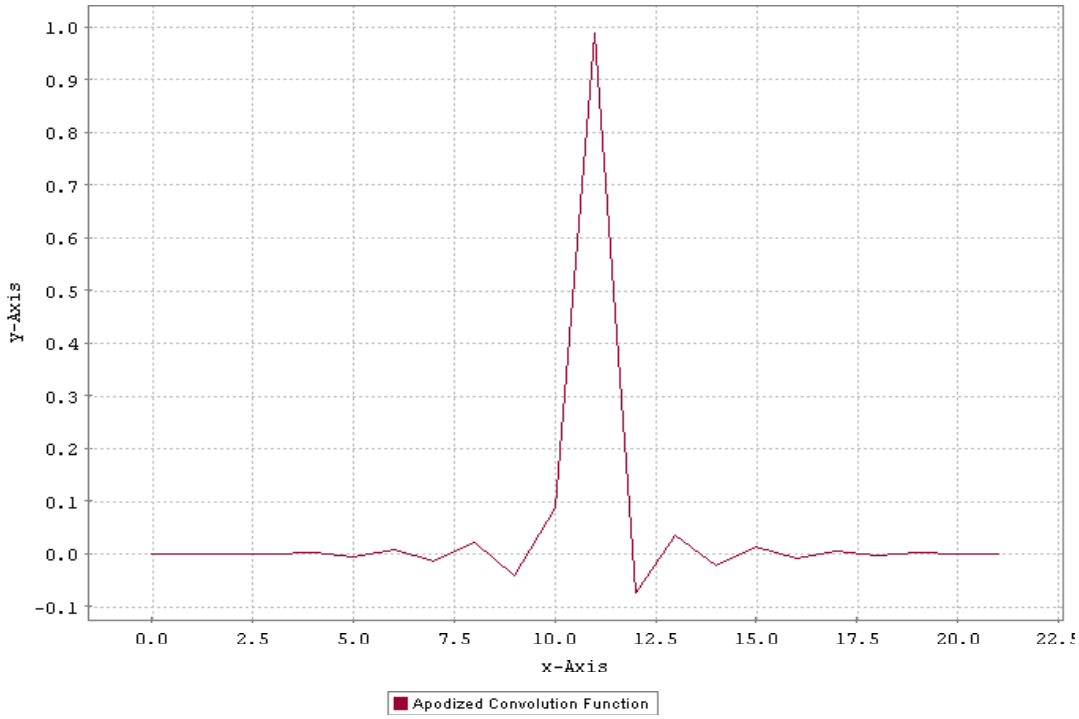

Figure 11: 22-point single-sided phase correction function apodized using Norton-Beer 1.9 FWHM

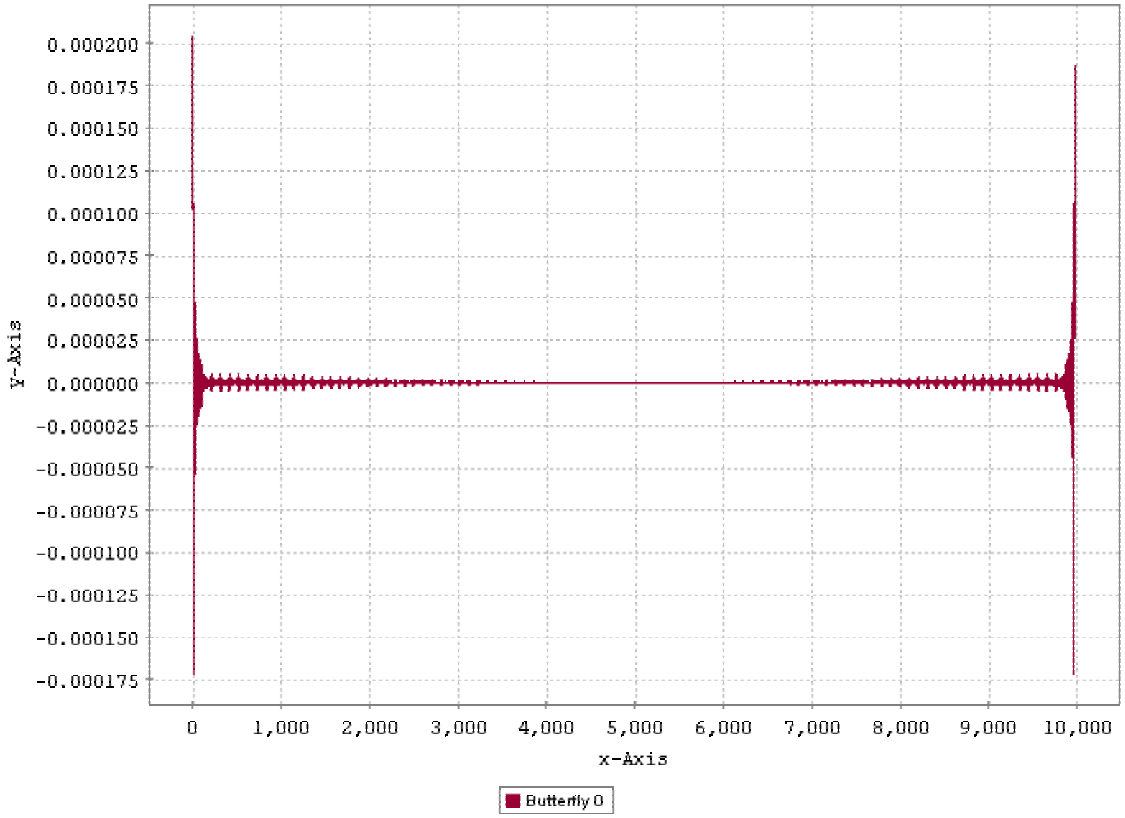

Figure 12: Apodized (NB 1.9 FWHM), phase corrected, and butterflied single-sided interferogram

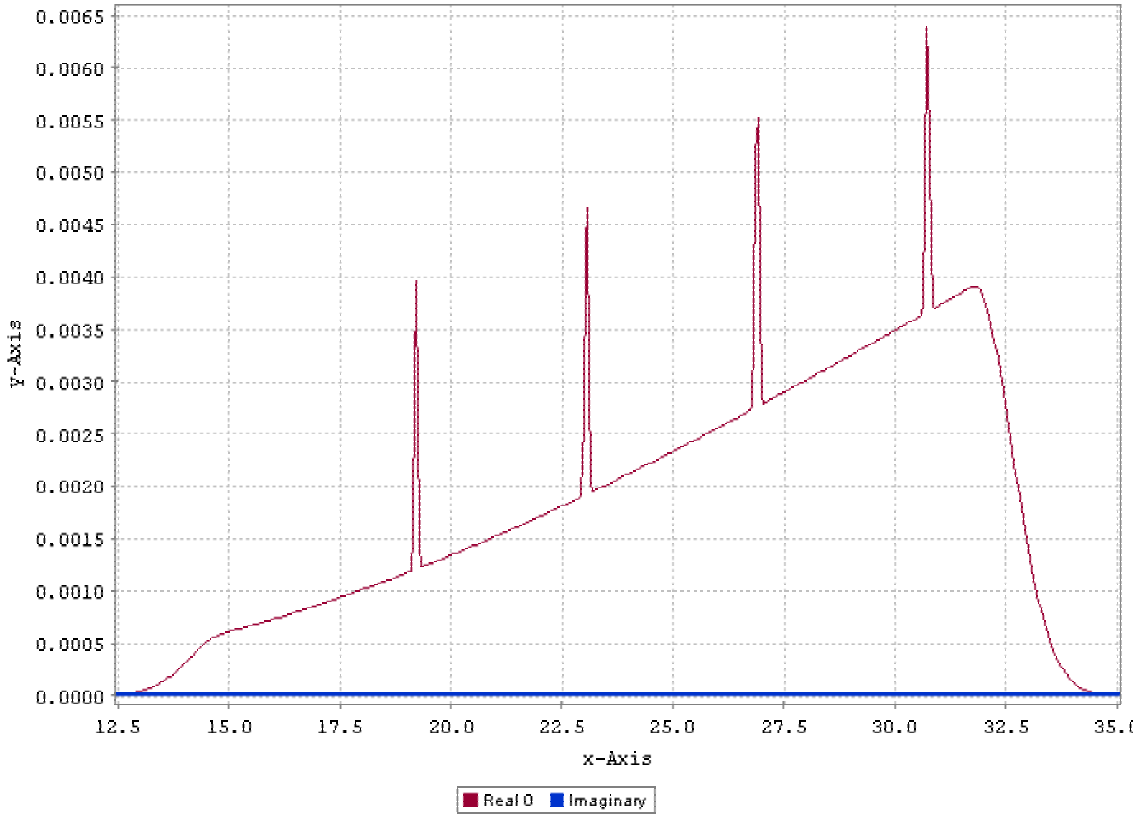

Figure 13: Apodized and phase corrected spectrum

Open points:

1) What is the ideal number of points to truncate the PCF to?
2) What apodization, if any, should be used on the PCF?
3) How should the PCF be convolved? In the current code, the convolution leads to the loss of half the number of points of the PCF in the interferogram.

Another optional apodization can be performed on the phase corrected interferogram.

The following table shows the results of spectrum produced with and without double-sided phase correction. BB Temp, β, and RMS are produced from six runs. The line centre and amplitude measurements are produced from 6 runs x 4 lines for a total of 24 lines.

Spectra	BB Temp (K)	Stdev BB Temp (K)	β	Stdev β	Avg Line Centre Difference (per Resolution Element)	Stdev Line Centre Difference (per Resolution Element)
No phase error – No phase correction	20.00206	5.8E-5	1.49997	1.1E-05	-0.0002	0.00013
Phase error - No phase correction	20.00072	1.1E-4	1.49922	2.7E-05	-0.0002	0.00013
Phase error - Phase correction	20.00213	1.1E-4	1.49994	2.7E-05	-0.0002	0.00013

Spectra	Avg Amplitude Difference (%)	Stdev Amplitude Difference (%)	RMS (abs(spectra) - abs(reference spectrum))	Stdev RMS
No phase error – No phase correction	-0.86959	0.00013	6.44E-08	2.78E-08
Phase error - No phase correction	-0.91992	0.00013	1.77E-07	5.78E-08
Phase error - Phase correction	-0.86923	0.00013	1.83E-07	5.84E-08

Table 4: Comparison of spectra produced from double-sided interferogram, SLW band, 0.3% jitter, no apodization on interferograms

The double-sided phase correction at a stage jitter level of 0.3% can be compared to spectral results when no phase error is introduced. For the two parameters describing the fit to the continuous background (T and β), the computed values agree within one standard deviation. For the lines centres, phase error and phase correction do not appear to have a significant effect. For the line amplitudes, the agreement between the results without phase error and phase-correction differ by less than three standard deviations while the difference is close to 400 standard deviations when no phase correction is applied.

The following table shows the shows the results of spectrum produced with and without single-sided phase correction.

Spectra	BB Temp (K)	Stdev BB Temp (K)	β	Stdev β	Avg Line Centre Difference (per Resolution Element)	Stdev Line Centre Difference (per Resolution Element)
No phase error – No phase correction	20.00237	0.00011	1.49990	2.2E-05	-0.0001	7.72E-05
Phase error - No phase correction	18.92538	0.00018	1.69580	4.5E-05	-0.0100	4.02E-03
Phase error - Phase correction	20.01763	0.00048	1.49697	8.7E-05	0.0002	5.02E-04

Spectra	Avg Amplitude Difference (%)	Stdev Amplitude Difference (%)	RMS (abs(spectra) - abs(reference spectrum))	Stdev RMS
No phase error – No phase correction	-0.8828	7.72E-05	3.63E-06	1.45E-11
Phase error - No phase correction	0.9695	4.02E-03	2.49E-05	3.19E-09
Phase error - Phase correction	-1.1543	5.02E-04	1.73E-05	3.06E-10

Table 5: Comparison of spectra produced from single-sided interferogram, SLW band, 0.3% jitter, no apodization on interferograms (PCF for SS apodized with NB 1.9)

Tables 4 and 5 show that phase correction of single sided interferograms improves the results considerably. The errors for T and β are on the order of several thousands of standard deviations from the reference case if the phase error is not corrected and are reduced to roughly 30 standard deviations when phase correction is performed.

Load Testing

The pipeline was tested with a large with data to simulate a normal load. The SMECT was filled with data for 6 scans, and the SDT was filled with data for 72 pixels, for a total of 432 interferograms. Each interferogram contained 6000 points (1000 small wing, 5000 large wing). This resulted in the following FITS file sizes:

SMECT – 3.13MB
SDT – 71.49MB
SDS – 108.89MB

The tasks were timed on a 2.66 Ghz Pentium 4 machine with 512 MB RAM running Windows 2000. The processing times for each task are as follows:

Task	**Time (s)**
CreateRegSampledInterferogram	9.297
RegSampledFT (double-sided)	4.094
PhaseCorrectRegSampled	5.281
ApodizeRegSampled	0.984
RegSampledFT (single-sided)	41.922
TOTAL	**61.578**

Table 6: Load testing of the data processing step Fourier Transformation in the single-sided case