CONSIGLIO NAZIONALE DELLE RICERCHE

`MPISUB`, a Fortran subroutine for processing Martin-Puplett interferometer data

A.Simonetto

FP 10/04 October 2010

ISTITUTO DI FISICA DEL PLASMA

Associazione EURATOM-ENEA-CNR

Via Cozzi 53 - 20125 Milano (Italy)

Introduction

This program was written in response to the basic need of having a reasonably comprehensive piece of software for processing off-line data from the *KK5* Martin-Puplett interferometer [1] at JET, instead of relying on ad-hoc programs performing only very basic processing on bidirectional interferograms only. For generality, the program was written as a subroutine, making as few assumptions as possible on the data.

This report is intended as the documentation of the program. As little theory is described as needed to explain and justify data processing. The treatment of data is very close to the one extensively described in [2], which is briefly summarized.

The program relies on the input interferogram being evenly spaced in the independent variable and containing at least a small bidirectional portion, i.e. asymmetric scans must extend on the *other* side of the Zero Path Difference feature (ZPD).

In very short summary, the program optionally removes a linear baseline from the data, then locates the approximate position of the ZPD in the record, extracts the largest bi-directional interferogram and Fourier transforms it to give a low-resolution "spectrum". The phase of the transform is unwrapped and fitted with a straight line, whose slope is proportional to the location of the ZPD in the record, and whose intercept is related to the sign of the data. The phase of the low resolution "spectrum" is shifted to remove the linear dependence on wavenumber, and the real and imaginary parts of the result are further fitted with a spline, since they contain the information necessary for correcting small misalignments in the data.

Reverting to the full interferogram, a linear weighting function is applied to the doubly sampled region, together with any other apodizing functions. The interferogram is then Fourier-transformed, the linear dependence on wavenumber is removed from the phase using the values found before, thus accounting for the position of the ZPD, and optionally the fit with splines is used to compensate the (non-linear) phase variation due to misalignments. The result is inverse-transformed to obtain the corrected interferogram.

The program returns basically the high-resolution spectrum, the corrected interferogram, the position of the ZPD and an error code. Lots of more detailed information can be output on files if required.

1. Basic theory

1.1 Notation

The Fourier transform is computed using the routines *DZFFTF* and *DZFFTB* in FFTPACK [3], whose definition of Discrete Fourier Transform of a real periodic sequence $\{r_i\}$ is the standard one (see e.g. [4]):

$$DFT\left(\{r_i\}_{i=1,N}\right) = \left\{\frac{1}{N}\sum_{i=1}^{N} r_i \exp\left(j\frac{2\pi}{N}k(i-1)\right)\right\}_{k=0,\frac{(N-\varepsilon_N)}{2}} = \{A_k + jB_k\} \tag{1a}$$

$$\begin{Bmatrix} A_k \\ B_k \end{Bmatrix} = \left\{\frac{2}{N}\sum_{i=1}^{N} r_i \begin{Bmatrix} \cos\left(2\pi k(i-1)/N\right) \\ \sin\left(2\pi k(i-1)/N\right) \end{Bmatrix}\right\}_{k=0,\frac{(N-\varepsilon_N)}{2}} \tag{1b}$$

where

$$\varepsilon_N = \begin{cases} 0 & even\ N \\ 1 & odd\ N \end{cases}$$

As customary, the normalization factor N is on the forward transform, and the inverse transform is written as

$$DFT^{-1}\left(\{c_k\}_{k=-(N-\varepsilon_N)/2,(N-\varepsilon_N)/2}\right) = \left\{\sum_{k=-(N-\varepsilon_N)/2}^{(N-\varepsilon_N)/2} c_k \exp\left(-j\frac{2\pi}{N}k(i-1)\right)\right\}_{i=1,N} \tag{1c}$$

$$DFT^{-1}\left(\left\{A_k,B_k\right\}_{k=0,\frac{N-\varepsilon_N}{2}}\right)=\left\{\sum_{k=0}^{\frac{N-\varepsilon_N}{2}}A_k\cos\left(\frac{2\pi}{N}k(i-1)\right)+jB_k\sin\left(\frac{2\pi}{N}k(i-1)\right)\right\}_{i=1,N} \tag{1d}$$

The cosine discrete transform function used in the program (DCOST) omits the normalization factor *2(N-1)*:

$$DFC\left(\left\{r_i\right\}_{i=1,N}\right)_{k=1,N}=\left\{r_1+(-1)^{k-1}r_N+2\sum_{i=2}^{N-1}r_i\cos\left(\frac{\pi}{N-1}(k-1)(i-1)\right)\right\}_{k=1,N} \tag{1e}$$

and, since it is its own inverse apart from the normalization factor, double application of DFC multiplies the sequence of data by *2(n-1)*.

The complex transform routines used include the normalization, therefore the diagnostic output (see chapter 3) includes both normalized and un-normalized versions of important data.

For ease of notation, the theoretical grounds below are described in terms of the (continuous) Fourier Transform, defined (consistently with the definition above) as

$$\begin{aligned}F(s)&\equiv FT\left(f(x)\right)\equiv\int_{-\infty}^{\infty}f(x)\exp(j2\pi sx)dx\equiv\int_{-\infty}^{\infty}f(x)\cos(2\pi sx)dx+j\int_{-\infty}^{\infty}f(x)\sin(2\pi sx)dx\equiv\\&\equiv FC\left(f(x)\right)+jFS\left(f(x)\right)\end{aligned} \tag{2}$$

$$\begin{aligned}f(x)&\equiv FT^{-1}\left(F(s)\right)\equiv\int_{-\infty}^{\infty}F(s)\exp(-j2\pi sx)ds\equiv\int_{-\infty}^{\infty}F(s)\cos(2\pi sx)ds-j\int_{-\infty}^{\infty}F(s)\sin(2\pi sx)ds\equiv\\&\equiv FC\left(F(s)\right)-jFS\left(F(s)\right)\end{aligned} \tag{3}$$

1.2 Model of measured quantity

Only a basic description of instrument theory is given here. See any text on Fourier spectrometry (e.g. [5] or [6]) for an in-depth treatment.

The Martin-Puplett interferometer measures the *difference* in radiation intensity (i.e. the modulus of the Poynting vector) reaching the detector *from the two input ports* as the moveable arm is scanned. This quantity is averaged over a time constant that is very large with respect to the path delay. Using a scalar field description (e.g. for linear polarization) and assuming that the field is stationary and that the inputs at the two ports are mutually incoherent, one can write for either port *i* (using if necessary the Wiener–Khintchine theorem for non-integrable noise-like fields)

$$\begin{aligned}\left\langle p_i(x)\right\rangle&=\left\langle\left(E_i(t)+E_i\left(t+\frac{x}{c}\right)\right)\left(H_i(t)+H_i\left(t+\frac{x}{c}\right)\right)\right\rangle=\\&=\left\langle E_i(t)H_i(t)\right\rangle+\left\langle E_i\left(t+\frac{x}{c}\right)H_i\left(t+\frac{x}{c}\right)\right\rangle+\frac{1}{2}FT^{-1}\left[\mathrm{Re}\left(E(s)H^*(s)\right)\right]\end{aligned} \tag{4}$$

that is,

$$\begin{aligned}V(x)-V_\infty\propto I(x)+I_\infty&\equiv\frac{1}{2}FT^{-1}\left[\mathrm{Re}\left(E_1(\sigma)H_1^*(\sigma)\right)-\mathrm{Re}\left(E_2(\sigma)H_2^*(\sigma)\right)\right]=\\&=\frac{1}{2}FT^{-1}\left[S_1(\sigma)-S_2(\sigma)\right]\equiv\frac{1}{2}FT^{-1}\left[S(\sigma)\right]\end{aligned} \tag{5}$$

where the subscript ∞ stands for the limit value for large path difference, i.e. the baseline of the interferogram, σ=ν/c is the wavenumber, and S(σ) is of course the measured spectrum. If the detector is not bolometric but uses direct rectification with square law response, (4) and (5) should be written using E.E instead of E.H, and the definition of *S(σ)* should be changed accordingly. There is of course no difference in formulation (apart from the factor 1/Z) for plane waves, and therefore for integrable fields, that allow such decomposition.

The measured AC-coupled signal $V(x)-V_\infty \div I(x)-I_\infty$ is therefore proportional to the Fourier transform of the difference between spectra at the two ports, one of which is normally terminated on a stationary reference load.

For an ideal instrument, (5) is a real even function of *x*, therefore $S_i(\sigma)$, $S(\sigma)$ are real, and one could as well write everything using the cosine transform

$$V(x) - V_\infty \propto I(x) + I_\infty = \frac{1}{2} FC^{-1}\left[S(\sigma)\right]. \tag{6}$$

The program does not include the factor 1/2 and the normalization 1/N, since a calibration function shuld be applied anyway to the data.

1.3 Location of Zero Path Difference feature ("Sampling error")

The interferogram at zero path difference (x=0) takes the value

$$I(0) - I_\infty = \int_{-\infty}^{\infty} S(\sigma) d\sigma \tag{7}$$

which is known as the Zero Path Difference feature, normally the maximum value of the interferogram.
The location of this point is usually unknown beforehand, at least to the accuracy required, and almost certainly not coincident with a sampling position. If ζ is the ZPD location in the laboratory reference frame for path difference, one can write, using the shift theorem

$$S_\zeta(\sigma) = FT\left(I(x-\zeta) - I_\infty\right) = \exp(j2\pi\sigma\zeta) S(\sigma). \tag{8}$$

Therefore, if one collects at least a small portion of interferogram on either side of the ZPD in order to apply (5), one can compute ζ from the slope of the phase of the complex FT of the interferogram (named for short "*complex low resolution spectrum*" in the following, even if this is nonsense in physical terms).

1.4 Finite measurement range ("Truncation error")

Of course the interferogram is available only over a finite range $[-X_1,X_2]$ with respect to the ZPD position. Following [2], one can assume that $(0<)X_1 \leq X_2$. One can describe the measured signal as the product between the so-called *scanning function*, in this case a shifted boxcar, due to the finite scan range and a hypothetical interferogram extending to infinity (symmetric as in an ideal instrument, if one assumes that the effect of misalignments was removed as shown in the next section).
The FT of the measured signal is the convolution between the spectrum of the infinite interferogram and the FT of the scanning function (*Instrument Line Shape, ILS*):

$$FT\left[\left(I(x) - I_\infty\right)A(x)\right] = S_0(\sigma) * ILS(\sigma) = \int_{-\infty}^{\infty} S_0(\sigma') ILS(\sigma - \sigma') d\sigma' \tag{9}$$

where $A(x)$ is the scanning function, and of course $ILS(\sigma)$ is the instrument line shape.
The scanning function can be decomposed in full generality in an even part (giving rise to the real part of the ILS) and a non-zero odd part due to the shift, giving rise to a finite imaginary part.
The even part of the shifted boxcar (see figure 1) is the sum of two (centered) boxcar functions, a narrow one, corresponding to the double-sided portion of interferogram $[-X_1,X_1]$ in the notation used above, and a wide one $[-X_2,X_2]$, corresponding to a double-sided measurement of the full extent of the interferogram. The odd part essentially provides cancellation of the part of the record that is not measured.
The real part of the ILS is therefore a sum of two *sinc* pulses, a wide one corresponding to the narrow boxcar, and a narrow one that is what one wants. Mis-location of the narrow boxcar function causes distortion (as a consequence of the shift theorem) of the wide *sinc* pulse, and this normally appears as baseline distortions.

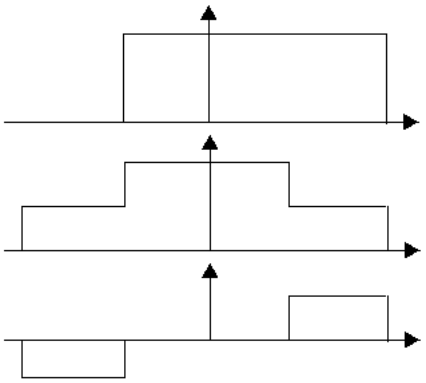

Figure 1 - Shifted boxcar function (top) split into even (center) and odd (bottom) parts

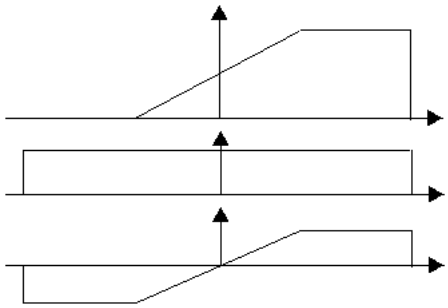

Figure 2 - Tapered shifted boxcar function (top) split into even (center) and odd (bottom) parts

Moreover, if that boxcar is of the same order as the sampling interval (which could happen if one committed an error of almost one sample in locating the origin when trying to select naively the largest portion of single sided interferogram) the wings of the corresponding *sinc* pulse would extend beyond the Nyquist frequency and heavy distortions would occur because of aliasing.

This can be avoided if one tapers the shifted boxcar function as shown in figure 2, to prevent double-weighing of the central portion of the interferogram.
The imaginary part of the ILS gives rise to an imaginary component in the Fourier transform, that can be neglected in the ideal case, since the interferogram is symmetric and the spectral information is only contained in the real part.
Sampling errors and misalignments (see below) that put physical information into the imaginary component should be accounted for before taking the real part of the FT.

1.5 Misalignment correction

The presence of misalignments can give rise (see e.g. [2,5,7,8]) to wavenumber-dependent phase shifts, i.e. the measured spectrum (9) becomes (in expanded notation)

$$S(\sigma) = \int_{-\infty}^{\infty} \left(I(x) - I_{\infty}\right) A(x) \exp\left[j2\pi\sigma x + j\varphi(\sigma)\right] dx = \exp\left[j\varphi(\sigma)\right] S_0(\sigma) * ILS(\sigma) \qquad (10)$$

where $S_0(\sigma)=S_1(\sigma)-S_2(\sigma)$ is the *ideal* spectrum one would get from a perfectly aligned instrument, which is a real function by definition (the interferograms are symmetric, once errors have been accounted for).
As obvious from the comparison of (8) and (10), a misalignment causing only first-order dependence of φ on σ is equivalent to a sampling error.
Taking the modulus of (10) one can see that the modulus of the spectrum is not affected by misalignments, as long as a double-sided interferogram is available, so that (5) can be used.
Provided that φ(s) is sufficiently smooth to allow a first order approximation over the extent of the ILS, one can recover it from the complex low-resolution spectrum (see [2,5,7]), and correct the one-sided interferogram using a proper combination of cosine and sine transforms. One can write (10) as

$$\begin{aligned} S(\sigma) &= \exp\left[j\varphi(\sigma)\right] \int_{-\infty}^{\infty} \exp\left[-j \left.\frac{d\varphi}{d\sigma}\right|_{\sigma} (\sigma - \sigma')\right] S_0(\sigma') ILS(\sigma - \sigma') d\sigma' = \\ &= \exp\left[j\varphi(\sigma)\right] S_0(\sigma) * FT\left[A\left(x + \frac{1}{2\pi} \left.\frac{d\varphi}{d\sigma}\right|_{\sigma}\right)\right]. \end{aligned} \qquad (11)$$

Since $S_0(\sigma)$ is real, the phase of $S(\sigma)$ is entirely due to misalignment, provided that the ILS is real, i.e. A is real and even.
The variable shift of the scanning function is a complication, that would require a separate FT for each frequency. In actual facts, for interferograms that are not too severely chirped (i.e. with frequency dependent ZPD -or more precisely "stationary phase position") one can include the variation in derivative into $\varphi(\sigma)$ which means writing

$$S(\sigma) \cong \exp\left[j\varphi'(\sigma)\right] \int_{-\infty}^{\infty} \exp\left[-j \left.\frac{d\varphi}{d\sigma}\right|_{\sigma_0} (\sigma - \sigma')\right] S_0(\sigma') ILS(\sigma - \sigma') d\sigma' \qquad (12)$$

where the derivative of the error phase is computed at a single position σ_0, and its point-to-point variation is incorporated in the definition of $\varphi'(\sigma)$:

$$\varphi'(\sigma) = \varphi(\sigma) + \left[\left.\frac{d\varphi}{d\sigma}\right|_{\sigma} - \left.\frac{d\varphi}{d\sigma}\right|_{\sigma_0}\right] \varepsilon \qquad (13)$$

where ε is an effective width of the ILS, incorporating the average dependence on σ'.
A further subtlety involved in the process is that the variation in the center of the scanning function has obvious effects at its boundaries, if the interferogram has non-negligible value there. The use of apodizing functions is highly beneficial in this case.
The parity requirement of the scanning function can thus be fulfilled with the short bidirectional interferogram, and one gets the correction formula required from (10):

$$S_0(\sigma) * ILS(\sigma) = \mathrm{Re}\left[\exp\left(-j\varphi(\sigma)\right) S(\sigma)\right] = \cos\varphi(\sigma) FC\left[I(x) - I_{\infty}\right] + \sin\varphi(\sigma) FS\left[I(x) - I_{\infty}\right]. \qquad (14)$$

This is as close as one can get to the real spectrum with a finite scan.

1.6 Empirical misalignment estimator

In order to assess the degree of misalignment in the input data, an empirical estimator can be computed as

$$Max\left\{\tan\varphi(\sigma)\frac{\left|\mathrm{Re}\left(S(\sigma)\right)\right|}{Max\left|\mathrm{Re}\left(S(\sigma)\right)\right|}\right\}=\frac{Max\left[\left|\mathrm{Im}\left(S(\sigma)\right)\right|\right]}{Max\left[\left|\mathrm{Re}\left(S(\sigma)\right)\right|\right]} \tag{15}$$

which is the maximum of $\tan\varphi(\sigma)$, weighed with the fractional amplitude of the spectral density. With JET's *kk5* local calibration data, all reasonably well-aligned channels had values around 0.1 or lower, the lower the better, whereas a channel with an appreciable (although not disastrous) misalignment had a value exceeding 0.2. This was in agreement with the qualitative estimate one could do by comparing the relative sensitivity of the channels, but is independent on input radiation temperature.

1.7 Apodization

Even correcting for all sources of error, the finite extent of the x scan affects the measured spectrum, transforming it into the convolution between the physical spectrum and the ILS, which is a *sinc* pulse if no weights are introduced on the data. This is sometimes undesirable because of the relatively large amplitude of the first minimum of this function, which gives rise to large negative sidelobes in presence of line spectra (which is not the case of Electron Cyclotron Emission spectra in tokamaks, but can make water absorption lines induce distortions in the spectral regions nearby). Tapering the data down to zero at the edges of the scan by multiplication with a suitably chosen apodizing function *(spectral window)* may reduce this distortion sufficiently, at the expense of a spectral smoothing that broadens lines. In ECE spectra, smoothing is normally the desired feature.
Three types of spectral windows are implemented in the program (see [9] for further information and details on respective merits). The formulae below assume $0\le x\le L$.

0. (Rectangular [Dirichlet] window): 1 inside the sampled range, 0 outside. This is just the boxcar function one applies implicitly when doing nothing to the data
1. Hanning $(\cos^2)$ window:
   $w(x)=\cos^2(\pi x/2L)$ for unidirectional interferograms of length L
   $w(x)=\sin^2(\pi x/L)$ for symmetric bidirectional interferograms of length L
2. Triangular window:
   $w(x)=(L-x)/L$ for unidirectional interferograms of length L
   $w(x)=1-|1-2x/L|$ for symmetric bidirectional interferograms of length L
3. Gaussian window (α=2.5):
   $w(x)=\exp\left[-(\alpha\, x/L)^2\right]$ for unidirectional interferograms of length L
   $w(x)=\exp\left[-\alpha^2(1-x/L)^2/2\right]$ for symmetric bidirectional interferograms of length L

1.8 Interpolation

If one computes the spectra at the maximum resolution available, the number of spectral points depends on the location of the ZPD feature in the record, which is only approximately predictable. This may be inconvenient when different data sets must be compared. To avoid that, the one-sided interferogram may optionally be padded with additional data just before computing the final spectrum. If the interferogram is known to end with zero (e.g. a triangular or Hanning window is used), one can simply zero-pad it to reach the number of points required. If this is not the case, zero-padding forces a sharp edge at the original record boundary, which may give rise to widespread spectral ripple. To avoid that, the user may choose to pad the data with a Gaussian tail that matches the value of the last datum. The width parameter of the Gaussian must give plausible data, but is not otherwise constrained by physical requirements. Therefore, it was chosen as one third of the number of data to be added:

$$d_i = d_N \exp\left[-\frac{1}{2}(i-N)^2\Big/\left(\frac{m-N-1}{3}\right)^2\right] \tag{16}$$

where N is the number of data in the interferogram and m the number of data actually requested.

1.9 Error propagation

The FT is a linear operator, therefore error propagation is reasonably straightforward.
With reference to (1b), the standard deviation on coefficient k of the DFT can be written as

$$\left\{\sigma^2_{\left\{\begin{matrix}A_k\\B_k\end{matrix}\right\}}\right\}_{k=0,\frac{(N-\varepsilon_N)}{2}} = \left\{\sum_{i=1}^{N}\left(\frac{\partial f_k}{\partial r_i}\right)^2\sigma_i^2\right\}_k + \left\{2\sum_{i=1}^{N}\sum_{l<i}\left(\frac{\partial f_k}{\partial r_i}\right)\left(\frac{\partial f_k}{\partial r_l}\right)\sigma_{il}\right\}_k \cong$$

$$= \frac{2}{N^2}\left\{\sum_{i=1}^{N} w_i^2\sigma_i^2 \pm \sum_{i=1}^{N} w_i^2\sigma_i^2\cos\left(4\pi k(i-1)/N\right)\right\}_{k=0,\frac{(N-\varepsilon_N)}{2}} \tag{17}$$

where f_k stands for either A_k or B_k, the plus(minus) sign holds for the coefficient of the real (imaginary) part $A_k(B_k)$, w_i are the coefficients of the apodizing functions and the σ^2 are data variances (and not wavenumbers). This formula is only approximate, since the covariance terms in σ_{il} have been neglected, on the assumption that they should be small. Orthogonality between cosine functions cannot be invoked to discard them, because of the finite extent of the path scan (i.e. the finite summation range). The inclusion of the terms neglected would improve the accuracy of the error bars, but is probably not worth the effort.
In a similar way, one can compute the error bar on the cosine transform DFC in (1e):

$$\left\{\sigma^2_{\{A_k\}}\right\}_{i=1,N} = \left\{\sigma_1^2 + \sigma_N^2 + 2\sum_{k=2}^{N-1}\sigma_k^2\left[1+\cos\left(\frac{2\pi(k-1)(i-1)}{N-1}\right)\right]\right\}_{i=1,N} \tag{18}$$

2. Data processing

Data processing is essentially the one described in [2].

Putting everything together, one should start by removing any linear trends in the data, that would cause spectral distortions by convolving the spectrum with the FT of a sawtooth function. This may not always be necessary, and sometimes can indeed be detrimental if data are known by other means to have exactly a horizontal baseline, so it is selectable as an option.

Then the approximate location of the ZPD is computed by fitting a 3-point parabolic spline around the maximum, and the interferogram is shifted *(rotated)* to place at the start of the record the first sample following the approximate ZPD location; of course the previous samples are placed in reverse order at the end of the record. The rotation is done to reduce the slope of phase-vs-wavenubmer (see 1.3), thus making the tricky unwrap procedure somewhat easier.
It is assumed that the approximate ZPD location is accurate enough that the number of data in the one-directional interferogram can be predicted with it, so that the data can be zero-padded in order to interpolate the low resolution spectrum to the same number of points as the final high-resolution one. This number is the one requested by the user or the one resulting from maximum resolution, whichever the greater (within allocated boundaries, of course).

The phase of the *complex low resolution spectrum* is then unwrapped and fitted with a straight line to obtain the correction ζ to the approximate ZPD location and the sign of the zero-frequency component (in practice the polarity of the signal, that should be preserved).
If the exact location of the ZPD is known, it can be given as an input and used instead of the one computed here.

The linear phase dependence on wavenumber is removed and the residual unwrapped and piecewise fitted with parabolic splines to be used for correction of misalignment (see 1.5).
To avoid noise amplification, the fit is only made in regions where the correction is significant: the spectral power must exceed 3% the average of the widest region with signal above average, and the phase correction must exceed 3 times the standard deviation of phase in the same region. Since the region with largest spectral power is the one driving the linear fit, one should expect the phase correction to leave mostly noise here, so that threshold seems adequate to make sure that a real misalignment is found and not random noise.

Having processed all the information available from the short bidirectional interferogram, the full record is analyzed.

First of all, the linear tapering described in 1.4 and the (optional) apodizing functions are applied to the data, taking care that the origin of functions is computed from the (now known) *exact* location of the ZPD.

The data are then rotated to place in the first position the first datum after the ZPD, and padded if necessary with zeros or with a Gaussian function as described in (1.8) in order to get the required number of spectral points.

The complex FT is then computed.
The phase is corrected by removing the linear terms due to the ZPD position (sampling error described in 1.3) and (optionally) the piecewise parabolic terms due to misalignment (described in 1.5) where available.
Having removed all the (known) sources of error giving rise to complex factors in front of the spectrum $S(\sigma)$, its residual imaginary part is only due to the imaginary part of the ILS (see 1.4) and can be safely discarded, the real part being the best estimate to the high resolution spectrum of the data.

For completeness, the spectrum is inverse-cosine-transformed to give the re-aligned unidirectional interferogram.

3. Program description

The "program" is actually a FORTRAN subroutine, and a demo program –described in the appendix- is provided in the package.

3.1 Calling arguments

The subroutine is called with a statement like

```
CALL mpisub2(nraw, ignoredata, tow, nout, corr, output, rawdata, rawstd,
             ndat, err, intfout, dintf, spout, dsp, zpdcoarse, misal, str)
```

where the input arguments are

NAME	TYPE	Description	Acceptable range
nraw	INTEGER	Number of input data	15<nraw<N (*)
ignoredata	INTEGER	Number of invalid data to ignore at record's ends	0< ignoredata<nraw/2
tow	INTEGER	Type of window: 0:none (rectangular), 1: Hanning, 2: triangular, 3: Gaussian	0≤tow≤3
nout	INTEGER	Number of spectral point requested. 0=Automatic (as many as allowed by resolution).	0 or ndat≤nout≤N (*). Condition nout<ndat is treated as nout=0 but rises a (soft) error flag.
corr	INTEGER	Type of data correction. Sum of independent values: 0-no correction, 1-baseline subtraction, 2-misalignment correction, 4-Gaussian data padding if interpolation is requested, 8-propagate error bars, 16-take ZPD positions from external file.	0≤corr≤1+2+4+8+16=31
output	INTEGER	Type of direct data output. Sum of independent values: 0-no output, 1-console output of error condition 4, 2-s-related information, 4- x-related information, 8- textual information, 16- s-related debug information	Any. Tests are made on ABS(output)
rawdata	DOUBLE PRECISION array	Array of input data	DIMENSION(rawdata)≤N (*)
rawstd	DOUBLE PRECISION array	Array of standard deviations of input data	DIMENSION(rawdata)≤N (*)
str	CHARACTER*10	String to prepend to file names when output>0	Any allowed by filesystem as part of a file name

*:see below for value of N

the output ones are

NAME	TYPE	Description
ndat	INTEGER	Number of independent data in the unidirectional interferogram (without data padding).
err	INTEGER	Sum of error conditions found. 1-64 reserved for runtime errors, 128-2048 for errors in input parameters, ≥4096 for fatal errors. 1-*ndat≥nout. Fewer data were requested than available in the spectrum (soft error - user input ignored);* 2-*computed ZPD location outside record (garbage data or bug);* 4- *too few data in symmetric interferogram;* 8-*loss of precision in phase fit (constant data?);* 16- *loss of precision in baseline fit (constant data?); 32-inconsistent inputs to take2 (bug); 64-too many data for internal array size (same as 4096, but discovered at runtime);* 512-*nout is negative or too large;* 1024-*unknown type of spectral window;* 2048-*ignoredata is negative or too large;* 4096-*too many or too few input data.* Unspecified values are reserved for further expansion
intfout	DOUBLE PRECISION array	Processed unidirectional interferogram (not normalized). DIMENSION(intfout)=MAX(nout,ndat)
dintf	DOUBLE PRECISION array	Standard deviation of processed interferogram. DIMENSION(intfout)=MAX(nout,ndat)
spout	DOUBLE PRECISION array	Output spectrum. DIMENSION(spout)=MAX(nout,ndat)-1
dsp	DOUBLE PRECISION array	Standard deviation of output spectrum. DIMENSION(spout)=MAX(nout,ndat)-1
misal	DOUBLE PRECISION	Parameter estimating misalignment as in (12) above. 0≤misal

*:see below for value of N

One parameter is either input or output, depending on the value of corr

zpdcoarse	DOUBLE PRECISION	Position of ZPD feature in record. 1≤zpdcoarse≤nraw

In addition to the arguments, there are also a few internal tuning parameters. Changing them requires recompilation.

NAME	TYPE	DESCRIPTION	Current value
N	INTEGER PARAMETER	Maximum size of input record. Used for dimensioning internal arrays	256*2+1, set in PARAMETER statement in mpisub
mincoadat	INTEGER	Minimum number of data in short two sided interferogram	15, set in DATA statement in mpisub
lows	INTEGER	Number of low-frequency points to neglect in unwrapping phase	6, det in DATA statement in mpisub
maxdel	DOUBLE PRECISION	Maximum acceptable point to point variation, as a fraction of 2π, after approximate unwrap	0.6d0, set in DATA statement in subroutine unwrap7 and unwraps
maxder	DOUBLE PRECISION	Maximum total variation of phase allowed in record (fraction of 2π)	2, set in DATA statement in subroutine unwrap7
frac	DOUBLE PRECISION	Fraction of average power (in wider area of continuous phase) below which no correction for alignment error takes place	0.03, set in DATA statement in subroutine realign

`mult`	DOUBLE PRECISION	correction for alignment takes place only if `frac` test successful and phase correction exceeds mult times the standard deviation of phase in the highest signal region.	set in DATA statement in subroutine `realign`

3.2 Library functions

This program uses FFTPACK [6]. Should it not be available as a library, these routines (and only these) should be included explicitly: `dcost.f, dcosti.f, dfftb.f, dfftf.f, dffti.f, dsint.f, dsinti.f, dzfftb.f, dzfftf.f, dzffti.f, ezfft1.f, radb2.f, radb3.f, radb4.f, radb5.f, radbg.f, radf2.f, radf3.f, radf4.f, radf5.f, radfg.f, rfftb1.f, rfftf1.f, rffti1.f, sint1.f`.

3.3 Output files (optional)

Several files with intermediate results that are of interest sometimes, can be written on request. The files are named
`t<str>YYMMDDhhmmsshh.txt`: textual information
`s<str>YYMMDDhhmmsshh.txt`: data in *s*- (Fourier-)space
`x<str>YYMMDDhhmmsshh.txt`: data in *x*- (path difference) space
`d<str>YYMMDDhhmmsshh.txt`: phase debug information in *s*- space
`e<str>YYMMDDhhmmsshh.txt`: error bar debug information in *s*- space

where `<str>` is the string contained in the input parameter `str`, YYMMDD is the date in the usual form, hhmmsshh the time with a resolution of a hundredth of a second. This is necessary to avoid overwriting data in subsequent calls to `mpisub` if `str` is not changed by the calling program.

Most information contained in these files is not relevant in normal operation, but the last column in *d*_file (or last column in *s*_file) are actually very useful for monitoring the accuracy of interferogram realignment (*d*_file only) and ZPD location when dealing with interferograms with low signal-to-noise ratio.

Any user should be cautious of the number of files generated when `output`>1. *Never* try to process several *kk5* JETDSP files with `output`=31, since 4 files per interferogram (plus presumably at least two by the main program), four interferograms per round, more than a thousand rpm are likely to give problems. You have been warned.

The output files are written in tab-separated format, ready for post-processing.
The columns are:
s_file:

- Independent variable s ($0<s\leq 1$) of high resolution spectrum
- High resolution spectrum from cosine transform of unidirectional interferogram (not normalized)
- *Standard deviation on previous column*
- Residual imaginary part of spectrum
- *Standard deviation on previous column*
- Spectral power ($\sqrt{A^2 + B^2}$) where A,B are the real (cosine transform) and imaginary parts
- Real part (A) of low resolution (coarse) spectrum
- Imaginary part (B) of low resolution (coarse) spectrum
- Un-normalized low resolution spectrum ($\sqrt{A^2 + B^2}$)
- Phase of low resolution (coarse) spectrum, including ZPD displacement [rad]
- Residual phase of low resolution spectrum after ZPD shift [rad]

Columns in italic are only included if error bar processing was requested.

x_file:

- Index (1..nraw)
- Raw data (interferogram) after optional baseline subtraction
- Previous column multiplied by apodizing function
- Short bidirectional interferogram multiplied by apodizing function

- Raw data
- *Standard deviation on raw data (input)*
- Input data multiplied by linear ramp in double-sided region
- Processed unidirectional interferogram
- *Standard deviation on previous column*

Columns in italic are only included if error bar processing was requested.

t_file:

- `Nraw:` number of input data
- `Ncoadat:` number of data in symmetric interferogram
- `Npts:` number of output data in spectrum(+1) and interferogram
- `Ndat:` number of independent data (setting the spectral resolution)
- `Start0:` start of short bidirectional interferogram in record
- `Stp0:` end of short bidirectional interferogram in record
- `Start:` position of first datum to consider in raw data
- `Stp:` position of last datum to consider in raw data
- `Abase:` slope of the baseline of raw data
- `Bbase:` intercept of the baseline of raw data
- `Afitphcoa:` slope of the linear best fit of phase vs. wavenumber
- `Bfitphcoa:` intercept of the linear best fit of phase vs. wavenumber
- `Deltai:` position of ZPD in record, neglecting invalid head and tail
- `Deltai0:` approximate position of ZPD in record, from parabolic fit around maximum
- `Zmax:` location of first point after ZPD in record
- `Zpdcoarse:` position of ZPD in record
- `Azero:` zero frequency spectral component of spectrum

d_file: (debug information on phase)

- Independent variable s ($0<s\leq1$) of high resolution spectrum
- Phase at s, interpolated from results of low resolution spectrum, in units of π
- Unwrapped phase at s, interpolated from results of low resolution spectrum, in units of π
- Residual phase in low resolution spectrum, after correction for ZPD position, in units of π
- Weight (0,1) used in unwrapping procedure
- Spectral power in low resolution spectrum
- Residual phase in low resolution spectrum, after correction for ZPD position and misalignment, in units of π

e_file: (debug information on error bars)

- Index, 1 to NPTS-1
- Input standard deviation on data
- Standard deviation on processed unidirectional interferogram
- Standard deviation on high resolution spectrum (same as column 3 of s-file)
- Taper function to multiply data
- Same, dropping `ignoredata` points either end
- Same, multiplied by apodizing function
- Standard deviation on real (A) part of low resolution spectrum
- Standard deviation on imaginary (B) part of low resolution spectrum
- Standard deviation on imaginary part (sine) transform of processed unidirectional interferogram

3.4 I/O logical units

The program reserves logical unit 37 for its output files. The calling program should not have files open with this logical unit when calling `mpisub`, unless `output` is zero.

3.5 Phase unwrap

While the short description in section *Basic theory* suffices for most of the program, unwrapping and fitting the phase of the low-resolution spectrum involves a rather elaborate procedure in `unwrap7,` (and in a simplified form in `unwraps)` that must be documented here.

First of all, phase data are normalized to 2π. Then an approximate unwrap is achieved by adding one (i.e. 2π) to any phase point that differs more than 2π∗`maxdel` from its predecessor. The result is a set of segments, separated by discontinuities.
The algorithm maps the segments using the criterion that, for all data in one segment, the derivative (i.e. the difference with next datum) should not differ more than 1-`maxdel` from the average slope of the full record.
Then the *dominant* region is found, using the weights defined in `mpisub`. They are defined as 1 when the power spectral density in the low-resolution spectrum is above average, and zero otherwise. This forces the algorithm to identify as d*ominant* region the one where most of the power resides.
The average slope in the dominant region is found, and the other phase points are shifted by addition of an integer number (of 2π) to lie as close as possible to the line defined by it.

4. Performance Tests

Analytic data were used first (a *sinc* function) for testing the precision in measuring the ZPD position and the behaviour of the spectral windows. The difference between the ZPD position set in the analytic data and the one computed by the program was always below 0.002 times the sampling interval. This is by no means a general measure of precision (it depends on the number of samples in the symmetric interferogram and the shape of the spectrum) but one should always expect a precision of the order of a fraction of the sampling interval.
Afterwards, all the program functions were tested using JET's *kk5* measurements on a blackbody radiator on its calibration line. Different ZPD positions, input temperatures and alignment conditions were examined.
The ZPD position computed by the program is always consistent with the tiny differences between sectors of the wheel scanning the movable arm [1], the direction of rotation and the physical layout of the instrument. This gives enough confidence to use this analysis to extract information on fine alignment of the optics.

Conclusion

The program in its present status appears reasonably robust in providing meaningful results when fed with meaningful data, tolerating some amount of misalignment and gently handling (i.e. without crashing) garbage data. The results of the high-resolution spectrum are compatible with the low-resolution one everywhere.
The source code will be placed on IFP's website next to this report as soon as possible. If you actually *use* it, I would be grateful for a citation of this report in your paper. If you find bugs, I would appreciate being informed.

Appendix

The Appendix includes information that is related to the program or its use, but not essential for describing it or documenting its use.

Demo program

`mpisub` is a subroutine, so a demo program was written to use it for one flavour of JET's *kk5* local calibration data and to show examples of use. Its detailed description is not useful, since anyone interested had better just look at the source code and comments therein. In its most used incarnation, the program reads data written in Labview's *.lvm* format, containing the averaged interferograms (no error bars). More than one 1024-samples turn can be included, thus allowing some estimate of error bar.

The program is normally invoked through a unix script named `shuffleplot` (or `shuffle` if one does not want plots to be made automatically) that calls all the programs required for post-processing, and shuffles -hence the name- all the files produced by the program into a standard directory structure.

In order to use the program, one should provide a text file (unix LF termination, if that is an issue) named `parameters.txt` like the one shown here

```
15      ;firstdatum (#data to drop at begin-end of interferogram)
1       ;type of window (0-none, 1-cos^2, 2-triangular, 3-Gaussian
256     ;nout (number of data wanted)(interpolate if necessary; 0=auto)
31      ;0: no internal output, +1: console output of err 4, +2: s-file, +4: x_file, +8: t_file, +16:  d-file
1       ;data processing: 0:none+1:slope subtraction+2:realign
.......(list of input files to process, one per line; data in .lvm format with proper LF termination; filenames limited to 31 characters in this version)
```

The program writes results for each input data file on two output files named *z_<name of input file>* and *sum_<name of input file>*.

The files are written in tab-separated format, and contain the following information for each interferogram:

*z_*file:

- channel number (1 to 6)
- start location of interferogram in input file (integer)
- error returned by `mpisub`
- number of data in processed interferogram
- ZPD position in mm
- misalignment index

*sum_* file:

- `nraw`: number of raw data per channel
- `nchan`: number of channels
- `firstdatum`: input parameter (see subroutine arguments above)
- `tow`: input parameter (see subroutine arguments above)
- `nout`: input parameter (see subroutine arguments above)
- `output`: input parameter (see subroutine arguments above)
- `corr`: input parameter (see subroutine arguments above)

For each interferogram, the program writes files named fc*S*siiiii*x_<name of input file>,* fc*S*siiiii*s_<name of input file>,* where *f* is the order of the input file in the list above (not necessarily single-digit), *c* is the single-digit channel number (i.e. data column in the *.lvm* format), *S* is the letter 'S', *s* the sector number (1 to 4, counting from the beginning of the file, so like-numbered sectors in different channels are not the same –but a one to one relation exists of course-), *iiiii* the five-digit location or the first point of the interferogram in the file. Five digits are used to accommodate data files of 10240 samples as those used for the absolute calibration. *x_, s_* are the strings 'x_', 's_'.

fc*S*siiiii*x_(s_)* files are written in tab-separated format, with two columns, one holding the independent variable (path difference or frequency), the other the processed interferogram and the high resolution spectrum respectively. The fc*S*siiiii*s_* file also holds the spectral resolution in GHz, the number of independent data and the misalignment index on the first data line.

In order to work, the demo program needs knowledge about the position of the first interferogram in the record. This piece of information is preset in a DATA statement, and was obtained using the program allin (see below)

Program structure and Subroutines

The main program demo calls mpisub and getlvm. (See below for this).
mpisub calls checkparms, fitsmooth, linfit, phacorr, take, unwrap6, window, wlinfit and the FFTPACK routines DCOST, DCOSTI, DSINT, DSINTI, DZFFTB, DZFFTF, DZFFTI.
The purpose of the subprograms is:

- checkparms: range check of call arguments
- erbars: error bars on cosine transform
- erbars2: error bars on forward FT
- linfit: linear least squares fit
- realign: fit a line through the phase data
- take2: take a portion of the data, accounting for padding and data rotation
- unwrap7: unwrap phase data
- unwraps: unwrap phase data, simplified version
- window3: apply a spectral window to the data
- wlinfit: linear least squares fit with external weights

CalibAn:

This program is a specialized version of demo optimized for processing the most recent version of JET's *kk5* in-vessel calibration data (hence the name *Calibration Analysis)*. There are differences in the input file format (simple text files, 1024 lines, no header row, 2*number-of-channels columns, respectively channel 1 datum, standard deviation, ..., channel n datum, standard deviation) and input parameters (file caliparms.txt):

```
218 91 90 217 90 91 ;approximate position of sector jumps
12   ;firstdatum (#data to drop at begin-end of interferogram)
1    ;type of window (0-none, 1-cos^2, 2-triangular, 3-Gaussian
256  ;nout (number of data wanted)(interpolate if necessary; 0=auto)
31 ;0:no output,+1:output of err 4, +2:s_file, +4:x_file, +8:t_file, +16:d-file
11  ;processing: 0:none,+1:slope subtraction,+2:realign,+4:Gaussian padding,+8:read
     and propagate error bars,+16:force ZPD position(if so, first filename is for
     ZPDs)
4    ;number of channels (WITH error bars)
2 6 3 5    ;identification numbers of all channels
zpdindx.txt ;zpd position input [only used if Mod(corr,32)>=16)] DO NOT CANCEL LINE
Ch2635Hot-Vessel.txt
```

The main difference with parameters.txt is the addition of the approximate position of sector jumps (necessary to handle situations where the relation between encoder positions and actual path scan was not homogeneous in the data sets because of hardware faults). The number of channels and their identification are contained in the parameters file, since the file format does not carry this information. The first filename is used for input of ZPD positions (e.g. from a previous run or to force a different correction), that is enabled with corr+16. It is not implemented in demo.
CalibAn is normally run with the unix script caliplot, that performs the very same functions as shuffleplot.

Ancillary programs

Several tiny programs were written to post-process the program's outputs or provide essential data for it. While none of them is worth documenting on its own, a short description is given here in case of need.

- getlvm.f: subroutine to read LabView™'s .lvm files
- allin.f: finds position of interferograms in record. Also useful for getting information on alignment.
- erbars.f: post-process a set of tab-separated files, considering the first column as the independent variable, the last column (P, compilation parameter) as holding the data. Writes a two-column tab-separated file with independent variable of the last input file, average and standard deviation of the data column of all files.

- `zpdstat.f`: post-process summary file with ZPD data, written by `demo.` It writes a tab-separated file with ZPD-position and its standard deviation, misalignment parameter and its standard deviation, averaging separately for each channel and sector over all interferograms available
- `shuffle:` a unix shell script to drive `demo`. Originally written to arrange the many tens of files produced into a proper directory structure (hence the name), it also drives the post-processing with `erbars.f` and `zpdstat.f.` `DEMO` is run and its outputs are moved in a set of directories named 1 to <number of raw data files>, each containing directories named 1 to <number of channels>, each containing four directories (1 to 4) for the four sectors. Each directory contains processed data for all the available interferograms of the file/channel/sector under consideration. `erbars` is run on the processed data (output from `demo`), and `zpdstat` on the summary ZPD data.
- `rette2.f:` post process spectra at different temperatures using a linear fit for each frequency

References

[1] the closest reference is at present S. Garavaglia et al., *Mechanical realization of a multichannel Martin-Puplett interferometer for perpendicular and oblique ECE measurements on JET,* Fusion Engineering and Design, 82, 1224-1230, (2007)

[2] C.D. Porter, D.B. Tanner *Correction of phase errors in Fourier spectroscopy* International Journal of infrared and Millimeter Waves, vol. 4, no. 2 (1983) 273-298

[3] P.N. Swarztrauber, FFTPACK, http://www.netlib.org/fftpack/index.html

[4] R.N. Bracewell *The Fourier Transform and its applications* McGraw-Hill

[5] R.J. Bell *Introductory Fourier Transform spectroscopy* Academic Press, 1972, New York, London

[6] D. H. Martin *Polarizing (Martin-Puplett) interferometric spectrometers for the near- and submillimeter spectra* in Infrared and Millimeter waves, Vol. 6 Systems and Components, Chapter 2, pp. 65-148, K.J. Button, Editor, Academic Press 1982

[7] H. Sakai, G.A. Vanasse, M.L. Forman *Spectral recovery in Fourier spectroscopy* Journal of the Optical Society of America Vol, 58,1 (January 1968), pp. 84-90

[8] L. Mertz *Auxiliary computation for Fourier spectrometry* Infrared Physics, vol 7 (1967), pp. 17-23

[9] F. J. Harris, *On the use of windows for harmonic analysis with the discrete Fourier transform,* Proceedings of the IEEE, Vol. 66, 1 (January 1978), pp. 51-83