Title:	Mathematical Formulation used by MATLAB Code to Convert FTIR Interferograms to Calibrated Spectra
Author(s):	Armstrong, Derek Elswick
Intended for:	Report
Issued:	2016-07-19

Mathematical Formulation used by MATLAB Code to Convert FTIR Interferograms to Calibrated Spectra

Derek E. Armstrong, XCP-8

Los Alamos National Laboratory, Los Alamos, NM

Abstract

This report discusses the mathematical procedures used to convert raw interferograms from Fourier transform infrared (FTIR) sensors to calibrated spectra. The work discussed in this report was completed as part of the Helios project at Los Alamos National Laboratory. MATLAB code was developed to convert the raw interferograms to calibrated spectra. The report summarizes the developed MATLAB scripts and functions, along with a description of the mathematical methods used by the code. The first step in working with raw interferograms is to convert them to uncalibrated spectra by applying an apodization function to the raw data and then by performing a Fourier transform. The developed MATLAB code also addresses phase error correction by applying the Mertz method. This report provides documentation for the MATLAB scripts.

1 Introduction

This report discusses mathematical procedures and MATLAB code to convert raw Fourier transform infrared (FTIR) spectroscopy data to calibrated spectra. The MATLAB code and work described in this report was completed for the Helios project at Los Alamos National Laboratory (LANL). The algorithms used to transform the raw FTIR data to calibrated spectra will be discussed. Also, information on running the MATLAB scripts will be given.

The developed MATLAB code has been initially applied to the analysis of recent Helios data. The MATLAB code consists of scripts and functions to transform the raw interferograms to *uncalibrated* spectra. The MATLAB code can then be used to calibrate the uncalibrated spectra by applying an estimated sensor gain and offset. The wavenumber dependent gain and offset are also computed with the MATLAB code. The gain and offset are estimated by using blackbody calibration source data at different temperatures.

The next section of this report discusses the procedures for converting raw FTIR data to uncalibrated spectra (see reference [1] for more information). The main technique to convert the raw data to uncalibrated spectra is the Fourier transform. This section also discusses apodization and phase error correction. Apodization is a method that multiplies the raw FTIR data (interferogram) by a suitable function to improve the estimation of the input source radiance to the sensor. Apodization is necessary due to the finite collection limitations of the FTIR sensor. Another necessary step is phase error correction, which adjusts the computed calibrated spectra to account for small errors from instrumental sources. The Mertz method (see reference [1] and [2]) was applied to the phase error correction problem. The third section discusses the calibration (see reference [3]) of the uncalibrated spectra. Calibration is performed by estimating a wavenumber dependent gain and offset from blackbody sources and then applying the gain and offset to uncalibrated data. The final section (section 4) provides information about the developed MATLAB functions and scripts.

2 Converting Interferograms to Spectra

FTIR sensors are based on the Michelson interferometer. A Michelson interferometer takes an input beam of radiation, divides the beam into two paths, and then recombines the two beams after a path difference has been introduced. The input beam is divided into two beams with the use of a beam splitter. The Michelson interferometer also consists of two mirrors, where one of the mirrors is movable so that a path difference is created after the beams reflect off of the mirrors and recombine at the beam splitter. A basic illustration of the Michelson interferometer is given in figure 1.

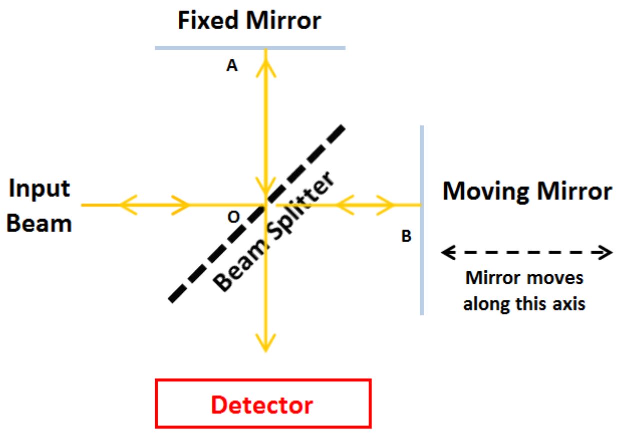


Figure 1: Basic diagram of a Michelson interferometer.

The differences in path length created by the moving mirror creates an interference pattern that is measured by a detector. This interference pattern is referred to as an interferogram and this is the raw data from a FTIR sensor. Let OA be the distance between the beam splitter and the fixed mirror. Similarly, let OB be the distance between the beam splitter and the moving mirror. The optical path difference for the two beams recombining at the beam splitter is $2(OB - OA)$ and is denoted by δ. The value of δ is also called the retardation or path retardation.

In the following subsections, the mathematical formulation for converting raw interferograms to spectra is given. The formulation given follows the reference [1] closely and is the basis of the developed MATLAB code.

2.1 Basic Formulation

Letting $B(\nu)$ be the input source intensity at wavenumber ν, it can be shown that the intensity at the detector for the simple case of monochromatic light is

$$I_{det,v}(\delta) = 0.5B(\nu)(1 + cos(2\pi\nu\delta)). \tag{1}$$

Integrating equation 1 across all wavenumbers, the intensity at the detector is given by

$$I_{det}(\delta) = \int_0^{+\infty} 0.5B(\nu)(1 + cos(2\pi\nu\delta))d\nu. \tag{2}$$

Discarding the term independent of δ in equation 2 and allowing $B(\nu) = B(-\nu)$ for negative values of v, the interferogram will be defined as

$$I(\delta) = \int_{-\infty}^{+\infty} B(\nu)cos(2\pi\nu\delta)d\nu. \tag{3}$$

Note that the functions I_{det} and I are related according to

$$I_{det}(\delta) = I(\delta)/4 + B_{total}/2 \tag{4}$$

where

$$B_{total} = \int_0^{+\infty} B(\nu)d\nu. \tag{5}$$

The interferogram $I(\delta)$ is the Fourier cosine transform of the input source $B(\nu)$. Since $I(\delta)$ and

$B(\nu)$ are even functions, the input source $B(\nu)$ is given by

$$B(\nu) = \int_{-\infty}^{+\infty} I(\delta) cos(2\pi\nu\delta) d\delta. \tag{6}$$

In other words, the input source $B(\nu)$ can be obtained by taking the inverse Fourier transform of the interferogram $I(\delta)$. Note that the imaginary (or sine) part of the Fourier transforms are zero due to the functions being even.

2.2 Apodization

Equation 6 allows for infinite retardation δ and would hold, only in theory, if the moving mirror could move infinitely far away. Since the moving mirror will impose a finite maximum path retardation, the integral 6 can be written as

$$\widetilde{B}(\nu) = \int_{-\infty}^{+\infty} I(\delta) D(\delta) cos(2\pi\nu\delta) d\delta \tag{7}$$

where the function $D(\delta)$ is given by

$$D(\delta) = \begin{cases} 1 & : \delta \in [-\Delta, \Delta] \\ 0 & : \delta \notin [-\Delta, \Delta]. \end{cases}$$

for some positive value Δ. For simplicity, the equation above is assuming that the mirror moves the same amount of distance in both directions relative to the point of zero retardation. This assumption creates double-sided interferograms which are the interferograms of interest in this report.

Therefore, the modified input source radiance $\widetilde{B}(\nu)$ is the Fourier cosine transform of the interferogram $I(\delta)$ multiplied by the function $D(\delta)$. In practice, the term $I(\delta)$ is also impacted by the detector response. By the convolution theorem, the modified input source $\widetilde{B}(\nu)$ is the convolution of the Fourier transform of $I(\delta)$ and the Fourier transform of $D(\delta)$. The Fourier transform of $I(\delta)$ is the true input spectrum $B(\nu)$ and the Fourier transform of $D(\delta)$ is the following modified sinc function

$$f(\nu) = 2\Delta sinc(2\pi\nu\Delta). \tag{8}$$

A plot of the sinc function is given in figure 2. By computing the Fourier transform of the obtained interferogram it is now seen that the result is the true spectrum convolved with the sinc type function of equation 8. The sinc function has relative large oscillations that includes negative values. To reduce the impact of these oscillations on the approximation of $B(\nu)$, an apodization procedure is performed on the interferogram data obtained from the FTIR sensor. An apodization function is used to taper or smooth the oscillations of the sinc function.

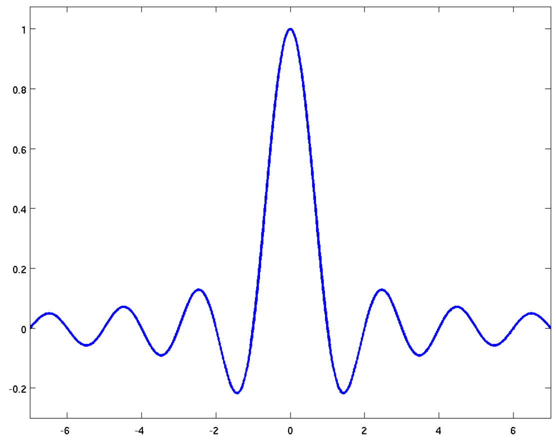


Figure 2: Graph of the $sinc$ function.

Many different apodization functions can be used. A triangular apodization function is used by the MATLAB code discussed in this report. A plot of this function is given in figure 3. This function equals one at zero and linearly decreases to zero at $\delta = \pm\Delta$. Let $T(\delta)$ represent this apodization function. Then equation 7 becomes

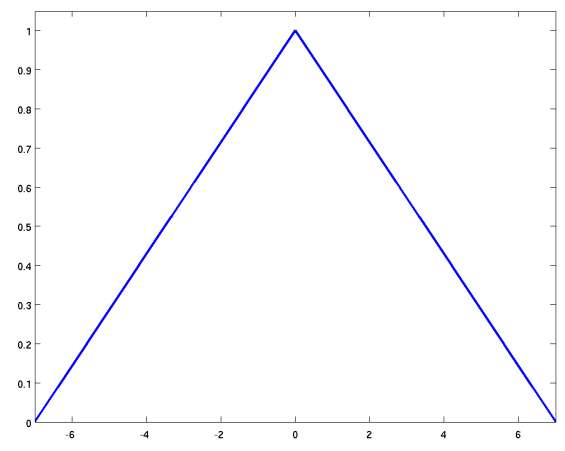


Figure 3: Graph of a triangular apodization function.

$$\widetilde{B}(\nu) = \int_{-\infty}^{+\infty} I(\delta)T(\delta)cos(2\pi\nu\delta)d\delta. \tag{9}$$

Now, applying the convolution theorem to equation 9, the term $\widetilde{B}(\nu)$ is the convolution of the Fourier transform of $T(\delta)$ and the Fourier transform of $I(\delta)$. The Fourier transform of $T(\delta)$ is equal to

$$f(\nu) = \Delta sinc^2(\pi\nu\Delta). \tag{10}$$

The function $sinc^2$ is plotted in figure 4 and has noticeably reduced oscillations. Therefore, by using this apodization function to multiply with the data interferogram, a more accurate representation of the input source radiance is obtained when performing a Fourier transform.

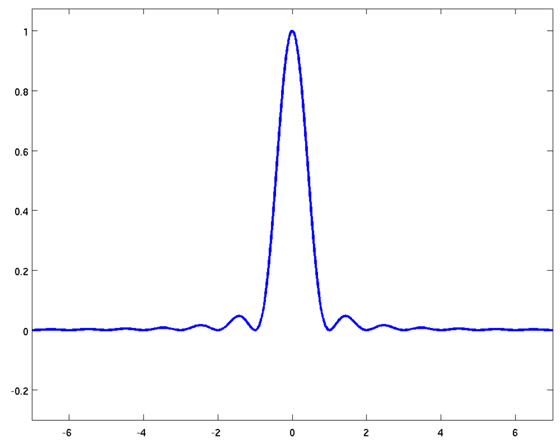


Figure 4: Graph of the $sinc^2$ function.

2.3 Correction for Phase Errors

Another issue with interferograms from FTIR sensors is the phase error introduced by the instrument. To address phase error correction, it is best to work with the complex Fourier transform. The complex Fourier transform of $I(\delta)T(\delta)$ is given by

$$\widetilde{B}_c(\nu) = \int_{-\infty}^{+\infty} I(\delta)T(\delta)e^{-2\pi i\nu\delta}d\delta. \tag{11}$$

Note that in the absence of phase errors, the Fourier sine transform (i.e., the imaginary part of the complex Fourier transform) is zero due to $I(\delta)T(\delta)$ being an even function. The phase error can be represented by a wavenumber dependent term $\theta(\nu)$ so that the interferogram is given by

$$I(\delta) = \int_{-\infty}^{+\infty} B(\nu)cos(2\pi\nu\delta - \theta(\nu))d\nu. \tag{12}$$

The addition of the phase error for equation 12 introduces a sine component to the transform. This can be seen from the following trigonometric equation

$$cos(\alpha - \beta) = cos(\alpha)cos(\beta) + sin(\alpha)sin(\beta). \tag{13}$$

Using the trigonometric identity 13, equation 12 can be written as

$$I(\delta) = \int_{-\infty}^{+\infty} B(\nu)cos(2\pi\nu\delta)cos(\theta(\nu))d\nu + \int_{-\infty}^{+\infty} B(\nu)sin(2\pi\nu\delta)sin(\theta(\nu))d\nu. \tag{14}$$

Equation 14 represents the interferogram $I(\delta)$ as an even function plus an odd function. Therefore, if the complex Fourier transform

$$\widetilde{B}_c(\nu) = \int_{-\infty}^{+\infty} I(\delta)T(\delta)e^{-2\pi i\nu\delta}d\delta \tag{15}$$

is computed, then it will be approximately equal to $B(\nu)e^{-i\theta(\nu)}$. It follows that

$$B(\nu) \approx e^{i\theta(\nu)} \int_{-\infty}^{+\infty} I(\delta)T(\delta)e^{2\pi i\nu\delta}d\delta. \tag{16}$$

In order to compute an estimate of $B(\nu)$, an estimate of $\theta(\nu)$ is needed. An estimate for $\theta(\nu)$ can be obtained from the following. The result from taking the Fourier transform of $I(\delta)T(\delta)$ is approximately $B(\nu)cos(\theta(\nu)) - iB(\nu)sin(\theta(\nu))$ where

$$\theta(\nu) = atan2(Imag(\widetilde{B}_c(\nu))/Real(\widetilde{B}_c(\nu))). \tag{17}$$

It then follows that the phase corrected spectrum $B(\nu)$ is estimated by

$$B(\nu) = cos(\theta(\nu))Real(\widetilde{B}_c(\nu)) + sin(\theta(\nu))Imag(\widetilde{B}_c(\nu)). \tag{18}$$

2.4 Algorithm Steps

In this subsection, step by step instructions are given for converting the raw interferogram data to uncalibrated spectra. The information given in this section corresponds to the steps taken in the MATLAB function Igm_to_Spectra() (see section 4). In the following, let **I** be a vector representing the raw interferogram. The length of **I** is N. The first step in the algorithm is to find the point in the interferogram that is closest to zero path retardation and center the vector around that point. The resulting vector is a double-sided interferogram. The next main step of the algorithm is to estimate the phase error using equation 17.

Step 1: Locate the centerburst in **I**. This is the maximum point and corresponds to the zero path difference. Subtract the mean of **I** from **I**. Subtracting the mean from **I** will not impact the results and puts the interferogram into the form of equation 3 (except for a constant multiplication factor). The constant multiplication factor is also not important since it is ”folded” into the computed gain when doing the calibration (see next section).

Step 2: Extract a small double-sided interferogram with the centerburst being the center of the extracted subvector. Let **S** represent the extracted subvector. For the MATLAB function Igm_to_Spectra(), the extracted interferogram can be of size 128, 256, or 512. For the remainder of this section, a size of 256 will be assumed. This small double-sided interferogram will be used to estimate the phase error. A smaller sized interferogram can be used since the phase error is expected to vary little as a function of wavenumber.

Step 3: Multiply the vector **S** pointwise with the triangular apodization vector **T** that equals one in the middle (at the centerburst location) and linearly decreases to zero on both sides. That is, $\mathbf{T}(1)$ and $\mathbf{T}(256)$ both equal zero when the length of vector **S** is assumed to be 256. Let $\mathbf{S} = \mathbf{S}. * \mathbf{T}$, where the operator .* represents pointwise vector multiplication.

Step 4: Swap the left and right halves of **S**. The swap is done so that the maximum value of **S** is the first element. This is a typical step when performing a discrete Fourier transform in a language such as MATLAB. For example, the fft() function in MATLAB expects the input to be sampled so that the zero component is first (e.g., an index order of 0, 1, 2, ..., $N/2-1$, $-N/2$, ..., -1). Let $Swap()$ denote the operator that swaps a vector as described in this step. Let $\mathbf{S} = Swap(\mathbf{S})$.

Step 5: Compute the discrete (fast) Fourier transform of **S**. Estimate the phase error according to equation 17,

$$\boldsymbol{\theta} = atan2(Imag(fft(\mathbf{S}))/Real(fft(\mathbf{S}))). \tag{19}$$

The phase error vector is interpolated to be size N, which is the size of the original interferogram. In the next step, the notation $\boldsymbol{\theta}$ will refer to the phase error vector interpolated out to size N.

Step 6: Compute the Fourier transform of the entire vector **I** using the steps above and apply the phase error correction. The vectors $cos(\boldsymbol{\theta})$ and $sin(\boldsymbol{\theta})$ are computed and multiplied pointwise with the real and imaginary parts of the Fourier transformed interferogram **I**. Before computing the Fourier transform of **I**, the vector **I** is multiplied by a ”triangular” vector and has its left and right halves swapped as mentioned above for vector **S**. Thus, the uncalibrated spectrum is given by

$$\begin{aligned} \mathbf{U} &= cos(\boldsymbol{\theta}). * Real(fft(Swap(\mathbf{I}. * \mathbf{T}))) \\ &\quad + sin(\boldsymbol{\theta}). * Imag(fft(Swap(\mathbf{I}. * \mathbf{T}))), \end{aligned} \tag{20}$$

where the ”triangular” vector **T** in equation 20 has length N.

3 Detector Response and Calibration

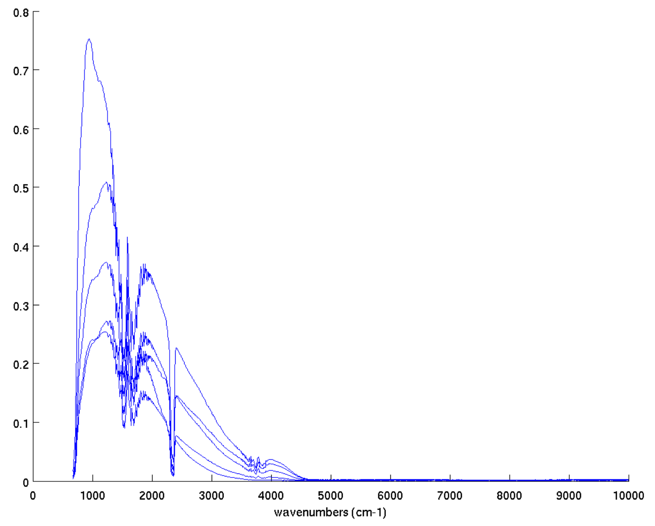
The MATLAB code for converting interferograms and for calibrating spectra has been applied to some recent data on the Helios project. A Bomem FTIR was used during this recent experimental campaign and it consisted of two detectors; a MCT (HgCdTe - Mercury Cadmium Telluride) detector and an InSb (Indium Antimonide) detector. The MCT detector is referred to as channel A or data A in the files obtained from the Bomem software. The InSb detector is referred to as channel B or data B. The MCT detector has a nominal wavenumber range of between 500 and 5000 cm^{-1} (2 μm - 20 μm). The InSb detector has a nominal range between 1800 and 8500 cm^{-1} (1.17 μm - 5.56 μm). However, the ranges for the detectors of the Bomem FTIR used is limited by the calibration data that was collected. The blackbody sources need to cover the wavenumber regions of interest in order to be useful for a given shot. Analysis of the calibration data provides information on the valid ranges of each detector.

The calibration data consists of using the FTIR sensor to collect data from blackbody sources at different temperatures. At a recent data collect, FTIR data was collected over several minutes from the blackbody sources at each of the different temperatures. When performing the calibration, an average of the collected spectra at each temperature was used to compute a mean spectra and these mean spectra are used in the calibration process. An average is used to reduce the impact of sensor noise in a single spectrum. The calibration data obtained from blackbody sources are converted to uncalibrated spectra (following the steps of the previous section) and then a wavenumber dependent gain and offset are estimated. The following linear model is used to represent the observed sensor radiance from the blackbody calibration data (see reference [3]),

$$L_i(\nu) = g(\nu)\frac{A_B}{d_B^2}B(\nu, T_i) + \sigma(\nu). \tag{21}$$

In equation 21, L_i is the average spectrometer response from the ith blackbody source (each i represents the blackbody data at a constant temperature), g is the estimated gain, σ is the estimated offset, B is the blackbody function, A_B is the blackbody aperture area, and d_B is the distance from the sensor to the blackbody. Given a collection of blackbody source data at different temperatures, the gain and offset are estimated with linear regression. Since the gain and offset are both dependent on the wavenumber, the linear regression is performed for each wavenumber. The equation 21 could include the atmospheric transmission between the blackbody source and FTIR sensor. The MATLAB code currently does not account for atmospheric transmission between the blackbody source and FTIR sensor.

Figure 5 illustrates blackbody calibration data from a recent data collect. The figure provides data for both detectors (channel A and channel B). The plots in the figure give the average sensor response L_i at the different temperatures. Therefore, the plots in the figure show the conversion of the raw interferograms to spectra. The figures are also given to demonstrate how the available blackbody data can constrain the valid wavenumber regions for an experimental shot. Based on figure 5 and the available calibration data, it can be seen that channel A is valid for a range of approximately 800 to 4300 cm^{-1} and and channel B is valid approximately 1800 to 7000 cm^{-1}.



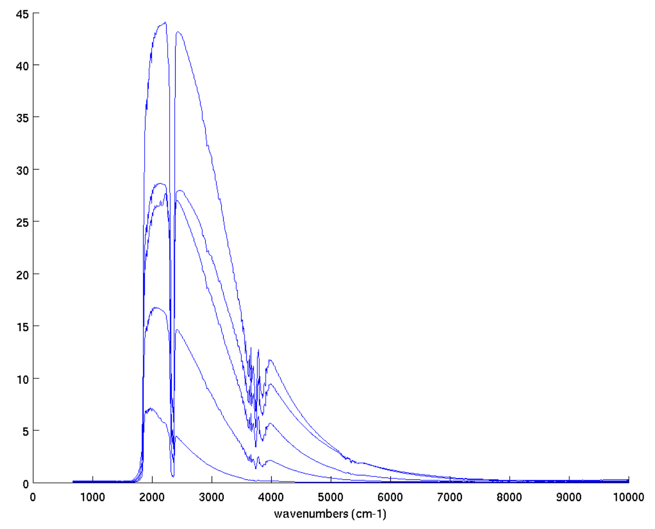


Figure 5: Example calibration data.

4 MATLAB Scripts

Listed below are the main MATLAB scripts for converting raw FTIR interferograms to calibrated spectra, together with corresponding information about the scripts. Given a new data set to calibrate, the scripts that need to be called are the following. First, the script getBBMean needs to be called to get the spectra from the blackbody sources. The script getBBMean calls helper functions that are currently called getCalibByIndex() and setNames() to set the directory paths and variable values needed by the code. These helper functions are specific to a data collect. Therefore, for a new data collect, new versions of getCalibByIndex() and setNames() would have to be written. A user can make new versions of these functions by copying the format of the existing functions and changing the corresponding values appropriately. The information returned by these helper functions consist of the directory location for the calibration data, the temperatures of the blackbody source data, distances between the sensor and blackbody, and the aperture areas of the blackbody sources. Second, the script convertIgm needs to be called to convert and calibrate the experimental shot data. This script also uses setNames() in a similar way as does getBBMean and, thus, requires that setNames() be changed for a new data collect.

In the following, references are made to the output from the Bomem software. The manufacturer (Bomem) of the FTIR sensor has software for analyzing data collected from their sensors. The Bomem software produces different types of files that will be referenced in this section. The Igm files from the Bomem software are the raw interferograms from the sensor. The RadInt files from the Bomem software contain calibrated spectra. Finally, the Spectrum files are uncalibrated spectra. That is, these files contain spectra that were converted from raw interferograms but have not been calibrated.

This section will provide a basic introduction to the different scripts and functions developed. Further documentation will be added to the MATLAB code itself in the near future.

Main Scripts and Functions

- **convertIgm.m**: Script to convert raw interferograms to calibrated spectra. The script calls function Igm_to_Spectra() to convert interferograms to uncalibrated spectra and function computeGainOffset() to get the gain and offset terms from the blackbody data. The function calibrateSpectra() is called by this script to calibrate the converted spectra using the computed gain and offset. This script writes out files with the computed calibrated spectra. The interferograms to convert and calibrate are read from a user specified file.
- **getBBMean.m**: Script to convert calibration source data to uncalibrated spectra. The uncalibrated spectra are written to files and used by the script convertIgm. The script only writes the mean spectra to a file. The calibration data usually consists of hundreds or thousands of spectra of the blackbody source at a constant temperature. The mean is used to reduce the impact of noise that exists in a single spectrum.
- **Igm_to_Spectra.m**: Function to convert raw interferograms to uncalibrated spectra. Follows the algorithm steps of section 2.4.
- **calibrateSpectra.m**: Function that takes uncalibrated spectra and computes the calibrated spectra, using the gain and offset. The gain and offset are input parameters. This function is called by the script convertIgm.

Auxiliary Functions

- **bb.m**: Function that computes the blackbody (Planckian) function for a given temperature. The function computes the blackbody function for an input vector of wavelengths. The returned radiance units are $W/m^2/sr/\mu m$.
- **bb_wn.m**: Function that computes the blackbody (Planckian) function for a given temperature. The function computes the blackbody function for an input vector of wavenumbers. The returned radiance units are $W/m^2/sr/cm^{-1}$.

- **getCalData.m**: This function is specific to a data collect. It takes as input an index that represents the day that the calibration data was collected. The function returns the blackbody calibration data for that day. The information returned by this function is the directory name of the calibration data and the corresponding file names.

- **getCalibByIndex.m**: Function that is specific to a data collect. The function returns a base file name, temperature, blackbody source area, and distance to the blackbody source for a given day and index number. The index number is assigned to each blackbody source collection on a given day and is arbitrary.

- **getNumberInStr.m**: Function that takes an input string and two substrings (*bound1* and *bound2*). The function then returns the number located between the two substrings. This function is called by readSpectrumTxt() and is used for convenience when parsing a Spectrum file.

- **getNumberPoints.m**: Function that returns the number of samples for both channel A and channel B. The function reads a Bomem file and returns the answers.

- **readLANLCal.m**: Function that reads calibrated spectra from a file that was generated by the MATLAB script convertIgm.

- **readRadInt.m**: Function that reads a Bomem RadInt text file.

- **readSpectrumTxt.m**: Function that reads a Bomem Spectrum text file. A Bomem Spectrum file contains uncalibrated spectra. That is, the spectra has been converted from the raw interferogram but it has not been calibrated.

- **setNames.m**: A function that is specific to a collect and it returns a MATLAB structure variable that contains information that is needed for a given shot. The returned information consists of directory names for the calibration data and interferogram data. Also, the returned structure variable contains information on the calibration sources, such as their temperatures, distances from the detector, and aperture areas.

- **wlrad2wnrad.m**: Function that converts radiance units from $W/sr/m^2/\mu m$ (wavelength based) to $W/sr/m^2/cm^{-1}$ (wavenumber based).

- **writeRadInt.m**: Function that writes out a calibrated spectrum to a file. This function is called by the MATLAB script convertIgm. The format of this file is similar to the RadInt files from Bomem but it is not exactly the same. Files written using the function writeRadInt() should be read with the function readLANLCal().

References

[1] P.R. Griffiths and J.A. de Haseth, *Fourier Transform Infrared Spectroscopy*. John Wiley & Sons, 1986.

[2] L. Mertz. *Transformations in Optics*. Wiley, New York, 1965.

[3] K.C. Gross, ”Phenomenological model for infrared emissions from high-explosive detonation fireballs,” Ph.D. dissertation, Air Force Institute of Technology, Dayton, OH, 2007.