Post processing field spectra in MATLAB

Guidelines for the FSF Post Processing Toolbox

I. Robinson and A. Mac Arthur
Field Spectroscopy Facility, Natural Environment Research Council, 2011

1 Introduction

The Field Spectroscopy Facility (FSF) Post Processing Toolbox is a MATLAB toolbox containg functions for importing spectra from field-portable specroradiometers and processing them. The following spectrometer file formats are supported:

- FieldSpec 3 and FieldSpec Pro, Analytical Spectral Devices
- GER 1500, Spectra Vista Corporation
- GER 3700, Spectra Vista Corporation
- HR-1024, Spectra Vista Corporation

Spectra acquired in the field must usually be post processed to select 'useful' spectra, remove invalid data, perform calibrations, make other calculations or apply processing functions. MATLAB provides an ideal environment for all these functions, as it incorporates many of the tools required for plotting, analysis and processing these data. This toolbox provides extra functions to perform the following tasks:

- Importing data from spectrometer data files directly into the MATLAB workspace
- Joining spectral regions of different detectors.
- Calculation of relative reflectance from target and reference spectra
- Calculation of absolute reflectance using a refrence panel calibration file
- Removal of erroneous water band data from reflectance spectra
- Savitzky–Golay smoothing
- Convolution of field spectra with the spectral response function (bands) of airborne and satellite instruments

This document outlines the installation, and use of these functions. The functions can be used for handling individual spectra or arrays of many spectra (up to several hundred). As field spectra are usually processed in batches, this documentation will describe the import and processing of multiple spectra together.

2 Installation

2.1 Check requirements

The functions require MATLAB 7.7 (R2008b) or later. The Savitzky–Golay smoothing functions additionally require the Signal Processing Toolbox.

For handling files from Analytical Spectral Devices (ASD) spectrometers ViewSpec Pro is recommended to convert ASD's binary format to the more easily-readible text format.

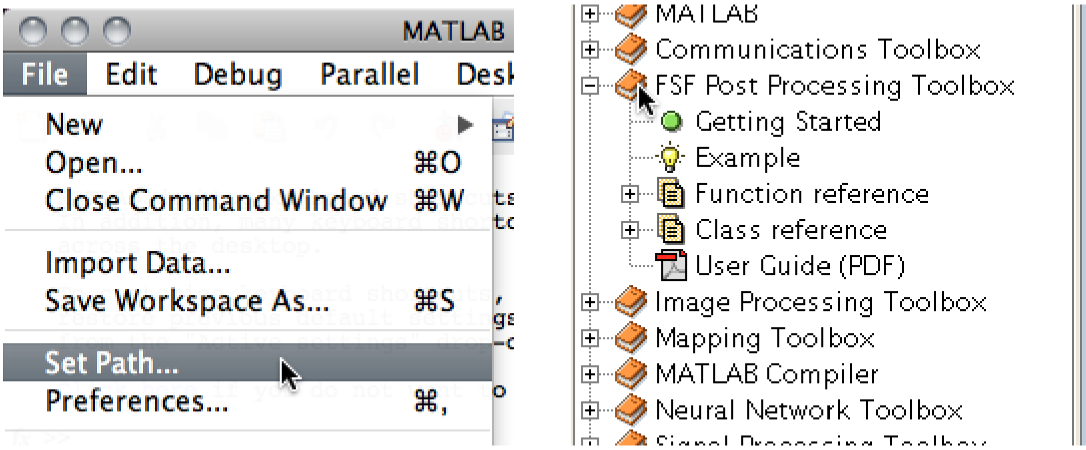


Figure 1: Add the directory containing the functions to your MATLAB path. Additional documentation is available in the help broswer.

Calculation of absolute reflectance requires a reference panel calibration file. For the Field Spectroscopy Facility's panels this file is available online.

Convolution of spectra to the spectral response functions of airborne or satellite instruments requires the band data for the specific instrument, available from the Field Spectroscopy Facility User Group.

2.2 Download and install

To install the functions:

1. Download and unzip `FSFPostProcessing1.2.5.zip` into a directory accessible to your MATLAB installation (e.g. on the same computer/server that runs MATLAB).
2. In MATLAB, add the folder `FSFPostProcessing1.2.5` to the path. This can be done from the Set Path dialog box, or by using the `path` function.

Once added to your path a brief description and syntax for each function can be printed using the help function; e.g. type `help smooth` for to check the syntax of the `smooth` function. Further documentation should appear in the help browser.

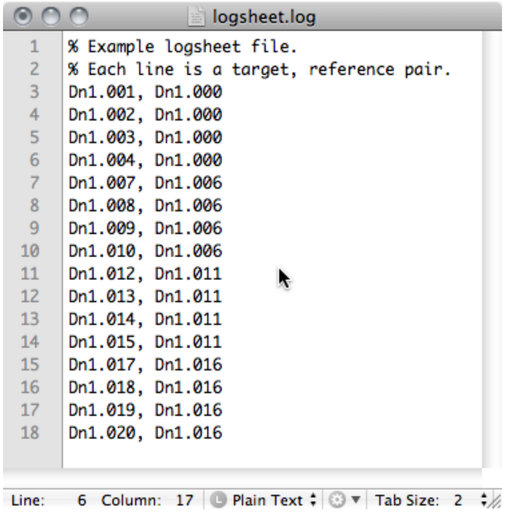
Locate examples

A number of example data files are in the examples directory.

3 Importing, plotting and handling

3.1 Importing

The first stage of post processing is to import the spectra from the spectrometer data files into the MATLAB workspace. This following subsections describe how to import data depending from different manufacturers' spectrometers.



```
% Example logsheet file.
% Each line is a target, reference pair.
Dn1.001, Dn1.000
Dn1.002, Dn1.000
Dn1.003, Dn1.000
Dn1.004, Dn1.000
Dn1.007, Dn1.006
Dn1.008, Dn1.006
Dn1.009, Dn1.006
Dn1.010, Dn1.006
Dn1.012, Dn1.011
Dn1.013, Dn1.011
Dn1.014, Dn1.011
Dn1.015, Dn1.011
Dn1.017, Dn1.016
Dn1.018, Dn1.016
Dn1.019, Dn1.016
Dn1.020, Dn1.016
```

Figure 2: Spectral data recorded in raw mode can be paired by creating a logsheet file.

3.1.1 Analytical Spectral Devices (ASD)

Files from ASD spectrometers (FieldSpec Pro and FieldSpec 3) must first be converted to text files using ViewSpec Pro[1]. This tool is available from ASD's Support Central web site. This toolbox has been tested with version 5.6.10 of ViewSpec Pro. The converted files will be in a text format with header information. Each file contains one spectrum.

To import the text files into the MATLAB workspace:

1. Change the current directory to the directory containg the files. This can be done with the current directory browser, by editing the current directory field on the desktop toolbar, or by using the `cd` command.

2. Type `spectra = ASDFieldSpec('filename')` where `filename` is the name of the file to be imported. Multiple files can be imported using wildcards. So to import all files with the `.txt` extension in the current directory type `s = ASDFieldSpec('*.txt')`.

The progress will be repoted during importing. The resulting spectra will be stored in an array in the workspace. If output is not suppressed (with a terminal semi-colon) a struct array will be shown:

```
s =

96x1 struct array with fields:

    name
    datetime
    header
    wavelength
    data
```

If the imported spectra were recorded in *white reference mode* the data will be reflectance values and further processing can continue.

If they were recorded in *raw mode*, the spectra must be sorted into reference and target spectra. Each target spectrum must be *paired* with its corresponding reference spectrum

[1] This toolbox cannot currently read ASD's binary file format.

using information from a logsheet recorded in the field. This information must be recorded in a separate text file. Figure 2 shows an example of a file called `logsheet.log`. This file can be used to sort and pair the imported spectra by typing `[tar, ref] = pair(s, 'logsheet.log')`. The original spectra will then be sorted and paired according to the pairings specified in the file:

```
tar =

1x16 struct array with fields:

    name
    datetime
    header
    wavelength
    data
    pair

ref =

1x4 struct array with fields:

    name
    datetime
    header
    wavelength
    data
```

Note that the target spectra have a field called `pair` which stores the name of the corresponding reference spectrum. This value is used later to calculate reflectance spectra.

3.1.2 Spectra Vista Corporation (SVC)

Files from SVC spectrometers (GER 1500, GER 3700 and HR-1024) are stored in the *signature* format. This is a text format with header information. Each file contains two spectra: a target spectrum and a reference spectrum.

To import the text files into the MATLAB workspace:

1. Change the current directory to the directory containg the files. This can be done with the current directory browser, by editing the current directory field on the desktop toolbar, or by using the `cd` command.

2. Type `[tar, ref] = SVCmodel('filename')` where model is the `model` of the spectrometer and `filename` is the name of the file to be imported. Multiple files can be imported using wildcards. For example, to import spectra recorded on a GER1500 spectrometer type `[tar, ref] = SVCGER1500('*.sig')`.

The progress will be repoted during importing. As each file contains a target spectrum and a reference specturm, the spectra are returned as two separate struct arrays for reference and target spectra respectively:

```
tar =

1x7 struct array with fields:

    name
    datetime
```

```
        header
        pair
        wavelength
        data

ref =

1x7 struct array with fields:

        name
        datetime
        header
        wavelength
        data
```

Note that the reference spectra may contain duplicates. This is because SVC signature files always contain the previously-recorded reference spectrum for each target spectrum.

When importing spectra from models with multiple detectors, such as the HR-1024, there may be overlapped data in the spectra. This occurs at the joins between the different detectors. The presence of overlapped data in imported spectra depends on the setting of the OverlapDataHandling.

Signature files from SVC instruments contain one target spectrum and its corresponding reference spectrum. Where the same reference spectrum is used for multiple target spectrum there will therefore be multiple copies of the reference spectrum in different files. This will lead to duplicates in the structure array of reference spectra when imported.

3.2 Plotting

To plot a struct array `s` of spectra type `plot([s.wavelength], [s.data])`. This plots spectral data columns againt a common set of wavelength values in the column vector `w.` . in the array. To plot an individual spectrum, say the sixth, type: `plot(s(6).wavelength, s(6).data)`. To plot a range of spectra, say the second to the fourth (inclusive), type: `plot([s(4:6).wavelength], [s(4:6).data])`.

When plotting reflectance data, it may be preferable to multiply the data values by 100 to produce a reflectance plot in per cent: `plot([s.wavelength], 100*[s.data])`

To change the horizontal scale from the default nanometres (nm) to micrometres (μm): `plot(0.001*[s.wavelength], [s.data])`

A legend can be added to the graph using MATLAB's `legend` function. For example, to add a legend showing the names of the spectra type: legend(spectra.name)

Axes labels, ranges and titles can be added using MATLAB's built-in comands or the plot editor. For example, the plot shown in Figure 3 was produced by typing:

```
plot([tar.wavelength], [tar.data]);
legend(tar.name);
xlabel('Wavelength (nm)');
ylabel('Intensity (a.u.)');
title('Target spectra recorded on SVC GER 1500');
```

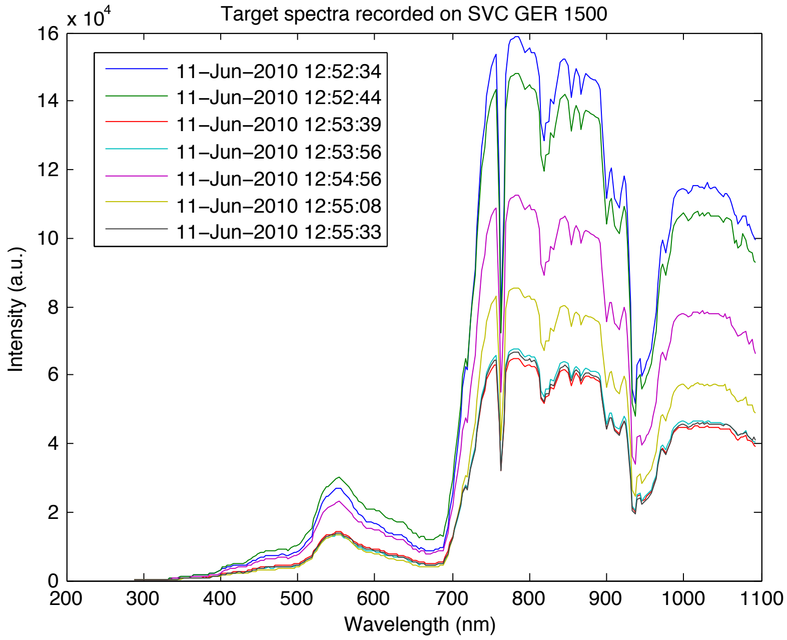


Figure 3: Plot of seven target spectra from a SVC GER 1500 spectrometer.

3.3 Handling spectra

Imported data are stored in a MATLAB structure array. This ensures that important header information (such as the name, date and integration times) is kept together with the spectral data (wavelength and data). There are a number of ways to view and manipulate spectrum data in the workspace, described in the following subsections.

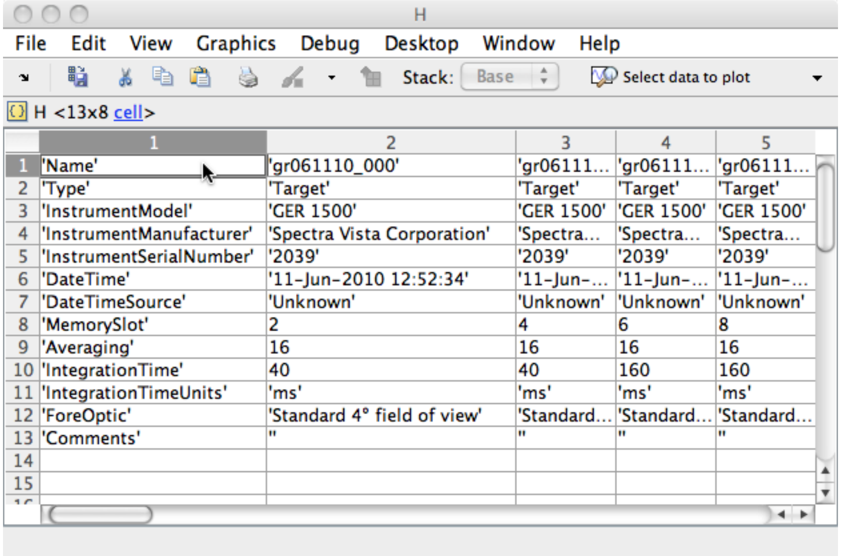
3.3.1 Structures

The spectra are stored as structures. A single spectrum would be a single structure, whereas multiple spectra are an array of structures (`struct array`).

The data are stored in the following fields:

- `name` is the name of the spectrum (usually taken from the filename)
- `datetime` is the date and time the spectrum was recorded[2].
- `header` is a struct containing all the header information from the original file.
- `pair` appears only in target spectra from SVC data files. It gives the name of the corresponding reference spectrum.
- `wavelength` is a column vector of wavelengths in nanometres.
- `data` is a column vector of the corresponding values. These will generally be uncalibrated digital numbers. For ASD spectrometers running in white reflectance mode these values will be reflectances in the range 0–1.

[2] The source of this data may be the spectrometer's internal clock, or may be the clock of the computer or PDA used to record the spectrum.



H <13x8 cell>

	1	2	3	4	5
1	'Name'	'gr061110_000'	'gr06111...	'gr06111...	'gr06111...
2	'Type'	'Target'	'Target'	'Target'	'Target'
3	'InstrumentModel'	'GER 1500'	'GER 1500'	'GER 1500'	'GER 1500'
4	'InstrumentManufacturer'	'Spectra Vista Corporation'	'Spectra...	'Spectra...	'Spectra...
5	'InstrumentSerialNumber'	'2039'	'2039'	'2039'	'2039'
6	'DateTime'	'11-Jun-2010 12:52:34'	'11-Jun-...	'11-Jun-...	'11-Jun-...
7	'DateTimeSource'	'Unknown'	'Unknown'	'Unknown'	'Unknown'
8	'MemorySlot'	2	4	6	8
9	'Averaging'	16	16	16	16
10	'IntegrationTime'	40	40	160	160
11	'IntegrationTimeUnits'	'ms'	'ms'	'ms'	'ms'
12	'ForeOptic'	'Standard 4° field of view'	'Standard...	'Standard...	'Standard...
13	'Comments'	''	''	''	''
14					
15					

Figure 4: Use the headers function to combine all headers into a single table.

3.3.2 Workspace browser

Open the workspace broswer by selecting Workspace from the Desktop menu, or type the command `workspace`. The spectra will be visible as struct arrays with the names given when imported. Double-click a struct array to browse its contents. If the strucure contains multiple spectra, they will be listed as a row vector. Double-clock to open a specific spectrum.

3.3.3 Listing fields

The fields of individual spectra can be viewed as a comma separated list. For example, typing `s.name` lists the names of all spectra in the structure `ref`.

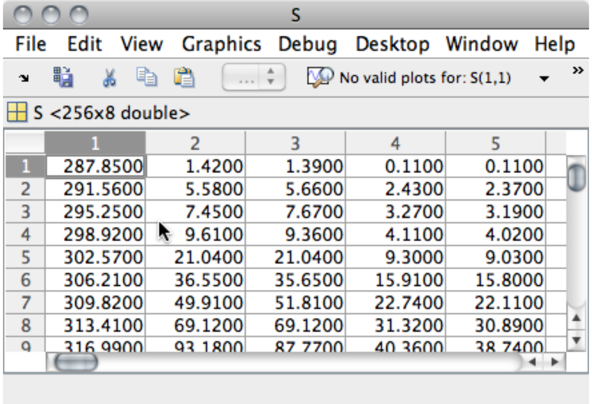
```
>> s.name

ans =
gr061110_000_reference

ans =
gr061110_001_reference

...
```

Similarly, typing `s.header` lists the headers of all spectra.

Whilst this is practical when working with a few spectra, for large numbers of spectra it will produce too much output. Instead data can be rearranged to a more suitable format as described in the following subsections.

3.3.4 Creating a table of header information

A function called headers provides a method to copy all header information into a single (large) cell array. For example, typing `H = headers(s)` will create a cell array `H` containing


S
File Edit View Graphics Debug Desktop Window Help
No valid plots for: S(1,1)
S <256x8 double>

	1	2	3	4	5
1	287.8500	1.4200	1.3900	0.1100	0.1100
2	291.5600	5.5800	5.6600	2.4300	2.3700
3	295.2500	7.4500	7.6700	3.2700	3.1900
4	298.9200	9.6100	9.3600	4.1100	4.0200
5	302.5700	21.0400	21.0400	9.3000	9.0300
6	306.2100	36.5500	35.6500	15.9100	15.8000
7	309.8200	49.9100	51.8100	22.7400	22.1100
8	313.4100	69.1200	69.1200	31.3200	30.8900
9	316.9900	93.1800	87.7700	40.3600	38.7400

Figure 5: Data can be copied into a numeric array if required.

all the header data. This can be viewed as a single table in the variable editor by typing `openvar H`.

3.3.5 Formatting data as a single numeric array

Spectrum data can be concatenated into a single numeric array by typing `A=[spectra.data]` where spectra is the name of the structure array. This way the spectrum data are arranged in columns, one column for each spectrum.

As all spectra usually have the same set of wavelength values, the wavelength of the first spectrum can be copied into a variable called `w` by typing: `w=s(1).wavelength`

Alternatively, it is often convenient to arrange data in a numeric array with wavelengths in the first column and data in subsequent columns. This can be achieved by typing `A=[s(1).wavelength, [spectra.data]]` where `w` is a column vector of wavelength values. Not that in doing this the data values of the spectrum 1 will be in column 1; the data values of spectrum 2 will be in column 3, etc.

4 Processing: basic

This section describes the processing functions in the toolkit.

4.1 Removing overlapped data

Spectroradiometers that cover both the visible near infared (VNIR) and short wave infared (SWIR) spectral regions have multiple detectors. For example the SVC HR1024 has a Si array detector for the VNIR spectral region and two InGaAs array detectors for the SWIR region. There is a small area of overlap at the long wave edge of the Si VNIR detector and the short wave edge of the first SWIR detector, typically 960 − −1050 nm.

Figure 6(A) shows the overlap region in a field spectrum. The overlapped data can cause problems for processing and must usually be removed prior to any further processing.

On ASD spectroradiometers the overlapped region is removed by the software, therefore ASD spectra will never contain overlapped data. On the SVC HR1024 the overlapped data

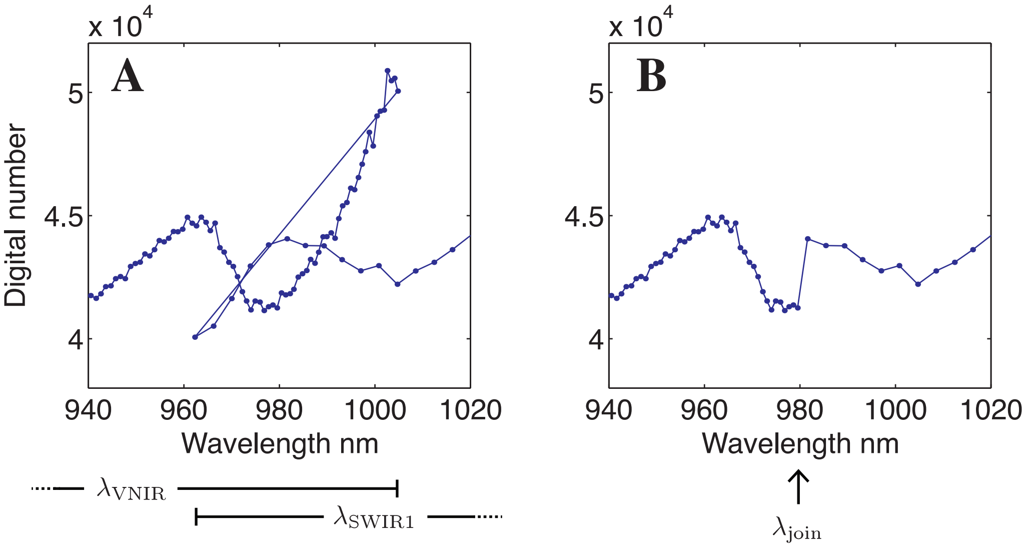


Figure 6: A small region of a spectrum recorded on an SVC HR1024 spectroradiometer showing overlapped data (A). The VNIR detector records wavelengths up to 1005 nm whilst the first SWIR detector starts at 962 nm. When the spectra are joined (B) at 980 nm a sharp jump in the intensity occurs.

may be *removed* or *preserved* depending on the settings chosen in the software (under the menu item 'Setup Overlap/Matching...'). We recommend setting this option to *preserve* and removing the overlap during post processing. (Spectra recorded with a handheld device (PDA) will always preserve the overlap.)

A function called `removeoverlap` is provided in the toolbox to join the VNIR and SWIR regions at a specified join wavelength in order to remove the overlap. Figure 6(B) shows an example of a spectrum with overlap removed. For example, to remove overlap from target and reference spectra type

```
tar_joined = removeoverlap(tar, 980)
ref_joined = removeoverlap(ref, 980)
```

After joining the spectra there is usually a sharp jump in the spectral data value either side of the join wavelength. This toolbox does not at present provide any matching algorithms to remove this feature, although some software provided by the manufacturers implements matching algorithms.

4.2 Relative reflectance

Spectral relative reflectance $R_{\mathrm{rel}}(\lambda)$ as a function of wavelength λ is defined as the ratio of the target spectrum $L(\lambda)$ to the corresponding relative reference spectrum $E_{rel}(\lambda)$

$$R_{rel}(\lambda) = \frac{L(\lambda)}{E_{rel}(\lambda)}$$

In the field $E_{rel}(\lambda)$ is measured by recording a spectrum of light reflected from a reference panel.

The function `relativereflectance` calculates the relative reflectance by dividing each target spectrum by its corresponding reference spectrum. Typing `R_rel=relativereflectance(tar,` `ref)` calculates relative reflectance spectra from the target spectra `tar` and reference spectra `ref`. Each spectrum in the struct array `tar` must have a field called `pair` which defines the corresponding reference spectrum in the series `ref`.

4.3 Absolute reflectance

Absolute spectral reflectance $R_{abs}(\lambda)$ as a function of wavelength λ is the ratio of the target spectrum L to the spectral irradiance $E(\lambda)$. However, in the field it is usually the relative reference spectrum $E_{rel}(\lambda)$ which is measured. This is the spectral irradiance which is reflected by the panel

$$E_{rel}(\lambda) = R_{panel}(\lambda)E(\lambda)$$

where $R_{panel}(\lambda)$ is the spectral reflectance of the panel.

The absolute reflectance can be calculated from the relative reflectance

$$R_{abs}(\lambda) = \frac{L(\lambda)}{E(\lambda)} = R_{panel}(\lambda)\frac{L(\lambda)}{E_{rel}(\lambda)} = R_{panel}(\lambda)R_{rel}(\lambda)$$

Therefore to convert absolute reflectance to relative reflectance the spectrum must be multiplied by the reflectance of the panel that was used to record it. The panel reflectance data are provided by a panel calibration file which records the panel reflectance over a set of wavelengths $R_{panel}(\lambda)$. If the calibration file is recorded with the same spectrometer as the field spectrum the set of wavelengths λ will be the same for both spectrum and panel. Otherwise the panel spectrum must be interpolated to the wavelengths of the relative reflectance spectrum.

Panel calibration files can be provided in one of two formats:

1. A comma-separated values (CSV) file with a header that gives some metadata about the panel.
2. An Excel spreadsheet downloaded form the Field Spectroscopy Facility's web site web site (for the Facility's panels).

For a file in CSV format, the file format is as follows:

rt the panel calibration file type

```
Panel calibration file for FSF Post Processing Toolbox
Name:  SRT #011
Date of calibration:  2011-04-03
350,0.980
351,0.980
352,0.981
```

...and so on. The first column is wavelength and the second is reflectance (in the range 0–1). Noe that the header data is *required.*

To import the panel calibration from a CSV file type

```
p = importpanelcalibration('panel.csv')
```

From an Excel spreadsheet the serial number of the panel must also be specified

```
P = importpanelcalibration('FSF Ref Panel Cal Data 2010_1.xls', 'SRT
#008')
```

MATLAB may give several warnings when importing an Excel spreadsheet. However if the import is successful these may be ignored.

The panel calibration spectrum can be plotted

```
plot(P.wavelength, P.data)
```

Once panel data are loaded, the absolute reflectance can be calculated using the function `absolutereflectance`. Typing

```
R_abs=absolutereflectance(R_rel, P)
```

calculates the absolute reflectance from the struct array or relative reflectance spectrum using the panel calibration data in the specified spreadsheet. `'SERIAL'` is the serial number of the panel, e.g. `'SRT #004'`.

Once calculated, absolute reflectance can be plotted using MATLAB's plot function. Reflectance values are (normally) in the range 0–1. A graph of reflectance as a percentage can be plotted by multiplying the reflectance values by 100:

```
plot([R_abs.wavelength], 100*[R_abs.data]);
legend(R_abs.name);
xlabel('Wavelength (nm)');
ylabel('Absolute reflectance (%)');
title('Reflectance spectra recorded on SVC GER 1500');
```

4.4 Smoothing

The function `smooth` smooths spectra with Savitzky–Golay smoothing [1]. Typing `S=sgsmooth(R_abs)` smooths a struct array of spectra `R_abs`, returning a struct array of smoothed spectra `S`. The smoothing reduces the wavelength range of the spectra, so the spectra in `S` will have a new wavelength scale.

4.5 Removing water bands

The function removewater deletes data in the water band regions which are usually erroneous. Type `W=removewater(S)` to delete water bands from a struct array `S`, returning a new struct array `W`.

The data points are removed by replacing them with the special value *not a number* (NaN). This special value is used to indicate that data have been removed. Further processing of data in this region causes NaN.

5 Processing: advanced

5.1 Convolution

The function `convolve` provides a means to convolve a spectrum [2] or spectra with the spectral response functions of airborne or satellite instruments. This procedure is used to compare data from high-resolution field-portable spectroradiometers with lower-resolution instruments.

The first step in this processing function is to download the spectral response functions (bands) for the specific instrument. These data are provided as XML files and are listed on the Field Spectroscopy Facility's web site.

The class `Band` can be used to load response data into MATLAB. For example, to load spectral response data for the POLDER instrument type

```
POLDER = Band('http://fsf.nerc.ac.uk/user_group/bands/POLDER.xml')
```

By default the response data are normalized such that the area under each curve is 1. The `normalizeHeight` method can be used to normalize such that the height of each response function is 1:

```
POLDER.plot
POLDER.normalizeHeight.plot
```

To convolve with a spectrum `S`, type

```
T = convolve(S, POLDER)
```

If the spectrum `S` and the bands `POLDER` are defined on different wavelength scales, the bands will be interpolated.

To plot the output spectrum `T`

```
plot(T.wavelength, T.data)
```

Note that the convolve function cannot be used on spectra where there are overlapping data.

5.2 Feature depth estimation

The depth of an absorption feature in a spectrum can be estimated using the method described by Clark and Roush [3]. The function `featuredepth` provides an illustration of the method.

References

[1] Savitzky, Abraham and Golay, M J E, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry*, volume 36, no. 8, pp. 1627–1639, 1964

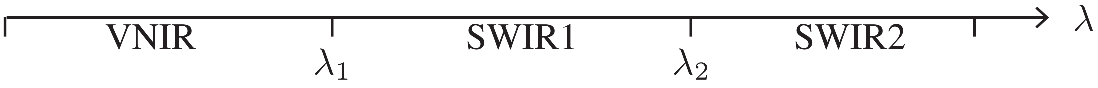
[2] James, J F, *Spectrograph Design Fundamentals*, Cambridge University Press, 2007

[3] Clark, R N and Roush, T L, "Reflectance spectroscopy: quantitative analysis techniques for remote sensing applications," *J Geophys Res*, volume 89, p. 6329, 1984

Appendix: normalization of ASD spectra in raw mode

ASD spectra recorded in *raw mode* must be normalized to account for the different responses of the detectors. The normalization is performed during import and the code is contained in the function `ASDFieldSpec`. (Reflectance spectra recorded in *white reference mode* are not normalized.)

The ASD FieldSpec 3 and FieldSpec Pro spectroradiometers have three separate detectors for three different spectral regions. A silicon photodiode array detects light in the visible-near-infrared (VNIR) region. Two InGaAs photodiodes cover two short wave infrared spectral regions labelled SWIR1 and SWIR2.



λ_1 and λ_2 are the join wavelengths. These may be different for different instruments, but the values are saved in ASD spectrum files. They can be found in the `header` structure in fields named `Join1Wavelength` and `Join2Wavelength`.

The pixel values DN are normalized differently in each spectral region:

$$\mathrm{DN}_{\mathrm{norm}}(\lambda) = \begin{cases} \frac{\mathrm{DN}(\lambda)}{T_{\mathrm{VNIR}}} & \lambda \leqslant \lambda_1 \\ \frac{G_{\mathrm{SWIR1}}\mathrm{DN}(\lambda)}{2048} & \lambda_1 < \lambda \leqslant \lambda_2 \\ \frac{G_{\mathrm{SWIR2}}\mathrm{DN}(\lambda)}{2048} & \lambda_2 < \lambda \end{cases}$$

where T_{VNIR} is the integration time of the VNIR photodiode array, G_{SWIR1} is the gain of the SWIR1 photodiode and G_{SWIR2} is the gain of the SWIR2 photodiode. These values are saved in ASD spectrum files, and are imported into the MATLAB workspace as fields in the `header` structure called `VNIRIntegrationTime`, `SWIR1Gain` and `SWIR2Gain` respectively.