Emissivity-temperature separation algorithm

Overview

The emissivity-temperature separation algorithm is applied to OTES calibrated radiance spectra and results in the derivation of the maximum brightness temperature and an emissivity spectrum (emissivity vs. wavenumber, where emissivity is a unitless quantity ranging from 0 - 1). Radiance spectra include contributions from both temperature and emissivity, so to compare spectra of surfaces having differing temperatures, it is desirable to remove the temperature component and compare emissivity spectra. Temperature is also an input to thermal models. The temperature is derived from a portion of the spectrum that is assumed to be spectrally featureless and have an emissivity of (or near) 1.0. The calibrated radiance spectrum is divided into a Planck function at the derived temperature to yield the emissivity spectrum.

Algorithm Description

1. Read in OTES calibrated radiance spectra
2. Run Davinci routine emissivity() using radiance spectra as input; options include (not a comprehensive listing):
    - filter size for determining maximum brightness temperature
    - optional maximum emissivity (e.g., 0.98 instead of default of 1.0)
    - optional starting wavenumber for brightness temperature calculation
    - optional ending wavenumber for brightness temperature calculation
3. Save output HDF structure containing identifying information for input calibrated radiance spectrum and wavenumber axis, derived maximum brightness temperature, emissivity spectrum, optional parameter values, and ancillary geometric information. Filenames will follow established SPOC convention.

Example

```
dv> calrad = load('otes_sim_calrad.hdf')
dv> e = emissivity(otes=1, calrad.spectral_qube.data, xaxis=xaxis)
dv> write(e, 'otes_detailed_survey_1030am_Tb-emissivity.hdf', format=hdf)
```

--> Calculates brightness temperature for input calibrated radiance spectra where e might look like this:

```
dv> e

struct, 9 elements

emissivity: 30x1x92 array of float, bsq format [120 bytes]
incidence: 30x1x1 array of float, bsq format [120 bytes]
latitude: 30x1x1 array of float, bsq format [120 bytes]
longitude: 30x1x1 array of float, bsq format [120 bytes]
maxbtemp: 30x1x1 array of float, bsq format [120 bytes]
xaxis: 1x1x92 array of float, bsq format [368 bytes]
max_emiss: 1x1x1 array of float, bsq format [268 bytes]

label: Text Buffer with 30 lines of text
```

1. OCK 3021, ICK 1500
2. OCK 3021, ICK 1501
...
30. OCK 3021, ICK 1529

emissivity_version: 2.0

.....

Note: This script is intended for use with many different data sets, of which OTES data is only one. As such, there are a number of options and outputs that may not be relevant to OTES data analysis. At present, the function emissivity() does not write out the output file - this requires a wrapper script that is in production.

Installation, Program Files, User's Guide, and Test Data

Installation: The script exists within the Davinci software distribution as the function "emissivity" - a copy is attached below as a Davinci resource file for reference. Users must run library_update(update=1, beta=1) and then run library_update(update=1,version=2.16) to access a non-public version of the emissivity function that includes OTES compatibility. The user's guide can be read in Davinci by typing "emissivity()" (without quotes); at present, the OTES functionality is not advertised but can be accessed by the option "xaxis=xaxis" (see example above), which assumes that a variable called "xaxis" exists and contains the OTES x-axis. This variable "xaxis" can be created using the command xaxis=make_band(inst="omtes").

The SPOC should ingest the emissivity spectra and temperatures; emissivity spectra should be returnable and temperatures should be searchable and returnable.

Input/output from/to the SPOC database is not yet implemented.