BErkeley High Resolution (BEHR) NO_2 product - User Guide

Josh Laughner

May 8, 2018

Contents

1 Reading

1. Russell et al. (2011) - description of original BEHR algorithm
2. Laughner and Cohen (2017) - description of v2.1C of the BEHR algorithm

2 Product overview

2.1 Product types

Currently, we have two data products available: one at the native OMI pixel resolution and one in which each swath has been gridded to a $0.05^\circ \times 0.05^\circ$ fixed grid. The gridded product is ideal for users who simply wish to obtain an NO_2 VCD, as the latitude and longitude of each grid point will remain fixed over time, whereas the native OMI pixels do not. However, the native OMI resolution files have additional variables compared to the gridded product, such as scattering weights, averaging kernels, our NO_2 *a priori* profile, etc. that will be useful to users wishing to modify the product in some way.

2.2 Version numbering

The BEHR version numbering system combines the OMNO2 version number with an internal version letter. So, v2.1A represents the first BEHR product based on version 2.1 of the NASA OMNO2 product. A subsequent version of BEHR still based on version 2.1 of OMNO2 would be v2.1B. If OMNO2 were to update to version 3, the BEHR product based on that would be v3.0A.

Should a minor change be made (e.g. one that adds information to the output but does not change the core algorithm), a revision number will be appended to the version number. For example, v2.1A and v2.1Arev0 will be the same, but v2.1Arev1 would indicate this sort of minor change.

The version number may also be formatted as, e.g. v2-1A. This is used as the version string in file names to prevent any issue with file systems unable to handle a . in a filename that is not separating the file extension. This form is completely equivalent to the one with the period and is used interchangeably.

2.3 File format

All products will be made available as HDF version 5 files (`https://www.hdfgroup.org`). Please note when trying to open these that many programming languages and utilities have different commands and tools for opening version 4 and 5 HDF files. If you are having trouble opening these files:

1. Ensure that you are using the correct command for an HDF5 file, not an HDF4 file.
2. Try using HDFView (available from `https://www.hdfgroup.org/products/java/index.html` to browse the file. This will confirm that it downloaded properly.
3. Check if the utility or programming language you are using requires the HDF5 library (`https://www.hdfgroup.org/HDF5/`) to be installed.

For both the native and gridded products, the HDF files are organized similarly. Under the `/Data` group, each swath is contained within its own group, named as `Swath#`. There will be 3–5 swaths per day. Each swath will contain all relevant variables as datasets.

Starting with BEHR version 2.1A, fill values will be directly stored in the HDF FillValue information for each dataset, along with four additional attributes: Description, Range, Product, and Unit. *Description* is a brief, one-line description of the meaning of each variable. *Range* is the range of values that variable may correctly take on. *Product* indicates whether this dataset is copied directly from the NASA standard product (represented by **SP**), the OMPIXCOR product (**PIXCOR**), or is added by the BEHR product (unsurprisingly represented by **BEHR**). Finally, *Unit* is the physical unit assigned to each dataset.

Starting a v3.0A, we are no longer providing the native pixels as comma separated values. With the proliferation of available data into separate products using daily and monthly profiles, providing both .hdf and .txt files was no longer practical.

2.4 Tools for working with HDF files

The following programs or programming languages are known to be able to read HDF files:

- MATLAB: current versions have high-level functions such as `h5info` and `h5read` which can easily read in HDF5 files. This does not seem to rely on the external HDF library.
- Python: the `h5py` package (http://www.h5py.org) can read HDF5 files, however it does depend on having the HDF library installed, at least on Unix based systems.
- IDL: various users have sucessfully read HDF5 files in IDL; however since we do not use it ourselves, we cannot offer specific advice on the best way to do so.
- Igor Pro, v. > 5.04: http://www.wavemetrics.com/products/igorpro/dataaccess/hdf5.htm
- GNU Octave: https://www.gnu.org/software/octave/

This is not an exclusive list, however these are common scientific software packages that indicate they have the capability to read HDF5 files. Note that our experience is focused on MATLAB and Python, so our ability to offer specific advice for other utilities is limited.

3 Variables

3.1 List of variables

Table 1 will list the attributes of all variables found in the BEHR files. Most categories are fairly self-explanatory. *Product* indicates whether the variable is directly copied from the NASA OMNO2 product (SP), the OMPIXCOR product (PIXCOR) or calculated from BEHR. *Gridding* indicates whether the variable will be contained in the native OMI pixel resolution files ("Native") or both the native and gridded files ("both").

For more information on the SP variables, see the links at https://disc.sci.gsfc.nasa.gov/datasets/OMNO2_V003/summary?keywords=omno2, especially the Readme and OMNO2 Data Format. Likewise, see https://disc.sci.gsfc.nasa.gov/datasets/OMPIXCOR_V003/summary?keywords=ompixcor for information on the PIXCOR variables.

Here we will describe primarily the BEHR variables in detail, although we will describe the presentation of some OMNO2 variables where necessary.

- **Areaweight**: field for weighting the temporal average of the gridded product; calculated as the inverse of the FoV75 Area.
- **BEHRAMFTrop**: the tropospheric AMF calculated by the BEHR algorithm using high resolution albedo, terrain pressure, and NO_2 *a priori* inputs. This AMF estimates a ghost column as well, and corresponds to the VCD in the BEHRColumnAmountNO2Trop field.
- **BEHRAMFTropVisOnly**: the tropospheric AMF calculated by the BEHR algorithm that does not attempt to estimate a ghost column. This corresponds to the VCD in BEHRColumnAmountNO2VisOnly.
- **BEHRAvgKernels**: a vector of averaging kernels that can be used when comparing model output to the BEHR product. See §4.4 for details.
- **BEHRColumnAmountNO2Trop**: the tropospheric NO_2 column calculated using the BEHR algorithm as $V_{\text{BEHR}} = V_{\text{NASA}} \cdot A_{\text{trop, NASA}}/A_{\text{trop, BEHR}}$, where V_{BEHR} and V_{NASA} are the vertical column densities for BEHR and NASA respectively, and $A_{\text{trop, NASA}}$ and $A_{\text{trop, BEHR}}$ are the tropospheric AMFs. This field includes estimated ghost columns that are obscured by clouds. The estimate essentially scales the above-cloud component by the ratio of the above-cloud to total column from the model *a priori* profile.
- **BEHRColumnAmountNO2TropVisOnly**: the tropospheric NO_2 column calculated using the BEHR algorithm and the same equation as for the previous variable, except that the visible-only AMF is used instead. This retrieves only the visible NO_2 column, thus this is the NO_2 column to ground for the clear fraction of the pixel, and only the above-cloud NO_2 column for the cloudy fraction of the pixel.
- **BEHRNO2Apriori**: this is the NO_2 profile used as the *a priori* for the AMF calculations. It is given as unscaled mixing ratios; so multiplying these numbers by the number density of air will directly give the number density of NO_2. See §4.4 for more information.
- **BEHRPressureLevels**: the pressure level that correspond to the BEHRAvgKernels, BEHRNO2Apriori, and BEHRScatteringWeights vectors. See §4.4 for more information.
- **BEHRQualityFlags**: flags field that combines the NASA XTrackQualityFlags and VcdQualityFlags summary bits with processing errors or warnings from the BEHR algorithm. The meaning of each bit is given in Table 2. Discussion of how to use these flags is given in §4.1.1.
- **BEHRScatteringWeightsClear**: the vectors of scattering weights for clear-sky (non-cloudy) conditions that can be used to recompute AMFs using custom *a priori* profiles. See §4.4 for more details.

- **BEHRScatteringWeightsCloudy**: the vectors of scattering weights for cloudy conditions that can be used to recompute AMFs using custom *a priori* profiles. See §4.4 for more details.

- **BEHRSurfacePressure**: The surface pressure of the pixel used in the AMF calculation; it is calculated by using the difference in surface elevation between WRF and the GLOBE database to scale the WRF surface pressure.

- **BEHRTropopausePressure**: the tropopause pressure derived from WRF temperature profiles and used as the upper limit in the BEHR AMF calculation.

- **GLOBETerrainHeight**: The average surface elevation of the pixel calculated from the GLOBE database (http://www.ngdc.noaa.gov/mgg/topo/globe.html).

- **MODISAlbedo**: The average surface reflectance of the pixel calculated from Band 3 of the MCD43D combined MODIS BRDF product for the viewing geometry of this pixel.

- **MODISCloud**: The cloud fraction of the pixel averaged from Aqua MODIS MYD06 cloud fraction data.

- **RelativeAzimuthAngle**: the azimuthal difference between solar and viewing angles, calculated as:

$$x = |\text{SAA} + 180 - \text{VAA}|$$
$$\text{RAA} = \begin{cases} 360 - x \text{ if } x > 180 \\ x \text{ otherwise} \end{cases}$$

  The extra factor of 180 transforms the RAA into the definition used in the TOMRAD look up table.

- **VcdQualityFlags**: directly taken from the OMNO2 product; however, while they should always be considered a bit array (and thus integers) in some versions of BEHR these are converted to floating point numbers. Also, in the gridded product, in any case where more than one pixel contributed to a grid cell, the value given will be the result of a bitwise OR operation applied to the flags from each pixel. Thus, if a flag is set in any contributing pixel, it will be set in the grid cell.

- **WRFSurfacePressure**: surface pressure calculated as a simple average of WRF profile falling within the pixel. Not used in the AMF calculation.

- **XTrackQualityFlags**: same notes as VcdQualityFlags.

Table 1: Variables found in the BEHR files.

Variable	Gridding	Product	Range	Unit
AmfStrat	Native	SP	$[0, \infty)$	unitless
AmfTrop	Both	SP	$[0, \infty)$	unitless
Areaweight	Gridded	BEHR	$[0, \infty)$	km^{-2}
BEHRAMFTrop	Both	BEHR	$[0, \infty)$	unitless
BEHRAMFTropVisOnly	Both	BEHR	$[0, \infty)$	unitless
BEHRAvgKernels	Native	BEHR	$[0, \infty)$	unitless
BEHRColumnAmountNO2Trop	Both	BEHR	$[0, \infty)$	molec. cm^{-2}
BEHRColumnAmountNO2TropVisOnly	Both	BEHR	$[0, \infty)$	molec. cm^{-2}
BEHRNO2Apriori	Native	BEHR	$(-\infty, \infty)$	unscaled mixing ratio
BEHRPressureLevels	Native	BEHR	$[0, \infty)$	hPa
BEHRQualityFlags	Both	BEHR	$[0, 2^{32} - 1]$	bit array flag
BEHRScatteringWeightsClear	Native	BEHR	$[0, \infty)$	unitless
BEHRScatteringWeightsCloudy	Native	BEHR	$[0, \infty)$	unitless
BEHRSurfacePressure	Both	BEHR	[0, 1013]	hPa
BEHRTropopausePressure	Both	BEHR	$[0, \infty)$	hPa
CloudFraction	Both	SP	[0, 1]	unitless
CloudPressure	Native	SP	$[0, \infty)$	hPa
CloudRadianceFraction	Both	SP	[0, 1]	unitless
ColumnAmountNO2	Native	SP	$[0, \infty)$	molec. cm^{-2}
ColumnAmountNO2Strat	Native	SP	$[0, \infty)$	molec. cm^{-2}
ColumnAmountNO2Trop	Both	SP	$[0, \infty)$	molec. cm^{-2}
ColumAmountNO2TropStd	Native	SP	$[0, \infty)$	molec. cm^{-2}
FoV75Area	Native	PIXCOR	$[0, \infty)$	km^2
FoV75CornerLatitude	Native	PIXCOR	[-90, 90]	degrees
FoV75CornerLongitude	Native	PIXCOR	[-180, 180]	degrees
GLOBETerrainHeight	Native	BEHR	$[-\infty, \infty)$	hPa
Latitude	Both	SP	[-90, 90]	degrees
Longitude	Both	SP	[-180, 180]	degrees
MODISAlbedo	Both	BEHR	[0, 1]	unitless
MODISCloud	Both	BEHR	[0, 1]	unitless
RelativeAzimuthAngle	Native	BEHR	[0, 180]	degrees
Row	Both	SP	[0, 59]	unitless
SlantColumnAmountNO2	Native	SP	$[0, \infty)$	molec. cm^{-2}
SolarAzimuthAngle	Native	SP	[-180, 180]	degrees
SolarZenithAngle	Native	SP	[0, 90]	degrees
SpacecraftAltitude	Native	SP	$[0, \infty)$	m
SpacecraftLatitude	Native	SP	[-90, 90]	degrees
SpacecraftLongitude	Native	SP	[-180, 180]	degrees
Swath	Native	SP	$[0, \infty)$	unitless
TerrainHeight	Native	SP	$(-\infty, \infty)$	m

Continued on next page

Table 1 – *Continued from previous page*

Variable	Gridding	Product	Range	Unit
TerrainPressure	Native	SP	[0, ∞)	hPa
TerrainReflectivity	Native	SP	[0, 1]	unitless
TiledArea	Native	PIXCOR	[0, ∞)	km^2
TiledCornerLatitude	Native	PIXCOR	[-90, 90]	degrees
TiledCornerLongitude	Native	PIXCOR	[-180, 180.0]	degrees
Time	Native	SP	[0, ∞)	s
VcdQualityFlags	Both	SP	N/A	bit array flag
ViewingAzimuthAngle	Native	SP	[-180, 180]	degrees
ViewingZenithAngle	Native	SP	[0, 90]	degrees
WRFSurfacePressure	Native	BEHR	[0, ∞)	hPa
XTrackQualityFlags	Both	SP	N/A	bit array flag

3.2 Attributes

3.2.1 Swath level

Each orbit group (`/Data/SwathN`) has a number of attributes that describe important aspects of the BEHR algorithm that relate to that swath.

- **Simple metadata:**
  - *Description* generic description of the data; indicates if at native pixel or regridded resolution
  - *Version* gives the BEHR version used to produce this file
  - *Date* gives the date the data in the file is of
  - *BEHRRegion* gives which region the product is for; current values as "us" (continental United States) and "hk" (Hong Kong).
- **Input files:**
  - *OMNO2File* gives the NASA SP file ingested as the base of the retrieval
  - *OMPIXCORFile* gives the OMPIXCOR file ingested for the FoV and Tiled pixel corners and areas.
  - *MODISCloudFiles* gives the MODIS MYD06_L2 files ingested to provide the data in the MODISCloud dataset.
  - *MODISAlbedoFile* lists the MCD43D files ingested to calculate the surface reflectivity given in the MODISAlbedo field
  - *BEHRWRFFile* gives the WRF-Chem file ingested to provide NO_2 and temperature profiles

Error bits		
Bit position	Bit value	Bit meaning
1	1	Low quality flag: set if this pixel should not be used to generate high quality full tropospheric VCD (visible + ghost column) data. Set for any pixel with the critical error bit set, MODIS BRDF quality warning set, or cloud fraction > 20%.
2	2	Critical error bit: set if any error present. Do not use either the standard or visible-only VCD from this pixel.
3	4	AMF error: set if either the BEHR AMF or BEHR visible-only AMF is less than or equal to the minimum allowed value
4	8	VcdQualityFlags: set if the VcdQualityFlags field was not an even integer (i.e. its own summary bit was set)
5	16	XTrackFlags: set if the XTrackFlags field was > 0, indicating the presence of the row anomaly
6–16	32–32768	Unused: reserved for future use
Warning bits		
Bit position	Bit value	Bit meaning
17	65536	High cloud fraction: OMI geometric cloud fraction exceeds 20%. Not recommended for use if trying to retrieve a full tropospheric column (visible + ghost).
18	131072	MODIS ocean flag: MODIS land use table indicates that this pixel is majority ocean, used reflectance from a look up table based on solar zenith angle derived from the Couple Ocean-Atmosphere Radiative Transfer Model (https://cloudsgate2.larc.nasa.gov/jin/coart.html)
19	262144	MODIS BRDF quality flag: the average quality value for the MCD43D* values averaged to this pixel exceeds 2.5 or > 50% of the MODIS grid cells in this OMI pixel are fill values. Accuracy of the surface reflectance used may be lower; use with caution.
20	2^{19}	Cloud pressure is less than BEHRTropopausePressure. This means that the AMF calculation assumed no above-cloud component.
21	2^{20}	Tropopause pressure was interpolated from neighboring pixels because the algorithm could not find a level in the temperature profile with a lapse rate < 2 K/km.
22–31	2^{21}–2^{30}	Unused: reserved for future use
32	2^{31}	Fill value. If this bit is set, it indicates the quality flag itself was set as a fill value. (This should not happen, if it does, please contact the BEHR maintainer.)

Table 2: Meaning of the various bits in the BEHRQualityFlags field.

- **Processing decisions:**
  - *BEHRProfileMode* will be the string "monthly" or "daily" indicating whether monthly or daily NO_2 and temperature profiles were used.
  - *BEHRWRFPressureMode* will be the string "precomputed" or "online". "precomputed" indicates that total pressure (for the vertical coordinate) was calculated with `calculated_met_quantities.nco` from https://github.com/CohenBerkeleyLab/WRF-nco-tools during subsetting or averaging of WRF output files. "online" indicates that total pressure was calculated from WRF `P` and `PB` variables within the Matlab code (`rProfile_WRF.m`).
  - *BEHRWRFTemperatureMode* will be the string "precomputed" or "online". Like BEHRWRFPressureMode, this indicates whether WRF's temperature output was converted from perturbation potential temperature to absolute temperature with `calculated_met_quantities.nco` ("precomputed") or in the Matlab code (`convert_wrf_temperature.m`).
- **Code commit IDs:**
  - Each of these attributes begins with *GitHead*
  - The middle segment indicates the repository that the commit is from:
    - `Core`: BEHR-core at https://github.com/CohenBerkeleyLab/BEHR-core
    - `BEHRUtils`: BEHR-core-utils at https://github.com/CohenBerkeleyLab/BEHR-core-utils
    - `GenUtils`: Matlab-Gen-Utils at https://github.com/CohenBerkeleyLab/Matlab-Gen-Utils
    - `PSM`: BEHR-PSM-Gridding at https://github.com/CohenBerkeleyLab/BEHR-PSM-Gridding
    - `MatPyInt`: MatlabPythonInterface at https://github.com/CohenBerkeleyLab/MatlabPythonInterface
    - `WRFUtils`: WRF_Utils at https://github.com/CohenBerkeleyLab/WRF_Utils
  - The last segment indicates which step of the algorithm that commit was the active commit for:
    - `Read`: reading in ancillary data (`read_main.m` in BEHR-core).
    - `Main`: BEHR AMF and VCD NO_2 calculation (`BEHR_main.m` in BEHR-core).
    - `Pub`: publishing, writing the .hdf files from the Matlab .mat files (`BEHR_publishing_main.m` in BEHR-core).
  - The hash given as the attribute value is the Git SHA-1 hash of the commit that was HEAD at the time that step was run.
    - So `GitHead_Core_Main = 82f51557e00b8227cd7ca5490d9b77112a72df4f` means that commit 82f5155 of the BEHR-core repository was the most recent commit when `BEHR_main.m` was run to produce the data for that file.

* If the value is "N/A" that means that repository was not used at the time to produce that file.

4 Considerations when working with BEHR files

4.1 Pixel filtering

4.1.1 Native pixel data

For users wishing to identify high quality NO_2 column information to ground, we have added our own flag field that combines relevant flags from NASA with processing flags from our algorithm. This is the BEHRQualityFlags field. For users who want the simplest way to identify good quality data for full tropospheric columns (i.e. BEHRColumnAmountNO2), only use pixels for which BEHRQualityFlags is an even integer (i.e. the least significant bit is 0). This automatically rejects any pixels that had an error during processing, any pixels for which the NASA VcdQualityFlags or XTrackQualityFlags fields indicate the pixel should not be used, any pixels with low quality MODIS BRDF data (see below), and (as of 3 Oct 2017) any pixels that have an OMI geometric cloud fraction $> 20\%$.

More advanced users who want more control over cloud filtering should instead reject pixels for which the second least significant bit of BEHRQualityFlags is non-zero. (One way to check this is to do a bitwise AND of the flag value with 2 and check if the result is > 0.) This will remove any pixels with a processing error or a NASA flag. BEHR contains three cloud fractions: OMI geometric, OMI radiance, and MODIS cloud fraction. We generally filter for OMI geometric or MODIS fraction to be < 0.2 (20%) when interested in total (to ground) columns. As discussed in Russell et al. 2011, MODIS cloud fraction is often less susceptible to identifying high albedo ground surfaces as clouds. Other applications (e.g. cloud slicing) may require different filtering.

A second flag advanced users should be aware of is bit #19, which indicates that the MODIS MCD43D BRDF data averaged for this pixel is of lower quality. This can happen if, for instance, the ground is snow covered. The first bit is set to true wherever this bit is true; the second bit is not. Users interested in winter retrievals over the northern US will likely need to allow this bit to be true in order to allow sufficient data through the filter. At this time, we do not have a quantitative estimate for how much the lower quality BRDF data increases the uncertainty of the retrieved VCD.

Users wishing extremely fine-grained control of which pixels are rejected have access to the VcdQualityFlags and XTrackQualityFlags fields as well as BEHRQualityFlags. Please refer to Table 2 and the NASA OMI NO_2 Readme file.

A previous version of this document indicated to removing negative VCDs and VCDs $> 1 \times 10^{17}$ molec. cm^{-2}; this is no longer recommended as removing negative VCDs can make the error in the stratospheric subtraction systematic instead of random (especially for small VCDs), and removing large VCDs should not be necessary as long as you check that XTrackQualityFlags = 0.

For non-NO_2 column density fields, users can be more judicious in the filtering of data. For example, if interested in the value of the MODIS fields, removing pixels for which BEHRQualityFlags indicates an error in the NO_2 algorithm will unnecessarily remove good

MODIS data, since MODIS data is not affected by the row anomaly nor by errors in the NO_2 algorithm. If you would like assistance figuring out the best filtering for a particular field, please contact one of the maintainers listed on the BEHR website.

4.1.2 Gridded data - CVM fields

Currently, all fields are gridded by the constant value method (CVM), also available in the `omi` package (https://github.com/gkuhl/omi). These data must be filtered the same as the native pixels (§4.1.1). These fields are indicated by the `grid_type` attribute having the value "constant value method".

4.1.3 Gridded data - flag fields

Flag fields (BEHRQualityFlags, VcdQualityFlags, XTrackQualityFlags) are put on a grid using the CVM; however, when multiple pixels overlap a single grid cell, their values are combined using a bitwise OR operation, rather than a weighted summation. This ensures that if any pixel contributing to a grid cell has a flag set, that flag will be set for the grid cell. This means that filtering CVM gridded fields by these flags will be conservative about including grid cells with good data. These fields are indicated by the `grid_type` attribute having the value "flag, bitwise OR".

4.1.4 Gridded data - other fields

Other fields that define the grid, such as Longitude and Latitude, are indicated by the `grid_type` attribute value "grid property". If the HDF publishing algorithm cannot identify a field's grid type, the `grid_type` attribute will have the value "undefined".

4.2 Weighting temporal averaging

Averaging over time is most easily accomplished using the gridded product, since the grid is consistent day-to-day. CVM gridded fields (§4.1.2) should be filtered for quality using BEHRQualityFlags or XTrackQualityFlags, VcdQualityFlags, and cloud fraction. The temporal average should weight each grid cell by the weight given in the Areaweight field for that swath.

4.3 To-ground vs. visible only columns

The field BEHRColumnAmountNO2Trop contains the to-ground column, which includes the estimated ghost column below the cloudy part of the pixel. The field BEHRColumnAmountNO2TropVisOnly does not include the ghost column; only the above-cloud NO_2 is included for the cloudy component of the pixel.

Most users should use the BEHRColumnAmountNO2Trop field. Users interested in cloud slicing approaches should use the BEHRColumnAmountNO2TropVisOnly field.

The difference in the VCDs stems from the difference in the computation of the cloudy AMF. The to-ground AMF (BEHRAMFTrop) is ultimately the ratio of a modeled slant

column density (accounting for the presence of clouds) to a modeled vertical column density that is integrated from ground to tropopause over the whole pixel:

$$A_{\text{to-ground}} = \frac{(1 - f_r) \int_{p_{\text{surf}}}^{p_{\text{tp}}} w_{\text{clear}}(p) g(p)\, dp + f_r \int_{p_{\text{cloud}}}^{p_{\text{tp}}} w_{\text{cloudy}}(p) g(p)\, dp}{\int_{p_{\text{surf}}}^{p_{\text{tp}}} g(p)\, dp} \tag{1}$$

where p_{surf} is the ground surface pressure, p_{cloud} is the cloud pressure, p_{tp} is the tropopause pressure, $w_{\text{clear}}(p)$ are the clear-sky scattering weights, w_{cloudy} are the cloudy-sky scattering weights, $g(p)$ is the *a priori* NO_2 profile, and f_r is the cloud radiance fraction.

The visible-only AMF, on the other hand, uses only the visible component of the modeled column in the denominator:

$$A_{\text{vis-only}} = \frac{(1 - f_r) \int_{p_{\text{surf}}}^{p_{\text{tp}}} w_{\text{clear}}(p) g(p)\, dp + f_r \int_{p_{\text{cloud}}}^{p_{\text{tp}}} w_{\text{cloudy}}(p) g(p)\, dp}{(1 - f_g) \int_{p_{\text{surf}}}^{p_{\text{tp}}} g(p)\, dp + f_g \int_{p_{\text{cloud}}}^{p_{\text{tp}}} g(p)\, dp} \tag{2}$$

where f_g is the geometric cloud fraction.

Therefore, dividing the satellite slant column by the to-ground AMF from Eq. (1) scales it by the ratio of a modeled slant column to a modeled total tropospheric column (including the ghost column). Dividing the same slant column by the visible only AMF from Eq. (2) scales it by the ratio of a modeled slant column to a modeled visible column (*excluding* the ghost column).

4.4 Using scattering weights/averaging kernels

4.4.1 Variable layout

These are for advanced users who might wish to either use their own *a priori* NO_2 profile or to compare modeled NO_2 VCDs correctly with BEHR VCDs. Starting with v3.0B separate clear and cloudy scattering weights are provided, which allows users to follow Eqs. (1) and (2) exactly, while a single combined averaging kernel vector is given for each pixel. The relationship between them is:

$$AK_i(p) = \frac{(1 - f) w_{i,\text{clear}}(p) + f w_{i,\text{cloudy}}(p)}{A_{i,\text{to-ground}}} \tag{3}$$

that is, the averaging kernel for pixel i at pressure p is the cloud radiance fraction weighted average of the clear and cloudy scattering weights for pixel i at pressure p, divided by the to-ground AMF for pixel i.

For new users wishing to begin using these variables, several resources will be helpful. Palmer et al. (2001) contains the original formulation of the relationship between scattering weights, *a priori* NO_2 profiles, and air mass factors and should definitely be studied. The first several chapters of *The Remote Sensing of Tropospheric Composition from Space* assembled by Burrows, Platt, and Borrell (Burrows et al., 2011) also describes the relationship of scattering weights, averaging kernels, and air mass factors.

BEHRScatteringWeights, BEHRAvgKernels, BEHRPressureLevels, and BEHRNO2apriori are 3D variables; the first dimension is the vertical dimension, the second and third correspond to the dimensions of the normal 2D variables. To put this another way, if X were

the array of values for these variables, then $X(:,1,1)$ would be the vector of values for the $(1,1)$ pixel, $X(:,1,2)$ the vector for the $(1,2)$ pixel and so on. (Note: some programs may read the data in with C-style ordering and so put the vertical dimension last.)

The BEHRPressureLevels variable gives the vertical coordinates for the other three. 30 of the pressure levels will be the same for every pixel; the remaining three will correspond to the terrain, cloud, and tropopause pressure. Should the terrain, cloud, or tropopause pressure match one of the standard 30 pressures, then the vector of values for this pixel will still be 33 elements long, but will end with fill values which should be removed. The scattering weights and averaging kernels can have additional fill values compared to the pressure levels; this occurs when the pressures are outside those available from WRF-Chem. Such fill values should always be above the tropopause, and because AMF calculations integrate from surface (or cloud) pressure to tropopause, can be safely removed.

4.4.2 Summary of use

Averaging kernels should be applied to model profiles before integrating the model profile to yield a tropospheric column for comparison with the BEHR VCD:

$$V_{\text{model}} = \sum_l AK_l g_l \tag{4}$$

where AK_l is the averaging kernel and g_l the model NO_2 partial column density for level l. The model profile should be interpolated to the pressure levels given in BEHRPressureLevels, rather than the other way around, since the averaging kernels can have sharp transitions at clouds. Since $AK_l = [(1-f)w_{i,\text{clear}}(p) + fw_{i,\text{cloudy}}(p)]/A$, where w_i is the scattering weight for level i and A the pixel's AMF, this calculation effectively converts the model column to a slant column first, then "retrieves" it using the BEHR AMF. **Note:** since BEHR provides a tropospheric NO_2 column density, it must be compared against a modeled tropospheric column only.

Similarly, scattering weights can be used with Eq. (1) and (2) to calculate custom AMFs with new *a priori* profiles. Unlike the averaging kernels, separate clear and cloudy scattering weights are stored in the .hdf files because in the BEHR AMF calculation, clear and cloudy scattering weights are used separately. Users interested in calculating their own AMFs with custom NO_2 profiles should download the following Matlab repositories:

- https://github.com/CohenBerkeleyLab/BEHR-core
- https://github.com/CohenBerkeleyLab/BEHR-core-utils
- https://github.com/CohenBerkeleyLab/Matlab-Gen-Utils

and follow the code in

https://github.com/CohenBerkeleyLab/BEHR-core/blob/develop/Production%20tests/test_published_scattering_weights.m

Before calculating custom AMFs, users should follow the code in the above link exactly to reproduce the BEHR AMFs from the published scattering weights and NO_2 profiles. The difference between user calculated and published AMFs should be $< 0.1\%$.

4.4.3 Calculation of averaging kernels

The averaging kernels presented (AK_i) are calculated as the weighted average of a vector of clear and cloudy scattering weights obtained by interpolating the NASA OMNO2 TOMRAD look up table to the appropriate values of SZA, VZA, RAA (relative azimuth angle), albedo, and surface pressure, divided by the AMF for that pixel. Mathematically:

$$AK_i = \frac{f_r w_{i,\text{cld}} + (1 - f_r) w_{i,\text{clr}}}{A} \tag{5}$$

$$w_{i,\text{clr}} = \begin{cases} 10^{-30} & \text{if } p_i > p_{\text{terr}} \\ w_{i,\text{clr}} & \text{otherwise} \end{cases} \tag{6}$$

$$w_{i,\text{cld}} = \begin{cases} 10^{-30} & \text{if } p_i > p_{\text{cld}} \\ w_{i,\text{cld}} & \text{otherwise} \end{cases} \tag{7}$$

where f_r is the radiance cloud fraction, $w_{i,\text{clr}}$ is the ith element in the clear sky scattering weight vector, $w_{i,\text{cld}}$ is the ith element in the cloudy scattering weight vector, A is the BEHR tropospheric AMF, p_i is the pressure level for the ith element in the vectors, p_{terr} is the terrain pressure for the pixel, and p_{cld} is the cloud top pressure for the pixel. Therefore the final vector of averaging kernels is the cloud radiance fraction-weighted average of the clear and cloudy scattering weight vectors after setting the clear sky vectors to essentially 0 below the ground, and the cloudy vectors to essentially 0 below the cloud top.

Two final notes: first, internally BEHR integrates the *a priori* profile in mixing ratio (or mixing ratio times scattering weight) over pressure, see Appendix B of Ziemka et al. (2001) for information on how mixing ratio integrated over pressure should be done to be equivalent to number density integrated over altitude. Second, since the clear and cloudy scattering weights are set to 0 below the surface and cloud pressures, respectively, before they are combined to yield the averaging kernel, so there will implicitly be no below-ground or below-cloud contribution to the respective subscenes.

References

Burrows, J., Platt, U., and Borrell, P.: The Remote Sensing of Tropospheric Composition from Space, Springer, 2011.

Laughner, J. L. and Cohen, R. C.: Quantification of the effect of modeled lightning NO_2 on UV-visible air mass factors, Atmos. Meas. Tech. Discuss., 2017, 1–24, doi:10.5194/amt-2017-263, URL https://www.atmos-meas-tech-discuss.net/amt-2017-263/, 2017.

Palmer, P., Jacob, D., Chance, K., Martin, R., Spurr, R., Kurosu, T., Bey, I., Yantosca, R., Fiore, A., and Li, Q.: Air mass factor formulation for spectroscopic measurments from satellites: Application to formaldehyde retrievals from the Global Ozone Monitoring Experiment, J. Geophys. Res., 106, 14 539–14 550, 2001.

Russell, A., Perring, A., Valin, L., Bucsela, E., Browne, E., Min, K., Wooldridge, P., and Cohen, R.: A high spatial resolution retrieval of NO_2 column densities from OMI: method and evalutation, Atmos. Chem. Phys., 11, 8543–8554, 2011.

Ziemka, J., Chandra, S., and Bhartia, P.: "Cloud slicing": A new technique to derive upper tropospheric ozone from satellite measurements, J. Geophys. Res. Atmos., 106, 9853–9867, 2001.