

---

# **pangaea Documentation**

***Release 0.0.4***

**Alan D. Snow**

**Nov 02, 2018**



---

## Contents:

---

<b>1</b>	<b>xarray extension</b>	<b>3</b>
<b>2</b>	<b>Read in LSM files</b>	<b>7</b>
<b>3</b>	<b>Logging</b>	<b>9</b>
<b>4</b>	<b>Indices and tables</b>	<b>11</b>



An xarray extension for gridded land surface & weather model output.

GitHub: <https://github.com/snowman2/pangaea>



**class** `pangaea.LSMGridReader` (*xarray\_obj*)

This is an extension for xarray specifically designed for land surface models.

Read with pangaea example:

```
import pangaea as pa

with pa.open_mfdataset('/path/to/ncfiles/*.nc',
                      lat_var='lat',
                      lon_var='lon',
                      time_var='time',
                      lat_dim='lat',
                      lon_dim='lon',
                      time_dim='time') as xds:
    print(xds.lsm.projection)
```

Read with xarray example:

```
import xarray as xr

with pa.open_dataset('/path/to/file.nc') as xds:
    print(xds.lsm.projection)
```

**affine**

The affine for the transformation.

**Type** `Affine()`

**center**

Return the geographic center point of this dataset.

**coords**

Returns y, x coordinate arrays

**Warning:** The grids always be returned with [0,0] as Northeast and [-1,-1] as Southwest.

**datetime**

Get datetime object for time variable

**dx**

Pixel size in x direction.

**Type** `float`

**dy**

Pixel size in y direction.

**Type** `float`

**epsg**

EPSG code

**Type** `str`

**geotransform**

The geotransform for grid.

**Type** `tuple`

**getvar** (*variable*, *yslice=slice(None, None, None)*, *xslice=slice(None, None, None)*, *calc\_4d\_method=None*, *calc\_4d\_dim=None*)

Get variable from model with subset options.

**Warning:** The grids will always be returned with [0,0] as Northeast and [-1,-1] as Southwest.

**Parameters**

- **variable** (`str`) – Name of variable in dataset.
- **yslice** (`slice`, optional) – Slice in y-direction of grid to extract data from.
- **xslice** (`slice`, optional) – Slice in x-direction of grid to extract data from.
- **calc\_4d\_method** (`str`) – Method to convert 4D variables to 3D variables (Ex. ‘mean’, ‘min’, or ‘max’).
- **calc\_4d\_dim** (`str`) – Dimension to reduce grid from 4D to 3D (Ex. ‘top\_bottom’).

**Returns**

**Return type** `xarray.DataArray()`

**latlon**

Returns lat,lon arrays

**Warning:** The grids always be returned with [0,0] as Northeast and [-1,-1] as Southwest.

**projection**

`osgeo.osr.SpatialReference()` The projection for the dataset.

**resample** (*variable*, *match\_grid*)

Resample data to grid.



**Parameters**

- **variable** (*str*) – Name of variable in dataset.
- **match\_grid** (*gdal.Dataset()* or *sloot.grid.GDALGrid()*) – Grid you want the data resampled to match resolution. You can also pass the path to the grid.

**to\_datetime** ()

Converts time to datetime.

**to\_projection** (*variable*, *projection*)

Convert Grid to New Projection.

**Parameters**

- **variable** (*str*) – Name of variable in dataset.
- **projection** (*osr.SpatialReference()*) – Projection to convert data to.

**Returns****Return type** *xarray.Dataset()***to\_tif** (*variable*, *time\_index*, *out\_path*)

Dump a variable at a time index to a geotiff.

**Parameters**

- **variable** (*str*) – Name of variable in dataset.
- **time\_index** (*int*) – 0-based time index,
- **out\_path** (*str*) – Path to output geotiff file,

**to\_utm** (*variable*)

Convert Grid to UTM projection at center of grid.

**Parameters** **variable** (*str*) – Name of variable in dataset.**Returns****Return type** *xarray.Dataset()***x\_size**

Number of columns in the dataset.

**Type** *int***y\_inverted**

Is the y-coord inverted

**y\_size**

Number of rows in the dataset.

**Type** *int*



---

### Read in LSM files

---

`pangaea.open_mfdataset` (*path\_to\_lsm\_files*, *lat\_var*, *lon\_var*, *time\_var*, *lat\_dim*, *lon\_dim*, *time\_dim*,  
*lon\_to\_180=False*, *coords\_projected=False*, *loader=None*, *engine=None*,  
*autoclose=True*)

Wrapper to open land surface model netcdf files using `xarray.open_mfdataset()`.

**Warning:** The time dimension and variable will both be renamed to ‘time’ to enable slicing.

#### Parameters

- **path\_to\_lsm\_files** (*str*) – Path to land surface model files with wildcard. (Ex. ‘/path/to/files/\*.nc’)
- **lat\_var** (*str*) – Latitude variable (Ex. lat).
- **lon\_var** (*str*) – Longitude variable (Ex. lon).
- **time\_var** (*str*) – Time variable (Ex. time).
- **lat\_dim** (*str*) – Latitude dimension (Ex. lat).
- **lon\_dim** (*str*) – Longitude dimension (Ex. lon).
- **time\_dim** (*str*) – Time dimension (ex. time).
- **lon\_to\_180** (*bool*, *optional*, *default=False*) – If True, will convert longitude from [0 to 360] to [-180 to 180].
- **coords\_projected** (*bool*, *optional*, *default=False*) – If True, it will assume the coordinates are already in the projected coordinate system.
- **loader** (*str*, *optional*, *default=None*) – If ‘hrrr’, it will load in the HRRR dataset.
- **engine** (*str*, *optional*) – See: `xarray.open_mfdataset()` documentation.
- **autoclose** (*str*, *optional*, *default=True*) – If True, will use autoclose option with `xarray.open_mfdataset()`.

### Returns

**Return type** `xarray.Dataset()`

Read with pangaea example:

```
import pangaea as pa

with pa.open_mfdataset('/path/to/ncfiles/*.nc',
                      lat_var='lat',
                      lon_var='lon',
                      time_var='time',
                      lat_dim='lat',
                      lon_dim='lon',
                      time_dim='time') as xds:
    print(xds.lsm.projection)
```

`pangaea.log_to_console` (*status=True, level=None*)

Log events to the console.

#### Parameters

- **status** (*bool, Optional, Default=True*) – whether logging to console should be turned on(True) or off(False)
- **level** (*string, Optional, Default=None*) – level of logging; whichever level is chosen all higher levels will be logged. See: <https://docs.python.org/2/library/logging.html#levels>

`pangaea.log_to_file` (*status=True, filename='/home/docs/.cache/pangaea/log/pangaea.log', level=None*)

Log events to a file.

#### Parameters

- **status** (*bool, Optional, Default=True*) – whether logging to file should be turned on(True) or off(False)
- **filename** (*string, Optional, Default=None*) – path of file to log to
- **level** (*string, Optional, Default=None*) – level of logging; whichever level is chosen all higher levels will be logged. See: <https://docs.python.org/2/library/logging.html#levels>



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





## A

affine (pangaea.LSMGridReader attribute), 3

## C

center (pangaea.LSMGridReader attribute), 3

coords (pangaea.LSMGridReader attribute), 3

## D

datetime (pangaea.LSMGridReader attribute), 4

dx (pangaea.LSMGridReader attribute), 4

dy (pangaea.LSMGridReader attribute), 4

## E

epsg (pangaea.LSMGridReader attribute), 4

## G

geotransform (pangaea.LSMGridReader attribute), 4

getvar() (pangaea.LSMGridReader method), 4

## L

latlon (pangaea.LSMGridReader attribute), 4

log\_to\_console() (in module pangaea), 9

log\_to\_file() (in module pangaea), 9

LSMGridReader (class in pangaea), 3

## O

open\_mfdataset() (in module pangaea), 7

## P

projection (pangaea.LSMGridReader attribute), 4

## R

resample() (pangaea.LSMGridReader method), 4

## T

to\_datetime() (pangaea.LSMGridReader method), 5

to\_projection() (pangaea.LSMGridReader method), 5

to\_tif() (pangaea.LSMGridReader method), 5

to\_utm() (pangaea.LSMGridReader method), 5

## X

x\_size (pangaea.LSMGridReader attribute), 5

## Y

y\_inverted (pangaea.LSMGridReader attribute), 5

y\_size (pangaea.LSMGridReader attribute), 5