Mapping Files to Model Variables

 Authors: Daniel Garijo, Deborah Khider, Yolanda Gil and Suzanne Pierce

[**General types of files**](#_a3argwrfosnx) **3**

[**Formats of files in detail**](#_wgv239buidf) **4**

[**List of Requirements and Competency Questions**](#_u5pz51imxcev) **5**

[Data Catalog](#_k13p0r6h2god) 5

[Model Catalog](#_ntin1yatigr8) 6

**Goal of this document:** Computational models from different domains such as hydrology, geology, agronomy, climate or economics use different files and formats to represent their input data. These inputs are key to produce the resultant predictions of each model, and are derived from heterogeneous data sources. This document aims to understand the transformation processes that lead to the creation of input files for models, as well as their relationship with the variables that the model represents and requires. This document also lists examples and common data format descriptions that are used to represent model variables.

**Rationale:** Modelers from different domains usually have to transform source data to extract information about the variables that are input to the target model. If we aim to make this process automatic or semi-automatic, we need to understand what are the steps required for such transformations, as well as the requirements needed by them.

## Scenarios

### **Example 1**

How can precipitation, as obtained from Nexrad data (<https://www.ncdc.noaa.gov/data-access/radar-data/nexrad>), be used to create the recharge file that MODFLOW groundwater model uses to produce its predictions for the Barton Springs in Austin? (it’s one of its many inputs). A workflow like the following one may be followed:

* First, you need to get as much additional data as you can: **stream data from** USGS, **anthropogenic data** from Austin water utility and **land use** from the city of Austin. These are additional data variables, alternative to precipitation.
* Then from satellite data you may estimate **pipeline density** (i.e., the amount of pipes pumping water from the area).
* Before you create your recharge file you must create a function, which is an interpretation of the situation based on your domain knowledge. For example, if only precipitation data is available (as opposed as to also having stream data and land use), a reasonable interpretation might be that only 25% of the precipitation infiltrates. This gives you as an output a recharge estimate.
* With the obtained recharge estimate you modify your precipitation file (multiply the estimate on each cell).
* The next step is to follow with an intersection of the grids: the grid from your target model and the grid from the precipitation file.
* Finally, you reformat the final file according to your target model. In this case, MODFLOW. There are software packages designed for this task. In this case, Flopy (https://water.usgs.gov/ogw/flopy/).

Although this is a simple example, one can realize that significant domain expertise is needed to run every model accurately:

1. Interpretations: required and selected according to the input data available. Wrong interpretations will lead to wrong predictions in the model.
2. Minimum data and missing data: The more available data, the more accurate model predictions are. Some of the inputs are “nice to have” to complement the required variables, which in this case is precipitation. If the minimum variable data is unavailable, scientists must estimate the data values. A common example is to interpolate data from belonging to close regions. In the example, data from the nearest airports could be used to get an estimate on precipitation if precipitation data was not available.

~~Also, this example highlights some initial requirements needed to describe model inputs in terms of format, location and time:~~

1. **~~R1:~~** ~~Input data must follow a certain file format accepted by the target model: if the file format is not according to what MODFLOW expects, the model won’t run.~~
2. **~~R2:~~** ~~Input data must follow a certain grid, or be mapped to the region expected by the model (and the rest of the inputs of the model).~~
3. **~~R3:~~** ~~Input data must conform to a certain time interval.~~

### Example 2

How can we calculate relative humidity for Topoflow using the information from one of the scenarios from [ECMWF](https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era-20c)? In this scenario, the input files are NetCDF (or GRIB) files modeled according to the Climate and Forecast (CF) standard. A typical set of actions is as follows:

1. Reconstruct a GRIB file from the time frame we are interested in: GRIB files contain multiple variables in a single time stamp. Therefore if we aim to retrieve a time series of a given variable, we have to compile GRIB files and reconstruct the time series.
2. Select the time interval needed for the model and adjust the data accordingly
3. Select the grid size of the target area.
4. Transform the GRIB inputs into the variables used by the model. Note that one or more variables may be calculated using a function. For example, relative humidity is calculated from temperature and precipitation:



The first column represents how the data from ECMWF is calculated from two other variables, which corresponds to topoflow (second column) and GSN (third column).

~~This example presents the following additional requirements:~~

1. **~~R4:~~** ~~Need to represent whether a file is contains one or multiple variables.~~
2. **~~R5:~~** ~~Units expected by the model versus the ones provided by the input source must match.~~
3. **~~R6~~**~~: Need to represent that an additional function is required to calculate data. If this function is simple, we may consider it a data transformation (e.g., calculating an average). If the function is a complex equation, we may consider it an additional component. In this case, we would have a component that takes as input 2 meter temperature and 2 meter dewpoint temperature and produces relative humidity.~~

The rest of variable mappings to GSN can be found in the following document: [https://docs.google.com/document/d/1aaMViWvvv83K1nHFPPOIdt\_ZNd2RXEBHh9IRYZSkNA4/edit#](https://docs.google.com/document/d/1aaMViWvvv83K1nHFPPOIdt_ZNd2RXEBHh9IRYZSkNA4/edit)

## General types of files

In climate and hydrology, the following seem to be the most common types of files. Note that the models tend to use as inputs a cleaned up version of these commonly used files.

1. Observation data: Observation data may come from **satellite observations**, or **gauge data.** Satellite observations tend to be more structured, and collected according to agreed standards, while gauge data is not necessarily standardized nor consistent, but it’s more representative of fine-scale variation. For example, on the one hand gauge data may bring a CSV file with a few data points in a very specific area, with different time series depending on what was recorded at the time. On the other hand, remote sensing products in NetCDF (GRIB) may cover the target grid consistently, but at less resolution.

 **Formats used:** CSV, textNetCDF and GRIB

1. Reanalysis files/Data assimilation:
	1. Reanalyses: Through a variety of methods, observations from various instruments are added together onto a regularly spaced grid of data. Placing all instrument observations onto a regularly spaced grid makes comparing the actual observations with other gridded datasets easier. In addition to putting observations onto a grid, reanalysis also holds the gridding model constant - it doesn’t change the programming - keeping the historical record uninfluenced by artificial factors. Reanalysis helps ensure a level playing field for all instruments throughout the historical record. A climate reanalysis gives a numerical description of the recent climate, produced by combining models with observations. It contains estimates of atmospheric parameters such as air temperature, pressure and wind at different altitudes, and surface parameters such as rainfall, soil moisture content, and sea surface temperature. The estimates are produced for all locations on Earth, and they span a long time period that can extend back by decades or more. Climate reanalysis generates large datasets that can take up many terabytes of space. They use standard conventions.

**Formats used:** NetCDF and GRIB

b) Data Assimilation: The purpose of data assimilation is to determine a best possible atmospheric state using observations and short range forecasts. Data assimilation is typically a sequential time-stepping procedure, in which a previous model forecast is compared with newly received observations, the model state is then updated to reflect the observations, a new forecast is initiated, and so on. The update step in this process is usually referred to as the *analysis*; the short model forecast used to produce the analysis is called the *background*.
When data assimilation is used to monitor climate change based on past observations, it is called reanalysis. The atmosphere is chaotic, meaning that even small differences in its state can lead to very different weather patterns occurring several days later – this is sometimes referred to as *the butterfly effect*. To account for the chaotic nature of the atmosphere and the associated uncertainty in prediction, ensembles are run.

1. Global Climate Model (GCM) projections. For example, the IPCC uses projections of 21st century climate. They have agreed upon scenarios, which model greenhouse gases concentrations according to possible changes in anthropogenic emissions, which are consistent with certain socio-economic assumptions. Founded under the World Climate Research Programme, CMIP provides a community-based infrastructure in support of climate model diagnosis, validation, intercomparison, documentation, and data access. The IPCC Fifth Assessment Report, published in 2014, was supported by CMIP5. The next AR (AR6) is underway and supported by CMIP6. Some model outputs for CMIP6 are already available. GCM outputs also follow the CF convention but tend to be lower resolution (both spatially and temporally).

**Formats used**: NetCDF

1. Maps: soil data and land use. This data may come in a table format (CSV, txt) or as JPEG files from which we have to deduct the grid, latitude and longitude of the represented map. Figure 5 of <http://onlinelibrary.wiley.com/doi/10.1002/2016EA000237/full> shows an example.

 **Formats used:** ShapeFiles, Pixel files JPEG (or similar)

## Formats of files in detail

NetCDF files (<https://www.unidata.ucar.edu/software/netcdf/docs/>): Format for sharing array-oriented data. NetCDF files may be organized in two different ways:

1. Time series: a NetCDF file contains a single variable on a grid across multiple periods of time.
2. Time point: a NetCDF file contains multiple variables on a grid, across a single period of time. (most likely)

The advantage of the former is that if you just want to query a variable, you can do so from a single netCDF file. For the latter approach you need to reconstruct the time series from multiple netCDF files.

Each NetCDF file contains at least a **variable of interest**, **dimensions (lat, long, time, and if needed altitude)** and **attributes** (both global and variable level).

Example (from <https://github.com/khider/netCDFTutorial/blob/master/Opening%20and%20reading%20NetCDF%20and%20GRIB%20file.ipynb>):

Local name of variable: tp
 dimensions: ('time', 'latitude', 'longitude')
 size: 618240
 type: dtype('int16')
 scale\_factor: 6.624043998318357e-07
 add\_offset: 0.021704342564889928
 \_FillValue: -32767
 missing\_value: -32767
 units: 'm'
 long\_name: 'Total precipitation' → Link to GSN!

GRIB files (<https://en.wikipedia.org/wiki/GRIB>): are a type of flexible NetCDF files that tend to have more metadata. Therefore, we can transform a NetCDF file to GRIB, but if we transform GRIB to NetCDF we will lose metadata. So far, each GRIB file we have encountered contains only a month worth of data (time point approach). See Section 4.2 of <https://github.com/khider/netCDFTutorial/blob/master/Opening%20and%20reading%20NetCDF%20and%20GRIB%20file.ipynb> to find an example on how to access a GRIB file

All NetCDF (GRIB) are provided in the “.nc” (“.grib”) format. Some versions of NetCDF are based on HD5. NetCDF files are also partly annotated to GSN, which should facilitate the process of figuring out the appropriate variable that the input file aims to represent.

Example of a NetCDF and GRIB file, how to open them and query for variables.

<https://github.com/khider/netCDFTutorial/blob/master/Opening%20and%20reading%20NetCDF%20and%20GRIB%20file.ipynb>

Shape files (<https://en.wikipedia.org/wiki/Shapefile>): are commonly used to describe map information in different layers.

## List of Requirements and Competency Questions

After understanding both examples, talking to domain scientists and also looking at the metadata available in NetCDF and GRIB files, the data catalog and model catalog need to address the following requirements:

### Data Catalog

Given an input file:

1. What standard does the input file follow?
	1. Example: NetCDF, GRIB.
2. What is the version/convention of the standard adopted in this input file?
	1. Example: CF-1.6 from NetCDF.
	2. Rationale: different countries have different conventions to represent their topographic data
3. What variables are represented in an input file? (Assumption: from GSN)
	1. Example: land\_surface\_\_temperature
4. What are the values for a given variable in a given input file?
	1. Example: Values for land surface temperature as NetCDF file. The type of result will be requested. Could be CSV, NetCDF or GRIB among others.
5. What are the units for a given variable in a given input file?
	1. Example: land\_surface\_\_temperature measures temperature in Celsius degrees
6. What are the time intervals used in an input file?
	1. Example: days, months, years.
7. Is the input file a time series, or a time point?
	1. Example: point
8. What are the values for a given variable? (output should be specified)
	1. Example: reconstructed GRIB file for temperature in Example 2.
9. What are the time units used to represent the intervals at which a variable is captured?
	1. Example: Time is represented according to a gregorian calendar.
10. What is the region of the grid represented in this file?
	1. Example: The region is South Sudan
11. Is the grid from these two variables consistent? What are the units of grids?
	1. Example: My first variable is temperature, and my second variable is precipitation.
12. What are the values for a variable with an overlap in a given grid region?
	1. Example: retrieve precipitation data from the closest airports to my target region in South Sudan
13. Is the variable a category, or a number? If a category, what are the target values?
	1. Example: What are the types of soil in a given location?
14. If the values of a variable have missing values, how are they represented?
	1. Example: Total precipitation in <https://github.com/khider/netCDFTutorial/blob/master/Opening%20and%20reading%20NetCDF%20and%20GRIB%20file.ipynb> represents missing values with -32767
15. Is it possible to convert a variable in units A to the same variable in units B?
	1. Example: Can the sea surface temperature in Celsius be converted to Fahrenheit?
16. What is the grid size used in a file? What is the scale factor used in a file?
	1. Rationale: needed to re-pack NetCDF files.
17. What is the offset of the variable in the file?
	1. Rationale: Needed for proper alignment of variables and data transformations.
18. Is a given input file produced as a reanalysis file? If so, who has produced the analysis? How was the file produced?
	1. Example: US reanalysis is not very good at the moment, and ECMWF (european) is one of the best products at the moment.
	2. Rationale: Users need to know which types of data are they dealing with. If it’s a data poor region, you need to know whether the the data is from reanalysis or not.
19. Is a given input file an assimilation file? Is it a projection?

###

### Model Catalog

Most of the requirements for the model catalog have a certain similarity with the data catalog. In order to avoid repeating the same examples, I will omit some of them

1. Is there a model or function that produces a target variable?
	1. Example: give me a function that calculates relative humidity.
2. Is there a function or model that produces a target parameter?
	1. Example: To run SIMMeteo, I need an estimate of the maximum humidity from last year.
3. What are the requirements of the model variables in terms of:
	1. Unit convention (i.e., model expects data to be in this unit convention)
	2. Time intervals (model expects data to be daily, monthly, etc.)
	3. Time representation (model expects data according to the gregorian calendar)
	4. Standards (version of NetCDF that input “temp” of the model uses)
	5. Time extent (the amount of time needed to produce a simulation)
	6. Grid/resolution of the model (this might be difficult to determine, need real hydrology expertise)
	7. Assumptions
		1. Example: What is the solver used for this model? Is it compatible with the analysis that I am performing later, using another model?
4. Which input to the model maps to which variable(s) in GSN?
	1. Example: In Topoflow, Surface Temperature maps to land\_surface\_\_temperature.
5. What is the variable associated to the output type of the model?

The model catalog and the data catalog should be used alternatively to create workflows: with the model catalog you first retrieve all the requirements you have for all inputs of your workflow, including those functions needed to do non trivial conversions between variables. Then, with the data catalog you get the transformations needed to convert the available sources into workflow inputs.

## ~~Under discussion/ Brainstorm~~

~~This section~~ **~~reflects Daniel’s thoughts at the moment. They are a draft.~~**

~~It seems like there is ambiguity when we refer to variables. We can distinguish different types of variables:~~

1. ~~Global Standard Variables, which are the concepts as described in GSN. They don’t have associated units or grids, resolution, etc., but they unambiguously refer to the variable concept.~~
2. ~~Model variables: They describes variables with localized information requirements that may depend on other factors. For example, the variable units might need to be in longitude/time.~~
3. ~~Data variables: The variables that are in datasets/files in the data catalog. These seem to be concrete variables in the sense that they refer to a variable concept but also to the concrete units and time. Note that some of the metadata refers to the file (e.g, time interval), and some to the variables~~

~~Then we have inputs and outputs types. An input type may have associated a variable, spatial and temporal extent, etc. Input types seem to be broader than just variables, the variable is just one of the things they cover. At the same time, model variables seem to be more concrete than global standard variables. And what we want to do is transform a given dataset (which may have several instances of dataset variables) into an input type for a model.~~

~~Basically: InputType --hasVariable-->ModelVariable. ModelVariable have units and correspond to a GSN variable. The input may have other requs such as format, time, etc (see list above)~~

~~So what are the inputs variables of a model? We would have to query for the variables of an input, and their link to GSN.~~

~~Difference between dataset variables and model variables? It could be that model variables are more flexible. For example, the units are generic. In the ontology world, model variables would be types (classes) while the datasets would be instances of those classes. But we need to define concrete axiomatization (e.g., type1 only(hasUnits longByTime)). This is a little too complex in my opinion.~~

~~An alternative would be to add all this info at the component level, using rules. For example, component has Input1, and Input1 hasUnits ?unit, and ?unit corresponds to a derivation to long by time → rework this with example.~~

~~Example: TO DO.~~

~~This at the instance level (dataset), input level (wings data catalog), component (rules). How would they be connected?~~