CFU_load

Description

This function loads experimental and corresponding observational data from the repository and stores them into two similar multi-dimensional matrices that will be used later to compute diagnostics.

This data structure was agreed at a previous stage to facilitate the computation of the diagnostics.

Usage

The user must specify at least the name of the variable to load, a list of experimental dataset names, a list of observational dataset names and a list of starting dates.

The first experiment in the list of experimental datasets must be the one with the greatest number of members and, if possible, of lead-times. Very often it is not possible and a parameter with the greatest number of lead-times across the various experiments must be provided.

The dimensions of the first experiment will determine the dimensions of the two data structures that CFU_load will fill and output. Below it is explained how to sort the experiments properly.

If various experiments are specified, it is very possible that they are stored with different gridding. Then a common gridding must be specified and all the data will be adapted to it.

To check how the parameters should be introduced and what ranges of values are accepted, use info_cd('CFU_load') after loading the common_diagnostics source file.

Details

First, the two data structures are created. These are two multi-dimensional arrays (matrices).

The experimental data matrix will contain the values of the specified variable for all the experiments in the specified starting dates. The observational data matrix will contain the observed values of the same variable gathered from the specified observational datasets scanning the data repository. The observational data is chosen so as to date-correspond the experimental data.

To create these matrices, we take as reference the dimensions of the first experimental dataset.

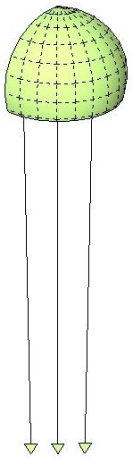
An experiment usually consists of multi-member predictions initialized at several starting dates. From each starting date, a sequence of lead-times is stored when the model runs for a subset of the variables describing the climate state. The lead-times have a given constant frequency which can be daily or monthly and the variables can be either two-dimensional or

area averages. While the model runs, values are generated and stored into data files that will be part of an experimental dataset.

The data of each starting date is stored in a different file. To compute diagnostics after CFU_load we only need to load values of a single specific variable and in a specified zone but from all the chosen starting dates.

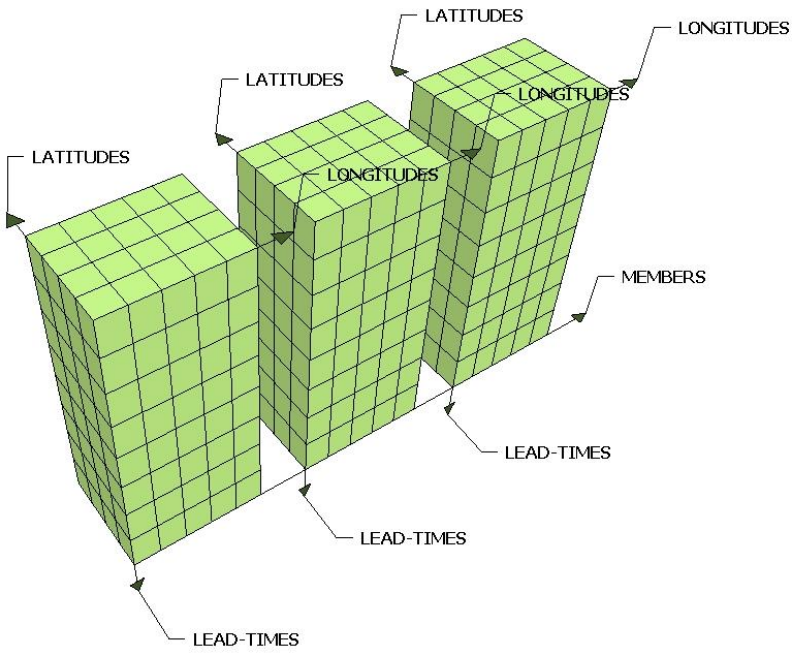
When we load the data from a single experimental dataset file, we obtain a 4-dimensional array with the dimensions in the following order: longitudes, latitudes, members and lead-times **[1]**. We consider a 2-dimensional variable because it is the most complete example.

In the picture at the right we can see the representation of a variable from an experimental dataset at a single given starting date. The slice of earth is the area we want to consider for the analysis. This data is stored in the same experimental data file, that stores a value corresponding to every cell of the longitude by latitude grid that delimits the area. The time line goes downwards in the figure and the arrows that come out from the slice symbolize the members, each as long as the simulation time (i.e. the number of lead-times times the step).



Single variable values in a delimited zone from an experimental dataset at a given starting date.

In the picture below, a representation of the corresponding 4-dimensional array we load with labeled axes that represent the dimensions. The experiment in this example was delimited to a zone with a few more longitudes than latitudes, several lead-times and three members.



Representation of the 4-dimensional array we load from an experimental dataset file.

[1] In 'DePreSys' experiments not only the initial conditions change from one member to another, but also the adjustments of the model parameters. This is why in a 'DePreSys' experimental dataset there is a separate file for each of its 9 members. When we load one of its files, we obtain a 3-dimensional array because the members dimension is missing. We add a dimension of length 1 at the third position after loading it to have the same structure in all the kinds of experiments.

The first file we load is the one of the first starting date of the first experimental dataset specified in the list. Once we have its 4-dimensional matrix with the variable data, we are able to work out the number of longitudes, latitudes, members and lead-times of the first experiment by simply measuring its dimensions.

Known the dimensions of the first experimental dataset we build the two output data matrices.

The experimental data matrix is built with the dimensions in the following order and with the following lengths:
1- The number of experimental datasets determined by the user (we need to store data of all the experiments in the same array).
2- The number of members of the first experiment.
3- The number of starting dates determined by the user (we need to store data of each prediction of the model in each starting date).
4- The greatest number of lead-times **[2]**.
5- The number of latitudes of the zone we want to consider.
6- The number of longitudes of the zone we want to consider.

Dimensions 5 and 6 will depend on whether the variable loaded is 2-dimensional or an area average. In the case of an area average the dimensions of the matrix will be only the first 4. Furthermore, if we are loading a 2-dimensional variable, we can set CFU_load to load it as area averages, as longitudinal averages in function of latitudes or as in latitudinal averages in function of longitudes. In these cases the 5^{th} and/or 6^{th} dimension will also disappear. We can constrain CFU_load to fetch only values that are inside a zone delimited by minimum and maximum longitudes and minimum and maximum latitudes **[3]**. Otherwise, the zone the experiment was run over will be taken as default.

As other optional parameters, we can disable certain values by specifying masks or change the gridding and the interpolation method used for this finality. The default grid is the original model grid (only when all the models are in the same grid! If not, specify a common gridding). We will keep on with 2-dimensional examples without constraints.

The observational data matrix is built very similarly to the experimental matrix. The only differences are that the first dimension turns to be the number of observational datasets and the length of the 2^{nd} dimension turns to be the number of members of the observational datasets **[4]**. Later, for each lead-time we will store observational data into it if possible.

[2] Either the number of lead-times of the first experiment or the greatest number of lead-times specified by the user in the cases that another experiment has a greater number of lead-times. A sub-sampling range of time and can be specified to load data only between two lead-times and also a sub-sampling period to load only a subset of lead-times. Check '-nleadtime', '-leadtimemin', '-leadtimemax' and '-sampleperiod' in info_cd('CFU_load').

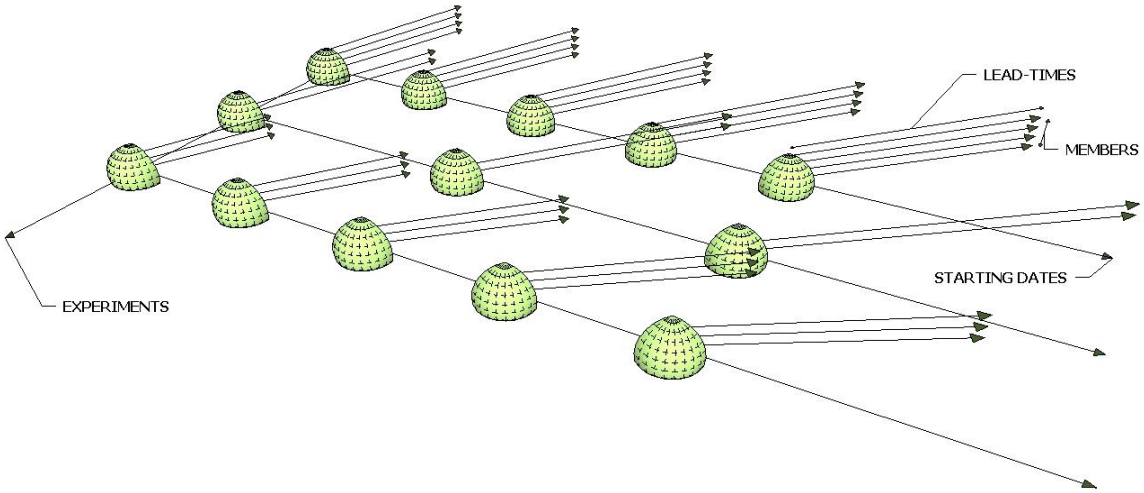
[3] Check parameters '-output', '-lonmin', '-lonmax', '-latmin', '-latmax' in info_cd('CFU_load') for more information.

[4] Currently (march 2013) the multi-member observational dataset load is not a fully developed feature. When the user chooses 2-dimensional oceanic variables, the members dimension in the observational matrix is set to 5. The only observational dataset with more than one member existing by now is ORAS4 ocean reanalysis: it has 5 members. In any other case the length of the 2^{nd} dimension is set to 1.

At this point, we start to fill in the experimental data matrix with the data in the repository by looping on the experiments.

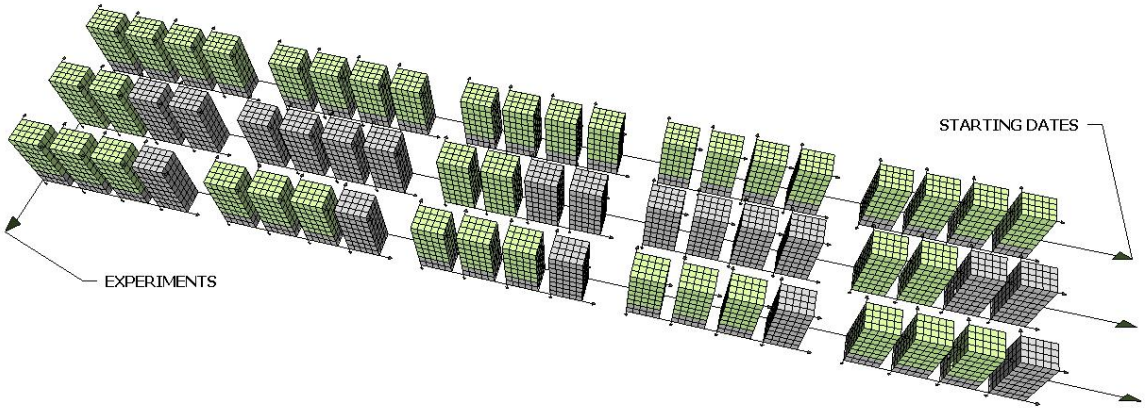
If we looped on an experiment with more members or lead-times than the first experiment, its data would not fit the structure of the matrix and would not be loaded. This is why the first experiment should be the one with the greatest number of members and, if possible, lead-times. If not possible, the greatest number of lead-times must be specified.

Besides, if we loop on an experiment that has less members or lead-times, we will not be able to load data for all the cells in the array. These will remain as NA values. The same will happen when an experiment was not run for any of the starting dates specified by the user. This can be appreciated clearly in the following example, where the starting dates specified are 5, the experimental datasets are 3 (with 4, 2 and 3 members respectively) and the second one was not run in the 2nd and 4th specified starting dates but its number of lead-times is greater than the others'. Recall we always load a single variable.



Representation of an example of use of CFU_load where the starting dates specified by the user are 5, the experimental datasets are 3 (with 4, 2 and 3 members from back to front) and the second was not run in the 2nd and 4th specified starting dates but its number of lead-times is greater than the others'.

The array that corresponds to this example is depicted in the picture below. Lead-times go downwards and NA values are colored in gray.



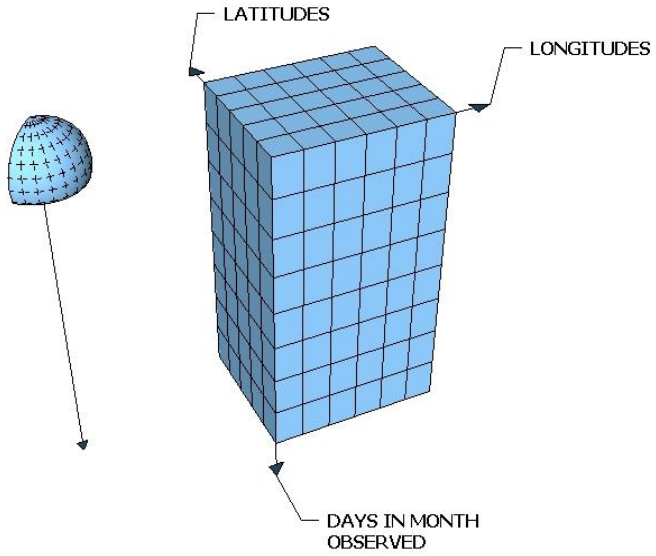
Example of an experimental data array loaded with 5 starting dates and 3 experiments (with 4, 2 and 3 members from back to front). The 2nd experiment has a greater number of lead-times than the others and has no data for the 2nd and 4th starting dates. NA values are colored in gray.

In this example we also see how the number of experiments, the number of starting dates, the greatest number of members and lead-times and the longitudes and latitudes determine the size of the experimental data matrix, regardless of the difference in number of members or lead-times among experiments or missing data for a starting date.

After the experimental data is loaded, CFU_load tries to find the observational data that date corresponds to the experimental data and stores it into the observational data matrix. For a given starting date, data is kept in the array only once per observational dataset instead of once per experimental dataset, that is, we do not store an observational value corresponding to each experimental value; we store the observational values that were observed on any date that matches any of the lead-times of any experiment. Thus a single value may match more than one experimental value.

Whenever we have no observation at any date, which is very common, the observational matrix is filled with NA.

Each observational data file contains observed data of a month and usually with only one member. In most observational data files there is only one day in month observed because most of observational datasets are monthly and record one observation per month. The others are daily.

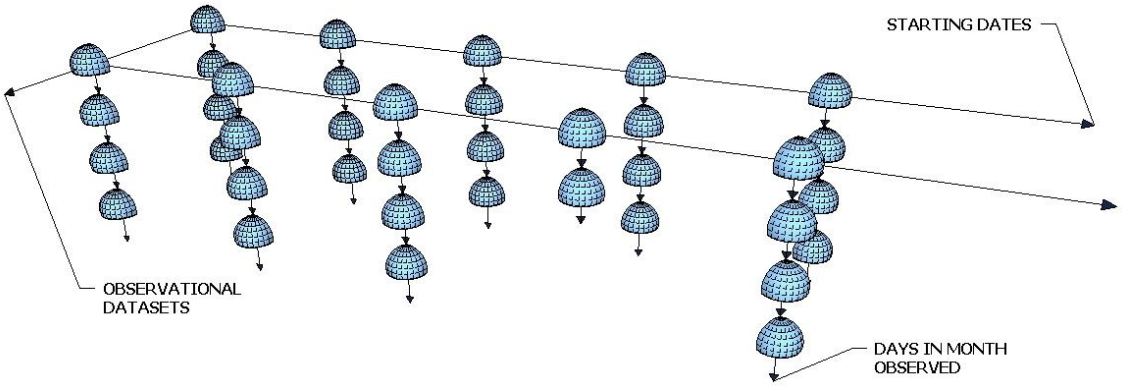


Representation, at the left, of a variable from an observational dataset at a given month and the 3-dimensional array we obtain when we load it. The observational dataset is daily because it contains more than one observation per month.

In the picture we can see the representation of a variable from an observational file and the 3-dimensional array we obtain when we load it. The time line goes downwards.

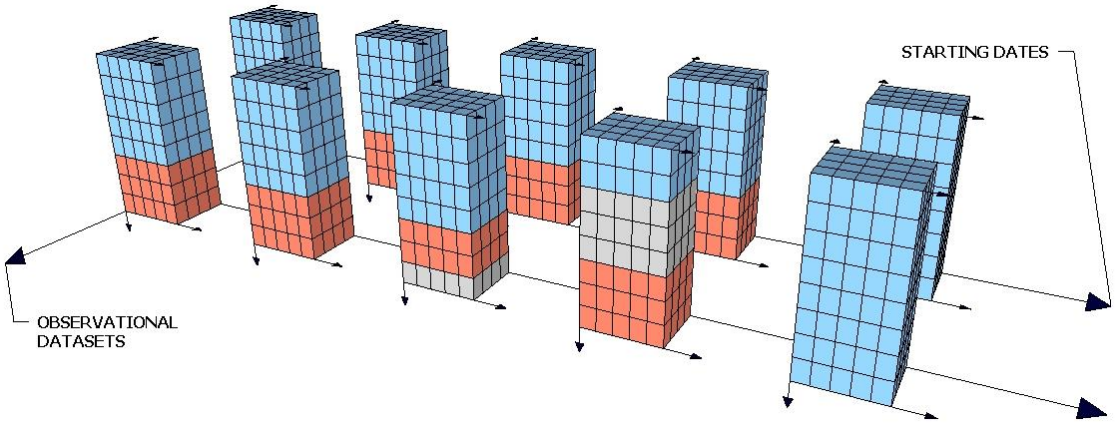
In contrast to an experimental dataset, an observational dataset does not contain multiple data for the same date: an experiment consists of predictions starting at several starting dates. These predictions usually last longer than the time between two starting dates and hence we store two different predictions for some dates in one experimental dataset. This does not happen in observational datasets, where the data is recorded month to month (or day to day). However, the observational matrix has a similar structure to the experimental matrix and, for each lead-time (from 1 to the greatest number of lead-times) in every starting date, CFU_load tries to fetch an observational value that matches. This is why in the observational matrix we will have repeated values in the ranges of time that overlap the next starting date.

In the next picture we can see the representation of the observational data involved in the experimental data example in the previous page. To preserve the same axes, the time is measured in starting dates. Then, whereas in experimental dataset there was one file each starting date, in observational datasets there are several files between two starting dates because one file stores data from a single month.



Representation of the observational data involved in a call to CFU_load. The user specified a 2-dimensional variable, 2 observational datasets (with 1 member each) and 5 starting dates. Every month of data is stored in a different file. The files that match the starting dates are depicted wider. The 2nd observational dataset has no data registered for the second half of the 4th simulation time.

The next picture represents the observational data array that will result in the case of the example. NA values are colored in gray and overlapping values that are repeated to the first values of the next starting date are colored in red.



5-dimensional array that contains the observational data from 2 observational datasets and 5 starting dates specified by the user. The 2nd observational dataset has no recorded values for the second half of the 4th simulation time. NA values are colored in gray and overlapping values between consecutive starting dates are colored in red.