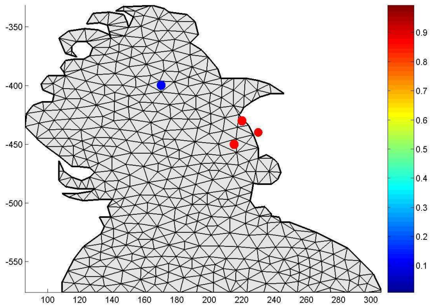
Diva User's Guide

Four Dimensional Climatology analysis tool

(last modified March 15, 2011)



J.-M. Beckers[1] (JM.Beckers@ulg.ac.be),
M. Ouberdous[2] (M.Ouberdous@ulg.ac.be),
C. Troupin[3] (ctroupin@ulg.ac.be).

Université de Liège

GHER GeoHydrodynamics and Environment Research

GeoHydrodynamics and Environment Research, MARE,
University of Liège, Sart-Tilman B5, 4000 Liège, BELGIUM
http://modb.oce.ulg.ac.be

[1] Honorary Research Associate, National Fund for Scientific Research, BELGIUM
[2] Research Associate, GHER, MARE, University of Liege, BELGIUM
[3] PhD Student, Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture (F.R.I.A), BELGIUM

Acknowledgments

This report was written for helping oceanographers to work with the Diva *software. This would not have been possible without the help of scientists involved in Data Analysis projects.*

We would like to thank:

the participants to the Diva *workshops in Liège (November 2006) and Calvi (September 2007) for their valuable comments to improve the software;*

the GHER *team for its constant support;*

the National Fund for Scientific Research (Belgium) for the funding of a supercomputer,

the Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture the European Union (FRIA) for funding Damien and Charles PhD grants.

Conditions of use

Diva is a public domain graphical software and can be found on the GHER 'www' server. This program is subject to conditions of use that can be found on the 'www' server.
Diva is a Liege University software, which will be further developed for SeaDataNet scientific data products in JRA4 activities.
All SeaDataNet products (data and softwares) are freely distributed to the scientific community at the following conditions:

- ☑ The products should be used for scientific purposes only.
- ☑ Articles, papers, or written scientific works of any form, based in whole or in part on data or software supplied by SeaDataNet, will contain a suitable acknowledgment to the SeaDataNet program of the European Union. Related publications (see bibliography) should also be cited.
- ☑ The applications of SeaDataNet products are on the full responsibility of the users; neither the Commission of the European Communities nor the SeaDataNet partners shall be held responsible for any consequence resulting from the use of SeaDataNet products.
- ☑ The recipient of these data will accept responsibility of informing all data users of these conditions.
- ☑ The Diva software cannot be incorporated into another software package without permission.

Contents

III Godiva **5**

1 Diva 4D **6**

1.1 Introduction 6
1.2 Climatology production 6
1.2.1 Climatology definition 7
1.2.2 Actions to be performed 7
1.3 `Climatology` directory: input files 9
1.4 Domain definition: Praparing the topo.gebco 12
1.5 Climatologies production: Actions control 13
1.5.1 `driver` file: actions and flag values 13
1.6 "Advection" constraint and Reference fields 17
1.6.1 "Advection" constraint fields generation 17
1.6.2 Reference fields generation 17
1.6.3 Advection constraint and Reference fields usage 18
1.7 `newinput` and `output` 19

2 Diva 4D Installation and Use **20**

2.1 Installation 20

2.2 Using GODIVA: an example 21
2.2.1 Domain definition and coast line contour files generation 22
2.2.2 Data extraction 27
2.2.3 Variable analysis 30

Part III

Godiva

Chapter 1

Diva 4D

This document is the third part of Diva documentation. The first part is **DivaUserGuide** which describes the diva theory and use of diva 2D-scripts and related tools. The **Diva3DUserGuide** describes Diva 3D-tools and scripts to generate analysis of a given variable data on an oceanic basin and a time interval. This document describes Diva tools to generate climatologies for several variables on different time periods, starting from data extraction to 4D-Netcdf analysis files for each variable.

1.1 Introduction

To produce Diva4D analysis, the main script to be run is the `divadoall` (located in directory `GODIVA/JRA/Climatology/`. The actions to be performed are defined by the flag values read in the `driver` file. The `divadoall` program allows you to extract data from an ODV generic spreadsheet file (to write it in Diva input data files format), and to perform analysis of the extracted data. The data extraction and analyses can be done for a set of variables on different (chosen) time periods. Variables to be treated are defined in an input file `varlist`, and time periods in `yearlist` and `monthlist` files.

1.2 Climatology production

Produced climatologies are the analysis performed by Diva of a given variable observation (measures) data sets. Diva is adapted to produce climatologies using ODV4 spreadsheets. Climatology variables and time periods definition is made by providing necessary information in three input files: `varlist`, `yearlist` and `monthlist`. All the necessary actions and steps to be performed to produce a climatology are defined through flag values in the `driver` file.

1.2.1 Climatology definition

To define the climatology to be produced one has to provide the following files:

- `varlist`: contains on each line the (short) name `VAR` of the variable to be treated (this is typically the header name from the ODV export).
- `yearlist`: contains on each line the period to be covered. Ex `19502000` from 1950 to 2000 included.
- `monthlist`: contains on each line the range of months covered by the climatology. Ex 0103 means January, February and March together. Values as 1202 are allowed and cover December, January and February. For a climatology for each month, file `monthlist` therefore will contain 12 lines from 0101 to 1212.

```
Temperature
Salinity
```

Example file 1.2.1: `varlist` *file example.*

```
19501960
19701980
19902000
```

Example file 1.2.2: `yearlist` *file example.*

```
0103
0406
0709
1012
```

Example file 1.2.3: `monthlist` *file example.*

1.2.2 Actions to be performed

All actions performed by `divadoall` are prescribed in the input file `driver`. These actions are:

- Data extraction.
- Boundary lines and coastlines generation (contour files).
- "Advection" field (based on coast lines) generation.
- Data cleaning on mesh.
- Outliers elimination from data sets.
- Generation of Relative Length fields.
- Optimization of the correlation length parameter (for each data set).
- Optimization of the signal to noise ratio parameter (for each data set).
- Calculation of variable (semi normed) reference fields.
- Performance of variable analysis fields.
- Gnuplot plots production.
- Performances of analysis using data detrending method.
- Metadata XML files generation.

Note:

- When data extraction is activated in `driver`, the execution is made for all levels found in `contour.depth` file provided in the subdirectory `input`. It is also taking into account a minimum number of data in a layer with regard to the corresponding flag value given in `driver`.
- When "boundary lines and coastlines" generation is activated in `driver`, the execution is made for all levels found in `contour.depth` file provided in the subdirectory `input`.
- When parameter optimization and/or analysis is activated, the execution is made for the levels between the values choosen in the `driver` for the lower and upper level numbers.
- When parameter optimization (correlation length and/or signal to noise parameter) is activated in `driver`, the execution is made taking into account the bound values (maximum and minimum) given in the `driver`

1.3 Climatology directory: input files

In the Climatology directory, divadoall is the main script to be executed to produce climatologies. It uses several input files where it reads information about variables to be treaded, time periods, names, units, and actions to be performed. It is very important, before executing divadoall, to check if all the necessary input files are present and if they contain the right information.

Input files in Climatology directory:

- driver: The main input file to divadoall, all actions to be done are activated through corresponding flag values (see example 1.5.1)

In order to extract data sets from an ODV4 spreadsheets and define variable and climatologies:

- datasource: Contains filename of the ODV generic spreadsheet data file, from which data sets are extracted when data extraction is activated.
- varlist: contains in each line the name VAR of the variable to be treated (this is typically the header name from the ODV export) (see example 1.2.1).
- yearlist: contains in each line the period to be covered (see example 1.2.1). Ex 19301980 from 1930 to 1980 included.
- monthlist: contains on each line the range of months to be covered (see example 2.2.2). Ex 0103 means January, February and March together. Values as 1202 are allowed and cover December, January and February. For a climatology for each month, file monthlist therefore will contain 12 lines from 0101 to 1212.
- qflist: contains quality flags to be used when data extraction is activated (see example 1.3). If there is a qflist file, the selection with divaselectorODV4 will only use those measurements for which the quality flag is one of those found in the file qflist. When qflist is not provided, no quality flag analysis is done and all data will be taken.

The next two files are generated by divadoall when data extraction is activated (if data sets are not extracted from an ODV dpreadsheet, they have to be provided):

- VAR.units: the units of the variable. It is created when extracting data from an ODV spreadsheet data file. If you do not extract data using divadoall and prepare your data yourself, you need to provide a file VAR.units in the main climatology directory.
- VAR.longname: contains a long name of the variable (eg. temp.longname contains a line with "Temperature" if your variable name is temp and stands for temperature (see example 1.3)).

The next three files are used when gnuplot plots are required (option in `driver`:

- `VAR.bounds`: contains the lower and upper bounds during the plotting for the variable `VAR` (which is one of the variable names found in `varlist`)(see example 1.3).
- `VAR.pal`: contains the color palette for the same variable (see example 1.3).
- `plotboundingbox.dat`: contains the box for plotting (see example 1.3). This is typically used to plot only the region of interest, without overlapping regions with other climatologies (the numerical fields include the overlapping regions, only the plotting is limited with the `plotboundingbox.dat` file). File contains `xmin, xmax` on the first line, then `ymin, ymax` on the second line.

For Analysis calculation using " Advection constraint" and or reference fields, one has to provide:

- `constandrefe`: Contains flag values for " Advection constraint" and "Reference field use" activation, and related time period for reference fields when used (see example in section 2.2.3.

When generating metadata Xml files, one has to provide:

- `VAR.SDNconv`: Contains 4 additional informations: product label, product ID, parameter group and parameter name as shown in the example (1.3.7)

Input files in `Climatology/input` directory:

Here in `Climatology/input` directory, one must provide:

- `NCDFinfo`: Contains general information which will be used as self-documentation for climatology NetCDF files. A template file is provided which you have to edit and adapt to your application.
- `contour.depth`: Containe all considered depth values
- A topography file: `topogebco.asc`, `topo.gebco`, `topo.dat` or `topo.grd` and its related `TopoInfo.dat`

When generating metadata Xml files, one has to provide in `Climatology/input`:

- `general.info` Provides general information needed for identification of user organisations and rules for data access and delivery. The information related to each organisation must be identical to the one regestred in the European Directory of Marine Organisations (EDMO Seadatanet), and can be retrieved from: "http://seadatanet.maris2.nl/edmo".

```
0
3
1
```

Example file 1.3.1: `qflist` *file content.*

```
Temperature
```

Example file 1.3.2: `TEMP.longname` *file content.*

```
GMIN
-3
GMAX
32
```

Example file 1.3.3: `TEMP.bounds` *file content.*

```
set palette defined ( \
0 "gray20" ,\
4 "gray60" ,\
5 "light-blue", \
6 "skyblue",\
8 "aquamarine", \
10 "dark-turquoise",\
15 "light-green", \
16 "sea-green",\
35 "green", \
37 "yellow", \
38 "red", \
39.5 "dark-red",\
39.5 "dark-red",\
40 "violet",\
42 "gray10")
```

Example file 1.3.4: `TEMP.pal` *file content.*

```
-10 20
40 50
```

Example file 1.3.5: `plotboundingbox.dat` *file content.*

```
# advection flag
0
# reference field flag
1
# variable year code
19951995
# variable month code
0810
```

Example file 1.3.6: `constandrefe` *file content.*

```
TEMP-GHER-TST
GHER_ULG-MED-TEMP-OBS
TEMP: Temperature of the water column
sea_water_temperature,K,CF
```

Example file 1.3.7: Temperature.SDNconv file

1.4 Domain definition: Praparing the topo.gebco

The `topo.grd` in a binary file that Diva uses to generate `coast.cont` files, one for each depth and which define the domain on which the analyses are performed. The file `topo.grd` can be generated by providing an `ascii` topography file covering the region of interest. To generate `topo.grd` the topography file must provided in `input` directory. There are different options for naming the topography file:

- `topogebco.asc`: for GEBCO topography files, which may include a header and have negative values for sea's depths.
- `topo.gebco`: for GEBCO topography file which has no header and positive values for sea's depths are written with points (dots) and not comas.
- `topo.dat`: any topography three column file (lon, lat, val) with positive values for depths and decimal numbers are written with points (dots) and not comas

One can execute `gebcomodif` to eliminate header, change sign of depth values and replaces commas by dots for decimals (from `topogebco.asc`) as well as excluding regions that are not of interests (lowlands, lakes, unconnected regional seas, estuaries) and generate `topo.grd` file. If `topogebco.asc` is present in the `input`, `gebcomodif` will transform it to `topo.gebco` file (with no header and positive values for sea's depths and with points for decimals). If a file `takeout.coord` is present in the `input` directory, `gebcomodif` will apply the masking of the regions listed in it.

```
-10. -6. 33. 38.
-10. -1. 42. 49.
26.5 40. 40. 49.
```

Example file 1.4.1: `takeout.coord` *file: one line for each region to mask (minlon maxlon minlat maxlat).*

1.5 Climatologies production: Actions control

Numerouse outputs are generated by "**GODIVA**", which is the four dimentional extention of Diva for climatologies production and the reference the SeaDtaNet tool for this purpose. The main output of climatologies production is a four-dimentional netcdf file of a variable analysis, aside to 2D, 3D and 4D binary and ascii files. To produce a climatology several actions has to be taken. One has to generate the 3D-domaine (the region) of the application, by producing a set set of "coast contour" files from a aprovided topography file, extract variable data sets from an ODV4 spreadsheet and define the parameters and options for the analysis calculations. All actions and there different options are performed through flag values prescribed in the `driver` file located in the `Climatology` directory.

`driver` file must be edited and adapted before each execution of the `divadoall` execution.

1.5.1 `driver` file: actions and flag values

In the `driver` file, each flag may be associated to different actions through different value:

- **Data extraction**: Possible flag values: 0,1,-1 and -10. If you activate the data extraction (flag value $\neq 0$) in the `driver` file, the execution of `divadoall` will run the `divaselectorODV4` automatically, including interpolation to the levels specified in `contour.depth`. Data will be extracted from the ODV spreadsheat file(s) specified in `datasource`. `divaselectorODV4` will recognises if the data export to ODV file was done with depths in "meters" or it was done with pressure (in dbar) vertical coordinate, you can either choose to map it as if they were meters or apply the Saunders correction. Choose flag $= -1$ to use pressure coordinate and assume they are meters, and flag value $= -10$ to use pressure coordinates and transform to meters by using the Saunders approach.

  If there is a `qflist` file, the selection with `divaselectorODV4` will only use those measurements for which the quality flag is one of those found in the file `qflist`. In the absence of `qflist`, no quality flag analysis is done and all data taken.

  Note: you can specify several ODV spreadsheet files as input files, one name per line, in `datasource`, *provided they have the same naming, depth and quality conventions.*

- **Boundary lines and coastlines generation**: Possible flag values are 0, 1, 2 and 3. When this action is activated (flag ≥ 1), you must provide in the `input` directory the files `TopoInfo.dat` and `topo.grd` in addition to `contour.depth` file.

  - = 1 if contour files are to be generated,
  - = 2 if advection constraint (Anisotropic correlation along topography) files are to be generated from `topo.grd`,
  - = 3 if contour files and advection constraint are to be generated.

- **Cleaning data and Relative Length**: Possible flag values are 0, 1, 2, 3, 4 and 5:
  - = 1 if data files are to be cleaned,
  - = 2 if relative length files are to be generated,
  - = 3 if data files are to be cleaned and relative length files are to be generated.
  - = 4 if outliers are to be cleaned from data files.
  - = 5 if outliers are to be cleaned from data files and, relative length files to be generated.

- **Parameter optimization**: Possible flag values are 0, 1, 2, 3, −1, −2, −3, 10, −10, 30 and −30:
  - = 1 if correlation length parameters are to be estimated,
  - = 2 if signal to noise ratio (S/N) parameters are to be estimated,
  - = −1 if correlation length parameters are to be estimated and vertically filtered,
  - = −2 if signal to noise ratio (S/N) parameters are to be estimated and vertically filtered,
  - = 3 if both correlation length and signal to noise ratio parameters are to be estimated,
  - = −3 if both correlation length and signal to noise ratio parameters are to be estimated and vertically filtered,
  - = 10 if correlation length parameters are to be estimated using data mean distance as a minimum,
  - = −10 if correlation length parameters are to be estimated using data mean distance as a minimum and vertically filtered,
  - = 30 if both correlation length and signal to noise ratio parameters are to be estimated using data mean distance as a minimum (for CL),
  - = −30 if both correlation length and signal to noise ratio parameters are to be estimated using data mean distance as a minimum (for CL), and both parameters vertically filtered.

- **Analysis**: analysis and reference fields can be performed in different ways:
  - **Perform analysis**: Possible flag values are 1, 10, 11, 12, 13 and 14:
    - = 1 if analysis fields of the given variable are to be performed for all the layers between L_1 and L_2 which are the flag values for *lowerlevel number* and *upperlevel number* in the `driver`.

    * = 10 if analysis fields of the given variable are to be performed with exp(data)-log(analysis) transformation
    * = 11 if analysis fields of the given variable are to be performed with log(analysis)-exp(data) transformation
    * = 12 if analysis fields of the given variable are to be performed with $(\text{data})^2$-$\sqrt{(\text{analysis})}$ function transformation
    * = 13 if analysis fields of the given variable are to be performed with anamorphose transformation
    * = 14 if analysis fields of the given variable are to be performed with user choosen transformation
  - **Perform reference fields**: Possible flag values are 2, 20, 21, 22, 23 and 24:
    * = 2 if semi normed reference fields of the given variables (prescribed in `varlist` and for time periods described in `yearlist` and `monthlist`) are to be performed for all the layers between L_1 and L_2, which are the flag values for *lowerlevel number* and *upperlevel number* in the `driver`.
    * = 20 if analysis fields of the given variable are to be performed with exp(data)-log(analysis) transformation
    * = 21 if analysis fields of the given variable are to be performed with log(analysis)-exp(data) transformation
    * = 22 if analysis fields of the given variable are to be performed with $(\text{data})^2$-$\sqrt{(\text{analysis})}$ function transformation
    * = 23 if reference fields of the given variable are to be performed with anamorphose transformation
    * = 24 if reference fields of the given variable are to be performed with user choosen transformation
  - **Adding 100 to the flag value**:
    * = 101 or = $11x$ allows performing analysis using reference fields for each layer using all data from the two neighbouring layers in addition to the layer data set. Only reference fields are performed
    * = 102 or = $12x$ allows performing reference fields for each layer using all data from the two neighbouring layers in addition to the layer data set.

- **Metadata xml and NetCdf files**: Possible flag values are 0, 1, 2, 10, 11 and 12:
  * = 0 For each variable in `varlist` and each period in `yearlist` a NetCdf file is generated containing climatologies for all months in `monthlist`.
  * = 1:
    - For each variable in `varlist` and each period in `yearlist` a NetCdf file is generated containing climatologies for all months in `monthlist`.
    - Metadata Xml file is generated for each variable in `varlist` for **Sea-DataNet** CAMIOON catalogue.
  * = 2:
    - For each variable in `varlist` and each period in `yearlist` a NetCdf file is generated containing climatologies for all months in `monthlist`.

  - Metadata Xml file is generated for each variable in `varlist` for **EMODNET** CAMIOON catalogue.
- * = 10 For each variable in `varlist` a NetCdf file is generated containing climatologies and for all periods in `yearlist` and all months in `monthlist`.
- * = 11:
  - For each variable in `varlist` a NetCdf file is generated containing climatologies and for all periods in `yearlist` and all months in `monthlist`.
  - Metadata Xml file is generated for each variable in `varlist` for **SeaDataNet** CAMIOON catalogue.
- * = 12:
  - For each variable in `varlist` a NetCdf file is generated containing climatologies and for all periods in `yearlist` and all months in `monthlist`.
  - Metadata Xml file is generated for each variable in `varlist` for **EMODNET** CAMIOON catalogue.

- **Gnuplot plots**: Possible flag values are 0 and 1. Activate this action for a quick visualization (and assessment) of the climatology production, `gnuplot` executions can be included in the production process.

  There are a few controls you can apply for these gnuplot plots:

  * `VAR.bounds`: contains the lower and upper bounds during the plotting for the variable `VAR` (which is one of the variable names found in `varlist`)
  * `VAR.pal`: contains the color palette for the same variable.
  * `plotboundingbox.dat`: contains the box for plotting. This is typically used to plot only the region of interest, without overlapping regions with other climatologies (the numerical fields include the overlapping regions, only the plotting is limited with the `plotboundingbox.dat` file).

  Note: the gnuplot colorbars use a scale that is actually remapped to the bounds found in `VAR.bound`. Exemple: if your colorbar definition goes from 0 to 10 and the `VAR` bounds are from 0 and 100, a value of 50 in the variable analyzed will use the color found in the colorbar definition at value 5. To help you designing a specially adapted colorbar lets say for salinity, it is therefore a good idea to define the colorbar with the same bounds as those in `VAR.bounds`.

  Note: for adapting the color palette, file `gnuplotcolornames` contains a list of preexisting colors and their hexadecimal codes you can use instead of names.

- **Detrending** Possible flag values are 0 and n: the action is activated when choosing flag value an integer $n > 0$. The chosen value n must be equal or smaller to groups number in data files.

  Note: If you use `divadoall` (or `divaselectorODV4`) to extract data and create data input files, columns 5, 6 ,7 and 8 contain respectively groups years, month, days and hours (1 for the first year in the selection etc).

```
extract flag: 1 do it, 0 do nothing, -1 press coord, -10 pressure+Saunders
1
boundary lines and coastlines generation: 0 nothing, 1: contours, 2: UV, 3: 1+2
1
cleaning data on mesh: 1, 2: RL, 3: both, 4: 1 + outliers elimination, 5: =4+2
4
minimal number of data in a layer. If less, uses data from any month
10
Optimise parameters, possible values: 0, 1, 2, 3, -1, -2, -3, 10, -10, 30 and -30
-30
Minimal L
0.5
Maximal L
4
Minimal SN
0.05
Maximal SN
0.5
Analysis: pssible values: 0, 1, 2, 10 to 14, 20 to 24 and +100
1
lowerlevel number
1
upperlevel number
25
Metadata xml files genaration: 0, 1, 2, 10, 11, 12
1
isplot 0 or 1
1
Detrendind: 0 if no detrending, number of groups if detrending.
0
```

Example file 1.5.1: `driver` *file content.*

1.6 "Advection" constraint and Reference fields

1.6.1 "Advection" constraint fields generation

"Advection" constraint (Anisotropic correlation along topography) files are generated from `topo.grd` when this action is activated (corresponding flag ≥ 2). The generated files are stored in `newinput/divaUVcons_all`

1.6.2 Reference fields generation

Reference fields when generated (corresponding flag value = 2 in driver) are stored in `newinput/divarefe_all`

1.6.3 Advection constraint and Reference fields usage

If you have prepared advection constraint fields and/or reference fields, using `divadoall` you have to copy the resulting `newinput/divaUVcons_all` and `newinput/divarefe_all` to `input` to be able to use it.

Note: You have to provide in `/input/divaUVcons_all` an additional file: `constraint.dat`

If you are using Advection constraint field and/or Reference fields not generated by `divadoall`, you will have to store them respectively in `/input/divaUVcons_all` and `input/divarefe_all`. The advection constraint fields can be named as: `Uvel.level` and `Vvel.level`,or `Uvel.month.level` and `Vvel.month.level` or `Uvel.year.month.level` and `Vvel.year.month.level` where `year, month` take values specified in `yearlist, monthlist` and `level` is the level code `10000+ linenumber`, where `linenumber` is the number of the line of `contour.depth`.

Note: You have to provide in `/input/divaUVcons_all` two additional files: `constraint.dat` and `UVinfo.dat`.

The reference fields must be named as: `VAR.year.month.level` and copied in `/input/divarefe_all`. You have to provide a `GridInfo.dat` file in the same directory.

If you want to activate advection constraint and/or reference fields, you will have to provide a file `constandrefe` (aside `divadoall` and `driver`) with the flag value equal to 1 for the action to be taken. In this file the year code and month code common to reference variables must be given.

Note: When making analyses for several variables for different time periods (several months for example), the reference fields for all variable must have the same year code and same month code (as prescribed in constandrefe).

```
10 0
```

Example file 1.6.1: `constraint.dat` *file content.*

```
#Advection constraint flag: 1 do it, 0 no
1
#Reference fields flag: 1 use it, 0 no
1
#Reference variable year code
199051995
#Reference variable month code
0810
```

Example file 1.6.2: `constandrefe` *file content.*

1.7 newinput and output

All performed actions result on production of output files. Resulting files will be stored in newinput and in output/3Danalysis.

In newinput are stored divadoall output files that may be used as inputs for future runs:

- "Advection" constraint fields in divaUVcons_all
- Reference fields in divarefe_all
- coast.cont and param.par.VAR.year.month.level and related files when parameter optimization is done.

Note: The content of the newinput directory has to be copied to input after each divadoall execution and then removed.

In output/3Danalysis are stored (after each divadoall execution) all output files related to analyses.

- 3D and 4D netcdf files in output/3Danalysis
- 2D analysis (files for all levels) in output/3Danalysis/Fields
- 2D analysis (files for all levels) performed with original data sets (before outliers cleaning) in output/3Danalysis/FieldWithOutliers.
- Text and Xml metadata files in MetaDataXMLs

Chapter 2

Diva 4D Installation and Use

2.1 Installation

Copy the file "GODIVA.tar.gz" to you home directory. If you have a Diva directory "GODIVA" in you home, make a copy of it before copying the tar file.

unzip and untar "GODIVA.tar.gz":

```
> gunzip GODIVA.tar.gz
> tar xvf GODIVA.tar
```

You will obtain the new Diva directory: GODIVA.

Change directory to Fortran:

```
> cd ~/GODIVA/DIVA3D/src/Fortran
```

Edit the two script file divacompileall, and adapt the following lines:

```
compiler=
flags='-O'
nclib='-L/<Path>/lib -lnetcdf'
```

to your fortran compiler and netcdf library. And then run divacompile. Execute the the `divacompileall` to compile Fortran programs:

```
> divacompileall
```

Follow the compilation process and make sure that all fortran programs are compiled.

Change directory to input:

```
> cd ~/GODIVA/JRA4/Climatology/input
```

Edit the file `NCDFinfo` and adapt it to your case and application.

```
 Title string for 3D analysis NetCDF file:
'Diva 3D analysis '
 Reference time for data (ie: days since since 1900-01-01), if not climatological data
'months since since xxxx-01-01'
 Time value (that represents the data set), if not climatological data
1200
 Cell_method string:
'time: mean (this month data from all years)'
 Institution name: where the dataset was produced.
'My_Institution'
 Production group and date
'Diva group'
 Source (observation, radiosonde, database, model-generated data,...)
' data'
Comment
'No comment'
```

Example file 2.1.1: `NCDFinfo` *to be adapted*

2.2 Using GODIVA: an example

In this example, we will make monthly gridded climatology of meditramean sea temperatures. To this end, we prepare first an ODV4 spreadsheet of data, from which we will extract input data sets to our analysis process.

2.2.1 Domain definition and coast line contour files generation

Topography preparation

We use here an `ascii` file of mediterranean topography extracted from the GEBCO global topography (http://www.bodc.ac.uk/data/online_delivery/gebco/). The is named tt topogebco.asc and put in `input` directory. It contains a hader, and has negative values for depths and comas for decimals:

```
Gridded data provided through the GEBCO Centenary Edition software interface
Data source GEBCO One Minute Grid version 1.02
Minimum longitude : -6 degrees (+ve East)
Maximum longitude : 37 degrees (+ve East)
Minimum latitude : 30 degrees (+ve North)
Maximum latitude : 46 degrees (+ve North)
Data spacing : 1 minute
Rows*Columns : 961x2581
Units : Depths in Metres (terrestrial values are positive)
Points start at the North-West corner working eastwards across to the North-East
 before restarting in the West
Longitude Latitude Depth
354,0000  46,0000 -4657
354,0167  46,0000 -4654
354,0333  46,0000 -4652
354,0500  46,0000 -4650
354,0667  46,0000 -4648
354,0833  46,0000 -4646
354,1000  46,0000 -4645
```

Example file 2.2.1: `topogebco.asc` *file: head of file*

Now, in `Climatology` directory, we execute the shel script tt gebcomodif to "clean" teh file and generate a `topo.gebco`:

```
354.0000 46.0000 4657
354.0167 46.0000 4654
354.0333 46.0000 4652
354.0500 46.0000 4650
354.0667 46.0000 4648
354.0833 46.0000 4646
354.1000 46.0000 4645
354.1167 46.0000 4643
354.1333 46.0000 4642
354.1500 46.0000 4641
354.1667 46.0000 4640
354.1833 46.0000 4639
354.2000 46.0000 4638
```

Example file 2.2.2: `topo.gebco` *file: head of file*

Note The used `ascii` topography `topogebco.asc` file represents a squared area which includes parts of the Atlantic ocean, the Black Sea and some lakes and very low lands (sea figure 2.1)

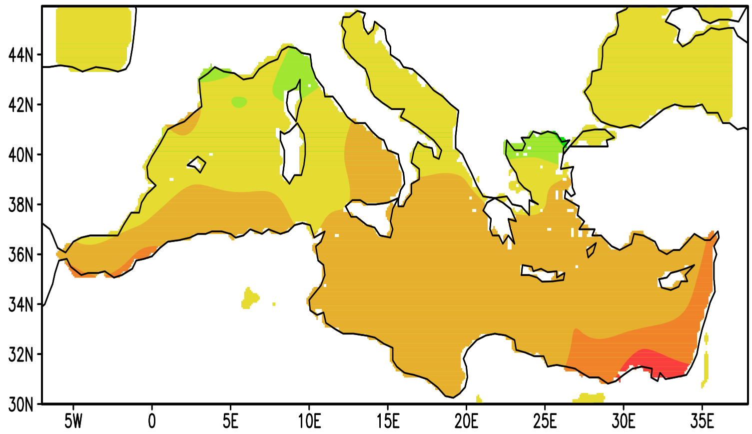


Figure 2.1: surface coast contour file: original topography

Masking of not desired areas can be done by providing in the `input` a file listing the squared areas we want to take out from topography: `takeout.coord` with `topogebco.asc` and executing `gebcomodif`

```
-10. -6. 33. 38.
-10. -1. 42. 49.
26.5 40. 40. 49.
5. 9. 33. 35.
20. 30. 30. 30.5
35. 37. 31. 33.
```

Example file 2.2.3: `takeout.coord` *file*

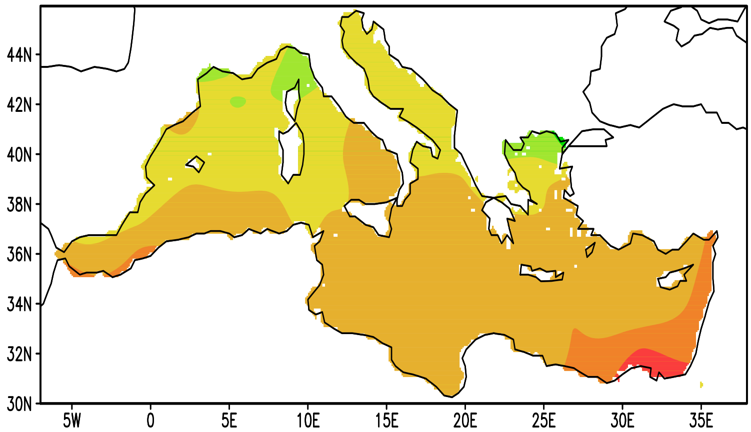


Figure 2.2: surface coast contour file: masked topography

To generate the boundary and coastline files. We provide in addition to `topo.gebco` a `param.par` and `contour.depth` files (see the Diva usermanual for details on this files) in the `Climatology/input` directory.

```
4000
3500
3000
2500
2000
1500
1200
1000
800
600
500
400
300
250
200
150
125
100
75
50
30
20
10
5
0
```

Example file 2.2.4: `contour.depth`

```
# Correlation Length lc, in km or degree, according to param icoordchange
2
# icoordchange (=0 if position of data in km ; =1 if position of data in degree)
1
# ispec (output files required)
11
# ireg (mode selected for background field: 0=null guess; 1=mean of data; 2=regression
1
# xori (origin of output regular grid, min values of X)
-7
# yori (origin of output regular grid, min values of Y)
30
# dx (step of output grid)
.09
# dy (step of output grid)
0.0625
# nx number of columns of the output grid
500
# ny number of lines of the output grid
256
# valex (exclusion value)
-9999.0
# snr signal to noise ratio
2
# varbak variance of the background field
0.35
```

Example file 2.2.5: `param.par` *file*

We use the following `driver`:

```
extract flag: 1 do it, 0 do nothing, -1 press coord, -10 pressure+Saunders
0
boundary lines and coastlines generation: 0 nothing, 1: contours, 2: UV, 3: 1+2
1
cleaning data on mesh: 1, 2: RL, 3: both, 4: 1 + outliers elimination, 5: =4+2
0
minimal number of data in a layer. If less, uses data from any month
10
Optimise parameters, possible values: 0, 1, 2, 3, -1, -2, -3, 10, -10, 30 and -30
0
Minimal L
0.5
Maximal L
4
Minimal SN
0.05
Maximal SN
5.
Analysis: pssible values: 0, 1, 2, 10 to 14, 20 to 24
0
lowerlevel number
1
upperlevel number
25
Metadata xml files genaration: 0 do nothing, 1 do it
0
isplot 0 or 1
0
Detrendind: 0 if no detrending, number of groups if detrending.
0
```

Example file 2.2.6: `driver` *file*

and execute `divadoall`.

After `divadoall` execution a `newinput` subdirectory is created in the `Climatology` directory and contains:

```
3Dinfo         divaparam      param.par   topo.grd
contour.depth  dodetrend.dat  topo.gebco  TopoInfo.dat
```

Example file 2.2.7: Content of `input` *directory*

The generated `coast.cont` coastline files (one for each depth in `contour.depth`) are stored in `newinput/divaparam/`:

```
coast.cont        coast.cont.10006  coast.cont.10012  coast.cont.10018  coast.cont.10024
coast.cont.10001  coast.cont.10007  coast.cont.10013  coast.cont.10019  coast.cont.10025
coast.cont.10002  coast.cont.10008  coast.cont.10014  coast.cont.10020
coast.cont.10003  coast.cont.10009  coast.cont.10015  coast.cont.10021
coast.cont.10004  coast.cont.10010  coast.cont.10016  coast.cont.10022
coast.cont.10005  coast.cont.10011  coast.cont.10017  coast.cont.10023
```

Example file 2.2.8: Content of `newinput/divaparam` *directory*

We move the `newinput/divaparam/` directory to `input` as well as the generated binary gridded topography `topo.grd` and its ralated info file `TopoInfo.dat`, after what we cab remove the `newinput` directory.

2.2.2 Data extraction

First step is data extraction, we check the ODV spreadsheet content by editing its header. In this case we have two variables: Temperature and salinity. We also notice that the export was done with pressure vertical coordinate.

```
//<Version>ODV Spreadsheet V4.0</Version>
//<Creator>mohamed@jilocasin</Creator>
//<CreateTime>2009-10-16T18:35:36</CreateTime>
//<Software>Ocean Data View Version 4.2.1 - 2009</Software>
//<Source>/media/disk-1/ALL_ARCHIVES/home/mohamed/medatlas.odv</Source>
//<SourceLastModified>2009-10-16T18:27:33</SourceLastModified>
//<DataField>GeneralField</DataField>
//<DataType>GeneralType</DataType>
//
Cruise Station Type yyyy-mm-ddThh:mm:ss.sss Longitude [degrees_east]
Latitude [degrees_north] Bot. Depth [m]:METAVAR:FLOAT:4 Pressure [decibar] QF
Temperature [degrees Celsius] QF Salinity [psu] QF QV:ODV:SAMPLE
```

Example file 2.2.9: header of odv spreadsheet file `data_from_medatlas.txt`

We provide the input information file for semestrial data sets extraction: `varlist`, `yearlist` and `monthlist` in `Climatology` directory.

```
Temperature
```

Example file 2.2.10: `varlist` *file example.*

```
19002010
```

Example file 2.2.11: `yearlist` *file example.*

```
0101
0202
0303
0404
0505
0606
0707
0808
0909
1010
1111
1212
```

Example file 2.2.12: `monthlist` *file example.*

And use the following `driver` and `datasource` files:

```
extract flag: 1 do it, 0 do nothing, -1 press coord, -10 pressure+Saunders
-10
boundary lines and coastlines generation: 0 nothing, 1: contours, 2: UV, 3: 1+2
0
cleaning data on mesh: 1, 2: RL, 3: both, 4: 1 + outliers elimination, 5: =4+2
0
minimal number of data in a layer. If less, uses data from any month
10
Optimise parameters, possible values: 0, 1, 2, 3, -1, -2, -3, 10, -10, 30 and -30
0
Minimal L
0.5
Maximal L
4
Minimal SN
0.05
Maximal SN
5.
Analysis: pssible values: 0, 1, 2, 10 to 14, 20 to 24
0
lowerlevel number
1
upperlevel number
25
Metadata xml files genaration: 0 do nothing, 1 do it
0
isplot 0 or 1
0
Detrendind: 0 if no detrending, number of groups if detrending.
0
```

Example file 2.2.13: `driver` *file*

```
data_from_medatlas.txt
```

Example file 2.2.14: `datasource` *file*

content of `input` before running `divadoall`:

```
3Dinfosalinity  contour.depth  divaparam           Stabinfo   NCDFinfo\_temperature
3Dinfotemper    NCDFinfo       NCDFinfo\_salinity  topo.gebco topo.grd
```

Example file 2.2.15: Content of `input` *directory*

We execute `divadoall` in `Climatology` directory, and after the execution, the

a subdirectory `divadata` is created in `Climatology/input` directory:

```
3Dinfosalinity  contour.depth  divaparam NCDFinfo_salinity    Stabinfo
3Dinfotemper    divadata       NCDFinfo  NCDFinfo_temperature topo.dat
topo.grd
```

Example file 2.2.16: Content of `input` *directory*

the subdirectory `divadata` contains extracted data files:

```
Temperature.19002010.0101.10001
Temperature.19002010.0101.10002
Temperature.19002010.0101.10003
Temperature.19002010.0101.10004
Temperature.19002010.0101.10005
Temperature.19002010.0101.10006
Temperature.19002010.0101.10007
.
.
Temperature.19002010.1212.10018
Temperature.19002010.1212.10019
Temperature.19002010.1212.10020
Temperature.19002010.1212.10021
Temperature.19002010.1212.10022
Temperature.19002010.1212.10023
Temperature.19002010.1212.10024
Temperature.19002010.1212.10025
```

Example file 2.2.17: Content of `input/divadata` *subdirectory*

And `Temperature.unit` and `Salinity.unit` were generated in `Climatology`:

```
degrees Celsius
```

Example file 2.2.18: `Temperature.unit` *file*

2.2.3 Variable analysis

We do the analysis in two steps:

Step 1: we configure the `driver` in a way to:

- clean data on the mesh and outliers (flag= 5 in driver)
- estimate Correlation Length (L), Signal to Noise (λ) and the Varbak parameters, using bounds based on "mean distance" of data distribution calculation and a vertical filtering of L and λ parameters (flag= -30 in driver)

```
extract flag: 1 do it, 0 do nothing, -1 press coord, -10 pressure+Saunders
0
boundary lines and coastlines generation: 0 nothing, 1: contours, 2: UV, 3: 1+2
0
cleaning data on mesh: 1, 2: RL, 3: both, 4: 1 + outliers elimination, 5: =4+2
5
minimal number of data in a layer. If less, uses data from any month
10
Optimise parameters, possible values: 0, 1, 2, 3, -1, -2, -3, 10, -10, 30 and -30
-30
Minimal L
0.5
Maximal L
4
Minimal SN
0.05
Maximal SN
5.
Analysis: pssible values: 0, 1, 2, 10 to 14, 20 to 24
0
lowerlevel number
1
upperlevel number
25
Metadata xml files genaration: 0 do nothing, 1 do it
0
isplot 0 or 1
0
Detrendind: 0 if no detrending, number of groups if detrending.
0
```

Example file 2.2.19: `driver` *file configuration.*

After executing `divadoall`, the generated `newinput` contains the clean data sets in `newinput/divadata` and the new `param.par` files in `newinput/divaparam`

```
# Correlation Length lc, in km or degree, according to param icoordchange
2.96449
# icoordchange (=0 if position of data in km ; =1 if position of data in degree)
1
# ispec (output files required)
11
# ireg (mode selected for background field: 0=null guess; 1=mean of data; 2=regression plan if at le
1
# xori (origin of output regular grid, min values of X)
-7
# yori (origin of output regular grid, min values of Y)
30
# dx (step of output grid)
.09
# dy (step of output grid)
0.0625
# nx number of columns of the output grid
500
# ny number of lines of the output grid
256
# valex (exclusion value)
-9999.0
# snr signal to noise ratio
0.8680236
# varbak variance of the background field
0.9127094
```

Example file 2.2.20: an example of a generated `param.par` *file:* `param.par.Temperature.19002010.0404.10025` .

To use the new generated data sets and `param.par` files, we move `newinput/divadata` and `newinput/divaparam` to `input/divadata` and `input/divaparam`, after what we can deleat `newinput`.

Step 2: we want to make analysis of variable `Temperature` for the 12 months. we use the following `driver`:

```
extract flag: 1 do it, 0 do nothing, -1 press coord, -10 pressure+Saunders
0
boundary lines and coastlines generation: 0 nothing, 1: contours, 2: UV, 3: 1+2
0
cleaning data on mesh: 1, 2: RL, 3: both, 4: 1 + outliers elimination, 5: =4+2
0
minimal number of data in a layer. If less, uses data from any month
10
Optimise parameters, possible values: 0, 1, 2, 3, -1, -2, -3, 10, -10, 30 and -30
0
Minimal L
0.5
Maximal L
4
Minimal SN
0.05
Maximal SN
5.
Analysis: pssible values: 0, 1, 2, 10 to 14, 20 to 24
1
lowerlevel number
1
upperlevel number
25
Metadata xml files genaration: 0 do nothing, 1 do it
0
isplot 0 or 1
1
Detrendind: 0 if no detrending, number of groups if detrending.
0
```

Example file 2.2.21: `driver` *file configuration.*

Note that only analysis (flag = 1) and gnuplot plots (flag = 1) actions are activated.

After `divadoall` execution, we obtain the results in `output/3Danalysis`:

- `Fields`: a subdirectory where are stored Temperature and Salinity 2D fields analyses, performed with cleaned data sets.
- `FieldWithOutliers`: a subdirectory where are stored Temperature and Salinity 2D fields analyses, performed with original data sets before outliers cleaning.
- `GnuPlots`: a subdirectory where are stored all gnuplot plots produced while analyzing fields using cleaned data sets.
- `GPlotsWOutlrs`: a subdirectory where are stored gnuplot plots produced while analyzing fields using original data sets.
- `Temperature.19002010.0101.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0202.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file

- `Temperature.19002010.0303.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0404.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0505.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0606.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0707.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0808.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0909.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.1010.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.1111.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.1212.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19002010.0101.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0202.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0303.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0404.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0505.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0606.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0707.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0808.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.0909.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.1010.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.1111.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.1212.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19002010.4Danl.nc`: 4D Temperature analyses netcdf file