Diva User's Guide

Four Dimensional Climatology analysis tool

(last modified October 19, 2009)

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

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

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 definition

1.2.1 Variables and time periods

To decide what to produce 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 `19301980` from 1930 to 1980 included.
- `monthlist`: contains on each line the range of months to be covered. 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.

```
TEMP
PSAL
```

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.

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 Checklist of input files

In the `Climatology` directory, `divadoall` 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 the necessary input files are present and if they contain the right information.

Input files in `Climatology`:

- `driver`: The main input file to `divadoall`, all actions to be done are activated through corresponding flag values (see example 1.4)
- `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 2.2.4). 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`. In the absence of `qflist`, no quality flag analysis is done and all data taken.
- `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)).
- `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.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.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.
- `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 2.2.5.

Input files in `Climatology/input`:

- `NCDFinfo`: Contains information about global attributes for Netcdf files. You have to edit this file and adapt it to your application.

```
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.*

1.4 Actions control

For the control of the execution edit `driver` and adapt the self explaining parameters

```
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
isoptimise 0 nothing, 1 L, 2 SN, 3 both,  negative values filter vertically
0
Minimal L
0.1
Maximal L
1
Minimal SN
0.05
Maximal SN
0.5
2 do reference, 1 do analysis and 0 do nothing
1
lowerlevel number
7
upperlevel number
11
reference (to come)
0
isplot 0 or 1
1
number of groups for data detrending, 0 if no detrending.
0
```

Example file 1.4.1: `driver` *file content.*

1.4.1 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`. Choose flag value $= 1$ when data export to ODV file was done with depths in "meters". If original data export 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.

- **Perform analysis**: Possible flag values are 0, 1 and 2:
  * = 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`.
  * = 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`.

- **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).

1.5 "Advection" constraint and Reference fields

1.5.1 "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`

1.5.2 Reference fields generation

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

1.5.3 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` 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 1.5.1: `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 1.5.2: `constandrefe` *file content.*

1.6 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.

In `output/3Danalysis` are stored `divadoall` output files of 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`.

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 files divacompile and divahccompile, and adapt the following lines:

```
compiler=
flags='-O'

noglobals=
nocyg=
nclib='-L/<Path>/lib -lnetcdf'
```

Example file 2.1.1: `divacompile` *and* `divahccompile` *lines to be adapted*

to your fortran compiler and netcdf library. And then run divacompile:

```
> divacompile
```

and divahccompile:

```
> divahccompile
```

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`:

```
 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 2.1.2: `NCDFinfo` *to be adapted*

2.2 Using GODIVA: an example

In this example, we will make monthly gridded climatology of temperature and salinity from the Medatlas data base (at http://www.ifremer.fr/medar/. To this end, we prepare first an ODV4 spreadsheet of data, from which we will extract input data sets to our analysis process. We will need a semestrial data sets to make (semi normed) reference fields, and monthly data sets to make the monthly climatology. **Topography**: we use a topography file `topo.grd`, transformed by `gebco` export of an `ascii` topography and transform to Diva file by `gebco2diva 15 15` (see the Diva user manual for details).

2.2.1 Domain definition and coast line contour files generation

First we have to generate the boundary lines and coastlines files. We have to provide in addition to `topo.grd` and `TopoInfo.dat`, `param.par` and `contour.depth` files in the `Climatology/input` directory. We also generate at this step "advection" constraint files based on topography.

```
4000.e0
3500.e0
3000.e0
2500.e0
2000.e0
1500.e0
1200.e0
1000.e0
800.e0
600.e0
500.e0
400.e0
300.e0
250.e0
200.e0
150.e0
125.e0
100.e0
75.e0
50.e0
30.e0
20.e0
10.e0
5.e0
0.e0
```

Example file 2.2.1: `contour.depth`

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
3
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
isoptimise 0 nothing, 1 L, 2 SN, 3 both,  negative values filter vertically
0
Minimal L
0.1
Maximal L
1
Minimal SN
0.05
Maximal SN
0.5
2 do reference, 1 do analysis and 0 do nothing
0
lowerlevel number
1
upperlevel number
25
reference (to come)
0
isplot 0 or 1
0
number of groups for data detrending, 0 if no detrending.
0
```

Example file 2.2.2: `driver` *file*

and the following `param.par`:

```
# 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
20
# varbak variance of the background field
0.35
```

Example file 2.2.3: `param.par` *file*

and execute `divadoall`.

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

```
3Dinfo          contour.depth  divarefe_all    NCDFinfo_salinity     topo.grd
3Dinfosalinity  data.dat       divaUVcons      NCDFinfo_temperature  TopoInfo.dat
3Dinfotemper    divadata       divaUVcons_all  param.par             valatxyascii.anl
coast.cont      divamesh       dodetrend.dat   Stabinfo
constraint.dat  divaparam      NCDFinfo        topo.dat
```

Example file 2.2.4: Content of `input` *directory*

The generated boundary lines and coastlines files 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 2.2.5: Content of `newinput/divaparam` *directory*

And "advection" files are stored in `newinput/divaUVcons_all/`:

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

Example file 2.2.6: Content of `newinput/divaUVcons_all` *directory*

We (have to) add a `constraint.dat` file in `newinput/divaUVcons_all` (specifying "advection" and "diffusion" coefficients:

```
10 0
```

Example file 2.2.7: `constraint.dat` *file content.*

To proceed with the following steps we remove the `input` directory, and rename `newinput` as `input`.

2.2.2 Data extraction for semi normed analysis

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 2.2.8: 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
Salinity
```

Example file 2.2.9: `varlist` *file example.*

```
19502000
```

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

```
0106
0712
```

Example file 2.2.11: `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
-1
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
isoptimise 0 nothing, 1 L, 2 SN, 3 both,  negative values filter vertically
0
Minimal L
0.1
Maximal L
1
Minimal SN
0.05
Maximal SN
0.5
2 do reference, 1 do analysis and 0 do nothing
0
lowerlevel number
1
upperlevel number
25
reference (to come)
0
isplot 0 or 1
0
number of groups for data detrending, 0 if no detrending.
0
```

Example file 2.2.12: `driver` *file*

```
data_from_medatlas.txt
```

Example file 2.2.13: `datasource` *file*

content of `input` before running `divadoall`:

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

Example file 2.2.14: Content of `input` *directory*

We execute `divadoall` in `Climatology` directory, and after the execution, the `Climatology/input` directory content becomes:

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

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

a subdirectory `divadata` was created, and its content is:

```
Salinity.19502000.0106.10001  Salinity.19502000.0712.10010     Temperature.19502000.0106.10019
Salinity.19502000.0106.10002  Salinity.19502000.0712.10011     Temperature.19502000.0106.10020
Salinity.19502000.0106.10003  Salinity.19502000.0712.10012     Temperature.19502000.0106.10021
Salinity.19502000.0106.10004  Salinity.19502000.0712.10013     Temperature.19502000.0106.10022
Salinity.19502000.0106.10005  Salinity.19502000.0712.10014     Temperature.19502000.0106.10023
Salinity.19502000.0106.10006  Salinity.19502000.0712.10015     Temperature.19502000.0106.10024
Salinity.19502000.0106.10007  Salinity.19502000.0712.10016     Temperature.19502000.0106.10025
Salinity.19502000.0106.10008  Salinity.19502000.0712.10017     Temperature.19502000.0712.10001
Salinity.19502000.0106.10009  Salinity.19502000.0712.10018     Temperature.19502000.0712.10002
Salinity.19502000.0106.10010  Salinity.19502000.0712.10019     Temperature.19502000.0712.10003
Salinity.19502000.0106.10011  Salinity.19502000.0712.10020     Temperature.19502000.0712.10004
Salinity.19502000.0106.10012  Salinity.19502000.0712.10021     Temperature.19502000.0712.10005
Salinity.19502000.0106.10013  Salinity.19502000.0712.10022     Temperature.19502000.0712.10006
Salinity.19502000.0106.10014  Salinity.19502000.0712.10023     Temperature.19502000.0712.10007
Salinity.19502000.0106.10015  Salinity.19502000.0712.10024     Temperature.19502000.0712.10008
Salinity.19502000.0106.10016  Salinity.19502000.0712.10025     Temperature.19502000.0712.10009
Salinity.19502000.0106.10017  Temperature.19502000.0106.10001  Temperature.19502000.0712.10010
Salinity.19502000.0106.10018  Temperature.19502000.0106.10002  Temperature.19502000.0712.10011
Salinity.19502000.0106.10019  Temperature.19502000.0106.10003  Temperature.19502000.0712.10012
Salinity.19502000.0106.10020  Temperature.19502000.0106.10004  Temperature.19502000.0712.10013
Salinity.19502000.0106.10021  Temperature.19502000.0106.10005  Temperature.19502000.0712.10014
Salinity.19502000.0106.10022  Temperature.19502000.0106.10006  Temperature.19502000.0712.10015
Salinity.19502000.0106.10023  Temperature.19502000.0106.10007  Temperature.19502000.0712.10016
Salinity.19502000.0106.10024  Temperature.19502000.0106.10008  Temperature.19502000.0712.10017
Salinity.19502000.0106.10025  Temperature.19502000.0106.10009  Temperature.19502000.0712.10018
Salinity.19502000.0712.10001  Temperature.19502000.0106.10010  Temperature.19502000.0712.10019
Salinity.19502000.0712.10002  Temperature.19502000.0106.10011  Temperature.19502000.0712.10020
Salinity.19502000.0712.10003  Temperature.19502000.0106.10012  Temperature.19502000.0712.10021
Salinity.19502000.0712.10004  Temperature.19502000.0106.10013  Temperature.19502000.0712.10022
Salinity.19502000.0712.10005  Temperature.19502000.0106.10014  Temperature.19502000.0712.10023
Salinity.19502000.0712.10006  Temperature.19502000.0106.10015  Temperature.19502000.0712.10024
Salinity.19502000.0712.10007  Temperature.19502000.0106.10016  Temperature.19502000.0712.10025
Salinity.19502000.0712.10008  Temperature.19502000.0106.10017
Salinity.19502000.0712.10009  Temperature.19502000.0106.10018
```

Example file 2.2.16: Content of `input/divadata` *subdirectory*

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

```
degrees Celsius
```

Example file 2.2.17: `Temperature.unit` *file*

```
psu
```

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

2.2.3 Reference fields generation

To generate Reference fields we use the following `driver` configuration:

```
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
isoptimise 0 nothing, 1 L, 2 SN, 3 both,  negative values filter vertically
0
Minimal L
0.1
Maximal L
1
Minimal SN
0.05
Maximal SN
0.5
2 do reference, 1 do analysis and 0 do nothing
2
lowerlevel number
1
upperlevel number
25
reference (to come)
2
isplot 0 or 1
1
number of groups for data detrending, 0 if no detrending.
0
```

Example file 2.2.19: `driver` *file*

Note that data cleaning and outliers elimination are activated (corresponding flag = 5). **Note** flag number for analysis is = 2 for reference fields generation.

It is also our choice to use one `param.par` for all calculation. The used `param.par` is:

```
# 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
0
# 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
20
# varbak variance of the background field
0.35
```

Example file 2.2.20: `param.par` *file*

Note the value of the `ireg` parameter (= 0).

We execute `divadoall` and obtain the following results in `newinput` and `output/3Danalysis`:

- **newinput**:

  contains a copy of `input` contents (used for this run)

  the reference fields are stored in `divarefe_all`

  in `divadata` cleaned data sets are stored with a copy original data sets

- **output/3Danalysis**:

  `Fields`: a subdirectory where are stored Temperature and Salinity 2D reference fields analyses, performed with cleaned data sets.

  `FieldWithOutliers`: a subdirectory where are stored Temperature and Salinity 2D reference fields analyses, performed with original data sets before outliers cleaning.

  `GnuPlots`: a subdirectory where are stored all gnuplot plots produced while analyzing reference fields using cleaned data sets.

  `GPlotsWOutlrs`: a subdirectory where are stored gnuplot plots produced while analyzing reference fields using original data sets.

`Salinity.19502000.0106.10001.10025.fieldgher.ref`: Salinity 3D binary GHER format file

`Salinity.19502000.0106.10001.10025.ref.nc`: Salinity 3D netcdf file

`Salinity.19502000.0712.10001.10025.fieldgher.ref`: Salinity 3D binary GHER format file

`Salinity.19502000.0712.10001.10025.ref.nc`: Salinity 3D netcdf file

`Temperature.19502000.0106.10001.10025.fieldgher.ref`: Temperature 3D binary GHER format file

`Temperature.19502000.0106.10001.10025.ref.nc`: Temperature 3D netcdf file

`Temperature.19502000.0712.10001.10025.fieldgher.ref`: Temperature 3D binary GHER format file

`Temperature.19502000.0712.10001.10025.ref.nc`: Temperature 3D netcdf file

```
Fields
FieldWithOutliers
GnuPlots
GPlotsWOutlrs
Salinity.19502000.0106.10001.10025.fieldgher.ref
Salinity.19502000.0106.10001.10025.ref.nc
Salinity.19502000.0712.10001.10025.fieldgher.ref
Salinity.19502000.0712.10001.10025.ref.nc
Temperature.19502000.0106.10001.10025.fieldgher.ref
Temperature.19502000.0106.10001.10025.ref.nc
Temperature.19502000.0712.10001.10025.fieldgher.ref
Temperature.19502000.0712.10001.10025.ref.nc
```

Example file 2.2.21: Content of `output/3Danalysis` *directory*

To use the generated reference fields, we remove the `input` directory and rename `newinput` as `input`.

2.2.4 Data extraction for monthly analysis

we modify the `monthlist` file to work with monthly data:

```
0101
0202
0303
0404
0505
0606
```

Example file 2.2.22: `monthlist` *file example.*

We first extract the data using the following `driver`:

```
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
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
isoptimise 0 nothing, 1 L, 2 SN, 3 both,  negative values filter vertically
0
Minimal L
0.1
Maximal L
1
Minimal SN
0.05
Maximal SN
0.5
2 do reference, 1 do analysis and 0 do nothing
0
lowerlevel number
1
upperlevel number
25
reference (to come)
2
isplot 0 or 1
0
number of groups for data detrending, 0 if no detrending.
0
```

Example file 2.2.23: `driver` *file*

we execute `divadoall` and obtain the new extracted data sets in `input/divadata`:

```
Salinity.19502000.0101.10001              Temperature.19502000.0101.10001
Salinity.19502000.0101.10002              Temperature.19502000.0101.10002
Salinity.19502000.0101.10003              Temperature.19502000.0101.10003
Salinity.19502000.0101.10004              Temperature.19502000.0101.10004
Salinity.19502000.0101.10005              Temperature.19502000.0101.10005
Salinity.19502000.0101.10006              Temperature.19502000.0101.10006
Salinity.19502000.0101.10007              Temperature.19502000.0101.10007
Salinity.19502000.0101.10008              Temperature.19502000.0101.10008
Salinity.19502000.0101.10009              Temperature.19502000.0101.10009
Salinity.19502000.0101.10010              Temperature.19502000.0101.10010
Salinity.19502000.0101.10011              Temperature.19502000.0101.10011
Salinity.19502000.0101.10012              Temperature.19502000.0101.10012
Salinity.19502000.0101.10013              Temperature.19502000.0101.10013
.....
Salinity.19502000.0303.10001              Temperature.19502000.0303.10001
Salinity.19502000.0303.10002              Temperature.19502000.0303.10002
Salinity.19502000.0303.10003              Temperature.19502000.0303.10003
Salinity.19502000.0303.10004              Temperature.19502000.0303.10004
Salinity.19502000.0303.10005              Temperature.19502000.0303.10005
Salinity.19502000.0303.10006              Temperature.19502000.0303.10006
Salinity.19502000.0303.10007              Temperature.19502000.0303.10007
Salinity.19502000.0303.10008              Temperature.19502000.0303.10008
Salinity.19502000.0303.10009              Temperature.19502000.0303.10009
Salinity.19502000.0303.10010              Temperature.19502000.0303.10010
Salinity.19502000.0303.10011              Temperature.19502000.0303.10011
Salinity.19502000.0303.10012              Temperature.19502000.0303.10012
Salinity.19502000.0303.10013              Temperature.19502000.0303.10013
Salinity.19502000.0303.10014              Temperature.19502000.0303.10014
Salinity.19502000.0303.10015              Temperature.19502000.0303.10015
Salinity.19502000.0303.10016              Temperature.19502000.0303.10016
Salinity.19502000.0303.10017              Temperature.19502000.0303.10017
Salinity.19502000.0303.10018              Temperature.19502000.0303.10018
Salinity.19502000.0303.10019              Temperature.19502000.0303.10019
Salinity.19502000.0303.10020              Temperature.19502000.0303.10020
Salinity.19502000.0303.10021              Temperature.19502000.0303.10021
Salinity.19502000.0303.10022              Temperature.19502000.0303.10022
Salinity.19502000.0303.10023              Temperature.19502000.0303.10023
Salinity.19502000.0303.10024              Temperature.19502000.0303.10024
Salinity.19502000.0303.10025              Temperature.19502000.0303.10025
.........
Salinity.19502000.0606.10016              Temperature.19502000.0606.10016
Salinity.19502000.0606.10017              Temperature.19502000.0606.10017
Salinity.19502000.0606.10018              Temperature.19502000.0606.10018
Salinity.19502000.0606.10019              Temperature.19502000.0606.10019
Salinity.19502000.0606.10020              Temperature.19502000.0606.10020
Salinity.19502000.0606.10021              Temperature.19502000.0606.10021
Salinity.19502000.0606.10022              Temperature.19502000.0606.10022
Salinity.19502000.0606.10023              Temperature.19502000.0606.10023
Salinity.19502000.0606.10024              Temperature.19502000.0606.10024
Salinity.19502000.0606.10025              Temperature.19502000.0606.10025
........
Salinity.19502000.0712.10001.DATABINS     Temperature.19502000.0712.10001.DATABINS
Salinity.19502000.0712.10001.DATABINSinfo Temperature.19502000.0712.10001.DATABINSinfo
Salinity.19502000.0712.10001.notclean     Temperature.19502000.0712.10001.notclean
Salinity.19502000.0712.10001.withoutliers Temperature.19502000.0712.10001.withoutliers
Salinity.19502000.0712.10002.DATABINS     Temperature.19502000.0712.10002.DATABINS
......
```

Example file 2.2.24: Content of `input/divadata` *directory*

2.2.5 Variables 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
isoptimise 0 nothing, 1 L, 2 SN, 3 both,  negative values filter vertically
-30
Minimal L
1.
Maximal L
10.
Minimal SN
0.1
Maximal SN
10.0
analysis 1 do it 0 do nothing
0
lowerlevel number
1
upperlevel number
25
reference (to come)
0
isplot 0 or 1
0
number of groups for data detrending, 0 if no detrending.
0
```

Example file 2.2.25: `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`

To use the new generated data sets and `param.par` files, we replace the `input` directory by `newinput` (delete `input` and rename `newinput` as `input`).

```
param.par.Temperature.19502000.0303.10023  Salinity.19502000.0202.CL.dat.filtered
param.par.Temperature.19502000.0303.10024  Salinity.19502000.0202.CL.dat.notfiltered
param.par.Temperature.19502000.0303.10025  Salinity.19502000.0202.gcv.dat.10021
param.par.Temperature.19502000.0404.10021  Salinity.19502000.0202.gcv.dat.10022
param.par.Temperature.19502000.0404.10022  Salinity.19502000.0202.gcv.dat.10023
param.par.Temperature.19502000.0404.10023  Salinity.19502000.0202.gcv.dat.10024
param.par.Temperature.19502000.0404.10024  Salinity.19502000.0202.gcv.dat.10025
param.par.Temperature.19502000.0404.10025  Salinity.19502000.0202.SN.dat.filtered
param.par.Temperature.19502000.0505.10021  Salinity.19502000.0202.SN.dat.notfiltered
param.par.Temperature.19502000.0505.10022  Salinity.19502000.0202.VAR.dat.filtered
param.par.Temperature.19502000.0505.10023  Salinity.19502000.0202.VAR.dat.notfiltered
param.par.Temperature.19502000.0505.10024  Salinity.19502000.0303.CL.dat.filtered
param.par.Temperature.19502000.0505.10025  Salinity.19502000.0303.CL.dat.notfiltered
param.par.Temperature.19502000.0606.10021  Salinity.19502000.0303.gcv.dat.10021
param.par.Temperature.19502000.0606.10022  Salinity.19502000.0303.gcv.dat.10022
param.par.Temperature.19502000.0606.10023  Salinity.19502000.0303.gcv.dat.10023
param.par.Temperature.19502000.0606.10024  Salinity.19502000.0303.gcv.dat.10024
param.par.Temperature.19502000.0606.10025  Salinity.19502000.0303.gcv.dat.10025
RLinfo.dat                                 Salinity.19502000.0303.SN.dat.filtered
RL.Salinity.19502000.0101.10021            Salinity.19502000.0303.SN.dat.notfiltered
RL.Salinity.19502000.0101.10022            Salinity.19502000.0303.VAR.dat.filtered
RL.Salinity.19502000.0101.10023            Salinity.19502000.0303.VAR.dat.notfiltered
```

Example file 2.2.26: an example of divaparam *content.*

```
Salinity.19502000.0505.10021.DATABINS        Temperature.19502000.0505.10021.DATABINS
Salinity.19502000.0505.10021.DATABINSinfo    Temperature.19502000.0505.10021.DATABINSinfo
Salinity.19502000.0505.10021.notclean        Temperature.19502000.0505.10021.notclean
Salinity.19502000.0505.10021.ori             Temperature.19502000.0505.10021.ori
Salinity.19502000.0505.10021.withoutliers    Temperature.19502000.0505.10021.withoutliers
Salinity.19502000.0505.10022                 Temperature.19502000.0505.10022
Salinity.19502000.0505.10022.DATABINS        Temperature.19502000.0505.10022.DATABINS
Salinity.19502000.0505.10022.DATABINSinfo    Temperature.19502000.0505.10022.DATABINSinfo
Salinity.19502000.0505.10022.notclean        Temperature.19502000.0505.10022.notclean
Salinity.19502000.0505.10022.ori             Temperature.19502000.0505.10022.ori
Salinity.19502000.0505.10022.withoutliers    Temperature.19502000.0505.10022.withoutliers
Salinity.19502000.0505.10023                 Temperature.19502000.0505.10023
Salinity.19502000.0505.10023.DATABINS        Temperature.19502000.0505.10023.DATABINS
Salinity.19502000.0505.10023.DATABINSinfo    Temperature.19502000.0505.10023.DATABINSinfo
Salinity.19502000.0505.10023.notclean        Temperature.19502000.0505.10023.notclean
Salinity.19502000.0505.10023.ori             Temperature.19502000.0505.10023.ori
Salinity.19502000.0505.10023.withoutliers    Temperature.19502000.0505.10023.withoutliers
Salinity.19502000.0505.10024                 Temperature.19502000.0505.10024
Salinity.19502000.0505.10024.DATABINS        Temperature.19502000.0505.10024.DATABINS
Salinity.19502000.0505.10024.DATABINSinfo    Temperature.19502000.0505.10024.DATABINSinfo
Salinity.19502000.0505.10024.notclean        Temperature.19502000.0505.10024.notclean
Salinity.19502000.0505.10024.ori             Temperature.19502000.0505.10024.ori
Salinity.19502000.0505.10024.withoutliers    Temperature.19502000.0505.10024.withoutliers
Salinity.19502000.0505.10025                 Temperature.19502000.0505.10025
Salinity.19502000.0505.10025.DATABINS        Temperature.19502000.0505.10025.DATABINS
Salinity.19502000.0505.10025.DATABINSinfo    Temperature.19502000.0505.10025.DATABINSinfo
Salinity.19502000.0505.10025.notclean        Temperature.19502000.0505.10025.notclean
Salinity.19502000.0505.10025.ori             Temperature.19502000.0505.10025.ori
Salinity.19502000.0505.10025.withoutliers    Temperature.19502000.0505.10025.withoutliers
```

Example file 2.2.27: an example of `divadata` *content.*

```
# 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 2.2.28: an example of a generated `param.par` *file:* `param.par.Temperature.19502000.0404.10025` .

Step 2: we want to make analysis of the two variables `Temperature` and `Salinity` for the first six months. To use the references and the "advection" constraint fields prepared before, we have to provide the information in the file `constandrefe` as follow:

```
# advection flag
1
# reference field flag
1
# variable year code
19502000
# variable month code
0106
```

Example file 2.2.29: `constandrefe` *file content.*

and the `driver` as:

```
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
isoptimise 0 nothing, 1 L, 2 SN, 3 both,  negative values filter vertically
0
Minimal L
1.
Maximal L
10.
Minimal SN
0.1
Maximal SN
10.0
analysis 1 do it 0 do nothing
1
lowerlevel number
1
upperlevel number
25
reference (to come)
0
isplot 0 or 1
1
number of groups for data detrending, 0 if no detrending.
0
```

Example file 2.2.30: `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.
- `Salinity.19502000.0101.10001.10025.fieldgher.anl`: Salinity 3D binary GHER format file
- `Salinity.19502000.0202.10001.10025.fieldgher.anl`: Salinity 3D binary GHER format file

- `Salinity.19502000.0303.10001.10025.fieldgher.anl`: Salinity 3D binary GHER format file
- `Salinity.19502000.0404.10001.10025.fieldgher.anl`: Salinity 3D binary GHER format file
- `Salinity.19502000.0505.10001.10025.fieldgher.anl`: Salinity 3D binary GHER format file
- `Salinity.19502000.0606.10001.10025.fieldgher.anl`: Salinity 3D binary GHER format file
- `Salinity.19502000.0101.10001.10025.anl.nc`: Salinity 3D netcdf file
- `Salinity.19502000.0202.10001.10025.anl.nc`: Salinity 3D netcdf file
- `Salinity.19502000.0303.10001.10025.anl.nc`: Salinity 3D netcdf file
- `Salinity.19502000.0404.10001.10025.anl.nc`: Salinity 3D netcdf file
- `Salinity.19502000.0505.10001.10025.anl.nc`: Salinity 3D netcdf file
- `Salinity.19502000.0606.10001.10025.anl.nc`: Salinity 3D netcdf file
- `Salinity.19502000.4Danl.nc`: 4D Salinity analyses netcdf file
- `Temperature.19502000.0101.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19502000.0202.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19502000.0303.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19502000.0404.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19502000.0505.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19502000.0606.10001.10025.fieldgher.anl`: Temperature 3D binary GHER format file
- `Temperature.19502000.0101.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19502000.0202.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19502000.0303.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19502000.0404.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19502000.0505.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19502000.0606.10001.10025.anl.nc`: Temperature 3D netcdf file
- `Temperature.19502000.4Danl.nc`: 4D Temperature analyses netcdf file