

# Diva workshop 2014

## Diva in 4 dimensions (GODIVA)

Alexander Barth, Aida Alvera-Azcárate, Mohamed Ouberdous,  
Charles Troupin, Sylvain Watelet & Jean-Marie Beckers

**Acknowledgements:** SeaDataNet, EMODnet Chemistry,  
EMODnet Biology, STARESO



# Outline

See Tutorial on installation ([pdf](#))

# Outline

# Diva input info files

In `input` directory:

- Edit info files and adapt them to your case by providing in the relevant information

File name	content
<code>contour.depth</code>	list file of all depths in meters
<code>NCDFinfo</code>	metadata information for climatology NetCDF files
<code>general_info</code>	information for metadata XML files generation

# Outline

# Data extraction: input files preparation

In `Climatology` directory:

- `datasource` file: list of paths to `ODV4` spreadsheet(s) from which data sets will be extracted.
- `varlist`, `yearlist` and `monthlist` files.
- `qflist` (quality flags) file if desired.

<code>varlist</code>	<code>yearlist</code>	<code>monthlist</code>
Temperature	19002012	0101
Salinity		0202
		0303

# Data extraction: driver configuration & divadoall

In Climatology directory:

- Edit the driver file and put in a flag number for data extraction.

```
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.
0
```

*Figure 1: driver file configuration example.*

- Run divadoall or godiva (basic error check-up included)
- Rem : do not forget to adapt the PATH (for ex. in .bashrc)

A subdirectory divadata is created in input directory, and contains the data sets.



# Outline

# Topography preparation: `gebcomodif`

For a GEBCO topography file, use the script file `gebcomodif` to:

- Eliminate header lines
- Change depth values from negative to positive values
- Change comas to dots in decimal numbers
- Change longitude values from  $[0:360]$  to  $[-180:180]$  range
- Mask rectangle regions by giving coordinates in a `takeout.coord` file

# Topography preparation

In `input`:

- Provide a topography file named `topogebco.asc` extracted from GEBCO Global Elevation Data.

In the `Climatology` directory:

- Provide a `takeout.coord` file:

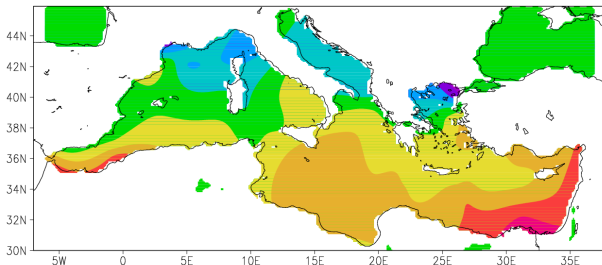
Minlon1	Maxlon1	Minlat1	Maxlat1
Minlon2	Maxlon2	Minlat2	Maxlat2
Minlon3	Maxlon3	Minlat3	Maxlat3
.	.	.	.
.	.	.	.

- Run `gebcomodif` script file.

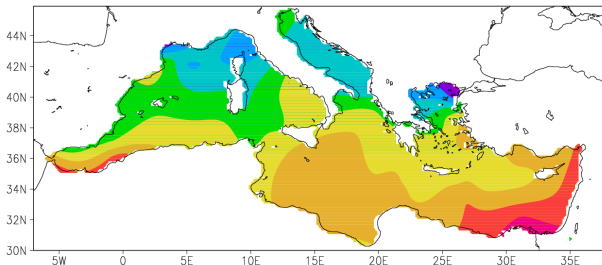
A `topo.gebco` file is generated in `input`.

# Masking regions in topography

Coastlines with full topography



Coastlines with masked topography



# Example of topography preparation

- In `input`, we provide `topogebco.asc` covering the Mediterranean Sea area: 30°N to 46°N and 6°W to 37°E.
- In `Climatology`, we provide a `takeout.coord` file:

```
-6. -1. 42. 46.  
26.5 40. 40. 46.  
5. 9. 33. 35.  
20. 30. 30. 30.5  
35. 37. 31. 33.
```

After running the command `gebcomodif` in `Climatology` directory, we obtain a `topo.gebco` in `input` directory.

- Or you can extract topography from diva-on-web !

# Coastline files generation: input files

In `input` directory provide:

- (a) a `topo.gebco` file
- (b) a `topo.dat` file
- (c) `topo.grd` + `TopoInfo.dat` files
- the `contour.depth` file
- a `param.par` file

OR  
OR

33.966702	35.116699	12.000000
33.983299	35.116699	30.000000
34.000000	35.116699	51.000000
34.016701	35.116699	90.000000
34.033298	35.116699	179.000000
34.049999	35.116699	382.000000
34.066700	35.116699	543.000000
34.083302	35.116699	640.000000
34.099998	35.116699	691.000000

[topo.gebco](#)

[Contour.depth](#)

```
# Correlation Length lc in km or degree??? according to param icoordchange
1
# icoordchange (=0 if position of data in km ; =1 if position of data in degree)
1
# ispec (output files required, comments to come)
11
# lreg (mode selected for background field: 0; 1; 2)
1
# xori (origin of output regular grid, min values of X)
-9.25
# yori (origin of output regular grid, min values of Y)
30
# dx (step of output grid)
.125
# dy (step of output grid)
.125
# rx max x of output grid
367
# ry max y of output grid
129
# valex (exclusion value)
-9999.0
# snr signal to noise ratio (not yet used as such, still set as 4th value of data.dat)
0.5
# varbak variance of the background field
0.6
```

[param.par](#)

3000
2500
2000
1750
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
250
200
150
125
100
75
50
30
20
10
0

# Coastlines files generation: driver configuration

In **Climatology** directory:

- Edit the **driver** file and choose a flag number for boundary lines and coastlines generation:

*Table 1: driver options for coastlines generation*

Comment line	Flag value and corresponding action	
Boundary lines and coastlines generation:	0 :	no action is performed
	1 :	generation of contour files of boundaries and coastlines
	2 :	generation of advection UV files of velocities along coasts
	3 :	generation of contour files and advection UV files

```
Data extraction: 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.
0
```

*Figure 2: driver file configuration example.*

# Coastlines files generation: output

In `Climatology` directory

- Run `divadoall`

A `newinput` directory is created which contains:

- `divaparam`: a subdirectory where coastline files `coast.cont.100xx` are stored
- `divaUVcons_all`: a subdirectory where velocity field files are stored

Copy `divaparam` and `divaUVcons_all`  
to your `input` directory.



# Outline

# Data Cleaning: input files

In `input` directory:

- `divadata`: directory which contains data set files of the considered layers.
- `divaparam`: directory which contains coastline `coast.cont.100xx` files for all considered layers.
- the `contour.depth` file.
- a `param.par` file.

# Data Cleaning: input files

In `Climatology` directory

- Provide `varlist`, `yearlist` and `monthlist` files.
- Edit the `driver` file,
- Choose a flag number for data cleaning and
- give the considered minimum layer and maximum layer numbers.

# Data Cleaning: driver configuration

*Table 2: driver options for data cleaning*

Comment line	Flag value and corresponding action
cleaning data on mesh	0 : no action is performed
	1 : cleaning data out of the mesh
	2 : generation of relative length (RL) fields
	3 : cleaning data out of the mesh and generations of RL fields
	4 : cleaning data set files from outliers
	5 : generations of RL fields and cleaning data set files from outliers

Data extraction: 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.  
 0  
 Parameters estimation and vertical filtering:  
 0  
 Minimal L  
 0.5  
 Maximal L  
 3.  
 Minimal SN  
 0.1  
 Maximal SN  
 50.0  
 Analysis and reference field:  
 0  
 lowerlevel number  
 3  
 upperlevel number  
 25

*Figure 3: driver file configuration example.*

# Data Cleaning: output

In `Climatology` directory:

Run `divadoall`.

A `newinput` directory is created and contains:

- `divadata` subdirectory which contains cleaned data sets
- `divadata` subdirectory which contains relative length files if generated

Copy the content of  
`newinput/divadata` and  
`newinput/divaparam`  
to `input/divadata` and `input/divaparam`  
directories.

# Outline

# Parameters optimisation: input

In `input` directory provide:

- `divadata` directory which contains the data set files of the considered depths.
- `divaparam` directory which contains coastline `coast.cont.100xx` files of the considered basin.
- The `contour.depth` file.
- A (template) `param.par` file.

# Parameters optimisation: input files

In **Climatology** directory:

- Provide **varlist**, **yearlist** and **monthlist** files
- Edit the **driver** file and give a flag number for parameters optimisation and bounds for correlation length ( $L$ ) and signal-to-noise ( $\lambda$ ) parameters.

```
0
Parameters estimation and vertical filtering:
-30
Minimal L
0.5
Maximal L
3.
Minimal SN
0.5
Maximal SN
50.0
Analysis and reference field:
0
```

*Figure 4: driver file configuration example.*



# Parameters optimisation: driver configuration

*Table 3: driver options for parameters optimisation.*

Comment line	Flag value and corresponding action	
Parameters optimisation and vertical filtering	0 :	no action is performed
	1 :	estimation for each level of correlation length $L$ parameter
	2 :	estimation for each level of signal to noise ratio ( $\lambda$ ) parameter
	-1 :	estimation and vertical filtering of $L$ parameter
	-2 :	estimation and vertical filtering of $\lambda$ parameter
	3 :	estimation for each level of $L$ and $\lambda$ parameters
	-3 :	estimation and vertical filtering of $L$ and $\lambda$ parameters
	10 :	estimation of $L$ parameter for each level using data mean distance as a minimum
	-10 :	estimation of $L$ parameter using data mean distance as a minimum and vertical filtering
	30 :	estimation of $\lambda$ and $L$ parameters for each level, using data mean distance as a minimum for $L$
	-30 :	estimation and vertical filtering of $\lambda$ and $L$ parameters, using data mean distance as a minimum for $L$ ,

# Parameters optimisation: output

In `Climatology` directory:

- Run the `divadoall` script file.

A `newinput` directory is created and contains:

`divaparam` subdirectory with `param.par.100xx` files and summary files of the optimisation and filtering procedure.

Copy the content of `newinput/divaparam`  
to `input/divaparam` directory

# Outline

# Producing a Climatology: input

In `input` directory:

- `divadata` directory which contains data sets for the considered layers,
- `divaparam` directory which contains:
  - coastlines `coast.cont.100xx` files,
  - coastlines `param.par.100XX` files.
- the `contour.depth` file,
- a `param.par` file if not provided in `divaparam`

# Producing a Climatology: input & and driver

In `Climatology` directory:

- Provide

`varlist`,  
`yearlist` and  
`monthlist` files.

- Edit the `driver` file and choose a flag number for analysis.

```
Analysis and reference field:
1
lowerlevel number
5
upperlevel number
25
4D netcdf and Metadata XML metadata files generation:
1
gnuplot plots: 0 or 1
0
Data detrending: number of groups, 0 if no detrending.
0
```

*Figure 5: `driver` file configuration example.*

# Producing a Climatology: input & and driver

In `Climatology` directory:

*Table 4: `driver` options analyses & climatologies production.*

Comment line	Flag value and corresponding action
Analysis and reference fields	0 : no action is performed
	1 : Perform analyses defined by a set of input files: <code>varlist</code> , <code>yearlist</code> , <code>monthlist</code> , <code>constandrefe</code> and the files in <code>input/</code> directory
	2 : generation of reference field
	3 : perform analyses as in 1 based on vertically filtered background
	11 : perform analyses using a $\log(\text{data})\text{-exp}(\text{analysis})$ transformations
	13 : perform analyses using the anamorphosis transformation
	14 : perform analyses using a user defined transformation
	21 : perform reference fields using a $\log(\text{data})\text{-exp}(\text{analysis})$ transformations
	23 : perform reference fields using the anamorphosis transformation
	24 : perform reference fields using user defined transformation
	<u>Adding 100 to flag values 1, 11, 13 and 14 allows to perform the same action using a reference field for each layer generated on the basis of all data from the two neighbouring layers in addition to the layer data set.</u>
	<u>Adding 100 to flag values 2, 21, 23 and 24 allows to perform reference fields with the same action using all data from the two neighbouring layers in addition to the layer data set</u>

Run `divadoall` script file.

# Producing a Climatology: output

An `output/3Danalysis` directory  
is created and contains:

- The 4D climatology NetCDF file:

`Temperature.19002010.4Danl.nc`

- subdirectories:

`Fields`: contains all Diva analyses 2D-fields

`Meshes`: contains depths meshes for each layer

- 3D NetCDF and binary (GHER format) files:

`Temperature.19002010.nnmm.100xx.100yy.anl.nc`

`Temperature.19002010.nnmm.100xx.100yy.fieldgher.anl`

- + 4D netcdf files (`Temperature.4Danl.nc`) if netcdf flag =  
11 or -11 !

# Outline



# Production of a Climatology using advection fields

In `input` directory provide:

- `divadata` directory (data sets)
- `divaparam` directory  
(`coast.cont.100xx` and `param.par.100xx` files)
- `divaUVcons_all` directory which contains velocity fields:  
(GHER-format) binary files. (+ see `asctobin`)
- the `contour.depth`
- a `param.par` if not provided in `divaparam`

In `input/divaUVcons_all` provide

- `constraint.dat` (one line) file.

10	0
----	---

example: `constraint.dat` file

# Production of a Climatology using advection fields

In `Climatology` directory:

- provide a `constandrefe` file:

*Table 5: Example of `constandrefe` file.*

```
# advection flag  
1  
# reference field flag  
0  
# variable year code  
00000000  
# variable month code  
0000
```

- Provide `varlist`, `yearlist` and `monthlist` files.
- Edit the `driver` file and choose a flag number for analysis.
- Execute `divadoall`.

# Outline

# Data extraction for reference field

In `input` directory:

- the `contour.depth` file

In `Climatology` directory provide:

- `datasource` file (ODV4 spreadsheet(s) path)
- `varlist`, `yearlist` and `monthlist` files

<code>varlist</code>	<code>yearlist</code>	<code>monthlist</code>
Temperature	19002010	0103

- `qflist` file if desired
- Edit the `driver` file and choose a flag number for data extraction
- Run `divadoall` script file.

The variable(s) data set files are stored in `input/divadata` directory

# Production reference fields: inputs

In `input` directory:

- `divadata` directory (data sets)
- `divaparam` directory  
(`coast.cont.100xx` and `param.par.100xx` files)
- the `contour.depth`
- a `param.par` if not provided in `divaparam` with value equal to zero for `ireg` (`ireg= 0`)

In `Climatology` directory:

- Provide `varlist`, `yearlist` and `monthlist` files.
- Edit the `driver` and choose flag value 1 for data cleaning.
- and flag value 2, 21, 23 or 24 for analysis.
- Run `divadoall` script file.

## Production reference fields: output

A `newinput` directory is created and contains:

- `divarefe` subdirectory which contains reference fields (Diva 2D binary files) in GHER-format.

In `output/3Danalysis` directory:

- `Fields`: contains all Diva analyses 2D-fields.
- 3D NetCDF files:

`Temperature.19002010.0103.100xx.100yy.ref.nc`

- Binary 3D files (GHER-format):

`Temperature.19002010.0103.100xx.100yy.fieldgher.ref`

Copy the content of `newinput/divarefe` to  
`input/divarefe_all`

# Producing Climatology using reference fields

In `input` directory:

- `divadata` directory (data sets)
- `divaparam` (`coast.cont.100xx` and `param.par.100xx`)
- `divarefe_all` directory which contains reference fields
- the `contour.depth` file.

In `Climatology` directory:

- `constandrefe` file:

```
# advection flag
0
# reference field flag
1
# variable year code
19002010
# variable month code
0103
```

# Using reference fields

In `Climatology` directory:

- `varlist`, `yearlist` and `monthlist` files
- Edit `driver` file and choose a flag number for analysis.
- Run `divadoall` script file.

Results will be stored in `output/3Danalysis` directory.



# Outline

# Detrending

In `input` directory provide:

- `divadata` directory where data set files have more than five columns (5th, 6th, ... contain the information in which class the data point belongs)
- same other inputs as for normal run

In `Climatology` directory  
provide the usual input text files and:

- Edit the `driver` file and
- choose a flag number for detrending a value (less or equal to the number of groups) present in your data set

Run `divadoall` script file.

- Results will be stored in

`output/3Danalysis` directory

# To go further...

- Result layers are *stacked* together



## To go further...

- Result layers are *stacked* together
- Problems may occur between two levels...

# To go further...

- Result layers are *stacked* together
- Problems may occur between two levels...
- ...so stabilisation is required

