DIVADEMECUM

Input	Action `Execution`	Output
`mycase/input/*`	load new case `divaload mycase`	`./input/*`
`topo.dat` `param.par`	make gridded topography `divatopo [-r]`	`TopoInfo.dat [./input/TopoInfo.dat]` `topo.grd [./input/topo.grd ]`
`topo.asc`	use dbdb topography `dbdb2diva [-r]`	`TopoInfo.dat [./input/TopoInfo.dat]` `topo.grd [./input/topo.grd ]`

`TopoInfo.dat`
`topo.grd`
`[contour.depth]`
	make contours
	`divacont [-r]`
		`coast.cont.* [./input/coast.cont.*]`

`param.par` `coast.coa`	use ODV contours `divacoa2cont [-r]`	`coast.cont [./input/coast.cont]`
`param.par` `coast.cont`	check hand-made contours `divacck [-r] [-v]`	`coast.cont.checked [./input/coast.cont]`
`../*/fort.*`	clean up directories `divaclean`	
`data.dat` `coast.cont`	eliminate useless data `divadataclean [`f_{min} f_{max}`]`	`./input/data.dat`

`param.par`
`data.dat`
	estimate L and S/N
	`divafit [n] [-r]`
		`covariance.dat`
		`covariancefit.dat`
		`paramfit.dat`
		`param.par.fit [./input/param.par]`

`param.par`
`coast.cont`
`[coast.cont.dens]`
	make FE mesh
	`divamesh`
		`divamesh outputs`

`gcvsampling.dat`
`param.par`
`data.dat`
`divamesh outputs`
`[Uvel.dat,Vvel.dat`
`UVinfo.dat, constraint.dat]`
`[RL.dat, RLinfo.dat]`
	optimise S/R by cross-validation
	`divacv [-r]`
	`divacvrand ns nt [-r]`
	`divagcv [-r]`
		`gcv.dat`
		`gcvsnvar.dat`
		`gcvval.dat`
		`param.par.gcv[./input/param.par]`

`param.par`
`data.dat`
`divamesh outputs`
`[Uvel.dat,Vvel.dat`
`UVinfo.dat, constraint.dat]`
`[RL.dat, RLinfo.dat]`
`[valatxy.coord]`
	make analysis
	`divacalc`
		`GridInfo.dat`
		`field*.anl`
		`error*.anl`
		`*.nc`
		`[valatxyascii.anl]`

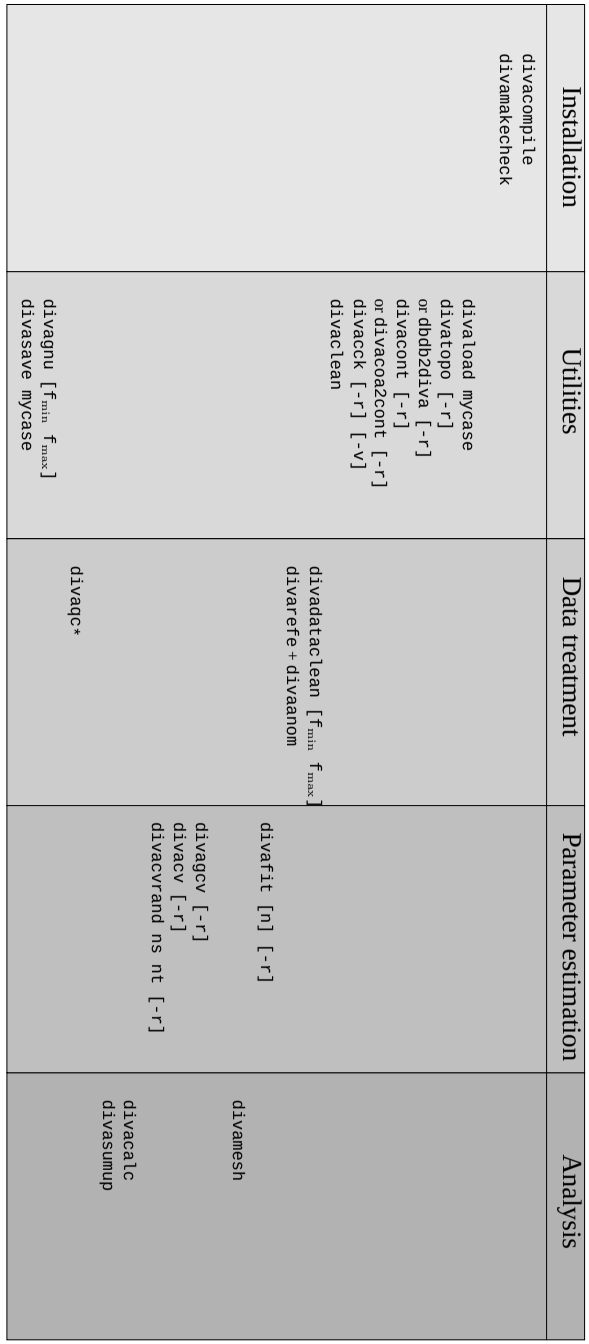
`param.par`
`data.dat`
`./output/meshvisu/*`
`[Uvel.dat,Vvel.dat`
`UVinfo.dat, constraint.dat]`
`[RL.dat, RLinfo.dat]`
	perform full quality control
	`divaqc`
		`outliers.dat`
		`outliers.normalized.dat`

`param.par`
`data.dat`
`divacalc outputs`
	perform simple quality control
	`divaqcbis`
		`outliersbis.dat`
		`outliersbis.normalized.dat`

`param.par`
`data.dat`
`divacalc outputs`
	perform simple quality control
	`divaqcter`
		`outlierster.dat`
		`outlierster.normalized.dat`

`./output/*`	make some plots `divagnu [`f_{min} f_{max}`]`	`./gnuwork/plots/*`
`./output/*`	save results `divasave mycase`	`mycase/outut/*`

Table 1: DIVA in- and outputs. When not specified differently, input files are from directory `./input` and output files are placed in directory `./output`. Script `divarefe` takes the same inputs as `divacalc` while `divaanom` and`divasumup` use no other user-provided files than the other scripts. Brackets `[ ]` enclose optional files or parameters. Ex. `[-r]` will replace an input file by the outputs from the scrips.



Minimal execution to get an analysis: `divamesh+divacalc`.

`divaseminorm` = `divarefe+divaanom+divacalc+divasumup`.

`divadress` = `divaclean+divamesh+divacalc+divaqcbis`.

Figure 1: Scripts used in the command-line version of DIVA; optional arguments are between `[]`.

Input files

```
# Correlation length (in units of data, if degrees: S-N)
1
# icoordchange (-xscale, 0=none, 1=degtokm, 2=sin projection)
0
# ispec (error output files required)
7
# ireg (subtraction of reference field 0: no, 1:mean, 2:plane)
0
# xori (origin of output regular grid, min values of x)
-4.999
# yori (origin of output regular grid, min values of y)
-4.999
# dx (x-step of output grid)
0.1999
# dy (y-step of output grid)
0.1999
# nx number of x points of output grid
51
# ny number of y of output grid
51
# valex (exclusion value)
-9999.0
# snr signal to noise ratio
10
# varbak variance of the background field
1
```

`param.par` file content. Parameters are self-explaining, except for error output specification. `ispec=0` means no error field requested; add +1 for a gridded error field, +2 for error at data location and +4 for error at coordinates defined in `valatxy.coord`. From there if you want

- error based on real covariance: `ispec` ← `-ispec`
- error based on real covariance with boundary effect: `ispec` ← `-ispec-10`
- poor man's error estimate (quick and underestimated error field): `ispec` ← `ispec+10`

(ex: `ispec=12` makes a poor mans error estimate at data locations)

```
-5  22 34.8
-2  19 36.1
...
```

`data.dat` file content. Simple ascii file with `x,y,val` and optional fourth column containing the relative weight on the data (large value = high confidence). `topo.dat` is just a special case where the third column represents depth (positive for sea values).

```
100 10
```

`constraint.dat` file content. First value= weight on advection constraint, second value=diffusion coefficient in advection/diffusion equation.

```
0.01
0.03
0.1
0.3
1
3
10
30
100
```

`gcvsampling.dat` file content. A list of trial values for the signal/noise ratio used in cross validation tools.

```
2000
1500
1000
800
700
600
500
400
300
200
100
0
```

`contour.depth` file content with depth for contours and subsequent analysis.

```
0
10
0.1
0.2
101
51
```

`*info.dat` file desribing the gridding parameters of binary gridded files such as `Uvel.dat`, `topo.grd`, `RL.dat`, `fieldgher.anl`, `errorfieldgher.anl`. Here first grid point in (0,10), with steps (0.1,0.2) and 101x51 grid points. Look at examples how to read/write binary files with Fortran or Matlab

Output files

```
 0.50E+01 1131 0.16E+02 0.43E+02 0.38E+02 0.37E+02 0.13E+00
 ...
  flag    ident   x   y   dataval  analysis expected-misfit
```

`outliers*.dat` Sorted ouliers, from most suspect to least suspect. Column 1: outlier indicator (larger than 3 suspect), following columns: data identifier, x and y coordinates, original data value, analysed data value, expected misfit.

```
Correlation length (in degrees latitude)
  3.69890785
 Signal to noise ratio
  0.902823746
 VARBAK
  16.7839489
 For information: correlation length in km is  412.962524
```

`paramfit.dat` Self explaining output from `divafit`. When option `[-r]` is used with `divafit`, an adapted `param.par` will be placed in `./input`.

```
 S/N
  1.99764168
 VARBAK
  1.14215052
```

`gcvsnvar.dat` Self explaining output from `divacv, divagc, divacvrand`. When option `[-r]` is used with cross validation, an adapted `param.par` will be placed in `./input`.