pydiva2d: a python interface to the DIVA interpolation software

Spatial interpolation | Oceanography | Data analysis | Finite-element method | python | SeaDataCloud | Python | Leaflet

1 What is DIVA?

A scientific tool designed to efficiently interpolate oceanographic observations.
The finite-element technique employed ensures that we can work with very big data sets.
The code itself consists of bash scripts that call a set of Fortran executables.

3 reasons to use it

- ✓ Free & open
- ✓ Deal with huge oceanographic datasets (> 1000000 points)
- ✓ Take physical boundaries (coastline, topography) into account

Table 1: How to get DIVA tool?

Zenodo	DOI 10.5281/zenodo.836727
	https://github.com/gher-ulg/DIVA

2 Why a python interface?

There are many input files to prepare prior to an analysis with DIVA, so the interface makes things easier for the new users: just select the data files and the analysis parameters, the run the code without worrying about the file names and formats.

Table 2: How to get the module?

Zenodo	DOI 10.5281/zenodo.838193
	https://github.com/gher-ulg/DivaPythonTools

What's inside the box?

`pydiva2d` defines *classes* representing the main DIVA objects:

data consisting of geolocalised measurements,
contours representing the physical domain (coastlines),
parameters specifying how the interpolation is performed,
mesh used in the solver
analysis the result produced by the interpolation.

3 An example in the Black Sea

We interpolate mixed-layer depth values obtained from in situ profiles in the Black Sea. The different steps for the interpolation are as follows: first we load the package and define the DIVA files:

```
import pydiva2d
divadir = "/home/ctroupin/diva-4.7.1/"
DivaDirs = pydiva2d.DivaDirectories(divadir)
DivaFiles = pydiva2d.Diva2Dfiles(DivaDirs.diva2d)

datafile = 'MLD1.dat'
coastfile = 'coast.cont'
paramfile = 'param.par'
```

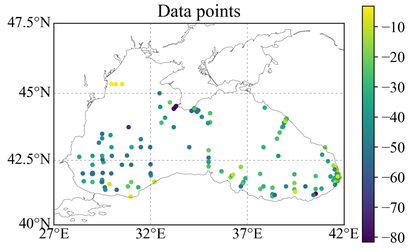

Figure 1: Data points representing the Mixed-layer depth (in meters). Some regions are well covered, while others have no measurements.

```
# Load parameters
param = pydiva2d.Diva2DParameters().
                        read_from(paramfile)
# Create projection (basemap)
m = Basemap(projection='merc', llcrnrlon
                        =param.xori,
                        llcrnrlat=param.yori
                        , urcrnrlon=param.
                        xend, urcrnrlat=
                        param.yend,
                        resolution='i')

# Load data
data = pydiva2d.Diva2DData()
data.read_from(datafile)
# Make a plot
fig = plt.figure()
ax = plt.subplot(111)
m.ax = ax
dataplot = data.add_to_plot(m=m, s=10)
```

```
# Load the contours from file
contour = pydiva2d.Diva2DContours()
contour.read_from(coastfile)
# Add to plot
contour.add_to_plot(m=m, linewidth=3)
```

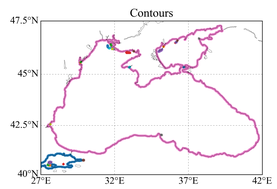

Figure 2: Surface contours created from the bathymetry. They delimit the interpolation domain, on which the finine-element mesh will be built.
In this application there are 28 contours, the main contour being the Black Sea.

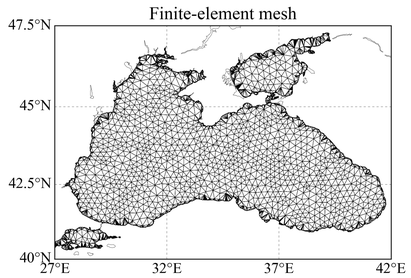

Figure 3: Triangular mesh used in the numerical solving of the interpolation.
Number of nodes: 4636.
Number of interfaces: 10089.
Number of elements: 5360.

```
# Read mesh files
mesh = pydiva2d.Diva2DMesh()
mesh = read_from(DivaFiles.mesh,
                        DivaFiles.meshtopo)
# Get a description
mesh.describe()
# Add to plot
mesh.add_to_plot(m, linewidth=0.25,
                        color='k')
```

```
# Perform the analysis using the
                        specified files
results = pydiva2d.Diva2DResults().make(
                        divadir, datafile=
                        datafile,
                        contourfile=
                        coastfile, paramfile
                        =paramfile)
# Add results to plot
resultplot = results.add_to_plot(field='
                        analysis', m=m)
```

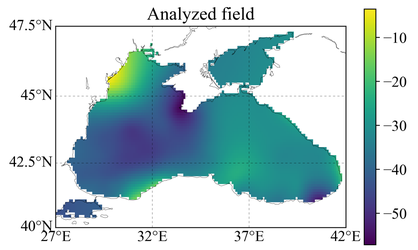

Figure 4: Interpolated mixed-layer depth (in meters). The shallowest values (yellow color) are found in the north-eastern part of the domain, influenced by the Danube river runoff.

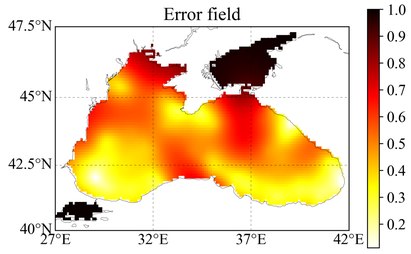

Figure 5: As expected, the error field displays lower values where the data coverage is higher.
Note the large error in the Seas of Azov and Marmara.

```
# Add the error field
errorplot = results.add_to_plot(field='
                        error', m=m, cmap=
                        plt.cm.hot_r)
```

4 Using Leaflet library

Thanks to the format-conversion methods available in the `pydiva2d`, one can easily generate GeoJSON files (http://geojson.org/), directly ingestible in Leaflet.

Leaflet?

Leaflet is a library for mobile-friendly interactive maps (http://leafletjs.com/). It comes with a bunch of plugins to create customized maps with a lot of information as layers.

Temperature in the MedSea

The following figures illustrate the DIVA input and output.

```
# Write contour and mesh to geoJSON
contour.to_geojson('medsea-contours.js')
mesh.to_geojson('medsea-mesh.js')
```

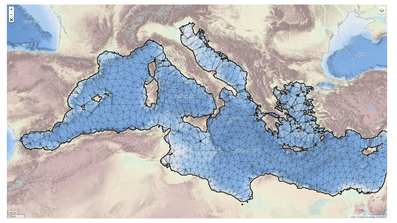

Figure 6: Contours and finite-element mesh. Both have the MultiPolygons geometry.
Bathymetry from http://www.emodnet-bathymetry.eu/.

```
# Write data to geoJSON
data.to_geojson('medsea-data.js')
```

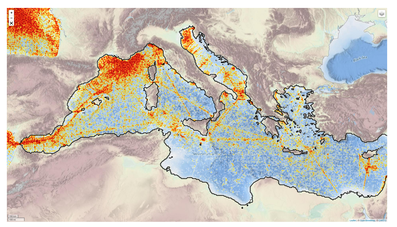

Figure 7: Heatmap displaying the data density.
As in the Black Sea example, the distribution is heterogeneous.
Data source: Mediterranean Sea – Temperature and salinity observation collection V2 doi:10.12770/8c3bd19b-9687-429c-a232-48b10478581c

```
# Write field to geoJSON
results.to_geojson('medsea-results.js')
```

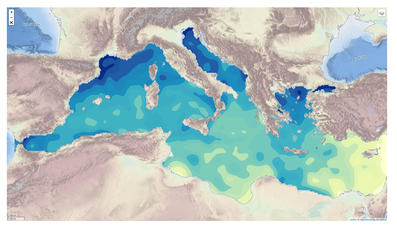

Figure 8: Analysis field of temperature.
The isotherms are stored in a FeatureCollection, each feature being a MultiPolygon with the temperature value as a property.

Summary

- ✓ DIVA is a software tool for data interpolation.
- ✓ the module `pydiva2d` helps with:
  the manipulation of input files,
  the generation of figures and
  the conversion to other formats.
- ✓ Leaflet can be used to represented the DIVA input and outputs using GeoJSON format.

Acknowledgements

The DIVA development has received funding from the European Union Sixth Framework Programme (FP6/2002–2006) under grant agreement no. 026212, SeaDataNet, Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 283607, SeaDataNet II, SeaDataCloud and EMODnet (MARE/2008/03 - Lot 3 Chemistry - SI2.531432) from the Directorate-General for Maritime Affairs and Fisheries.

Authors: Charles Troupin, Alexander Barth, Sylvain Watelet and Jean-Marie Beckers