pydiva2d: a python interface to the DIVA interpolation software

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>
<thead>
<tr>
<th>Table 1: How to get DIVA tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zenodo 851</td>
</tr>
</tbody>
</table>

What’s inside the box?
pydiva2d defines classes representing the main DIVA objects:
- `data` consisting of geolocated 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.

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>
<thead>
<tr>
<th>Table 2: How to get the module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zenodo 681</td>
</tr>
</tbody>
</table>

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:

```python
# load pydiva2d
diva = pydiva2d.Diva2DParameters()
# read from files
data, mesh = read_from('datafile.dat', 'meshfile.dat')

# create pyprojection (coordinates)
proj = pyprojection.Spherical()  # instance of the projection
m = proj.project(data, data)  # project the data

# create error
errorfield = pyprojectionErrorField(m)  # create the error field from the projection

# add the error to the plot
plt.plot(errorfield, label='errorfield')
```

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 represent the DIVA input and outputs using GeoJSON format.

Acknowledgements

Leaflet?
Leaflet.js 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.

- **Figure 3**: Surface contours created from the bathymetry. They delimit the interpolation domain, so which the finite-element mesh will be built.
- **Figure 4**: Error field displays lower values where the data coverage is higher. The large error in the Sea of Azov and Marmara is influenced by the Danube river runoff.
- **Figure 5**: Analyzed field. The shallower 'results' are displayed in yellow color. They are found in the north-eastern part of the domain, influenced by the Danube river runoff.
- **Figure 6**: Bathymetry from MultiPolygons finite-element mesh. Both have the same geometrical distribution.
- **Figure 7**: Median bathymetry. The coastlines are stored in a GeoJSON file, each feature being a MultiPolygon with the temperature value as a property.

Authors: Charles Troupin, Alexander Barth, Sylvain Watelet and Jean-Marc Becker