

Analyzing ocean observations in a HPC infrastructure with DIVAnd

Alexander Barth, Charles Troupin University of Liège





The PHIDIAS project has received funding from the European Union's Connecting Europe Facility under grant agreement n° INEA/CEF/ICT/A2018/1810854.





Image creation: Center for Environmental Visualization, University of Washington

- Many ocean processes are present simultaneously
- Non-linear
 Interaction between them
- Wide time/space spectrum of scales
- → High diversity of ocean observations





Image credits: ICTS SOCIB

The types of observations observations is quite diverse Ocean observations are sparse (because expensive) Yet scientifically very valuable (a measurement not taken it lost forever, the state of the climate and ocean in particular changes)



Fast access to data, multitude of formats, general trend towards netCDF

Different programming environments/languages used by scientists:

- Fortran (still used in numerical models)
- Matlab (very widespread ~10 years ago, but less use today)
- Python
- R

But also Julia, C, C++, shell scripts,...



- At GHER, ULiège: started to use Julia to use in 2017
- Julia version 1.0 was released on 8 August 2018

nature > toolbox > article	a naturerese				
MENU V International journal of science		Subscribe	Search	Login	

TOOLBOX · 30 JULY 2019

Julia: come for the syntax, stay for the speed

Researchers often find themselves coding algorithms in one programming language, only to have to rewrite them in a faster one. An up-and-coming language could be the answer.

Jeffrey M. Perkel



- DIVA: <u>Data Interpolating Variational</u> <u>Analysis</u>
- Objective: derive a gridded climatology from in situ observations
- The variational inverse methods aim to derive a continuous field which is:
 - close to the observations (it should not necessarily pass through all observations because observations have errors)
 - "smooth"
- Spline interpolation









SeaDataNet





Jupyter



- Workshops
- Virtual Research Environment (VRE) in SeaDataCloud
- Jupyter Notebooks
- CI (Continuous Integration) testing (Linux, Mac OS, Windows)
- Docker and Singularity images with preconfigured software



HIDIAS DIVAND in a virtual research environment https://vre.seadatanet.org/

					VRE - Mo	zilla Firefox			-		×
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	Hi <u>s</u> tory	<u>B</u> ookmarks	<u>T</u> ools	<u>H</u> elp					
VRE			×	+							
(ϵ)	→ Cª	۵	(i) 🔒	https://orca. d	krz.de				⊻	»	≡^
_	i.c.	eaData	Net	PAN-EUROPE OCEAN & MAI	AN INFRAST RINE DATA M	RUCTURE FOR MANAGEMENT	a.	barth@ulç	g.ac.be	•	
	T&S L	ab									
				•)			0	+		
		Private		webOD	V	webODV		web0D\	/		
	v	workspac	ce	import		data extractor	q	uality cont	trol		
	D	IVA Jupy Noteboo	rter	DIVA GU		VIZ					

				15-examp	ole-analy	ysis - Mozilla Firefox		-		×
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	Hi <u>s</u> tory	<u>B</u> ookmarks	<u>T</u> ools	<u>H</u> elp				
₹ 15	-examp	le-analy	sis X	+						
¢	→ C	ŵ	i loc	alhost:8906/nc	tebooks	/15-example-analysis.ipy	⊠ ☆	⊻	»	≡
C J	upyte	er 15	-example	e-analysis (#	utosaved)				Log	gout
File	Edit	View	/ Insert	Cell Ken	nel He	əlp	Not Trusted	Ju	ilia 1.1	.0 🔵
8	+ 🔀	2	• •	N Run	C H	Markdown 🗾 📼				

DIVAnd analysis using the sample data set

This example performs a salinity analysis using data from the Black Sea. The analysis is done for every season and year (using all data with 10-year sliding windows form the same season).

For testing purposes, let's start with a low resolution.

A slightly large test case:

- · horizontal resolution of 0.1 degree for the Black Sea
- 51 depths levels
- 8 time instance
- fixed correlation length
- CPU time: 21 minutes
- CPU time increases linearly with the number of time instance.









- DIVAnd needs to solve a large matrix system
- The solvers:
 - direct solver (SuiteSparse, Cholmod) requiring a significant amount of memory but a very fast
 - iterative solvers (preconditioned conjugate gradient) are more memory efficient but slower
- In practice: the direct solver is preferred as long as the problems fits into the available memory
- But having access to computing resources with sufficient memory has been a problem for our users (SeaDataCloud, EMODnet Chemistry)
- Code portability via Singularity container



- Paper: <u>Data INterpolating Convolutional Auto-Encoder</u>
- Neural network to reconstruct missing data in satellite images (in particular clouds in remotely sensed Sea Surface Temperature)
- Originally written in Python using TensorFlow 1
- Many changes in TensorFlow 2 -> better alternatives?
- Use Julia and with the Knet library
- Training time of the network was reduced from 3.5 hours to 1.9 hours (on a NVidia 1080 GPU)
- We use "data augmentation" (in particular perturbing input data, add additional clouds,...) using vectorized numpy code, but it could be made significantly faster by using Julia instead.



- Sea Surface
 Temperature (SST)
 reconstruction with
 DINCAE
- Some data is withheld during the reconstruction (i.e. additional clouds)
- SST is reconstructed and a reliable the expected error standard deviation is computed







DINCAE reconstruction using MODIS sea surface temperature in the Adriatic



- The types of available ocean data is quite diverse
- Fortran is still widely used in the oceanographic HPC community
 - But there are significant challenges to support users outside of a typical HPC environment
 - Julia has been a good fit for us for data analysis
- The original Fortran tool DIVA has been rewritten in Julia (DIVAnd)
- Jupyter notebooks provide the users a convenient interface that can also be used in a Virtual Research Environment (especially for data exploration)
- In future: adapt existing tools or adopt new algorithm able to leverage GPUs (or other accelerators)





