

Post-processing and diagnosing ICON data

Nils Brüggemann^{1,2}, Wolfgang Müller¹, Peter Korn¹

¹Max Planck Institute for Meteorology, ²University of Hamburg



Max-Planck-Institut
für Meteorologie

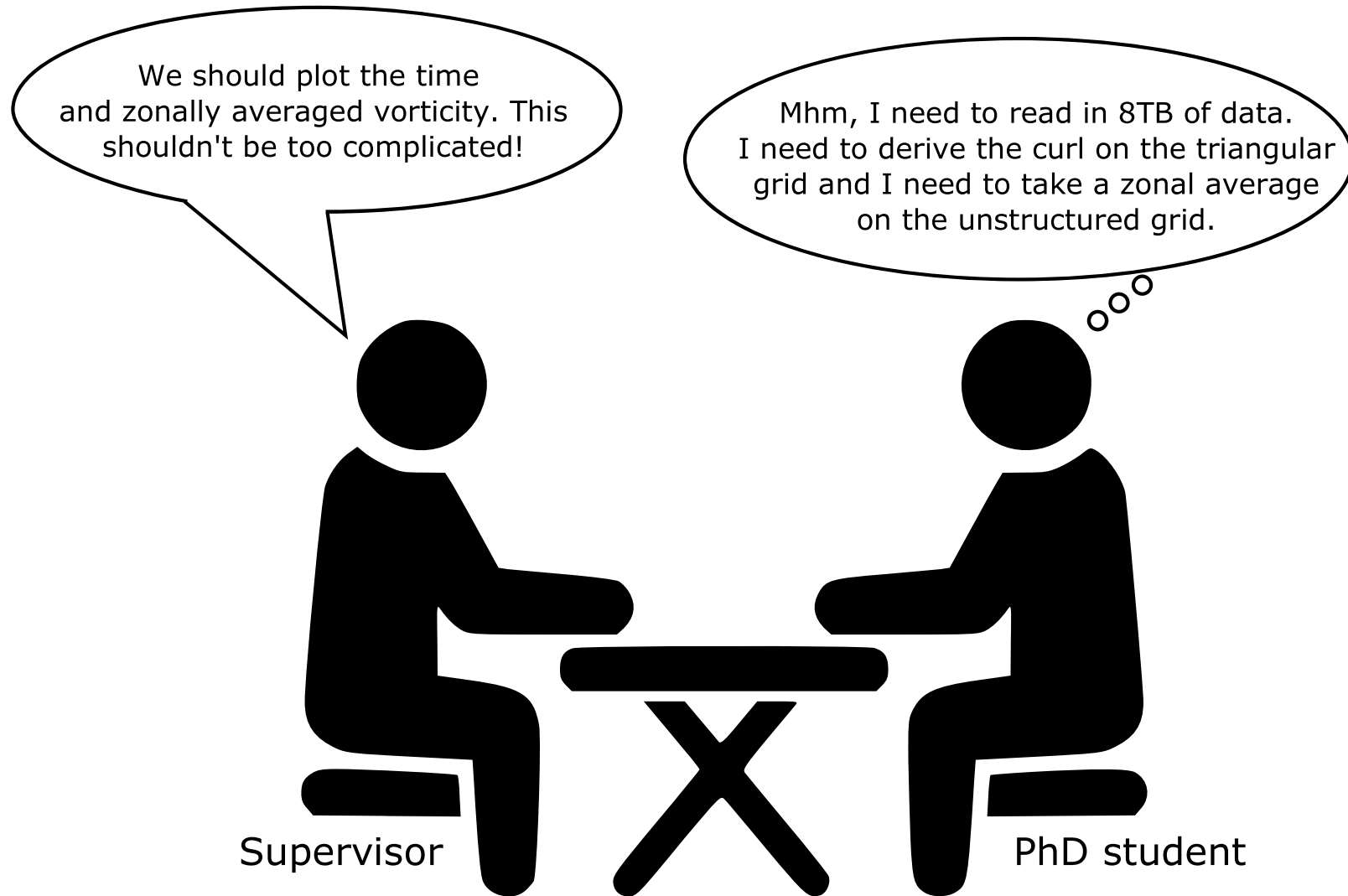


Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

The challenge

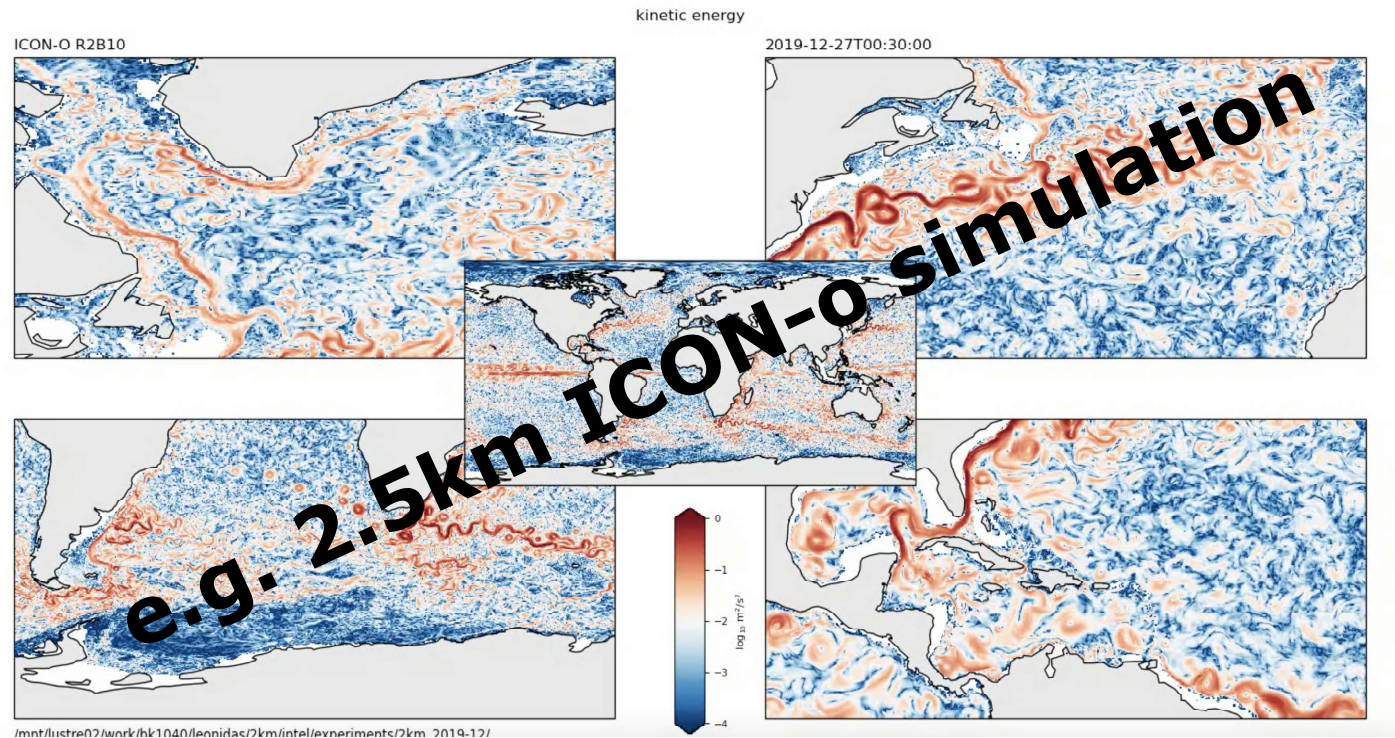
1



What we have

2

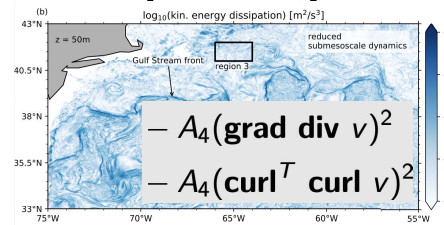
- ▶ many different ICON simulations – producing huge data sets



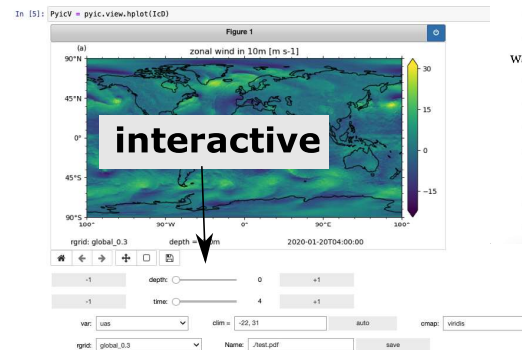
What we have

- ▶ many different ICON simulations – producing huge data sets
- ▶ various post-processings tools, developed by domain programmers at different institutions for different purposes – not all HPC-ready

1. Complex diagnostics



2. Sophisticated visualizations



3. Monitoring simulations

public website
 r2b6_idemix
 m300602 | 2021-12-21 15:38:12 | Compare: nibo033, nibo037, nibo038, nibo041, nibo042, nibo043
 list comparisons
Water masses
 Temp. diff. depths
 Temp. glob. zon. ave.
 Temp. Atl. zon. ave.
 Temp. Indo.-Pac. zon. ave.
 Salt. diff. depths
 Salt. glob. zon. ave.
 Salt. Atl. zon. ave.
 Temp. glob. zon. ave.

many diagnostics

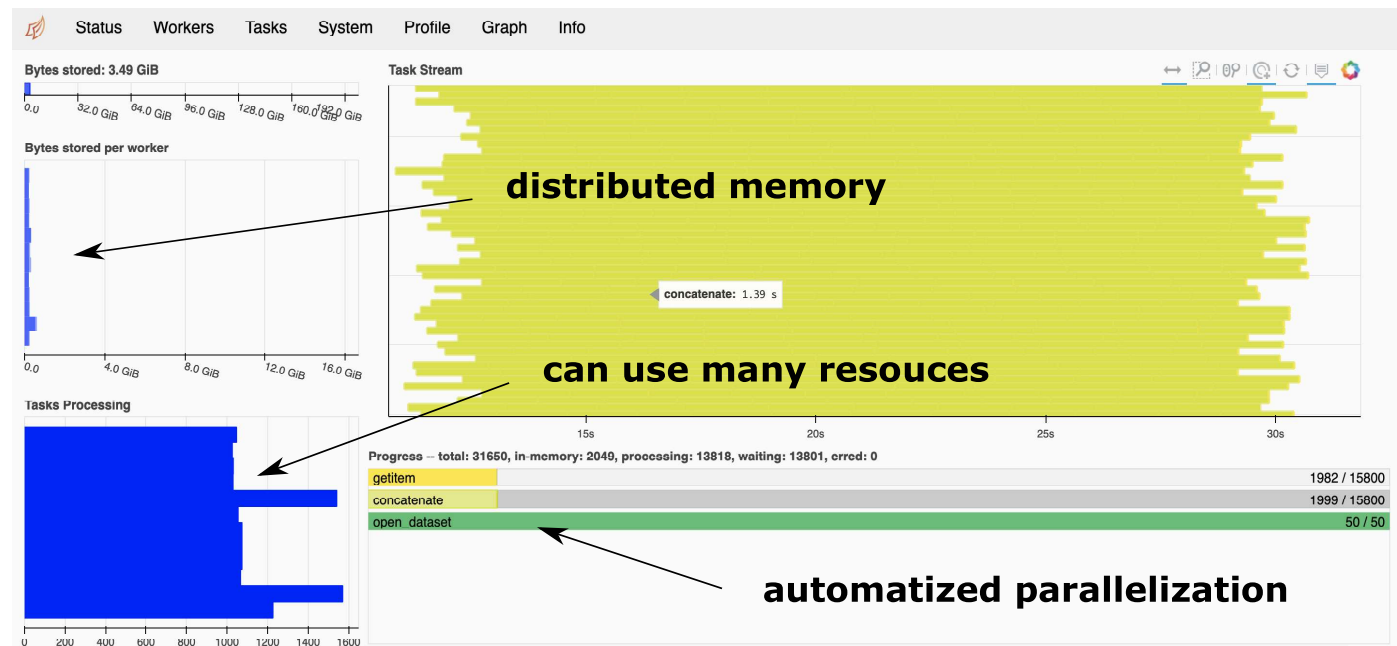
many simulations

What we have

2

- ▶ many different ICON simulations – producing huge data sets
- ▶ various post-processings tools, developed by domain programmers at different institutions for different purposes – not all HPC-ready
- ▶ some ideas how to make our post-processing tools HPC-ready

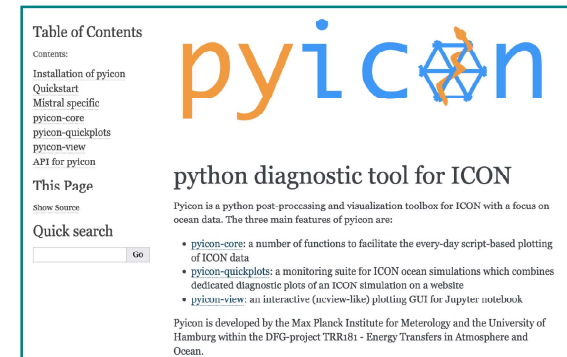
The advantages of Python / Dask



What we need

We need support for

- ▶ improving usability of our post-processing tools
- ▶ smarter ways for documentation
- ▶ better designed APIs



The screenshot shows the Pyicon website. On the left is a 'Table of Contents' with links: Contents, Installation of pyicon, Quickstart, Mistral specific, pyicon-core, pyicon-quickplots, pyicon-view, and API for pyicon. Below this is a 'Quick search' field with a 'Go' button. The main content area features the 'pyicon' logo (with a cube icon) and the title 'python diagnostic tool for ICON'. A paragraph describes Pyicon as a python post-processing and visualization toolbox for ICON with a focus on ocean data. Three main features are listed: pyicon-core (every-day script-based plotting), pyicon-quickplots (monitoring suite), and pyicon-view (interactive plotting GUI). At the bottom, it states Pyicon is developed by the Max Planck Institute for Meteorology and the University of Hamburg within the DFG-project TRR181 - Energy Transfers in Atmosphere and Ocean.

What we need

3

We need support for

- ▶ improving usability of our post-processing tools
 - ▶ smarter ways for documentation
 - ▶ better designed APIs
- ▶ improving performance of post-processing
 - ▶ effectively use parallel computing (e.g. with Python/Dask)
 - ▶ better I/O options by new file formats (e.g. Zarr)



Table of Contents

- [Contents](#)
- [Installation of pyicon](#)
- [Quickstart](#)
- [Mistral specific](#)
- [pyicon-core](#)
- [pyicon-quickplots](#)
- [pyicon-view](#)
- [API for pyicon](#)

This Page

Show Source

Quick search

pyicon

python diagnostic tool for ICON

Pyicon is a python post-processing and visualization toolbox for ICON with a focus on ocean data. The three main features of pyicon are:

- **pyicon-core**: a number of functions to facilitate the every-day script-based plotting of ICON data
- **pyicon-quickplots**: a monitoring suite for ICON ocean simulations which combines dedicated diagnostic plots of an ICON simulation on a website
- **pyicon-view**: an interactive (ncview-like) plotting GUI for Jupyter notebook

Pyicon is developed by the Max Planck Institute for Meteorology and the University of Hamburg within the DFG-project TRR181 - Energy Transfers in Atmosphere and Ocean.



python™

xarray

DASK

What we need

3

We need support for

- ▶ improving usability of our post-processing tools
 - ▶ smarter ways for documentation
 - ▶ better designed APIs
- ▶ improving performance of post-processing
 - ▶ effectively use parallel computing (e.g. with Python/Dask)
 - ▶ better I/O options by new file formats (e.g. Zarr)
- ▶ coordinating ICON post-processing
 - ▶ improve maintainability of existing tools
 - ▶ adapt to new requirements



The screenshot shows the Pyicon website. On the left is a 'Table of Contents' with links to 'Contents', 'Installation of pyicon', 'Quickstart', 'Mistral specific', 'pyicon-core', 'pyicon-quickplots', 'pyicon-view', and 'API for pyicon'. Below this is a 'Quick search' box with a 'Go' button. The main content area features the 'pyicon' logo (with a hexagonal icon) and the text 'python diagnostic tool for ICON'. It describes Pyicon as a python post-processing and visualization toolbox for ICON with a focus on ocean data. Three main features are listed: 'pyicon-core' (facilitates plotting), 'pyicon-quickplots' (monitoring suite), and 'pyicon-view' (interactive GUI). The footer mentions development by the Max Planck Institute for Meteorology and the University of Hamburg.



This block contains three logos: the Python logo (a blue and yellow snake), the Xarray logo (a 3D grid of colored blocks), and the Dask logo (a stylized orange and black flame-like shape).



The GitLab logo, featuring a stylized orange and yellow fox head icon above the text 'GitLab' in white on a dark purple background.

What we need

3

We need support for

- ▶ improving usability of our post-processing tools
 - ▶ smarter ways for documentation
 - ▶ better designed APIs
- ▶ improving performance of post-processing
 - ▶ effectively use parallel computing (e.g. with Python/Dask)
 - ▶ better I/O options by new file formats (e.g. Zarr)
- ▶ coordinating ICON post-processing
 - ▶ improve maintainability of existing tools
 - ▶ adapt to new requirements
- ▶ This is in particular relevant for ICON-Seamless



Table of Contents

- Contents
- Installation of pyicon
- Quickstart
- Mistral specific
- pyicon-core
- pyicon-quickplots
- pyicon-view
- API for pyicon

This Page

Show Source

Quick search

Go

pyicon

python diagnostic tool for ICON

Pyicon is a python post-processing and visualization toolbox for ICON with a focus on ocean data. The three main features of pyicon are:

- **pyicon-core**: a number of functions to facilitate the every-day script-based plotting of ICON data
- **pyicon-quickplots**: a monitoring suite for ICON ocean simulations which combines dedicated diagnostic plots of an ICON simulation on a website
- **pyicon-view**: an interactive (ncview-like) plotting GUI for Jupyter notebook

Pyicon is developed by the Max Planck Institute for Meteorology and the University of Hamburg within the DFG-project TRR181 - Energy Transfers in Atmosphere and Ocean.



python™

xarray

DASK



GitLab

What we propose

4

We ask for

- ▶ support for improving our post-processing tools
- ▶ learn from software engineers how to develop more efficient post-processing methods for HPC



What we propose

4

We ask for

- ▶ support for improving our post-processing tools
- ▶ learn from software engineers how to develop more efficient post-processing methods for HPC

We aim for support for

- ▶ 2 months full time for teaching and code development
- ▶ 4 months part time to account for new problems

**2 months
full time**

**4 months
one day per week**

