# An Online (Compression) Laboratory for Climate Science and Meteorology

## Exploring Data Compression and more from the Comfort of your web browser

ESiWACE3 WP3.2: Juniper Tyree[1], Sara Faghih-Naini[2], Peter Dueben[2], Karsten Peters-von Gehlen[3], Heikki J. Järvinen[1]

[1] University of Helsinki, [2] ECMWF, [3] DKRZ

CASTIEL2 Code of the Month, 19.02.2025

*Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) under grant agreement No 101093054.*

Co-funded by the European Union

# Towards exascale weather and climate simulations

Supporting the community of weather and climate modelling in Europe

03

Develop tools to tackle the data challenge of high-resolution models

3.2

Domain-specific data compression

Why do we need but avoid to use Lossy Compression?

The Compression Laboratory

An Online Open CliMet Science Laboratory

Future Outlook

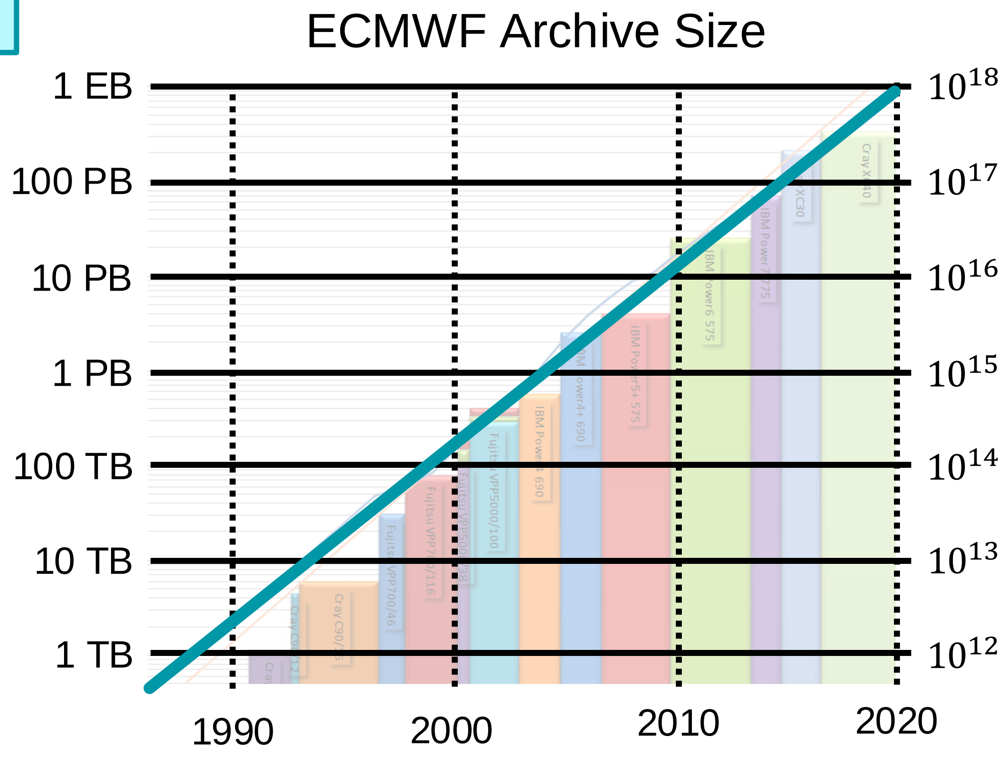
Why do we need but avoid to use Lossy Compression?

The Compression Laboratory

An Online Open CliMet Science Laboratory

Future Outlook

## The Problem – Exponential data growth



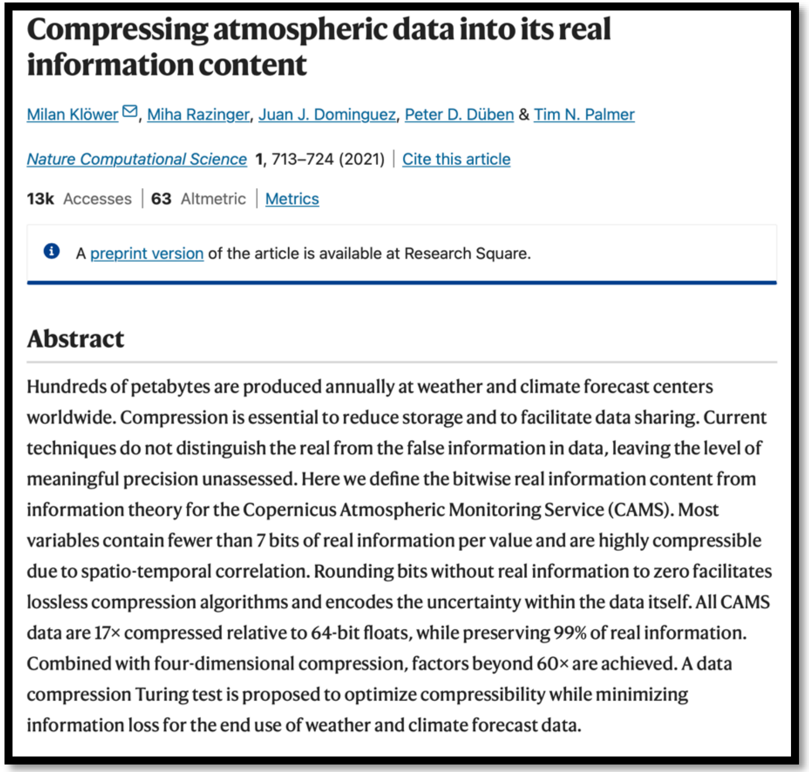
https://climate.copernicus.eu/sites/default/files/2019-11/Martin%20Palkovic.pdf

Weather and climate science has reached the exascale for data storage

Data needs to be produced, stored, analyzed, transferred, published, and archived, which all have I/O costs

Cost per stored petabyte (PB) is ~10k € for cold (tapes) and ~100k € for hot (disk) storage

## The Solution? – Lossy compression can provide huge savings



**Compressing atmospheric data into its real information content**

Milan Klöwer, Miha Razinger, Juan J. Dominguez, Peter D. Düben & Tim N. Palmer

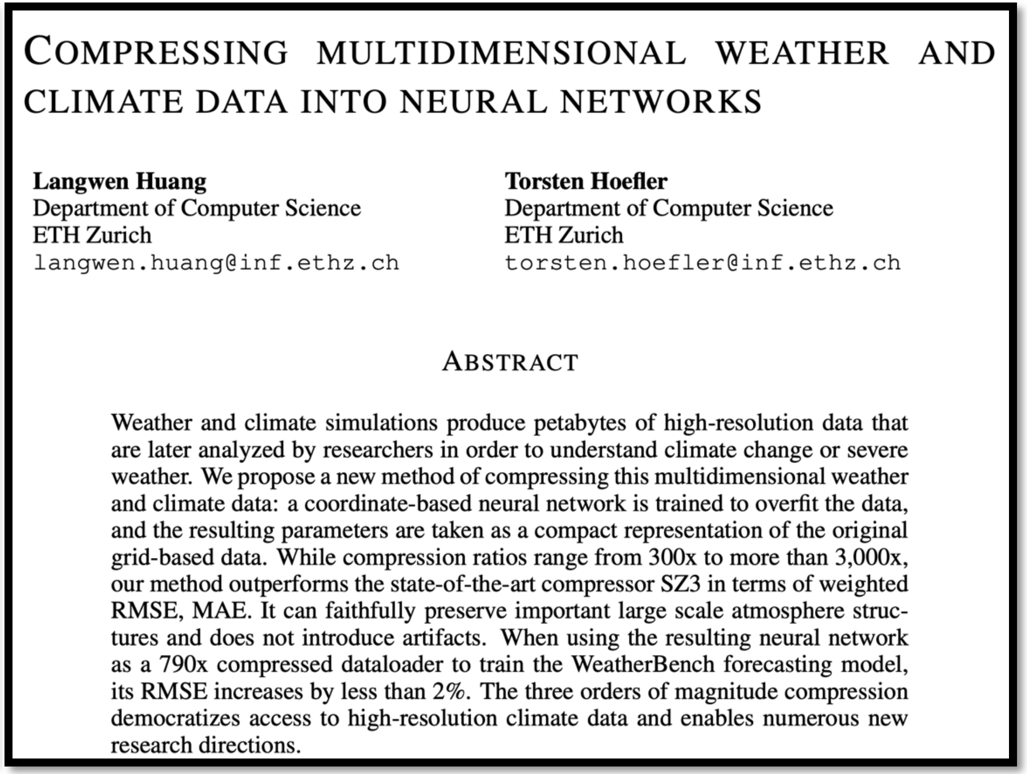
*Nature Computational Science* **1**, 713–724 (2021) | Cite this article

**13k** Accesses | **63** Altmetric | Metrics

A preprint version of the article is available at Research Square.

**Abstract**

Hundreds of petabytes are produced annually at weather and climate forecast centers worldwide. Compression is essential to reduce storage and to facilitate data sharing. Current techniques do not distinguish the real from the false information in data, leaving the level of meaningful precision unassessed. Here we define the bitwise real information content from information theory for the Copernicus Atmospheric Monitoring Service (CAMS). Most variables contain fewer than 7 bits of real information per value and are highly compressible due to spatio-temporal correlation. Rounding bits without real information to zero facilitates lossless compression algorithms and encodes the uncertainty within the data itself. All CAMS data are 17× compressed relative to 64-bit floats, while preserving 99% of real information. Combined with four-dimensional compression, factors beyond 60× are achieved. A data compression Turing test is proposed to optimize compressibility while minimizing information loss for the end use of weather and climate forecast data.



**COMPRESSING MULTIDIMENSIONAL WEATHER AND CLIMATE DATA INTO NEURAL NETWORKS**

**Langwen Huang**
Department of Computer Science
ETH Zurich
`langwen.huang@inf.ethz.ch`

**Torsten Hoefler**
Department of Computer Science
ETH Zurich
`torsten.hoefler@inf.ethz.ch`

ABSTRACT

Weather and climate simulations produce petabytes of high-resolution data that are later analyzed by researchers in order to understand climate change or severe weather. We propose a new method of compressing this multidimensional weather and climate data: a coordinate-based neural network is trained to overfit the data, and the resulting parameters are taken as a compact representation of the original grid-based data. While compression ratios range from 300x to more than 3,000x, our method outperforms the state-of-the-art compressor SZ3 in terms of weighted RMSE, MAE. It can faithfully preserve important large scale atmosphere structures and does not introduce artifacts. When using the resulting neural network as a 790x compressed dataloader to train the WeatherBench forecasting model, its RMSE increases by less than 2%. The three orders of magnitude compression democratizes access to high-resolution climate data and enables numerous new research directions.

## The Problem – Lossy compression is scary

**Computer scientist:** "I can give you great savings by using lossy compression."

**Domain scientist:** "Nice! … Actually, did you remember to keep the …

- **logarithmic error** norms small for **specific humidity**
- $L_2$ **error** small for **temperature**
- **budgets** correct when **integrating** over long climate trajectories
- $L_\infty$ **norm** small for **extreme precipitation events**
- **integral in the vertical** correct even though it sums across **several orders of magnitude**
- **delta** at high precision for $CO_2$

…

And how does lossy data compression change the **enthalpy budget** when cold rain is entering a warm ocean and changing the vertical layering of the vertical column in an ocean model???"

**Computer scientist:** "Goodbye."

## The Problem – We need lossy compression but avoid to use it

Lossy Compression is crucial to handle Data Growth

Safety Requirements for lossy compression
in weather and climate science are vague

**Lack of Trust** in Lossy Compression
to uphold the safety requirements
for everyday domain-scientist users

## The Roadmap to *Fearless* Lossy Compression

Lossy Compression is crucial to work with Data Growth

Establish clear Safety Requirements

Simplify building **Trust** in Lossy Compression:
(1) Benchmark existing methods for safety and performance
(2) Provide safeguards to make any compressor safe
(3) **Convince yourself** in an Online Compression Laboratory

Why do we need but avoid to use Lossy Compression?

The Compression Laboratory

An Online Open CliMet Science Laboratory

Future Outlook

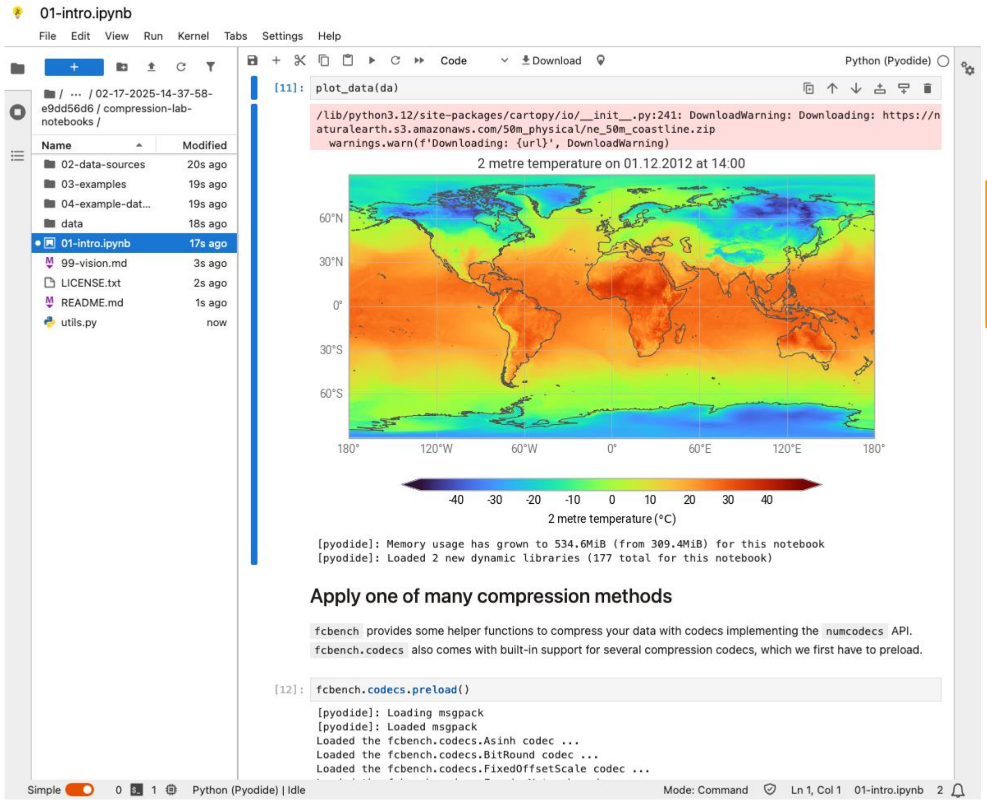
## A brief overview of the Compression Laboratory Notebooks

How to compress?

What compression?

Which datasets?

Examples case studies


01-intro.ipynb

File Edit View Run Kernel Tabs Settings Help

/ … / 02-17-2025-14-37-58-e9dd56d6 / compression-lab-notebooks /

| Name | Modified |
|---|---|
| 02-data-sources | 20s ago |
| 03-examples | 19s ago |
| 04-example-dat… | 19s ago |
| data | 18s ago |
| 01-intro.ipynb | 17s ago |
| 99-vision.md | 3s ago |
| LICENSE.txt | 2s ago |
| README.md | 1s ago |
| utils.py | now |

```
[11]: plot_data(da)

/lib/python3.12/site-packages/cartopy/io/__init__.py:241: DownloadWarning: Downloading: https://naturalearth.s3.amazonaws.com/50m_physical/ne_50m_coastline.zip
  warnings.warn(f'Downloading: {url}', DownloadWarning)
```

```
[pyodide]: Memory usage has grown to 534.6MiB (from 309.4MiB) for this notebook
[pyodide]: Loaded 2 new dynamic libraries (177 total for this notebook)
```

Apply one of many compression methods

`fcbench` provides some helper functions to compress your data with codecs implementing the `numcodecs` API.
`fcbench.codecs` also comes with built-in support for several compression codecs, which we first have to preload.

```
[12]: fcbench.codecs.preload()

[pyodide]: Loading msgpack
[pyodide]: Loaded msgpack
Loaded the fcbench.codecs.Asinh codec ...
Loaded the fcbench.codecs.BitRound codec ...
Loaded the fcbench.codecs.FixedOffsetScale codec ...
```

Simple 0 1 Python (Pyodide) | Idle Mode: Command Ln 1, Col 1 01-intro.ipynb 2

Example: Introduction to loading and compressing data

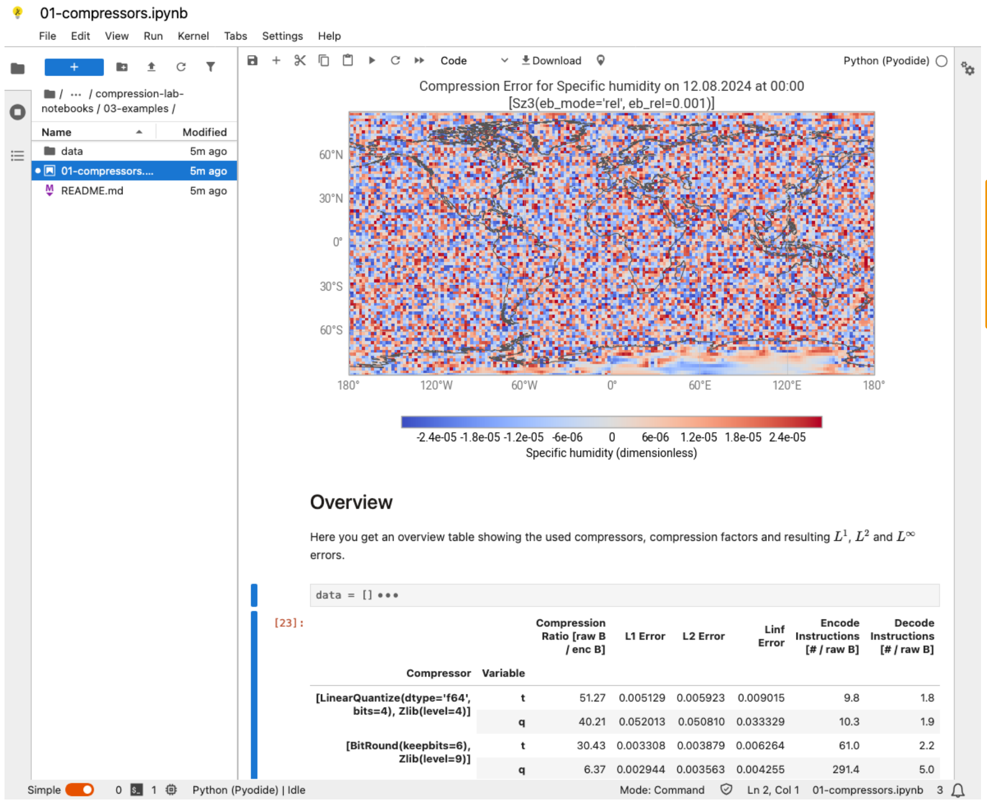
## A brief overview of the Compression Laboratory Notebooks

How to compress?

What compression?

Which datasets?

Examples case studies


01-compressors.ipynb
File Edit View Run Kernel Tabs Settings Help
Code
Download
Python (Pyodide)
/ … / compression-lab-notebooks / 03-examples /
Name
Modified
data 5m ago
01-compressors.... 5m ago
README.md 5m ago
60°N
30°N
0°
30°S
60°S
180°
120°W
60°W
0°
60°E
120°E
180°
-2.4e-05 -1.8e-05 -1.2e-05 -6e-06 0 6e-06 1.2e-05 1.8e-05 2.4e-05
Specific humidity (dimensionless)
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 2, Col 1
01-compressors.ipynb

Compression Error for Specific humidity on 12.08.2024 at 00:00
[Sz3(eb_mode='rel', eb_rel=0.001)]

### Overview

Here you get an overview table showing the used compressors, compression factors and resulting $L^1$, $L^2$ and $L^\infty$ errors.

```
data = [] •••
```

[23]:

| Compressor | Variable | Compression Ratio [raw B / enc B] | L1 Error | L2 Error | Linf Error | Encode Instructions [# / raw B] | Decode Instructions [# / raw B] |
|---|---|---|---|---|---|---|---|
| [LinearQuantize(dtype='f64', bits=4), Zlib(level=4)] | t | 51.27 | 0.005129 | 0.005923 | 0.009015 | 9.8 | 1.8 |
| | q | 40.21 | 0.052013 | 0.050810 | 0.033329 | 10.3 | 1.9 |
| [BitRound(keepbits=6), Zlib(level=9)] | t | 30.43 | 0.003308 | 0.003879 | 0.006264 | 61.0 | 2.2 |
| | q | 6.37 | 0.002944 | 0.003563 | 0.004255 | 291.4 | 5.0 |

Example: Comparing lossy compressor performance

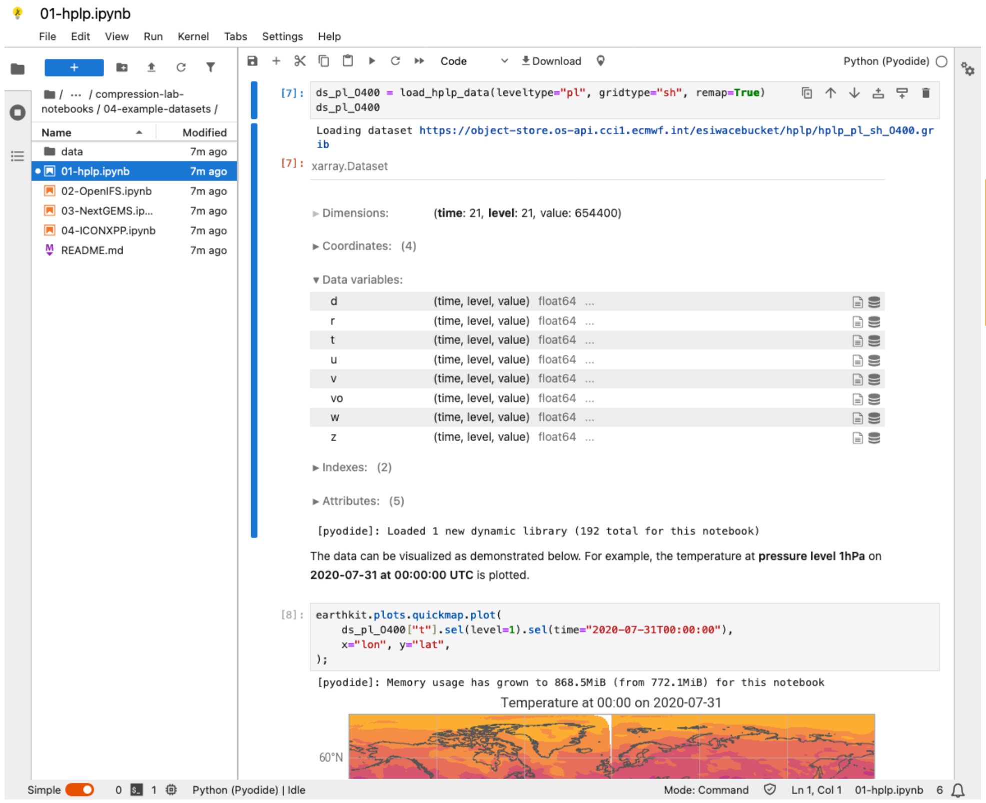
## A brief overview of the Compression Laboratory Notebooks

How to compress?

What compression?

Which datasets?

Examples case studies


01-hplp.ipynb
File Edit View Run Kernel Tabs Settings Help
Code
Download
Python (Pyodide)
/ … / compression-lab-notebooks / 04-example-datasets /
Name
Modified
data 7m ago
01-hplp.ipynb 7m ago
02-OpenIFS.ipynb 7m ago
03-NextGEMS.ip… 7m ago
04-ICONXPP.ipynb 7m ago
README.md 7m ago
Simple 0 1 Python (Pyodide) | Idle
Mode: Command Ln 1, Col 1 01-hplp.ipynb 6
60°N

```
[7]: ds_pl_0400 = load_hplp_data(leveltype="pl", gridtype="sh", remap=True)
     ds_pl_0400
```

```
Loading dataset https://object-store.os-api.cci1.ecmwf.int/esiwacebucket/hplp/hplp_pl_sh_0400.gr
ib
[7]: xarray.Dataset

▸ Dimensions:      (time: 21, level: 21, value: 654400)
▸ Coordinates:  (4)
▾ Data variables:
    d        (time, level, value)  float64 ...
    r        (time, level, value)  float64 ...
    t        (time, level, value)  float64 ...
    u        (time, level, value)  float64 ...
    v        (time, level, value)  float64 ...
    vo       (time, level, value)  float64 ...
    w        (time, level, value)  float64 ...
    z        (time, level, value)  float64 ...
▸ Indexes:  (2)
▸ Attributes:  (5)

[pyodide]: Loaded 1 new dynamic library (192 total for this notebook)
```

The data can be visualized as demonstrated below. For example, the temperature at **pressure level 1hPa** on **2020-07-31 at 00:00:00 UTC** is plotted.

```
[8]: earthkit.plots.quickmap.plot(
         ds_pl_0400["t"].sel(level=1).sel(time="2020-07-31T00:00:00"),
         x="lon", y="lat",
     );
```

```
[pyodide]: Memory usage has grown to 868.5MiB (from 772.1MiB) for this notebook
```

Temperature at 00:00 on 2020-07-31

Example: Opening large remote datasets

## A brief overview of the Compression Laboratory Notebooks

How to compress?

What compression?

Which datasets?

Examples case studies

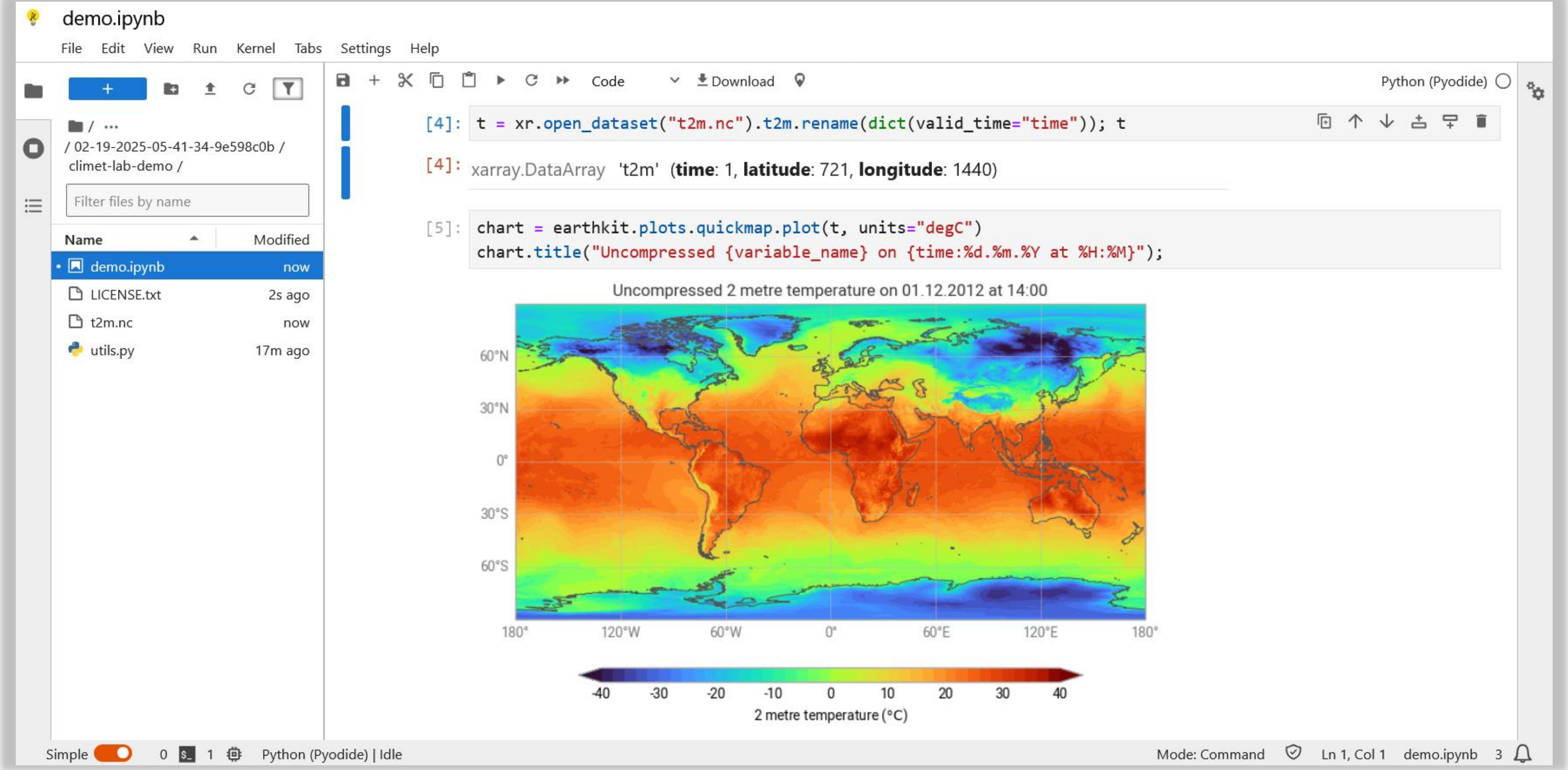
*Work in Progress:*

Compressing precipitation

with Clément Bouvier (UH), Joona Cornér (UH), Peter Dueben (ECMWF), and Milan Klöwer (Oxford)

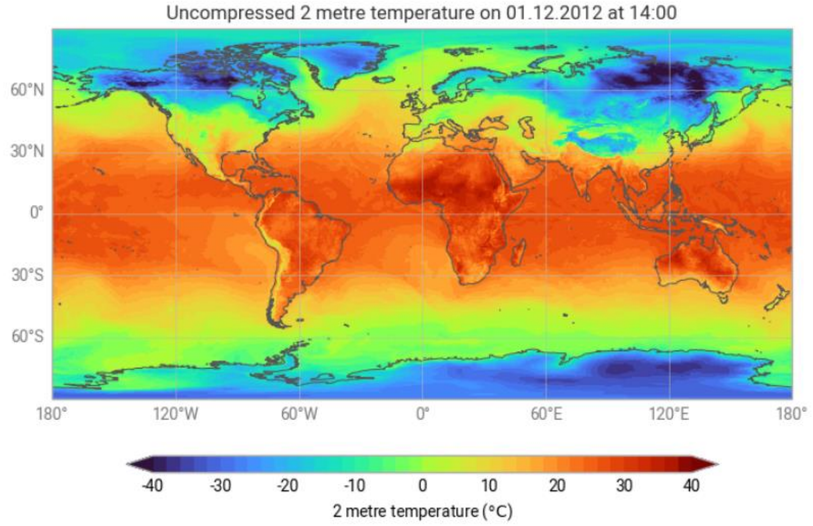
Compressing model states during runtime

with Madeleine Ekblom (FMI) and Milan Klöwer (Oxford)

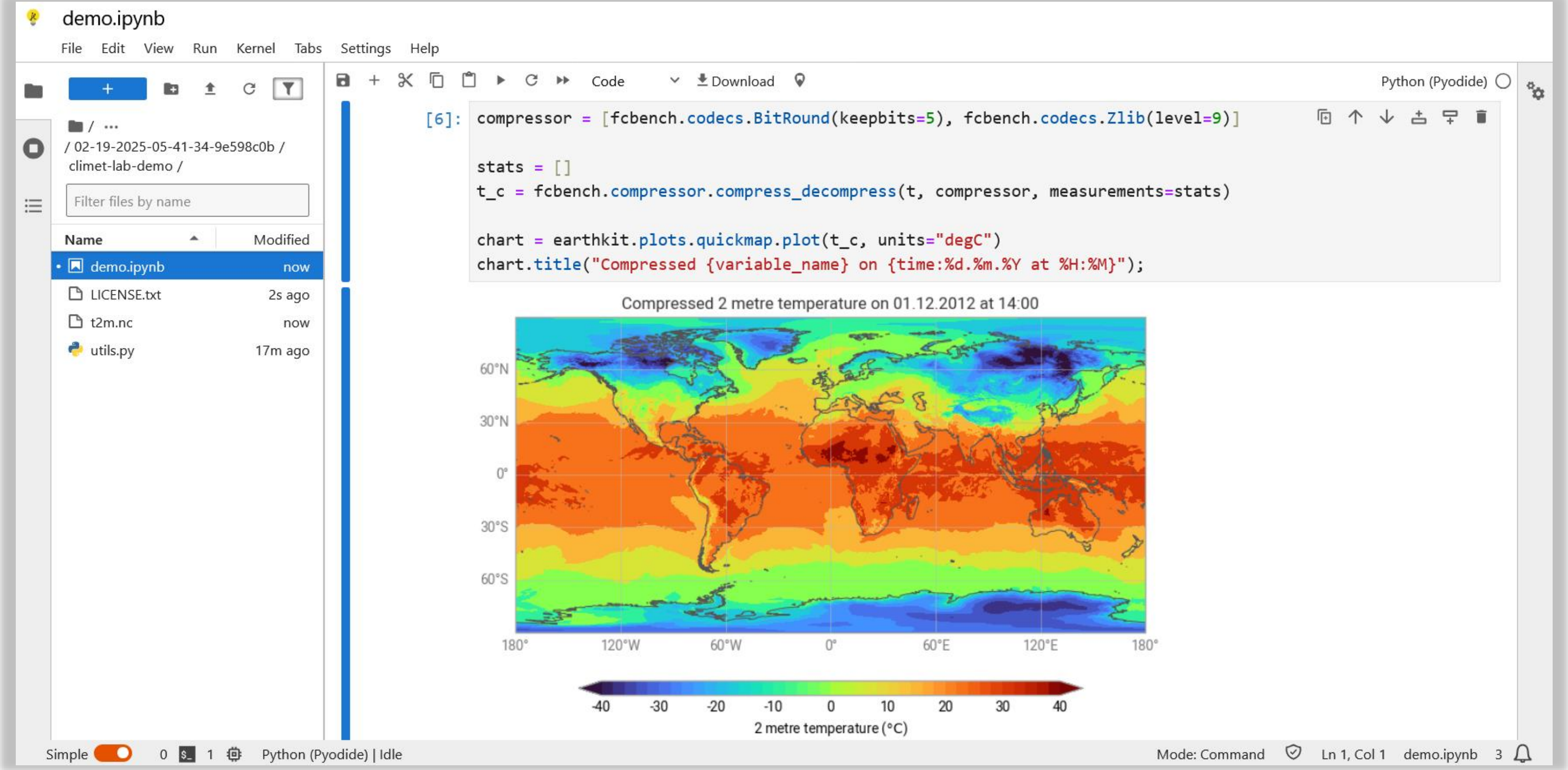


| Name | Modified |
|---|---|
| demo.ipynb | now |
| LICENSE.txt | 2s ago |
| t2m.nc | now |
| utils.py | 17m ago |

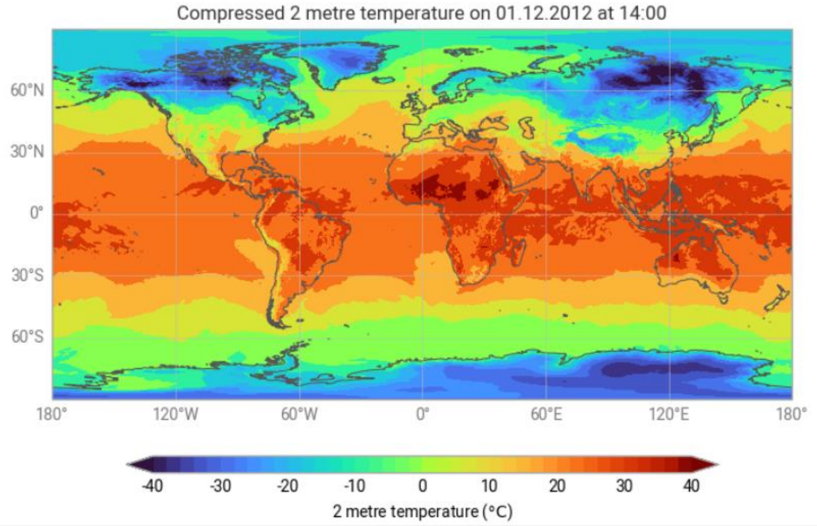
```
[4]: t = xr.open_dataset("t2m.nc").t2m.rename(dict(valid_time="time")); t
[4]: xarray.DataArray 't2m' (time: 1, latitude: 721, longitude: 1440)
[5]: chart = earthkit.plots.quickmap.plot(t, units="degC")
     chart.title("Uncompressed {variable_name} on {time:%d.%m.%Y at %H:%M}");
```



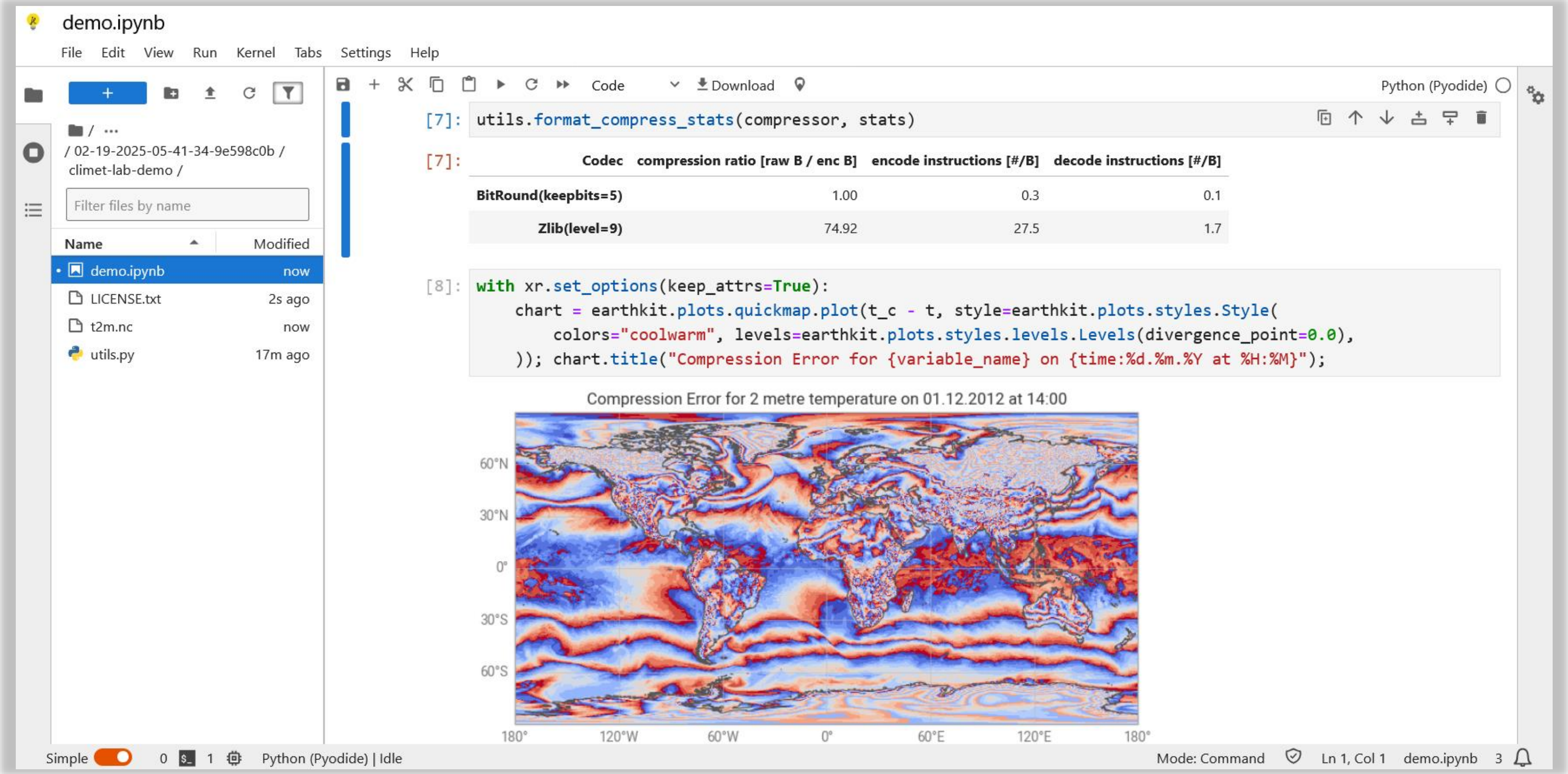
https://lab.climet.eu/v0.2/github/juntyr/climet-lab-demo/v0.2.0/demo.ipynb


demo.ipynb
File Edit View Run Kernel Tabs Settings Help
Code
Download
Python (Pyodide)
/ 02-19-2025-05-41-34-9e598c0b / climet-lab-demo /
Filter files by name
Name Modified
demo.ipynb now
LICENSE.txt 2s ago
t2m.nc now
utils.py 17m ago
Simple 0 1 Python (Pyodide) | Idle
Mode: Command Ln 1, Col 1 demo.ipynb 3

```
[6]: compressor = [fcbench.codecs.BitRound(keepbits=5), fcbench.codecs.Zlib(level=9)]

     stats = []
     t_c = fcbench.compressor.compress_decompress(t, compressor, measurements=stats)

     chart = earthkit.plots.quickmap.plot(t_c, units="degC")
     chart.title("Compressed {variable_name} on {time:%d.%m.%Y at %H:%M}");
```



https://lab.climet.eu/v0.2/github/juntyr/climet-lab-demo/v0.2.0/demo.ipynb



| Name | Modified |
|---|---|
| demo.ipynb | now |
| LICENSE.txt | 2s ago |
| t2m.nc | now |
| utils.py | 17m ago |

```
[7]: utils.format_compress_stats(compressor, stats)
```

[7]:

| Codec | compression ratio [raw B / enc B] | encode instructions [#/B] | decode instructions [#/B] |
|---|---|---|---|
| BitRound(keepbits=5) | 1.00 | 0.3 | 0.1 |
| Zlib(level=9) | 74.92 | 27.5 | 1.7 |

```
[8]: with xr.set_options(keep_attrs=True):
         chart = earthkit.plots.quickmap.plot(t_c - t, style=earthkit.plots.styles.Style(
             colors="coolwarm", levels=earthkit.plots.styles.levels.Levels(divergence_point=0.0),
         )); chart.title("Compression Error for {variable_name} on {time:%d.%m.%Y at %H:%M}");
```

Compression Error for 2 metre temperature on 01.12.2012 at 14:00

https://lab.climet.eu/v0.2/github/juntyr/climet-lab-demo/v0.2.0/demo.ipynb

Why do we need but avoid to use Lossy Compression?

The Compression Laboratory

An Online Open CliMet Science Laboratory

Future Outlook

## Why build an Online Laboratory? – Try and Convince Yourself

Trust in lossy compression requires **convincing yourself**

Convincing yourself requires **trying things out yourself**, with your own data and analyses

Trying out code has a setup/installation time cost
Reduce this friction to **engage** more stakeholders

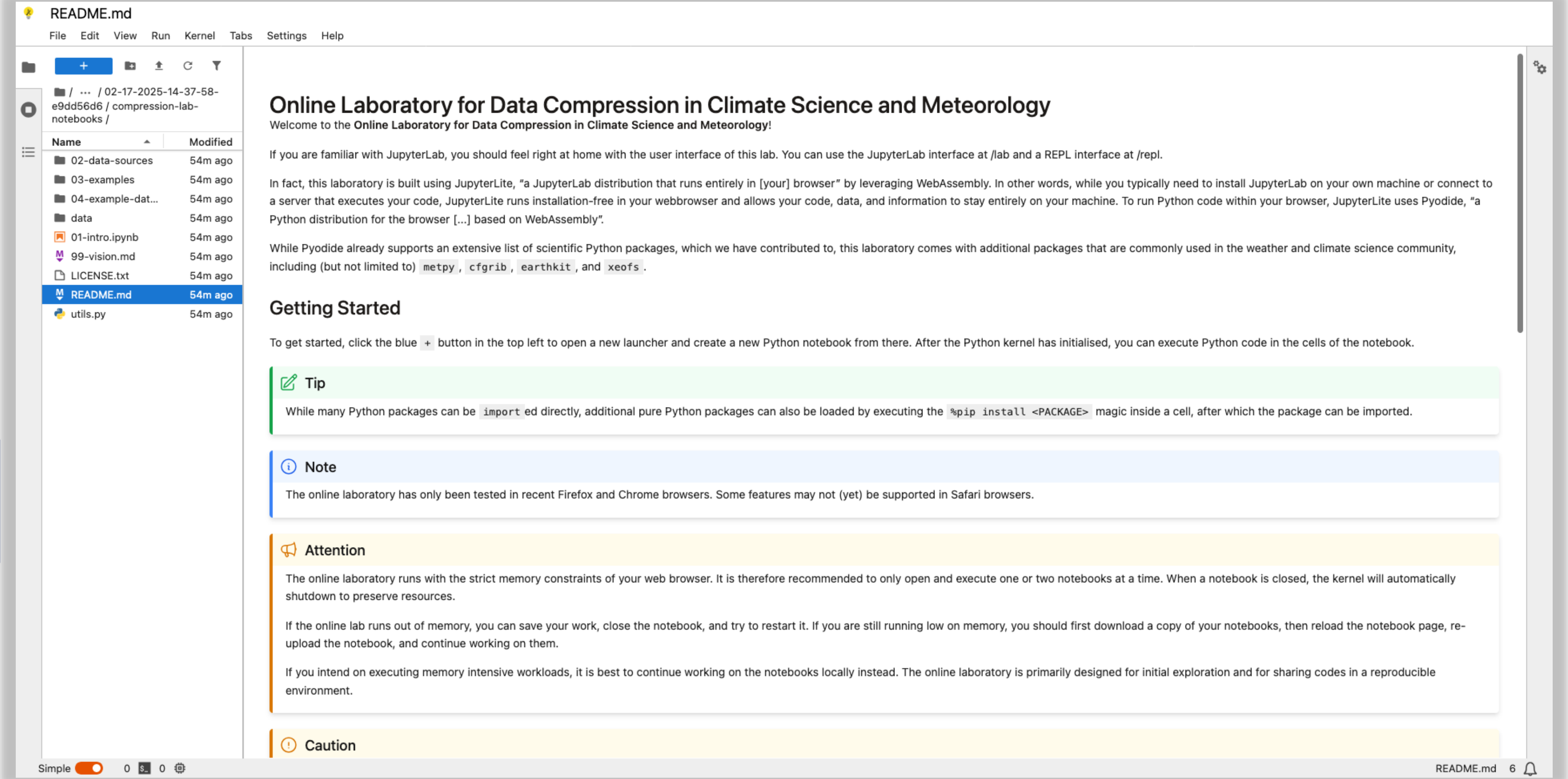
## Why build an Online Laboratory? – Try and Convince Yourself

**Trying out compression … the *easy* way**

**Compression should be easy to try out**

Let’s reduce the barrier to experiment and convince yourself,
a serverless WebAssembly container within your browser:

compression.lab.climet.eu



# Online Laboratory for Data Compression in Climate Science and Meteorology

Welcome to the **Online Laboratory for Data Compression in Climate Science and Meteorology**!

If you are familiar with JupyterLab, you should feel right at home with the user interface of this lab. You can use the JupyterLab interface at /lab and a REPL interface at /repl.

In fact, this laboratory is built using JupyterLite, "a JupyterLab distribution that runs entirely in [your] browser" by leveraging WebAssembly. In other words, while you typically need to install JupyterLab on your own machine or connect to a server that executes your code, JupyterLite runs installation-free in your webbrowser and allows your code, data, and information to stay entirely on your machine. To run Python code within your browser, JupyterLite uses Pyodide, "a Python distribution for the browser [...] based on WebAssembly".

While Pyodide already supports an extensive list of scientific Python packages, which we have contributed to, this laboratory comes with additional packages that are commonly used in the weather and climate science community, including (but not limited to) `metpy`, `cfgrib`, `earthkit`, and `xeofs`.

## Getting Started

To get started, click the blue `+` button in the top left to open a new launcher and create a new Python notebook from there. After the Python kernel has initialised, you can execute Python code in the cells of the notebook.

**Tip**

While many Python packages can be `import`ed directly, additional pure Python packages can also be loaded by executing the `%pip install <PACKAGE>` magic inside a cell, after which the package can be imported.

**Note**

The online laboratory has only been tested in recent Firefox and Chrome browsers. Some features may not (yet) be supported in Safari browsers.

**Attention**

The online laboratory runs with the strict memory constraints of your web browser. It is therefore recommended to only open and execute one or two notebooks at a time. When a notebook is closed, the kernel will automatically shutdown to preserve resources.

If the online lab runs out of memory, you can save your work, close the notebook, and try to restart it. If you are still running low on memory, you should first download a copy of your notebooks, then reload the notebook page, re-upload the notebook, and continue working on them.

If you intend on executing memory intensive workloads, it is best to continue working on the notebooks locally instead. The online laboratory is primarily designed for initial exploration and for sharing codes in a reproducible environment.

**Caution**

## Why build an Online Laboratory? – Ease of Use

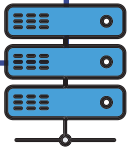
**Easy and quick to pick up and go**

Just open a website, **no installation, try it out!**

Open as easily as documentation, but interactive
3x-5x slower Python is sufficient for examples
Motivate and excite first, only then ask for setup

## The technology behind the Online Laboratory

Familiar Stack
on a **server**



## The technology behind the Online Laboratory

Familiar Stack
on a ~~**server**~~

python™

jupyter

*Same* Code
*Same* Interface
*Same* Results

No installation
Easy sharing

**Serverless container**
in your browser

PYODIDE

WA

jupyterlite

Benefit from the shared **ecosystem** of Python packages:

**Pure Python**
(`xarray`, `zarr`, `fsspec`, …)
PyPi

**Scientific Ports**
(`numpy`, `scipy`, `sklearn`, …)
Pyodide Community

**Earth Science Ports**
(`eccodes`, `netCDF4`, `cartopy`, …)
Online Laboratory

## Focus on reducing user friction and working with big data

**Remote streaming access to large datasets (GB, TB, …)**

| **Zarr** | **NetCDF4** | **GRIB** |
|---|---|---|
| natively chunked `fsspec` filesystems | separate metadata JIT with `kerchunk` | requires full scan prior with `gribscan` |

**Downloading smaller datasets:**
Access via `urllib`, `ecmwfapi`, `cdsapi`, `earthkit.data`, …
Load via the native `cfgrib` / `netCDF4` / `h5netcdf` / `zarr` driver

**Local datasets** of any size can be mounted and read in-place

## Why build an Online Laboratory? – Community Collaboration

**Service to the greater community**

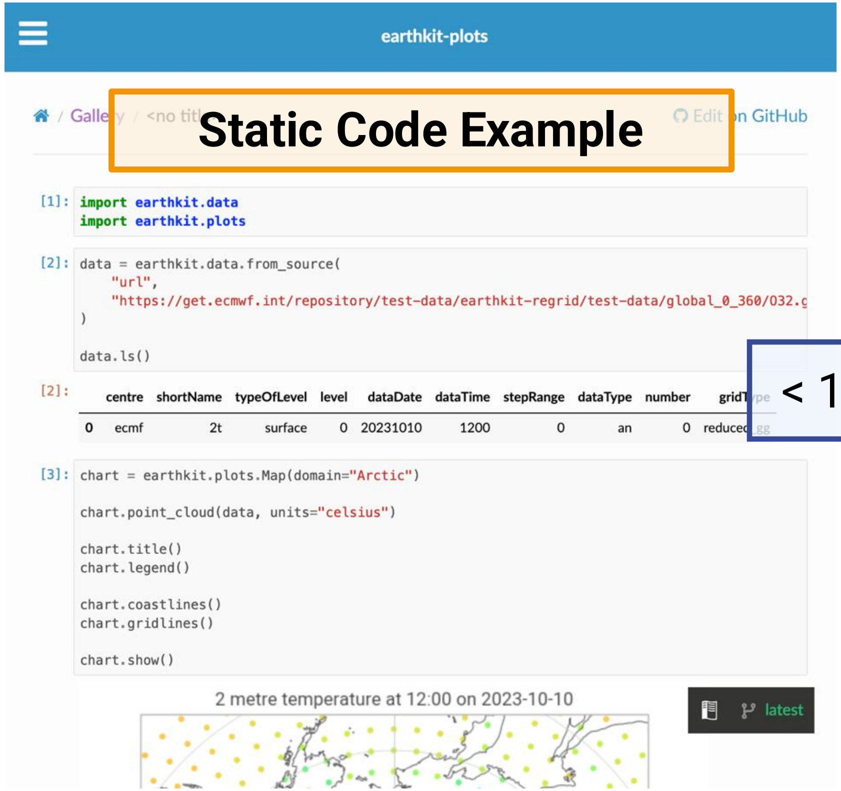
Locked Python environment with pre-installed packages

Your local notebooks and data never leave your device

Easy **sharing** of code samples and experiments:

e.g. lab.climet.eu/v0.2/github/:org/:repo/:branch/*

## Why build an Online Laboratory? – Interactive Documentation



earthkit-plots

Static Code Example

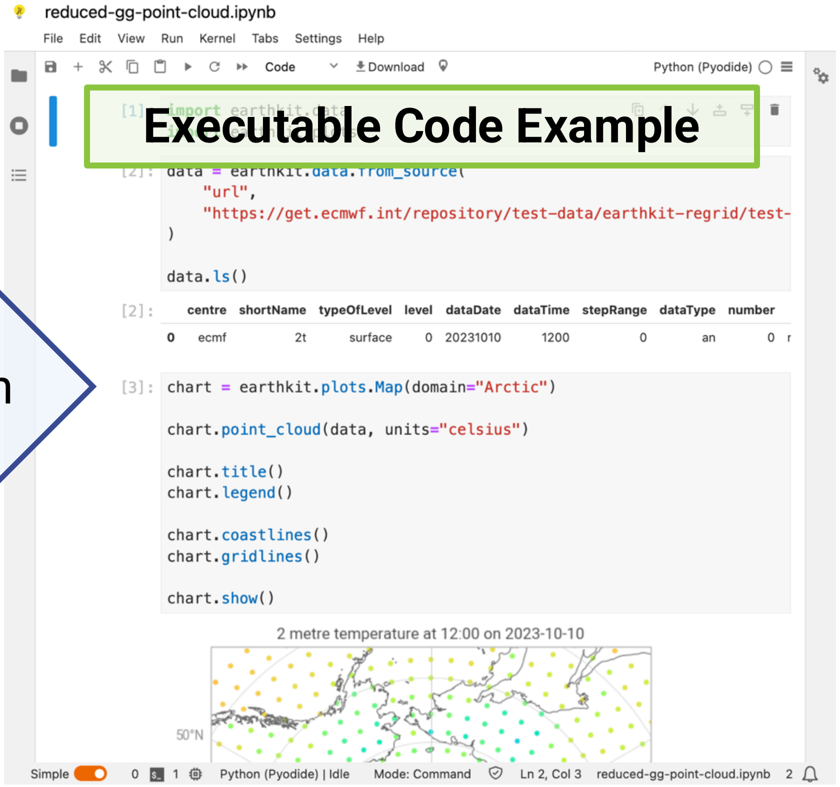
```
[1]: import earthkit.data
     import earthkit.plots

[2]: data = earthkit.data.from_source(
         "url",
         "https://get.ecmwf.int/repository/test-data/earthkit-regrid/test-data/global_0_360/032.g
     )

     data.ls()
```

```
[2]:     centre  shortName  typeOfLevel  level  dataDate  dataTime  stepRange  dataType  number  gridType
      0  ecmf           2t      surface      0  20231010      1200          0        an       0  reduced_gg
```

```
[3]: chart = earthkit.plots.Map(domain="Arctic")

     chart.point_cloud(data, units="celsius")

     chart.title()
     chart.legend()

     chart.coastlines()
     chart.gridlines()

     chart.show()
```

2 metre temperature at 12:00 on 2023-10-10

< 1 min



reduced-gg-point-cloud.ipynb

Executable Code Example

```
[2]: data = earthkit.data.from_source(
         "url",
         "https://get.ecmwf.int/repository/test-data/earthkit-regrid/test-
     )

     data.ls()
```

```
[2]:     centre  shortName  typeOfLevel  level  dataDate  dataTime  stepRange  dataType  number
      0  ecmf           2t      surface      0  20231010      1200          0        an       0
```

```
[3]: chart = earthkit.plots.Map(domain="Arctic")

     chart.point_cloud(data, units="celsius")

     chart.title()
     chart.legend()

     chart.coastlines()
     chart.gridlines()

     chart.show()
```

2 metre temperature at 12:00 on 2023-10-10

https://earthkit-plots.readthedocs.io/en/latest/examples/gallery/grid-types/reduced-gg-point-cloud.html

https://lab.climet.eu/v0.2/raw/github/ecmwf/earthkit-plots/main/docs/examples/gallery/grid-types/reduced-gg-point-cloud.ipynb

Why do we need but avoid to use Lossy Compression?

The Compression Laboratory

An Online Open CliMet Science Laboratory

Future Outlook

## Upcoming Online Laboratory Community Service Features

now

Open existing repositories or notebooks

05/25

Customize the locked Python package versions

2025

Anyone can produce, share, and try out ready-to-go interactive documentation, analyses, or code examples

## The Roadmap to *Fearless* Lossy Compression

2025

Establish clear **Safety Requirements**

Benchmark the Safety and Performance of compressors

**Community** Outreach facilitated by the Online Laboratory

**Collaboration** with and **Service** for other projects

2026

**Safeguards** to safely use any lossy compressor

**Recommendations** for Operational Lossy Compression

CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

## Thank you very much!

**Thank you also to everyone who has helped with this project:**
Clément Bouvier (UH), Hood Chatham (Pyodide @ MIT), Joona Cornér (UH), Alex Crichton (WASM @ Fermyon), Milan Klöwer (Oxford), Daniel Köhler (UH), Juha Lento (CSC), Iain Russell (ECMWF), Jeremy Tuloup (Jupyter @ QuantStack), and James Varndell (ECMWF)

/climet-eu/lab

EuroHPC Joint Undertaking

*Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) under grant agreement No 101093054.*

Co-funded by the European Union