PANGEO
A COMMUNITY-DRIVEN EFFORT FOR BIG DATA GEOSCIENCE
What Drives Progress in Oceanography?

New Ideas

\[
E = \frac{\rho_0|U|}{\pi} \int_{\|U\|}^{N/|U|} P_{1D}(k) \sqrt{N^2 - |U|^2 k^2} \sqrt{|U|^2 k^2} \, dk
\]

New Observations

New Simulations

\[
P_{2D}(k, l) \quad \text{and} \quad P_{1D}(k)\]

is the effective one-dimensional (1D) topographic spectrum. Hence, the wave radiation from 2D topography reduces to an equivalent problem of wave radiation from 1D topography with the effective spectrum given by

\[
P_{1D}(k).
\]

The effective 1D spectrum captures the effects of 2D topography on lee-wave radiation in the subcritical to-supercritical topography limit, that is, \(P_{2D}(k, l)\) and \(P_{1D}(k)\) result into identical radiation estimates for small steepness parameters. However, the suppression of wave radiation in the critical topography limit is different in 1D and 2D; 1D critical topography leads to blocking of the mean flow, while in 2D both blocking and splitting of the mean flow can occur. Hence, the radiation from \(P_{2D}(k, l)\) and \(P_{1D}(k)\) is expected to diverge as the topographic steepness is increased.

Bottom topography

Simulations are configured with multiscale topography characterized by small-scale abyssal hills a few kilometers wide based on multibeam observations from Drake Passage. The topographic spectrum associated with abyssal hills is well described by an anisotropic parametric representation proposed by Goff and Jordan (1988):

\[
P_{2D}(k, l) = 2 \frac{H_2}{k_0 l_0} k^2 l^2 \frac{m}{2} k^2 l^2 \frac{m}{2} ! \]

where \(k_0\) and \(l_0\) set the wavenumbers of the large hills, \(m\) is the high-wavenumber spectral slope, related to the parameter \(n\) used in Goff and Jordan (1988) as \(m = 2(n + 1)\), and \(H\) is the root-mean-square (rms) topographic height.

Nikurashin and Ferrari (2010b) estimated with a least squares fit to multibeam data that representative values in the Drake Passage region are \(k_0 = 2.3 \times 10^{-2}\), \(l_0 = 1.3 \times 10^{-2}\), \(H = 305\) m, and \(m = 3.5\). Given that the abyssal hills are only slightly anisotropic with a horizontal aspect ratio of \(k_0 / l_0 < 1.8\), we hereinafter assume that the topography is isotropic and characterized by the averaged rolloff wavenumber \(k_0 = l_0 = 1.8 \times 10^{-2}\).
Altimetry Past, Present & Future

1978 1.5 degree resolution
Seasat

Credit: NASA SVS / PODAAC
I would like all the data please…
How should scientific data infrastructure be organized for the petabyte era?
WHAT IS PANGEO?

“A community platform for Big Data geoscience”

• Open Community
• Open Source Software
• Open Source Infrastructure
PANGEO COMMUNITY

HTTP://PANGEO.IO
PANGEO TIMELINE

- **2013**: Xarray created
- **2014**: Xarray starts to catch on
- **2015**: first Pangeo workshop
- **2016**: Earthcube Pangeo proposal awarded
- **2017**: Pangeo binder

2013 - 2019
Growth of major programming languages
Based on Stack Overflow question views in World Bank high-income countries

Stack Overflow Traffic to Questions About Selected Python Packages
Based on visits to Stack Overflow questions from World Bank high-income countries

source: stackoverflow.com
PANGEO SOFTWARE
Scientific Python for Geoscience

Iris

cartopy

GCM

aospy

TensorFlow

DASK

pandas

xarray

SciPy

matplotlib

NumPy

IPython

Jupyter

Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)
XARRAY DATASET: MULTIDIMENSIONAL VARIABLES WITH COORDINATES AND METADATA

Data variables used for computation

Coordinates describe data

Indexes align data

Attributes metadata ignored by operations

“netCDF meets pandas.DataFrame”

Credit: Stephan Hoyer
import xarray as xr

ds = xr.open_dataset('NOAA_NCDC_ERSST_v3b_SST.nc')

ds

<xarray.Dataset>
Dimensions:  (lat: 89, lon: 180, time: 684)
Coordinates:
  * lat   (lat) float32 -88.0 -86.0 -84.0 -82.0 -80.0 -78.0 -76.0 -74.0 ...  
  * lon   (lon) float32 0.0 2.0 4.0 6.0 8.0 10.0 12.0 14.0 16.0 18.0 20.0 ...  
  * time  (time) datetime64[ns] 1960-01-15 1960-02-15 1960-03-15 ...  
Data variables:
  sst   (time, lat, lon) float64 nan nan nan nan nan nan nan nan nan ...  
Attributes:
  Conventions:  IRIDL  
  source:  https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/...
XARRAY: LABEL-BASED SELECTION

```python
# select and plot data from my birthday
ds.sst.sel(time='1982-08-07', method='nearest').plot()
```

![Map of time = 1982-08-15 with temperature ranges from -30 to 30]
# zonal and time mean temperature
ds.sst.mean(dim=('time', 'lon')).plot()
XARRAY: GROUPING AND AGGREGATION

```python
sst_clim = sst.groupby('time.month').mean(dim='time')
sst_anom = sst.groupby('time.month') - sst_clim
nino34_index = (sst_anom.sel(lat=slice(-5, 5), lon=slice(190, 240))
    .mean(dim=('lon', 'lat'))
    .rolling(time=3).mean(dim='time'))
nino34_index.plot()
```
xarray
https://github.com/pydata/xarray

• label-based indexing and arithmetic

• interoperability with the core scientific Python packages (e.g., pandas, NumPy, Matplotlib)

• out-of-core computation on datasets that don’t fit into memory (thanks dask!)

• wide range of input/output (I/O) options: netCDF, HDF, geoTIFF, zarr

• advanced multi-dimensional data manipulation tools such as group-by and resampling
LEGACY SOFTWARE

FERRET
An Analysis Tool for Gridded and Non-Gridded Data

NCL
NCAR Command Language

NASA Panoply

INGRID
Preventive Tool for Fire Activity in Chancellor, Malawi
Complex computations represented as a graph of individual tasks.

Scheduler optimizes execution of graph.

ND-Arrays are split into chunks that comfortably fit in memory.

D A S K

https://github.com/dask/dask/
EXAMPLE CALCULATION: TAKE THE MEAN!

**multidimensional array**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>(x', 0, 0)</td>
<td>(x', 0, 1)</td>
</tr>
<tr>
<td>5</td>
<td>(x', 1, 0)</td>
<td>(x', 1, 1)</td>
</tr>
<tr>
<td>5</td>
<td>(x', 2, 0)</td>
<td>(x', 2, 1)</td>
</tr>
<tr>
<td>5</td>
<td>(x', 3, 0)</td>
<td>(x', 3, 1)</td>
</tr>
</tbody>
</table>

**serial execution (a loop)**

1. read chunk from disk → reduce
2. read chunk from disk → reduce
3. read chunk from disk → reduce
4. store
5. store
6. store
7. reduce
Example Calculation: Take the Mean!

Multidimensional array:

<table>
<thead>
<tr>
<th></th>
<th>8</th>
<th>8</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>((x', 0, 0))</td>
<td>((x', 0, 1))</td>
<td>((x', 0, 2))</td>
</tr>
<tr>
<td>5</td>
<td>((x', 1, 0))</td>
<td>((x', 1, 1))</td>
<td>((x', 1, 2))</td>
</tr>
<tr>
<td>5</td>
<td>((x', 2, 0))</td>
<td>((x', 2, 1))</td>
<td>((x', 2, 2))</td>
</tr>
<tr>
<td>5</td>
<td>((x', 3, 0))</td>
<td>((x', 3, 1))</td>
<td>((x', 3, 2))</td>
</tr>
</tbody>
</table>

Parallel execution (dask graph)
“Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.”
FILE-BASED APPROACH

Data provider's responsibilities

Step 1: Download

files

Step 2: Analyze

local disk

End user's responsibilities

file

file

file
SERVER-SIDE DATABASE

Data provider’s responsibilities

End user’s responsibilities
Cloud-Native Approach

Data provider’s responsibilities

- object
- object
- object

Object store

End user’s responsibilities

- worker
- scheduler
- notebook

Compute cluster

Cloud region
PANGEO PRINCIPLES FOR CLOUD-NATIVE SCIENCE INFRASTRUCTURE

- **Community-driven** - Our needs are no different from those of our peer institutions. By developing infrastructure collaboratively, we can accomplish much more than any one institution can alone.

- **Open source** - Because infrastructure is code, the code should be licensed in a way that enables the entire research community to reuse and build upon it.

- **Modular** - “all in one” solutions are impossible to maintain long term. Separation of concerns is a key principle of good software and systems engineering.

- **Vendor neutral** - Academic research infrastructure should use only vendor-neutral services APIs. If this principle is followed, it means we can redeploy our infrastructure anywhere.
**PANGEO ARCHITECTURE**

“Analysis Ready Data” stored on globally-available distributed storage.

Parallel computing system allows users deploy clusters of compute nodes for data processing.

Dask tells the nodes what to do.

**Jupyter** for interactive access remote systems

**Xarray** provides data structures and intuitive interface for interacting with datasets

**Cloud / HPC**
## Build Your Own Pangeo

<table>
<thead>
<tr>
<th>Storage Formats</th>
<th>HDF</th>
<th>OPeNDAP</th>
<th>Cloud Optimized COG/Zarr/Parquet/etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ND-Arrays</td>
<td>NumPy</td>
<td>DASK</td>
<td>More coming…</td>
</tr>
<tr>
<td>Data Models</td>
<td>xarray</td>
<td>Iris</td>
<td></td>
</tr>
<tr>
<td>Processing Mode</td>
<td>jupyter</td>
<td>Batch</td>
<td>Serverless</td>
</tr>
<tr>
<td>Compute Platform</td>
<td>HPC</td>
<td>Cloud</td>
<td>Local</td>
</tr>
</tbody>
</table>
PANGEO DEPLOYMENTS

NASA Pleiades

NCAR Cheyenne

kubernetes

Google Cloud Platform

Jetstream

Microsoft Azure

aws

http://pangeo.io/deployments.html
CONTINUOUS DEPLOYMENT

- https://github.com/pangeo-data/pangeo-cloud-federation
- Cloud-based clusters managed with helm / kubernetes
- Deployment is completely automated via GitHub / circleci
- Resources scale elastically with demand
CLOUD DATA CATALOG

- [https://pangeo-data.github.io/pangeo-datastore/](https://pangeo-data.github.io/pangeo-datastore/)

- Datasets stored in **zarr** format (cloud-native HDF-replacement)

- Cataloged using **intake**

- Automated testing of datasets
CLOUD COSTS
DEMO

https://tinyurl.com/pangeo-ocean
<table>
<thead>
<tr>
<th>Access</th>
<th>Government HPC</th>
<th>Commercial Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✅ Available to all federally funded projects</td>
<td>✅ Available globally to anyone with a credit card</td>
</tr>
<tr>
<td></td>
<td>✗ Available <em>only</em> to federally funded projects</td>
<td>✗ Authentication is not integrated with existing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>research infrastructure</td>
</tr>
<tr>
<td>Cost</td>
<td>✅ Cost is hidden from researchers and billed by</td>
<td>✗ Cost is borne by individual researchers and hidden</td>
</tr>
<tr>
<td></td>
<td>funding agencies</td>
<td>from funding agencies</td>
</tr>
<tr>
<td></td>
<td>✗ Allocations, quotas, limits</td>
<td>✅ Economics of scale, unlimited resources</td>
</tr>
<tr>
<td>Compute</td>
<td>✅ Homogeneous, high performance nodes</td>
<td>✅ Flexible hardware (big, small, GPU)</td>
</tr>
<tr>
<td></td>
<td>✗ Queues, batch scheduling, ssh access</td>
<td>✅ Instant provisioning of unlimited resources</td>
</tr>
<tr>
<td></td>
<td>✗ Fixed-size compute</td>
<td>✅ Spot market: burstable, volatile</td>
</tr>
<tr>
<td>Storage</td>
<td>✅ Fast parallel filesystems (e.g. GPFS)</td>
<td>✅ Fast object storage</td>
</tr>
</tbody>
</table>
HOW TO GET INVOLVED

HTTP://PANGEO.IO

• Use and contribute to xarray, dask, zarr, jupyterhub, etc.

• Access an existing Pangeo deployment on an HPC cluster, or cloud resources (http://pangeo.io/deployments.html)

• Adapt Pangeo elements to meet your projects needs (data portals, etc.) and give feedback via github: github.com/pangeo-data/pangeo

• Provide data in a cloud-optimized format
The following slide is a backup in case live demo is not possible.
```python
[5]: <xarray.DataArray 'SST' (time: 14965, nlat: 2400, nlon: 3600), dtype=float32, chunksize=(1, 2400, 3600)>
   Coordinates:
   * time (time) float64 1.679e+04 1.679e+04 ... 3.175e+04 3.176e+04
   * nlon (nlon) int64 0 1 2 3 4 5 6 7 ... 3593 3594 3595 3596 3597 3598 3599
   * nlat (nlat) int64 0 1 2 3 4 5 6 7 ... 2393 2394 2395 2396 2397 2398 2399
   lon (nlat, nlon) float64
dask.array=shape=(2400, 3600), chunksize=(2400, 3600)>
   lat (nlat, nlon) float64
dask.array=shape=(2400, 3600), chunksize=(2400, 3600)>
Attributes:
   cell_methods: time: mean
   grid_loc: 2110
   long_name: Surface Potential Temperature
   units: degC

[ ]: %output holomap='scrubber' fps=1

sst_ds = hv.Dataset(sst, kdims=['time', 'nlon', 'nlat'])
hv_sst = sst_ds.to(hv.Image, kdims=['nlon', 'nlat'], dynamic=True)

%opts Image [width=700 height=400] {cmap='magma'}
regrid(hv_sst, precompute=True)
```
• 12 GB / 24 hours -> must be a mistake