Modernizing a Bias Correction Code

Kenton Wu, Thomas Cram, Riley Conroy, Dr. Cindy Bruyère

BACKGROUND

A methodology proposed by Bruyère et al. (2015) entails correcting the spatial bias in Global Climate Models (GCMs) in order to force Regional Climate Models (RCMs). This was done specifically for the CESM1 GCM under Coupled Model Intercomparison Project 5 (CMIP5) specifications. With the release of CMIP6, the method's NCAR Command Language/Fortran implementation has become outdated. Our goal is to **investigate whether Python re-implementations can improve accessibility, generality, and scalability while maintaining the efficiency and accuracy of the legacy bias correction code.**

METHODS

Accessibility/Generality

Xarray and **cf-xarray** were chosen to interface with data due to positionless indexing, standard name indexing, and label-based alignment.

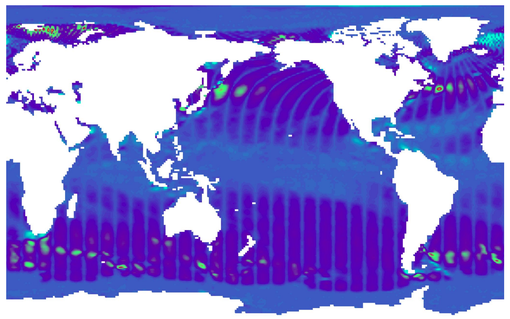
Accuracy/Scalability

Mitigation of Python overhead was investigated by **vectorizing** algorithms through combinations of **NumPy** and the **Numba JIT compiler**, as well as choosing packages such as **GeoCAT** and **xESMF** with **Dask compatibility.** Performance and accuracy were evaluated on a small subset of CMIP5 CESM data. Timings were reported by the *timeit* utility, on a Casper login node, and the NCL cpu timer.

OBJECTIVES

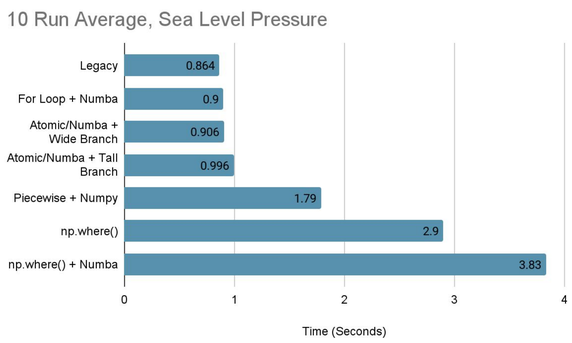
- Identify major geoscience calculations performed by legacy NCL/Fortran code:
  - Regridding of displaced pole grid from ocean model to lat/lon atmospheric model
  - Pressure levels on hybrid levels
  - ECMWF formulation of sea level pressure
  - Geopotential on hybrid levels from temperature
  - Interpolation of variables to hybrid levels
- Match calculations to external libraries
- Write in functions with no matching library
- Assess error when compared to original outputs
- Assess runtimes of serial execution relative to legacy code
- *Minor objective: ability to ingest generic cf-formatted data*

RESULTS



Absolute difference in regriddings of sea surface temperature done by xESMF. The maximal difference is 0.003, or relatively 5 digits of precision. This is two orders of magnitude above single precision (7 digits). Note the striations which are an artifact of floating point errors magnifying far away from the data. This is also evidenced by magnified errors in regions of high gradient.

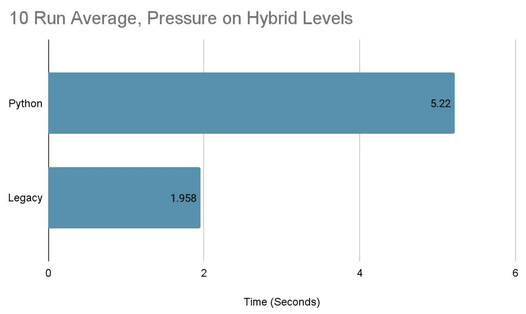
Not pictured: Maximal error for pressure on hybrid levels: 0.003, Maximal error for pslec: 0.007, both similarly around two orders of magnitude above machine precision



Top to bottom:

1. NCL with Fortran backend
2. One-to-one for loop accelerated with Numba njit compiler
3. Element wise accelerated with Numba vectorize compiler and a shallow branch structure
4. Same as 3. but with the original branch structure
5. Conditionals replaced by logical indexing with boolean masks
6. Using np.where
7. np.where with Numba njit

Not pictured: cz2ccm, which is not completed at this time, as well as interpolation to hybrid levels and relative humidity which are one-to-one functions provided by GeoCAT.



Legacy calculation speed as compared to an atomic function accelerated with Numba's vectorize.

CONCLUSIONS

The reported functions are shown to be reasonably within machine precision of legacy results. An atomic design pattern vectorized using Numba appears to strike a balance between speed, adaptability, and readability. No functions perform as fast as in the legacy code.

However, the nature of the code is embarrassingly parallel across time. The natural integration with Dask should theoretically allow a future implementation to scale linearly and perform better overall.

ACKNOWLEDGMENTS

I would like to acknowledge Anissa Zacharias and Orhan Eroglu of the GeoCAT team, as well as David Stepaniak from DECS for their guidance and indelible debugging skill. Of course, none of this would be possible without the amazing people running the SiParCS program, such as Virginia Do and Kristen Pierri.