Modernizing a Bias Correction Code

Kenton Wu[1],

Thomas Cram[2], Riley Conroy[2], Dr. Cindy Bruyère[2]

University of Texas at Austin[1], SiParCS Intern[1], NCAR[2]

August 2, 2023

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Background

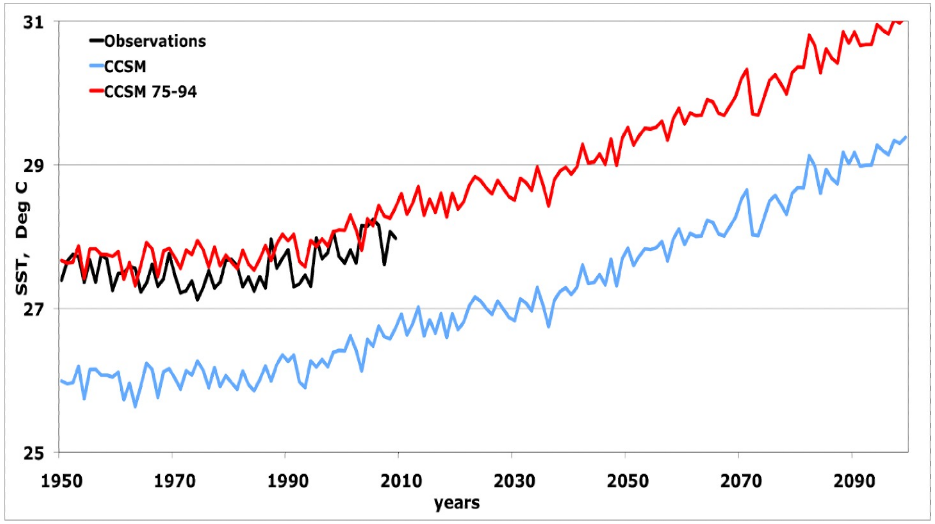
- How do we help regional scientists and policy makers make informed decisions as the climate changes?

- It's hard to look small and look far
  - Model insufficiencies
  - Numerical stability (Courant-Friedrichs-Lax condition)

- Leads to unavoidable biases

Bruyere’s Proposal

- Observed that the bias was not greatly time-varying
- So just correct forecasts with the averages of high-fidelity data, then use corrected forecast to force regional simulations



Time for a Makeover?

- Original code written in Fortran and NCL
- NCL is now deprecated
- Very little support for Fortran
- Lack of modern features
- Good timing for a refresh with the CMIP6 refresh

```
DO K = KMAX + 1,1,-1
DO I = 1,IMAX
ARG = HPRB*HYBA(2,KMAX-K+2) + PS(I)*HYBB(2,KMAX-K+2)
IF (ARG.GT.0.D0) THEN
PMLN(I,K) = DLOG(ARG)
ELSE
PMLN(I,K) = 0.0D0
END IF
END DO
END DO

DO K = 2,KMAX - 1
DO I = 1,IMAX
PTERM(I,K) = RBYG*TV(I,K)*0.5D0* (PMLN(I,K+1)-
PMLN(I,K-1))
END DO
END DO
DO K = 1,KMAX - 1
DO I = 1,IMAX
Z2(I,K) = PHIS(I)/G0 + RBYG*TV(I,K)*0.5D0*
+ (PMLN(I,K+1)-PMLN(I,K))
END DO
END DO


K = KMAX
C 3.a.109.5
DO I = 1,IMAX
Z2(I,K) = PHIS(I)/G0 + RBYG*TV(I,K)*
+ (DLOG(PS(I)*HYBB(1,1))-PMLN(I,K))
```

For Loops Considered Harmful

Interpreted loops are verrry slow….

Saved by the JIT

JITs are in between compilers and interpreters

The Job Description

- Calculate pressure on hybrid coordinates
- Calculate sea level pressure
- Calculate geopotential height from temperature
- Relative humidity
- Interpolate to hybrid coordinates

Sea Level Pressure

Naive

```
for i …
        for j…
                if
                else …
```

Where()

```
where(this is true,
calculate this,
else calculate this)
```

Sea Level Pressure

Piecewise

```
for every case:

    mask = where array meets condition

    answer[mask] = math(variable[mask])
```

Atomic + Numba vectorize

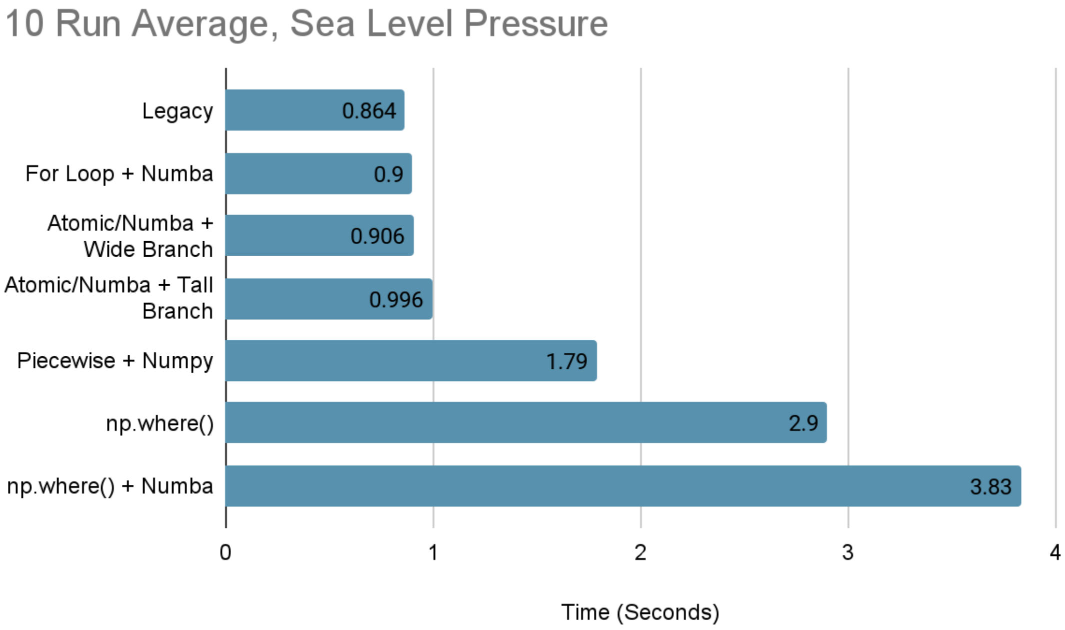
```
@vectorize
function atomic(scalar):
        return math(scalar)

answer = apply_everywhere(function,variable)
```

Sea Level Pressure

Naïve + Numba njit

```
@njit
for …
```



Future Work

- Dask parallelization
- Rewrite cz2ccm
- Finish up actual bias correction

Acknowledgements