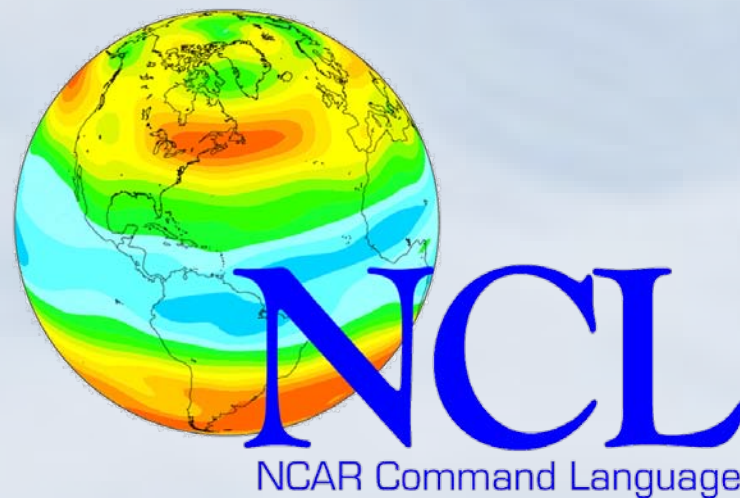


# Ocean model diagnostics package with NCL

Dave Brown  
NCL Development Team



# Topics

- Motivation
- Background and current status
- Implementation
- Comparison of graphical output
- Future plans
- Summary

**This work is part of the ParVis project which is sponsored by the Earth System Modeling program of the Office of Biological and Environmental Research of the U.S. Department of Energy's Office of Science. PI is Rob Jacob of ANL with collaborators from NCAR, PNNL, SNL and UC-Davis.**



# Goals

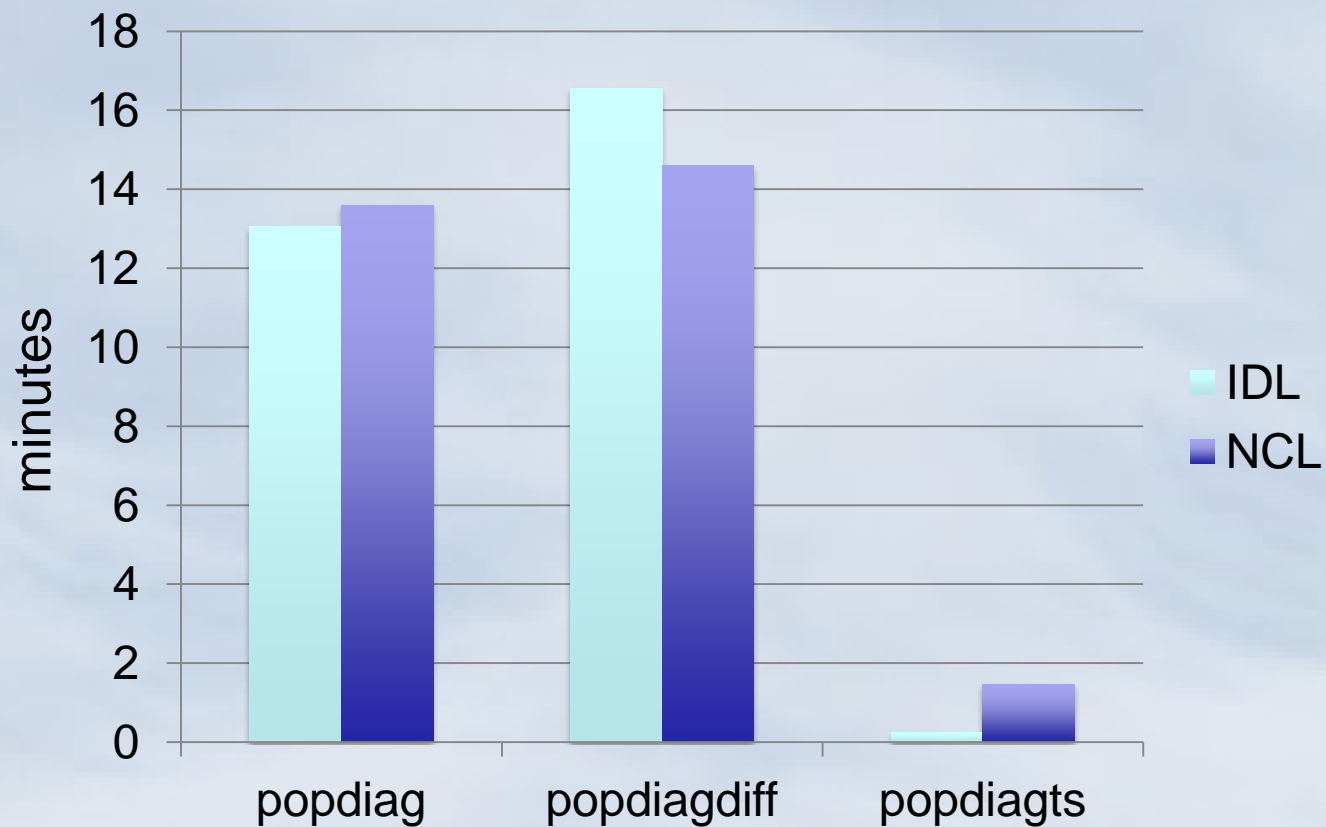
- Provide the community with a completely open-source diagnostics package with an option for dramatically improved performance
- Overall ParVis goal is to provide parallel-processing solutions for the big data problem facing climate researchers
- Short term: use Swift, a task-parallel scripting tool to improve performance for existing tasks
- Longer term: develop ParNCL, a parallel version of NCL

# Historical Background

- AMWG diagnostics historically used NCO for data reduction, NCL for analysis and viz, and C-shell scripts as a top-level driver
- Converted to use Swift originally by John Dennis
- Swift modified to accommodate the diagnostic package work flow.
- Provided a significant speed-up (~ 3x)
- The primarily IDL-based ocean diagnostics appeared to offer a good opportunity to extend this work
- Two part task:
  - Convert IDL code to NCL (my part)
  - Implement Swift code to manage the scripts in parallel

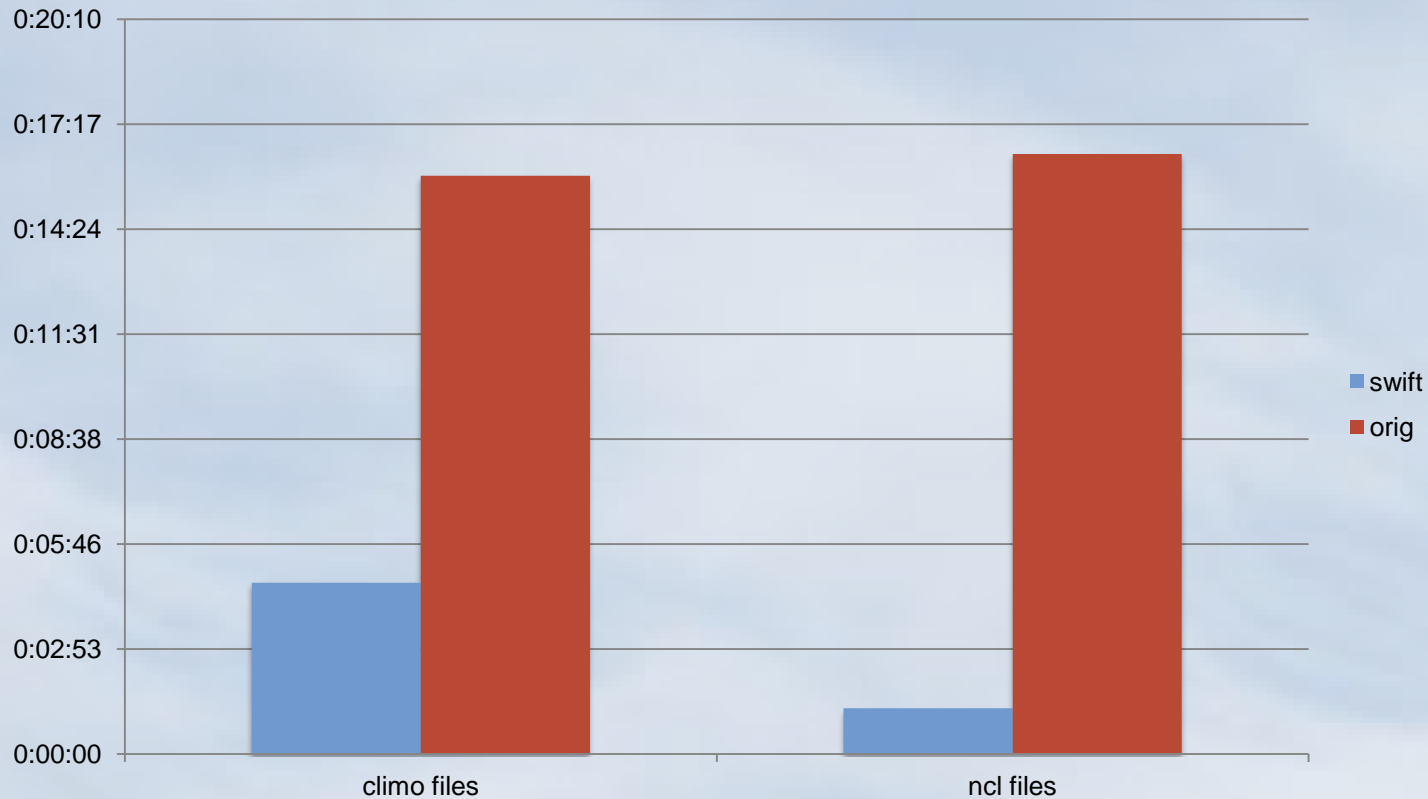
# OMWG diagnostics status

- All scripts have been converted
- 79 top-level scripts; 13 other functions
  - popdiag (39) -- 1 pre-existing (dwbc.ncl)
  - popdiagdiff (28)
  - popdiagts (13) – 1 pre-existing (enso\_wavelet\_asc.ncl)
- 2 Fortran routines added as built-in NCL functions
  - mixed\_layer\_depth
  - wgt\_area\_smooth – 2D 5-point area-weighted smoothing routine
- Without Swift, NCL and IDL timings are for the most part similar
- Basically transparent to user – scripts work the same as they always have



Old code vs. new code (without Swift)

## POPD DIAG - Original vs. Swift on lens

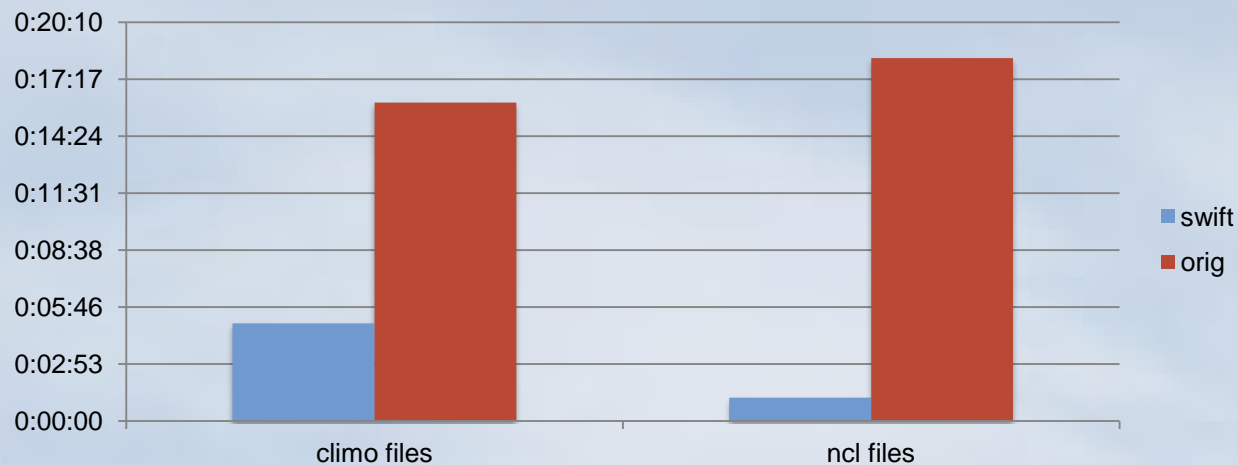


4 nodes – 8 tasks per node maximum – using lens at ORNL  
climo: averaging and other pre-processing  
HPSS access and image conversion not included  
10-year comparison

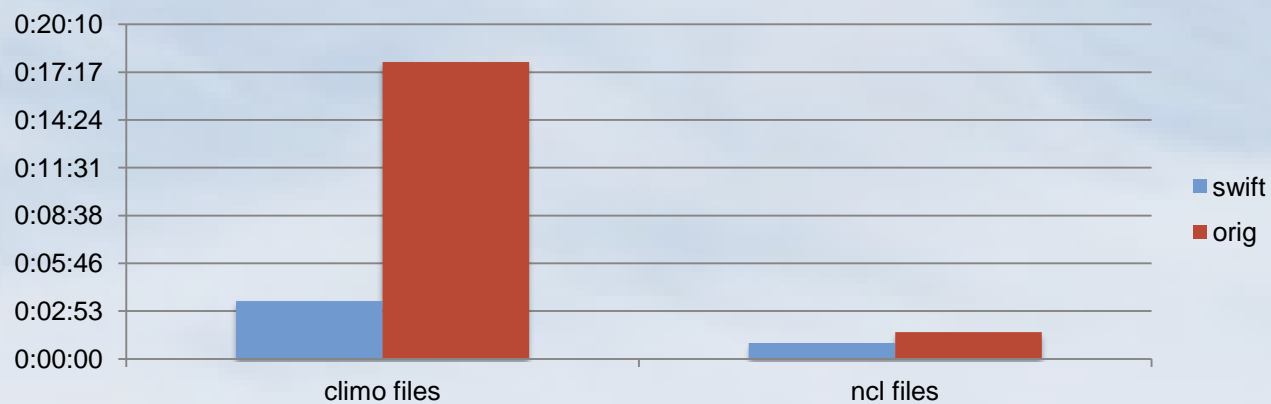
Courtesy Sheri Mickelson



## POPDIAGDIFF - Original vs. Swift on lens



## POPDIAGTS - Original vs. Swift on lens



Courtesy Sheri Mickelson

# Swift tutorial: 6:00 PM

## Aspen/Blue Spruce room

Sheri Mickelson and Rob Jacob will be giving a workshop this evening at 6:00 PM on how to use the task-parallel versions of both the OMWG and AMWG diagnostic packages.

I will be helping out as well.

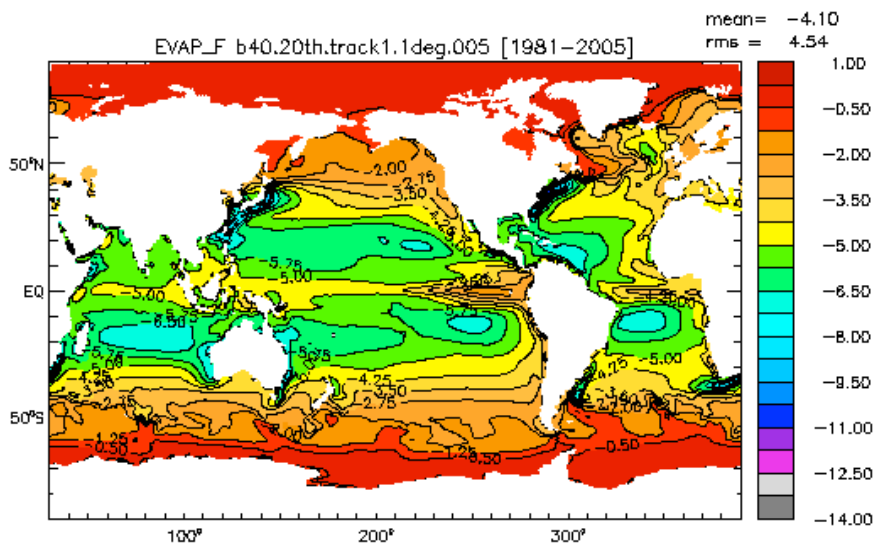
# Conversion project implementation

- Conservative approach (not an oceanographer!)
- Results must match numerically and graphically
- Therefore initial version retains original colormaps, contour levels, and line colors for ease of verification
- Similar positioning of annotations, but allowed some variation in font styles, etc.
- Fairly literal code translation where performance not affected
- Array arithmetic used more aggressively since NCL looping performance is slower than IDL

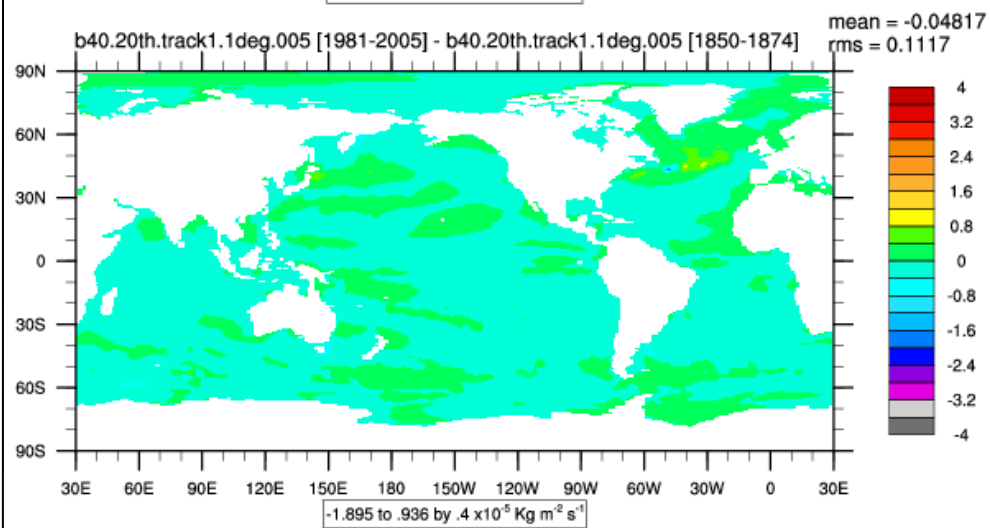
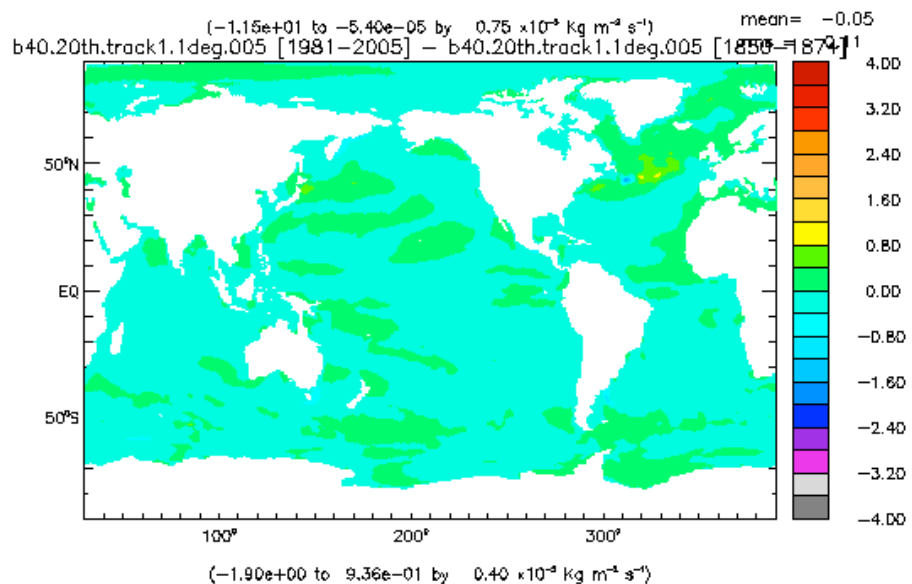
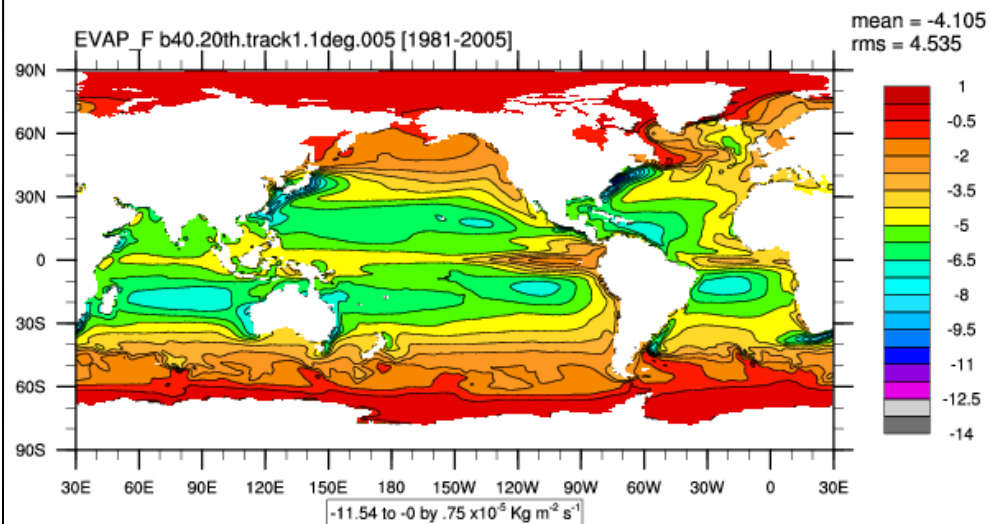
# OMWG diagnostic output comparisons

- NCL output online:
  - [http://www.ncl.ucar.edu/Applications/popdiag/pd.1981\\_2005/popdiag.html](http://www.ncl.ucar.edu/Applications/popdiag/pd.1981_2005/popdiag.html)
- Current IDL output online:
  - [http://www.cesm.ucar.edu/experiments/cesm1.0/diagnostics/b40.20th.track1.1deg.005/ocn\\_1981-2005-obs/popdiag.html](http://www.cesm.ucar.edu/experiments/cesm1.0/diagnostics/b40.20th.track1.1deg.005/ocn_1981-2005-obs/popdiag.html)
- Some samples follow:

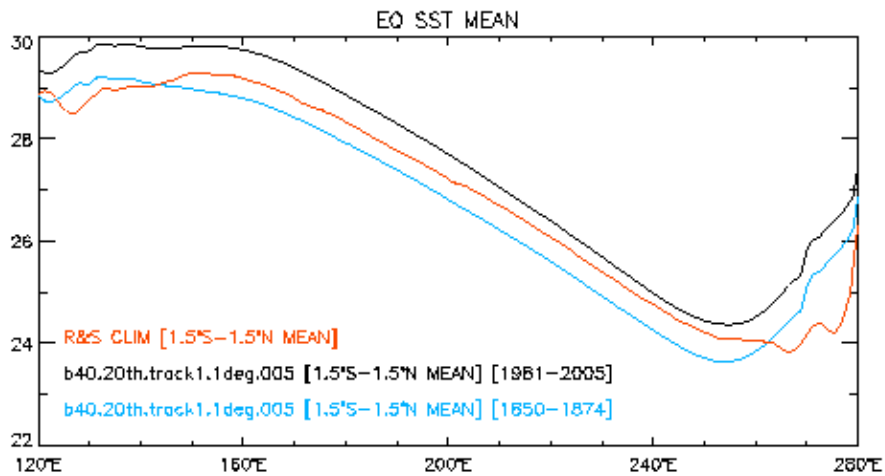
# IDL:



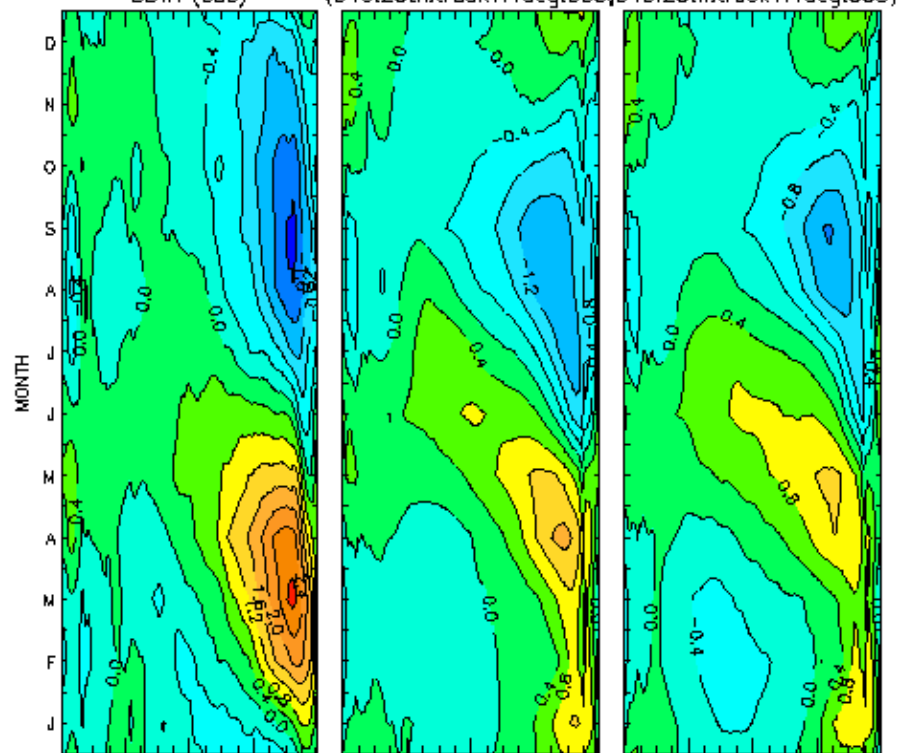
# NCL:



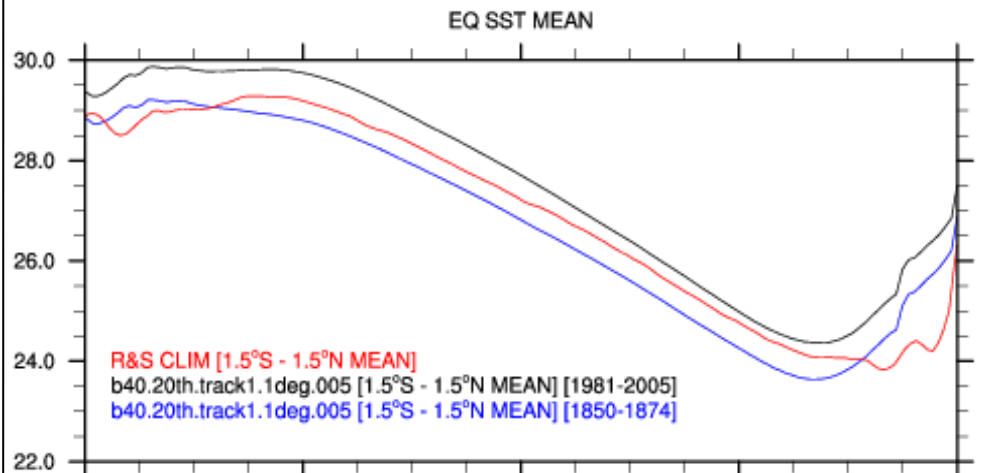
# IDL:



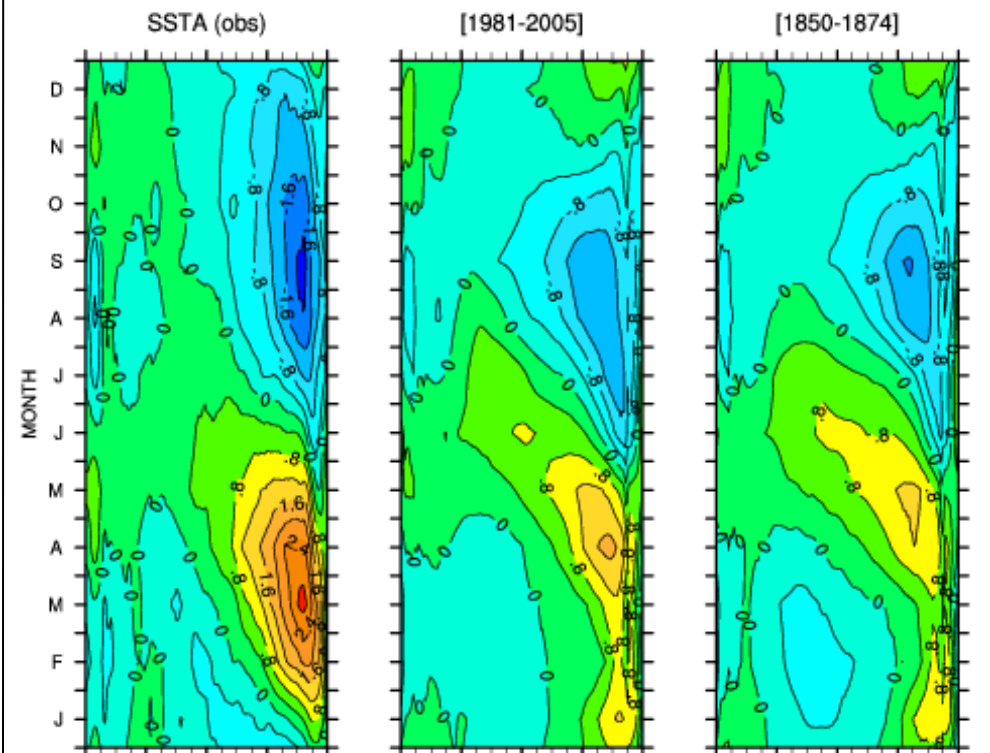
SSTA (obs) (b40.20th.track1.1deg.005) (b40.20th.track1.1deg.005)



# NCL:

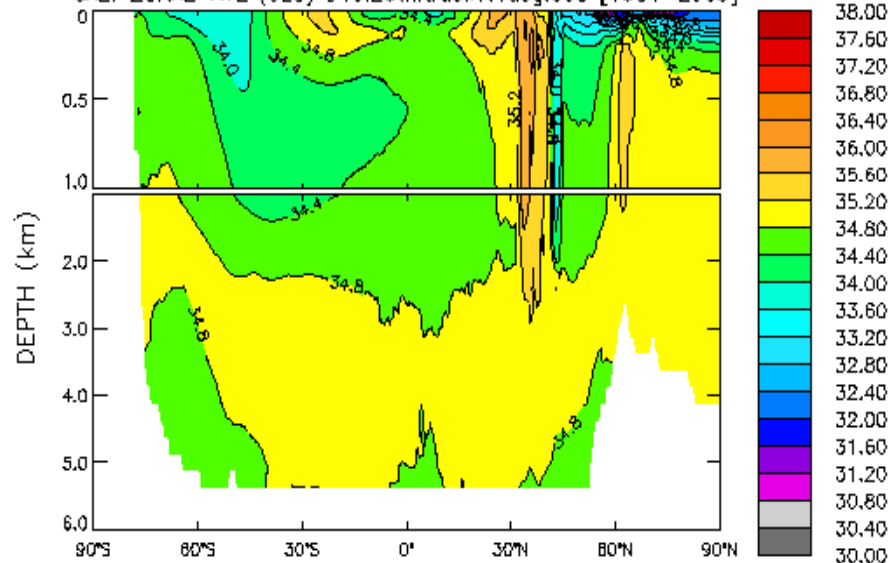


SSTA (obs) [1981-2005] [1850-1874]

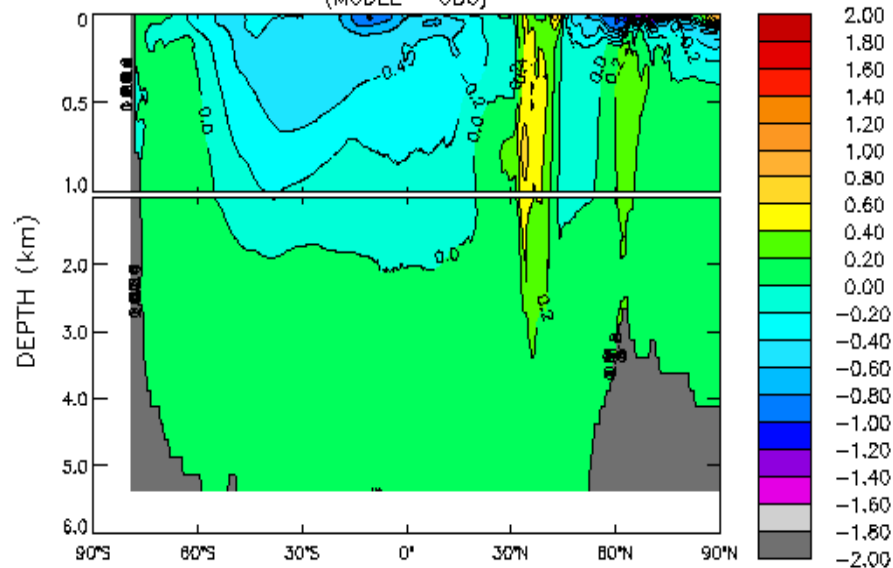


# IDL:

SALT ZONAL-AVE (GLO) b40.20th.track1.1deg.005 [1981-2005]



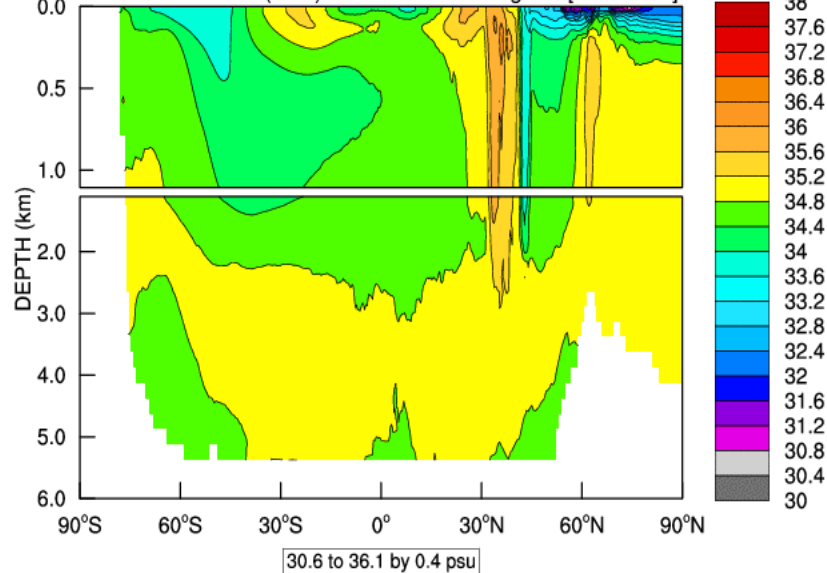
( 3.06e+01 to 3.61e+01 by 0.40psu)  
(MODEL - OBS)



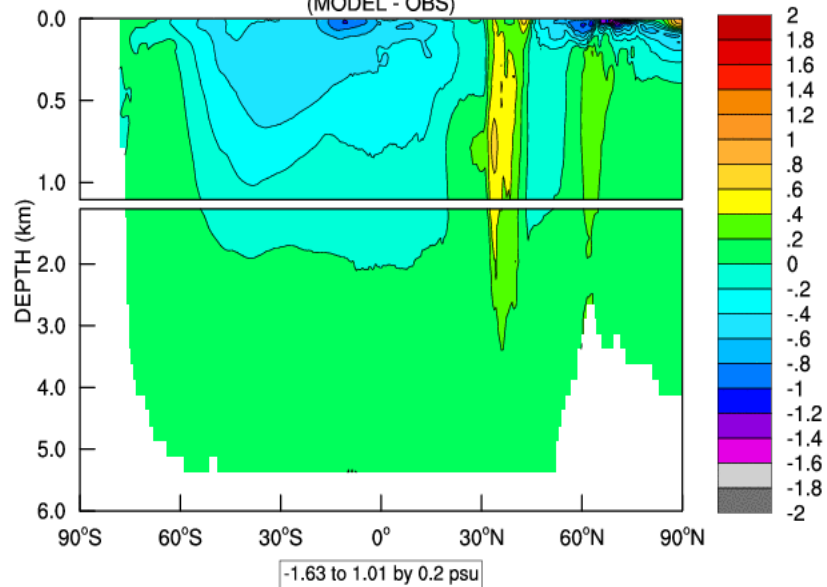
(-1.63e+00 to 1.01e+00 by 0.20psu)

# NCL:

SALT ZONAL-AVE (GLO) b40.20th.track1.1deg.005 [1981-2005]

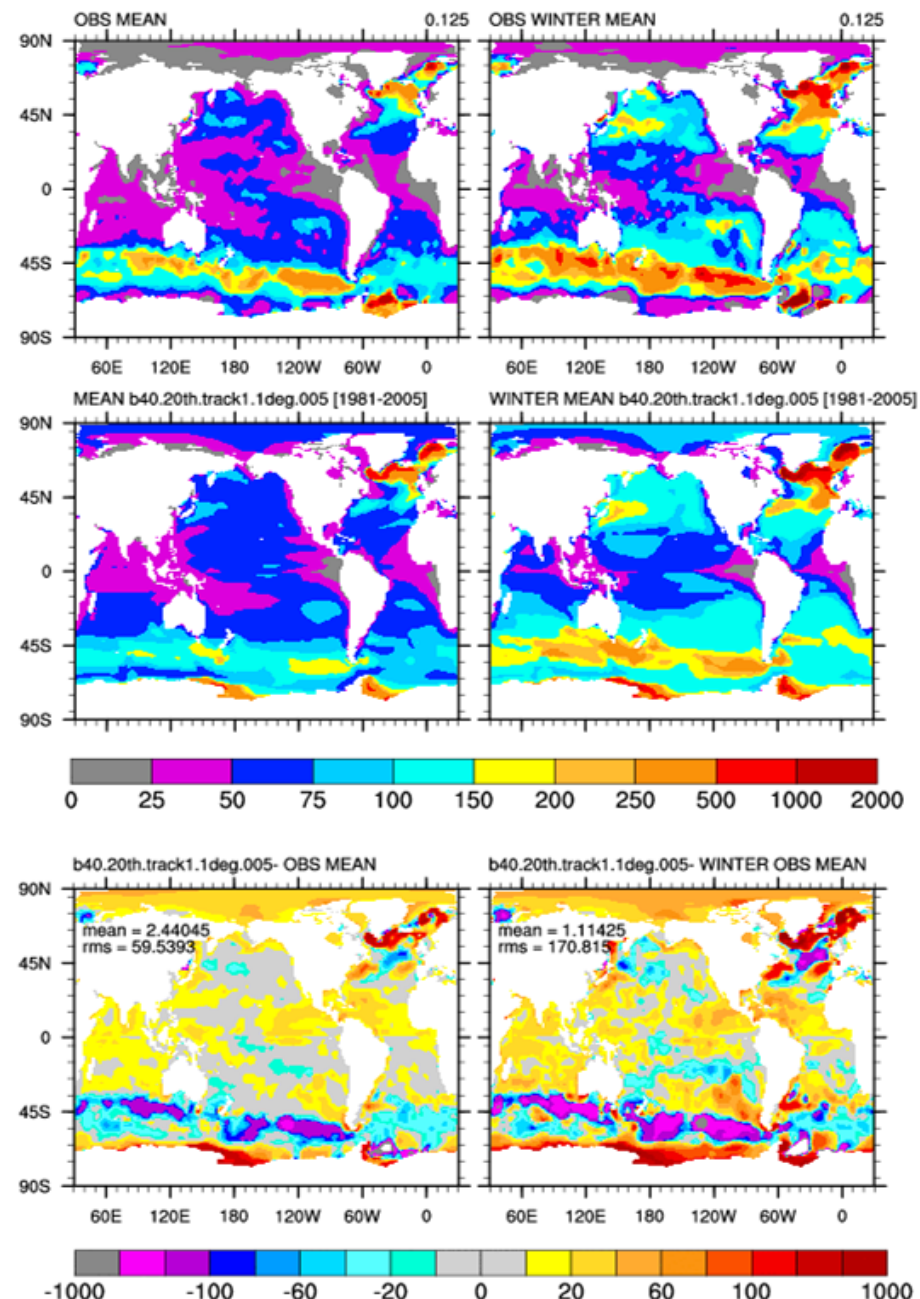
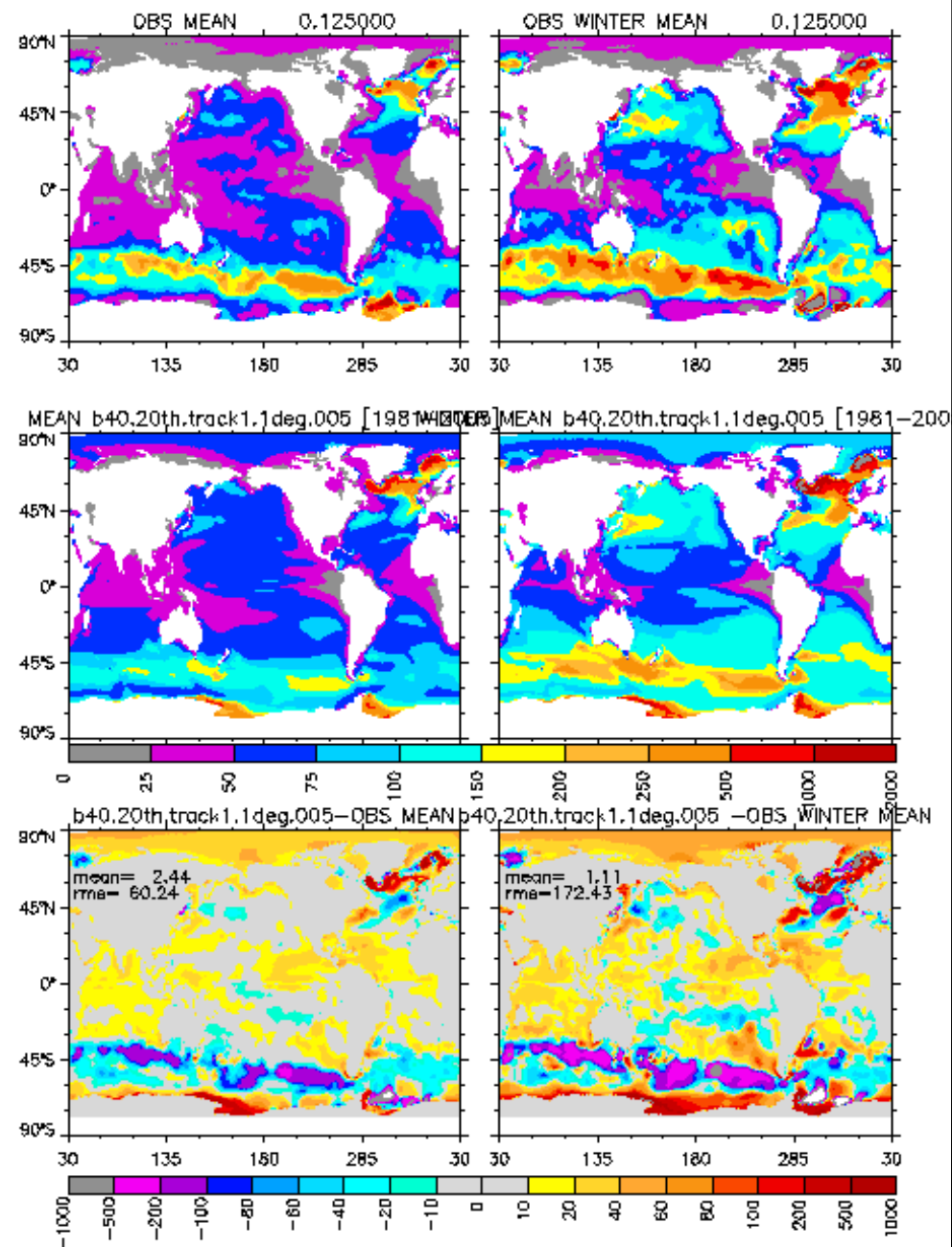


(MODEL - OBS)



# IDL:

# NCL:





# Future plans

- Verify and improve diagnostic suite to work with high-resolution versions of the model
- Deploy ParNCL to improve performance further
- Possible enhancements to graphics
  - Configurable color tables
  - Configurable level settings
  - Allow for zooming into regions with levels adapting to regional min/max data extent
- Experimental web-based version using a Python framework currently under development by summer intern
  - Replaces csh top-level scripts with Python

# Summary

- OMWG diagnostic suite soon to be available at
  - [https://svn-ccsm-release.cgd.ucar.edu/model\\_diagnostics/ocn/pop2/popdiag\\_rel\\_<yearmndy>](https://svn-ccsm-release.cgd.ucar.edu/model_diagnostics/ocn/pop2/popdiag_rel_<yearmndy>)
- Freely distributable open source
- Graphics equal to or better than existing suite
- Similar performance in single processor mode; much faster with Swift enabled
- Future improvements possible
- Suggestions welcome

Thanks to Susan Bates and Gokhan Danabasoglu for their support of this project, Sheri Mickelson for technical assistance, and Mary Haley and Dennis Shea for their good advice.

Questions?

Email: [dbrown@ucar.edu](mailto:dbrown@ucar.edu)



# NCL vs. IDL

- Many apparent similarities
  - ; (semicolon) starts a comment
  - Fortran-like syntax features: e.g. .eq. (NCL), eq (IDL)
  - Overall verbosity (lines of code): 14424 (NCL), 14388 (IDL)
  - Similar array syntax: 0-based element counting
- Significant differences
  - NCL: row-major like C; IDL: column-major like Fortran
  - Graphics code has a different model
  - NCL's built-in support for missing values helps simplify code
  - NCL's NetCDF-like variable model allows easier access for attributes and other metadata
  - IDL looping is definitely faster (script is compiled)
  - (Therefore) more important to use array syntax in NCL