The Future of the AMWG diagnostics package?

Cécile Hannay, Brian Medeiros, Jesse Nusbaumer
with contributions of Dani Coleman, Matt Long, and AMP/CGD.

March 11, 2020
Motivation

• NCL is going away
• NCAR is transitioning to Python for diagnostics
• How is going to affect the AMWG diagnostics?

Outline

• What are we using now?
• What are our needs for the future?
• How does it fit in the NCAR vision for diagnostics?
  (Other packages in development)
What are we using now?
The NCAR Command Language (NCL)

NCL is an interpreted language designed specifically for scientific data analysis and visualization.

Pipeable, robust, and free, NCL is available as binaries or open source.

Supports netCDF 3/4, GRIB 1/2, HDF 4/5, HDF-EOS 2/5, shapefile, ASCII, binary.

Numerous analysis functions are built-in.

High-quality graphics are easily created and customized with hundreds of graphic resources.

Many example scripts and their corresponding graphics are available.

Find a plot I like

Script to do that plot

Look at the examples

GeoCAT (https://geocat.ucar.edu) python tools related to NCL

NCL is retiring

Check examples

https://www.ncl.ucar.edu/
The AMWG Diagnostics package

Monthly means
Read files
Compute Climos
Regrid Climos
Plots
Webpage

cshell script calling
• NCO
• NCL
• ImageMagic
The AMWG Diagnostics package

Overview

The AMWG diagnostics package produces over 600 plots and tables from CCSM (CAM) monthly netcdf files.

The diagnostics package computes climatological means of the simulations and produced plots and tables of the mean climate in a variety of formats. The diagnostics package can be used to compare two CCSM (CAM) model simulations or for comparing a model simulation to the observational and reanalysis data. (Information about the AMWG datasets can be found in the Climate Data Guide.)

Example webpages created by the diagnostics package

Included in the package are HTML files which provide the infrastructure for a basic website for the display of all your plots and tables. The c-shell script has a switch for creating webpages automatically. When this is used the end result of running the script is a tar file of all the plots in gif, jpg or png format and the needed html files organized in the proper subdirectories. The user can then untar this file in a directory of their choosing and create a link to it.

- Model fields compared with observational data plots
- Comparison of two different models plots
- CAM-chem diagnostics plots

http://www.cesm.ucar.edu/working_groups/Atmosphere/amwg-diagnostics-package/
The CESM postprocessing package

- Common framework that produces diags for all components
- Produced CMIP6 diags automatically
- Well documented

- Complex framework to ultimately run NCL scripts
- Very complex (for scientists)
- Not portable (only run on Cheyenne)
- Dependency on outdated packages (python2, basemap,...)
- Not more support

What are our needs for the future?
<table>
<thead>
<tr>
<th>Package to have a quick look at simulations</th>
<th>Bank of scripts to replace NCL examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In absence of support + NCL going away:</strong></td>
<td></td>
</tr>
<tr>
<td>- Simple framework python based</td>
<td></td>
</tr>
<tr>
<td>- Scientists can understand and modify</td>
<td></td>
</tr>
<tr>
<td>- Portability (can run anywhere)</td>
<td></td>
</tr>
<tr>
<td>- Small numbers of python packages</td>
<td></td>
</tr>
</tbody>
</table>
What are our needs and where do we want to go from here?

Package to have a quick look at simulations

- Monthly means
  - Read files
    - xarray
    - numpy
    - scipy
  - Compute Climos
    - xesmf
  - Regrid Climos
    - matplotlib
    - cartopy
  - Plots
    - yaml

Bank of scripts to replace NCL examples

- How to run:
  - run_diag < namelist
    - namelist
      - casename
      - climo year

Code would live on github
- https://github.com/NCAR

Contributions would be through review process
What are our needs and where do we want to go from here?

Package to have a quick look at simulations

- Monthly means

<table>
<thead>
<tr>
<th>Library</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>xarray</td>
<td>Read files</td>
</tr>
<tr>
<td>numpy</td>
<td>Compute Climos</td>
</tr>
<tr>
<td>scipy</td>
<td>Regrid Climos</td>
</tr>
<tr>
<td>cartopy</td>
<td>Plots</td>
</tr>
<tr>
<td>xesmf</td>
<td></td>
</tr>
</tbody>
</table>

How to run:
```
run_diag < namelist
```

- namelist
  - casename
  - climo year

Code would live on github
[https://github.com/NCAR](https://github.com/NCAR)

Contributions would be through review process

Bank of scripts to replace NCL examples

- Collection of python scripts
- Jupyter Notebooks

An example of what it could be

[https://github.com/brianpm/hacknostics](https://github.com/brianpm/hacknostics)

- Some atmospheric diagnostics, focused on analysis of CESM.
- The repo is organized into Notebooks, utilities (util), and notes (docs).
What are our needs and where do we want to go from here?

Package to have a quick look at simulations

- Monthly means
- Read files
- Compute Climos
- Regrid Climos
- Plots

Bank of scripts to replace NCL examples

- Collection of python scripts
- Jupyter Notebooks

How to run:
run_diag < namelist

- namelist
  - casename
  - climo year

Code would live on github
https://github.com/NCAR

Contributions from the community welcome

Could be reorganized (like NCL Webpage)

Contributions would be through review process
What are our needs and where do we want to go from here?

Package to have a quick look at simulations

- Monthly means
- Read files
- Compute Climos
- Regrid Climos
- Plots

```yaml
How to run:
run_diag < namelist
```

- namelist
  - casename
  - climo year

Bank of scripts to replace NCL examples

- Collection of python scripts
- Jupyter Notebooks

Code would live on github
https://github.com/NCAR

Contributions from the community welcome

Could be reorganized (like NCL Webpage)

Contributions would be through review process

Deadline for first implementation: April 15, 2020
How does this fit in the bigger picture?
Pangeo: Core tools

A thin client
Interactive computing in the universal app: a browser; produce and publish “computational narratives.”

Python netCDF data model
Multi-dimensional datasets w/ coordinates; coordinate-aware selection and dim-reduction methods.

Parallel & “out-of-core” computation
Data bigger than memory; parallelized execution on a distributed resource “under the hood”!

Courtesy Matt Long
Vision: An interactive numerical laboratory for Earth system science

- Seamless integration of routine model evaluation and cutting-edge research
- Enable reproducible science
- Enable novel means of data interactivity and visualization
- Component models are not necessarily a natural organizing principle
  - Be as model agnostic as much as possible
- Scalable
  - Big Data
    - New applications, communities, etc.
- Fluid integration of observations and models
- Community-developed and open-source

Courtesy Matt Long
Vision: An interactive numerical laboratory for Earth system science

- **Analysis elements**: modular scripts or Notebooks that perform computation and support visualization
- **Data APIs**: abstract data access through APIs
- **Operators**: perform dimensions reductions, compute derived quantities, etc.
- **Visualization**: static plots, web apps
- **Workflow**: Automate, schedule, interact

- **Build library of analysis_element prototypes**
  - Aim for best practices (modularity!), but focus on scientific objectives and achieving basic function first
  - Coordinate sharing and standardization
  - Communicate!

- Reassess and coalesce as library grows

Matt will have a talk at OMWG workshop

Courtesy Matt Long
An open framework to run process-oriented diagnostics from research teams to inform model development.

- Focused on specific phenomena and process-level understanding
- Available for anyone to use the software package (python and ncl) to apply to a model simulation (cmip6 variables). Currently 7 packages. More coming including multi-model capability.

Diagnostics include:
- Cloud microphysical processes
- Tropical & extratropical cyclones
- ENSO teleconnections
- MJO moisture, convection, & radiative processes
- Precipitation diurnal cycle
- AMOC, Pacific sea level variability
- Arctic sea ice
- Lake effect processes
- North American Monsoon
- Radiative forcing & cloud-circulation feedbacks
- Temperature & precip. extremes

www.cesm.ucar.edu/working_groups/Atmosphere/mdtf-diagnostics-package/

Maloney et al. (2019, *BAMS*)

Courtesy Dani Coleman
Summary

• Package to have a quick look at diagnostics
  – simple, portable, small number of python packages

• Bank of python scripts and Jupyter Notebooks
  – Community contributions encouraged

• Code would live on github
  – https://github.com/NCAR

• Timeline
  – First implementation: April 15, 2020

• Inline with the NCAR vision for future diagnostics