Climate & Forecast (CF) Conventions for MPAS

Charlie Zender

Departments of Earth System Science and Computer Science

University of California, Irvine

Acknowledgements:

Wenshan Wang, Xylar Asay-Davis, Henry Butowsky, Daniel Lee, David Hassell, Brian Eaton

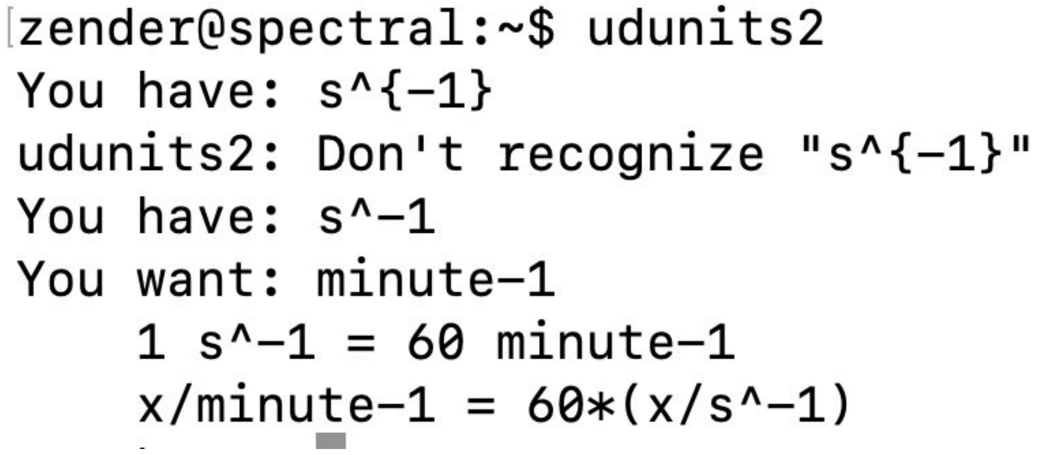
Support: DOE E3SM

CF History

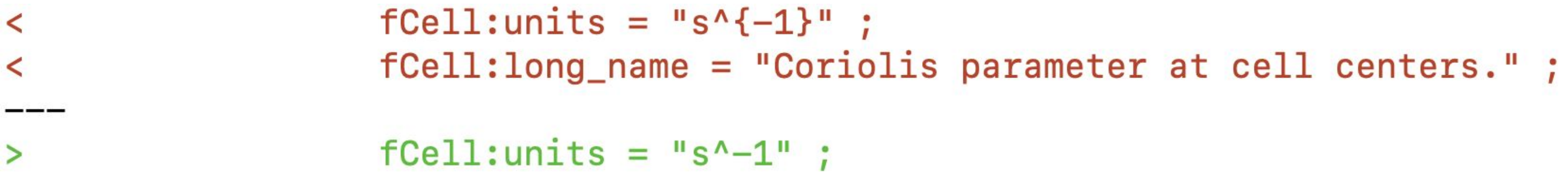
- Created in late 1990s by B. Eaton (NCAR), J. Gregory (U. Reading)
- Used in IPCC CMIP since SAR, leading to rapid adoption/expansion
- Self-organized, volunteer-based, hosted by PCMDI then GitHub
- Based on netCDF3 (classic) until about CF 1.8 in about 2017
- Annual meetings to spur discussion, governance

Units

- CF accepts UDUnits definitions for the dimensional `units` attribute
- Use udunits2 tool to check for compliance:



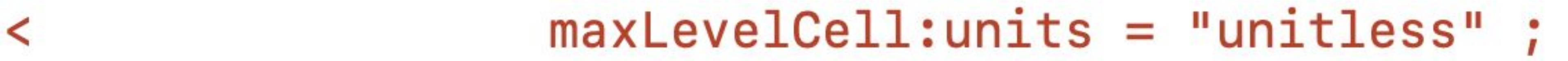
```
zender@spectral:~$ udunits2
You have: s^{-1}
udunits2: Don't recognize "s^{-1}"
You have: s^-1
You want: minute-1
    1 s^-1 = 60 minute-1
    x/minute-1 = 60*(x/s^-1)
```



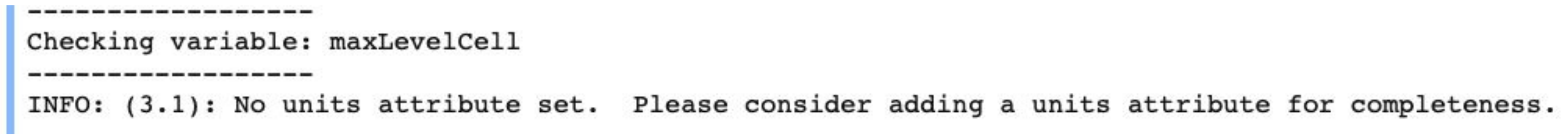
```
<               fCell:units = "s^{-1}" ;
<               fCell:long_name = "Coriolis parameter at cell centers." ;
---
>               fCell:units = "s^-1" ;
```

Units

- “A variable with no units attribute is assumed to be dimensionless.” However, eliminating units results in a (harmless) CF INFO message:

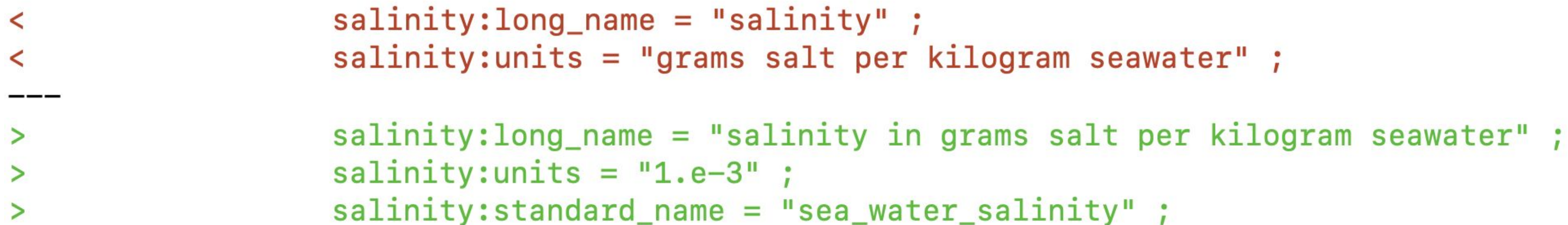


```
<                       maxLevelCell:units = "unitless" ;
```



```
------------------
Checking variable: maxLevelCell
------------------
INFO: (3.1): No units attribute set.  Please consider adding a units attribute for completeness.
```

- “The conforming unit for quantities that represent fractions, or parts of a whole, is "1". The conforming unit for parts per million is "1e-6".”



```
<                 salinity:long_name = "salinity" ;
<                 salinity:units = "grams salt per kilogram seawater" ;
---
>                 salinity:long_name = "salinity in grams salt per kilogram seawater" ;
>                 salinity:units = "1.e-3" ;
>                 salinity:standard_name = "sea_water_salinity" ;
```

Units

- CF prefers latitude, longitude use `degrees_north`, `degrees_east`
- Changing from `radians` perhaps fraught?

```
<                 latCell:units = "radians" ;
---
>                 latCell:units = "radians" ; // csz: CF prefers "degrees_north"
```

- Time units are not that bad…considering

```
>         double Time(Time) ;
>                 Time:long_name = "time" ;
>                 Time:units = "days since 1850-01-01 00:00:00" ;
>                 Time:calendar = "noleap" ;
>                 Time:bounds = "Time_bnds" ;
```

Units Uses

- Hyperslabbers might understand coordinate variables with UDUnits units:

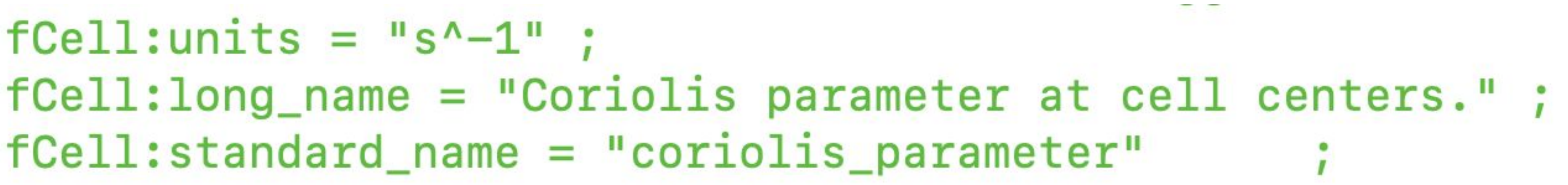
```
ncks -d Time,'2000-1-1','2001-12-31' in.nc out.nc
ncks -d lat,-10.,10. -d lon,180.,240. in.nc out.nc
```

- NCO understands lat/lon auxiliary coordinate variables in unstructured grids:

```
ncra -X -10.,10.,180.,240. in.nc out.nc
ncra --auxiliary -10.,10.,180.,240. in.nc out.nc
```

Standard Names

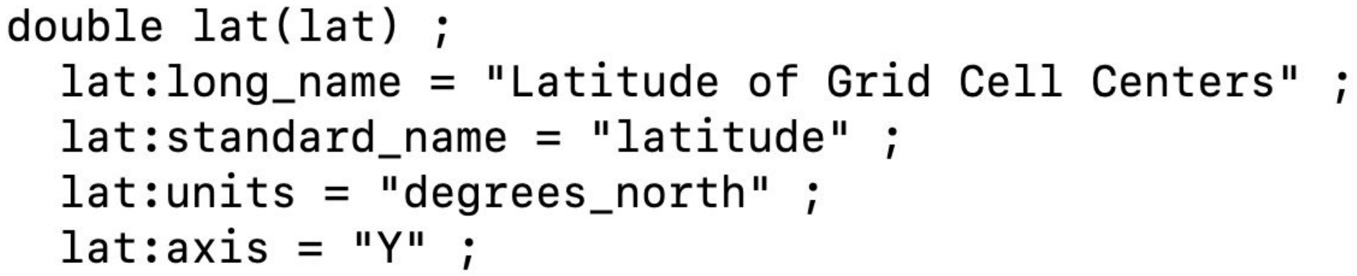
- CF defines names for common geophysical fields used in most ESMs
- Search Standard Name Table to find CF-equivalent to E3SM variable
- Store result in `standard_name` attribute
- Adopt “canonical units” if possible (not required)



```
fCell:units = "s^-1" ;
fCell:long_name = "Coriolis parameter at cell centers." ;
fCell:standard_name = "coriolis_parameter"      ;
```

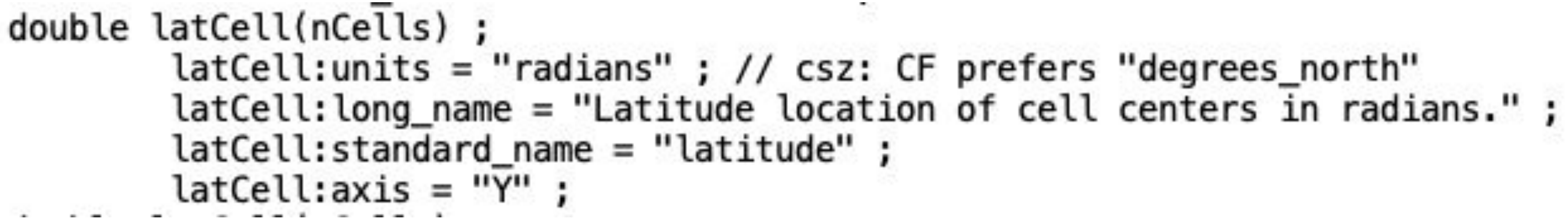
Coordinates

- Coordinate variable is 1D, same name as underlying dimension, monotonic, no missing values. Usually `time` and rectangular grids:



```
double lat(lat) ;
  lat:long_name = "Latitude of Grid Cell Centers" ;
  lat:standard_name = "latitude" ;
  lat:units = "degrees_north" ;
  lat:axis = "Y" ;
```

- Auxiliary coordinate variable can have any name, dimensionality. Required for curvilinear coordinate systems `lat(i,j)`, `lon(i,j)`



```
double latCell(nCells) ;
        latCell:units = "radians" ; // csz: CF prefers "degrees_north"
        latCell:long_name = "Latitude location of cell centers in radians." ;
        latCell:standard_name = "latitude" ;
        latCell:axis = "Y" ;
```

Coordinates

- Unstructured grids use horizontal cell number dimension (`col` or `nCells`) from which some coordinates cannot easily be determined. Solution: list auxiliary coordinate variables in `coordinates` attribute

```
 double temperature(Time,nCells,nVertLevels) ;
   temperature:coordinates = "latCell lonCell" ;
```

Cell Methods

Describe statistical properties along variable's dimensions: `min`, `max`, `mean`, `point` (i.e., instantaneous or location), `sum` (extensive). Use `area` instead of `lat: lon:`. Add `where` clause to restrict area-type:

```
    double snow_flux(Time,nCells) ;
     snow_flux:cell_methods = "Time: mean area: mean where sea" ;
     snow_flux:cell_methods = "Time: mean area: mean where sea_ice" ;
     snow_flux:cell_methods = "Time: mean within years Time: mean over years area: mean where sea" ;
```

Cell Measures

Identifies variables that contain cell area or volume with `area` or `volume`, respectively. Helps user determine how to properly weight statistics (means, integrals):

```
    double temperature(Time,nCells,nVertLevels) ;
      temperature:coordinates = "latCell lonCell" ;
      temperature:cell_methods = "Time: mean area: mean
where sea" ;
      temperature:cell_measures = "area: areaCell" ;
     double areaCell(nCells) ;
      areaCell:standard_name = "cell_area" ;
      areaCell:coordinates = "latCell lonCell" ;
```

Resources

CF Conventions here in HTML and here in PDF

CF Standard Names are here

UDUNITS Documentation is here in HTML

CF Checker (from NCAS) is here

NCO User Guide is here in HTML and here in PDF

That's all Folks!