grib2io

a Python interface to the NCEP GRIB2 Library (g2c)

Eric Engle

NOAA / NWS / MDL / Statistical Modeling Division

August 23, 2023

9th NOAA Ensemble Users Workshop

Outline

- Motivation
- History
- Collaboration
- Package Structure
- Key Features
- Live Demo *(Reading GEFS data via the xarray backend)*

Motivation

- MDL's National Blend of Models (NBM) is undergoing a major software transition from the Fortran-based MOS-2000 Software System to a new post-processing system written in Python

- NBM ingests GRIB2 data from ***many*** Numerical Weather Prediction (NWP) systems

- pygrib/ecCodes is restricted on NCEP Operational Supercomputer

- ***But we do have the NCEP g2c library!***

History

- Initial development under name of “ncepgrib2” from pygrib package by Jeff Whitaker (NOAA/OAR/PSL)
- Took over development in 2020 and renamed to “grib2io”
- ***v1.x:*** Mainly a cleanup of existing ncepgrib2 code; some performance improvements; introduction of NCEP GRIB2 tables
- ***v2.0:*** Major code refactoring to improve performance and efficiency; xarray backend; interpolation *(via grib2io-interp component package)*

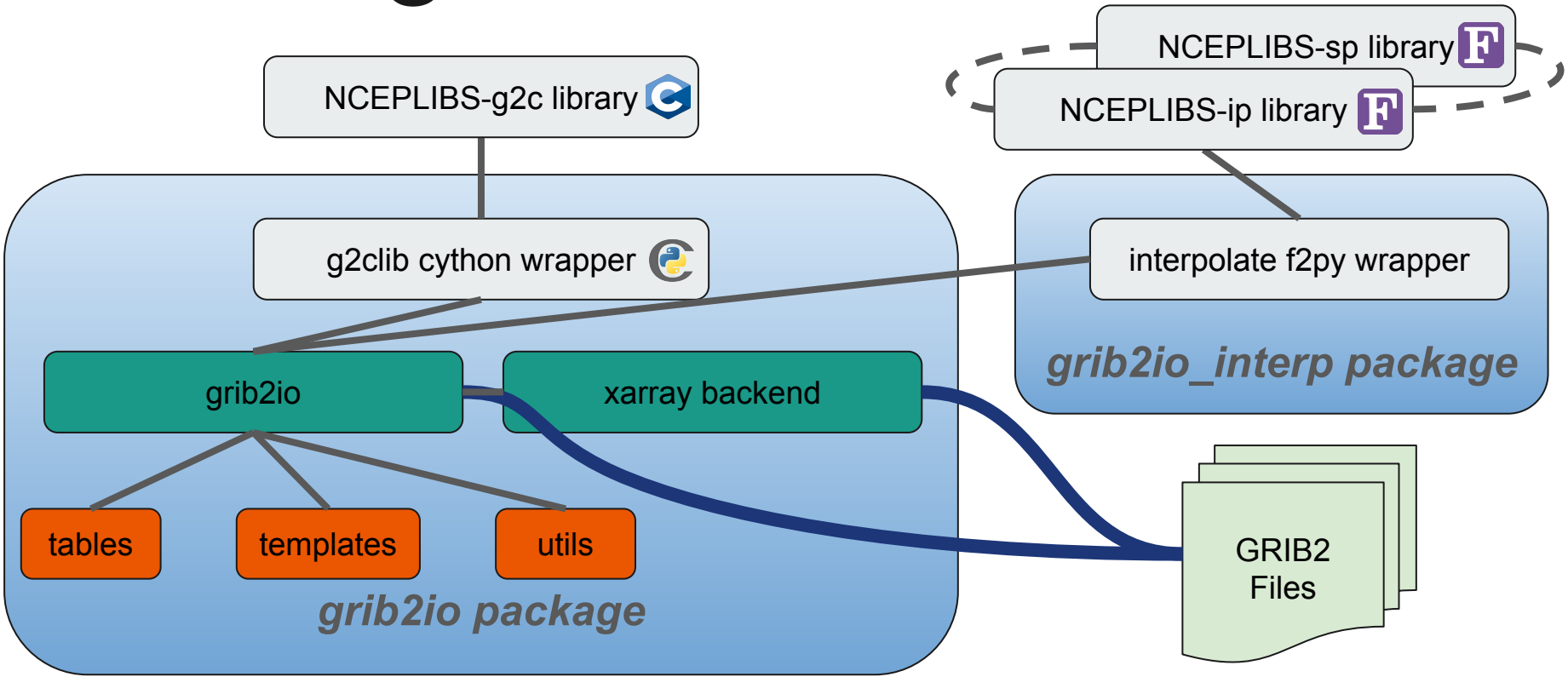
Collaboration

- The current state of **`grib2io`** would not be possible without the collaboration between MDL/SMD and NCEP/EMC NCEPLIBS Team

- MDL has assisted the NCEPLIBS team with features and bugs for g2c, sp, and ip libraries:
  - shared object support
  - interpolation to station points
  - cmake configuration and testing

- All collaboration via GitHub and pull requests.

Package Structure



Features - Reading GRIB2

- `f = grib2io.open(<file>)`
  - Iterable... `for msg in f:`

- GRIB2 files are indexed for fast access to messages and their metadata
  - Each GRIB2 message is decoded (metadata only) and stored in a **`Grib2Message`** object

- Index dictionary stores **`Grib2Message`** object and other information for decoding data when requested

```
In [1]: import grib2io

In [2]: f = grib2io.open('gfs.t00z.pgrb2.0p25.f024')

In [3]: print(f)
mode = rb
name = /Users/ericengle/Downloads/gfs.t00z.pgrb2.0p25.f024
messages = 743
current_message = 0
size = 537045998
closed = False
variables = ('4LFTX', 'ABSV', 'ACPCP', 'ALBDO', 'APCP', 'APTMP', 'CAPE', 'CFRZR', 'CICEP', 'CIN', 'CLMR', 'CNWAT', 'CPOFP', 'CPRAT', 'CRAIN', 'CSNOW', 'CWAT', 'CWORK', 'DLWRF', 'DPT', 'DSWRF', 'DZDT', 'FLDCP', 'FRICV', 'GFLUX', 'GRLE', 'GUST', 'HCDC', 'HGT', 'HINDEX', 'HLCY', 'HPBL', 'ICAHT', 'ICEC', 'ICEG', 'ICETK', 'ICETMP', 'ICMR', 'LAND', 'LCDC', 'LFT X', 'LHTFL', 'MCDC', 'MSLET', 'O3MR', 'PEVPR', 'PLPL', 'POT', 'PRATE', 'PRES', 'PRMSL', 'PWAT', 'REFC', 'REFD', 'RH', 'RWMR', 'SFCR', 'SHTFL', 'SNMR', 'SNOD', 'SOILL', 'SOILW', 'SOTYP', 'SPFH', 'SUNSD', 'TCDC', 'TMAX', 'TMIN', 'TMP', 'TOZNE', 'TSOIL', 'U-GWD', 'UFLX', 'UGRD', 'ULWRF', 'USTM', 'USWRF', 'V-GWD', 'VEG', 'VFLX', 'VGRD', 'VIS', 'VRATE', 'VSTM', 'VVEL', 'VWSH', 'WATR', 'WEASD', 'WILT')
levels = ('0-0.1 m underground', '0.01 mb', '0.02 mb', '0.04 mb', '0.07 mb', '0.1 mb', '0.1-0.4 m underground', '0.2 mb', '0.33-1 sigma layer', '0.4 mb', '0.4-1 m underground', '0.44-0.72 sigma layer', '0.44-1 sigma layer', '0.7 mb', '0.72-0.94 sigma layer', '0.995 sigma level', '0C isotherm', '1 hybrid level', '1 mb', '1-2 m underground', '10 m above ground', '10 m above mean sea level', '10 mb', '100 m above ground', '100 mb', '1000 m above ground', '1000 mb', '15 mb', '150 mb', '180-0 mb above ground', '1829 m above mean sea level', '2 hybrid level', '2 m above ground', '2 mb', '20 m above ground', '20 mb', '200 mb', '250 mb', '255-0 mb above ground', '2743 m above mean sea level', '3 mb', '30 m above ground', '30 mb', '30-0 mb above ground', '300 mb', '3000-0 m above ground', '350 mb', '3658 m above mean sea level', '40 m above ground', '40 mb', '400 mb', '4000 m above ground', '450 mb', '5 mb', '50 m above ground', '50 mb', '500 mb', '550 mb', '600 mb', '6000-0 m above ground', '650 mb', '7 mb', '70 mb', '700 mb', '750 mb', '80 m above ground', '800 mb', '850 mb', '90-0 mb above ground', '900 mb', '925 mb', '950 mb', '975 mb', 'PV=-2e-06 (Km^2/kg/s) surface', 'PV=2e-06 (Km^2/kg/s) surface', 'boundary layer cloud layer', 'cloud ceiling', 'convective cloud bottom level', 'convective cloud layer', 'convective cloud top level', 'entire atmosphere', 'entire atmosphere (considered as a single layer)', 'high cloud bottom level', 'high cloud layer', 'high cloud top level', 'highest tropospheric freezing level', 'low cloud bottom level', 'low cloud layer', 'low cloud top level', 'max wind', 'mean sea level', 'middle cloud bottom level', 'middle cloud layer', 'middle cloud top level', 'planetary boundary layer', 'surface', 'top of atmosphere', 'tropopause')
```

Features - Accessing GRIB2

- **`read([<N>])`** - read 1 or N GRIB2 messages from “current position”
  - `f.read()   # Read next message`
  - `f.read(20) # Read 20 messages`

- use index, key, or slice notation
  - `f[0]      # Returns 0th message`
  - `f[:100]   # Returns a list of first 100 messages`
  - `f['HGT'] # Returns all HGT messages`

- **`select(**kwargs)`** - select messages from file object using ***any GRIB2 metadata***
  - `f.select(shortName='HGT', level='500 mb')`

```
In [4]: import datetime

In [5]: lead = datetime.timedelta(hours=24)

In [6]: duration = datetime.timedelta(hours=6)

In [7]: apcp = f.select(shortName='APCP',level='surface',leadTime=lead,duration=duration)

In [8]: print(apcp[0])
595:d=2023-08-21 00:00:00:APCP:Total Precipitation (kg m-2):surface:1 day, 0:00:00

In [9]: print(apcp[0].duration)
6:00:00
```

Features - Grib2Message Object

- Grib2Message object is dynamically created from the GRIB2 Section 3, 4, and 5 template classes (unique metadata)
- Only stores GRIB2 metadata section data (1D arrays of integer values)
- GRIB2 metadata decoded in real-time via Python descriptor classes and their **`__get__`** and **`__set__`** methods
- Changing metadata attribute automatically updates the section data
- Values unpacked when **`data`** attribute referenced (i.e. “lazy reading”)

```
In [12]: hgt500 = f.select(shortName='HGT',level='500 mb')[0]

In [13]: hgt500.section0
Out[13]: array([1196575042,          0,          0,          2,     699593])

In [14]: hgt500.section1
Out[14]:
array([   7,    0,    2,    1,    1, 2023,    8,   21,    0,    0,    0,
          0,    1])

In [15]: hgt500.section2
Out[15]: b''

In [16]: hgt500.section3
Out[16]:
array([        0,   1038240,         0,         0,         0,         6,
               0,         0,         0,         0,         0,         0,
            1440,       721,         0,        -1,  90000000,         0,
              48, -90000000, 359750000,    250000,    250000,         0])

In [17]: hgt500.section4
Out[17]:
array([    0,     0,     3,     5,     2,     0,    96,     0,     0,
           1,    24,   100,     0, 50000,   255,     0,     0])

In [18]: hgt500.section5
Out[18]:
array([   1038240,          3, 1222951004,          2,          2,
               12,          0,          1,          0, 1649987994,
               -1,      28653,          0,          4,          1,
                1,         41,          7,          2,          2])
```

Features - Grib2Message Attributes

- Attributes that map to an NCEP GRIB2 table are a **`Grib2Metadata`** object with attributes of:
  - **`definition`** - “meaning” from table
  - **`table`** - table number
  - **`value`** - integer metadata code value

- Other attributes are floats (e.g. lats, lons); **`datetime.datetime`** objects (e.g. refDate); **`datetime.timedelta`** objects (e.g. leadTime, duration)

- NCEP GRIB2 tables are a part of grib2io formatted as dictionaries
  - several utility functions for accessing the tables

```
In [36]: hgt500.section0
Out[36]: array([1196575042,          0,          0,          2,     699593])

In [37]: hgt500.discipline
Out[37]: Grib2Metadata(0, table = '0.0')

In [38]: hgt500.discipline.definition, hgt500.discipline.table, hgt500.discipline.value
Out[38]: ('Meteorological Products', '0.0', 0)
```

Let's change the value of the **`discipline`** attribute...

```
In [39]: hgt500.discipline = 10

In [40]: hgt500.section0
Out[40]: array([1196575042,          0,         10,          2,     699593])

In [41]: hgt500.discipline
Out[41]: Grib2Metadata(10, table = '0.0')

In [42]: hgt500.discipline.definition, hgt500.discipline.table, hgt500.discipline.value
Out[42]: ('Oceanographic Products', '0.0', 10)
```

Features - Interpolation

- Interpolation to a new grid or station points via component package **`grib2io_interp`** which interfaces with NCEPLIBS-ip.

- **`interpolate()`** method for Grib2Message object
  - Grid to grid interpolation
  - Returns new Grib2Message object

- module-level **`interpolate()`** function
  - Can be used if data not in GRIB2 format
  - Interpolation to grid or station points
  - Returns data array

```
In [75]: hgt500.data.shape
Out[75]: (721, 1440)

In [76]: nbmco_gdtn = 30

In [77]: nbmco_gdt = [1, 0, 6371200, 255, 255, 255, 255, 2345, 1597, 19229000, 233723400, 48, 25000000, 265000000, 2539703, 2539703, 0, 6
    ...: 4, 25000000, 25000000, -90000000, 0]

In [78]: nbmco_griddef = grib2io.Grib2GridDef(nbmco_gdtn,nbmco_gdt)

In [79]: new_hgt500 = hgt500.interpolate('bilinear',nbmco_griddef)

In [80]: new_hgt500.data.shape
Out[80]: (1597, 2345)

In [81]: new_hgt500.attrs_by_section(3,values=True)
Out[81]:
{'interpretationOfListOfNumbers': Grib2Metadata(255, table = '3.11'),
 'gridDefinitionTemplateNumber': Grib2Metadata(30, table = '3.1'),
 'shapeOfEarth': Grib2Metadata(1, table = '3.2'),
 'earthRadius': 6371200.0,
 'earthMajorAxis': None,
 'earthMinorAxis': None,
 'resolutionAndComponentFlags': [0, 0, 1, 1, 0, 0, 0, 0],
 'ny': 1597,
 'nx': 2345,
 'scanModeFlags': [0, 1, 0, 0, 0, 0, 0, 0],
 'latitudeFirstGridpoint': 19.229,
 'longitudeFirstGridpoint': 233.7234,
 'latitudeTrueScale': 25.0,
 'gridOrientation': 265.0,
 'gridlengthXDirection': 2539.703,
 'gridlengthYDirection': 2539.703,
 'projectionCenterFlag': 0,
 'standardLatitude1': 25.0,
 'standardLatitude2': 25.0,
 'latitudeSouthernPole': -90.0,
 'longitudeSouthernPole': 0.0,
 'projParameters': {'a': 6371200.0,
  'b': 6371200.0,
  'proj': 'lcc',
  'lat_1': 25.0,
  'lat_2': 25.0,
  'lat_0': 25.0,
  'lon_0': 265.0}}
```

Where to get grib2io?

- GitHub: https://github.com/NOAA-MDL/grib2io
- Documentation: https://noaa-mdl.github.io/grib2io/
- PyPI: https://pypi.org/project/grib2io/
- Anaconda: https://anaconda.org/conda-forge/grib2io
    - MDL/SMD are maintaining NCEPLIBS g2c, sp, and ip on conda-forge

Thank You!

...now for a live demo...

Extra Slides

Features - Grib2Message Object

- Grib2Message object methods...
- **`latlons()`** - Returns latitudes and longitudes for the message
- **`map_keys()`** - Returns grid where numeric values have been unenumerated into strings (e.g. DPTYPE)

GRIB2 - CODE TABLE 4.201

PRECIPITATION TYPE

Revised 05/29/2019
Red text depicts changes made since 04/02/2015

Code Figure	Meaning
0	Reserved
1	Rain
2	Thunderstorm
3	Freezing Rain
4	Mixed/Ice
5	Snow
6	Wet Snow
7	Mixture of Rain and Snow
8	Ice Pellets
9	Graupel
10	Hail
11	Drizzle
12	Freezing Drizzle
13-191	Reserved
192-254	Reserved for Local Use
255	Missing