FlexGrid

Melissa Sulprizio
Jacob Group Meeting
29 May 2019
FlexGrid Overview

• Use HEMCO’s I/O interface and regridding/cropping capability to:
  • Explicitly control meteorological variables at run time to perform sensitivity runs
  • Set up nested-grid simulations over custom domains without changing source code

• **Stage 1:** Read meteorology fields through HEMCO
  • Developer: Jiawei Zhuang (Harvard)
  • Implemented in GEOS-Chem 12.1.0

• **Stage 2:** Define custom grids at run time
  • Developer: Melissa Sulprizio (Harvard)
  • To be added in GEOS-Chem 12.4.0
FlexGrid Stage 1: Read met fields in HEMCO

• Provides consistency with other GEOS-Chem input.
  • Emissions, non-emissions fields, restart fields, and now meteorology fields are all read in via HEMCO.

• Simplifies using new meteorology fields in GEOS-Chem.

• Allows users to do sensitivity tests and modify meteorology fields. Some examples include:
  • Holding meteorology fields constant
  • Applying scale factors
  • Applying masks
FlexGrid Stage 1: Implementation

- Consolidate meteorology code.
  - Remove `geosfp_read_mod.F90, merra2_read_mod.F90` and use `flexgrid_read_mod.F90` instead

- Add entries for met fields to `HEMCO_Config.rc`.
  - Use tokens `$MET, $met, $CNYR, $NC`

- Get fields from HEMCO in routine `Get_Met_Fields` in `hcoi_gc_main_mod.F90`.

- Still have meteorology-specific code in HEMCO extensions (e.g. DustDead, LightNOx), but soon we will have grid-independent emission fields (expected in 12.4.0).
FlexGrid Stage 1: HEMCO_Config.rc

# BEGIN SECTION SETTINGS

ROOT: /n/holylfs/EXTERNAL_REPOS/GEOS-CHEM/gcgrid/data/ExtData/HEMCO
METDIR: /n/holylfs/EXTERNAL_REPOS/GEOS-CHEM/gcgrid/data/ExtData/GEOS_4x5/GEOS_FP

# NON-EMISSIONS DATA (subsection of BASE EMISSIONS SECTION)

Non-emissions data. The following fields are read through HEMCO but do not contain emissions data. The extension number is set to wildcard character denoting that these fields will not be considered for emission calculation. A given entry is only read if the assigned species name is an HEMCO species.

# Meteorology fields for FlexGrid

# CN fields

* FRLAKE  $METDIR/$CNYR/01/$MET.$CNYR0101.CN.$RES.$NC  FRLAKE  */1/1/0  C xy 1 * - 1 1
* FRLAND  $METDIR/$CNYR/01/$MET.$CNYR0101.CN.$RES.$NC  FRLAND  */1/1/0  C xy 1 * - 1 1
* FRLANDIC $METDIR/$CNYR/01/$MET.$CNYR0101.CN.$RES.$NC  FRLANDIC  */1/1/0  C xy 1 * - 1 1
* FROCEAN $METDIR/$CNYR/01/$MET.$CNYR0101.CN.$RES.$NC  FROCEAN  */1/1/0  C xy 1 * - 1 1
* PHIS    $METDIR/$CNYR/01/$MET.$CNYR0101.CN.$RES.$NC  PHIS  */1/1/0  C xy 1 * - 1 1

# A1 fields

* ALBEDO  $METDIR/$YYYY/$MM/$MET.$YYYY$MM$DD.A1.$RES.$NC  ALBEDO  1980-2018/1-12/1-31/0-23/+30minute CY xy 1 * -
* CLDTOT  $METDIR/$YYYY/$MM/$MET.$YYYY$MM$DD.A1.$RES.$NC  CLDTOT  1980-2018/1-12/1-31/0-23/+30minute CY xy 1 * -
* EFLUX   $METDIR/$YYYY/$MM/$MET.$YYYY$MM$DD.A1.$RES.$NC  EFLUX  1980-2018/1-12/1-31/0-23/+30minute CY xy 1 * -
* EVAP    $METDIR/$YYYY/$MM/$MET.$YYYY$MM$DD.A1.$RES.$NC  EVAP  1980-2018/1-12/1-31/0-23/+30minute CY xy 1 * -
FlexGrid Stage 2: Custom nested grids

• Jiawei Zhuang’s original idea was to simplify how custom nested grids are defined in GEOS-Chem by taking advantage of HEMCO’s regridding and cropping capability.

• For consistency and transparency, the way all grids are defined has been updated.

• Instead of using compiler flags MET=, GRID= and NEST=, the meteorology fields, grid resolution, grid domain are defined at run time in input.geos.

• A new derived type object State_Grid stores the user-defined grid parameters and those computed by GEOS-Chem at run time
  • Replaces fields previously in CMN_SIZE_mod.F (e.g. IIPAR, JIPAR) and gc_grid_mod.F90 (e.g. XMid, YMid, Area_M2)

• CPP statements throughout code are replaced with IF statements, so changing meteorology field or grid doesn’t require recompiling.
### FlexGrid Stage 2: State_Grid object

<table>
<thead>
<tr>
<th>CHARACTER (LEN=255)</th>
<th>GridRes</th>
<th>Grid resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL (fp)</td>
<td>DX</td>
<td>Delta X [degrees longitude]</td>
</tr>
<tr>
<td>REAL (fp)</td>
<td>DY</td>
<td>Delta Y [degrees latitude]</td>
</tr>
<tr>
<td>REAL (fp)</td>
<td>XMin</td>
<td>Minimum X value [degrees longitude]</td>
</tr>
<tr>
<td>REAL (fp)</td>
<td>XMax</td>
<td>Maximum X value [degrees longitude]</td>
</tr>
<tr>
<td>REAL (fp)</td>
<td>YMin</td>
<td>Minimum Y value [degrees latitude]</td>
</tr>
<tr>
<td>REAL (fp)</td>
<td>YMax</td>
<td>Maximum Y value [degrees latitude]</td>
</tr>
<tr>
<td>INTEGER</td>
<td>NX</td>
<td># of grid boxes in X-direction</td>
</tr>
<tr>
<td>INTEGER</td>
<td>NY</td>
<td># of grid boxes in Y-direction</td>
</tr>
<tr>
<td>INTEGER</td>
<td>NZ</td>
<td># of grid boxes in Z-direction</td>
</tr>
<tr>
<td>LOGICAL</td>
<td>HalfPolar</td>
<td>Use half-sized polar boxes?</td>
</tr>
<tr>
<td>LOGICAL</td>
<td>NestedGrid</td>
<td>Is it a nested grid sim?</td>
</tr>
<tr>
<td>INTEGER</td>
<td>NorthBuffer</td>
<td># buffer grid boxes on North edge</td>
</tr>
<tr>
<td>INTEGER</td>
<td>SouthBuffer</td>
<td># buffer grid boxes on South edge</td>
</tr>
<tr>
<td>INTEGER</td>
<td>EastBuffer</td>
<td># buffer grid boxes on East edge</td>
</tr>
<tr>
<td>INTEGER</td>
<td>WestBuffer</td>
<td># buffer grid boxes on West edge</td>
</tr>
</tbody>
</table>
Grid fields computed in gc_grid_mod.F90

INTEGER :: GlobalNX ! NX on the global grid
INTEGER :: GlobalNY ! NY on the global grid
INTEGER :: NativeNZ ! NZ on the native-resolution grid
INTEGER :: MaxChemLvl ! Max # levels in chemistry grid
INTEGER :: MaxStratLvl ! Max # levels below strat
INTEGER :: MaxTropLvl ! Max # levels below trop
INTEGER :: XMinOffset ! X offset from global grid
INTEGER :: YMaxOffset ! X offset from global grid
INTEGER :: XMinOffset ! Y offset from global grid
INTEGER :: YMaxOffset ! Y offset from global grid

! Arrays
REAL(fp), POINTER :: GlobalXMid(:, :) ! Lon centers on global grid [deg]
REAL(fp), POINTER :: GlobalYMid(:, :) ! Lat centers on global grid [deg]
REAL(fp), POINTER :: XMid (:, :) ! Lon centers [degrees]
REAL(fp), POINTER :: XEdge (:, :) ! Lon edges [degrees]
REAL(fp), POINTER :: YMid (:, :) ! Lat centers [degrees]
REAL(fp), POINTER :: YEdge (:, :) ! Lat edges [degrees]
REAL(fp), POINTER :: YMid_R (:, :) ! Lat centers [radians]
REAL(fp), POINTER :: YEdge_R (:, :) ! Lat edges [radians]
REAL(fp), POINTER :: YSIN (:, :) ! SIN( lat edges )
REAL(fp), POINTER :: Area_M2 (:, :) ! Grid box area [m2]
FlexGrid Stage 2: Defining met and grid

- A Grid Menu has been added to `input.geos`:

```plaintext
%% SIMULATION MENU %%
Start YYYYMMDD, hhmmss : 20160701 000000
End   YYYYMMDD, hhmmss : 20160703 000000
Run directory       : ./
Root data directory  : /n/holylfs/EXTERNAL_REPOS/GEOS-CHEM/gcgrid/data/ExtData
Met field            : GEOS-FP

%% GRID MENU %%
Grid resolution       : 4.0x5.0
Longitude min/max    : -180.0 180.0
Latitude  min/max     : -90.0  90.0
Half-sized polar boxes?: T
Number of levels      : 47
Nested grid simulation?: F
Buffer zone (N S E W ) : 3 3 3 3
```

- Theoretically, any meteorology field, grid resolution, and grid domain may be defined but the first step is to validate the existing grids supported in GEOS-Chem.
Example: 4° x 5° global grid

%% GRID MENU %%
Grid resolution : 4.0x5.0
Longitude min/max : -180.0 180.0
Latitude min/max : -90.0 90.0
Half-sized polar boxes?: T
Number of levels : 47
Nested grid simulation?: F
Buffer zone (N S E W ) : 3 3 3 3
Example: $2^\circ \times 2.5^\circ$ global grid

%% GRID MENU %%

Grid resolution : 2.0x2.5
Longitude min/max : -180.0 180.0
Latitude min/max : -90.0 90.0
Half-sized polar boxes?: T
Number of levels : 47
Nested grid simulation?: F
Buffer zone (N S E W ) : 3 3 3 3
Example: $0.5^\circ \times 0.625^\circ$ nested Asia
Example: 0.5° x 0.625° nested Europe

```plaintext
%% GRID MENU %%
Grid resolution : 0.5x0.625
Longitude min/max : -30.0 50.0
Latitude min/max : 30.0 70.0
Half-sized polar boxes?: F
Number of levels : 47
Nested grid simulation?: T
Buffer zone (N S E W ) : 3 3 3 3
```
Example: $0.5^\circ \times 0.625^\circ$ nested N. America
Example: $0.25^\circ \times 0.3125^\circ$ nested China

```plaintext
%% GRID MENU %%
Grid resolution : 0.25x0.3125
Longitude min/max : 70.0 140.0
Latitude min/max : 15.0 55.0
Half-sized polar boxes? : F
Number of levels : 47
Nested grid simulation? : T
Buffer zone (N S E W) : 3 3 3 3
```
Example: 0.25° x 0.3125° nested Europe
Example: 0.25° x 0.3125° nested N. America

%% GRID MENU %%
Grid resolution : 0.25x0.3125
Longitude min/max : -130.0 -60.0
Latitude min/max : 9.75 60.0
Half-sized polar boxes?: F
Number of levels : 47
Nested grid simulation? : T
Buffer zone (N S E W ) : 3 3 3 3
FlexGrid Stage 2: Nested grids

- Boundary conditions are now saved out to netCDF files using the History component. In `HISTORY.rc`:

```plaintext
COLLECTIONS: 'Restart',
             'BoundaryConditions',
             ...

# ==THE BoundaryConditions COLLECTION==
#
# GEOS-Chem boundary conditions for use in nested grid simulations
#
# Available for all simulations

BoundaryConditions.template: '$y%m2%d2_%h%n2z.nc4',
BoundaryConditions.format:   'CFIO',
BoundaryConditions.frequency: 00000000 030000
BoundaryConditions.duration:  00000001 000000
BoundaryConditions.mode:      'instantaneous'
BoundaryConditions.fields:    'SpeciesBC_?ALL? ', 'GIGCchem',
```

**NOTE:** Currently only global BC files can be saved out, but the capability to save out regional BC files is being added to History now.
FlexGrid Stage 2: Nested grids

- Boundary conditions are read in via HEMCO. In `HEMCO_Config.rc`:

```
# ExtNr ExtName     on/off  Species
0     Base          on       *
  --> GC_RESTART    :         true
  --> GC_BCs       :         true
...
```

- Get BCs from HEMCO in routine `Get_Boundary_Conditions` in `hcoi_gc_main_mod.F90`.
- BCs are applied to the buffer zone defined in the Grid Menu of `input.geos`. 
FlexGrid Status

• FlexGrid Stage 2 was validated to ensure:
  • The structural code updates don’t impact model output.
  • All existing meteorology fields (GEOS-FP, MERRA-2), grid resolutions (4°x5°, 2°x2.5°, 0.5°x0.625°, 0.25°x0.3125°), and nested domains (China/Asia, Europe, North America) still work as expected.

• The GCST will need support from the user community to experiment with other meteorology fields and grid definitions for further refinement to FlexGrid.

• FlexGrid will be included in GEOS-Chem 12.4.0 (benchmarking in progress).

• Nested Model community has requested a python script to convert BC files from BPCH to netCDF.
FlexGrid application

- Defined custom 0.25 x 0.3125 nested grid over Vietnam to model air quality conditions under present-day and future coal-fired power plant emissions scenarios.