New features in GEOS-Chem 13.0.0
(+ GCHP 13.0.0 + HEMCO 3.0.0 + GCPY 1.0.0)

Melissa Sulprizio
Jacob Group Meeting
4 March 2020
The GCST plans to simultaneously release:

- GEOS-Chem 13.0.0
- GCHP 13.0.0
- HEMCO 3.0.0
- GCPy 1.0.0
The GCST plans to simultaneously release:

- GEOS-Chem 13.0.0
- GCHP 13.0.0
- HEMCO 3.0.0
- GCPy 1.0.0

All will be major version releases involving significant structural changes and breaking backwards compatibility.
Today’s Plan

Background motivation and overview of:

1. GCHP 13.0.0
2. HEMCO 3.0.0
3. GEOS-Chem 13.0.0
4. GCPy 1.0.0
Background motivation and overview of:

1. GCHP 13.0.0
2. HEMCO 3.0.0
3. GEOS-Chem 13.0.0
4. GCPy 1.0.0
GCHP 13.0.0

Credit: Lizzie Lundgren
GCHPctm repository on Github

Wrapper for GEOS-Chem chemical-transport model to enable the high performance option (GCHP).
Today’s Plan

Background motivation and overview of:

1. GCHP 13.0.0
2. HEMCO 3.0.0
3. GEOS-Chem 13.0.0
4. GCPy 1.0.0
HEMCO was developed by Christoph Keller to streamline emissions in GEOS-Chem.

Recent GEOS-Chem versions are taking advantage of HEMCO’s I/O, regridding, scaling, and masking capabilities to process all geospatial data.

- HEMCO now also reads meteorology, chemistry input data, restart files, and boundary conditions.
- This has the advantage of consistently reading, regridding, cropping, and scaling input data.

Several external models are interested in leveraging HEMCO’s capabilities to handle input data.
HEMCO repository on Github

- HEMCO is currently distributed within the GEOS-Chem source code.

- A new Github repository has been set up to for HEMCO development:
  - HEMCO will be a Git submodule in GEOS-Chem 13.0.0
  - This will also allow for HEMCO to be easily dropped into other models.

- The repository includes a run directory for the HEMCO standalone.

- Source code can now be compiled with CMake.

- Up next: Restructuring and cleanup of HEMCO source code to facilitate interfacing with external models.
HEMCO repository on Github
HEMCO Structure

HEMCO:
AUTHORS.txt  CMakeLists.txt  CMakeScripts/  LICENSE.txt  README.md  run/  src/
HEMCO Structure

• Source code is currently split into Core, Extensions, and Interfaces
HEMCO Structure

• Source code is currently split into Core, Extensions, and Interfaces

  HEMCO:
  AUTHORS.txt  CMakeLists.txt  CMakeScripts/  LICENSE.txt  README.md  run/  src/

  HEMCO/src:
  CMakeLists.txt  Core/  Extensions/  Interfaces/  Makefile  Shared/

• run/ contains template files and a setup script to generate HEMCO standalone run directories

  HEMCO/run:
  createRunDir.sh*
  HEMCO_Diagn.rc
  HEMCO_sa_Config.template  HEMCO_sa_Grid.025x03125.rc  HEMCO_sa_Grid.4x5.rc  HEMCO_sa_Grid.05x0625.rc  HEMCO_sa_Spec.rc  HEMCO_sa_Time.rc  OutputDir/
  runHEMCO.sh*
HEMCO source code

- **Core/** is currently a file dump with code for:
  - Defining HEMCO derived type objects
  - Reading settings from configuration file
  - Reading data from files
  - Regridding horizontally (and vertically) to model grid
  - Calculating emissions (applying masks, scale factors, hierarchy)

- **Interfaces/** contains code for running HEMCO in standalone mode or interfaced with other models

- **Extensions/** contains code for computing emissions that require knowledge about environmental fields

- **Shared/** is newly added and contains copies of GEOS-Chem routines also used by HEMCO
Background motivation and overview of:

1. GCHP 13.0.0
2. HEMCO 3.0.0
3. GEOS-Chem 13.0.0
4. GCPy 1.0.0
Structural changes for GCHP and HEMCO have necessitated changes within GEOS-Chem “Classic”

1. Change build process to CMake and retire GNU Make
   • GCHPctm and HEMCO standalone now solely use CMake
   • CMake was added as an option to GNU Make in GEOS-Chem 12.6.0
Structural changes for GCHP and HEMCO have necessitated changes within GEOS-Chem “Classic”

2. Move run directory files to `run/` within the GEOS-Chem repository
   • The GEOS-Chem repository already includes run directory files and setup scripts for GCHPctm and GEOS-GC
   • Storing run directory files separately in the GEOS-Chem Unit Test repository has been a source of error and confusion among users
   • Run directory updates will now be pegged to a GEOS-Chem version
   • A setup script will walk the users through run time options
Structural changes for GCHP and HEMCO have necessitated changes within GEOS-Chem “Classic”

3. Make HEMCO a Git submodule within GEOS-Chem
   • This will allow for HEMCO to be easily interfaced into other models
   • A separate HEMCO repository will hopefully foster feedback and developments from users outside of the GEOS-Chem community
Other anticipated structural changes include:

4. Dynamic allocation of memory for netCDF diagnostics

5. Capability for REAL*8 output from HISTORY and input for HEMCO to reduce differences between single and multi-segmented runs

6. Conversion of species database from Fortran to YAML file

7. GEOS-Chem classic wrapper to encompass GEOS-Chem, HEMCO, and shared utilities

8. Additional updates to improve model performance? (more later)
Background motivation and overview of:

1. GCHP 13.0.0
2. HEMCO 3.0.0
3. GEOS-Chem 13.0.0
4. GCPy 1.0.0
GCPy 1.0.0

- GCPy is nearing maturity in terms of benchmark plotting capabilities
- GCPy is nearing maturity in terms of benchmark plotting capabilities
GCPy 1.0.0

- GCPy is nearing maturity in terms of benchmark plotting capabilities
GCPy 1.0.0

• GCPy is nearing maturity in terms of benchmark plotting capabilities
GCPy 1.0.0

- GCPy is nearing maturity in terms of benchmark plotting capabilities
GCPy 1.0.0

- Recent and near-term developments include:
  - Parallelization of benchmark plotting routines
  - Unit testing capability
  - Creation of single plots

- The official GCPy release (1.0.0) will be announced with the other releases
The GEOS-Chem community has expressed concern in the slowdown of the model. Luckily, we can leverage the official GEOS-Chem 1-month benchmarks along with internal benchmarks to identify potential causes.
The GEOS-Chem community has expressed concern in the slowdown of the model. Luckily, we can leverage the official GEOS-Chem 1-month benchmarks along with internal benchmarks to identify potential causes.

<table>
<thead>
<tr>
<th>Timer name</th>
<th>DD-hh:mm:ss.SSS</th>
<th>Total Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEOS-Chem</td>
<td>00-08:39:54.476</td>
<td>31194.477</td>
</tr>
<tr>
<td>Initialization</td>
<td>00-00:01:20.156</td>
<td>80.156</td>
</tr>
<tr>
<td>Timesteps</td>
<td>00-08:37:56.539</td>
<td>31076.539</td>
</tr>
<tr>
<td>HEMCO</td>
<td>00-01:36:09.328</td>
<td>5769.328</td>
</tr>
<tr>
<td>All chemistry</td>
<td>00-03:57:44.851</td>
<td>14264.852</td>
</tr>
<tr>
<td>=&gt; Gas-phase chem</td>
<td>00-03:18:36.046</td>
<td>11916.047</td>
</tr>
<tr>
<td>=&gt; FAST-JX photolysis</td>
<td>00-00:11:37.242</td>
<td>697.242</td>
</tr>
<tr>
<td>=&gt; All aerosol chem</td>
<td>00-00:21:44.007</td>
<td>1304.008</td>
</tr>
<tr>
<td>=&gt; Strat chem</td>
<td>00-00:01:09.328</td>
<td>69.328</td>
</tr>
<tr>
<td>=&gt; Unit conversions</td>
<td>00-00:18:01.062</td>
<td>1081.062</td>
</tr>
<tr>
<td>Transport</td>
<td>00-00:33:17.242</td>
<td>1997.242</td>
</tr>
<tr>
<td>Convection</td>
<td>00-00:30:52.710</td>
<td>1852.711</td>
</tr>
<tr>
<td>Boundary layer mixing</td>
<td>00-00:40:54.734</td>
<td>2454.734</td>
</tr>
<tr>
<td>Dry deposition</td>
<td>00-00:01:27.101</td>
<td>87.102</td>
</tr>
</tbody>
</table>
All simulations use the 4°x5° GEOS-FP benchmark simulation and ran on 24 CPUs
GEOS-Chem Performance

FlexChem bug introduced
GEOS-Chem Performance

FlexChem bug fixed
GEOS-Chem Performance

Halogen chemistry (Br, Cl, I)
GEOS-Chem Performance

FlexGrid Stage 1 + Restart file updates
GEOS-Chem Performance

Related to diagnostics?
GEOS-Chem Performance

Offline emissions
GEOS-Chem Performance

HEMCO updates and fixes
GEOS-Chem Performance

Unit conversion fix?
GEOS-Chem Performance

Inefficient I/O caused by SfcFix
Next steps: Profile with TAU; Run time tests on AWS