Software Engineering, Code Development, User Support Issues

Bob Yantosca
Philippe Le Sager
Claire Carouge
Atmospheric Chemistry Modeling Group
School of Engineering & Applied Sciences
Harvard University

geos-chem-support@as.harvard.edu

GEOS–Chem Meeting
Thursday, April 9th, 2009
Outline

• User Resources:
  – Where to find information?

• Code development:
  – Who does what?
  – And how to do it?

• Debugging
  – Tips and tricks

• Next?
  – Moving towards netCDF?
GEOS–Chem User Resources

Subscribe to the GEOS–Chem wiki

• The wiki facilitates communication among GEOS–Chem users and developers. *It is the first place you should look for info!*

• The wiki contains information about
  - GEOS–Chem versions
  - Benchmark simulations
  - Bugs, fixes, and machine issues
  - Emissions options
  - Chemistry mechanism
  - Offline simulations
  - Met fields
  - Adjoint simulations
  - GAMAP

• Create a login and password, then you can edit pages
"Recent changes" wiki page has RSS feed option!

RSS Feed Readers
- Google Reader
- Mozilla Thunderbird
- Internet Explorer
- Akregator
- Safari
- Mail
- etc...

SUBSCRIBE for automatic notification
Subscribe to GEOS–Chem email lists

- **GEOS–Chem general**
  - geos-chem@seas.harvard.edu
  - Intended for ALL geos chem users!
  - Version updates & other info will be sent to this list

- **GEOS–Chem aerosols**
  - geos-chem-aerosols@seas.harvard.edu
  - Intended for developers of aerosol simulations

- **GEOS–Chem adjoint**
  - geos-chem-adjoint@seas.harvard.edu
  - Intended for users of the adjoint model

For info about how to subscribe, please see this wiki page:

GEOS-Chem_welcome_letter_for_new_users#Subscribing_to_the_GEOS-Chem_email_lists
GEOS–Chem User Resources

GEOS–Chem web page and user manuals

- GEOS–Chem web page, contains
  - Recent publications and presentations
  - Information about GEOS–Chem user groups
  - Information about parallelization and coding style
  - Links to programming resources (e.g. F90 tutorials, HDF info, etc.)
  - Links to various types of GEOS–Chem output
  - And more...

- User manuals
  - GEOS–Chem Users' Guide
  - GAMAP Users' Guide
  - GEOS–Chem Style Guide
  - TESTRUN
Who supports what?

- Bob, Philippe, and Claire support GEOS–Chem's “full-chemistry” simulation (NOx – Ox – HC's – Aerosols)
  - 1 month & 1 year benchmark simulations
  - User manual on the GEOS–Chem website … etc

- The GEOS–Chem user community is responsible for the upkeep and validation of other simulations, including:
  - Offline tagged Ox
  - Offline tagged CO
  - Offline tagged aerosols
  - Offline CH4
  - Mercury
  - Nested grid simulations, … etc.

- Users should provide B/P/C with updates for offline simulations, for inclusion into the GEOS–Chem std code
GEOS–Chem Code Development

GEOS–Chem user responsibilities

- GEOS–Chem users are responsible for:
  - Setting up their systems
  - Learning F90, IDL, shell scripts, etc...
  - Periodically updating to the latest version of G–C
    - i.e. at the start of a new project
  - Debugging code
- Promptly report any bug fixes you have made to B/P/C
  - Other users may be struggling with the same problem and could benefit from the fix that you have made
- Code submission guidelines
  - When any code you have written is mature, submit it to B/P/C for inclusion into the GEOS–Chem std code.
  - Also provide the appropriate documentation
Documentation: Do's

- Please provide copious comments in your code
  - List the name and purpose of each module, subroutine, and function
  - Label each input/output argument and its units
  - Provide citations to journal references where appropriate
  - Use descriptive names for variables, subroutines, functions
  - Clearly identify your modifications in the code using your initials

- For data that is meant as input to GEOS–Chem, provide:
  - A README file describing:
    - Individual data files and their contents (and units!!)
    - Source of the data and any journal references
    - The FTP/web site from which the data was obtained
  - Plots, tables and/or sums of emissions data
  - IDL/F90 etc. code that was used to create individual data files

- Feel free to post documentation for code & data on the wiki!
Documentation: Don'ts

- Don't use a language other than English
- Avoid partial citations such as:
  - “see Smith & Jones, 2001”
  - “see Smith's paper”
  - “see Amanda's slide”
  - List the full citation first, then abbreviate thereafter...
- Avoid cryptic comments
  - e.g. “NOx 1.2, Ox 2.0”
- Don't assume that the person reading your documentation will have any prior knowledge about this code and/or data
  - Leave nothing unexplained!
Debugging / Testing Tips

Use GEOS-Chem utilities

- **CHECK_STT** if NaN or a negative tracer appears
- **SAFE_DIV** to avoid potential overflow (division by zero)
- **IS_SAFE_DIV** to check both overflow & underflow in division
- **CHECK_VALUE** to check for NaN or INFINITY

Look for numerical traps

- **“catastrophic cancellation”** when subtracting huge and small
- use **F90 utilities**
  - TINY to get a small positive number
- compile with **warnings** on
  - check bounds, check uninit, check pointers (ifort)
Use a Debugger

- F90 -> Totalview, dbx, GDB
- IDL, Matlab -> included

Report a bug

- Use this address: geos-chem-support@as.harvard.edu
- Please provide this information:
  - GEOS-Chem version number & Type of Simulation
  - Met field type & horizontal resolution (e.g. 4 x 5 GEOS–5)
  - Platform, Compiler, Number of Processors
  - Description of problem
  - Log file output from the simulation w/ the error message
  - **Important!** Have you made any changes to the standard code?
Some Useful Resources:

- **Floating-point math issues** page on the GEOS-Chem wiki
  - Brief description of how floating-point math works
  - How to avoid some common numerical pitfalls

- **Programming resources** page on GEOS–Chem website
  - Links to tutorials, compiler sites & manuals, and other information
  - [http://www.as.harvard.edu/ctm/geos/geos_resources.html](http://www.as.harvard.edu/ctm/geos/geos_resources.html)
Pros and Cons of netCDF

PROS:

● Binary format with header
● Easy access to the data from command line
● Easy to share with other teams

CONS:

● Every user has to install the netCDF library by himself
● Do not give more information than .bpch files
● Reading netCDF files w/ Fortran is relatively easy
● Writing netCDF files w/ Fortran is more involved
● A substantial amount of work to implement in G-C

It's not a priority for us. But feel free to begin the work if you want to
Questions?