Status Report:
Adding MARBL to (CESM) MOM

Mike Levy, NCAR
August 3, 2020
(With lots of input from Keith Lindsay, Matt Long, Gustavo Marques, Andrew Shao, and the GFDL developers)
Outline

• What is MARBL?
• How does MARBL run?
• How does MARBL fit into MOM? (brief)
  • Includes titular status report
• How does MARBL fit into MOM? (long answer)
• Summary
What is MARBL?

• Software library to add biogeochemistry tracers to ocean models
• Modular / flexible design independent of any GCM
  • Computations done column by column (no horizontal operators)
  • Initial (Fall 2014) goal: share POP’s BGC code with MPAS-O [DOE grant for NCAR & LANL]
  • Ideal world: other GCMs adopt it as well

We are [almost] here
MARBL Design Principles

1. Open development on github: [https://github.com/marbl-ecosys/MARBL](https://github.com/marbl-ecosys/MARBL)
   - Fortran computation, support wide range of compilers
   - Python for tools
   - Documentation [hosted on github.io](https://github.io)

2. Column-wise computation

3. Requirements on the GCM:
   a. GCM handles all I/O
   b. GCM handles all parallelization
   c. GCM responsible for time-stepping
   d. GCM holds *some* data in memory for MARBL from one time step to the next
   e. GCM provides horizontal operators (if necessary)

4. Consistent units in MARBL
   - Currently cgs, switch to mks is planned ([issue #41 on github](https://github.com))
MARBL Design
What Does MARBL Provide?

- Adds 32 tracers to the system
  - Optional carbon isotope module adds 14 more tracers when enabled
- Computes surface fluxes (air-sea CO$_2$, chlorophyll concentration for shortwave absorption, etc)
- Computes source / sink terms for tracers (requires GCM to apply them)
- “Stand-alone” MARBL provides some small test cases (separate from code shared with GCMs)
How Does MARBL Run?

Initialization
- Set parameters
- Define tracers
- Define diagnostics
- Allocate memory

Per time-step
- Compute surface fluxes
- Compute interior tendencies (source / sink terms)
- Provide GCM with diagnostics

Shut down
- Make sure log is empty
- Deallocate memory
How Do We Add MARBL to MOM?

1. Infrastructure updates
   • Create compsets that turn on BGC tracers by default
   • Interact with MARBL python scripts
     o Parameter settings
     o Determining diagnostic output
   • Introduce new `USE_MARBL_TRACERS` parameter, only add tracers if `.true.`
   • Only build if compiled with `-D_USE_MARBL_TRACERS`
   • Update MOM diagnostic package to handle MARBL output
How Do We Add MARBL to MOM?

2. Fortran changes
   1. Read MARBL parameters from disk
   2. Call MARBL’s `init()` via `call_tracer_register()`
   3. Call MARBL’s `surface_flux_compute()` and `interior_tendency_compute()` via `call_tracer_column_fns()`
   4. Add MARBL diagnostics to desired history files
   5. Call MARBL’s `shutdown()` via `tracer_flow_control_end()`
Progress Report

• CESM can build MARBL as part of MOM
• MOM can add all MARBL tracers to the model
  • MOM can passively advect all MARBL tracers
  • MOM can not apply surface flux or source / sink terms
• MARBL diagnostics can be added to output, but not in a user-friendly way
• MARBL log messages are written to CESM log files.
MARBL Tracers in MOM output

• DISCLAIMER: MOM ignores MARBL’s surface flux and tendency computations and just passively advects tracers!
More Details

• Testing in CESM framework; that’s primary focus for initial implementation

• Trying to avoid showing code snippets, but all code is available on github
  • Branch on my fork of MOM6
  • Branch on my fork of MOM_interface

• Keeping up-to-date CESM-MOM development:
  • Merged in dev/ncar_20200724 and mi_20200724

• Following slides use the following schema to discuss progress:
  • Done! (✔)
  • In progress (✔️)
  • Not yet begun (⚠️)
Infrastructure Status

• **Build MARBL via** `case.build` **and MOM_interface’s buildlib** (✔)

• **Add** `USE_MARBL_TRACERS` **to MOM parameter list** (✔)
  • Default value is false unless user specifies valid `OCN_BGC_CONFIG` value
  • Created `C1850MOMECO` compset to set `OCN_BGC_CONFIG=latest`

• **Build MOM with** `-D_USE_MARBL_TRACERS` (⚠️)
  • My CIME sandbox’s `config/cesm/machines/config_compilers.xml` includes flag whenever we build MOM
  • Need to talk to Alper and CIME developers about what we want
    • Maybe something in `env_build.xml` the triggers addition of the flag?
Infrastructure Status (cont)

- Provide initial condition and surface forcing files
  - Initial condition file generated by [a notebook in my fork of marbl-forcing](https://github.com/username/notebook)
    - Some WOA (2013), some GLODAP (v2), rest from CESM2.1 piControl for CMIP6
  - Some surface forcing will eventually come from files, currently hard-coding “reasonable” values just for testing
Infrastructure Status (cont)

• Tie in to MARBL’s `generate_settings_file` script to provide runtime parameters (확정)
  • `buildnml generates marbl_in, but`
    1. Still need to implement `user_nl_marbl` to allow non-default parameters
    2. Still need to modify Fortran code to read `marbl_in`

• Tie in to MARBL’s `generate_diagnostics_file` to provide list of diagnostic output (확정)
  • I’ve hard-coded tracers and CO\textsubscript{2} piston velocity into `diag_table.yaml`
  • Will add a few additional diagnostics by hand as development warrants, and won’t connect to MARBL infrastructure until we think Fortran driver is done

• Tie in to MOM diagnostic package (확정)
Fortran Status

• Follows “How Does MARBL Run” slide pretty nicely:

  Initialization
  • Set parameters (☐)
  • Define tracers (☑)
  • Define diagnostics (☑)
  • Allocate memory (☑)

  Per time-step
  • Compute surface fluxes (☑)
  • Compute interior tendencies (source / sink terms) (☐)
  • Provide GCM with diagnostics (☑)

  Shut down
  • Make sure log is empty (☑)
  • Deallocate memory (☑)

• In src/tracer/, MARBL_tracers.F90 is called from MOM_tracer_flow_control.F90
Fortran Status (cont)

• Finished tasks
  • MOM6 gets list of BGC tracers from MARBL in call_tracer_register()
  • Every diagnostic MARBL can return is reported to MOM in tracer_flow_control_init()

• Tasks in progress
  • Compute surface fluxes
    • All calls to MARBL are in place (✔)
    • Some forcing fields are properly filled from MOM memory (✔)
    • Remaining forcing fields need to be read from disk or pulled from coupler
    • Need to tell MOM what to do with the surface fluxes
    • MARBL can compute fluxes for multiple columns simultaneously, current implementation is column-by-column
Fortran Status (cont)

• Tasks in progress (cont)
  • Allocating memory: creating internal datatypes as needed, so more to come
  • Write diagnostics
    • All diagnostics from computing surface fluxes are available (☐)
    • Interior tendency computation not called, so those diagnostics are not accumulated
  • The MARBL log is written to cesm.log, but most content belongs in ocn.log

• Still to do
  • Read marbl_in, pass values to MARBL
  • Call interior_tendency_compute()
    • Similar calling sequence to what I’ve done for surface fluxes
    • Handling vertical coordinate will be tricky
MARBL Diags in MOM output

• DISCLAIMER: MOM ignores MARBL’s surface flux and tendency computations and just passively advects tracers!
Summary

• MARBL driver for MOM is a work in progress
• Most of the infrastructure is in place for CESM+MOM
  • MOM+MARBL without CESM will be supported as well, but isn’t yet
• Have a few more MARBL calls to make
  • Also need to tie MARBL’s `intent(out)` back to MOM