Newton-Krylov Methods for Tracer Spinup

Keith Lindsay (NCAR)
primary collaborators: François Primeau & Ann Bardin (UC-Irvine)
August 3, 2020
Statement of Problem

• Generate tracer distributions that are in balance with respect to (non-stationary) ocean model circulation (advection and mixing).

• Applications:
  – Initializing transient experiments
  – Analyze dynamics/properties of spun-up tracers
  – Compare tracers to observations
  – Optimize parameters to reduce model bias
    • Requires ability to spin-up repeatedly

• Brute force is prohibitively expensive
  – wall-clock time and computing allocation
  – \( \frac{2000 \text{ years}}{50 \text{ years/day}} = 40 \text{ days} \)
Online Target Model Configurations

- POP2 in CESM, x1 grid: 320x384x60 (~4.2x10^6 grid points)
  - nominal 1° resolution

- MOM6 in CESM, t06 grid: 540x458x34 (~4.1x10^6 grid points)
  - nominal 2/3° resolution, on WOA09 z* vertical grid

- Tracer modules
  - Abiotic radiocarbon: 2 tracers
  - Dye tracers, regional He isotopes: arbitrary # of independent tracers
  - CESM 2 ecosystem: 32 tracers
  - Active tracers: temperature & salinity
Mathematical Formulation of Problem

• Let $c(t)$ denote tracer state, i.e., tracer concentrations.
  – for 1 tracer on target grids, $c$ is vector of length $\approx 4 \times 10^6$

• Model Map: $c(t) = \Phi(c(0), t)$

• $\Phi$ incorporates advection, mixing, surface fluxes, interior source-sink terms, etc.

• Find $c^*$ such that $\Phi(c^*, T) = c^*$.
  – Tracer end-state is the same as the initial condition.
  – $T$ is period of forcing and circulation.

• Rewrite as $G(c) \equiv \Phi(c, T) - c = 0$. 

Newton-Krylov Methods for Tracer Spinup
Newton-Krylov Solvers


(not Newton vs. Krylov)
Newton’s Method

• Iterative method for solving $G(c) \equiv \Phi(c,T) - c = 0$
• Generate sequence $c_1, c_2, \ldots, c_k, \ldots$ that converge to solution of system of equations
• $0 = G(c_{k+1}) = G(c_k) + (\frac{\partial G}{\partial c})^*(c_{k+1} - c_k) + \ldots$
  \[ c_{k+1} = c_k - (\frac{\partial G}{\partial c})^{-1} \ast G(c_k) \]
• Newton increment $\delta c_k$ solves linear equation
  \[ (\frac{\partial G}{\partial c})(\delta c_k) = -G(c_k) \]
Computing the Increment in Newton’s Method

• We cannot compute, or store, \( \partial G / \partial c \), but we can evaluate matrix-vector products such as \( \partial G / \partial c ) (\delta c) \) with the finite difference approximation

\[
\frac{(\partial G / \partial c ) (\delta c)}{\delta c} \approx \frac{(G(c + \sigma \delta c) - G(c))}{\sigma}
\]

• Note this is a forward model run of length T.

• Krylov iterative methods are well suited for this scenario.
Krylov Solvers

• Use Krylov iterative method (GMRES) to solve:
  \[(\partial G/\partial c)\delta c_k = -G(c_k)\]

• Construct Krylov basis
  \[y_0, (\partial G/\partial c)y_0, (\partial G/\partial c)^2y_0, (\partial G/\partial c)^3y_0, \ldots\]

• Find linear combination of basis that minimizes
  \[|{(\partial G/\partial c) x + G(c_k)}|^2\]

• Each GMRES iteration evaluates \((\partial G/\partial c)(\delta c)\)
  – Note this is a forward model run of length T.
Preconditioner for GMRES

• We apply a preconditioner to improve the convergence of GMRES
  I.e., transform \((\partial G/\partial c)(\delta c_k) = -G(c_k)\) into
  \[(P(\partial G/\partial c))(\delta c_k) = (P)(-G(c_k))\]

• To improve convergence, \(P \approx (\partial G/\partial c)^{-1}\)

• To be practical, multiplying by \(P\) should be feasible.

• We construct \(P\) as the inverse of a sparse approximation to \(\partial G/\partial c\),
  based on time mean advection and mixing operators, extracted from
  diagnostics that have been added to the ocean model. Terms are
  related to \(\partial(\partial C_i/\partial t)/\partial C_j\).

• Multiplication by \(P\) is implemented with a parallel sparse matrix
  solver, SuperLU_DIST.
Putting it all together

• Find $c^*$ such that $G(c^*) \equiv \Phi(c^*, T) - c^* = 0$.
  - Tracer end-state is the same as the initial condition.

• Use Newton’s Method, $c_{k+1} = c_k - (\partial G/\partial c)^{-1} * G(c_k)$

• Use Krylov solver (GMRES) to solve $(\partial G/\partial c)(\delta c_k) = -G(c_k)$
  - Matrix-vector multiplies in GMRES are approximated with a model run
  - Use a preconditioner based on a sparse approximation to $\delta G/\delta c$

• Numerous technical details omitted
Convergence

Ideal Age tracer (IAGE)
4 Krylov iterations per Newton iteration

a) $\|G\|_2 = \|\Delta IAGE\|_2$

b) Volume where $\Delta IAGE$ exceeds threshold

c) Histogram of $\log_{10} \Delta IAGE$ after 5 Newton iterations

c) Histogram of $\log_{10} \Delta IAGE$ after 10 Newton iterations

figure from Lindsay, Ocean Model. (2017)
Shadow Tracers Applying NK Solver to MARBL

• Ecosystem variables are very non-linear on timescales much shorter than timescale that NK solver integrates over

• Apply NK solver to system without these non-linearities:
  1. Do short run to spin up biological pump (BP)
  2. Apply NK solver to spinup nutrients with BP prescribed from 1.
  3. goto 1.

• Running with prescribed biological pump implemented using shadow tracers; run model with full ecosys + shadow nutrients that get BP from real tracers; shadow tracer do not feed back to BP

• challenge: with BP uncoupled from shadow nutrients, they can go negative; restore to real tracers to prevent this
Convergence of Solver for Different Restoring Strategies in 1D Phosphorus Model

\[
\frac{1}{\tau} = \frac{1}{\text{hr}} \text{ in top layer} \quad \text{and} \quad \frac{1}{\tau} = \frac{\text{d (PO4 uptake)}}{\text{d (PO4)}}
\]
Ongoing Work

• Interannual variability of circulation
  – Is $\Phi(c^*, T) = c^*$ appropriate for non-cyclostationary circulation?
  – How many years are needed to be ‘representative’?

• What is the impact of approximations in preconditioner on the solver’s rate of convergence?

• Shadow tracer formulation for multiple nutrients
  – Solving preconditioner equations for multiple coupled tracers is memory intensive

• Deal with strong non-linearity of $O_2$ consumption at low $O_2$ values
  – Enforce physical constraints in line search along Newton increment direction

• Apply shadow tracer approach to temperature and salinity
  – Appropriate restoring formulation is not obvious
MOM6
(thanks to Andrew Shao for valuable input)

• Interpreting $G(c) \equiv \Phi(c,T) - c$ with time-varying vertical coordinate
  – initial approach is to evaluate both sides on fixed $z^*$ coordinate

• Extend implementation of lateral terms in preconditioner to handle tripole seam.

• Get appropriate diagnostics for preconditioner
  – Some thought required for Shao neutral density mixing scheme. Naïve implementation may lead to significant fill in and memory hit. Approximations to scheme may be needed.
Extra Slides
Some Solver Implementation Features

• implemented in python, facilitating file manipulation and data processing in a single environment

• solver can persist in memory, but this is not necessary
  – solver has restart-like capability to exit and later resume where it exited
  – exiting is done after submitting a model run to the batch system

• solver works on an abstract ModelState class that has a method for computing a function that the Newton-Krylov solver finds a zero of

• subclasses of ModelState have specifics for particular applications
  – cime_pop: for tracers in POP run through CESM
  – another subclass simulates 1d tracers with source-sinks and time-varying vertical mixing; a phosphorus tracer module is a BGC prototype

• git/github is used for version control

• documentation available at readthedocs.io (work in progress)
Krylov Method Algorithm

**Algorithm 9.4 GMRES with Left Preconditioning**

1. \[ \text{Compute } r_0 = M^{-1}(b - Ax_0), \beta = ||r_0||_2 \text{ and } v_1 = r_0 / \beta \]
2. For \( j = 1, \ldots, m \) Do:
3. \[ \text{Compute } w := M^{-1} A v_j \]
4. For \( i = 1, \ldots, j \), Do:
5. \[ h_{i,j} := (w, v_i) \]
6. \[ w := w - h_{i,j} v_i \]
7. EndDo
8. \[ \text{Compute } h_{j+1,j} = ||w||_2 \text{ and } v_{j+1} = w / h_{j+1,j} \]
9. EndDo
10. Define \( V_m := [v_1, \ldots, v_m] \), \( \bar{H}_m = \{ h_{i,j} \}_{1 \leq i \leq j+1; 1 \leq j \leq m} \)
11. Compute \( y_m = \text{argmin}_y ||\beta e_1 - \bar{H}_m y||_2 \), and \( x_m = x_0 + V_m y_m \)
12. If satisfied Stop, else set \( x_0 := x_m \) and GoTo 1

'Some implementation details:

Multiplication by matrix \( A \) invokes a forward model run. In batch environment, solver exists after submitted job, and solver is reinvoked after job completes. Coefficients are stored in solver state 'restart' file.

Additional dimensions for scalars:

Region dimension: Applications are to enable simultaneous spin-up in open-ocean and Black Sea for an ocean model, separate ice sheets for a land ice model, separate columns for a land model.

Tracer module dimension: Tracers are grouped into modules of related/coupled tracers. Separate modules enables simultaneous spin-up of uncoupled tracers.