OM-II

- Chris Hill
- A small adjoint trick
- Some parameterization motivation for future plans
- Some future plans

About me - One of my favorite pets….

Actual start was ~May 21, 1993 – but we (myself, Alistair Adcroft, Lev Perelman,……) didn't discover version control until 1998…..

My favorite adjoint trick

Using fictitious (0 in forward run) terms to compute specific adjoint sensitivities.

Evaluating carbon sequestration efficiency in an ocean circulation model by adjoint sensitivity analysis

Chris Hill, Véronique Bugnion, Mick Follows, and John Marshall

$$\frac{\partial C}{\partial t} = -\vec{U} \cdot \nabla C + \nabla \cdot (K \nabla C) + \Gamma(C) - \mu C + S$$

S is a fictitious source that is set to 0 in forward run. In adjoint it will accumulate sensitivity of cost function to a continuous source.

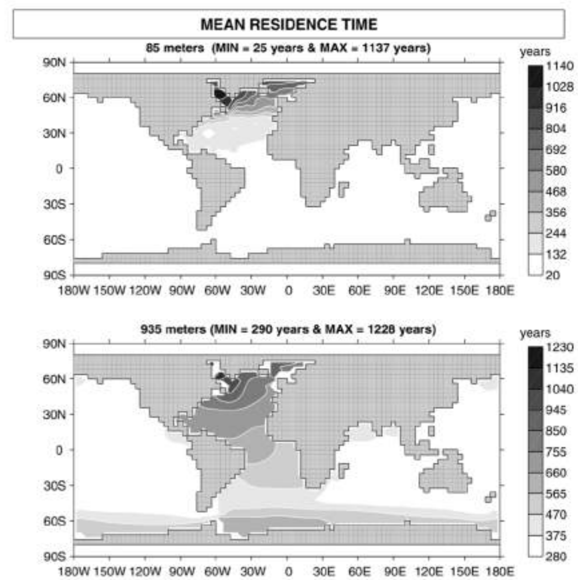
$$J(t = T) = \int_{t=0}^{t=T} \int_A \mu C \Delta z dA dt$$

Cost function outgassing of a carbon like gas

$$\widetilde{S} = \frac{1}{tV} \frac{\partial J}{\partial S}(\lambda, \phi, z, t)$$

$$S^*(\lambda, \phi, z, t) = \frac{\partial J}{\partial S}(\lambda, \phi, z, t)$$

Adjoint sensitivity S* is computed even if S=0



Adjoint trick works for dynamics fields too

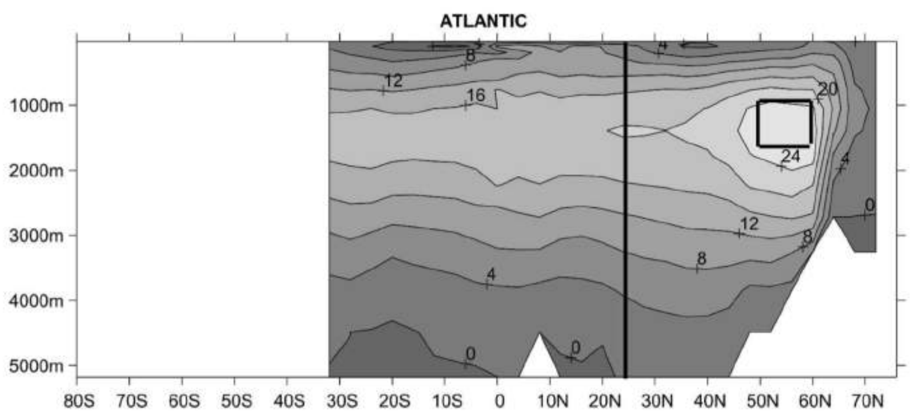
An Adjoint Analysis of the Meridional Overturning Circulation in a Hybrid Coupled Model

VÉRONIQUE BUGNION, CHRIS HILL, AND PETER H. STONE

Massachusetts Institute of Technology, Cambridge, Massachusetts

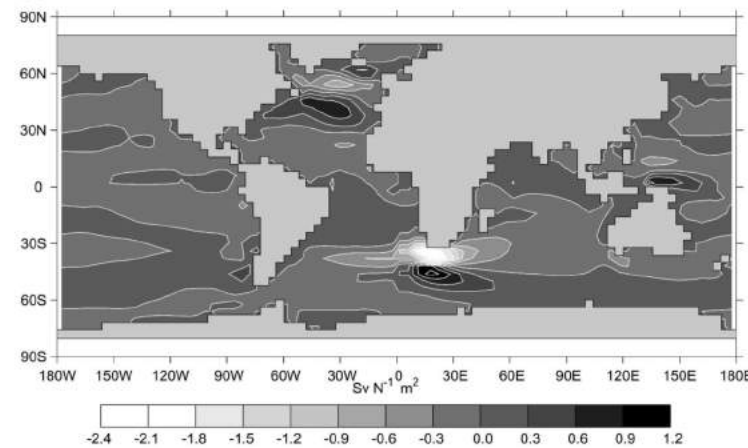
$$J = \psi_{max}$$

$$\psi_{\mathrm{MAX}} = \overline{\psi}(52 - 60°\mathrm{N}, 80 - 0°\mathrm{W}, 1055 - 1395\ \mathrm{m})$$



Overturning strength

(a) Windstress sensitivity



$$\frac{\partial u}{\partial t} = \cdots + \frac{\partial \Psi}{\partial y}$$

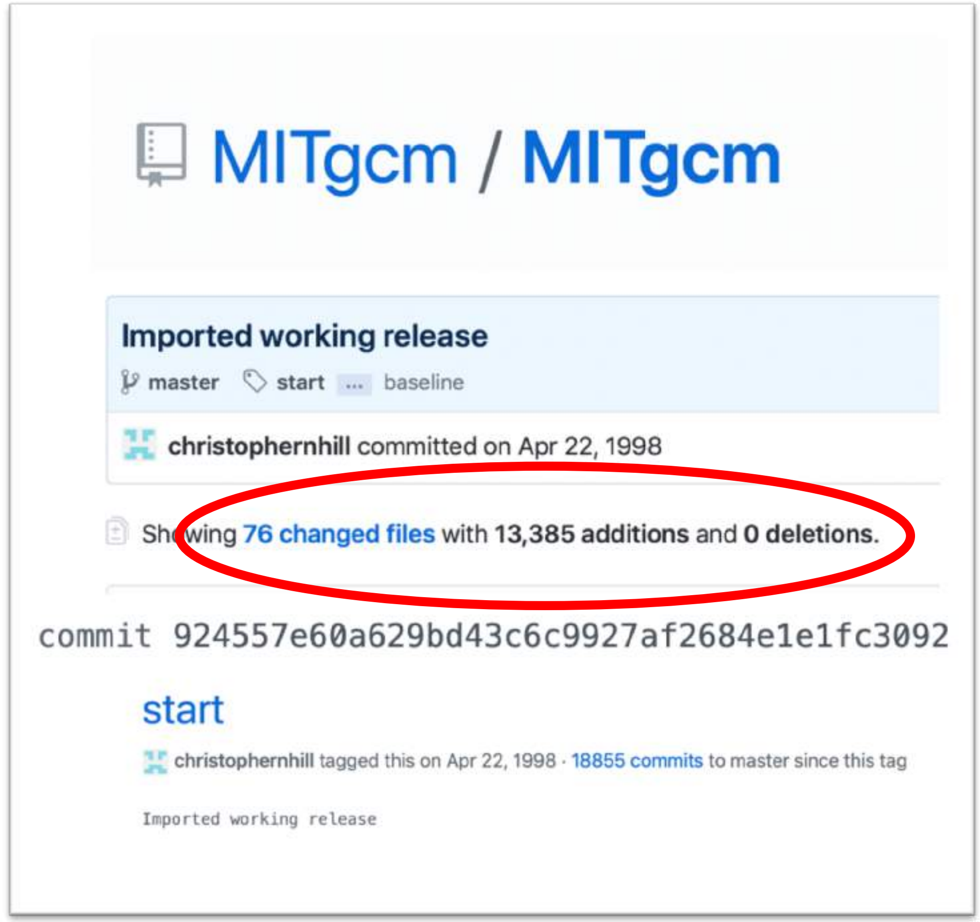
$$\frac{\partial v}{\partial t} = \cdots - \frac{\partial \Psi}{\partial x}$$

$\Psi = 0$ can be used to calculate sensitivity without altering dynamics.

OM-II

- Chris Hill
- A small adjoint trick
- Some parameterization motivation for future plans
- Some future plans

About me - One of my pets....



Some parameterization motivation for future plans

- Numerical ocean models are finite resolution in space and time
  - Underlying PDEs assume infinite resolution in space and time
  - Truncation of a finite dimensional model inevitably creates a need for parameterization/approximation terms
- e.g.
  - Sub-grid parameterizations/closures
    - Representing bulk effect of small scale in space and time (full talk Sonya next Thur)
  - Other approximations e.g.
    - Surface flux computations
    - Equation of state
    - Riverine fluxes
    - Mass v. volume conserving
    - etc…
- **Parameterizations and other approximations play a O(1) role in solutions.**

Velocity

horiz. velocity; $\sum$ stress, coriolis, viscosity, transport, …

$$\frac{\partial \vec{u}_h}{\partial t} + \nabla_h (g \rho_{ref} \eta + p_{hyd}) = \vec{G}_{\vec{u}_h}$$

pressure gradient; sea-surface height anomaly

Thermodynamics and tracers

$\sum$ sources/sinks, transport, mixing, …

$$\frac{\partial \gamma}{\partial t} = \vec{G}_\gamma, \gamma = \{\theta, s, \lambda_m\}$$

temperature; salinity; other tracers

Pressure and continuity

$$\nabla_h^2 \eta + \frac{\partial \eta}{\partial t} = \nabla_h \cdot \vec{G}_{\vec{u}_h}$$

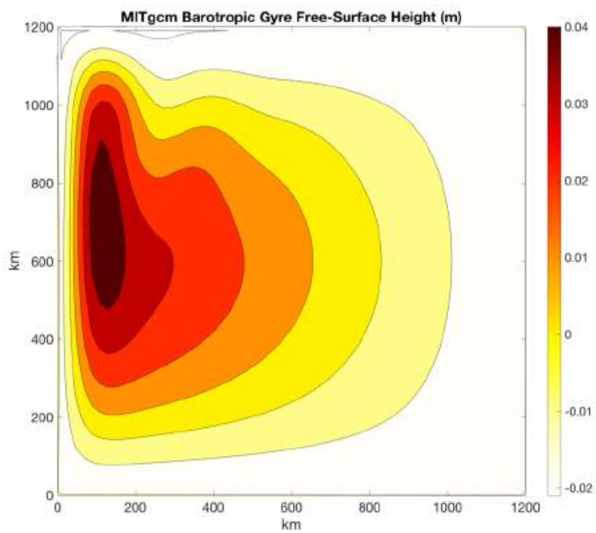
equation of state

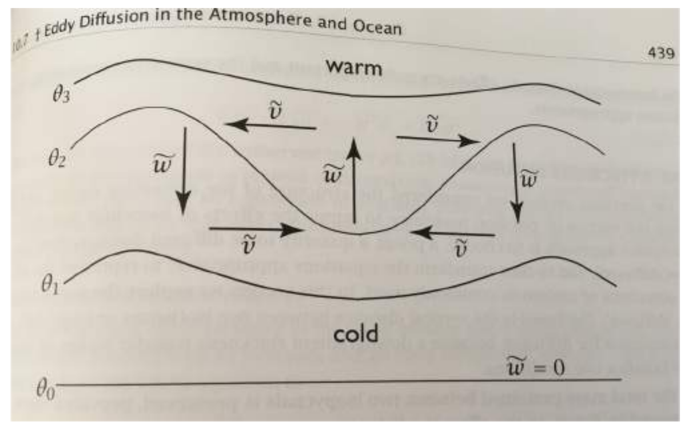
$$\frac{\partial p_{hyd}}{\partial z} = -g\rho(\theta, s, p(z)) \qquad \frac{\partial w}{\partial z} = -\nabla_h \cdot \vec{u}_h$$

https://mitgcm.readthedocs.io/en/latest/overview/overview.html#continuous-equations-in-r-coordinates

The O(1) role of "parameterization/closures"

- e.g. Stommel Gyre, Gent McWilliams





Recap - prognostic MITgcm equations basic timestep

$$\frac{\gamma^{n+1} - \gamma^{n}}{\Delta t} = \vec{G}_{\gamma}^{n+\frac{1}{2}}, \qquad \gamma = \{\theta, s, \lambda_m\}$$

Thermodynamics and tracers

$$\frac{\partial p_{hyd}^{n+1}}{\partial z} = -g\rho(\theta^{n+1}, s^{n+1}, z)$$

$$\nabla_h^2 \eta^{n+1} + \frac{\eta^{n+1} - \eta^{n}}{\Delta t} = \nabla_h \cdot \int_{-H}^{0} \left(\vec{G}_{\vec{u}_h}^{n+\frac{1}{2}} + \frac{\vec{u}_h^{n}}{\Delta t} - \nabla p_{hyd}^{n+\frac{1}{2}} \right)$$

$$\frac{\vec{u}_h^{n+1} - \vec{u}_h^{n}}{\Delta t} + \nabla_h (\eta^{n+1} - p_{hyd}^{n+\frac{1}{2}}) = \vec{G}_{\vec{u}_h}^{n+\frac{1}{2}}$$

$$\frac{\partial w^{n+1}}{\partial z} = -\nabla_h \cdot \vec{u}_h^{n+1}$$

Pressure, velocity and continuity.

Adams Bashforth

$$\phi^{n+\frac{1}{2}} = \left(\frac{3}{2} + \epsilon\right) \phi^{n} - \left(\frac{1}{2} + \epsilon\right) \phi^{n-1}$$

https://mitgcm.readthedocs.io/en/latest/algorithm/algorithm.html#synchronous-time-stepping-variables-co-located-in-time

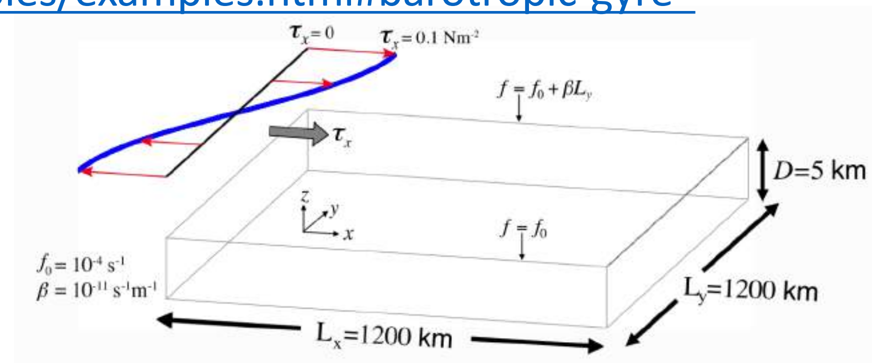
Simplest example?

- Given the basic equations a lot of the specialization to a specific model configuration is in defining the G terms.

$$\frac{\partial \vec{u}_h}{\partial t} + \nabla_h (g \rho_{ref} \eta + p_{hyd}) = \vec{G}_{\vec{u}_h}$$

horiz. velocity; $\sum$ *stress, coriolis, viscosity, transport, …*; *pressure gradient*; *sea-surface height anomaly*

  - Single layer, Gyre example (https://mitgcm.readthedocs.io/en/latest/examples/examples.html#barotropic-gyre-mitgcm-example)



$$\vec{G}_{\vec{u}_h} = (G_u, G_v), \vec{u}_h = (u, v)$$

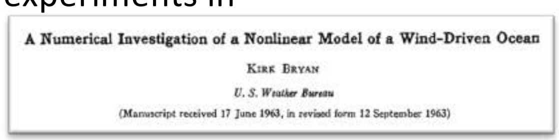
$$G_u = f v + A \nabla^2 u + \tau_x(y) - u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y}$$

wind stress, *coriolis*, *eddy visc.*, *transport*

$$G_v = -f u + A \nabla^2 v - u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y}$$

$$\nabla^2 \eta + \frac{\partial \eta}{\partial t} = \nabla \cdot \vec{G}_{\vec{u}_h}$$

This elementary, illustrative setup is useful for understanding circulation model basics. It is similar to experiments in

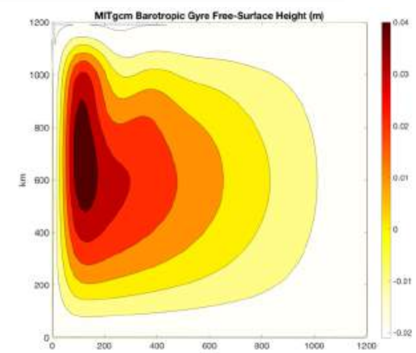


A Numerical Investigation of a Nonlinear Model of a Wind-Driven Ocean

KIRK BRYAN

U. S. Weather Bureau

(Manuscript received 17 June 1963, in revised form 12 September 1963)



Even simpler .....

- We can reduce to a linear problem by modifying G terms.

$$\frac{\partial \vec{u}_h}{\partial t} + \nabla_h (g\rho_{ref}\eta + p_{hyd}) = \vec{G}_{\vec{u}_h}$$

horiz. velocity; $\sum$ stress, coriolis, viscosity, transport, ...; pressure gradient; sea-surface height anomaly

  - Single layer, **linear** Gyre example (https://mitgcm.readthedocs.io/en/latest/examples/examples.html#barotropic-gyre-mitgcm-example)

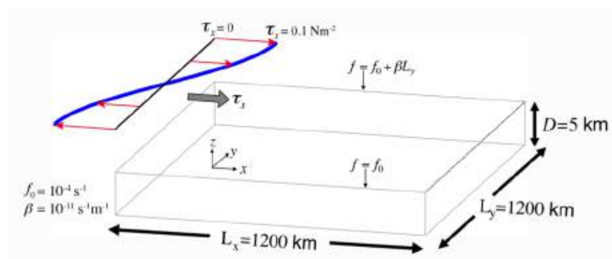
$$\vec{G}_{\vec{u}_h} = (G_u, G_v), \vec{u}_h = (u, v)$$

$$G_u = fv + A\,\nabla^2 u + \tau_x(y) - u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y}$$

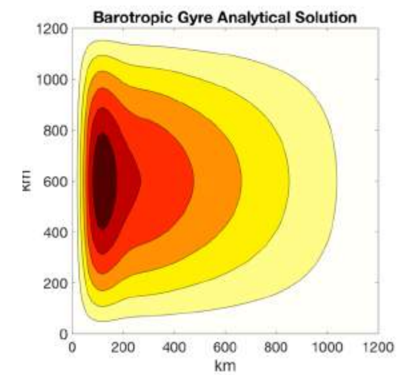
$$G_v = -fu + A\,\nabla^2 v - u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y}$$

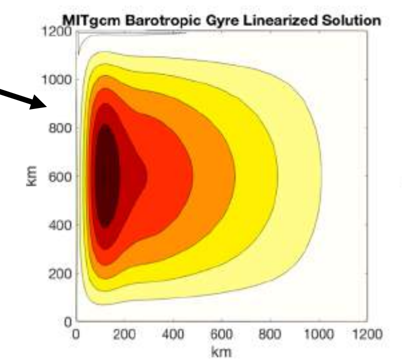
$$\nabla^2 \eta + \frac{\partial \eta}{\partial t} = \nabla \cdot \vec{G}_{\vec{u}_h}$$

(The advective terms $u\frac{\partial u}{\partial x}$, $v\frac{\partial u}{\partial y}$, $u\frac{\partial v}{\partial x}$, $v\frac{\partial v}{\partial y}$ are crossed out.)



modifying G terms gives slightly different mathematical model (and solution)





This is similar to analyses in

THE WESTWARD INTENSIFICATION OF WIND-DRIVEN OCEAN CURRENTS

Henry Stommel

(Contribution No. 408, Woods Hole Oceanographic Institution)

ON THE WIND-DRIVEN OCEAN CIRCULATION

By Walter H. Munk

Institute of Geophysics and Scripps Institution of Oceanography, University of California

(Manuscript received 24 September 1949)

Some MITgcm numerical modeling details from this example

- Discrete grid (https://mitgcm.readthedocs.io/en/latest/algorithm/algorithm.html#notation, https://mitgcm.readthedocs.io/en/latest/algorithm/horiz-grid.html)

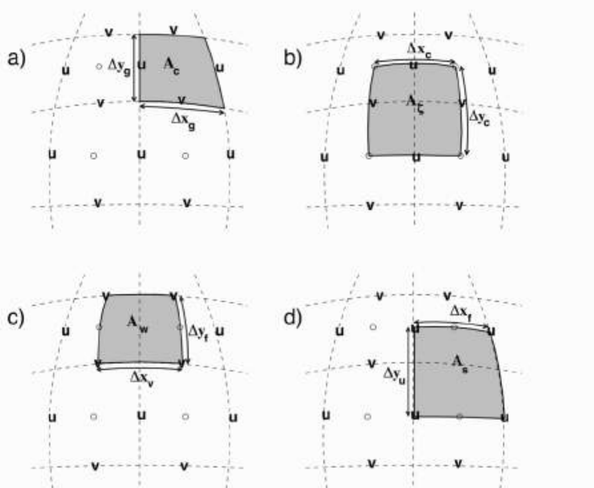


Figure 2.6 Staggering of horizontal grid descriptors (lengths and areas). The grid lines indicate the tracer cell boundaries and are the reference grid for all panels. a) The area of a tracer cell, A_c, is bordered by the lengths Δx_g and Δy_g. b) The area of a vorticity cell, A_ζ, is bordered by the lengths Δx_c and Δy_c. c) The area of a u cell, A_w, is bordered by the lengths Δx_v and Δy_f. d) The area of a v cell, A_s, is bordered by the lengths Δx_f and Δy_u.

General notation:

$\Delta x, \Delta y, \Delta r$ grid spacing in X, Y, R directions

A_c, A_w, A_s, A_ζ : horizontal area of a grid cell surrounding θ, u, v, ζ point

$\mathcal{V}_u, \mathcal{V}_v, \mathcal{V}_w, \mathcal{V}_\theta$: Volume of the grid box surrounding u, v, w, θ point

i, j, k : current index relative to X, Y, R directions

Basic operators:

$\delta_i : \delta_i \Phi = \Phi_{i+1/2} - \Phi_{i-1/2}$

$\overline{\ \ }^i : \overline{\Phi}^i = (\Phi_{i+1/2} + \Phi_{i-1/2})/2$

$\delta_x : \delta_x \Phi = \frac{1}{\Delta x} \delta_i \Phi$

$\overline{\nabla}$ = horizontal gradient operator : $\overline{\nabla} \Phi = \{\delta_x \Phi, \delta_y \Phi\}$

$\overline{\nabla} \cdot$ = horizontal divergence operator : $\overline{\nabla} \cdot \vec{\mathrm{f}} = \frac{1}{\mathcal{A}} \{\delta_i \Delta y \, \mathrm{f}_x + \delta_j \Delta x \, \mathrm{f}_y\}$

$\overline{\nabla}^2$ = horizontal Laplacian operator : $\overline{\nabla}^2 \Phi = \overline{\nabla} \cdot \overline{\nabla} \Phi$

Some numerical modeling details from this example (and code!)

$$- u \frac{\partial u}{\partial x}$$

- Momentum advection (https://mitgcm.readthedocs.io/en/latest/algorithm/algorithm.html#advection-of-momentum)

2.14.1. Advection of momentum

The advective operator is second order accurate in space:

$$A_w \Delta r_f h_w G_u^{adv} = \delta_i \overline{U}^i \overline{u}^i + \delta_j \overline{V}^i \overline{u}^j + \delta_k \overline{W}^i \overline{u}^k \quad (2.89)$$

$$A_s \Delta r_f h_s G_v^{adv} = \delta_i \overline{U}^j \overline{v}^i + \delta_j \overline{V}^j \overline{v}^j + \delta_k \overline{W}^j \overline{v}^k \quad (2.90)$$

$$A_c \Delta r_c G_w^{adv} = \delta_i \overline{U}^k \overline{w}^i + \delta_j \overline{V}^k \overline{w}^j + \delta_k \overline{W}^k \overline{w}^k \quad (2.91)$$

and because of the flux form does not contribute to the global budget of linear momentum. The quantities U, V and W are volume fluxes defined:

$$U = \Delta y_g \Delta r_f h_w u \quad (2.92)$$

$$V = \Delta x_g \Delta r_f h_s v \quad (2.93)$$

$$W = A_c w \quad (2.94)$$

The advection of momentum takes the same form as the advection of tracers but by a translated advective flow. Consequently, the conservation of second moments, derived for tracers later, applies to u^2 and v^2 and w^2 so that advection of momentum correctly conserves kinetic energy.

S/R MOM_U_ADV_UU, MOM_U_ADV_VU, MOM_U_ADV_WU

uu, vu, wu : fZon, fMer, fVerUkp (local to MOM_FLUXFORM.F)

```
#include "MOM_FLUXFORM_OPTIONS.h"

CBOP
C !ROUTINE: MOM_U_ADV_UU

C !INTERFACE: ==========================================================
      SUBROUTINE MOM_U_ADV_UU(
     I           bi,bj,k,
     I           uTrans, uFld,
     O           AdvectFluxUU,
     I           myThid)

C !DESCRIPTION:
C Calculates the zonal advective flux of zonal momentum:
C \begin{equation*}
C F^x = \overline{U}^i \overline{u}^{i}
C \end{equation*}

C !USES: ===============================================================
      IMPLICIT NONE
#include "SIZE.h"
#include "EEPARAMS.h"
#include "PARAMS.h"
#include "GRID.h"

C !INPUT PARAMETERS: ===================================================
C  bi,bj                :: tile indices
C  k                    :: vertical level
C  uTrans               :: zonal transport
C  uFld                 :: zonal flow
C  myThid               :: thread number
      INTEGER bi,bj,k
      _RL uTrans(1-OLx:sNx+OLx,1-OLy:sNy+OLy)
      _RL uFld(1-OLx:sNx+OLx,1-OLy:sNy+OLy)
      INTEGER myThid

C !OUTPUT PARAMETERS: ==================================================
C  AdvectFluxUU         :: advective flux
      _RL AdvectFluxUU(1-OLx:sNx+OLx,1-OLy:sNy+OLy)

C !LOCAL VARIABLES: ====================================================
C  i,j                  :: loop indices
      INTEGER i,j
CEOP

      DO j=1-Oly,sNy+Oly-1
       DO i=1-Olx,sNx+Olx-1
        AdvectFluxUU(i,j) =
     &  0.25*( uTrans(i,j) + uTrans(i+1,j) )
#ifdef MOM_BOUNDARY_CONSERVE
     &      *(   uFld(i,j)*_maskW(i+1,j,k,bi,bj)
     &         + uFld(i+1,j)*_maskW(i,j,k,bi,bj) )
#else
     &      *(   uFld(i,j) +   uFld(i+1,j) )
#endif
       ENDDO
      ENDDO

      RETURN
      END
```

https://github.com/MITgcm/MITgcm/blob/master/pkg/mom_fluxform/mom_u_adv_uu.F

Some numerical modeling details from this example (and code!)

- Coriolis (https://mitgcm.readthedocs.io/en/latest/algorithm/algorithm.html - coriolis-terms)

```
#include "MOM_FLUXFORM_OPTIONS.h"

CBOP
C !ROUTINE: MOM_U_CORIOLIS

C !INTERFACE: ==========================================================
      SUBROUTINE MOM_U_CORIOLIS(
     I        bi,bj,k,vFld,
     U        uCoriolisTerm,
     I        myThid)

C !DESCRIPTION:
C Calculates the horizontal Coriolis term in the zonal equation:
C \begin{equation*}
C \overline{f}^i \overline{v}^{ij}
C \end{equation*}

C !USES: ===============================================================
      IMPLICIT NONE
#include "SIZE.h"
#include "EEPARAMS.h"
#include "PARAMS.h"
#include "GRID.h"
#include "SURFACE.h"

C !INPUT PARAMETERS: ===================================================
C  bi,bj                :: tile indices
C  k                    :: vertical level
C  vFld                 :: meridional flow
C  myThid               :: thread number
      INTEGER bi,bj,k
      _RL vFld(1-OLx:sNx+OLx,1-OLy:sNy+OLy)
      INTEGER myThid

C !OUTPUT PARAMETERS: ==================================================
C  uCoriolisTerm        :: Coriolis term
      _RL uCoriolisTerm(1-OLx:sNx+OLx,1-OLy:sNy+OLy)
```

fv

```
      IF (useEnergyConservingCoriolis) THEN
C Energy conserving discretization
       DO j=1-Oly,sNy+Oly-1
        DO i=1-Olx+1,sNx+Olx
         uCoriolisTerm(i,j) =
     &     0.5*( _fCori( i ,j,bi,bj)
     &          *0.5*( vFld( i ,j)+vFld( i ,j+1) )
     &         + _fCori(i-1,j,bi,bj)
     &          *0.5*( vFld(i-1,j)+vFld(i-1,j+1) ) )
        ENDDO
       ENDDO
      ELSE
C Original discretization
       DO j=1-Oly,sNy+Oly-1
        DO i=1-Olx+1,sNx+Olx
         uCoriolisTerm(i,j) =
     &     0.5*( _fCori( i ,j,bi,bj) +
     &           _fCori(i-1,j,bi,bj)  )
     &   *0.25*(
     &     vFld( i ,j)+vFld( i ,j+1)
     &    +vFld(i-1,j)+vFld(i-1,j+1)
     &          )
        ENDDO
       ENDDO
      ENDIF

      IF (useJamartWetPoints) THEN
C Scale term so that only "wet" points are used
C Due to: Jamart and Ozer, 1986, JGR 91 (C9), 10,621-10,631
C         "Numerical Boundary Layers and Spurious Residual Flows"
       DO j=1-Oly,sNy+Oly-1
        DO i=1-Olx+1,sNx+Olx
         uCoriolisTerm(i,j) = uCoriolisTerm(i,j)
     &   *4. _d 0/MAX( one,
     &                maskS( i , j ,k,bi,bj)+maskS( i ,j+1,k,bi,bj)
     &               +maskS(i-1, j ,k,bi,bj)+maskS(i-1,j+1,k,bi,bj) )
        ENDDO
       ENDDO
      ENDIF

      RETURN
      END
```

2.14.2. Coriolis terms

The "pure C grid" Coriolis terms (i.e. in absence of C-D scheme) are discretized:

$$\mathcal{A}_w \Delta r_f h_w G_u^{Cor} = \overline{f \mathcal{A}_c \Delta r_f h_c \overline{v}^j}^i - \epsilon_{nh} \overline{f' \mathcal{A}_c \Delta r_f h_c \overline{w}^k}^i \quad (2.95)$$

$$\mathcal{A}_s \Delta r_f h_s G_v^{Cor} = -\overline{f \mathcal{A}_c \Delta r_f h_c \overline{u}^i}^j \quad (2.96)$$

$$\mathcal{A}_c \Delta r_c G_w^{Cor} = \epsilon_{nh} \overline{f' \mathcal{A}_c \Delta r_f h_c \overline{u}^i}^k \quad (2.97)$$

where the Coriolis parameters f and f' are defined:

$$f = 2\Omega \sin \varphi$$
$$f' = 2\Omega \cos \varphi$$

where φ is geographic latitude when using spherical geometry, otherwise the β-plane definition is used:

$$f = f_o + \beta y$$
$$f' = 0$$

This discretization globally conserves kinetic energy. It should be noted that despite the use of this discretization in former publications, all calculations to date have used the following different discretization:

$$G_u^{Cor} = f_u \overline{v}^{ji} - \epsilon_{nh} f'_u \overline{w}^{ik} \quad (2.98)$$

$$G_v^{Cor} = -f_v \overline{u}^{ij} \quad (2.99)$$

$$G_w^{Cor} = \epsilon_{nh} f'_w \overline{u}^{ik} \quad (2.100)$$

where the subscripts on f and f' indicate evaluation of the Coriolis parameters at the appropriate points in space. The above discretization does *not* conserve anything, especially energy, but for historical reasons is the default for the code. A flag controls this discretization: set run-time logical useEnergyConservingCoriolis to .TRUE. which otherwise defaults to .FALSE..

S/R CD_CODE_SCHEME, MOM_U_CORIOLIS, MOM_V_CORIOLIS

G_u^{Cor}, G_v^{Cor} : cF (local to MOM_FLUXFORM.F)

Some numerical/physical parameter detail for Gyre example

- The experiment also illustrates standard numerical stability criteria
  - Velocity and timestep

$$2\left(\frac{|u|\Delta t}{\Delta x}\right) < 0.5$$

  - Coriolis and timestep

$$f\Delta t < 0.5$$

  - for Gyre example these implies

$$\Delta t = 1200 \text{ s}$$

- The eddy viscosity term has a basic stability too

$$2\left(4\frac{A_h \Delta t}{\Delta x^2}\right) < 0.6$$

- however, it cannot be too small because it also determines the Western boundary dissipation of vorticity. This must be resolved on the grid ➔

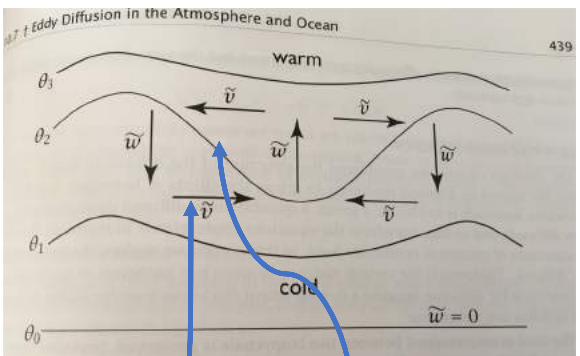
$$\frac{2\pi}{\sqrt{3}}\left(\frac{A}{\beta}\right)^{\frac{1}{3}} \approx 4\Delta x$$

Even for simplest possible problem the "eddy closure" has an O(1) effect on solution.

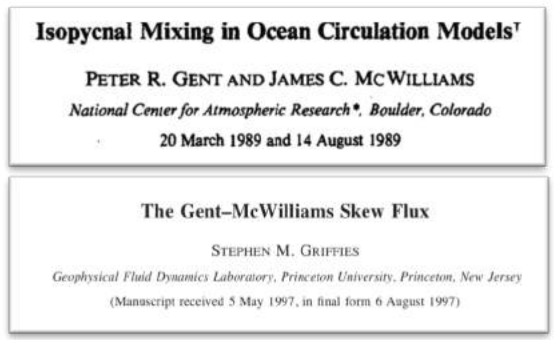
https://mitgcm.readthedocs.io/en/latest/examples/examples.html#numerical-stability-criteria

What about parameterization role in more realistic model.

- One big term that is lost when coarsening model to $\Delta \geq 10$km is mesoscale eddy effect.



Vallis AOFD, 2d illustration example



Isopycnal Mixing in Ocean Circulation Models

PETER R. GENT AND JAMES C. MCWILLIAMS

National Center for Atmospheric Research, Boulder, Colorado*

20 March 1989 and 14 August 1989

The Gent–McWilliams Skew Flux

STEPHEN M. GRIFFIES

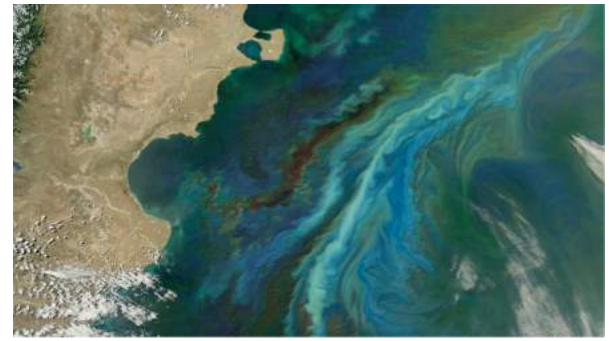
Geophysical Fluid Dynamics Laboratory, Princeton University, Princeton, New Jersey

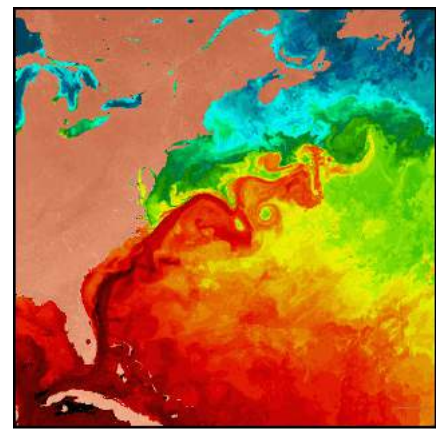
(Manuscript received 5 May 1997, in final form 6 August 1997)

$$\psi^* = \kappa_{GM} S^x$$
$$\tilde{v} = -\psi_z^*$$
$$\tilde{w} = \psi_x^*$$

Mathematically, shown here in 2d, GM defines an overturning stream function, ψ^*, that depends on the isopycnal slope, $S^x = -\frac{\rho_x}{\rho_z}$ and that works to flatten slope without changing moments.

Ocean Color off Patagonia





Gulf Stream SST

Gent McWilliams parameterization was a big step in ocean modeling.

Subgrid Parameterization GM – "Gent McWilliams" - modeling details I

Mathematically GM (1990) defined an overturning stream function that depended on the isopycnal slope and acts to "flatten" sloping isopycnals. Where κ_{GM} is a mixing coefficient that reflects the efficiency of geostrophic eddies in extracting potential energy ($\kappa_{GM}$is uncertain and is used as a control in some ECCO calculations). In practice, however, the GM velocities, written $(\tilde{u}, \tilde{v}, \widetilde{w})$ in full 3d, in their "advective" form.

Griffies (1998) showed that numerically it can be preferable to formulate the stream function transport in terms of a skew flux. In full 3d (acting on θ for example) this gives a term

$$G_{gmredi} = -\underline{A}\nabla\theta \, , A = \kappa_{GM} \begin{bmatrix} 0 & 0 & -S^x \\ 0 & 0 & -S^y \\ S^x & S^y & 0 \end{bmatrix}$$

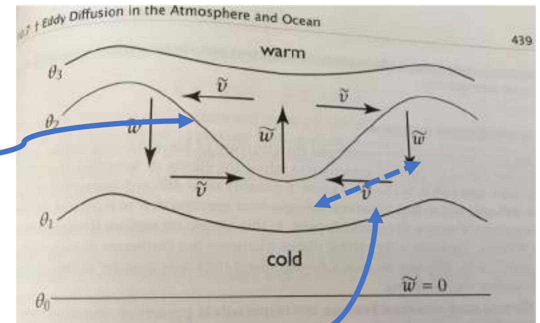
$$S^x = -\frac{\rho_x}{\rho_z}$$

that then appears as a subgrid term in the tracer equations.

Subgrid Parameterization GM – "Gent McWilliams" - modeling details II

$$G_{gmredi} = -\underline{A}\nabla\theta \,, A = \kappa_{GM}\begin{bmatrix} 0 & 0 & -S^x \\ 0 & 0 & -S^y \\ S^x & S^y & 0 \end{bmatrix}$$

$$S^x = -\frac{\rho_x}{\rho_z}$$



Oceanic Isopycnal Mixing by Coordinate Rotation

MARTHA H. REDI[1,2]

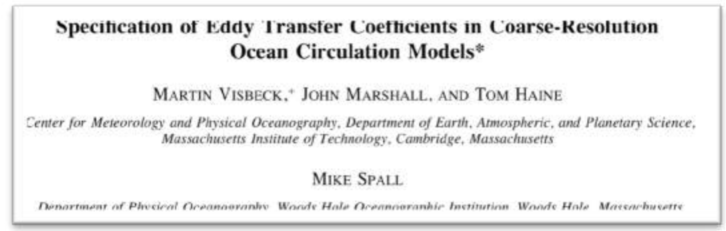
The skew flux term can combine nicely with the background along isopycnal diffusion. When the isopycnal slopes are small (i.e. the ocean interior) and $\kappa_{GM} = \kappa_{Redi}$, various terms cancel and a combined Redi/GM operator for sub-grid eddy diffusivity is

$$\boldsymbol{A} = \boldsymbol{\kappa}_{GM}\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{2S^x} & \mathbf{2S^y} & |\boldsymbol{S}^2| \end{bmatrix}$$

https://mitgcm.readthedocs.io/en/latest/phys_pkgs/gmredi.html#griffies-skew-flux

But we still have free parameters!

- GM is very elegant and physically based but….



Specification of Eddy Transfer Coefficients in Coarse-Resolution Ocean Circulation Models*

MARTIN VISBECK,+ JOHN MARSHALL, AND TOM HAINE

Center for Meteorology and Physical Oceanography, Department of Earth, Atmospheric, and Planetary Science, Massachusetts Institute of Technology, Cambridge, Massachusetts

MIKE SPALL

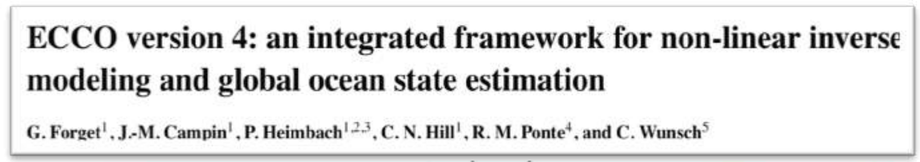
Department of Physical Oceanography, Woods Hole Oceanographic Institution, Woods Hole, Massachusetts

- Variable κ_{GM}: Variants on GM have dynamic κ_{GM} (based on local Richardson number, $\frac{N^2}{u_z^2}$) etc…

https://mitgcm.readthedocs.io/en/latest/phys_pkgs/gmredi.html#variable-kappa-gm

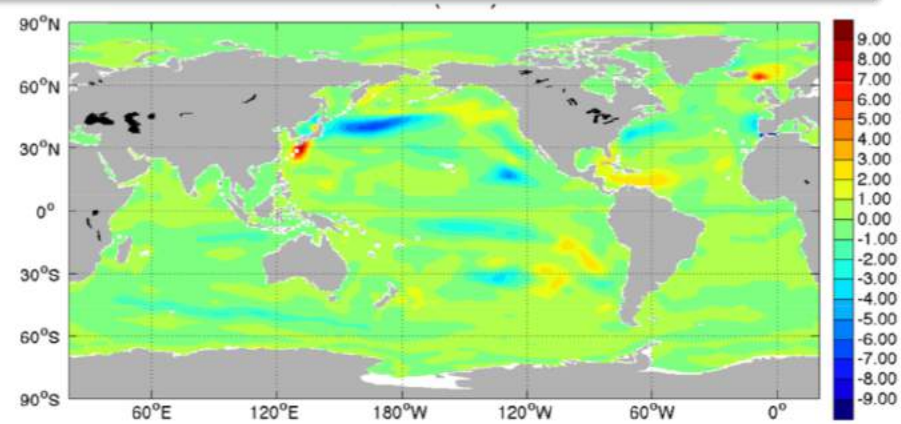
- Tapering: In the mixed layer isopycnal slopes can become large (or even infinite). Tapering schemes smooth the transition between ocean interior eddy mixing and mixed layer dynamics. Various tapering schemes exist, with sizable effects on solutions. The right approach is an open area of research

https://mitgcm.readthedocs.io/en/latest/phys_pkgs/gmredi.html#tapering-and-stability



ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation

G. Forget[1], J.-M. Campin[1], P. Heimbach[1,2,3], C. N. Hill[1], R. M. Ponte[4], and C. Wunsch[5]

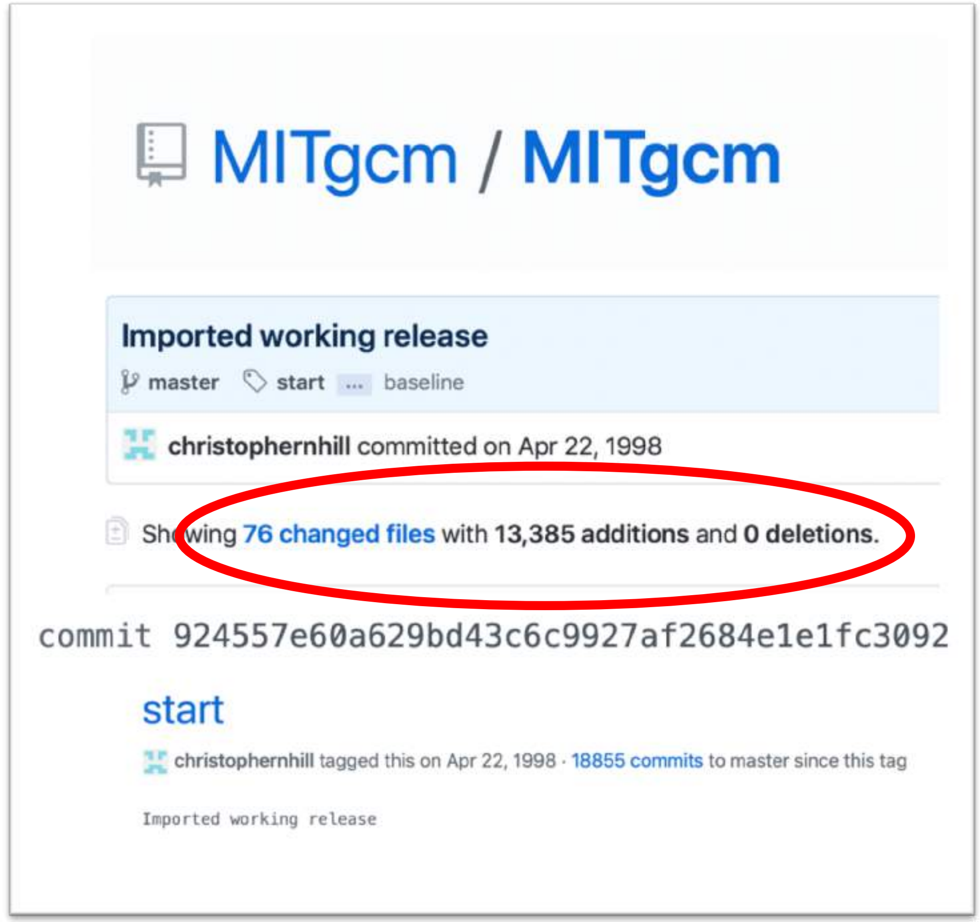


Sensitivities of Argo data misfit to κ_{GM} in Forget, 2015.

OM-II

- Chris Hill
- A small adjoint trick
- Some parameterization motivation for future plans
- Some future plans

About me - One of my pets….

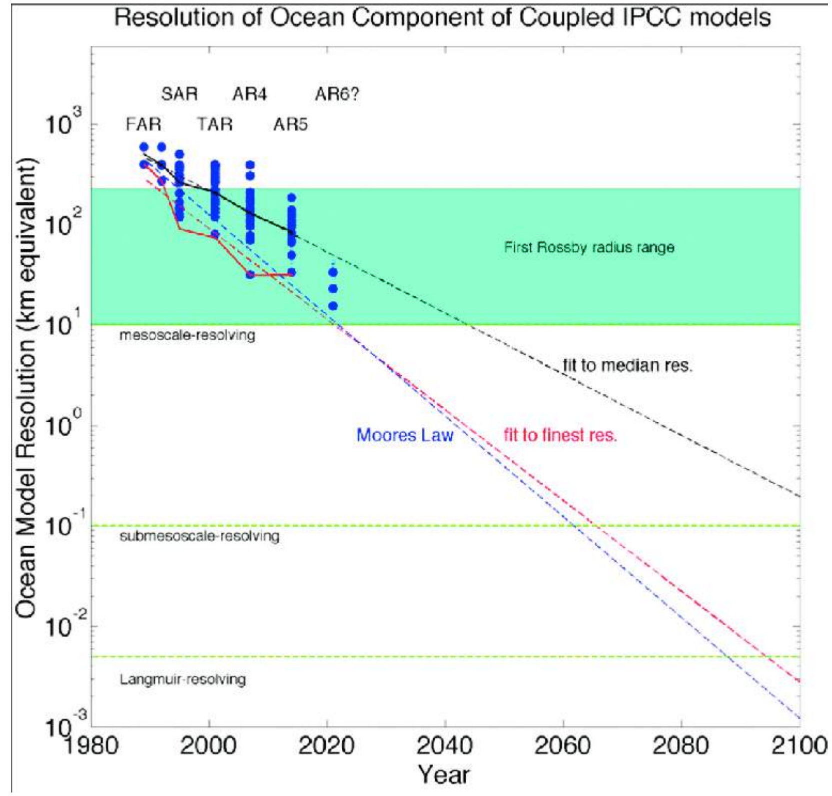


Looking forward

Even with worlds biggest computers, iterative, assimilation problems like ECCO will need to parameterize unresolved processes for the foreseeable future.

Core ocean model terms we would like to improve
- eddy parameterization
- mixed layer column
- transport

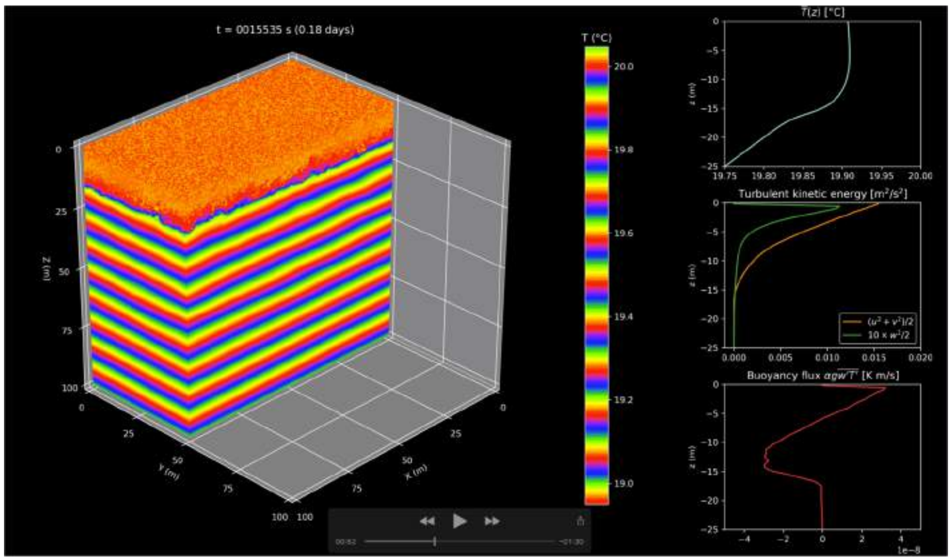
Relying on computing power improvements alone will likely not suffice.



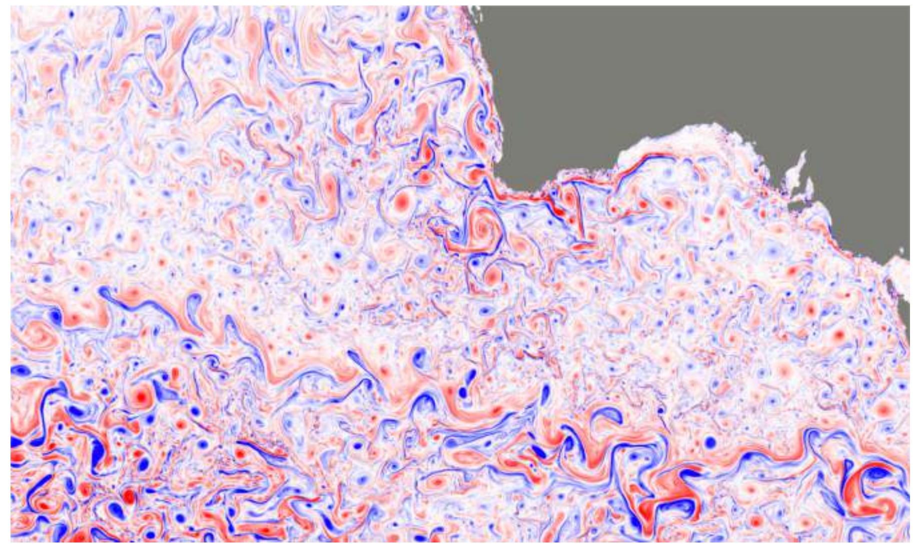
Baylor Fox-Kemperer

Can we create accurate models for resolving some of the parametrization challenges?

Mixed layer



Mesoscale eddy

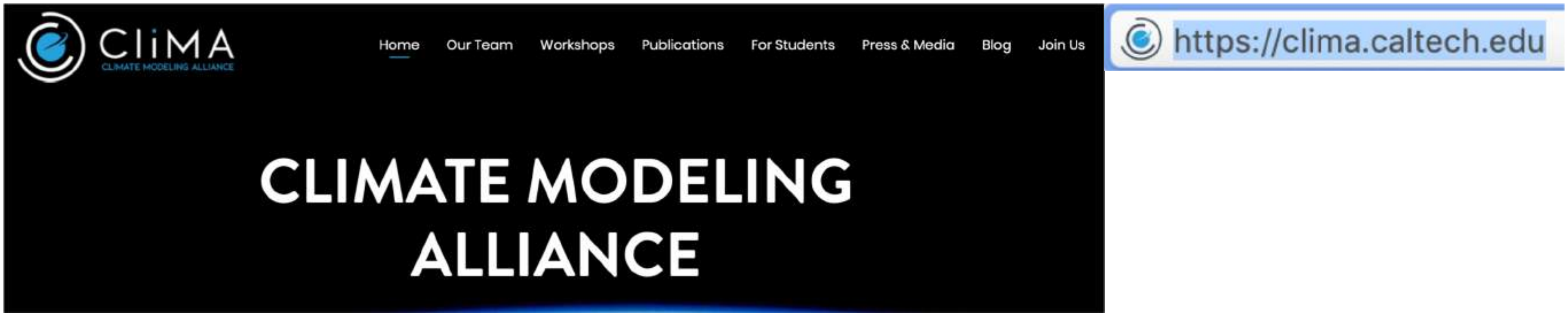


LES model for mixed layer column
Δ=50cm. (Ali Ramadhan)

LLC4320 – mesoscale resolving

These models can not run for long enough at scale for use in iteration, but (if they are accurate) they can in principle create synthetic fields that for developing parameterizations.

Can we frame parameterization improvement as an "assimilation like" problem?



Can we create highly resolved models that have enough skill to use as synthetic "obs" input for developing better parameterizations for use in projects like ECCO?

Starting to look at three areas

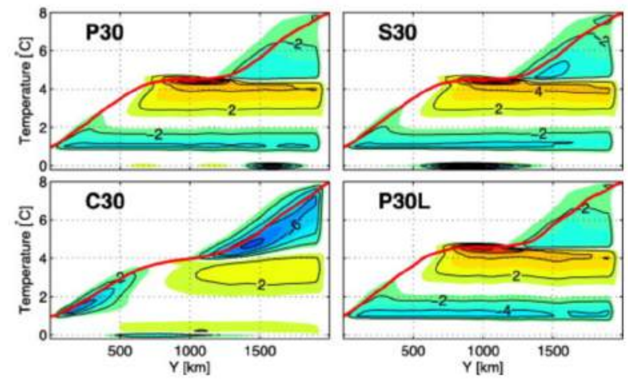
1. Improved numerics to yield better models
2. Application to LES style process models embedded in ECCO
3. Application to "super parameterization"+sub-mesoscale capable large scale models for mesoscale closures

1. High order numerics

- What numerics should we explore to ensure we can represent adiabatic interior and turbulent boundary layers all in same core.
- Earlier work shows the importance of high order tracer transport/advection schemes to maintaining an adiabatic interior.
- Exploratory work looking at potential role for discontinuous Galerkin (DG) techniques in moving more terms to high-order schemes.
- DG is a finite-volume based approach (like existing MITgcm) but considerably more complex.
- Used in engineering flows for accuracy and scaling.


Controlling spurious diapycnal mixing in eddy-resolving height-coordinate ocean models – Insights from virtual deliberate tracer release experiments

hris Hill David Ferreira Jean-Michel Campin John Marshall Ryan Abernathey Nicolas Barrier



LES style process models embedded in ECCO

1. **Current** - FV and DG process model configurations to train new parameterizations.
2. ➔ Future - New global model configurations using new parameterizations.

https://github.com/climate-machine

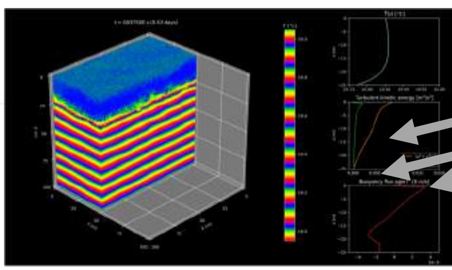
Mixed layer processes are a significant parameterization in current ECCO (KPP based). Together with Raf Ferrari, Greg Wagner, Andre Souza and others we are exploring using LES that samples the ECCO solution space. The explicit mixed layer solutions will be used for creating new ML parameterizations.

FV

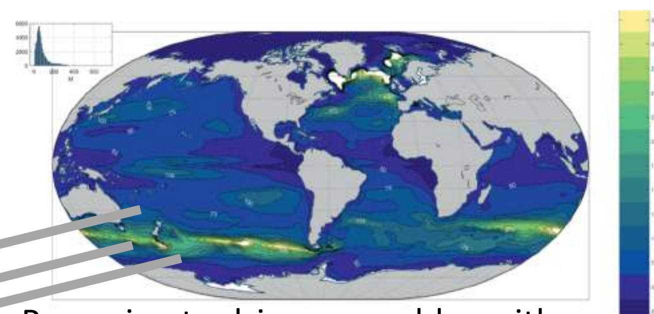
Oceananigans.jl

Development team

- Ali Ramadhan (@ali-ramadhan)
- Chris Hill (@christophernhill)
- Jean-Michel Campin (@jm-c)
- John Marshall (@johncmarshall54)
- Greg Wagner (@glwagner)
- Mukund Gupta (@mukund-gupta)
- Andre Souza (@sandreza)
- Zhen Wu (@zhenwu0728)
- Also big thanks to Valentin Churavy (@vchuravy) and Peter Ahrens (@peterahrens)!



Stand alone process model for idealized scenarios running today. Produces some initial synthetic training data.



Preparing to drive ensembles with realistic bcs sampled from ECCO ocean state estimates.

➔ New ML column dynamics parameterization; initially evaluate in ECCO setup.

DG

CLIMA

+ NPS, Caltech

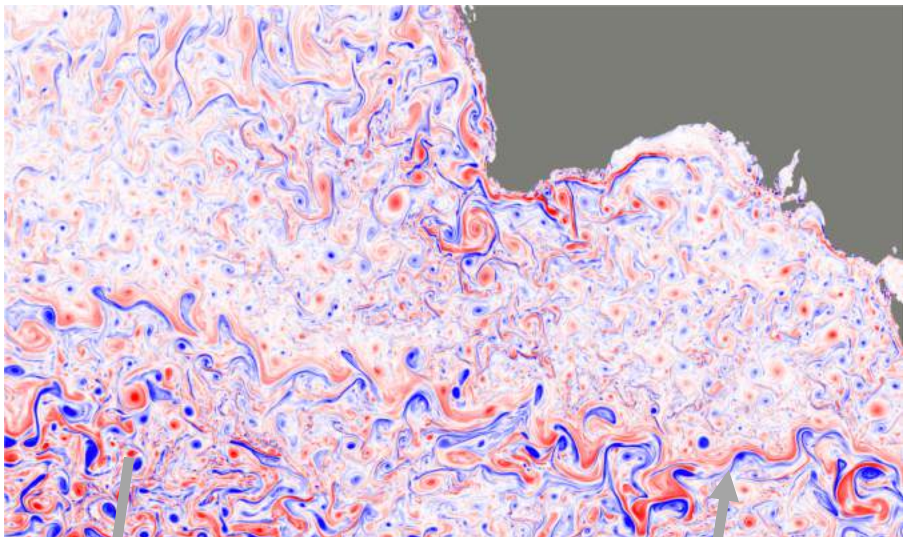
Currently making DG atmos kernel incompressible

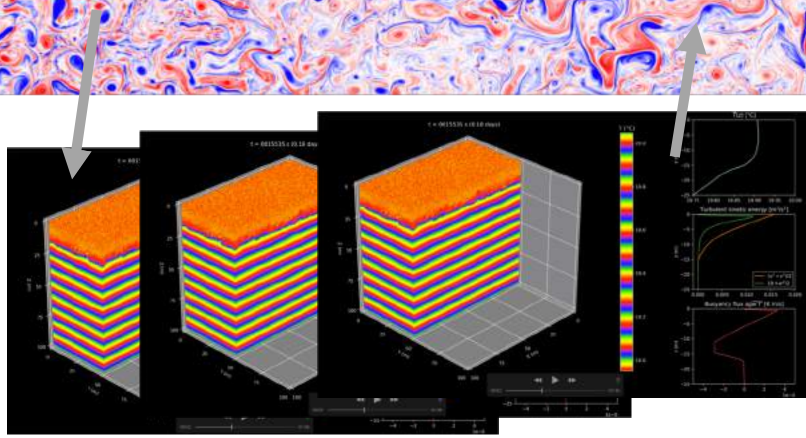
- approach based on CG/HDG concepts
- should allow equivalent process models to FV

Plausible hydrostatic form equations for large scale also identified.

What about synthetic systems for mesoscale parameterization formulation.

Superparameterization enhahanced sub-meso models.





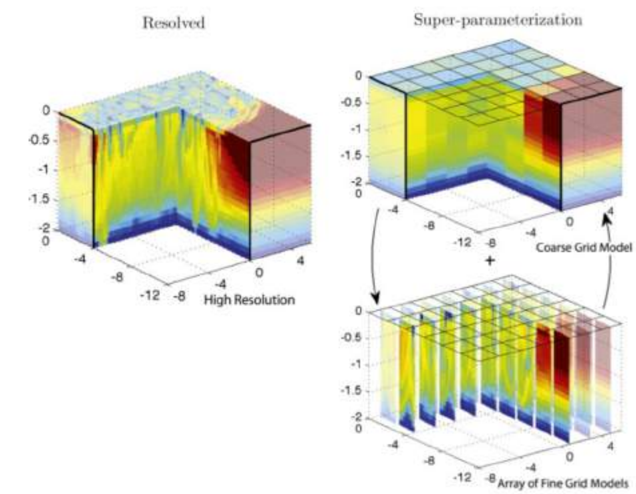
ELSEVIER

Ocean Modelling

Volume 36, Issues 1–2, 2011, Pages 90-101

Super-parameterization in ocean modeling: Application to deep convection

Jean-Michel Campin, Chris Hill, Helen Jones, John Marshall



$$\text{Fine}: \frac{\partial \underset{\sim}{\mathbf{v}}_{\mathbf{f}}}{\partial t} = -\underset{\sim}{\mathbf{v}}_{\mathbf{f}} \cdot \nabla \underset{\sim}{\mathbf{v}}_{\mathbf{f}} - 2\underset{\sim}{\Omega} \times \underset{\sim}{\mathbf{v}}_{\mathbf{f}} - \frac{1}{\rho_0}\nabla(P_h + P_{nh})_{\mathbf{f}} + D_{\mathbf{f}}$$

$$\text{Coarse}: \frac{\partial \underset{\sim}{\mathbf{u}}_{\mathbf{c}}}{\partial t} = -\underset{\sim}{\mathbf{v}}_{\mathbf{c}} \cdot \nabla \underset{\sim}{\mathbf{u}}_{\mathbf{c}} - 2\underset{\sim}{\Omega} \times \underset{\sim}{\mathbf{u}}_{\mathbf{c}} - \frac{1}{\rho_0}\nabla_h(P_h)_{\mathbf{c}} + D_{\mathbf{c}} + F^{SGS}_{\underset{\sim}{\mathbf{u}}}$$

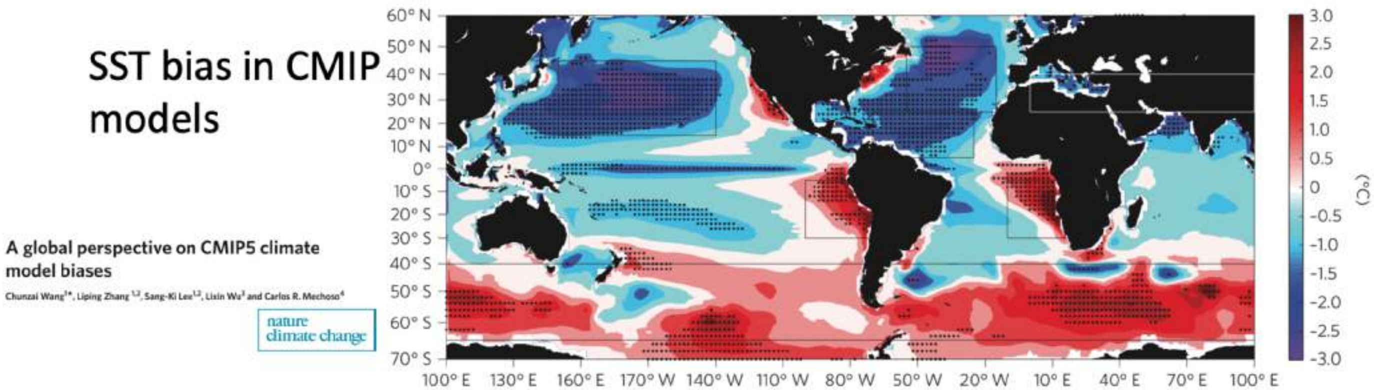
“Assimilation like” approaches to creating parametrizations

1. At this stage approaches are more rooted in Bayesian, gradient free methods
   - Emphasis on statistical fitting e.g. mean, variance etc… in algorithms rather than precise single trajectory following
   - Exploiting correlations within process studies as a cost term e.g. patterns and timescales of response to a windburst
2. Looking forward there is interest in where gradient information (through adjoint counterpart) could be leveraged.
3. It is 2019 so there is interest in various techniques from ML/AI. These include regression style networks, using unsupervised pattern detection to find structure, parsimonious bounded search for algebraic forms to create better reduced order/compressed models.

Summary

- A few new tricks for ECCO adjoints

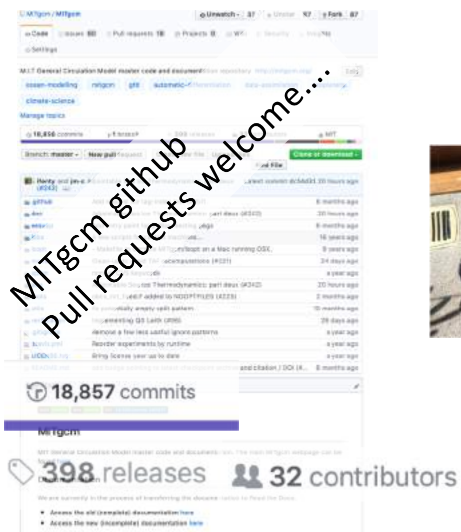
- Sizable efforts to address parametrization biases that are generally believed to be pervasive in much ocean/climate modeling



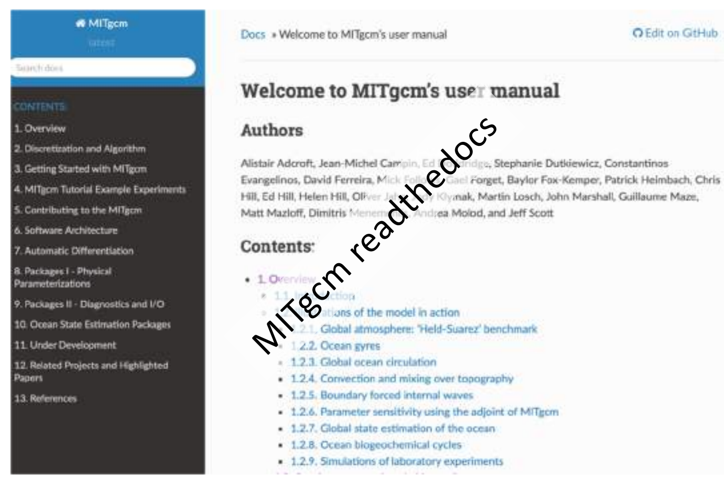
- Using ECCO solutions to provide realistic state space for parameterization development and for cost function to test new schemes.


MITgcm current web
MITgcm
M.I.T General Circulation Model News and Information
About MITgcm
Latest News and Features
High Latitude Channel Flow Revisited


MITgcm 2019 look web
MITgcm Massachusetts Institute of Technology General Circulation Model
ABOUT MITgcm
SOURCE CODE
DOCUMENTATION
LATEST NEWS
2018 Research Roundup


MITgcm github
Pull requests welcome….
18,857 commits
398 releases
32 contributors

Questions?


MITgcm readthedocs
Welcome to MITgcm's user manual
Authors
Contents: