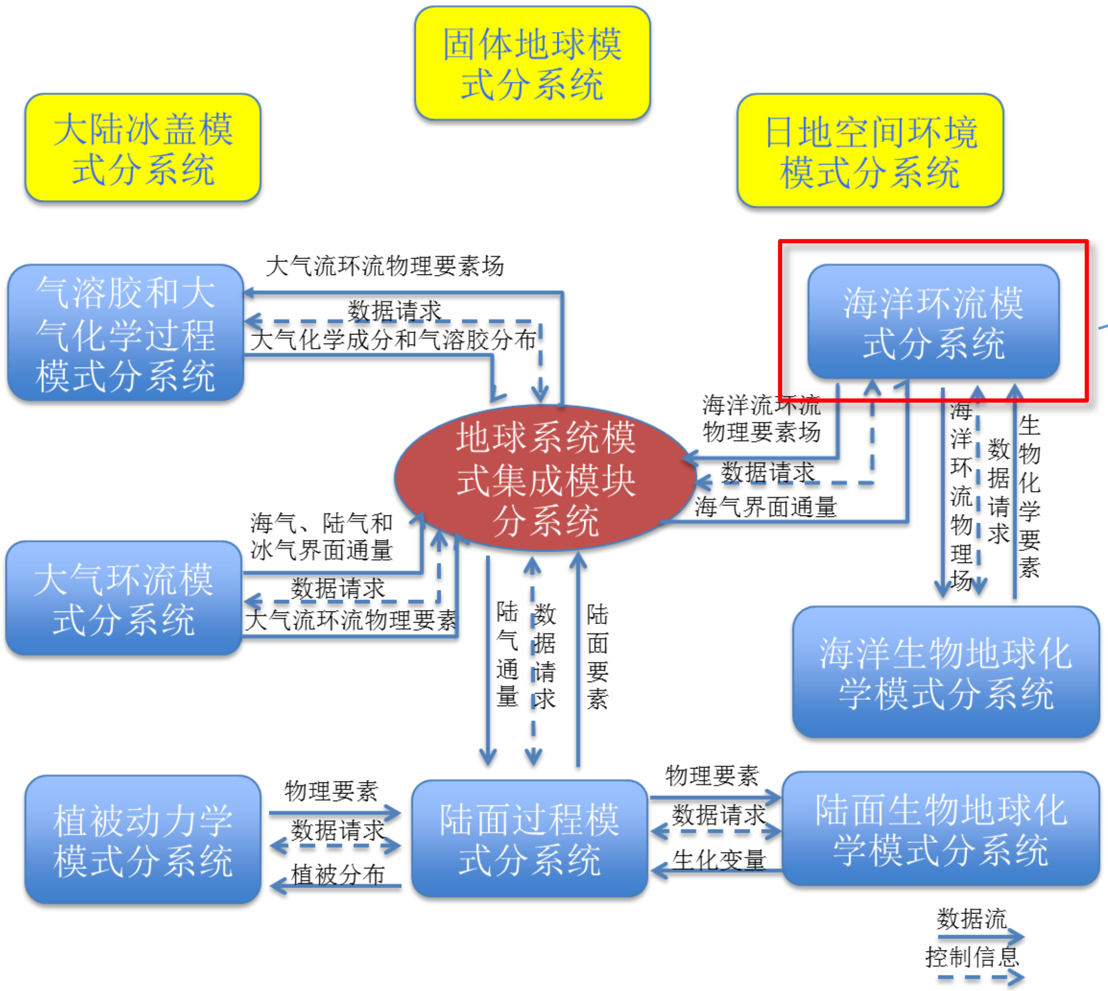
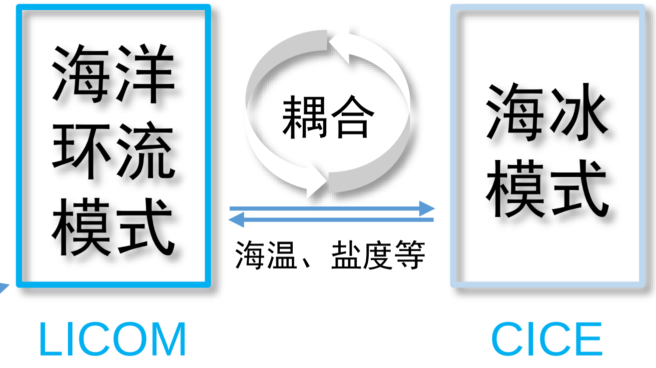
地球系统数值模拟装置项目（地球系统模式数值模拟系统）海洋环流模式分系统培训

培训人：于子棚

2022年6月1日

1.分系统介绍



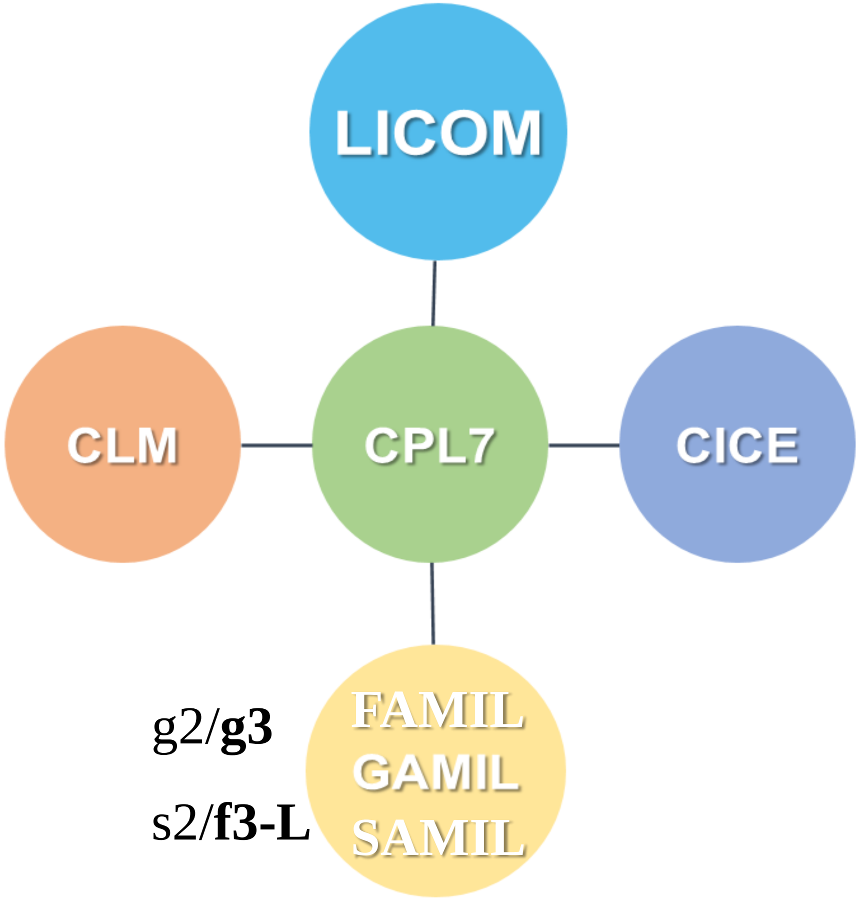


海洋环流模式分系统是海洋-海冰耦合的分系统，包括单独的海洋环流模式和海冰模式，二者通过耦合器进行通量交换，可以在给定的边界条件和初始条件下，模拟海水温度、盐度、流速、海表高度、海冰流速、海冰厚度和密集度等基本的特征，同时其模拟结果可以为海洋生物地球化学模式提供海洋环流背景场，并可以为大气模式提供下边界条件，为区域海洋模式提供侧边界条件。

1.分系统介绍

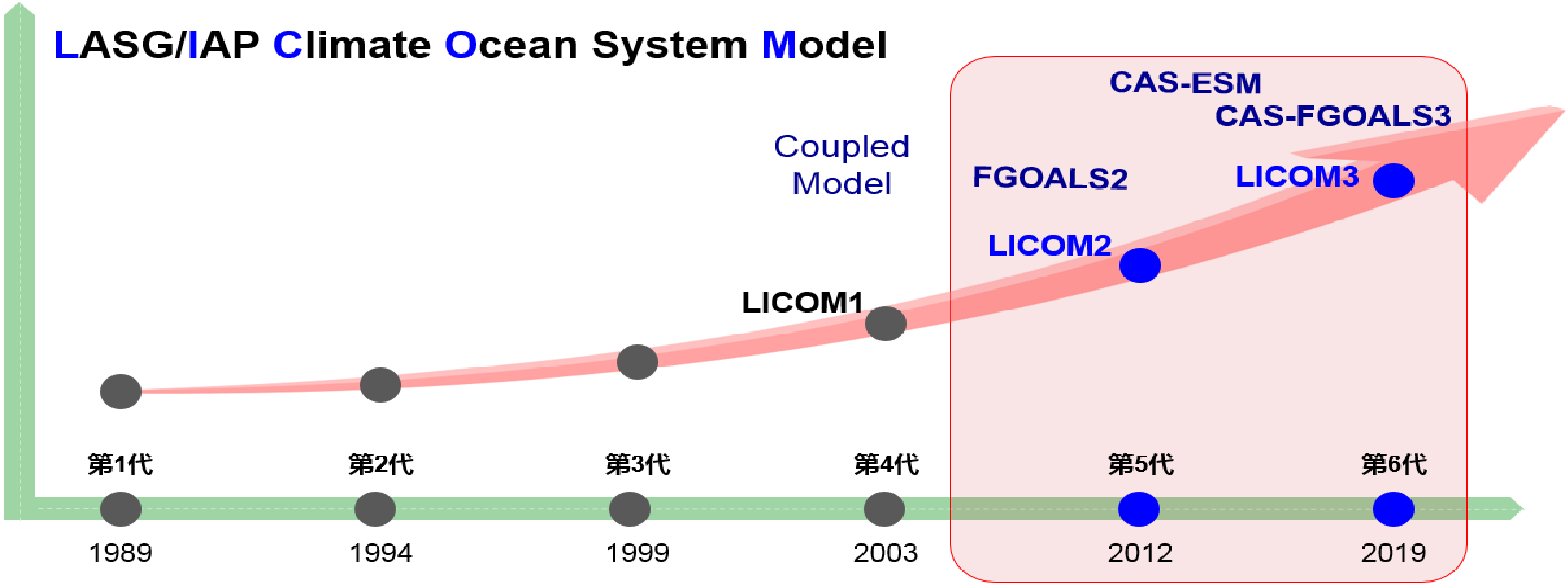
地球系统模式CAS-ESM

CAS-FGOALS



模拟数据提交国际耦合模式比较计划CMIP5，CMIP6，相关结果被IPCC报告引用。

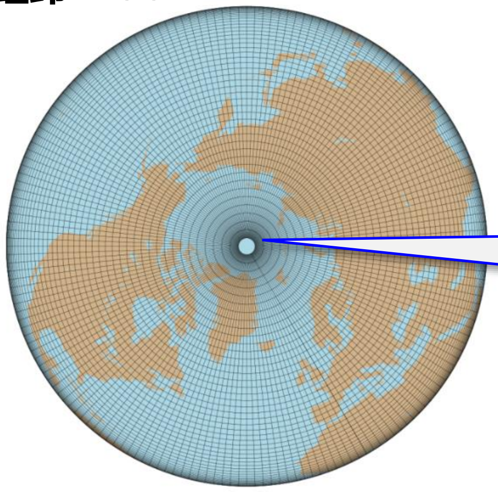
1.分系统介绍



	Resolution / Domain	References
1st	4°x5°, 4 levels ; 60°S-60°N	Zhang et al. 1989
2nd L20	4°x5°, 20 levels ; Global	Zhang et al. 1996
3rd L30T63	T63(1.875°x1.875°), 30 levels (20 in upper 1000m); Global	Jin et al. 1999
4th LICOM1.0	1° / 0.5°, 30 levels ; Global	Liu et al. 2004
5th LICOM2.0	1°/0.1°, 30/55 levels; Global/Semi-global	Liu et al. 2012; Yu et al. 2012
6th LICOM3.0	1/0.1°, 30/55/80 levels; Global	Lin et al. 2019

1.分系统介绍

经纬-LICOM2



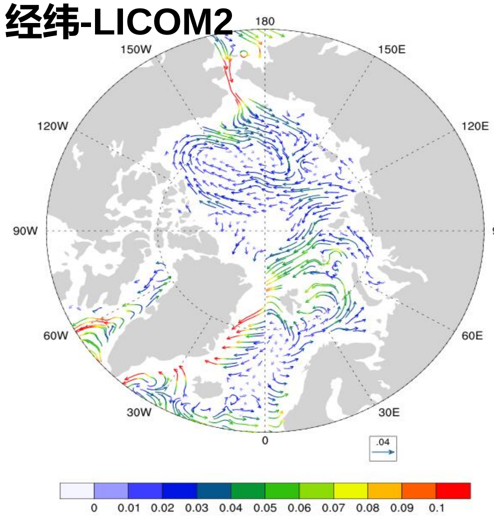
北极点，位于海上，处理成岛屿

引入适合任意水平正交的曲线坐标，采用三极网格

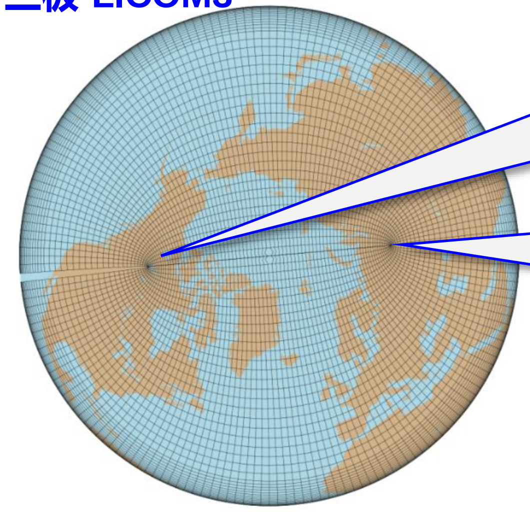
(南极+两个北极点)

模拟范围从准全球扩展到全球

经纬-LICOM2



三极-LICOM3

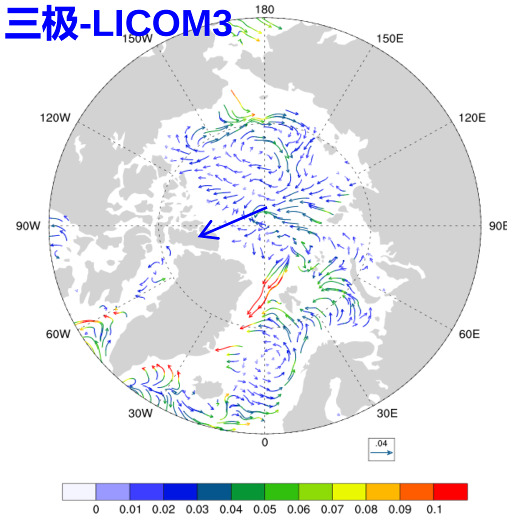


北极点，位于陆地

优点

- ✓ 消除奇点，克服北极点附近的计算不稳定
- ✓ 消除滤波，提高并行效率
- ✓ 更好模拟北冰洋物理过程，如穿极流

三极-LICOM3



1.分系统介绍

配置		LICOM2	LICOM3
接口和并行	耦合接口	第6代耦合器（CPL6）	**第7代耦合器（CPL7）**
	并行	1维 MPI+OMP	**2维**MPI + OMP
动力框架	坐标系，网格/水平，垂直	经纬正交，经纬度坐标/1º, 30层; 0.1°,55层	**任意水平正交曲线坐标，三极格点**/1º, 30/80层; 0.1°,55层
	示踪物平流	中央差	中央差/保型平流
	动量时间积分	显式	隐式
物理过程	垂直混合	2阶闭合，主针对混合层 (Canuto et al. 2001, 2002)	2阶闭合(Canuto et al. 2001, 2002) **内潮混合 (St. Laurent et al., 2002)**
	涡旋混合	等密度面混合 (Redi 1982) 涡致平流 (Gent & McWilliams 1990)	等密度面混合 (Redi 1982) 涡致平流(Gent & McWilliams 1990) **N^2 厚度扩散(Ferreira et al., 2005)**
数据	地形	DBDB5	ETOPO2
	初值	WOA01	PHC3.0

2.模式原理

模式发展是一项工程，既有海洋基本理论和原理，又有数值方法和计算机技术问题。

1. 选择坐标系，推导方程组和边界条件。
2. 确定参数化方案。
3. 确定分辨率，水平网格和垂直分层，实现区域和变量的离散化。
4. 构造时间-空间差分格式，将微分方程组化为代数方程组。
5. 完成程序设计，实现计算机模型。

2.模式原理

基于大尺度地球流体动力学的特点，**原始方程组由黏性流体力学基本方程组（即Navier-Stokes方程）简化和修改得到。**

静力平衡近似

尺度分析：海洋环流水平运动尺度远大于海洋平均深度；

垂直动量方程简化为静力平衡方程：

$$\boxed{\frac{\partial p}{\partial z} = -\rho g}$$

Boussinesq近似

压力对密度的贡献：海洋每下潜1000米(相当于增加100个大气压)，海水密度约增加5‰。

→不可压缩！

- 忽略连续方程中密度的个别变化 ；
- 动量方程压力梯度项中的密度取为常数;
- 保留状态方程和静力方程中密度的变化（浮力和水平压力梯度的来源)。

→即可压，又不可压。

湍流黏性假定

- 湍流黏性（扩散）远大于分子黏性（扩散），引入湍流黏性（扩散），略去分子黏性（扩散）。
- 方程闭合和参数化问题。

$$\frac{\partial u}{\partial t} = -\frac{\partial uu}{\partial x} - \frac{\partial vu}{\partial y} - \frac{\partial wu}{\partial z} \boxed{-\frac{\partial \langle u'u' \rangle}{\partial x} - \frac{\partial \langle v'u' \rangle}{\partial y} - \frac{\partial \langle w'u' \rangle}{\partial z}} \ldots$$

2.模式原理

原始方程组

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} - w\frac{\partial u}{\partial z} + fv - \frac{1}{\rho_o}\frac{\partial p}{\partial x} + F_u^x + F_u^y + F_u^z$$

$$\frac{\partial v}{\partial t} = -u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} - w\frac{\partial v}{\partial z} - fu - \frac{1}{\rho_o}\frac{\partial p}{\partial y} + F_v^x + F_v^y + F_v^z$$

水平动量方程

$$\frac{\partial T}{\partial t} = -u\frac{\partial T}{\partial x} - v\frac{\partial T}{\partial y} - w\frac{\partial T}{\partial z} + F_T^x + F_T^y + F_T^z + \frac{1}{\rho_0 c_p}\frac{\partial I}{\partial z}$$

$$\frac{\partial S}{\partial t} = -u\frac{\partial S}{\partial x} - v\frac{\partial S}{\partial y} - w\frac{\partial S}{\partial z} + F_S^x + F_S^y + F_S^z$$

温、盐方程

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

连续方程

$$\frac{\partial p}{\partial z} = -\rho g$$

静力方程

$$\rho = \rho(T, S, z)$$

状态方程

2.模式原理

动力子系统主要负责求解海洋和海冰模式方程组的数值解，方程包扩：考虑Boussinesq近似的动量方程，位温和盐度两种示踪物的守恒方程，表示密度、盐度、温度和压力之间非线性关系的状态方程、海表高度方程，以及海冰的热力学和动力学方程。

- 动力子系统
  - 正压模块
  - 斜压模块
  - 正斜压耦合模块
  - 垂直速度计算模块
  - 温盐模块
  - 温盐平流模块
  - 动量平流模块
  - 密度计算模块
  - 瞬时和平均输出模块
  - 经圈流函数诊断模块
  - 经向热输送诊断模块
  - 正压流函数诊断模块
  - 全球平均能量诊断模块
  - 垂直热动力模块
  - 热力变化模块
  - 动力传输成脊模块

2.模式原理

正压模块

对连续方程从海底到海表作垂直积分，并交换微分和积分次序：

$$\int_{-H}^{z_0}\left(\nabla\cdot\boldsymbol{v}+\frac{\partial w}{\partial z}\right)dz=0$$

$$w\Big|_{z=-H}=-\left(\mathbf{v}\cdot\nabla H\right)_{z=-H}$$

海底边界条件

$$\frac{\partial z_0}{\partial t}=-\left(\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}\right)$$

海表高度预报方程

$$\left(U,V\right)\equiv\int_{-H}^{z_0}\left(u,v\right)dz$$

正压输送

海表高度的倾向就取决于垂直积分流的散度。

```
      do iblock = 1, nblocks_clinic
         this_block = GET_BLOCK(blocks_clinic(iblock),iblock)
         call DIV(1,div_out,wka(:,:,1,iblock),wka(:,:,2,iblock),this_block)
         do j = 3, jmt-2
            do i = 3,imt-2
               work(i,j,iblock) = vit(i,j,1,iblock)*(-1)*div_out(i,j)*p25
            end do
         enddo
      end do

      call POP_HALOUPDATE(work,POP_haloClinic,POP_gridHorzLocCenter,&
                          POP_fieldKindScalar,errorCode,fillValue = 0.0_r8)

      !!$OMP PARALLEL DO PRIVATE(IBLOCK,J,I)
      do iblock = 1, nblocks_clinic
         do j = 1, jmt
            do i = 1,imt
               h0(i,j,iblock)= h0p(i,j,iblock) + work(i,j,iblock)*dtb
            end do
         end do
      end do

      !-----------------------------------------------------------------
      !  COMPUTE THE "ARTIFICIAL" HORIZONTAL VISCOSITY
      !-----------------------------------------------------------------
#if( defined SMAG1)

#else
#if(defined BIHAR)
      !!$OMP PARALLEL DO PRIVATE(IBLOCK,this_block, hduk,hdvk)
      do iblock = 1, nblocks_clinic
         this_block = GET_BLOCK(blocks_clinic(iblock),iblock)
         call HDIFFU_DEL4(1, hduk, hdvk, ubp(:,:,iblock), &
                          vbp(:,:,iblock), this_block)
```

```
   ! compute divergence using a 4 point stencil
   !-----------------------------------------------------------------
   bid = this_block%local_id

   div_out = c0
   do j = 2,ny_block
      do i = 1,nx_block-1
         if (k <= kmt(i,j,bid)) then
            div_out(i,j) = &
               p5*((ux(i+1,j  ) + ux(i+1,j-1))*htw(i+1,j,bid) -    &
                   (ux(i  ,j  ) + ux(i  ,j-1))*htw(i  ,j,bid) +    &
                   (uy(i+1,j  ) + uy(i  ,j  ))*hts(i  ,j,bid) -    &
                   (uy(i+1,j-1) + uy(i  ,j-1))*hts(i  ,j-1,bid))*  &
                   tarea_r(i,j,bid)
         endif
      end do
   end do
end subroutine DIV
```

2.模式原理

正斜压耦合模块

对水平动量方程做垂直积分，可得：

快过程　慢过程

$$\frac{\partial U}{\partial t}=\boxed{-g(H+z_0)\frac{\partial z_0}{\partial x}+fV}+\boxed{\frac{1}{\rho_0}(\tau_x-\tau_{b,x})}-\boxed{\frac{1}{\rho_0}\int_{-H}^{z_0}\frac{\partial p_C}{\partial x}dz}+\boxed{X}$$

$$\frac{\partial V}{\partial t}=\boxed{-g(H+z_0)\frac{\partial z_0}{\partial y}-fU}+\boxed{\frac{1}{\rho_0}(\tau_y-\tau_{b,y})}-\boxed{\frac{1}{\rho_0}\int_{-H}^{z_0}\frac{\partial p_C}{\partial y}dz}+\boxed{Y}$$

正压压力梯度项　科氏力项　源汇项　斜压压力梯度项的垂直积分　水平平流项和水平黏性项对正压输送的贡献

$$X\equiv-\frac{\partial}{\partial x}\int_{-H}^{z_0}uudz-\frac{\partial}{\partial y}\int_{-H}^{z_0}vudz+\int_{-H}^{z_0}\left(F_u^x+F_u^y\right)dz$$

$$Y\equiv-\frac{\partial}{\partial x}\int_{-H}^{z_0}uvdz-\frac{\partial}{\partial y}\int_{-H}^{z_0}vvdz+\int_{-H}^{z_0}\left(F_v^x+F_v^y\right)dz$$

$$\begin{cases}\dfrac{\partial p}{\partial z}=-\rho g\\ p\big|_{z=0}=\rho_0 g z_0\end{cases}$$

$$p=\rho_0 g z_0+\int_z^0\rho g\,dz\equiv\rho_0 g z_0+p_C$$

代表压力的斜压部分，只与密度有关。

压力梯度项

$$-\frac{1}{\rho_0}\frac{\partial p}{\partial x}=-g\frac{\partial z_0}{\partial x}-\frac{1}{\rho_0}\frac{\partial p_C}{\partial x}$$

$$-\frac{1}{\rho_0}\frac{\partial p}{\partial y}=-g\frac{\partial z_0}{\partial y}-\frac{1}{\rho_0}\frac{\partial p_C}{\partial y}$$

Earth Lab

2.模式原理

正斜压耦合模块

模态分解方法（Blumberg and Mellor，1987）

假定当前时刻所有的模式变量值均已得到，考虑如何由当前时刻的变量值计算出下一时刻的变量值。由于我们的主要目的是将表面波分离出来，故以下只讨论动量方程从当前时刻到下一时刻的积分（其间也要涉及连续方程）。为叙述简单起见，以下提到的“方程”均指和微分方程相应的差分方程。

第一步，从当前时刻出发，用时间步长Δt_C将完全的动量方程（4.1）和（4.2）积分一步，得到下一时刻u，v的预估值u^*，v^*。

第二步，由于动量方程（4.1）和（4.2）中既包含斜压分量也包含正压分量，而积分的时间步长Δt_C远大于正压过程所能允许的时间步长Δt_B，所以预估值u^*，v^*的误差应当主要来源于其中的正压部分。为避免正压部分误差的积累，将u^*，v^*中的正压分量扣除，只保留斜压分量u'，v'：

$$u' = u^* - \bar{u}$$

$$v' = v^* - \bar{v}$$

其中

$$\bar{u} = \frac{1}{H+z_0}\int_{-H}^{z_0} u^*\,dz$$

$$\bar{v} = \frac{1}{H+z_0}\int_{-H}^{z_0} v^*\,dz$$

$$\frac{\partial z_0}{\partial t} = -\left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y}\right)$$

$$\frac{\partial U}{\partial t} = -g(H+z_0)\frac{\partial z_0}{\partial x} + fV + \frac{1}{\rho_0}(\tau_x - \tau_{b,x}) - \frac{1}{\rho_0}\int_{-H}^{z_0}\frac{\partial p_C}{\partial x}dz + X$$

$$\frac{\partial V}{\partial t} = -g(H+z_0)\frac{\partial z_0}{\partial y} - fU + \frac{1}{\rho_0}(\tau_y - \tau_{b,y}) - \frac{1}{\rho_0}\int_{-H}^{z_0}\frac{\partial p_C}{\partial y}dz + Y$$

第三步，为了得到正压分量的比较精确的估计，回到当前时刻，用表面波过程所允许的时间步长Δt_B将（4.20）、（4.25）和（4.26）构成的正压模态方程组积分N步（总的时间跨度为Δt_C），得到下一时刻的海表高度z_0以及正压输送量U，V，由U，V可以计算正压速度分量，用于和第二步得到的斜压速度组成下一时刻的速度：

$$u = u' + \frac{U}{H+z_0}$$

$$v = v' + \frac{V}{H+z_0} \qquad (4.54)$$

和 u^*，v^* 相比，u，v 中的正压分量的误差得到了控制，因此这种算法有利于在长时期积分过程中保持计算稳定性。

```
!-----------------------------------------------------------------------
! INTERACTION BETWEEN BAROTROPIC AND BAROCLINIC MODES
!-----------------------------------------------------------------------
call VINTEG(U,WORK)

!!$OMP PARALLEL DO PRIVATE(IBLOCK,K,J,I)
do iblock = 1, nblocks_clinic
   do k = 1,km
      do j = 1, jmt
         do i = 1,imt
            u(i,j,k,iblock) = (u(i,j,k,iblock) - work(i,j,iblock) + &
                               ub(i,j,iblock))*viv(i,j,k,iblock)
         end do
      end do
   end do
end do

call VINTEG(V,WORK)

!!$OMP PARALLEL DO PRIVATE(IBLOCK,K,J,I)
do iblock = 1, nblocks_clinic
   do k = 1,km
      do j = 1, jmt
         do i = 1,imt
            v(i,j,k,iblock) = (v(i,j,k,iblock) - work(i,j,iblock) + &
                               vb(i,j,iblock))*viv(i,j,k,iblock)
         end do
      end do
   end do
end do
```

```
            ! now invert
            do k = kz,1, -1
               g0       = 1.0/(b8(k) - c8(k)*e8(k))
               e8(k-1)  = a8(k)*g0
               f8(k-1)  = (d8(k) + c8(k)*f8(k))*g0
            end do

            ! b.c. at surface
            wk(i,j,1,iblock) = (e8(0)*topbc(i,j,iblock) + &
                                f8(0))*viv(i,j,1,iblock)
            do k = 2,kz
               wk(i,j,k,iblock) = (e8(k-1)*wk(i,j,k-1,iblock) + &
                                   f8(k-1))*viv(i,j,k,iblock)
            end do
         end if   ! if(kmu(i,j,iblock) > 0) then
      end do   ! do i = 3, imt-2
   end do   ! do j = 3, jmt-2
end do   ! do iblock = 1, nblocks_clinic

return
end subroutine INVTRIU
```

2.模式原理

动量平流模块

“两点格式”

整点

$$(\Delta_x F)_{i+\frac{1}{2}} \equiv F_{i+1} - F_i \,, i = 1,2,\ldots,I-1$$

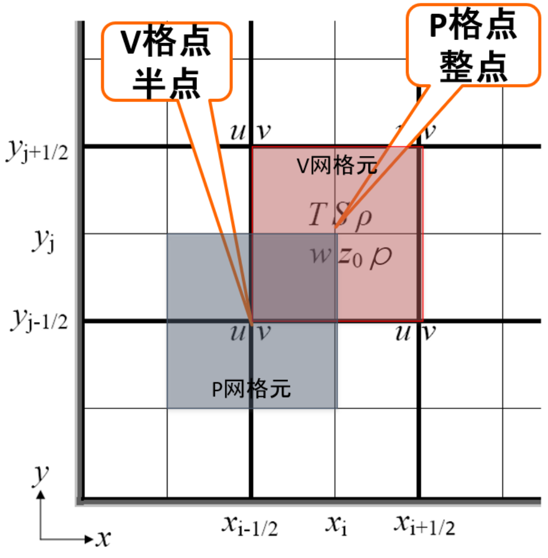
$$(\bar{F}^x)_{i+\frac{1}{2}} \equiv \frac{F_{i+1} + F_i}{2} \,, i = 1,2,\ldots,I-1$$

半点

$$(\Delta_x G)_i \equiv G_{i+\frac{1}{2}} - G_{i-\frac{1}{2}} \,, i = 1,2,\ldots,I$$

$$(\bar{G}^x)_i \equiv \frac{G_{i+\frac{1}{2}} + G_{i-\frac{1}{2}}}{2} \,, i = 1,2,\ldots,I$$

水平网格（B-grid）和变量的安排



以u-动量方程为例：

$$\left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \approx \left[\frac{1}{\Delta x}\left(\overline{\overline{u}^x \Delta_x u}^x\right) + \frac{1}{\Delta y}\left(\overline{\overline{u}^y \Delta_y u}^y\right)\right]_{i+\frac{1}{2},j+\frac{1}{2}}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \approx \frac{1}{\Delta x^2}\left[\Delta_x(\Delta_x u)\right]_{i+\frac{1}{2},j+\frac{1}{2}}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \approx \frac{1}{\Delta y^2}\left[\Delta_y(\Delta_y u)\right]_{i+\frac{1}{2},j+\frac{1}{2}}$$

```fortran
         do k = 1, km
            do j = 1,jmt-1
               do i = 2,imt
                  u_wface(i,j,k) = (uuu(i-1,j,k) + &
                                    uuu(i,  j,k))*P25*hue(i-1,j,  iblock)
                  v_sface(i,j,k) = (vvv(i,j,  k) + &
                                    vvv(i,j+1,k))*P25*hun(i,  j+1,iblock)
               end do
            end do
         end do

      else if ( adv_momentum(1:4) == 'flux' ) then

         do k = 1, km
            do j = 1,jmt-1
               do i = 2,imt
                  u_wface(i,j,k) = (uuu(i-1,j,k)*dyu(i-1,j,iblock) + &
                                    uuu(i,  j,k)*dyu(i,  j,iblock))*P25
                  v_sface(i,j,k) = (vvv(i,j,  k)*dxu(i,j,  iblock) + &
                                    vvv(i,j+1,k)*dxu(i,j+1,iblock))*P25
               end do
            end do
         end do
      else
         call EXIT_LICOM(sigAbort,'The false advection option for tracer')
      end if   ! if ( adv_momentum(1:8) == 'centered' ) then
```

```fortran
                  if(k == 1 ) then
                      adv_z1 = 0.0D0
                      adv_z3 = 0.0D0
                  else
                      adv_z1 = www(i,j,k)*(uuu(i,j,k-1) + uuu(i,j,k))*p5
                      adv_z3 = www(i,j,k)*(vvv(i,j,k-1) + vvv(i,j,k))*p5
                  end if

                  if(k == km ) then
                       adv_z2 = 0.0D0
                       adv_z4 = 0.0D0
                  else
                       adv_z2 = www(i,j,k+1)*(uuu(i,j,k ) + uuu(i,j,k+1))*p5
                       adv_z4 = www(i,j,k+1)*(vvv(i,j,k ) + vvv(i,j,k+1))*p5
                  end if
                  adv_uu(i,j,k) = adv_uu(i,j,k) - odzp(k)*(adv_z2-adv_z1)
                  adv_vv(i,j,k) = adv_vv(i,j,k) - odzp(k)*(adv_z4-adv_z3)
               end do   ! do i = 3,imt-2
            end do   ! do j = 3,jmt-2
         end do   ! do k = 1,km
      else
         write(6,*) "adv_momentum =", adv_momentum
         call EXIT_LICOM(sigAbort,'The false advection option for momentum')
      end if   ! if( adv_momentum(1:8) == 'centered' ) then

   end subroutine ADVECTION_MOMENTUM
```

2.模式原理

垂直速度计算模块

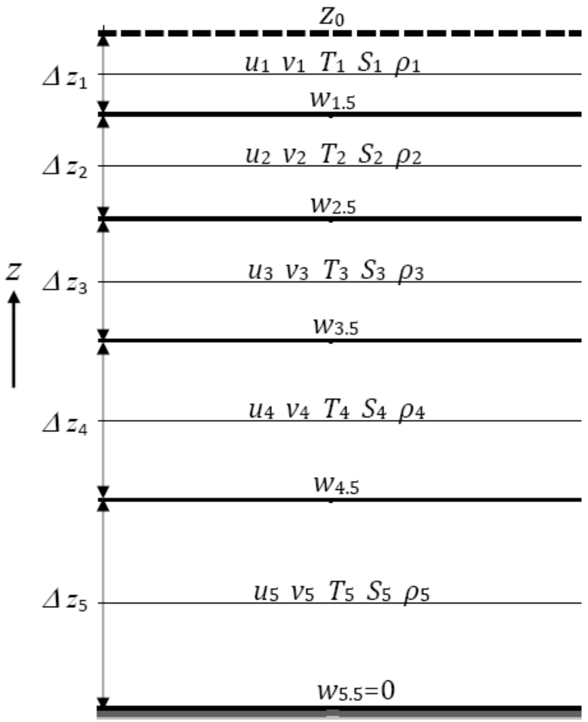
z方向上“两点格式”

$$\frac{1}{\Delta z}(\Delta_z w)_k = \frac{w_{k-\frac{1}{2}} - w_{k+\frac{1}{2}}}{\Delta z} = -D_k$$

$$2 \le k \le K$$

$$w_{K+\frac{1}{2}} = 0$$

其中， D_k表示该层的水平散度。




```fortran
     !!$OMP PARALLEL DO PRIVATE(IBLOCK,J,I)
     do iblock = 1, nblocks_clinic
        do j = 2, jmt-1
           do i = 2, imt-1
              work(i,j,iblock) = 1.0d0/(1.0d0 + h0wk(i,j,iblock)* &
                                 ohbt(i,j,iblock))
           end do
        end do
     end do

     !!$OMP PARALLEL DO PRIVATE(IBLOCK,K,J,I)
     do iblock = 1, nblocks_clinic
        do k = 2,km
           do j = 2, jmt-1
              do i = 2,imt-1
                 ws(i,j,k,iblock) = ws(i,j,k,iblock)*work(i,j,iblock)
              end do
           end do
        end do
     end do

     deallocate(uk,vk)
     return

  end subroutine UPWELL
```

$$D_{i,j} \equiv \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i,j} \approx \frac{1}{\Delta x}\left[\frac{u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}}{2} - \frac{u_{i-\frac{1}{2},j+\frac{1}{2}} + u_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\right] + \frac{1}{\Delta y}\left[\frac{v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{v_{i+\frac{1}{2},j-\frac{1}{2}} + v_{i-\frac{1}{2},j-\frac{1}{2}}}{2}\right]$$

在模式表层(k=1)，

$$w_{i,\frac{1}{2}} = \left(\frac{\partial z_0}{\partial t}\right)_i + \left[\overline{u\Delta_x z_0}^{x}\right]_{i,1}$$

$$\left(\frac{\partial z_0}{\partial t}\right)_i = -\frac{1}{\Delta x}\Delta_x\left\{\sum_{k=K}^{1} u \times \Delta z\right\}_{i,k} - \frac{1}{\Delta x}\Delta_x\left\{\sum_{k=K}^{1} u \times \overline{\Delta z_0}^{x}\right\}_{i,1}$$

2.模式原理

温度平流模块

通常情况下，一个单纯经历平流的标量场，在平流的过程中，其值的变化范围会在初值的取值范围内。根据这个规则，在结合 Lax-Wendroff 二阶精度差分方案和迎风差方案的基础上，提出了两步正定保形平流方案。

具体计算：

在不考虑扩散的情况下，示踪物方程的通量形式可以写成如下形式：

$$\frac{\partial F}{\partial t}+\nabla\cdot(\vec{V}F)=0$$

其中，$F=F(x,y,z,t)$为无扩散的示踪物，$V=V$（u,v,w）为三维速度场。

为了简单起见，考虑其在一维的情况，介绍两步正定保形平流方案的具体计算方法，多维的情况可以简单推广。对于一维的情况，将上式可以写成：

$$\frac{\partial F}{\partial t}+\frac{\partial uF}{\partial x}=0$$

1：运用 Lax-Wendroff 格式进行预积分。设第 n 时刻的值为 F_i^n，利用 Lax-Wendroff 格式预积分一步后得到的值为 F_i^*。即：

$$F_i^* = F_i^n - \frac{\Delta t}{2\Delta x}[u^n_{i+\frac{1}{2}}(F^n_{i+1}+F_i^n)-u^n_{i-\frac{1}{2}}(F_i^n+F^n_{i-1})]$$
$$+\frac{\Delta t}{2\Delta x}[|u^2|^n_{i+\frac{1}{2}}\frac{\Delta t}{\Delta x}(F^n_{i+1}-F_i^n)-|u^2|^n_{i-\frac{1}{2}}\frac{\Delta t}{\Delta x}(F_i^n-F^n_{i-1})]$$

然后，对预积分得到的 F_i^*用保形传输规则进行判断，判断是否满足：

$$F^n_{i\,\min} \le F_i^* \le F^n_{i\,\max}$$

其中：

$$F^n_{i\,\max} = \max(F^n_{i-1},F_i^n,F^n_{i+1})$$
$$F^n_{i\,\min} = \min(F^n_{i-1},F_i^n,F^n_{i+1})$$

如果满足该判定条件，则表示满足保形传输规则，否则表示破坏了保形传输规则。

A）如果不满足保形传输规则，则对该点采用迎风差格式，即：

$$F_i^{n+1} = F_i^n - \frac{\Delta t}{2\Delta x}[u^n_{i+\frac{1}{2}}(F^n_{i+1}+F_i^n)-u^n_{i-\frac{1}{2}}(F_i^n+F^n_{i-1})]$$
$$+\frac{\Delta t}{2\Delta x}[|u|^n_{i+\frac{1}{2}}(F^n_{i+1}-F_i^n)-|u|^n_{i-\frac{1}{2}}(F_i^n-F^n_{i-1})]$$

B）如果满足保形传输规则，则利用 F_i^*对 Lax-Wendroff 格式作校正积分，校正方法如下：

$$F_i^{n+1} = F_i^n - \frac{\Delta t}{2\Delta x}[u^n_{i+\frac{1}{2}}(F^n_{i+1}+F_i^n)-u^n_{i-\frac{1}{2}}(F_i^n+F^n_{i-1})]$$
$$+\frac{\Delta t}{2\Delta x}[|\tilde{u}|^n_{i+\frac{1}{2}}(F^n_{i+1}-F_i^n)-|\tilde{u}|^n_{i-\frac{1}{2}}(F_i^n-F^n_{i-1})]$$

令$A_i = (F_i^* - F^n_{i\,max})(F_i^* - F^n_{i\,min})$

在满足保形规则（即 $A_i<0$）的情况下，$|\tilde{u}|^n_{i+\frac{1}{2}}$和$|\tilde{u}|^n_{i-\frac{1}{2}}$的取值有以下四种情况：

当 $A_{i+1}>0$，$A_{i-1}>0$ 时：$|\tilde{u}|^n_{i+\frac{1}{2}}=|u|^n_{i+\frac{1}{2}}$，$|\tilde{u}|^n_{i-\frac{1}{2}}=|u|^n_{i-\frac{1}{2}}$

当 $A_{i+1}>0$，$A_{i-1}<0$ 时：$|\tilde{u}|^n_{i+\frac{1}{2}}=|u|^n_{i+\frac{1}{2}}$，$|\tilde{u}|^n_{i-\frac{1}{2}}=|u^2|^n_{i-\frac{1}{2}}\frac{\Delta t}{\Delta x}$

当 $A_{i+1}<0$，$A_{i-1}>0$ 时：，$|\tilde{u}|^n_{i+\frac{1}{2}}=|u^2|^n_{i+\frac{1}{2}}\frac{\Delta t}{\Delta x}$，$|\tilde{u}|^n_{i-\frac{1}{2}}=|u|^n_{i-\frac{1}{2}}$

当 $A_{i+1}<0$，$A_{i-1}<0$ 时：，$|\tilde{u}|^n_{i+\frac{1}{2}}=|u^2|^n_{i+\frac{1}{2}}\frac{\Delta t}{\Delta x}$，$|\tilde{u}|^n_{i-\frac{1}{2}}=|u^2|^n_{i-\frac{1}{2}}\frac{\Delta t}{\Delta x}$

```
        do k = 1, km
          do j = 1, jmt
            do i = 2, imt-1
              if( adv_tracer(1:8) == 'centered' .or. &
                  adv_tracer(1:5) == 'tspas') then
                v_sface(i,j,k) = (vvv(i,  j,k) + &
                            vvv(i+1,j,k))*hts(i,j,iblock)*P25
              else if( adv_tracer(1:4) == 'flux' ) then
                v_sface(i,j,k) = (vvv(i,  j,k)*dxu(i,  j,iblock) + &
                            vvv(i+1,j,k)*dxu(i+1,j,iblock))*P25

              end if
            end do
          end do

          do j = 2,jmt-1
            do i = 1,imt
              if( adv_tracer(1:8) == 'centered' .or. &
                  adv_tracer(1:5) == 'tspas') then
                u_wface(i,j,k) = (uuu(i,j-1,k) + &
                            uuu(i,j,  k))*htw(i,j,iblock)*P25
              else if( adv_tracer(1:4) == 'flux' ) then
                u_wface(i,j,k) = (uuu(i,j-1,k)*dyu(i,j-1,iblock) + &
                            uuu(i,j,  k)*dyu(i,j,iblock))*P25
              end if
            end do
          end do
```

```
          adv_xx = -(adv_x0 + adv_xy1 + adv_xy2 + adv_c1)
          adv_yy = -(adv_y0 + adv_xy3 + adv_xy4 + adv_c2)
          adv_zz = -(adv_za + adv_zb1 + adv_zb2 + adv_zc)
          adv_tt =    adv_xx + adv_yy  + adv_zz

          ax(:,:,:,mtracer,iblock) = adv_xx(:,:,:)
          ay(:,:,:,mtracer,iblock) = adv_yy(:,:,:)
          az(:,:,:,mtracer,iblock) = adv_zz(:,:,:)

          deallocate( adv_c1,adv_c2,adv_zc)
          deallocate( adv_x0, adv_y0, adv_xx, adv_yy)
          deallocate( adv_xy1, adv_xy2, adv_xy3, adv_xy4)
          deallocate( at00, atmax, atmin )
          deallocate( adv_zz, adv_za, adv_zb1 )
          deallocate( adv_zb2, atmaxz, atminz, atz)

        else
          call EXIT_LICOM(sigAbort,'The false advection option for tracer')
        end if   ! if( adv_tracer(1:8) == 'centered' ) then

      end subroutine ADVECTION_TRACER
```

密度计算模块

模式中位密度计算采用的是一个三次多项式拟合的UNESCO（United Nations Educational, Scientific and Cultural Organization）公式，扣除所在深度的参考层结，只计算密度对参考位温和参考盐度的扰动量（Bryan和Cox，1972；UNESCO，1981）。计算公式为：

$$\delta\rho = c_1\delta T + c_2\delta S$$

$$+ c_3\delta T^2 + c_4\delta T\delta S + c_5\delta S^2$$

$$+ c_6\delta T^3 + c_7\delta T\delta S^2 + c_8\delta T^2\delta S + c_9\delta S^3$$

$$\delta\rho \equiv \rho - \rho_{\mathrm{r}},\ \ \delta T \equiv T - T_{\mathrm{r}},\ \ \delta S \equiv S - S_{\mathrm{r}}$$

ρ_r，T_r和S_r分别表示参考密度、参考位温和参考盐度。ρ、T和S的单位分别为kg m^{-3}、^{o}C和（psu－35）/1000。特别需要说明的是，方程中S定义为（Sp－35）/1000。九个系数考虑了压力对密度的影响，故而是随深度而变化的。

```
subroutine DENSITY


   !LOCAL VARIABLES
   integer    :: iblock
   real(r8)   :: tq,sq

   ! ...........................................................................
   ! Start of code body
   ! ...........................................................................

   !!$OMP PARALLEL do PRIVATE(IBLOCK,K,J,I,TQ,SQ)
   do iblock = 1, nblocks_clinic
      do k = 1,km
         do j = jst,jet
            do i = 1,imt
               if(vit(i,j,k,iblock) > 0.0) then
                  tq = atb(i,j,k,1,iblock) - to(k)
                  sq = atb(i,j,k,2,iblock) - so(k)

                  pdensity(i,j,k,iblock) = 1.0d+3 + po(k)+ (c(k,1) + &
                                          (c(k,4) + c(k,7)*sq)*sq + &
                                          (c(k,3) + c(k,8)*sq+c(k,6)*tq)* &
                                          tq)*tq + (c(k,2) + (c(k,5) + &
                                          c(k,9)*sq)*sq)*sq
               else
                  pdensity(i,j,k,iblock) = 0.0
               end IF
            end do
         end do
      end do
   end do

   return

end subroutine DENSITY
```

```
function DENS(tq,sq,kk)


    ! INPUT PARAMETERS:
    real(r8),intent(in)   :: tq    !< @param tq
    real(r8),intent(in)   :: sq    !< @param sq
    integer, intent(in)   :: kk    !< @param kk

    ! LOCAL PARAMETER
    real(r8) :: dens ! dens


    ! ...........................................................................
    ! Start of code body
    ! ...........................................................................

    dens = (c(kk,1) + (c(kk,4) + c(kk,7)*sq)*sq +          &
           (c(kk,3) +  c(kk,8)*sq + c(kk,6)*tq)*tq)*tq + &
           (c(kk,2) + (c(kk,5) + c(kk,9)*sq)*sq)*sq

    return
end function DENS
```

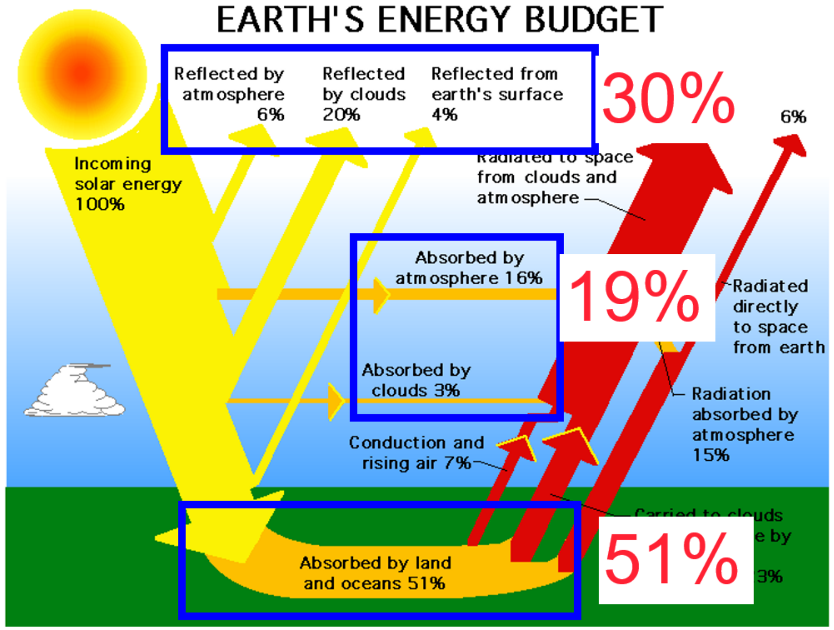
2.模式原理

参数化子系统负责处理海洋和海冰中的物理过程，主要分两类：一类是使用简单的近似方程表述复杂的海洋物理过程、海冰特征状态和复杂物理过程，如太阳短波辐射穿透过程，融池分布、海冰的强度、含盐胞的海冰比热、冰气-冰海间的扰动通量和应力、短波辐射反照和传输过程；另一类是次网格过程，即模式网格无法分辨的过程，主要包括海表湍流通量、水平粘性、对流调整方案、沿等密度面（或中性面）的混合以及穿越等密度面的混合等。

- 参数化子系统
  - 短波穿透模块
  - 中尺度涡参数化方案模块
  - 水平粘性（扩散）模块
  - Canuto湍流混合模块
  - 潮汐（内波）混合模块
  - 底边界层模块
  - 边界条件模块
  - 对流混合模块
  - 冰点温度模块

2.模式原理

短波辐射穿透模块



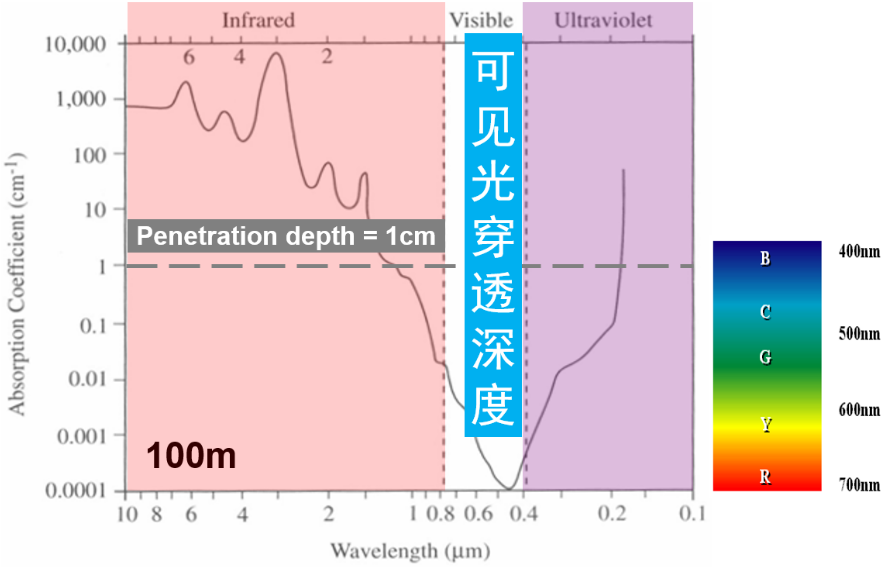
$$
\frac{\partial T}{\partial t} = -(u+u^*)\frac{\partial T}{\partial x} - (v+v^*)\frac{\partial T}{\partial y} - (w+w^*)\frac{\partial T}{\partial z}
$$

$$
+ K_h \Delta T + \frac{\partial}{\partial z}\left(K_v \frac{\partial T}{\partial z}\right) + R(K_I, T)
$$

$$
+ \frac{1}{\rho C_p}\frac{\partial I_0}{\partial z}
$$

$$
+ C_T
$$

太阳短波辐射穿透对海洋的加热，I_0是短波辐射在海洋中的分布廓线。



海洋的典型吸收系数 (取自 Thomas et al., 1999)

短波穿透影响：

- 上层海洋层结；
- 混合过程；
- 海表温度，进而影响海气通量交换以及海洋和大气环流。

Jerlov（1968）方案

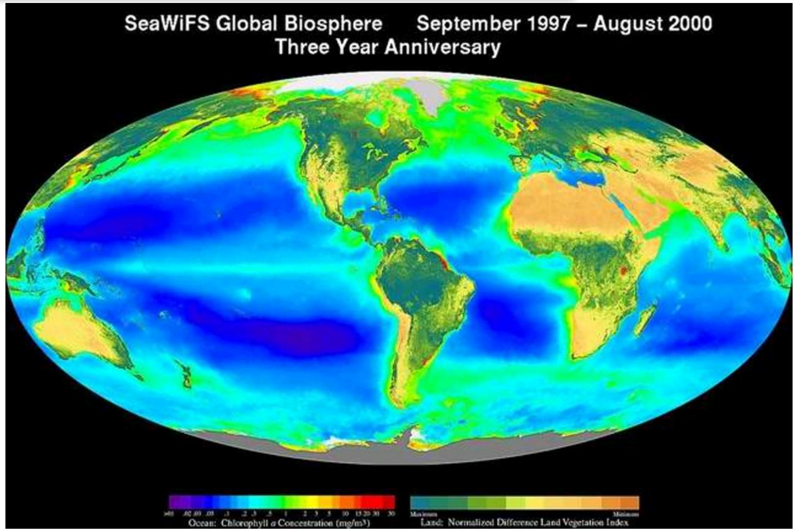
使用双 e 指数表示短波辐射随深度的变化，公式如下：

$$
I_0 = S_{w0}\left[A_1 e^{\frac{-z}{B_1}} + A_2 e^{\frac{-z}{B_2}}\right]
$$

其中，取 $A_1=0.58$，$A_2=1-A_1=0.42$，穿透深度 $B_1=0.35$m，$B_2=23.0$m。这意味着 58%的能量在 0.35m 的深度上呈 e 指数衰减，42%的能量在 23.0m 的深度内呈 e 指数衰减。公式第一部分表示的主要是红外频段的能量，而第二部分主要是可见光和紫外这些波长相对较短的频段的能量。

2.模式原理

短波辐射穿透模块



卫星观测的海洋浮游植物和陆面植被

Ohlmann（2003）方案

在开洋面上，海水对短波辐射的吸收在很大程度上受浮游植物的影响。因此，近期发展了依赖于浮游植物含量的方案，这里的浮游植物含量用海水中叶绿素含量表示。方案仍采用双指数形式，但是吸收的比例系数和穿透的深度都是叶绿素的函数，公式如下：

$$I_0 = S_{w0}\left[A_1(chl)e^{\frac{-z}{B_1(chl)}} + A_2(chl)e^{\frac{-z}{B_2(chl)}}\right]$$

对于叶绿素含量大的海域，穿透系数 B 更小，红外吸收系数 A_1 越大，短波吸收系数 A_2 越小。

```
   subroutine SW_ABSOR


#include <def-undef.h>

       use precision_mod
       use param_mod
       use pmix_mod
       use pconst_mod
       use constant_mod, only: OD0CP
       use sw_mod
       use forc_mod
       use domain

       implicit none
       integer :: iblock

       ! ...........................................................
       ! Start of code body
       ! ...........................................................

#if (defined SOLARCHLORO)
       !-----------------------------------------------------------
       ! chl for table look-up
       !-----------------------------------------------------------
       chloc=(/ 0.001D0,0.005D0, 0.01D0, 0.02D0,&
                 .03D0,   .05D0,   .10D0,   .15D0, &
                 .20D0,   .25D0,   .30D0,   .35D0, &
```

2.模式原理

对流调整模块

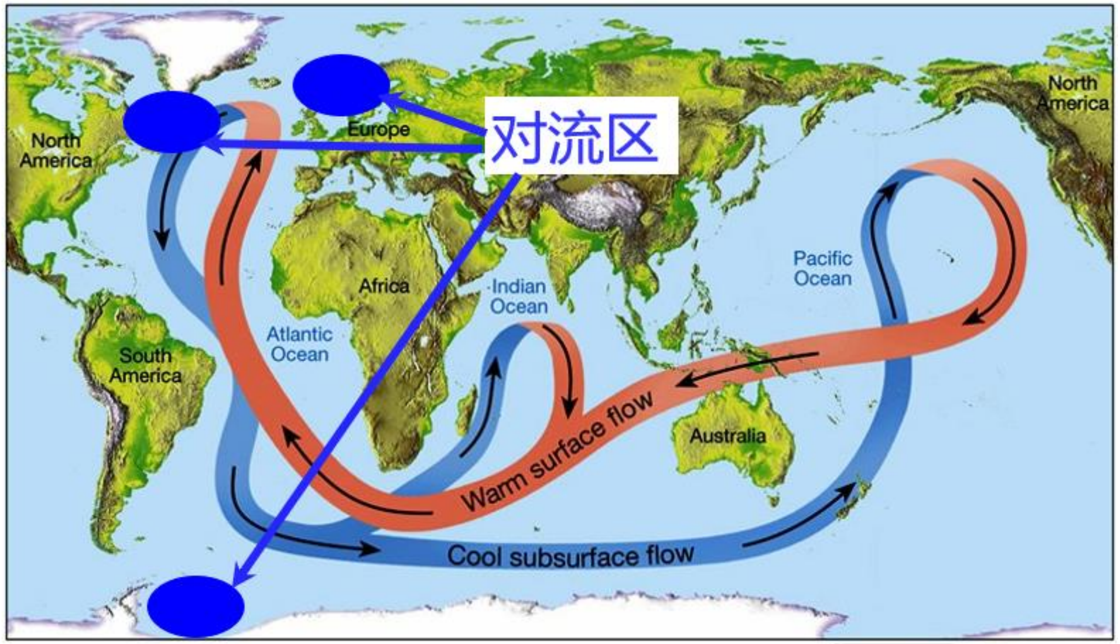
表面冷却
海冰盐析

$$\frac{\partial T}{\partial t} = -(u+u^*)\frac{\partial T}{\partial x} - (v+v^*)\frac{\partial T}{\partial y} - (w+w^*)\frac{\partial T}{\partial z}$$

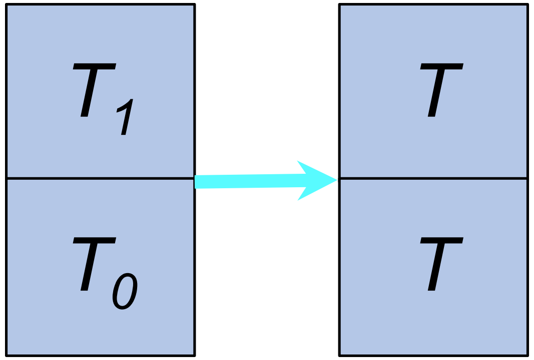
$$+ K_h \Delta T + \frac{\partial}{\partial z}\left(K_v \frac{\partial T}{\partial z}\right) + R(K_I, T)$$

$$+ \frac{1}{\rho C_p}\frac{\partial I_0}{\partial z}$$

$+ C_T$ 表示深对流，没有具体的表达式



静力不稳定！



$$\boxed{T = \frac{T_0 h_0 + T_1 h_1}{h_0 + h_1}}$$

2.模式原理

对流调整模块

具体步骤如下：

（1）顺序计算水柱中每一个网格元的位密度，相邻的网格采用相同的压力参考面；

（2）比较所有的位密度对，查找不稳定；

（3）混合最上面的不稳定对；

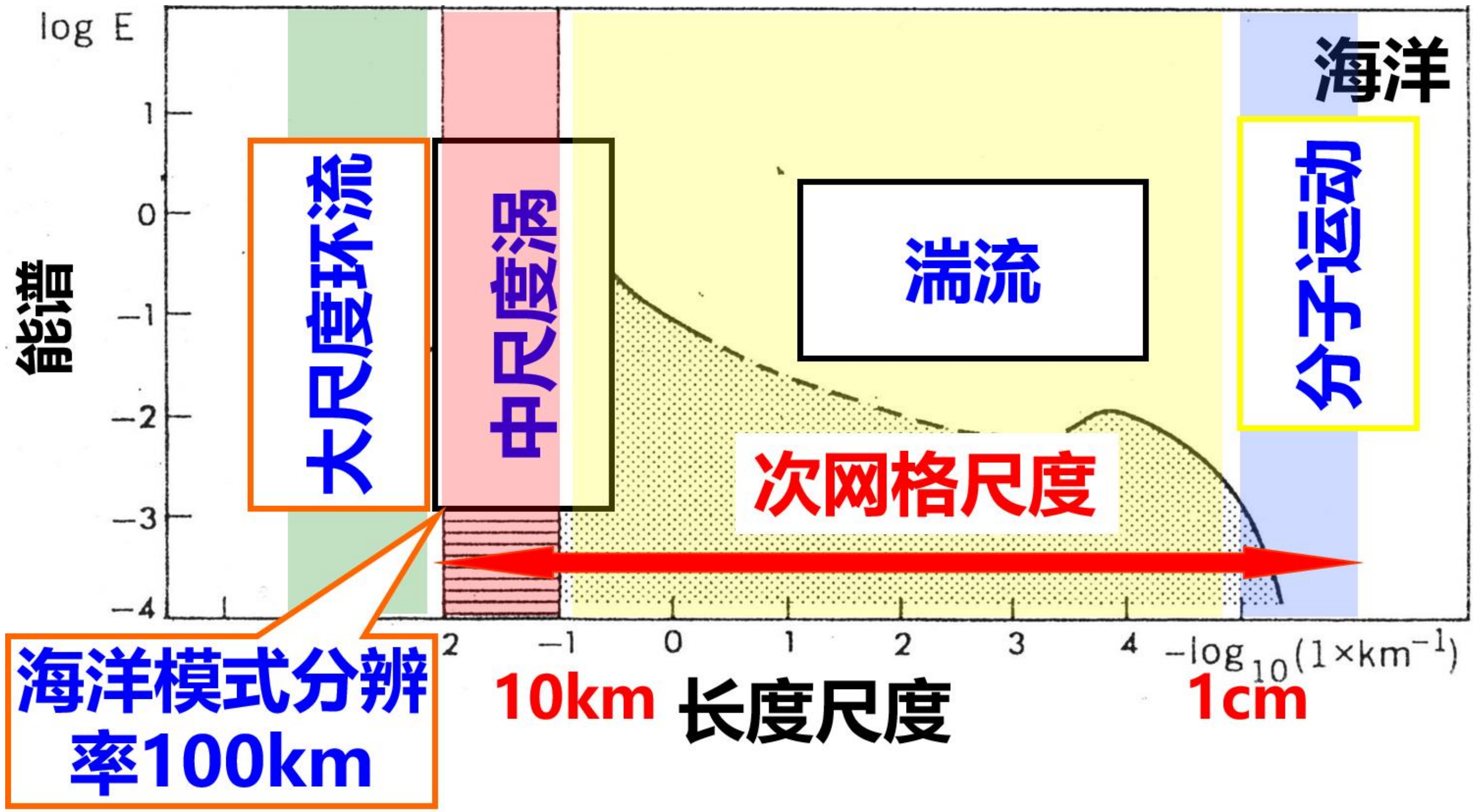
（4）检查下面的一层，如果这一层的位密度比混合后的位密度小，就把这三层混合起来，并继续按照这种方法处理更多的层，直到达到稳定；

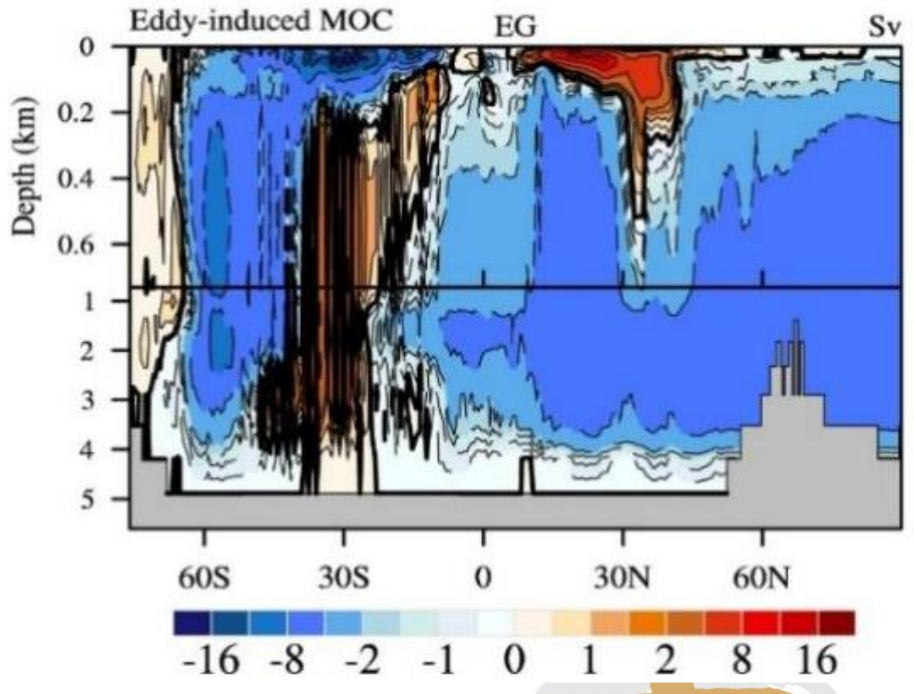
（5）检查混合部分之上的一层，看是否有不稳定出现。如果有不稳定出现，重复步骤(3)；如果没有，继续查找混合部分之下的各层， 若找到不稳定部分， 则重复步骤(3)。

```
      if( adv_tracer(1:8) == 'centered' )  then
         if(ist >= 1)then
            c2dtts = dts*2.0d0
         else
            c2dtts = dts
         end if
      else if( adv_tracer(1:5) == 'tspas') then
         c2dtts = dts
      else
         if(mytid == 0) write(16,*)'error in convadj'
      end if


      do n = 1, ntra
         do k = 1,km
            ! for output dt diffusion
            dt_conv(:,:,k,n,:) =(at(:,:,k,n,:) - atb(:,:,k,n,:))/ &
                                  c2dtts*vit(:,:,k,:)
         enddo
      enddo

      tend = tend + dt_conv    ! total tendency

      if(trim(adv_tracer) == 'tspas') then
         atb(:,:,1:km,:,:) = at(:,:,1:km,:,:)
      end if

      return
   end subroutine CONVADJ
```

2.模式原理

中尺度涡参数化模块





涡致经圈翻转环流

中尺度涡是次网格尺度的主要运动形式，也是参数化的主要对象

2.模式原理

中尺度涡参数化模块

以温度方程为例，GM90 方案在温度（以及盐度和其他示踪物）方程的平流项中引入了涡旋诱发的输送分量表达式如下：

$$\frac{\partial T}{\partial t}+\left(v+v^*\right)\frac{\partial T}{a\partial\theta}+\left(u+u^*\right)\frac{\partial T}{a\sin\theta\partial\lambda}+\left(w+w^*\right)\frac{\partial T}{\partial z}=R\left(A_I,T\right)$$

GM90方案 其中 $v^*=-\frac{\partial}{\partial z}\left(A_{ITH}S_\theta\right)$， $u^*=-\frac{\partial}{\partial z}\left(A_{ITH}S_\lambda\right)$

$$S_\theta=-\left(\frac{\partial\rho_{pot}}{\partial z}\right)^{-1}\frac{\partial\rho_{pot}}{a\partial\theta},\qquad S_\lambda=-\left(\frac{\partial\rho_{pot}}{\partial z}\right)^{-1}\frac{\partial\rho_{pot}}{a\sin\theta\partial\lambda}$$

u^*和 v^*为涡旋诱发的输送速度，A_{ITH}为沿等密度面的厚度扩散系数，S_θ和 S_λ为两个方向的等密度面的坡度。

调整后的水平混合系数具体计算公式如下：

R82方案

$$K=\frac{A_H}{(1+\delta^2)}\times\begin{bmatrix}1+\frac{\rho_y^2+\varepsilon\rho_x^2}{\rho_z^2} & (\varepsilon-1)\frac{\rho_x\rho_y}{\rho_z^2} & (\varepsilon-1)\frac{\rho_x}{\rho_z}\\ (\varepsilon-1)\frac{\rho_x\rho_y}{\rho_z^2} & 1+\frac{\rho_x^2+\varepsilon\rho_y^2}{\rho_z^2} & (\varepsilon-1)\frac{\rho_y}{\rho_z}\\ (\varepsilon-1)\frac{\rho_x}{\rho_z} & (\varepsilon-1)\frac{\rho_y}{\rho_z} & \varepsilon+\delta^2\end{bmatrix}$$

```
       !-----------------------------------------------------------------
       ! evaluate K2(,,3) centered on the northern face of "T" cells
       !-----------------------------------------------------------------
       call K2_3

       !-----------------------------------------------------------------
       ! evaluate K1(,,3) centered on eastern face of "T" cells
       !-----------------------------------------------------------------
       call K1_3

       !-----------------------------------------------------------------
       ! evaluate K3(,,1..3) centered on bottom face of "T" cells
       !-----------------------------------------------------------------
       call K3_123

       !-----------------------------------------------------------------
       ! compute isopycnal advective velocities for tracers
       !-----------------------------------------------------------------
       deallocate(e,rhoi)
       allocate(adv_vetiso(imt,km,jmt,max_blocks_clinic) )

       call ISOADV

       return
   end subroutine ISOPYC
```

2.模式原理

Canuto湍流混合模块

扩散系数 $K_\rho(r,N,R_i,\varepsilon)$取决于密度比 r、Brunt-Väisälä 频率 N、Richardson 数 R_i 和动能耗散率 ε，下标 ρ=(m,h,s)分别表示动量、热量和盐度。密度比的表达式为：

$$r = \alpha\partial_z\bar{\theta}/(\beta\partial_z\bar{S})$$

密度比和 Brunt-Väisälä 频率由海洋环流模式的大尺度场计算得到。剪切作用由 Richardson 数表征，并根据观测来调整临界 Richardson 数（~1），混合层内主要为风驱动的大尺度剪切，混合层之下则是小尺度剪切如内波为主。耗散率表示混合相关的物理过程，混合层内主要为风搅拌作用，在混合层下的开洋面上考虑内波破碎等，地形附近则考虑潮耗散或地热加热作用。该方案适用于双扩散稳定、双扩散不稳定、盐指和对流扩散情况：

双扩散稳定：$(\partial_z T>0, \partial_z S<0, r_\rho<0, Ri_T>0)$

双扩散不稳定：$(\partial_z T<0, \partial_z S>0, r_\rho>0, Ri_T<0)$

盐指：$(\partial_z T>0, \partial_z S>0, r_\rho>0, Ri_T>0)$

扩散对流：$(\partial_z T<0, \partial_z S<0, r_\rho>0, Ri_T>0)$

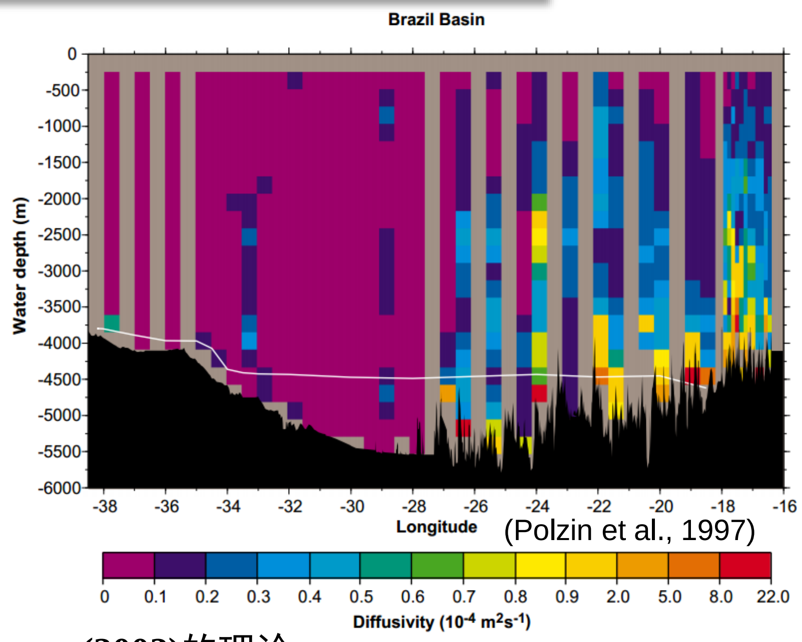
该方案首先计算压力网格点上的粘性、扩散系数，然后使用隐式方案求解垂直扩散方程，并将粘性廓线水平插值到速度网格点，最后求解动量垂直粘性方程。

```
    subroutine canuto_2010_interface(Km_out,     Kh_out,     Ks_out,     Kd_out,     mld_out,&
                                     ts_in,      ss_in,      rho_in,     ri_in,      rrho_in,&
                                     n2_in,      s2_in,      lat_in,     lev_in,     num_lev,ii

        real(r8), parameter :: Rrho_bound = 1.d-3
        real(r8), parameter :: Ri_low    = -1.d+10
        real(r8), parameter :: Ri_high    = 1.d+10
        integer, intent(in) :: num_lev, ii,jj, iblock, isc
        real(r8), intent(in) :: lev_in(km), s2_in(km-1), n2_in(km-1), lat_in
        real(r8), intent(in) :: ts_in(km), ss_in(km), rho_in(km), ri_in(km-1), rrho_in(km-1)
        real(r8), intent(out) :: Km_out(km), Kh_out(km), Ks_out(km), Kd_out(km), mld_out
        real(r8) :: struct_m,struct_h,struct_s, struct_rho
        real(r8) :: struct_m_inf,struct_h_inf,struct_s_inf, struct_rho_inf
        real(r8) :: mix_eff_m, mix_eff_h, mix_eff_s, mix_eff_rho
        real(r8) :: Rf, Rf_inf, R, Rnew, R_inf, Rnew_inf, TKE_mld
        real(r8) :: Ri(km-1), Rrho(km-1), Gm(km-1), out_pi(5), Gm0
        real(r8) :: cube(4)
        integer :: mld_lev, k

        call find_mix_layer_depth(mld_out,mld_lev,rho_in,0.03,lev_in,num_lev)


        Ri(:) = Ri_in(:)
        Rrho(:) = Rrho_in(:)

        where (Ri_in > Ri_high)
            Ri = Ri_high
        elsewhere (Ri_in < Ri_low)
```

```
        do k = 1, num_lev - 1
           call prepare_pi(1.0d10,Rrho(k),out_pi(:))
           call dyn_time_scale_calc(1.0d10,Rrho(k),out_pi(:),Gm(k),cube,ii,jj)
           call struct_function(1.0d10,Rrho(k),out_pi(:),Gm(k),struct_m_inf,struct_h_inf,struct_s_inf,&
                                                   struct_rho_inf,R_inf,Rnew_inf)
           call Rf_calc(Rf_inf,1.0d10,Rrho(k),struct_h_inf,struct_m_inf,Rnew_inf)
           call prepare_pi(Ri(k),Rrho(k),out_pi(:))
           call dyn_time_scale_calc(Ri(k),Rrho(k),out_pi(:),Gm(k),cube,ii,jj)
           call struct_function(Ri(k),Rrho(k),out_pi(:),Gm(k),struct_m,struct_h,struct_s,struct_rho,R,Rnew)
           call mixing_efficiency(mix_eff_m,struct_m,Ri(k),Gm(k))
           call mixing_efficiency(mix_eff_h,struct_h,Ri(k),Gm(k))
           call mixing_efficiency(mix_eff_s,struct_s,Ri(k),Gm(k))
           call mixing_efficiency(mix_eff_rho,struct_rho,Ri(k),Gm(k))
           call Rf_calc(Rf,Ri(k),Rrho(k),struct_h,struct_m,Rnew)

           if ( k <= mld_lev) then
               call mixed_layer_TKE_calc(TKE_mld,mld_out,Gm(k),s2_in(k),lev_in(k),Rf,Rf_inf)
               Km_out(k) = mix_eff_m*TKE_mld/(N2_in(k)+very_small)
               Kh_out(k) = mix_eff_h*TKE_mld/(N2_in(k)+very_small)
               Ks_out(k) = mix_eff_s*TKE_mld/(N2_in(k)+very_small)
               Kd_out(k) = mix_eff_rho*TKE_mld/(N2_in(k)+very_small)
           end if


           if (k > mld_lev) then
               call Thermocline_mixing_coeff_calc(Km_out(k), Kh_out(k), Ks_out(k), Kd_out(k),&
                                                  mix_eff_m,mix_eff_h,mix_eff_s,mix_eff_rho,&
                                                  N2_in(k),&
```

2.模式原理

内潮混合模块



(Polzin et al., 1997)

```
#if( defined TIDEMIX )
                  ak_tide(i,j,:,iblock)=0.0
                  do k = 1,int(kmt(i,j,iblock))-1
                     ak_tide(i,j,k,iblock) = &
                        back_tidalmixing + mixing_ef*                &
                        local_mixing_fraction*wave_dis(i,j,iblock)*      &
                        fz_tide(i,j,k,iblock)/(dmax1(rict(i,j,k,iblock),&
                        1.0d-8)*(wp3(k)*1000.0))

                     ak_tide(i,j,k,iblock) = &
                        dmin1(ak_tide(i,j,k,iblock),max_tidalmixing)

                     richardson(i,j,k,iblock) = rict(i,j,k,iblock)
                     fztidal(i,j,k,iblock)    = fz_tide(i,j,k,iblock)
                     wp3_tidal(i,j,k,iblock)  = wp3(k)
                  end do
"readyc.F90" 670 lines --51%--
```

```
#if( defined TIDEMIX )
        if(mytid == 0 ) then
           iret = NF_OPEN('tidal_energy.nc',nf_nowrite,ncid1)
           call CHECK_ERR(iret)
        endif

        do k = 1,1
           start2(1) = 1 ; count2(1) = imt_global
           start2(2) = 1 ; count2(2) = jmt_global

           if(mytid == 0) then
              iret = NF_GET_VARA_DOUBLE(ncid1, 5,start2,count2,buffer)
              call CHECK_ERR(iret)
              do j = 1 ,jmt_global/2
                 do i = 1, imt_global
                    xx_r8 = buffer(i,j)
                    yy_r8 = buffer(i,jmt_global+1-j)
                    buffer(i,jmt_global+1-j) = xx_r8
                    buffer(i,j) = yy_r8
                 end do
              end do
           endif

           call SCATTER_GLOBAL(wave_dis(:,:,:), buffer, master_task,    &
                               distrb_clinic, field_loc_center,          &
                               field_type_scalar)
        end do
"rdriver.F90" 663 lines --62%--
```

基于St. Laurent(2002)的理论：

$$k_v = k_0 + \frac{\Gamma\epsilon}{N^2} = k_0 + \frac{\Gamma}{N^2}\frac{qE(x,y)}{\rho}F(z),$$

背景扩散率k_0=0.1×$10^{-4}m^2s^{-1}$，混合效率$\Gamma = 0.2$(Osborn,1980)，q:局地耗散效率。
内潮能量通量：$E(x,y) = 1/2\rho_0 N_b \kappa h^2\langle U^2\rangle$，
N_b:沿海床的浮力频率，κ和h:地形粗糙度的波数和振幅尺度，$\langle U^2\rangle$:正压潮变化（方差）。

垂直耗散结构：$F(z) = \frac{e^{-(z+H)/z_s}}{z_s(1-e^{-H/z_s})}$，

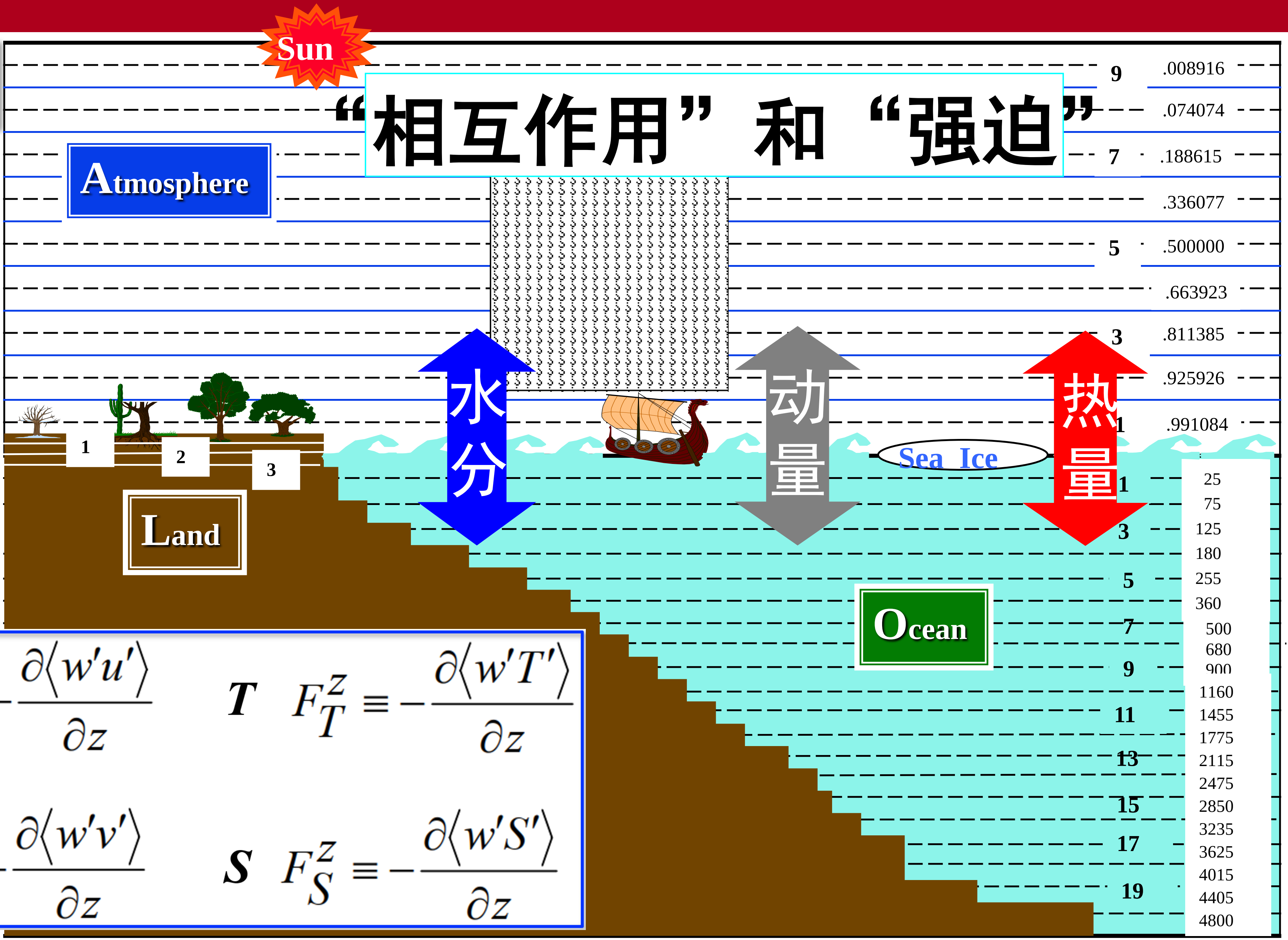
z_s=500m :垂直衰减尺度，H:水柱总深度，F(z)满足$\int_{-H}^{0} F(z)dz = 1$。

2.模式原理

海表边界条件模块

海洋环流主要由三种“外部”影响所驱动：

- ✓ 风
- ✓ 加热/冷却
- ✓ 蒸发/降水



$$U \quad F_u^z \equiv -\frac{\partial\langle w'u'\rangle}{\partial z} \qquad T \quad F_T^z \equiv -\frac{\partial\langle w'T'\rangle}{\partial z}$$

$$V \quad F_v^z \equiv -\frac{\partial\langle w'v'\rangle}{\partial z} \qquad S \quad F_S^z \equiv -\frac{\partial\langle w'S'\rangle}{\partial z}$$

2.模式原理

海表边界条件模块

动量方程、位温方程和盐度方程的海表边界条件分别是：

$$-\langle w'u'\rangle\big|_{z=z_0} = \frac{1}{\rho_0}\tau_x$$

$$-\langle w'v'\rangle\big|_{z=z_0} = \frac{1}{\rho_0}\tau_y$$

$$-\langle w'T'\rangle\big|_{z=z_0} = \frac{1}{\rho_0 c_p}\left(Q_{LW}+Q_S+Q_L\right)$$

$$-\langle w'S'\rangle\big|_{z=z_0} = -\frac{S_0}{\rho_0}\left(P-E+R\right)$$

以上三式的左端是海洋的垂直湍流通量，右端项分别涉及风应力、热通量和淡水通量，其中，参数 ρ_0=1029 kgm^{-3} 和 cp=3901$Jkg^{-1}K^{-1}$ 分别是海水的参考密度和定压比热。

```
      do iblock = 1, nblocks_clinic
         this_block = GET_BLOCK(blocks_clinic(iblock),iblock)
         do j = this_block%je, this_block%jb, -1
            do i = this_block%ib, this_block%ie
               tsf(i,j,iblock)   = (lat1(i,j,iblock) + sen(i,j,iblock)    +     &
                                   lwup(i,j,iblock)  + lwdn(i,j,iblock)  +     &
                                   netsw(i,j,iblock) + melth(i,j,iblock) -     &
                                   iceoff(i,j,iblock)*shr_const_latice   -     &
                                   snowl(i,j,iblock)*shr_const_latice)*OD0CP
                                   ! net heat flux

               swv(i,j,iblock)   = netsw(i,j,iblock)    ! net solar radiation
               nswv(i,j,iblock)  = lat1(i,j,iblock) + sen(i,j,iblock)  + &
                                   lwup(i,j,iblock) + lwdn(i,j,iblock) + &
                                   melth(i,j,iblock)-                    &
                                   iceoff(i,j,iblock)*shr_const_latice - &
                                   snowl(i,j,iblock)*shr_const_latice
                                   ! net heat flux error

               ssf(i,j,iblock)   = - (prec(i,j,iblock)  + evap(i,j,iblock) + &
                                     meltw(i,j,iblock)  + roff(i,j,iblock) + &
                                     iceoff(i,j,iblock) +                    &
                                     salt(i,j,iblock))*OD0*34.7*1.0D-3/DZP(1)

               tmp_su(i,j,iblock) =  taux(i,j,iblock)*cos(anglet(i,j,iblock))+&
                                     tauy(i,j,iblock)*sin(anglet(i,j,iblock))
               tmp_sv(i,j,iblock) = -taux(i,j,iblock)*cos(anglet(i,j,iblock))+&
                                     taux(i,j,iblock)*sin(anglet(i,j,iblock))
            end do
         end do
      end do

"POST_CPL_mod_add.F90" 202 lines --37%--
```

2.模式原理

底边界层模块

海底边界条件分为两部分：

✓ 海底地形的抬升作用；

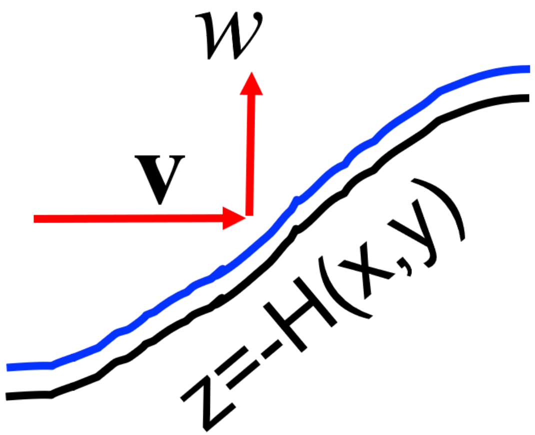
$$w\big|_{z=-H} = -(\mathbf{v}\cdot\nabla H)_{z=-H}$$

✓ 海底摩擦效应；

$$-\langle w'\mathbf{v}'\rangle_{z=-H} = \frac{1}{\rho_0}\boldsymbol{\tau}_{\mathrm{b}}$$

底摩擦应力 τ_{b} 由参数化给出：

$$(\tau_{b\theta},\tau_{b\lambda}) = \rho_0 C_0\sqrt{u_b^2+v_b^2}\,(u_b\sin\alpha + v_b\cos\alpha, u_b\cos\alpha - v_b\sin\alpha)$$




```fortran
      do iblock = 1, nblocks_clinic
         do j = 2, jmt-1
            do i = 2,imt-1
               kmb = kmu(i,j,iblock)
               if(kmb >= 1) then
                  sbcx(i,j,iblock) = su(i,j,iblock)* od0
                  sbcy(i,j,iblock) = sv(i,j,iblock)* od0
                  bbcx(i,j,iblock) = &
                     c0f*sqrt(up(i,j,kmb,iblock)*up(i,j,kmb,iblock)+     &
                           vp(i,j,kmb,iblock)*vp(i,j,kmb,iblock))*    &
                          (up(i,j,kmb,iblock)*cag+snlat(i,j,iblock)*  &
                           vp(i,j,kmb,iblock)*sag)
                  bbcy(i,j,iblock)= &
                     c0f*sqrt(up(i,j,kmb,iblock)*up(i,j,kmb,iblock)  + &
                           vp(i,j,kmb,iblock)*vp(i,j,kmb,iblock))*  &
                          (-snlat(i,j,iblock)*up(i,j,kmb,iblock)  *  &
                           sag+vp(i,j,kmb,iblock)*cag)
               else
                  sbcx(i,j,iblock) = 0.0d0
                  sbcy(i,j,iblock) = 0.0d0
                  bbcx(i,j,iblock) = 0.0d0
                  bbcy(i,j,iblock) = 0.0d0
               endif
               dlub(i,j,iblock) = dlub(i,j,iblock) + (sbcx(i,j,iblock) - &
                                  bbcx(i,j,iblock))*ohbu(i,j,iblock)
               dlvb(i,j,iblock) = dlvb(i,j,iblock) + (sbcy(i,j,iblock) - &
                                  bbcy(i,j,iblock))*ohbu(i,j,iblock)
            enddo
         enddo
      enddo
"READYC_mod_add.F90" 692 lines --88%--
```

其中，$\tau_{b\theta}$和$\tau_{b\lambda}$为海底的摩擦应力； $C_0=2.6\times10^{-3}$；

如果余纬θ>90°时α=10°，而θ<90°时α=-10°。

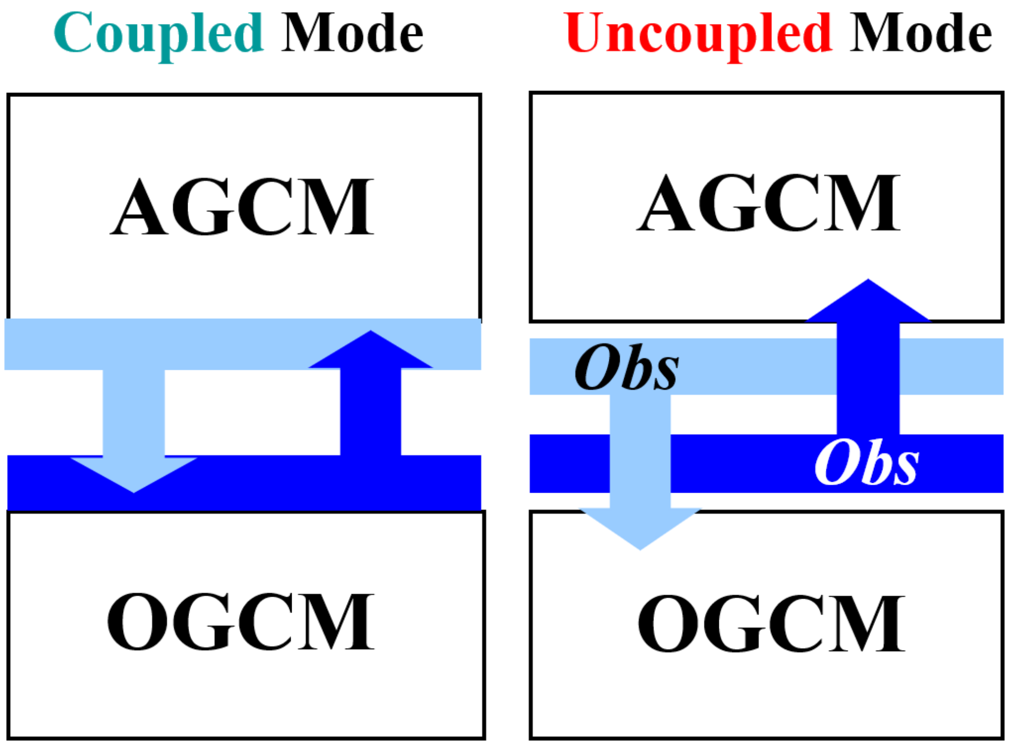
Earth Lab

3.数据制备

初始场

"冷"启动：
start_type = 'startup'
通常初始速度为0，海洋温、盐场取自观测，或者指定的数据；

"热"启动：
start_type = 'continue'
初始场取自模式运行过程中保存的瞬时重启动场文件。



强迫场

变量
10m空气密度
向下的长波辐射
降水
10m比湿
径流量
海平面气压
向下的短波辐射
向上的短波辐射
10m气温
10m纬向风
10m经向风

常用的海洋温、盐气候态观测数据集有： WOA、PHC3.0等；

常用的强迫场数据集有：CORE-I（NYF），CORE-II（IAF），JRA55-do，ERA20C，CRA，GFS等。

3.数据制备

模式输出结果

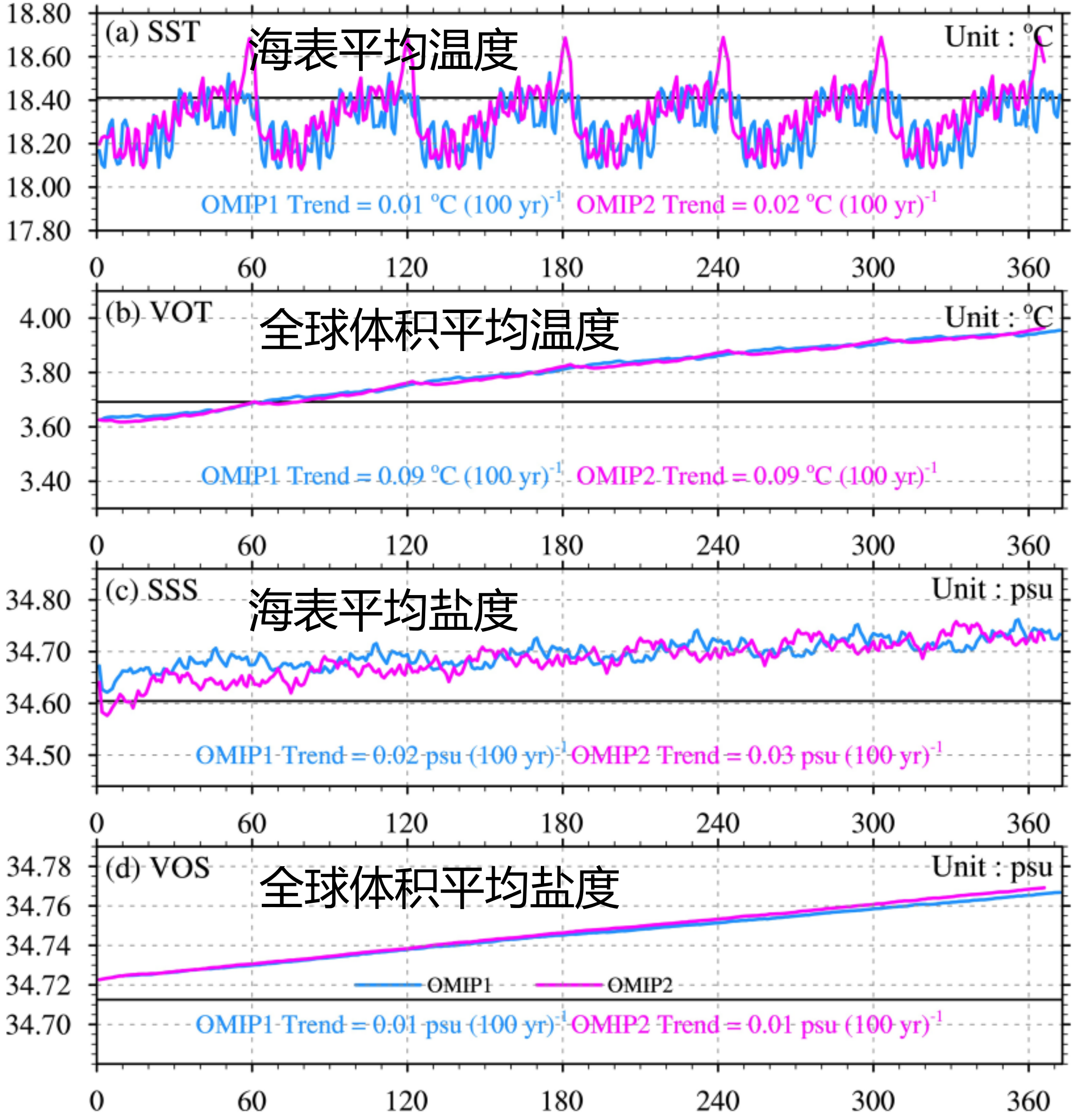
海洋模式瞬时重启动场文件 fort.22.0001(年)-02(月)-01(日)	
h0	海表面高度
u	纬向海流
v	经向海流
at(1)	海温
at(2)	盐度
ws	垂向速度
su	纬向风应力
sv	经向风应力
swv	短波辐射
lwv	长波辐射
sshf	感热
lthf	潜热
fresh	虚盐度通量

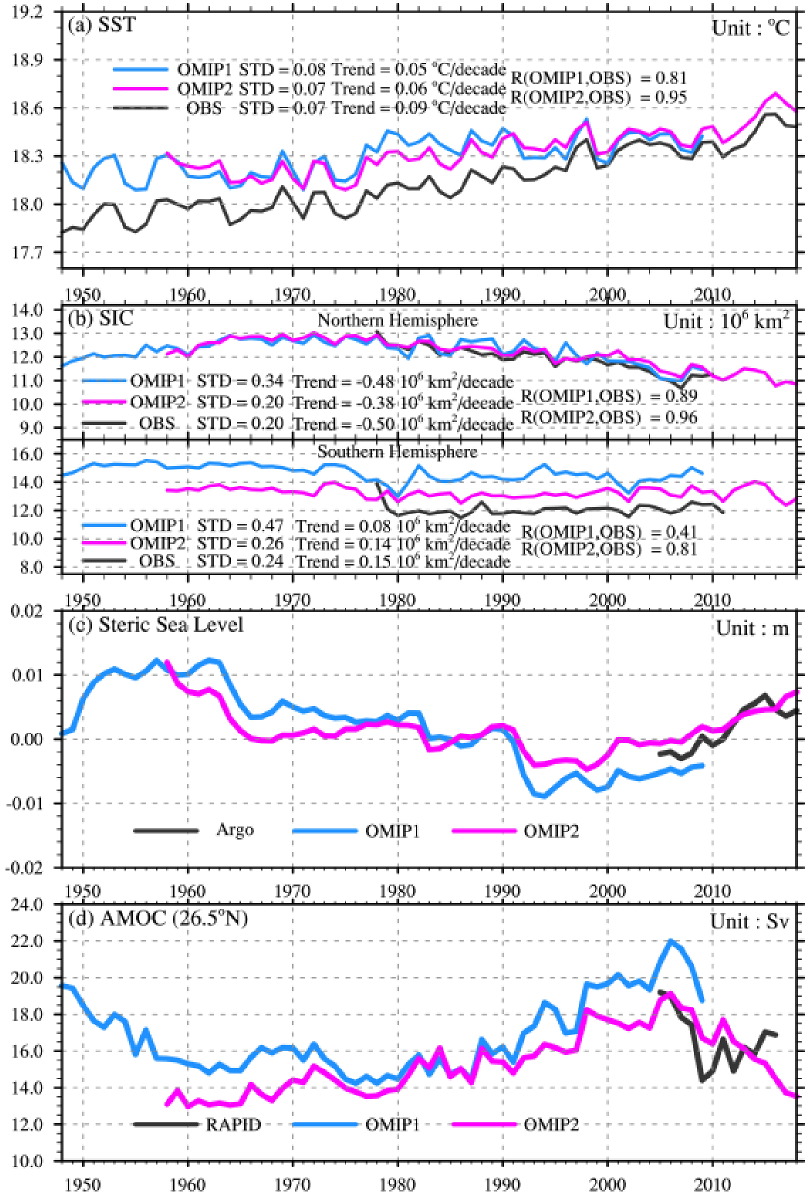
海洋模式月平均输出结果——MMEAN0001(年)-01(月).nc			
time	时间	akt/aks	湍流扩散系数
lat/lon	经纬度（T网格）	aktide	潮致扩散系数
ulat/ulon	经纬度（U网格）	athkdf	厚度扩散系数
lat_aux	纬向辅助网格	su	纬向风应力
lev1/lev	整/半层垂直网格	sv	经向风应力
area	网格面积（T点）	net1	净的海表热通量
z0	海表面高度	lthf	潜热
ts	海温	sshf	感热
ss	盐度	lwv	长波辐射
us	纬向海流	swv	短波辐射
vs	经向海流	qice	海冰溶解热
ifrac	海冰密集度	net2	净的盐度通量
ustar	涡致纬向流速	fresh	虚盐度通量
vstar	涡致经向流速	runoff	径流
wstar	涡致垂向流速	psi_euler	欧拉经圈翻转环流
ws	垂向速度	psi_eddy	涡致经圈翻转环流
mld	混合层深度	bsf	正压流函数
ic1	通风层数(包括对流)	mth_adv	经向热输送(欧拉)
ic2	通风层数	mth_adv_iso	经向热输送(中尺度涡)
akm	湍流黏性系数	mth_dif	经向热输送(扩散)

```
[yuzp@login01 run]$ ncdump -h MMEAN0001-02.nc
netcdf MMEAN0001-02 {
dimensions:
        y = 218 ;
        lat_aux = 316 ;
        tracer_dim = 2 ;
        x = 360 ;
        basin = 2 ;
        lev = 30 ;
        lev1 = 31 ;
        time = UNLIMITED ; // (1 currently)
variables:
        float lat(y, x) ;
                lat:long_name = "latitude (on T grids)" ;
                lat:units = "degrees_north" ;
        float lon(y, x) ;
                lon:long_name = "longitude (on T grids)" ;
                lon:units = "degrees_east" ;
        float ulat(y, x) ;
                ulat:long_name = "latitude (on U grids)" ;
                ulat:units = "degrees_north" ;
        float ulon(y, x) ;
                ulon:long_name = "longitude (on U grids)" ;
                ulon:units = "degrees_east" ;
        double lat_aux(lat_aux) ;
        float lev(lev) ;
                lev:long_name = "depth (on T grids)" ;
                lev:units = "meter" ;
        float lev1(lev1) ;
                lev1:long_name = "depth (on V grids)" ;
                lev1:units = "meter" ;
        double time(time) ;
                time:long_name = "time" ;
                time:units = "months since 0001-01-01" ;
        float area(y, x) ;
                area:long_name = "T grid cell area" ;
                area:units = "m*m" ;
                area:_FillValue = 1.e+35f ;
                area:coordinates = "lon lat" ;
        float z0(time, y, x) ;
                z0:long_name = "sea surface height" ;
                z0:units = "meter" ;
```

4.结果诊断分析与应用案例

海洋模式比较计划（OMIP）





海表平均温度

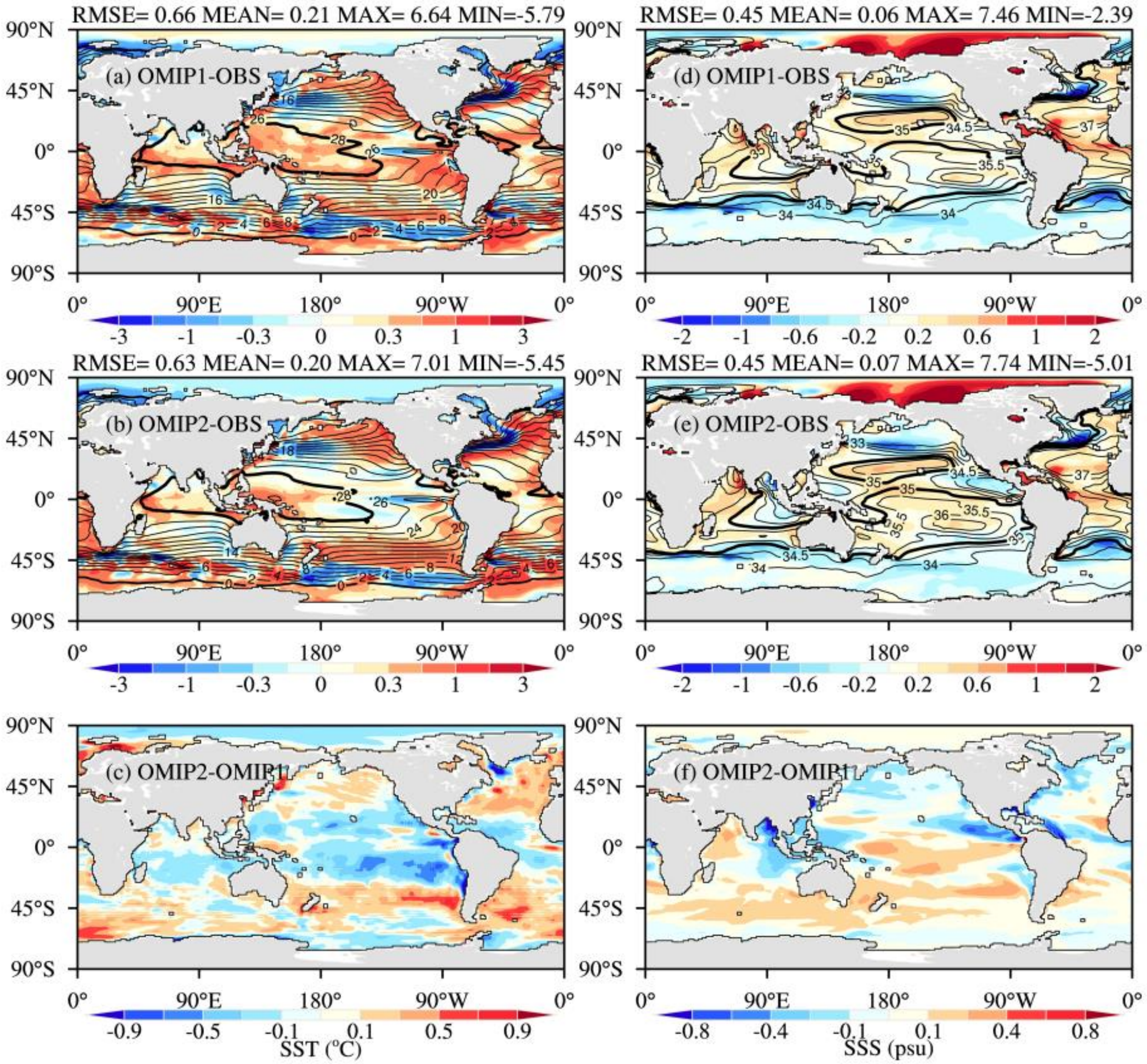
海冰密集度

比容海平面

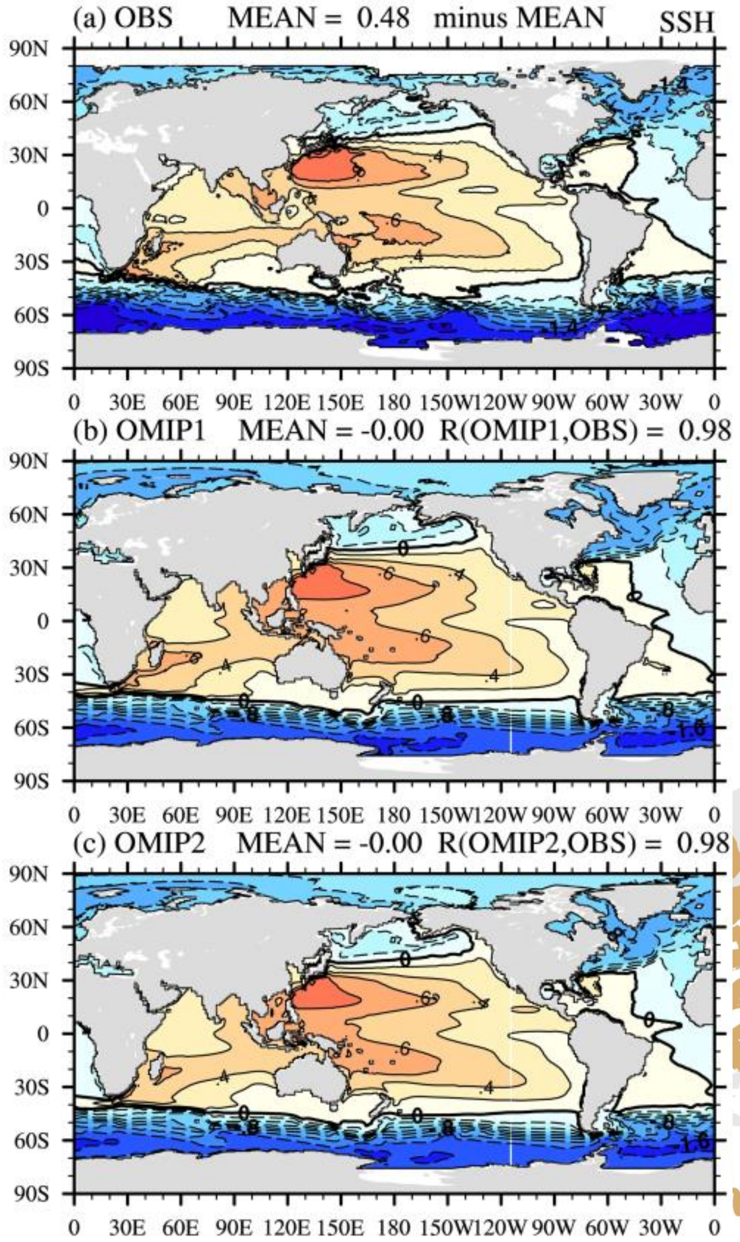
AMOC指数

4.结果诊断分析与应用案例

海表面温度、海表面盐度与观测的对比

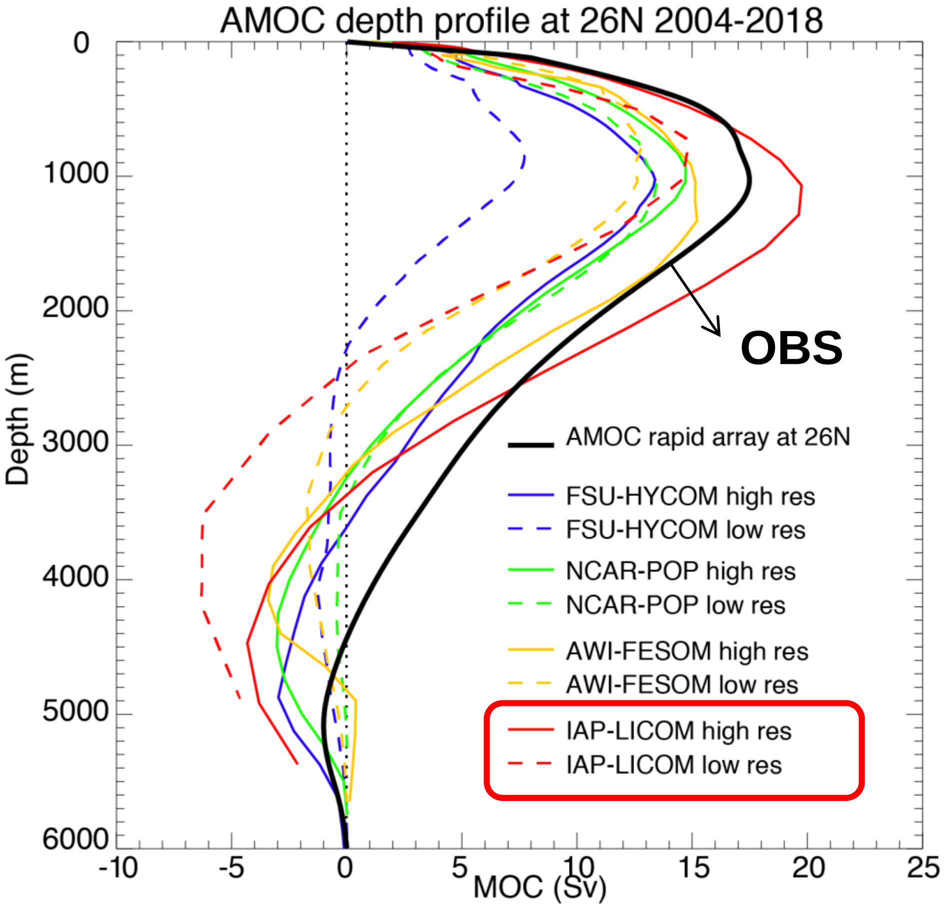


海表面高度异常与观测的对比



4.结果诊断分析与应用案例

北大西洋经圈翻转环流AMOC

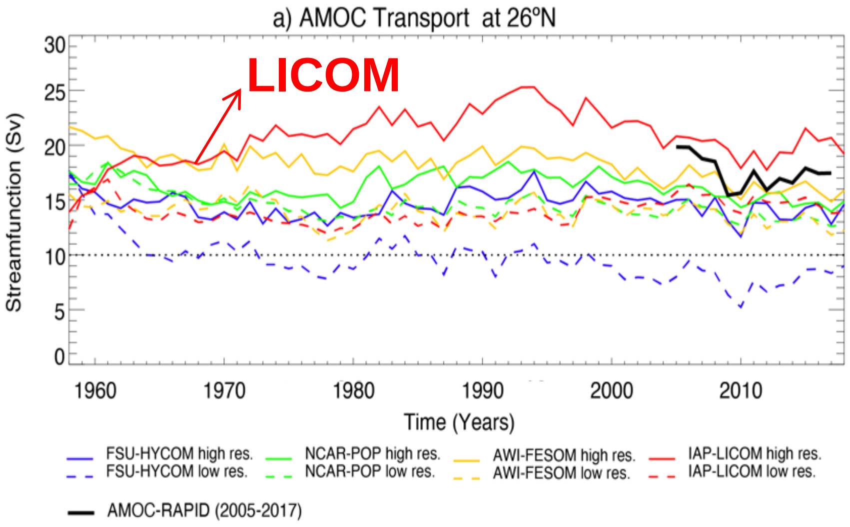


中国 LICOM
美国 HYCOM
美国 POP
德国 FESOM

模拟61年

LICOM3模拟AMOC上层结构与观测接近

AMOC时间变化



LICOM3模拟AMOC时间演变与观测基本一致

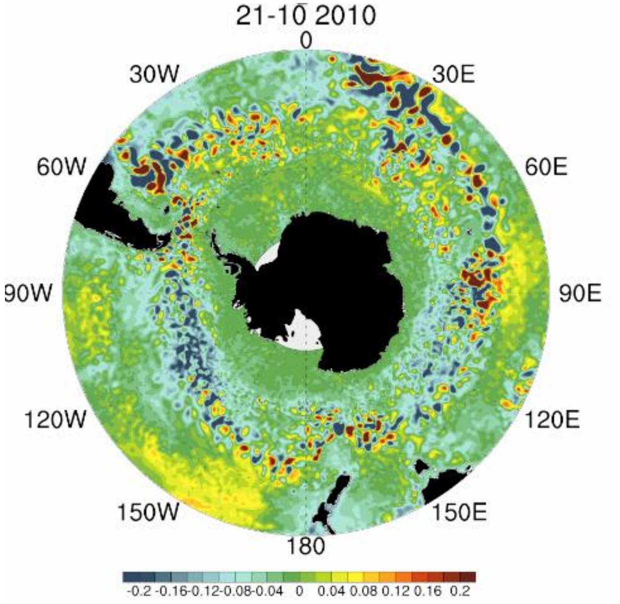
Chassignet et al. 2020, GMD
Li et al. 2020 AAS

4.结果诊断分析与应用案例

1/10°，平均7km

范围	全球， 0-6000米水深，垂直55层
网格	三极网格，3600X2302X55
物理参数化方案	垂直混合，潮混合，双调和+Leith 粘性
CPU	并行，提高3倍性能

LICOM3模拟的海表高度异常



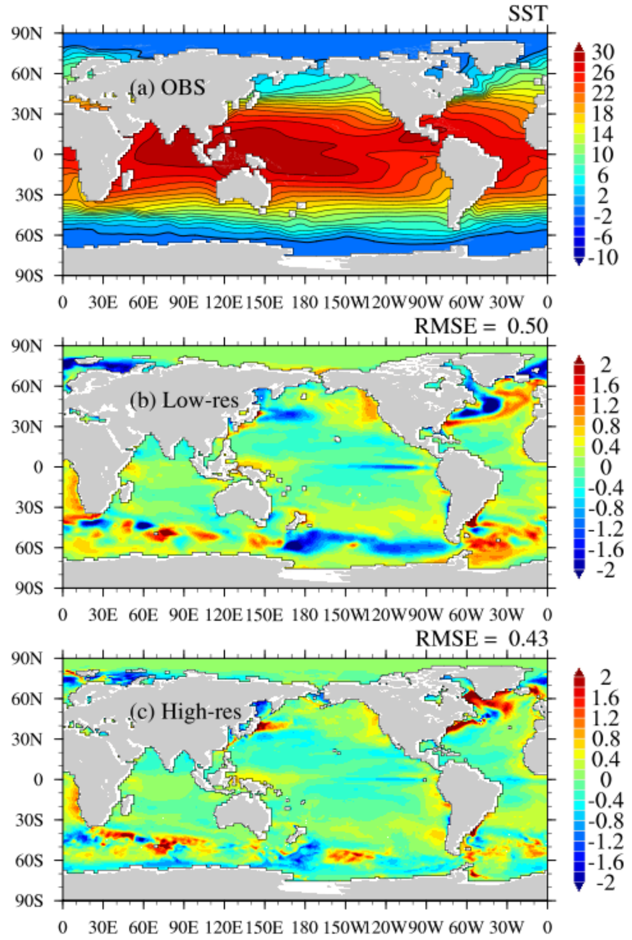
分辨海洋中尺度涡旋

海表面温度SST&偏差

观测

低分辨率LICOM

高分辨率LICOM

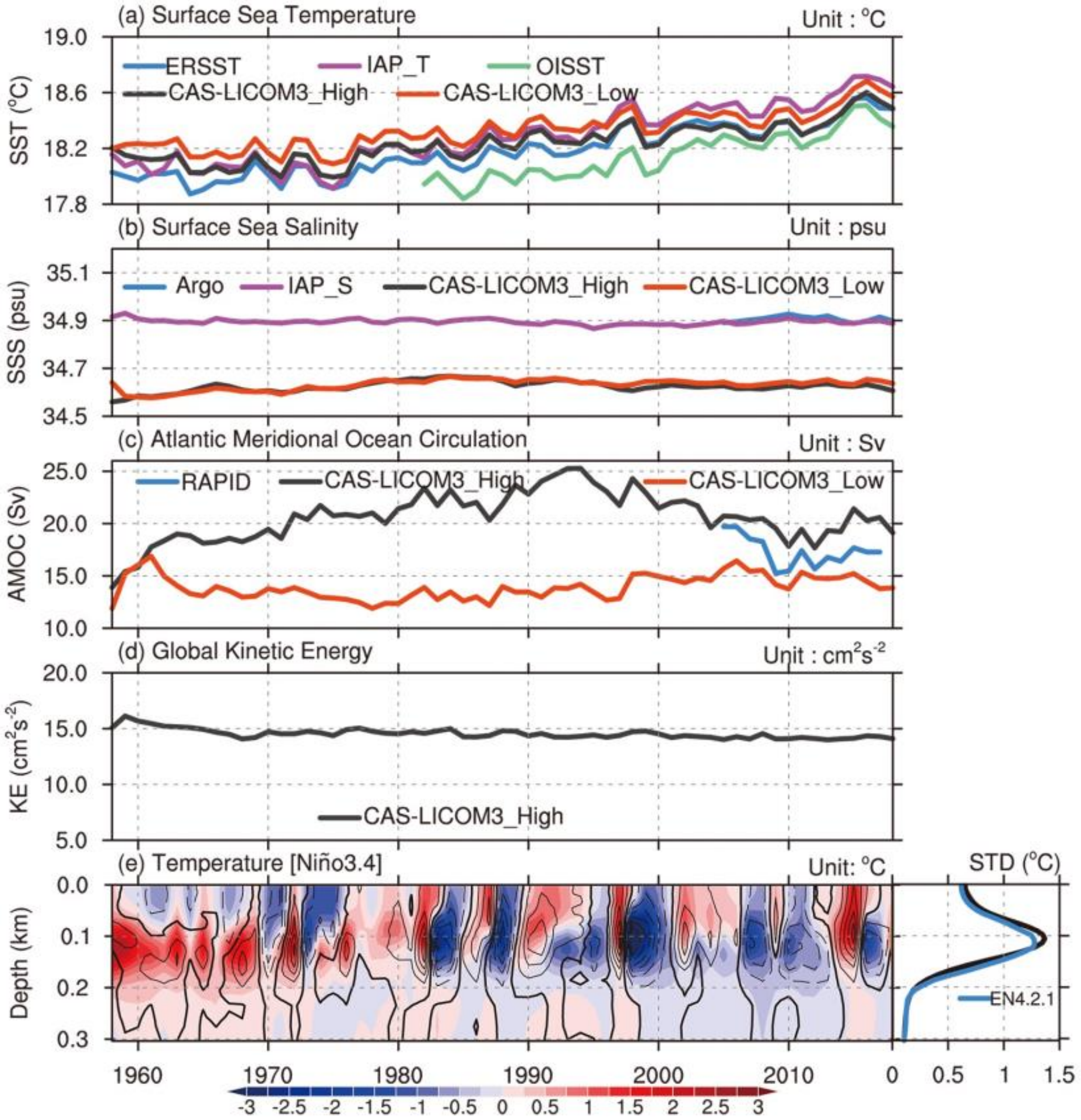


高分辨率模拟的SST偏差减小

Earth Lab

4.结果诊断分析与应用案例

高分辨率海洋模式比较计划



海表平均温度

海表平均盐度

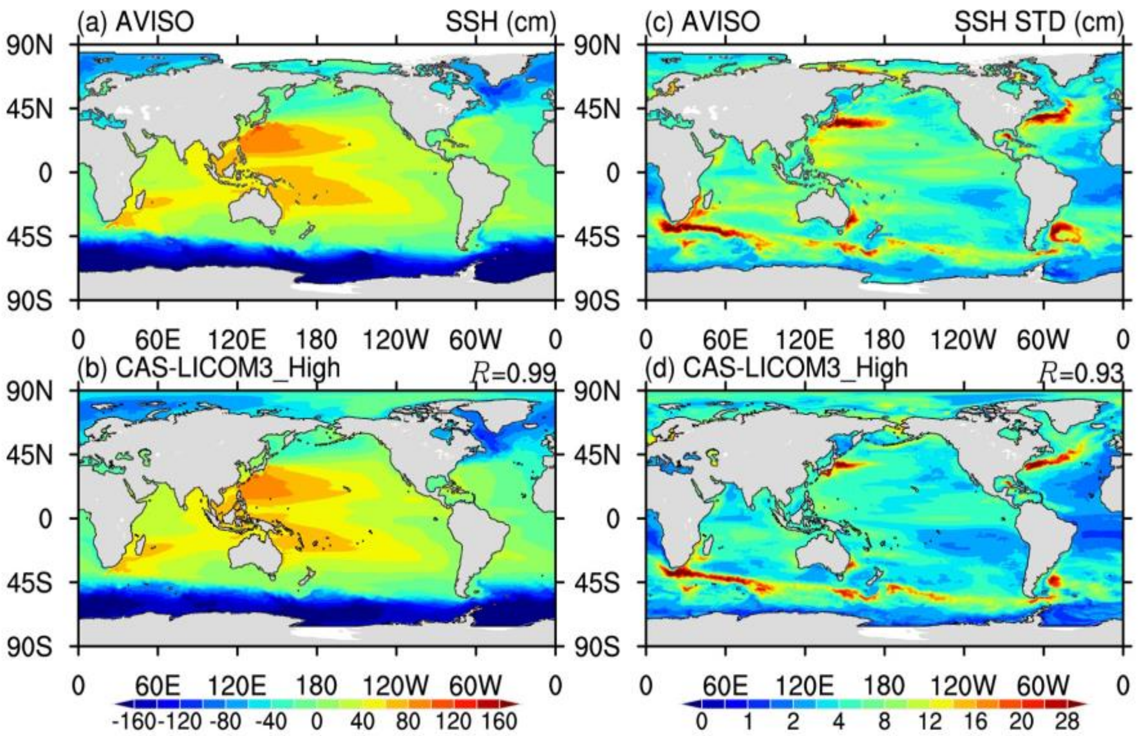
AMOC指数

全球动能

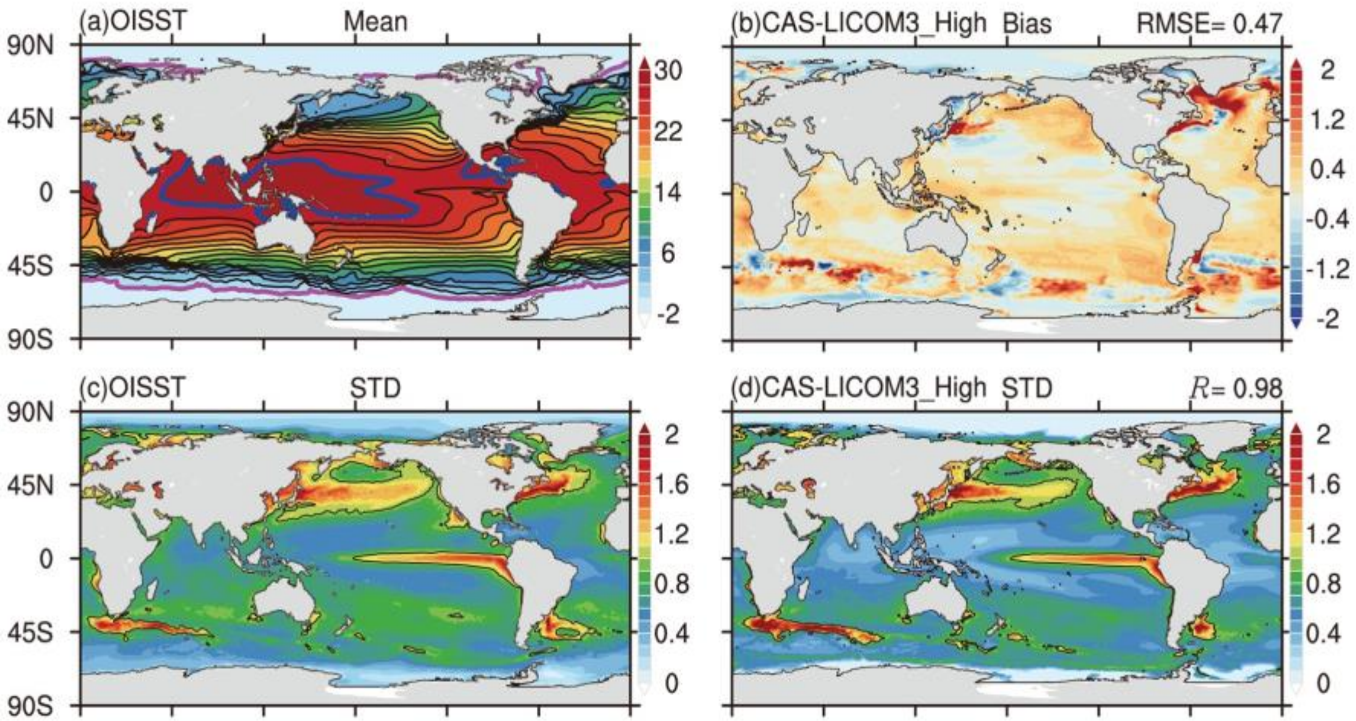
海温廓线

4.结果诊断分析与应用案例

海表面高度

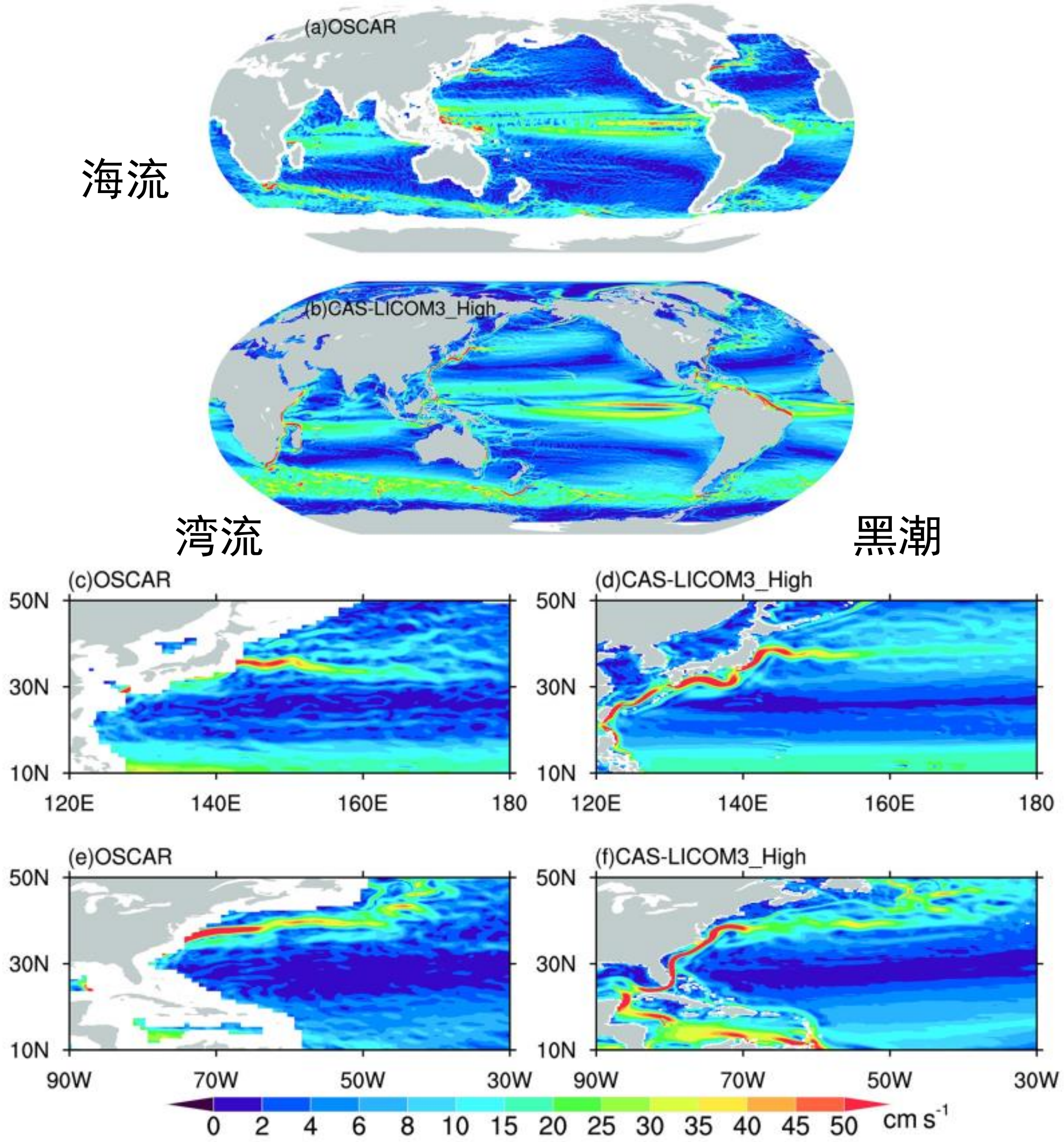


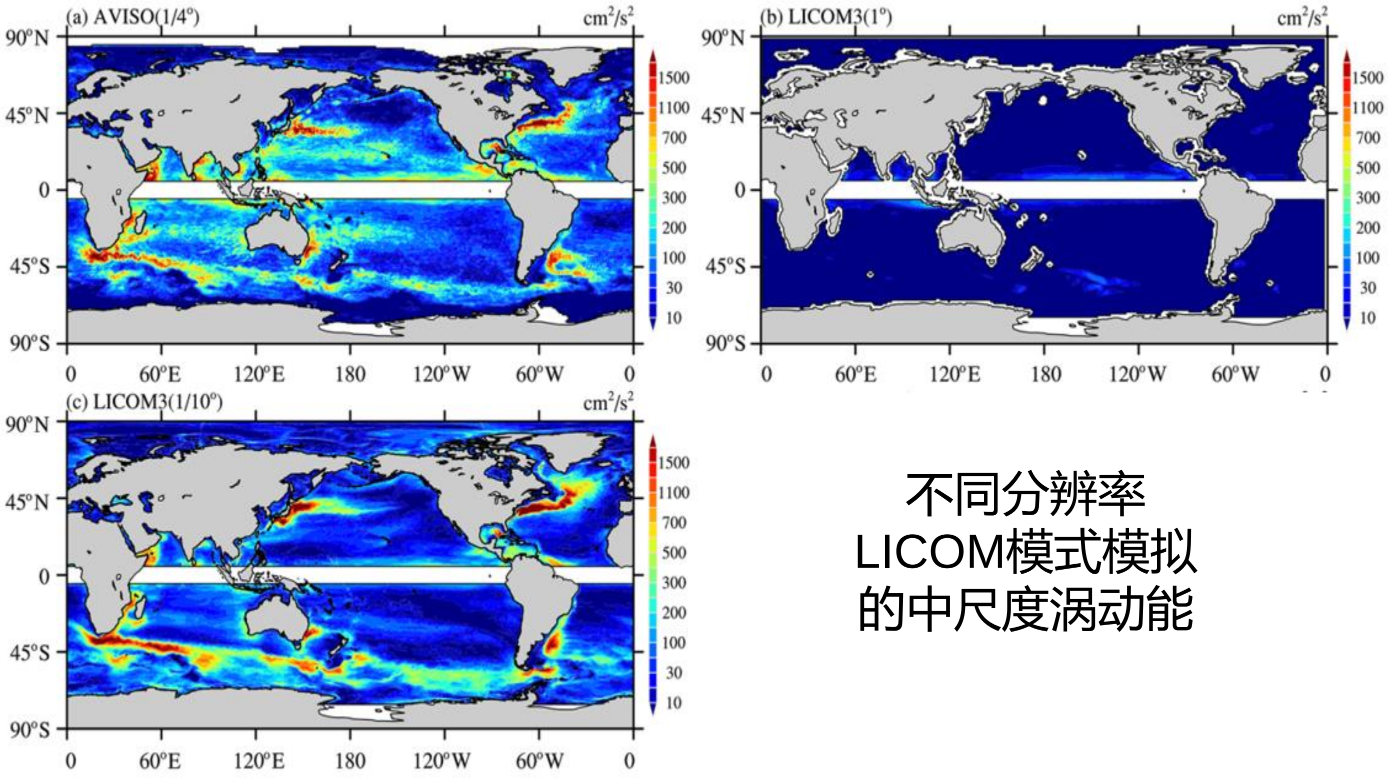
海表面温度



Earth Lab

4.结果诊断分析与应用案例





不同分辨率
LICOM模式模拟
的中尺度涡动能

谢　谢

寰

Earth Lab