全球植被动力学模式分系统

报告人：朱家文

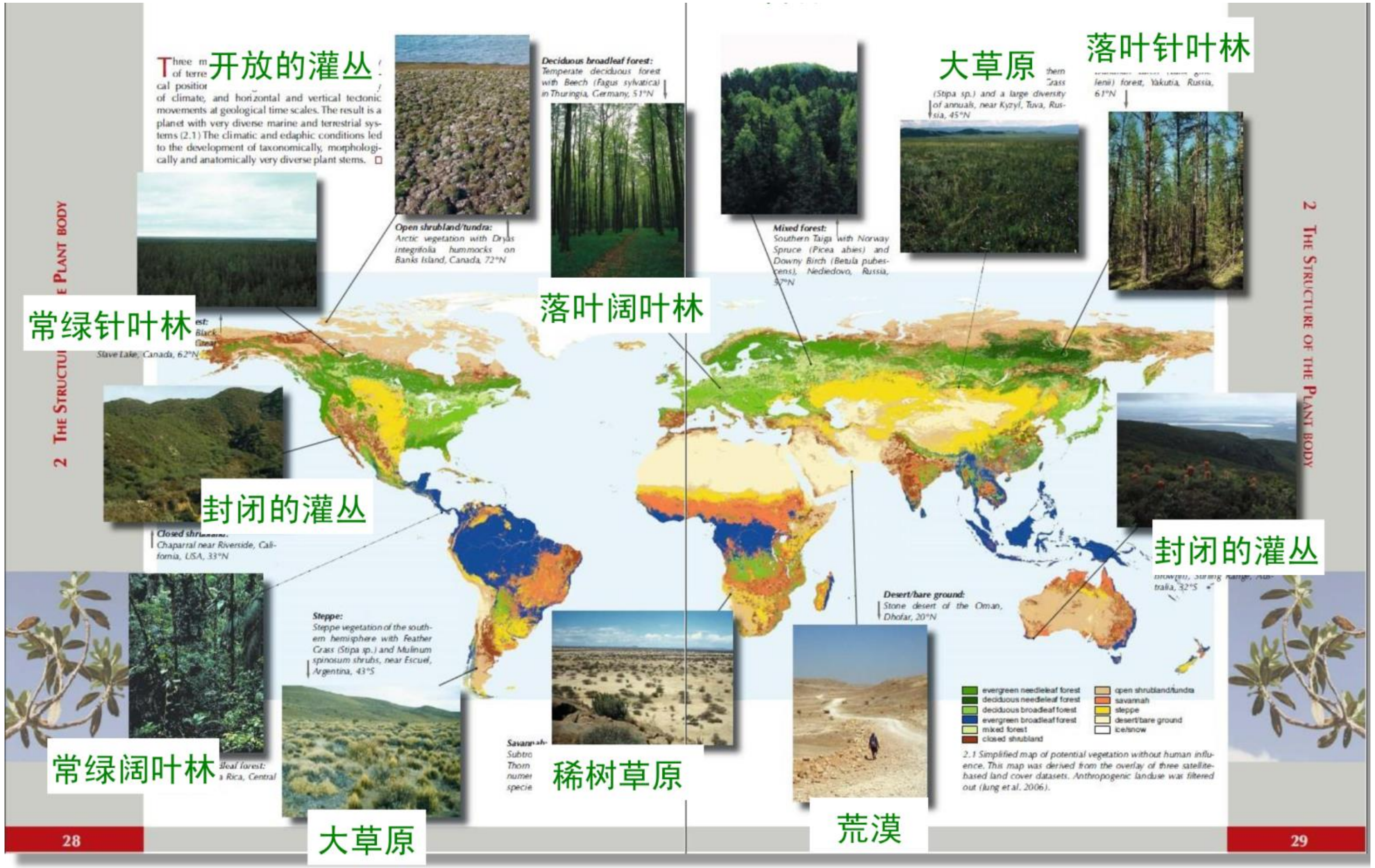
中国科学院大气物理研究所

2022 年 5 月 27 日

主要内容

植被动力学模式（IAP-DGVM）分系统介绍

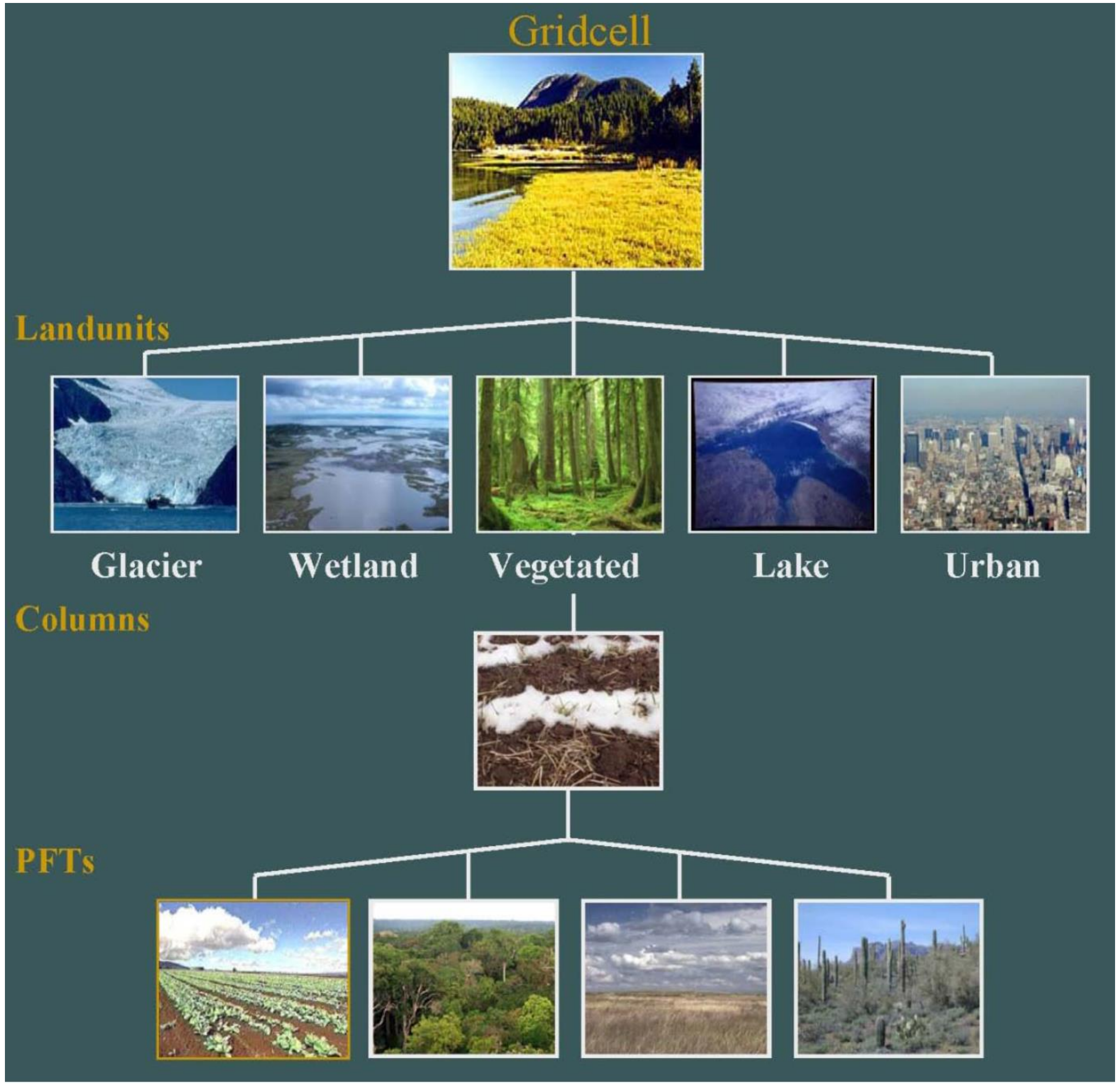
- 全球植被动力学模式（Dynamic Global Vegetation Model-DGVM）是模拟全球（自然界）植被的分布、结构及其变化的**中等复杂动力学模式**，其特点是符合现有的陆表过程模式的框架结构，便于与气候系统模式耦合，用于**全球气候和环境变化研究**，是地球系统模式的重要组成部分；



（*Schweingruber et al*, 2006）

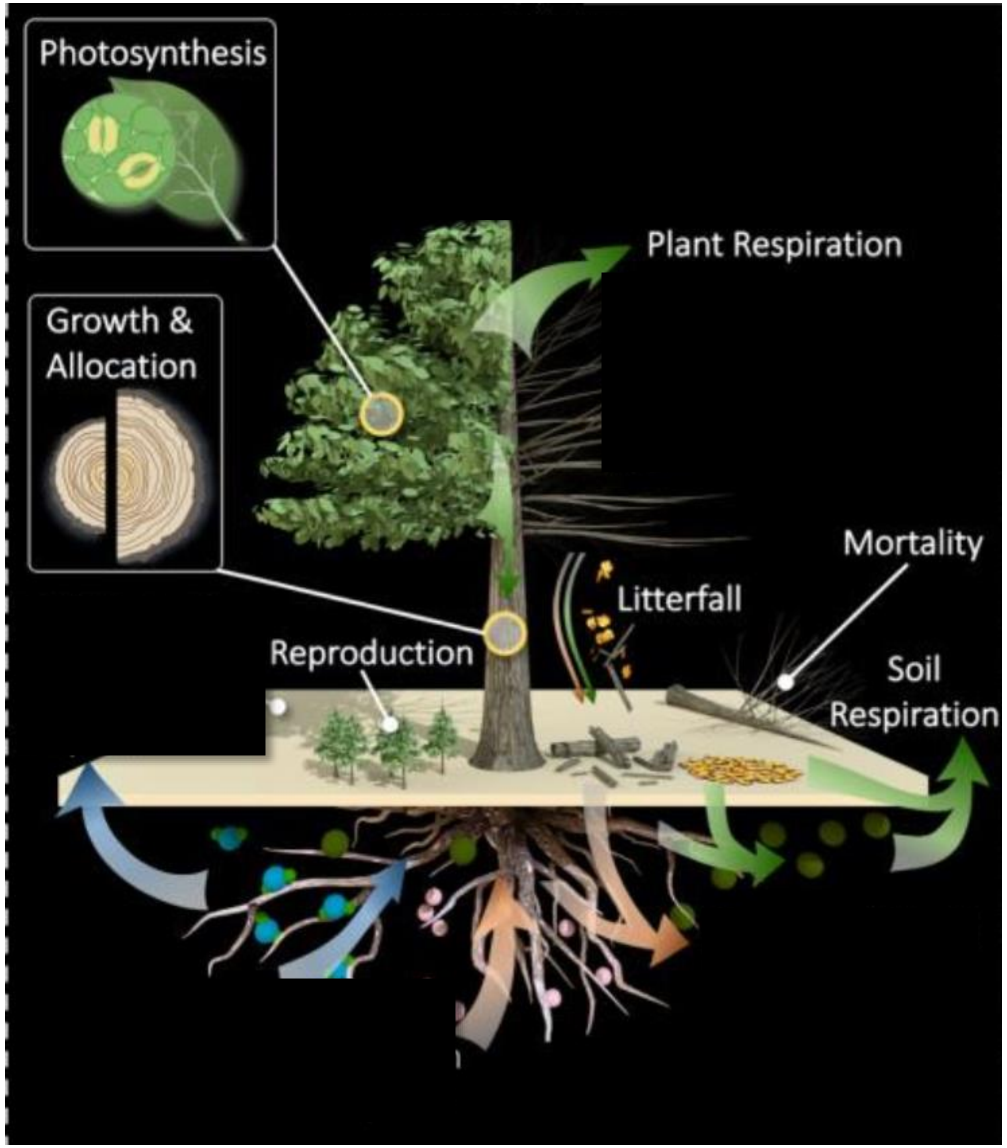
植被动力学模式（IAP-DGVM）分系统介绍

IAP-DGVM是中科院大气所**自主研发**的全球植被动力学模式



PFTs（**Plant Functional Types-植被功能型**）
热带阔叶常绿树
热带阔叶落叶树
温带阔叶常绿树
温带针叶常绿树
温带阔叶落叶树
寒带针叶常绿树
寒带阔叶落叶树
温带阔叶落叶灌木
寒带阔叶落叶灌木
C4 草
C3 非极地草
C3 极地草

植被动力学模式（IAP-DGVM）分系统介绍

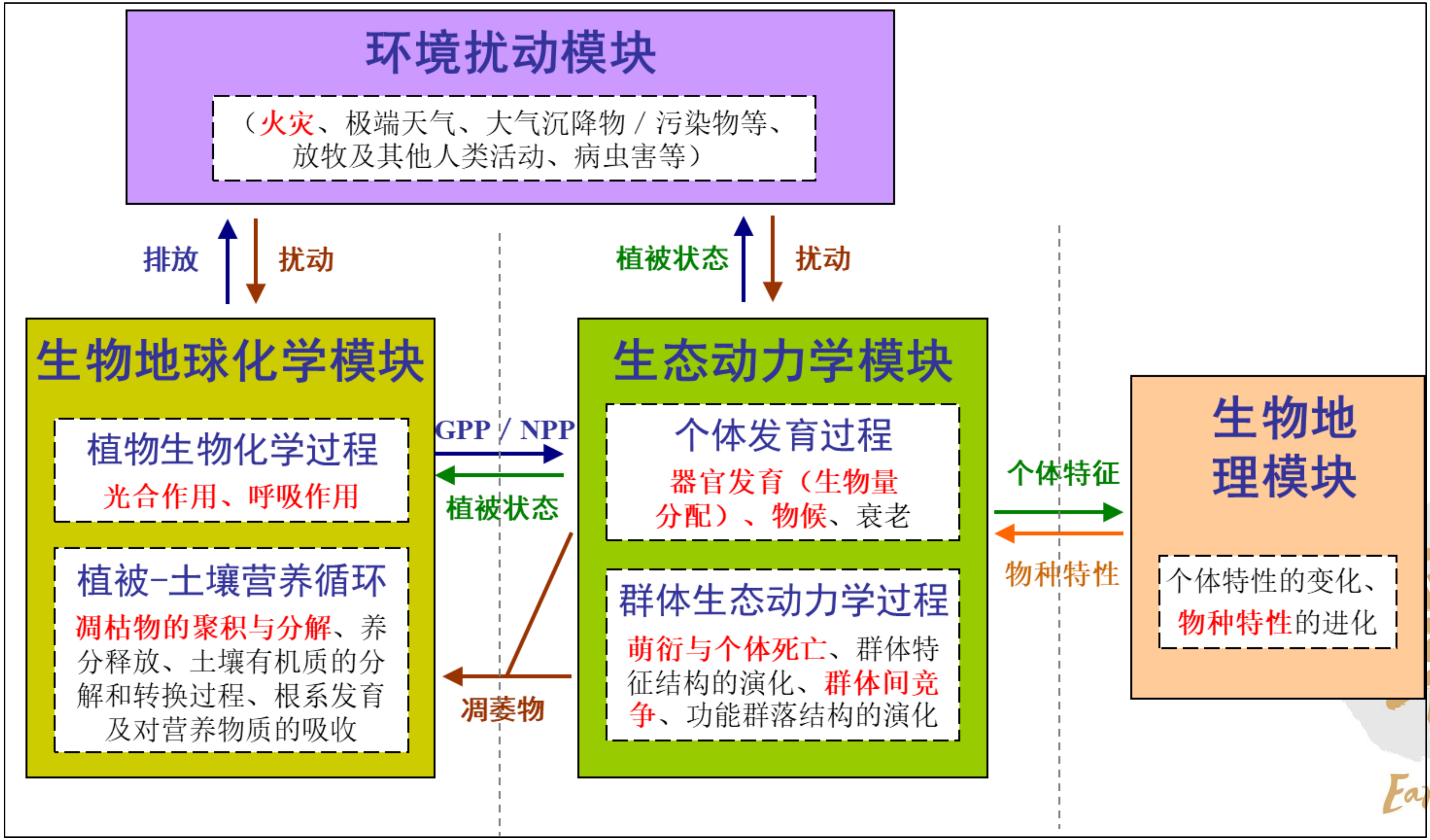


碳循环过程

- 碳库存储：叶（leaves）、根（roots）、茎/边材（stems/sapwood）和心材（heartwood）
- 光合作用：总初级生产力GPP（Gross Primary Production）
- 净初级生产力NPP（Net Primary Production）=GPP – Ra (autotrophic respiration)
- 再生/繁殖（reproduction）、凋落物（litterfall）、死亡（Mortality）、异养呼吸（Heterotrophic respiration）等等

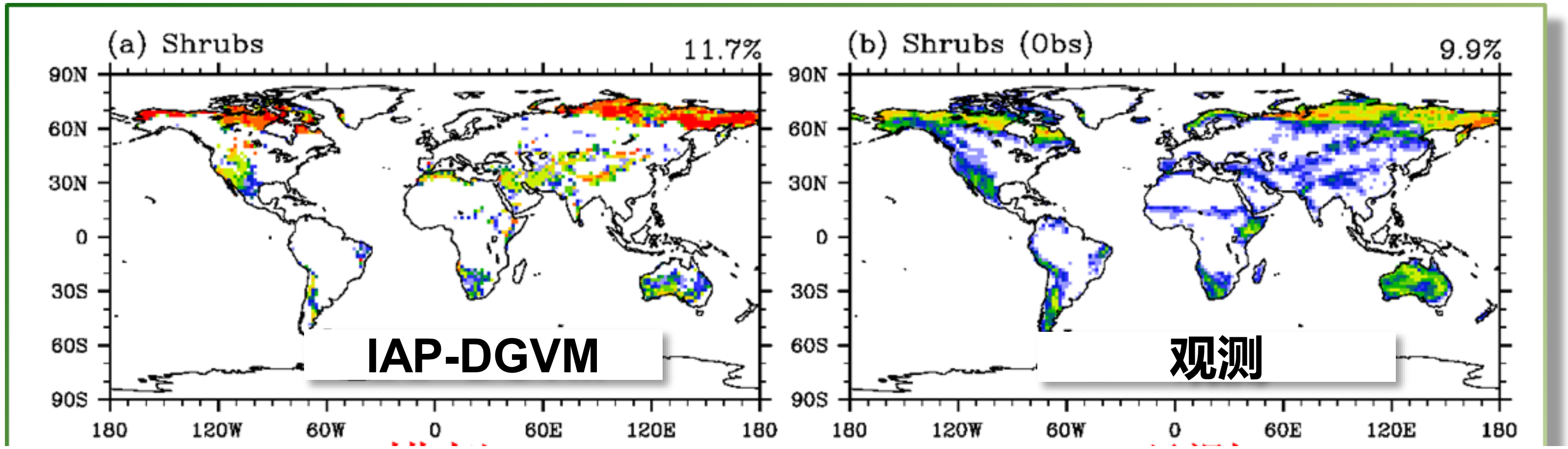
植被动力学模式（IAP-DGVM）分系统介绍

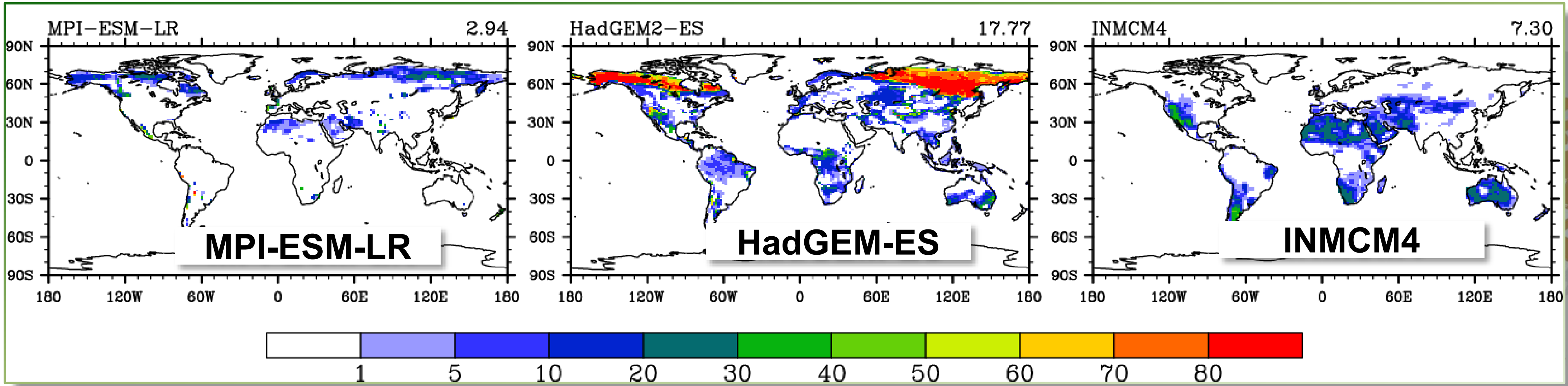
IAP-DGVM设计框图



植被动力学模式（IAP-DGVM）分系统介绍

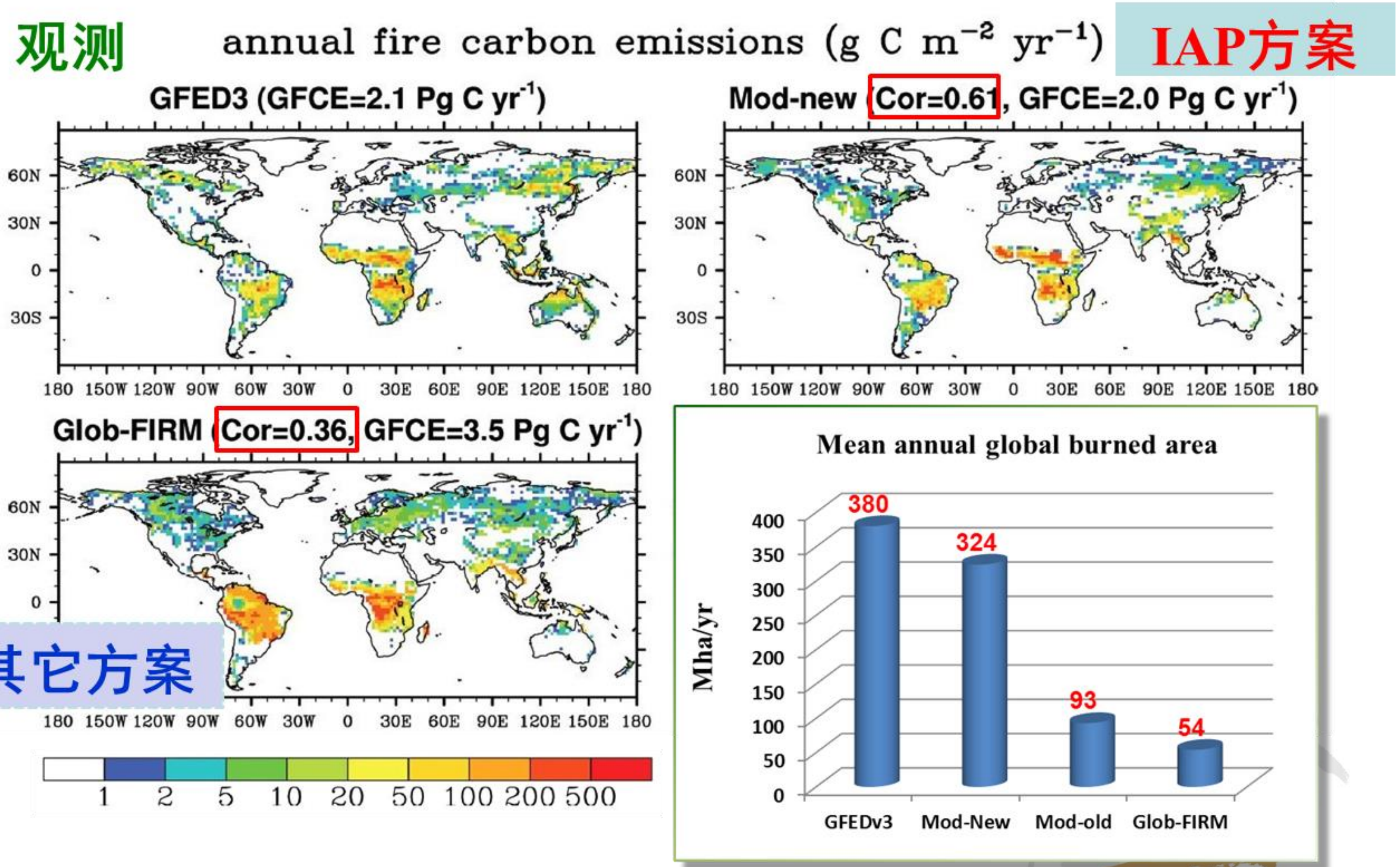
IAP-DGVM的特色创新模块**灌木林子模式（Zeng et al., 2008; 2010）**能准确模拟全球灌木的分布，其方案已被CLM4采用。





植被动力学模式（IAP-DGVM）分系统介绍

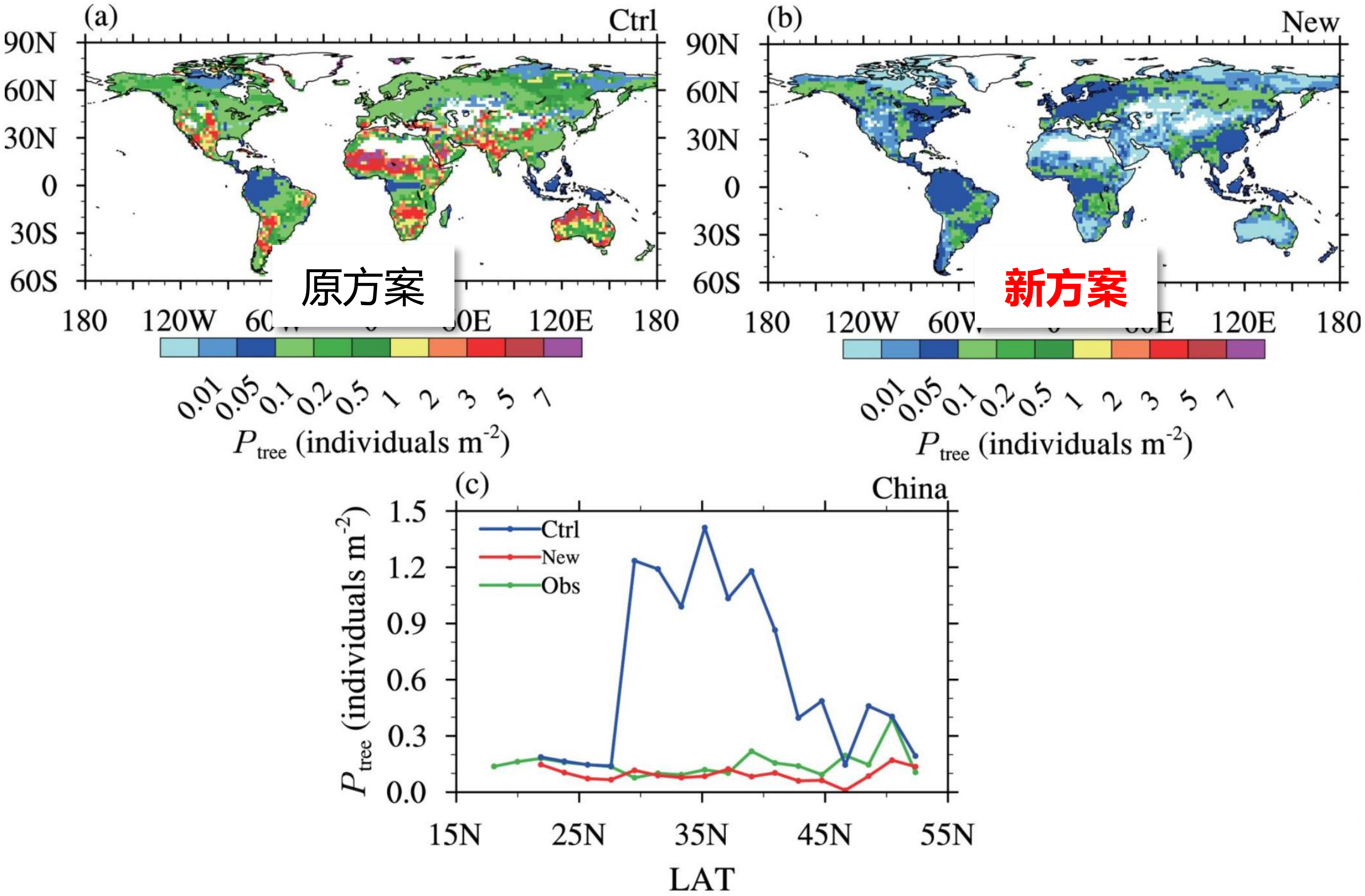
IAP-DGVM的特色创新模块**火干扰参数化方案（Li et al., 2012a; 2012b）**包括火的发生、传播及影响三部分，可模拟火引起的温室气体排放（国际上几乎没有），总排放量更为准确。该方案已被美国NCAR CESM、GFDL-ESM、美国能源部E3SM、加拿大CanESM、BCC-CSM等国内外模式广泛采用，是**国际上最先进**的火模式。



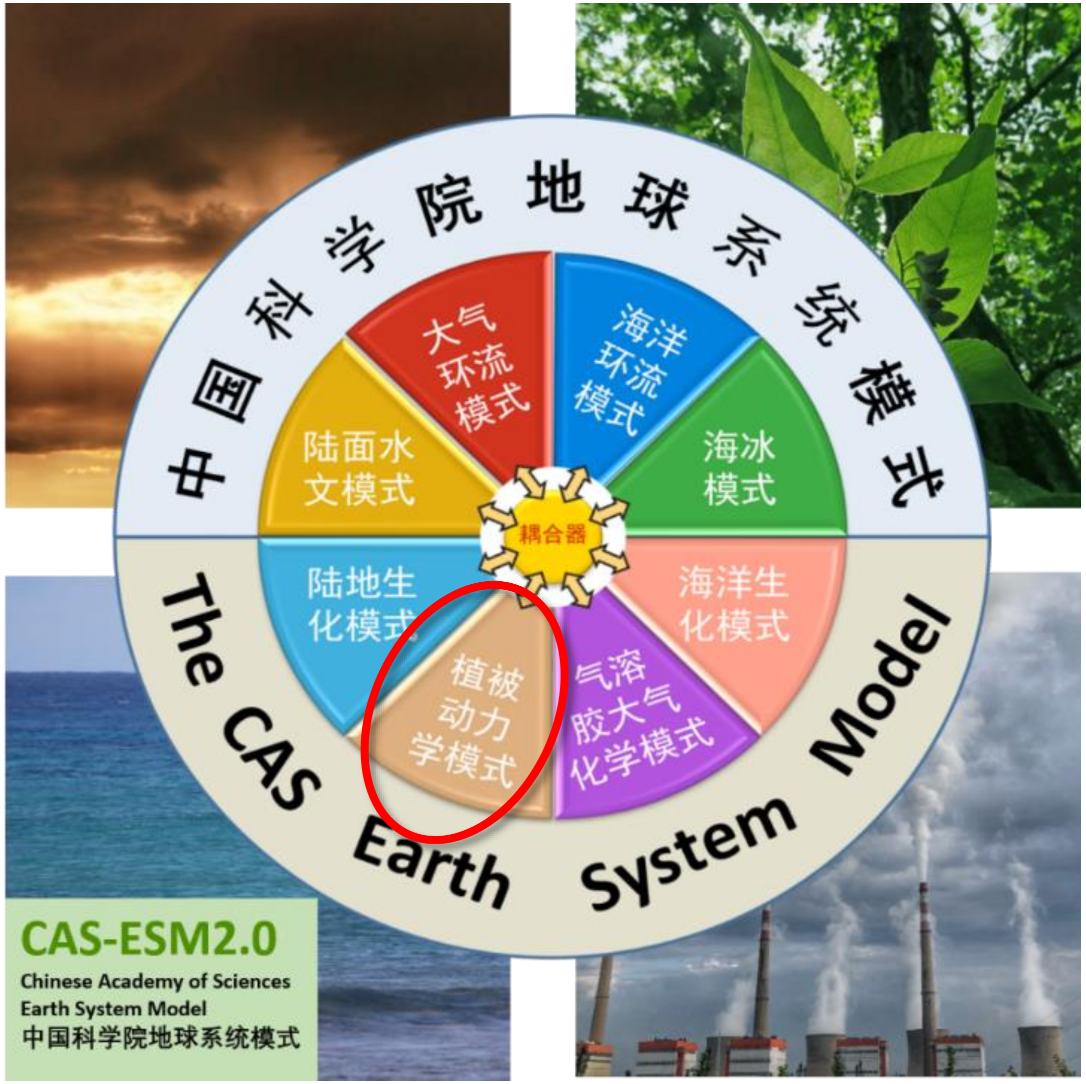
植被动力学模式（IAP-DGVM）分系统介绍

IAP-DGVM特色创新模块**群体动力学萌衍/竞争方案（Song et al.，2016）**显著改进全球森林区的个体数密度等，

从而提升模式的模拟性能；



植被动力学模式（IAP-DGVM）分系统介绍

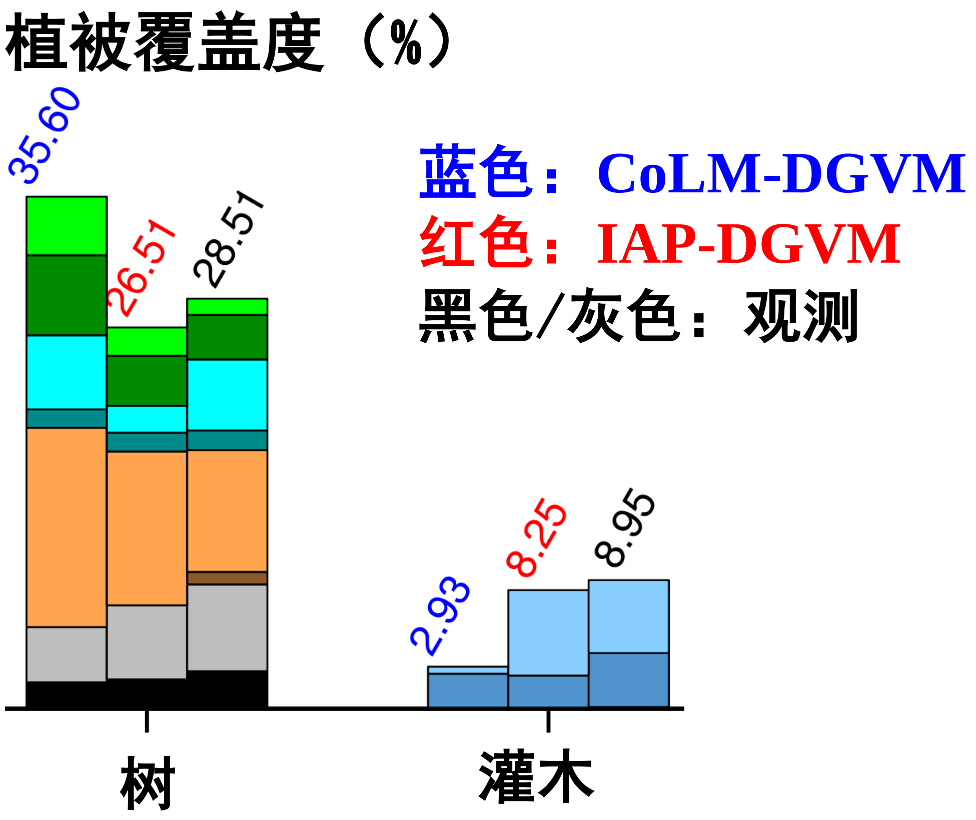


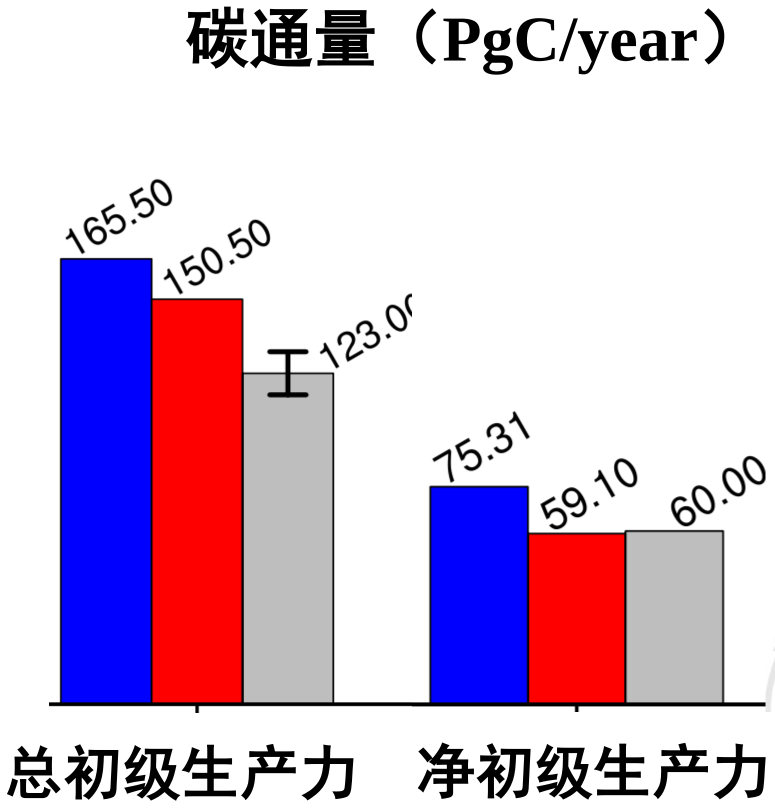
IAP-DGVM已实现与CAS-ESM的耦合，是CAS-ESM的重要组成部分；

分量模式	模式名称	研发单位	参考文献
大气	IAP AGCM5.0	大气所	Zhang et al. (2020)
海洋	LICOM2	大气所	Liu et el. (2012) Jin et al. (2020)
陆面	CoLM	中山大学/北师大	Dai et al. (2003)
海冰	CICE4	LNAL, USA	
大气化学和气溶胶	IAP AACM	大气所	Wei et al. (2019)
动态植被	IAP DGVM	大气所	Zeng et al. (2013) Zhu et al., (2018)
海洋生物地球化学	IAP OBGCM	大气所	Xu et al. (2013)
陆地生物地球化学	CoLM	北师大	Ji and Dai (2013)

植被动力学模式（IAP-DGVM）分系统介绍

IAP-DGVM能够**显著提升**CAS-ESM2对全球植被覆盖度和碳通量的模拟能力，为CAS-ESM2用于研究**全球植被动态过程与气候的相互作用**提供重要工具。





主要内容

参数化原理

IAP-DGVM主要源程序文件

文件名/子程序名	功能
IAP_DGVM.F90	植被动力学模式主程序
subroutine BuildNatVegFilter	子程序：判断自然植被是否存活
subroutine Reproduction	子程序：计算用于种子生产的生物量
subroutine Turnover	子程序：凋萎
subroutine Allocation	子程序：碳分配及形态发育
subroutine Light	子程序：光竞争
subroutine Mortality	子程序：生长胁迫和高温胁迫
subroutine Fire	子程序：火干扰
subroutine Establishment	子程序：萌衍
subroutine Kill	子程序：移除死亡的PFT
subroutine ResetBareFPC	子程序：更新裸土覆盖度
IAP_DGVM_daily.F90	植被动力学模式日尺度更新程序
subroutine Phenology	子程序：物候
subroutine Respiration	子程序：自养呼吸
subroutine LitterSOM	子程序：凋落物及土壤有机碳
IAPFireseason.F90	火模块主程序

参数化原理-自养呼吸（DGVMRespiration.F90）

- 目录：CAS-ESM2/src/models/lnd/colm/src/mainc
- 功能：得到自养呼吸和净初级生产力

Net primary production is defined as plant photosynthesis, A (see section 8, Oleson *et al.* (2004)) minus autotrophic respiration, R_a, where R_a is given by:

$$R_a = R_g + R_m \qquad \text{(Eq. 1)}$$

where R_g is growth respiration and R_m is the sum of maintenance respiration for leaves, R_{leaf}, sapwood, $R_{sapwood}$, and roots, R_{root}:

$$R_g = 0.25\,(A - R_m) \qquad \text{(Eq. 2)}$$

$$R_m = R_{leaf} + R_{sapwood} + R_{root} \qquad \text{(Eq. 3)}$$

$$R_{leaf} = r \cdot k \frac{C_{leaf}}{\mathrm{cn}_{leaf}} \phi \cdot g(T) \cdot \frac{2\times10^6 P}{28.5 \cdot \mathrm{FPC}} \qquad \text{(Eq. 4)}$$

$$R_{root} = r \cdot k \frac{C_{root}}{\mathrm{cn}_{root}} \phi \cdot g(T) \cdot \frac{2\times10^6 P}{28.5 \cdot \mathrm{FPC}} \qquad \text{(Eq. 5)}$$

$$R_{sapwood} = r \cdot k \frac{C_{sapwood}}{\mathrm{cn}_{sapwood}} \cdot g(T) \cdot \frac{2\times10^6 P}{28.5 \cdot \mathrm{FPC}} \qquad \text{(Eq. 6)}$$

Net primary production or dry biomass increment, Δm (μg biomass m^{-2} pft area s^{-1}), summed annually gives annual net primary production, NPP (g C m^{-2} pft area):

$$\Delta m = 28.5\,(A - R_a) \qquad \text{(Eq. 8)}$$

$$\mathrm{NPP} = \sum_{beginning}^{endofyear} \Delta m \cdot \Delta t \cdot 0.5\times10^{-6} \qquad \text{(Eq. 9)}$$

```
      frm   = frmf + frms + frmr

      ! growth respiration and production

      frg = 0.25 * max(assim - frm, 0.0)       !changed to ma
      respc = frm + frg
      ! npp gC/m2 vegt'd area /step
      dnpp = (assim - respc) * dmcf * 0.5 * dtime * fpcgrid
```

```
      frmf = respcoeff * k * lm_ind * nind / l_cton &
           * tf1 * dphen * 2.0 / dmcf / fpcgrid
      frms = respcoeff * k * sm_ind * nind / s_cton &
           * tf1 * 2.0 / dmcf / fpcgrid
      frmr = respcoeff * k * rm_ind * nind / r_cton &
           * tf2 * dphen * 2.0 / dmcf / fpcgrid
```

参数化原理-种子生物量（subroutine Reproduction）

- 目录：CAS-ESM2/src/models/lnd/colm/src/iap-dgvm/IAP_DGVM.F90
- 功能：计算用于种子生产的生物量

The cost of reproduction, ΔC_{reprod}, is assumed to be a constant fraction of annual net primary production, NPP. This fraction is set to 0.1 for all pfts (Sitch *et al.* 2003). Reproductive structures enter above ground litter, $C_{\text{L,ag}}$, directly. NPP is reduced by the same amount:

$$C_{\text{L,ag}} = C_{\text{L,ag}} + \Delta C_{\text{reprod}} \quad \text{(Eq. 11)}$$

$$\text{NPP} = \text{NPP} - \Delta C_{\text{reprod}} \quad \text{(Eq. 12)}$$

```fortran
   real(r8), parameter :: reprod_cost = 0.1         ! proportion of NPP lost tc
   integer  :: p, fp                                ! pft index
   real(r8) :: reprod                               ! temporary
!-----------------------------------------------------------------------

   ! Compute reproduction costs

   do fp = 1,num_natvegp
      p = filter_natvegp(fp)

      ! Calculate allocation to reproduction
      ! Reproduction costs taken simply as a constant fraction of annual NPP

      reprod = max(bm_inc(p) * reprod_cost, 0.0)

      ! assume the costs go to reproductive structures which will
      ! eventually enter the litter pool

      litter_ag(p) = litter_ag(p) + reprod

      ! Reduce biomass increment by reproductive cost

      bm_inc(p) = bm_inc(p) - reprod
   end do
```

参数化原理-凋萎（subroutine Turnover）

- 目录：CAS-ESM2/src/models/lnd/colm/src/iap-dgvm/IAP_DGVM.F90
- 功能：计算生物量的凋萎转化

The amount of living carbon that enters the above and below ground litter pools and the amount of sapwood that turns to heartwood annually are calculated given pft-

$$\Delta C_{\mathrm{turn}} = C_{\mathrm{leaf}} f_{\mathrm{leaf}} + C_{\mathrm{sapwood}} f_{\mathrm{sapwood}} + C_{\mathrm{root}} f_{\mathrm{root}}$$

$$\Delta C_{\mathrm{heartwood}} = C_{\mathrm{sapwood}} f_{\mathrm{sapwood}}$$

$$\Delta C_{\mathrm{L,ag}} = C_{\mathrm{leaf}} f_{\mathrm{leaf}} P$$

$$\Delta C_{\mathrm{L,bg}} = C_{\mathrm{root}} f_{\mathrm{root}} P$$

```
      ! Calculate the biomass turnover in this year

      lm_turn = lm_ind(p) * l_torate
      sm_turn = sm_ind(p) * s_torate
      rm_turn = rm_ind(p) * r_torate

      ! Update the pools

      lm_ind(p) = lm_ind(p) - lm_turn
      sm_ind(p) = sm_ind(p) - sm_turn
      rm_ind(p) = rm_ind(p) - rm_turn

      ! Convert sapwood to heartwood

      hm_ind(p) = hm_ind(p) + sm_turn

      ! Transfer to litter pools

      litter_ag(p) = litter_ag(p) + lm_turn * nind(p)
      litter_bg(p) = litter_bg(p) + rm_turn * nind(p)
```

参数化原理-碳分配（subroutine Allocation）

- 目录：CAS-ESM2/src/models/lnd/colm/src/iap-dgvm/IAP_DGVM.F90
- 功能：计算碳分配（叶、茎、根）及形态发育

1) Leaf area, LA (m^2 of leaves per individual), is proportional to the sapwood cross sectional area, SA (m^2 of sapwood cross sectional area per individual), according to the "pipe model" (Shinozaki *et al.* 1964a,b; Waring *et al.* 1982):

$$\mathrm{LA} = k_{\mathrm{la:sa}} \cdot \mathrm{SA} \qquad \text{(Eq. 22)}$$

2) Leaf mass, C_{leaf} (g C per individual), is proportional to root mass, C_{root} (g C per individual), with greater allocation to roots as water stress increases:

$$C_{\mathrm{leaf}} = \mathrm{lr}_{\mathrm{max}}\, \omega\, C_{\mathrm{root}} \qquad \text{(Eq. 23)}$$

where $\mathrm{lr}_{\mathrm{max}}$ is the ratio of leaf to root mass assuming unlimited soil water (Table 2) and ω (fraction ranging from 0 to 1) is the ratio of actual annual gross primary production to the potential annual production assuming unlimited soil water. Annual gross primary

3) Height, H (m), and crown area, CA (m^2), are allometric functions of stem diameter, D (m):

$$H = k_{\mathrm{allom2}} D^{k_{\mathrm{allom3}}} \qquad \text{(Eq. 24)}$$

$$\mathrm{CA} = k_{\mathrm{allom1}} D^{k_{\mathrm{rp}}} \qquad \mathrm{CA} \le 15\ \mathrm{m}^2 \qquad \text{(Eq. 25)}$$

where $k_{\mathrm{allom1}} = 100$, $k_{\mathrm{allom2}} = 40$, $k_{\mathrm{allom3}} = 0.5$, and $k_{\mathrm{rp}} = 1.6$.

```fortran
         hm = hm_ind(p)
         csum = bm_inc_ind(p) + lm_ind(p) + rm_ind(p) + sm_ind(p)

         k_lasa = latosa(ivt(p))
         ksla = sla(p)
         rou = wooddens
         k_a2 = allom2(ivt(p))
         k_a3 = allom3(ivt(p))
         lmtorm(p) = init_lmtorm(p) * wscal
         k_rl = 1.0 / lmtorm(p)

         lm0 = 0.0
         lm1 = csum / (1.0 + k_rl)     !give a guessed value of above ground cpool.
         err_hm = eps_alloc + eps_alloc                        !
         n_step = 0
#ifdef DGVM_BUGON
         print *, 'wscal', wscal, annpsn(p), annpsnpot(p), k_rl, lm1, csum
#endif

       ! Avoid unbelievable value for very small lm. Modified by zhq @09/14/2010
       ! do while ((lm1 - lm0) > eps_alloc .and. (err_hm > eps_alloc .or. err_hm < -eps_alloc) .and. n_step < 25)
         do while ((lm1 - lm0) > eps_alloc .or. (err_hm > eps_alloc .or. err_hm < -eps_alloc) .and. n_step < 25)
            lm = (lm0 + lm1) * 0.5
            rm = k_rl * lm
            sm = csum - lm - rm
            sa = lm * ksla / k_lasa
            hd = sm / (rou * sa)
            dd = (hd/k_a2) ** (1.0/k_a3)
            vd = PI * .25 * dd * dd * hd;
            err_hm = rou * vd - sm - hm;
            if (err_hm > 0) then
               lm0 = lm
            else
               lm1 = lm
            end if
```

参数化原理-光竞争（subroutine Light）
）

- 目录：CAS-ESM2/src/models/lnd/colm/src/iap-dgvm/IAP_DGVM.F90
- 功能：计算植被光竞争

竞争等级： 树>灌木>草

First, the fractional projective cover calculated in section 2.6 summed over all tree pfts, FPC_{woody}, is limited to 95% of the naturally vegetated landunit. Excess tree cover, FPC_{excess}, is removed from each tree pft in proportion to the pft's FPC increment, ΔFPC_{pft}:

$$FPC_{excess} = (FPC_{woody} - 0.95)\frac{\Delta FPC_{pft}}{\Delta FPC_{woody}} \quad \text{(Eq. 30)}$$

The amounts of leaf, sapwood, heartwood, and root carbon (C_{leaf}, $C_{sapwood}$, $C_{heartwood}$, C_{root}) corresponding to FPC_{excess} are transferred to above and below ground litter, $C_{L,ag}$ and $C_{L,bg}$. The population density, P, of trees is adjusted accordingly, representing a self-thinning process due to finite space in a grid cell:

$$P_{excess} = \frac{P \cdot FPC_{excess}}{FPC} \quad \text{(Eq. 32)}$$

$$P = P - P_{excess} \quad \text{(Eq. 33)}$$

$$C_{L,ag} = C_{L,ag} + P_{excess}\,(C_{leaf} + C_{sapwood} + C_{heartwood}) \quad \text{(Eq. 34)}$$

$$C_{L,bg} = C_{L,bg} + P_{excess}\, C_{root} \quad \text{(Eq. 35)}$$

```
fpc_shrub_max = 1.0 - min(fpc_tree_total, fpc_tree_max)
fpc_grass_max = max(0.0, fpc_shrub_max - fpc_shrub_total)

if (fpc_tree_total > fpc_tree_max) then

   if (fpc_inc_tree > 0.0) then
      excess = (fpc_tree_total - fpc_tree_max) * &
                  fpc_inc(p) / fpc_inc_tree
   else
    ! excess = (fpc_tree_total - fpc_tree_max) / &
    !             real(ntree)
      excess = (fpc_tree_total - fpc_tree_max) * &
                  fpcgrid(p) / fpc_tree_total
   end if
```

```
nind_kill = nind(p) * excess / fpcgrid(p)
nind(p) = nind(p) - nind_kill

! Transfer lost biomass to litter

litter_ag(p) = litter_ag(p) + nind_kill * (lm_ind(p) + sm_ind(p) + hm_ind(p))
litter_bg(p) = litter_bg(p) + nind_kill * rm_ind(p)
fpcgrid(p) = crownarea(p) * nind(p)
```

参数化原理-死亡（subroutine Mortality）

- 目录：CAS-ESM2/src/models/lnd/colm/src/iap-dgvm/IAP_DGVM.F90
- 功能：计算生长胁迫和高温胁迫

For each tree pft, a fraction of individuals is removed and converted to litter every year due to background mortality, $mort_{greff}$, and mortality due to heat stress, $mort_{heat}$:

$$mort_{greff} = \frac{k_{mort1}}{1 + k_{mort2} \cdot greff} \qquad \text{(Eq. 41)}$$

$$mort_{heat} = \frac{GDD_{23°C}}{300} \qquad 0 \le mort_{heat} \le 1 \qquad \text{(Eq. 42)}$$

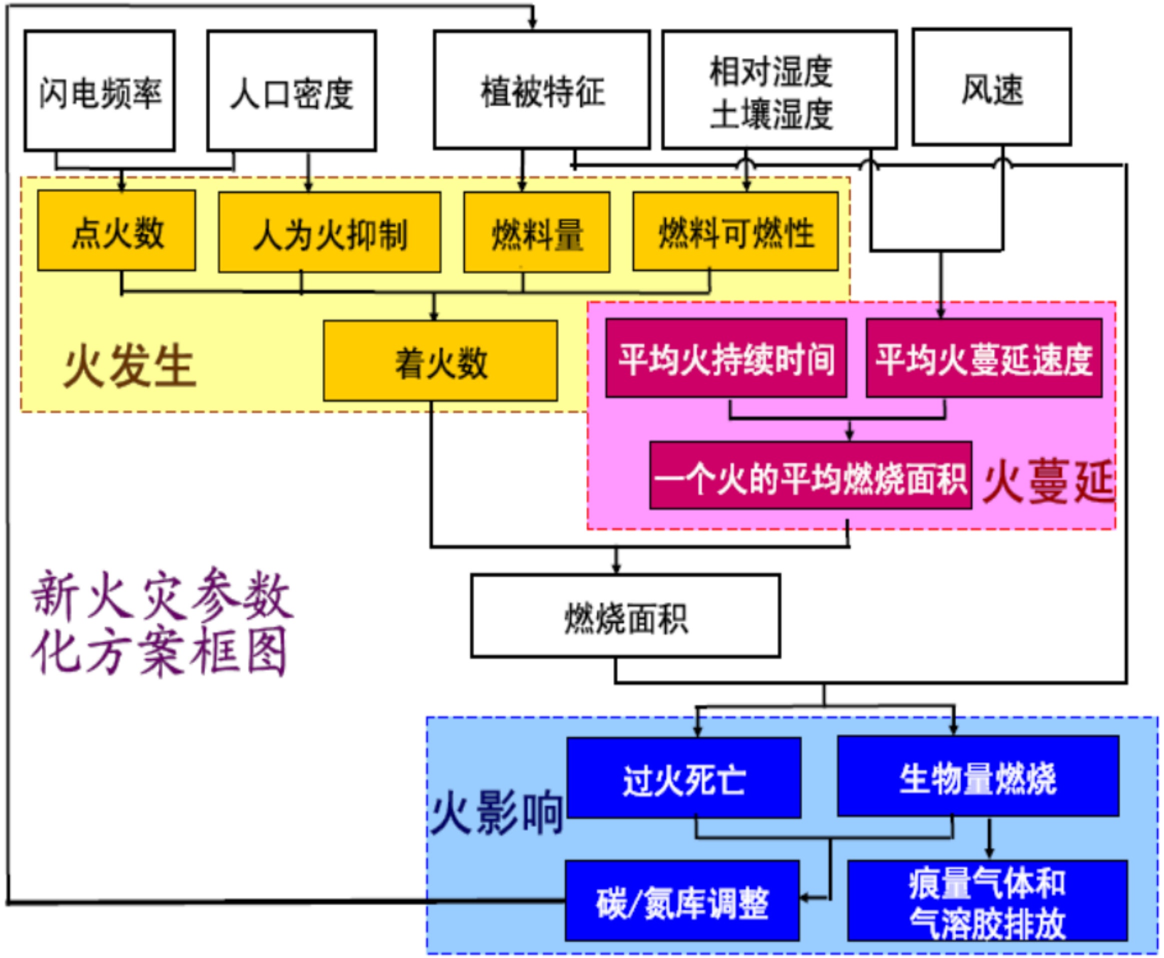
$$mort = mort_{greff} + mort_{heat} \qquad 0 \le mort \le 1 \qquad \text{(Eq. 43)}$$

$$P = P - P \cdot mort \qquad \text{(Eq. 44)}$$

```
heatstress = min(1.0, agddtw(p) / ramp_agddtw)

! Calculate net individual living biomass increment
bm_delta = max(0.0, bm_inc(p) / nind(p) - turnover_ind(p))

! Calculate growth efficiency (net biomass increment per un

!BUGFIX
if (lm_ind(p) == 0) then
   greffic = 0.
else
   greffic = bm_delta / lm_ind(p) / sla(p)
end if
!print *, ivt(p),'greffic', greffic
!BUGFIX

! Mortality rate inversely related to growth efficiency (Pr
mort = mort_max / (1.0 + k_mort * greffic)

! Reduce individual density (=> gridcell-level biomass) by
mort = min(1.0, mort + heatstress)
nind_kill = nind(p) * mort
nind(p) = nind(p) - nind_kill
```

参数化原理-火模块（subroutine Fire）

- 目录：CAS-ESM2/src/models/lnd/colm/src/iap-dgvm/IAP_DGVM.F90、IAPFireseason.F90
- 功能：火干扰过程



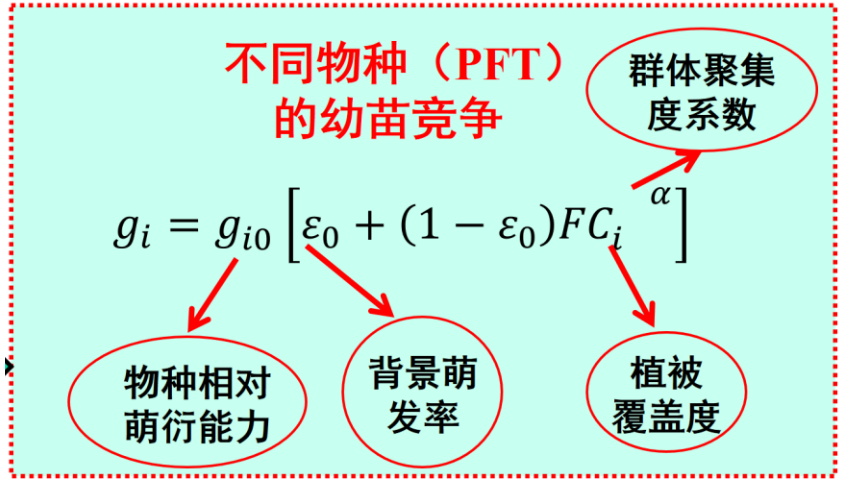

```
! ********fire occurrence**********
       ign=forc_i/3600.0*dtime
       fb=max(0.0,min(1.0,(agb_col-lfuel)/(ufuel-lfuel)))
       nfire=ign*fb*fm_col*(1.0-max(0.0,min(1.0,(min(max(forc_rh,0.0),1.0)-0.3)/(0.7-0.3))))

      ! ********fire spread **********
      spread_m=fm_col1*(1.0-max(0.0,min(1.0,(min(max(forc_rh,0.0),1.0)-0.3)/(0.7-0.3))))
      fd_lf=24*3600
      Lb_lf=1.0+10.0*(1.0-EXP(-0.06*forc_wind))
      baf_lf=(g0*spread_m*fa_col*fd_lf/1000)**2*nfire*PI*Lb_lf
      afirefrac1=afirefrac1+baf_lf

!============fire carbon release zhujw 2021-11-19=======
if (ivt .le. 16) then
  baf_lf=min(baf_lf,1.0)
! combustion first
      if(ivt==1.or.ivt==2.or.ivt==4.or.ivt==9)then !  evergreen vegwhose leaf turnover rate is 1/2
         lm_fire = lm_ind*ccl*baf_lf
      end if

      vc = vegclass
      if (vc==1.or.vc==2) then          !trees and shrubs
         sm_fire = sm_ind*(ccst*2.0)* baf_lf
         hm_fire = hm_ind*(ccst/4.0)* baf_lf
      end if
      rm_fire = rm_ind*ccr*baf_lf
      la_fire = ccd* baf_lf* litter_ag
      acflux_fire_step=(lm_fire+sm_fire+hm_fire+rm_fire)*nind+la_fire
end if
```

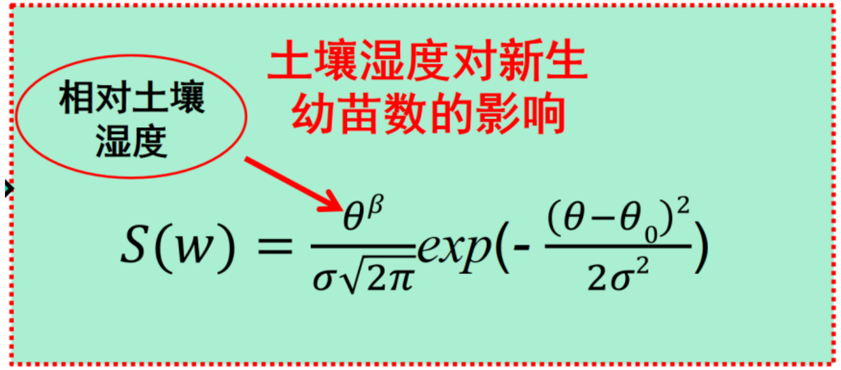

Li et al. (2012, Biogeosciences)   Li et al. (2017, J. Climate)

参数化原理-萌衍（subroutine Establishment）

- 目录：CAS-ESM2/src/models/lnd/colm/src/iap-dgvm/IAP_DGVM.F90
- 功能：种子萌发过程

$$\Delta P_i = \Delta P_{max}\left(1 - F_{woody}\right)\left[1 - e^{-5(1-F_{woody})}\right]\frac{g_i}{\sum_{k=1}^{n_{est}} g_k} S(w)$$






```
soil_liq_3lys=grid_wliq6mon**2/(sqrt(2.0*PI)*0.4)*exp(-(grid_wliq6mon-1.0)**2/(2*0.4**2))

if (grid_prec365 >= prec_min_estab .and. npft_estab_tree > 0) then
  ! Calculate establishment rate over available space, per tree PFT
  ! Maximum establishment rate reduced by shading as tree FPC approaches 1
  ! Total establishment rate partitioned equally among regenerating woody PFTs
   if (fpc_woody_total < 1.0) then     ! new test at Aug 5, 2008, 2:43pm,try to avoid negetive nind

    estab_rate(p) = estab_max *soil_liq_3lys*(1.0-exp(5.0*(fpc_woody_total-1.0))) * est_pot_tree(p) /est_sum_tree
  ! Calculate grid-level establishment rate per woody PFT
  ! Space available for woody PFT establishment is proportion of grid
  ! cell ! not currently occupied by woody PFTs
        estab_grid(p) = estab_rate(p) * (1.0-fpc_woody_total)
```

Song et al. (2016, AAS)

参数化原理-异养呼吸（subroutine LitterSOM）

- 目录：CAS-ESM2/src/models/lnd/colm/src/mainc/DGVMLitterSOM.F90
- 功能：异养呼吸过程

```
fmicr = cflux_litter_atmos + cflux_fast_atmos + cflux_slow_atmos
afmicr = afmicr + fmicr
fmicr = fmicr * 2.0 / dmcf / dtime
```

fmicr = **cflux_litter_atmos** + **cflux_fast_atmos** + **cflux_slow_atmos**

1，**凋落物的分解：cflux_litter_atmos** ! litter decomposition flux to atmosphere

2，**土壤碳库的分解：cflux_fast_atmos （快过程）** + **cflux_slow_atmos（慢过程）**

cflux_fast_atmos ! soil fast pool decomposition flux to atmos.
cflux_slow_atmos ! soil slow pool decomposition flux to atmos.

参数化原理-异养呼吸（subroutine LitterSOM）

- 目录：CAS-ESM2/src/models/lnd/colm/src/mainc/DGVMLitterSOM.F90
- 功能：异养呼吸过程

1，凋落物的分解：cflux_litter_atmos

```
! Calculate carbon flux to atmosphere and soil

cflux_litter_atmos = atmfrac  * litter_decom
```

1，地上和地下凋落物的总量
2，根的分布
3，土壤温度
4，土壤含水量

```
atmfrac = 0.7_r8
```

```
do j = 1, 10
   litter_decom_bg_layer(j) = litterbg * rootfr(j) * (1.0_r8-exp(-k_litter(j)))
end do

litter_decom_bg = sum(litter_decom_bg_layer)
```

```
litter_decom    = litter_decom_ag + litter_decom_bg
```

```
litter_decom_ag = litterag * (1.0_r8-exp(-k_litter(1)))
```

地上凋落物

DGVMtimevar.F90

```
do j = 1,nl_soil
   porslsum = porsl(j)*dz(j)
   wliqsum  = wliq(j)/denh2o + wice(j)/denice

   if(porslsum .gt. 0.)then
      wf(j) = min(wliqsum/porslsum,1.0)
   else
      wf(j) = 0.
   end if
```

```
k_litter(j) = k_litter10   * temp_resp(j) * moist_resp(j) * dtime / (86400._r8 * 365._r8)
        k_litter10 = 0.35_r8
```

```
temp_resp(j)=exp(308.56_r8*((1.0_r8/56.02_r8)-(1.0_r8/(tss(j)-227.13_r8))))
```

```
moist_resp(j) = 0.25_r8 + 0.75_r8 * wf(j)
```

土壤含水量

土壤温度

```
tss(nl_soil)     ! soil temperature to 0.25 m (Kelvin)
```

```
wf(nl_soil)      ! soil water as frac. of whc
```

参数化原理-异养呼吸（subroutine LitterSOM）

➢ 目录：CAS-ESM2/src/models/lnd/colm/src/mainc/DGVMLitterSOM.F90

➢ 功能：异养呼吸过程

2，土壤碳库的分解：cflux_fast_atmos （快过程） + cflux_slow_atmos（慢过程）

```
cflux_fast_atmos = sum(cflux_fast_atmos_layer)
cflux_slow_atmos = sum(cflux_slow_atmos_layer)
```

```
cflux_fast_atmos_layer(j) = cpool_fast(j) * (1.0_r8-exp(-k_fast(j)))
cflux_slow_atmos_layer(j) = cpool_slow(j) * (1.0_r8-exp(-k_slow(j)))
```

```
cpool_fast(j) = cpool_fast(j) + fastfrac * cflux_litter_soil_layer(j)
cpool_slow(j) = cpool_slow(j) + slowfrac * cflux_litter_soil_layer(j)
```

```
fastfrac = 0.985_r8
slowfrac = 1.0_r8 - fastfrac
```

碳库量

```
INTENT(inout) :: cpool_fast(nl_soil)        ! fast carbon pool
INTENT(inout) :: cpool_slow(nl_soil)        ! slow carbon pool
```

```
atmfrac = 0.7_r8
soilfrac = 1.0_r8 - atmfrac
```

```
do j = 1, 10
   cflux_litter_soil_layer(j) = soilfrac * litter_decom_bg_layer(j)
end do
cflux_litter_soil_layer(1) = cflux_litter_soil_layer(1) + soilfrac * litter_decom_ag
```

```
litter_decom_ag = litterag * (1.0_r8-exp(-k_litter(1)))
```

地上凋落物

```
do j = 1, 10
   litter_decom_bg_layer(j) = litterbg * rootfr(j) * (1.0_r8-exp(-k_litter(j)))
end do
```

地下凋落物

根的比例

```
INTENT(in) :: rootfr(nl_soil)   ! fraction of roots in each soil layer
```

1，土壤碳库量

2，地下凋落物的量

3， 根的分布

4，土壤温度和土壤含水量

主要内容

运行及后处理

- ./create_newcase -case test1 -compset B1850C5X -res fd14_licom -mach bigdata_intel
- ./test1.xxx.build
- Submit the job

./create_newcase -help 会列出其应用的所有选项，其中case, res,compset,mach一定要指定；
-list <compsets or grids or machines>
./create_clone －clone xxx －case xxx

运行create_newcase 后常用的主要文件：
env_mach_pes.xml： 设置cpu个数

env_run.xml ： 设置运行时间，restart频率

SourceMods ： 修改模式代码时使用

要在build之前才有效！

需要rebuild之前，要*.clean_build

```
src.cam
src.cice
src.colm
src_colm_new
src_colm_old
src.drv
src.licom
src.sglc
src.share
```

```
#define COUP_CSM
#undef  CPL6
#define CPL7
#undef  USGS
#undef  DyN
#define PFT
#define DGVM
#define RTM
#define SPMD
#define CMIP
#undef VEGDATA
#undef BNUDGVM
#define SOIL15
#define IAPDGVM
#undef FHNP
#undef FTF
#undef NP
#undef GWUSE
#define DUST
```

运行及后处理

lnd_in

```
&clmexp
fsrf           = 'CoLM-srf-IAP-CMIP-128x256'
!fini           = 'B1850C5X_IAPDGVM-colm-restart
!fsbc           = 'B1850C5X_IAPDGVM-colm-restart
fdust          = 'CoLM-surf-dust-128x256'
fnig           = 'ig2004.nc'
startup_date   = 1,1,0
frivinp_rtm    = 'rdirc.05'
lnd_cflux_year = 1
co2_type       = 'prognostic'
co2_ppmv       = 284.7
lhist_yearly   = .true.
lhist_monthly  = .true.
lhist_daily    = .false.
lhist_3hourly  = .false.
lon_points     = 256
lat_points     = 128
dtime          = 1800
/
```

```
B1850C5X_IAPDGVM_CO2C-colm-0450-01.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-02.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-03.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-04.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-05.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-06.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-07.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-08.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-09.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-10.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-11.nc
B1850C5X_IAPDGVM_CO2C-colm-0450-12.nc
B1850C5X_IAPDGVM_CO2C-colm-0450.nc
```

月输出变量

```
float gpp(time, lat, lon) ;
        gpp:long_name = "carbon mass flux out of atmosphere due to GPP on land" ;
        gpp:units = "kg/m^2/s" ;
        gpp:_FillValue = -9999.f ;
float npp(time, lat, lon) ;
        npp:long_name = "carbon mass flux out of atmosphere due to NPP on land" ;
        npp:units = "kg/m^2/s" ;
        npp:_FillValue = -9999.f ;
```

年输出变量

```
float pftFrac(time, pft, lat, lon) ;
        pftFrac:long_name = "faction of each PFT on gridlevel" ;
        pftFrac:units = "percent" ;
        pftFrac:_FillValue = -9999.f ;
```

运行及后处理

1) NetCDF : ncdump ***.nc | less

2) NCO :

ncra	file1.nc	file2.nc	avgfile.nc		(平均)
ncrcat	file1.nc	file2.nc	out12.nc		(整合)
ncea	file1.nc	file2.nc	file3.nc	out.nc	(集合平均)
ncdiff	file1.ncl	file2.nc	diff.nc		(差异)
ncks	-v T, U	in.nc	out.nc		(只输出T,U)
ncrcat	-x -v T,U	file1.nc	file2.nc	out.nc	(除了T,U)

NCO homepage：

http://nco.sourceforge.net

Reference Manual:

http://nco.sourceforge.net/nco.html#Operator-Reference-Manual

3) ncview file.nc

4) ImageMagick (display, convert)

convert – adjoin – delay 30 TLAI*.ps TLAI.gif

5) 诊断包 (Diagnostics Packages)

运行及后处理-NCL

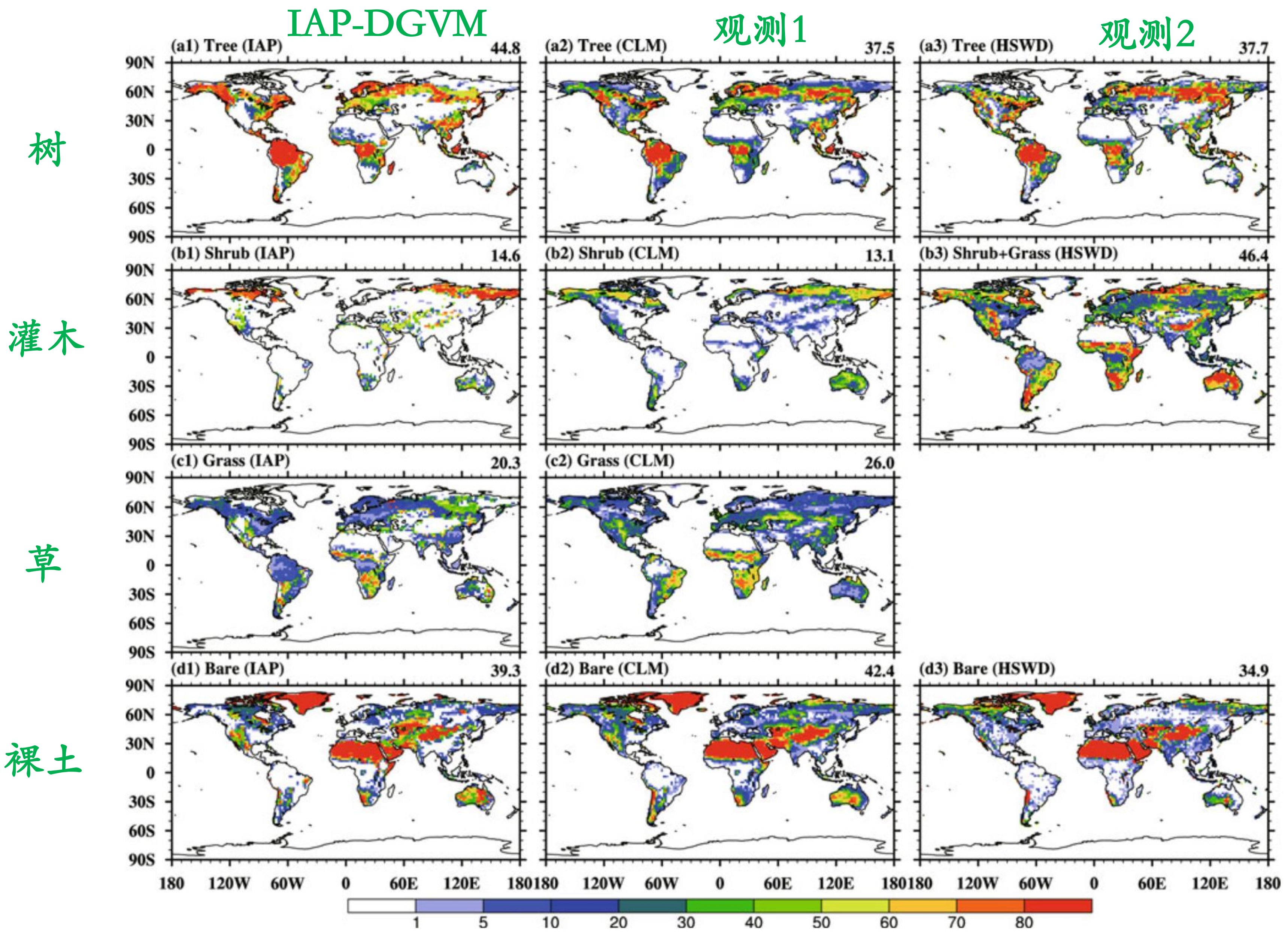
```
yr0=470
yr1=483
nyr=yr1-yr0+1
do i=yr0,yr1
   t=i-yr0
   yr=sprinti("%0.4i",i)
    datas=output+case+"/run/"+case+"-colm-"+yr+".nc"
    datas_in=addfile(datas,"r")
    bare(t,:,:) =datas_in->bare(0,:,:)
    tree(t,:,:) =datas_in->treeFrac(0,:,:)
    shrub(t,:,:)=datas_in->shrubFrac(0,:,:)
    grass(t,:,:)=datas_in->grassFrac(0,:,:)
    fc_pft(t,:,:,:)=datas_in->pftFrac(0,:,:,:)
    delete(datas_in)
end do
```

```
path_d="/public/home/zhujiawen/work/CAS-ESM2/output/"
path_f="/public/home/zhujiawen/work/ncl/B1850_IAPDGVM/picture/"
;case="I4804_qian_fd14"
;case="BNUDGVM_I4804_qian_fd14"
;case="IAPDGVM_I4804_qian_fd14"
;case="IAPDGVM_test"
case="B1850C5X_IAPDGVM_CO2C"
;case="B1850C5X_IAPDGVM_CO2C"
;case="I1850"
;case="B20TR_IAPDGVM_CO2A"
;run ="run_y100"
run ="run_150"
;========= run FC_eq.ncl =================
yr0 =380
yr1 =400
system("export path_d="+path_d+"; export path_f="+path_f+"; export case="+case+";export run="+run+"; export yr0="+yr0+"; export yr1="+yr1+"; ncl FC_eq.ncl")
system("export path_d="+path_d+"; export path_f="+path_f+"; export case="+case+";export run="+run+"; export yr0="+yr0+"; export yr1="+yr1+"; ncl FC_BET_eq.ncl")
system("export path_d="+path_d+"; export path_f="+path_f+"; export case="+case+";export run="+run+"; export yr0="+yr0+"; export yr1="+yr1+"; ncl FC_NEB_eq.ncl")
system("export path_d="+path_d+"; export path_f="+path_f+"; export case="+case+";export run="+run+"; export yr0="+yr0+"; export yr1="+yr1+"; ncl TREF_line.ncl")
;========= run FC_bar.ncl =================
yr0 =384
yr1 =384
system("export path_d="+path_d+"; export path_f="+path_f+"; export case="+case+"; export run="+run+"; export yr0="+yr0+"; export yr1="+yr1+"; ncl FC_bar.ncl")
system("export path_d="+path_d+"; export path_f="+path_f+"; export case="+case+"; export run="+run+"; export yr0="+yr0+"; export yr1="+yr1+"; ncl FC_PFT.ncl")
```

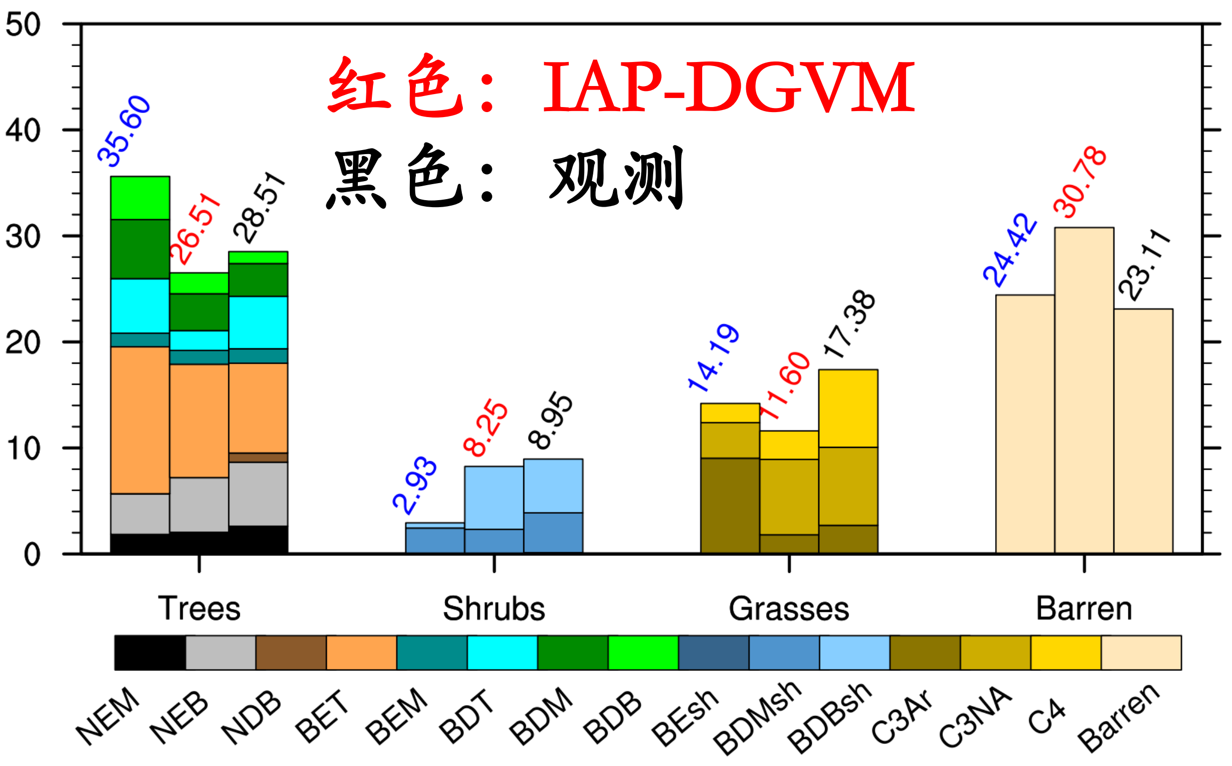
主要内容

Earth Lab

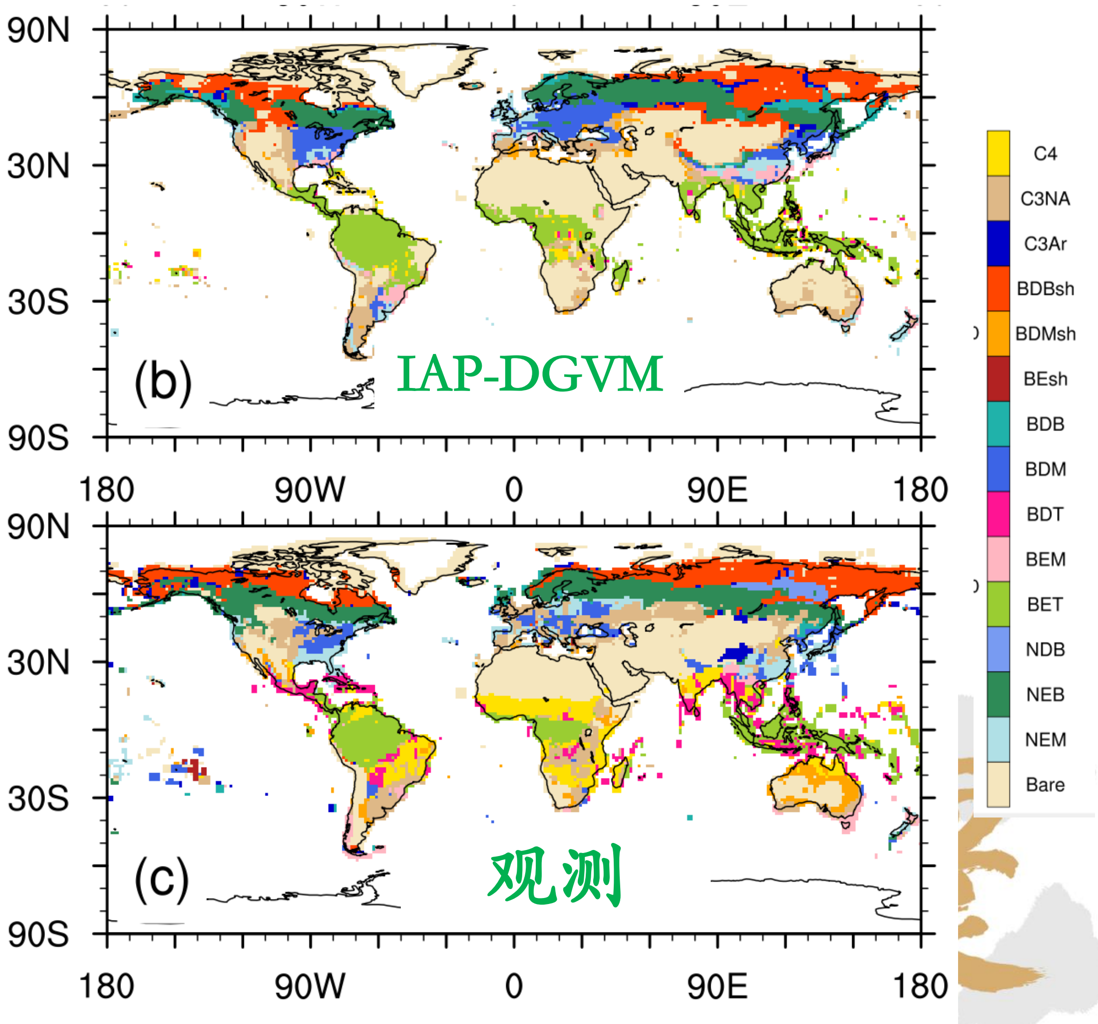
分系统应用1-模拟当前气候下自然植被状态



分系统应用1-模拟当前气候下自然植被状态



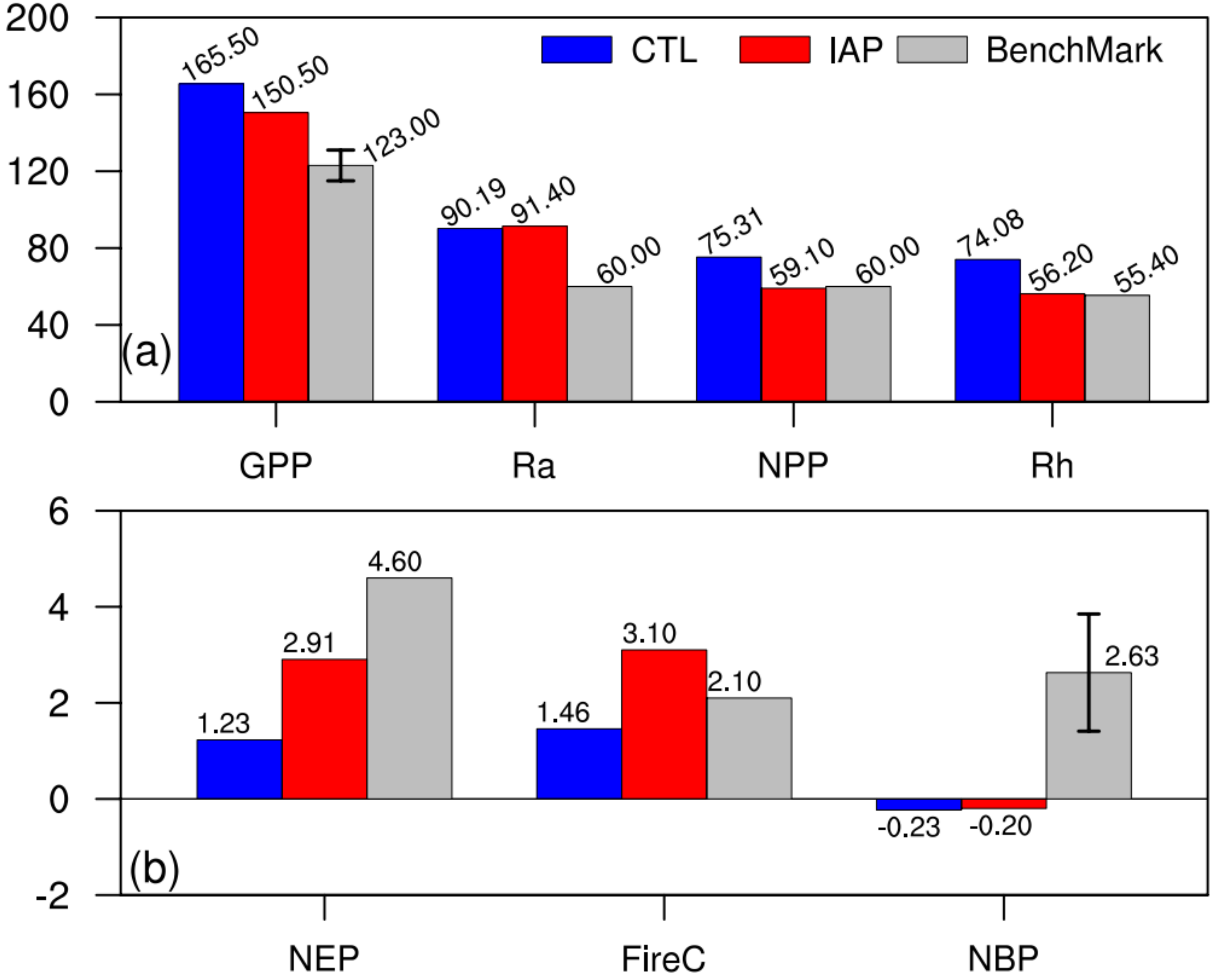
主导植被分布



EarthLab

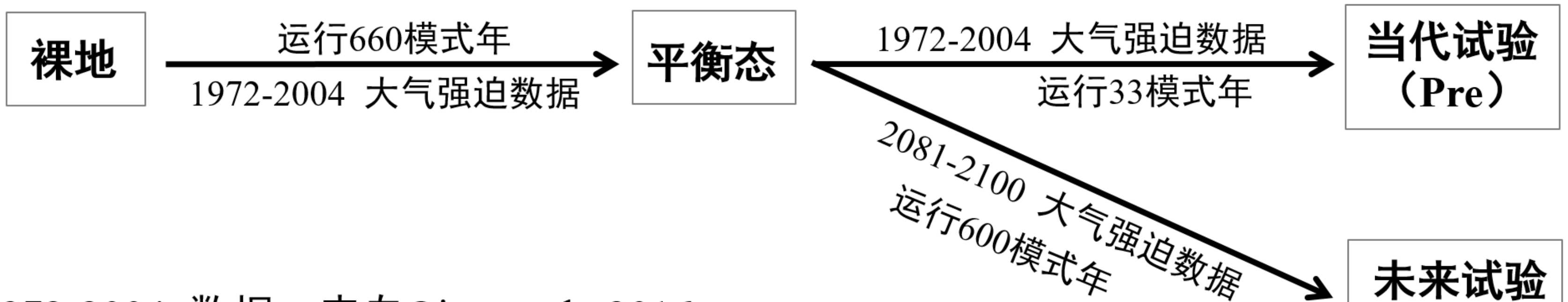
分系统应用1-模拟当前气候下自然植被状态

碳通量



分系统应用2-植被对未来气候变暖（RCP8.5）的响应研究

试验设计

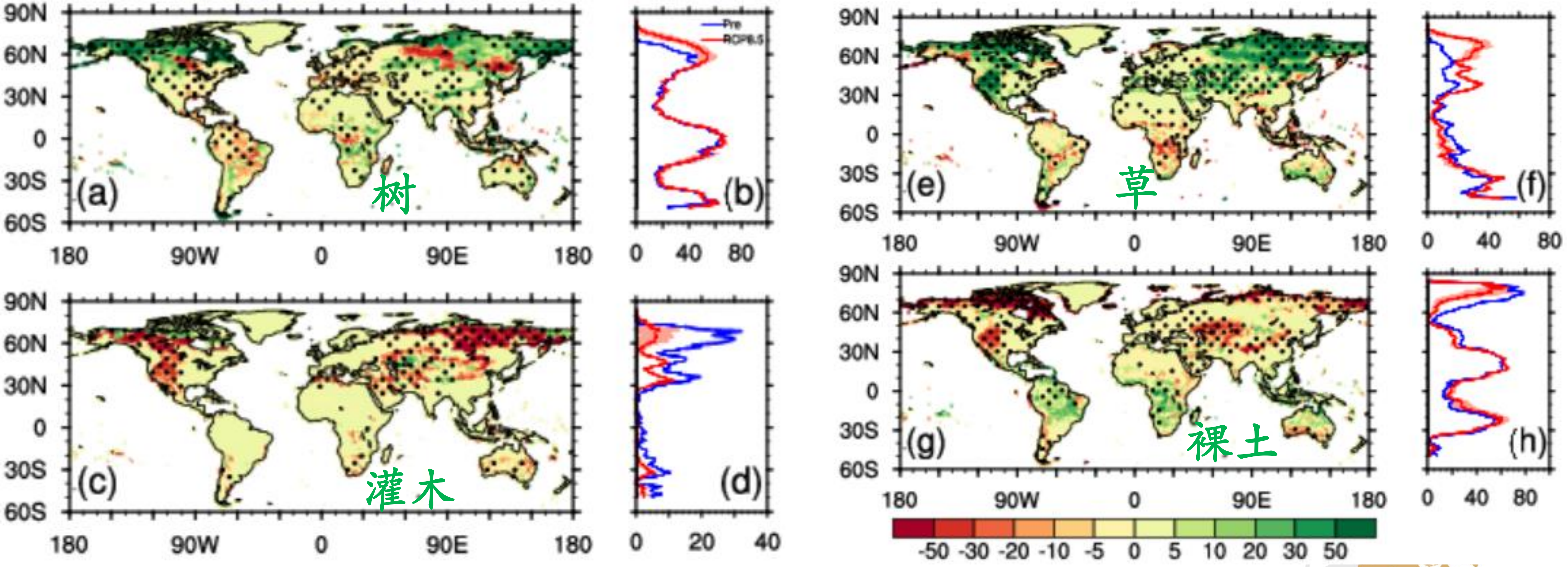


1972-2004 数据：来自Qian et al., 2016

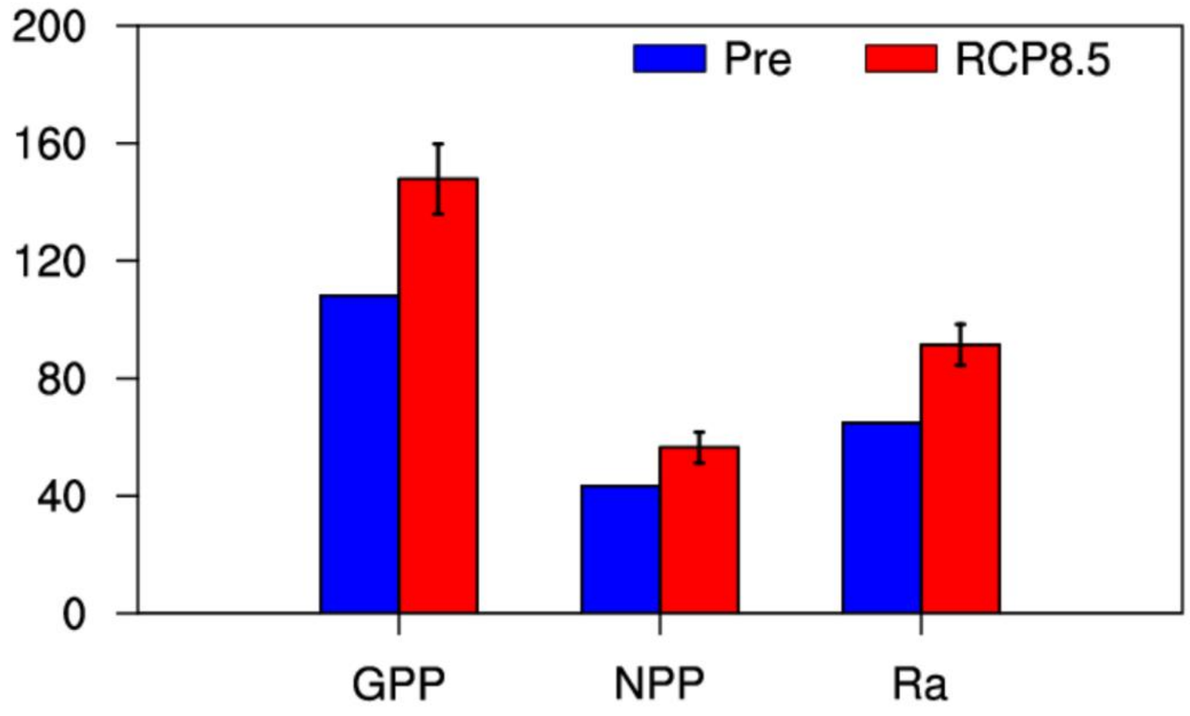
2081-2100 数据：$\boldsymbol{V}_{\text{new}}(i) = \overline{RCP8.5}_{2081-2100} - \overline{Hist}_{1981-2000} + \boldsymbol{V}_{Qian}(i), i = 1981, 1982, \ldots, 2000$

Model name	References
ACCESS1.3	(Bi et al. 2013)
BCC-CSM1.1(m)	(Wu et al. 2014)
BCC-CSM1.1	(Wu et al. 2013)
CNRM-CM5	(Voldoire et al. 2012)
GFDL-CM3	(Griffies et al. 2011)
GFDL-ESM2G	(Dunne et al. 2012)
GFDL-ESM2M	(Dunne et al. 2012)
GISS-E2-H	(Schmidt et al. 2014)
GISS-E2-R	(Schmidt et al. 2014)
INM-CM4	(Volodin et al. 2010)
IPSL-CM5A-LR	(Dufresne et al. 2013)
IPSL-CM5A-MR	(Dufresne et al. 2013)
MIROC5	(Watanabe et al. 2010)
MIROC-ESM-CHEM	(Watanabe et al. 2011)
MIROC-ESM	(Watanabe et al. 2011)
MRI-CGCM3	(Yukimoto et al. 2012)

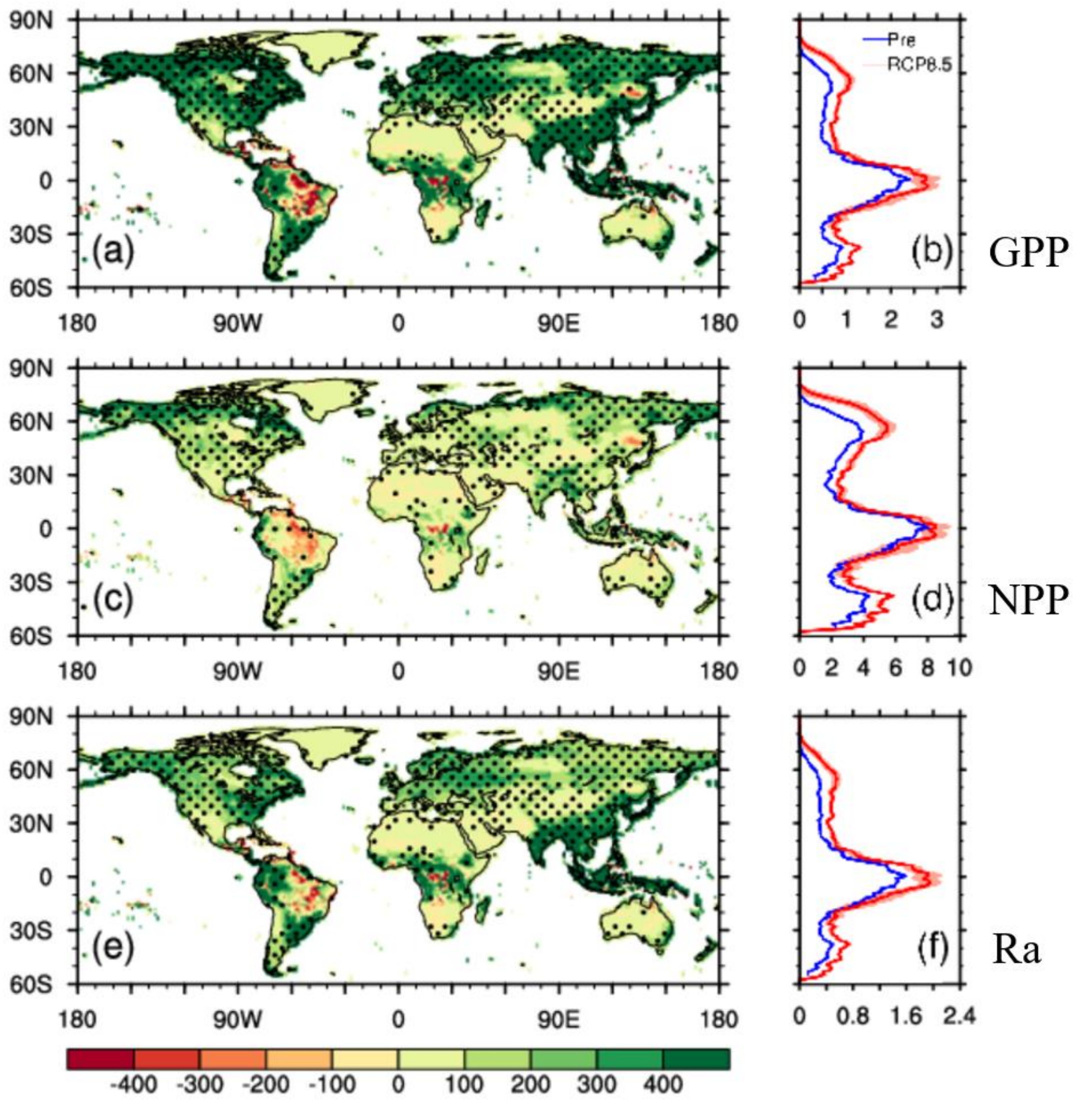
分系统应用2-植被对未来气候变暖（RCP8.5）的响应研究



分系统应用2-植被对未来气候变暖（RCP8.5）的响应研究



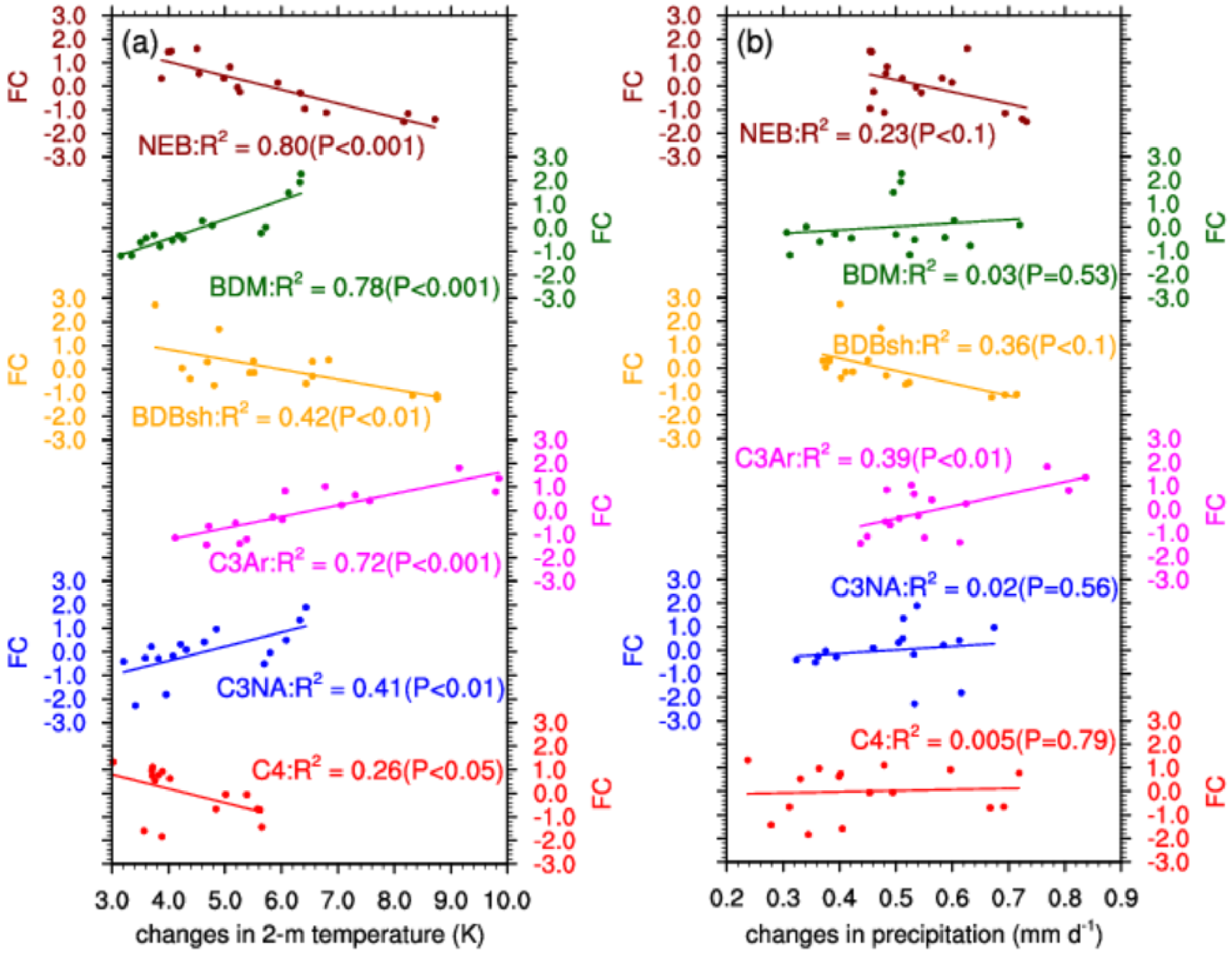
两类情景下全球平均碳通量（PgC yr^{-1}）



两类情景下碳通量的差异和纬向平均分布（gC m^{-2}yr^{-1}）

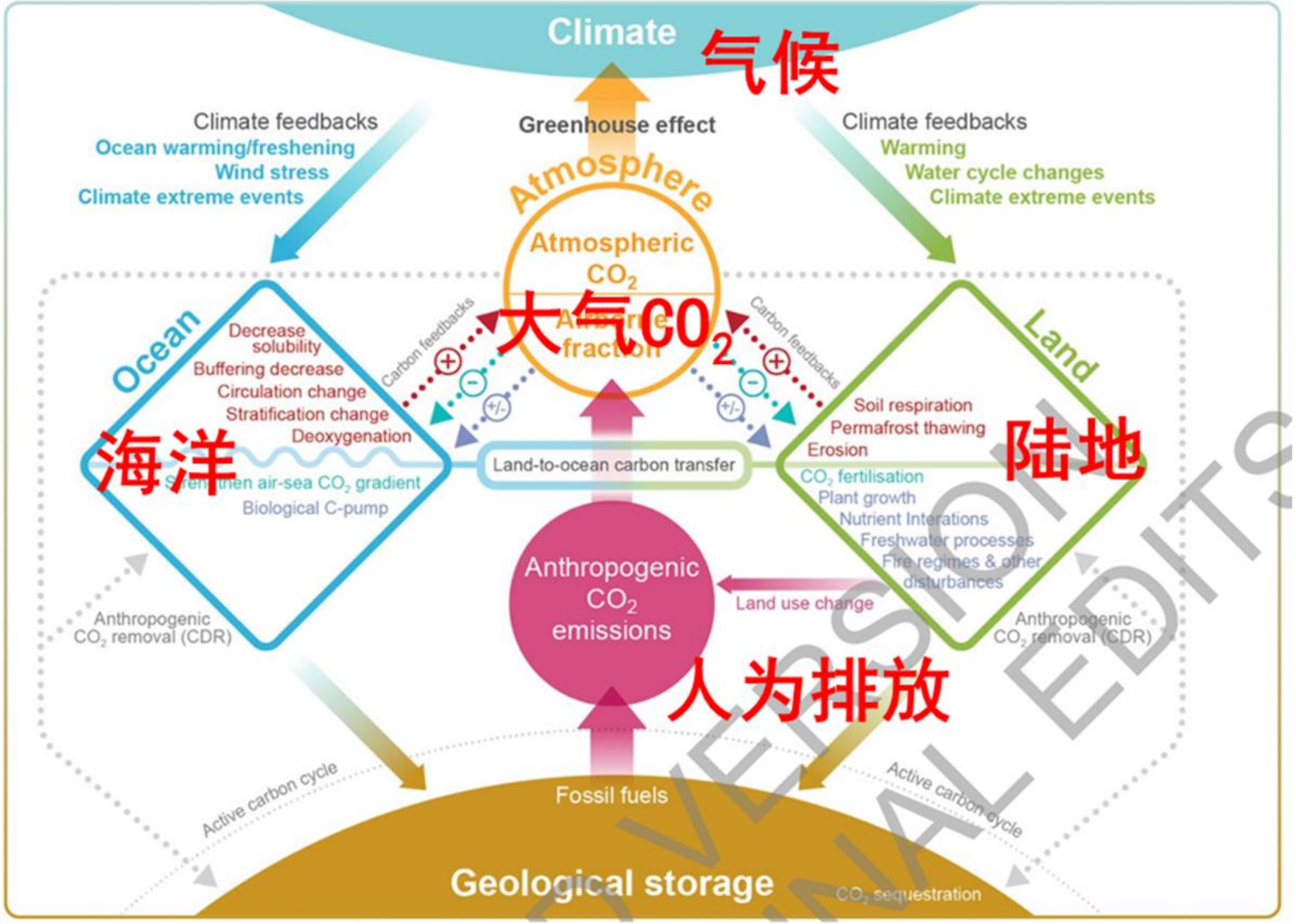
40

分系统应用2-植被对未来气候变暖（RCP8.5）的响应研究



Earth Lab

分系统应用3-CO2全耦合试验



AR6，2021

排放驱动试验：模式受人类排放驱动，计算大气CO2浓度，且模拟与陆地和海洋碳循环相互影响、相互作用

分系统应用3-CO2全耦合试验

排放驱动历史试验（esm-hist）：模拟历史时期（1850-2014年）大气CO2变化及其与陆地碳循环、海洋碳循环的相互作用

Table A1. Specifications in the DECK and CMIP6 historical simulations.

Experiment	Volcanic stratospheric aerosol	Solar variability	Anthropogenic forcings
amip	Time-dependent observations	Time-dependent observations	Time-dependent observations
piControl	Background volcanic aerosol that results in radiative forcing matching, as closely as possible, that was experienced, on average, during the historical simulation (i.e. 1850–2014 mean)	Fixed at its mean value (no 11-year solar cycle) over the first two solar cycles of the historical simulation (i.e. the 1850–1873 mean)	Given that the historical simulations start in 1850, the *piControl* should have fixed 1850 atmospheric composition, not true pre-industrial
esm-piControl	As in *piControl*	As in *piControl*	As in *piControl* but with CO_2 concentration calculated, rather than prescribed. CO_2 from both fossil fuel combustion and land-use change are prescribed to be zero.
esm-hist	As in *historical*	As in *historical*	As in *historical* but with CO_2 emissions prescribed and CO_2 concentration calculated (rather than prescribed)

Eyring et al., 2016

分系统应用3-CO2全耦合试验

- ACCESS-ESM1-5 (10)
- BCC-CSM2-MR (3)
- CESM2 (4)
- CNRM-ESM2-1 (4)
- CanESM5 (9)
- CanESM5-CanOE (3)
- EC-Earth3-CC (1)
- GFDL-ESM4 (1)
- MIROC-ES2L (3)
- MPI-ESM1-2-LR (10)
- MRI-ESM2-0 (1)
- NorESM2-LM (2)
- UKESM1-0-LL (4)

Institution ID

- BCC (3)
- CCCma (12)
- CNRM-CERFACS (4)
- CSIRO (10)
- EC-Earth-Consortium (1)
- MIROC (3)
- MOHC (4)
- MPI-M (10)
- MRI (1)
- NCAR (4)
- NCC (2)
- NOAA-GFDL (1)

Show All Replicas   Show All Versi

Search Constraints: esm-hist | co2

Total N
-1-
Please login to adc
Expert Users: you may display the search

1. **CMIP6.CMIP.BCC.BCC-CSM2-MR.esm-hist.r1i1p1f1.Amon.co2.gn**
Data Node: cmip.bcc.cma.cn is reported as unavailable. Any existing repl
Version: 20181221
Total Number of Files (for all variables): 5
Full Dataset Services: [**Show Metadata**] [**List Files**] [**WGET Script**

2. **CMIP6.CMIP.NCAR.CESM2.esm-hist.r1i1p1f1.AERmon.co2.gn**
Data Node: esgf-data.ucar.edu
Version: 20191105
Total Number of Files (for all variables): 4
Full Dataset Services: [**Show Metadata**] [**List Files**] [**WGET Script**

3. **CMIP6.CMIP.NCAR.CESM2.esm-hist.r1i1p1f1.Amon.co2.gn**
Data Node: esgf-data.ucar.edu
Version: 20191105
Total Number of Files (for all variables): 4
Full Dataset Services: [**Show Metadata**] [**List Files**] [**WGET Script**

4. **CMIP6.CMIP.NCAR.CESM2.esm-hist.r2i1p1f1.AERmon.co2.gn**
Data Node: esgf-data.ucar.edu
Version: 20191105
Total Number of Files (for all variables): 4
Full Dataset Services: [**Show Metadata**] [**List Files**] [**WGET Script**

5. **CMIP6.CMIP.NCAR.CESM2.esm-hist.r2i1p1f1.Amon.co2.gn**
Data Node: esgf-data.ucar.edu
Version: 20191105

分系统应用3-CO2全耦合试验

在13个模式中，只有2个模式同时包含火模式和全球植被动力学模式

火模式
动态植被

Modelling group	CSIRO	BCC	CCCma	CESM	CNRM	GFDL	IPSL	JAMSETC	MPI	NorESM2-LM	UK
ESM	ACCESS-ESM1.5	BCC-CSM2-MR	CanESM5	CESM2	CNRM-ESM2-1	GFDL-ESM4	IPSL-CM6A-LR	MIROC-ES2L	MPI-ESM1.2-LR	NorESM2-LM	UKESM1-0-LL
Land carbon/biogeochemistry component											
Model name	CABLE2.4 CASA-CNP	BCC-AVIM2	CLASS-CTEM	CLM5	ISBA-CTRIP	LM4p1	ORCHIDEE (2)	MATSIRO (phys) VISIT-e (BGC)	JSBACH3.2	CLM5	JULES-ES-1.0
Veg C pools	3	3	3	22	6	6	8	3	3	3	3
Dead C pools	6	8	2	7	7	4	3	6	18	7	4
PFTS	13	16	9	22	16	6	15	13	13	21	13
Fire	No	No	No	Yes	Yes	Yes	No	No	Yes	Yes	No
Dynamic Veg	No	No	No	No	No	Yes	No	No	Yes	No	Yes
Permafrost C	No	No	No	Yes	No	No	No	No	No	Yes	No
Nitrogen cycle	Yes	No	No	Yes	No	No	No	Yes	Yes	Yes	Yes
Ocean carbon/biogeochemistry component											
Model name	WOMBAT	MOM4_L40	CMOC (biol)	MARBL	PISCESv2-gas	COBALTv2	PISCES-v2	OECO2	HAMOCC6	HAMOCC5.1	MEDUSA-2.1
Phytoplankton	1	0	1	3	2	2	2	2	2	1	2
Zooplankton	1	0	1	1	2	3	2	1	1	1	2
Limiting Nutrients	P, Fe	P	N	N, P, Si, Fe	N, P, Si, Fe	N [illegible] F[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible] Si, Fe

IPCC AR 6; Arora et al., 2020

分系统应用3-CO2全耦合试验

陆地碳通量（cam_in%fco2_lnd=SFCO2_LND, kgCO2/m2/s）

1，陆面模式计算净碳通量

colm_cplMod.F90

```
lnd_nee(g)    = fldv%assim(g)-fldv%respc(g)-fldv%fmicr(g)

if(curr_yr.ge.lnd_cflux_year) then
 ! convert from molC/m2/s to kgCO2/m2/s
   lnd_nee(g) = lnd_nee(g)*44.0_r8*1.0e-3_r8
```

colm_cpl7.F90

```
clm_l2a%nee(g)          = lnd_nee(i)
```

2，耦合器接收碳通量并传给大气

lnd_comp_mct.F90

```
if (index_l2x_Fall_fco2_lnd /= 0) then
   l2x_l%rAttr(index_l2x_Fall_fco2_lnd,i) = l2a%nee(g)
end if
```

atm_comp_mct.F90

```
if (index_x2a_Faxx_fco2_lnd /= 0) then
ed converting to kg/m2/s, lnd co2 unit is already kg/m2/s, juanxiong he
   cam_in(c)%fco2_lnd(i) = -x2a_a%rAttr(index_x2a_Faxx_fco2_lnd,ig)
end if

if (index_x2a_Faxx_fco2_lnd /= 0) then

   cam_in(c)%cflx(i,c_i(4)) =cam_in(c)%cflx(i,c_i(4)) +  cam_in(c)%fco2_lnd(i)
```

3，输出陆气净碳通量

constituents.F90

```
sflxnam(pcnst+1)      = 'SFCO2_OCN'
sflxnam(pcnst+2)      = 'SFCO2_FFF'
sflxnam(pcnst+3)      = 'SFCO2_LND'
sflxnam(pcnst+4)      = 'CO2_OUT'
```

cam_diagnostics.F90

```
 call outfld(sflxnam(pcnst+1),cam_in%fco2_ocn, pcols, lchnk)
 call outfld(sflxnam(pcnst+3),cam_in%fco2_lnd, pcols, lchnk)
 call outfld(sflxnam(pcnst+4),cam_out%co2prog, pcols, lchnk)
f
o2_readFlux_fuel) then
 call outfld(sflxnam(pcnst+2),data_flux_fuel%co2flx, pcols, lchnk)
```

CO2全耦合试验-海洋碳通量

海洋碳通量（cam_in%fco2_ocn=SFCO2_OCN, kgCO2/m2/s）

1，海洋模式计算净碳通量

OBM.F90

```
uptake(i,j)=uptake(i,j)+ssfc(i,j)*vit(i,j,1)*1025.0_8*1.0E-6

  ENDDO
ENDDO

 COUP
if(ii==nss/ncpl) then
   uptake(:,:)=uptake(:,:)/ii

   co2_cpl = uptake
```

flux_cpl.F90

2，耦合器接收碳通量并传给大气

licomcpl.F90

```
  if (index_o2x_Faoo_fco2_ocn > 0) then
  o2x_o%rAttr(index_o2x_Faoo_fco2_ocn,n) =  co2_cpl(i,j)
  end if
```

atm_comp_mct.F90

```
 if (index_x2a_Faxx_fco2_ocn /= 0) then
 to kg/m2/s, juanxiong he
    cam_in(c)%fco2_ocn(i) = -x2a_a%rAttr(index_x2a_Faxx_fco2_ocn,ig)*mwco2*1.0e-3_r8
 end if

 if (index_x2a_Faxx_fco2_ocn /= 0) then

    cam_in(c)%cflx(i,c_i(4)) = cam_in(c)%fco2_ocn(i)
```

3，输出海气净碳通量

constituents.F90

```
sflxnam(pcnst+1)      = 'SFCO2_OCN'
sflxnam(pcnst+2)      = 'SFCO2_FFF'
sflxnam(pcnst+3)      = 'SFCO2_LND'
sflxnam(pcnst+4)      = 'CO2_OUT'
```

cam_diagnostics.F90

```
  call outfld(sflxnam(pcnst+1),cam_in%fco2_ocn, pcols, lchnk)
  call outfld(sflxnam(pcnst+3),cam_in%fco2_lnd, pcols, lchnk)
  call outfld(sflxnam(pcnst+4),cam_out%co2prog, pcols, lchnk)
 f
:o2_readFlux_fuel) then
  call outfld(sflxnam(pcnst+2),data_flux_fuel%co2flx, pcols, lchnk)
```

分系统应用3-CO2全耦合试验

人为碳排放（data_flux_fuel%co2flx=SFCO2_FFF, kgCO2/m2/s）

1，读取人为碳排放

co2_cycle.F90

```
character(len=256) :: co2flux_fuel_file ='/work/zhujiawen/data/co2_anthro_emission_1850-2014-monthly_iap_c20210225.nc'
if (co2_readFlux_fuel) then
   call read_data_flux ( co2flux_fuel_file, data_flux_fuel )
end if
```

2，人为排放累加到大气

atm_comp_mct.F90

```
if (co2_readFlux_fuel) then
   cam_in(c)%cflx(i,c_i(4)) =cam_in(c)%cflx(i,c_i(4))+ data_flux_fuel%co2flx(i,c)
```

3，输出人为碳通量

constituents.F90

```
sflxnam(pcnst+1)      = 'SFCO2_OCN'
sflxnam(pcnst+2)      = 'SFCO2_FFF'
sflxnam(pcnst+3)      = 'SFCO2_LND'
sflxnam(pcnst+4)      = 'CO2_OUT'
```

cam_diagnostics.F90

```
 call outfld(sflxnam(pcnst+1),cam_in%fco2_ocn, pcols, lchnk)
 call outfld(sflxnam(pcnst+3),cam_in%fco2_lnd, pcols, lchnk)
 call outfld(sflxnam(pcnst+4),cam_out%co2prog, pcols, lchnk)
f
o2_readFlux_fuel) then
 call outfld(sflxnam(pcnst+2),data_flux_fuel%co2flx, pcols, lchnk)
```

```
float CO2_flux(time, lat, lon) ;
        CO2_flux:missing_value = -999.f ;
        CO2_flux:units = "kg m-2 s-1" ;
        CO2_flux:long_name = "CO2 fossil fuel emission flux"
        CO2_flux:standard_name = "tendency_of_atmosphere_mass_
```

分系统应用3-CO2全耦合试验

1，添加CO2变量

co2_cycle.F90

```
character(len=7), dimension(ncnst), parameter :: & ! constituent names
   c_names = (/'CO2_OCN', 'CO2_FFF', 'CO2_LND', 'CO2    '/)

write(6,*)'CO2 is defined'
    call cnst_add(c_names(4), c_mw(4), c_cp(4), c_qmin(4), c_i(4),longname=c_names(4), mixtype='dry')
```

2，碳通量（海洋、陆地、人为排放）累加（kg CO2/m2/s）

atm_comp_mct.F90

```
if (index_x2a_Faxx_fco2_ocn /= 0) then

   cam_in(c)%cflx(i,c_i(4)) = cam_in(c)%fco2_ocn(i)

if (co2_readFlux_fuel) then

   cam_in(c)%cflx(i,c_i(4)) =cam_in(c)%cflx(i,c_i(4))+ data_flux_fuel%co2flx(i,c)

if (index_x2a_Faxx_fco2_lnd /= 0) then

   cam_in(c)%cflx(i,c_i(4)) =cam_in(c)%cflx(i,c_i(4)) +  cam_in(c)%fco2_lnd(i)
```

3, 碳通量水平扩散(kg/kg)

/physics/cam/diffusion_solver.F90

```
tmp1(:ncol) = ztodt * gravit * rpdel(:ncol,pver)
! Add the explicit surface fluxes to the lowest layer
q(:ncol,pver,m) = q(:ncol,pver,m) + tmp1(:ncol) * cflx(:ncol,m)
```

ztodt: s
gravit:m/s^2
rpdel (气压倒数S/F): $m^2/(kg*m/s^2)$
cflx: kg $/m^2/s$

4, 碳通量垂直输送(kg/kg)

/physics/cam/vertical_diffusion.F90

5, 输出大气CO2（kg/kg）

cam_diagnostics.F90

```
ftem(:ncol,:) = state%q(:ncol,:,pcnst)
call outfld('CO2', ftem,  pcols, lchnk)
```

分系统应用3-CO_2全耦合试验

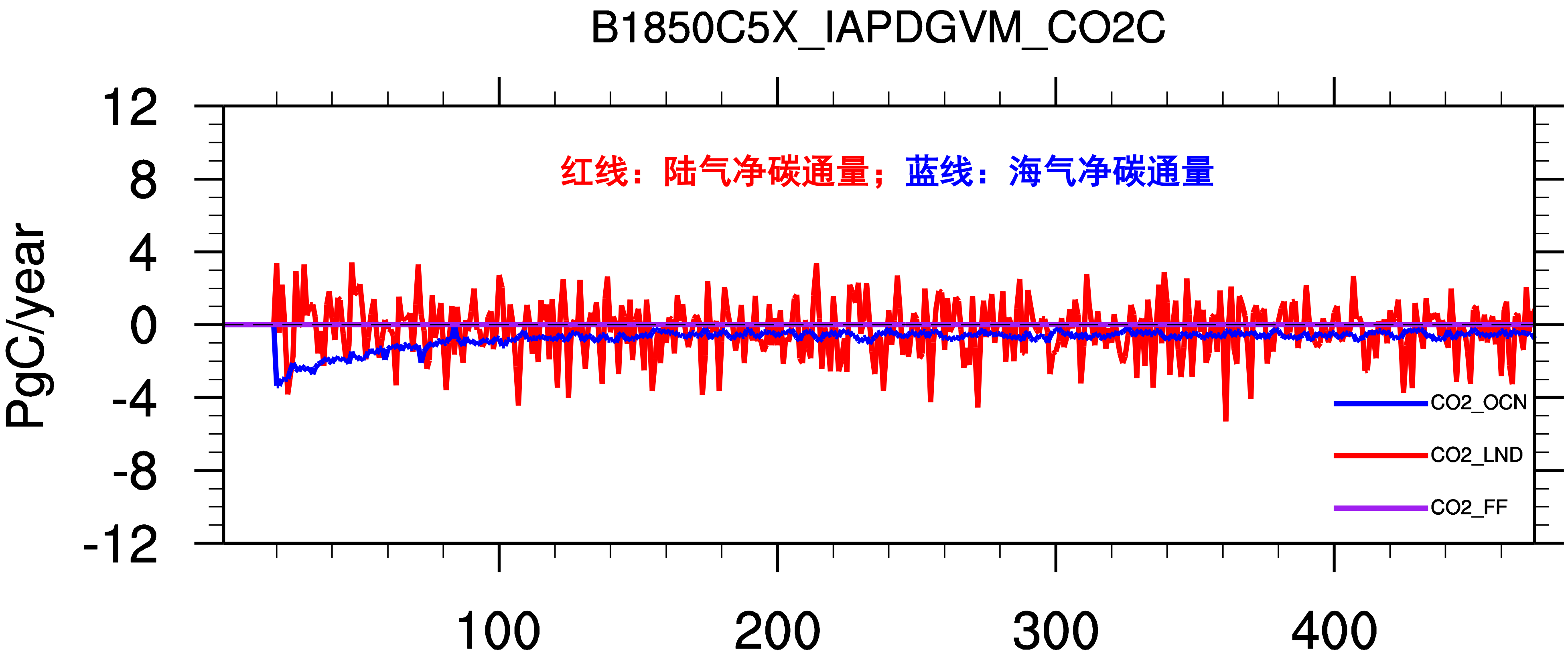
试验名称	大气CO_2	大气模式	陆地模式	动态植被模式	海洋模式
B1850_CO2C	模式计算	IAP-AGCM5.0	CoLM	IAP-DGVM1.0	LICOM2.0
B20TR_CO2C	模式计算	IAP-AGCM5.0	CoLM	IAP-DGVM1.0	LICOM2.0

B1850 (esm-pi)：工业革命前试验（1850年）

B20TR (esm-hist): 历史试验（1850-2014年）

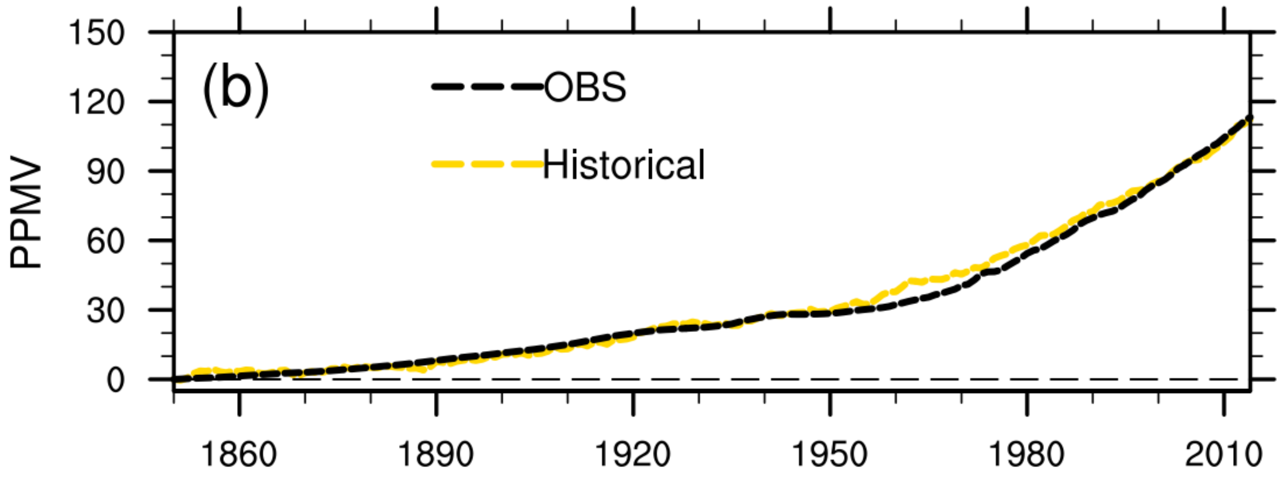
Earth Lab

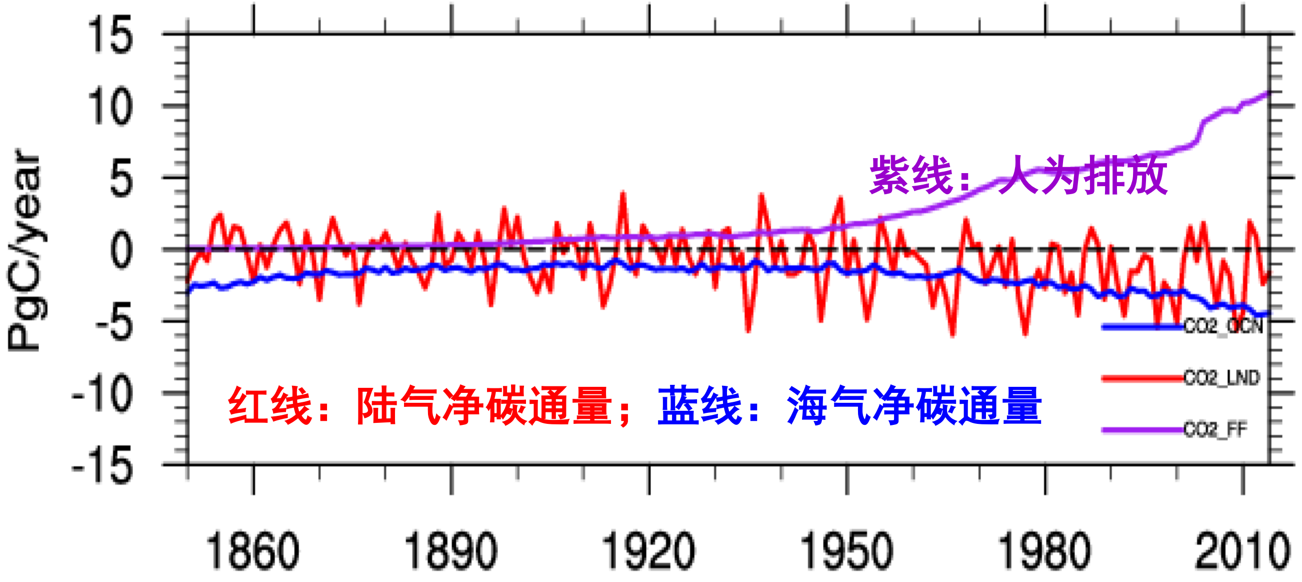
分系统应用3-CO2全耦合试验



Earth Lab

分系统应用3-CO2全耦合试验

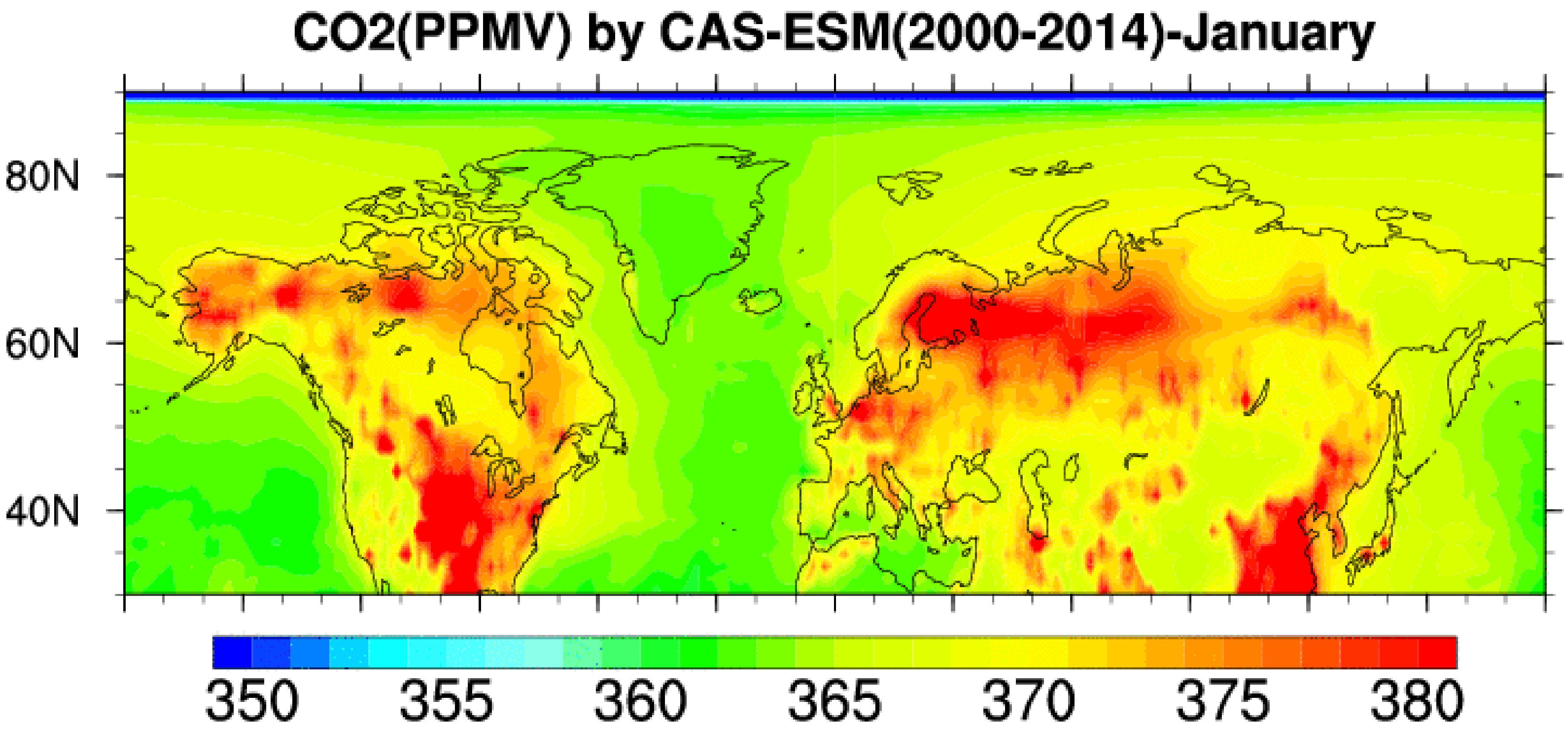




- 模拟的历史时期CO_2浓度增量与观测一致
- 陆地和海洋固碳量都增加，施肥效应

Earth Lab

分系统应用3-CO2全耦合试验



总结与展望

- 分系统简介、参数化原理、运行和后处理、分系统应用；
- 实现全球植被动力学模式（IAP-DGVM）与中科院地球系统模式（CAS-ESM）的耦合，广泛应用于研究植被演替、陆地碳循环与气候的相互作用；
- CAS-ESM2大气CO2的全耦合模拟试验，进一步优化后将提交CMIP6;


谢谢