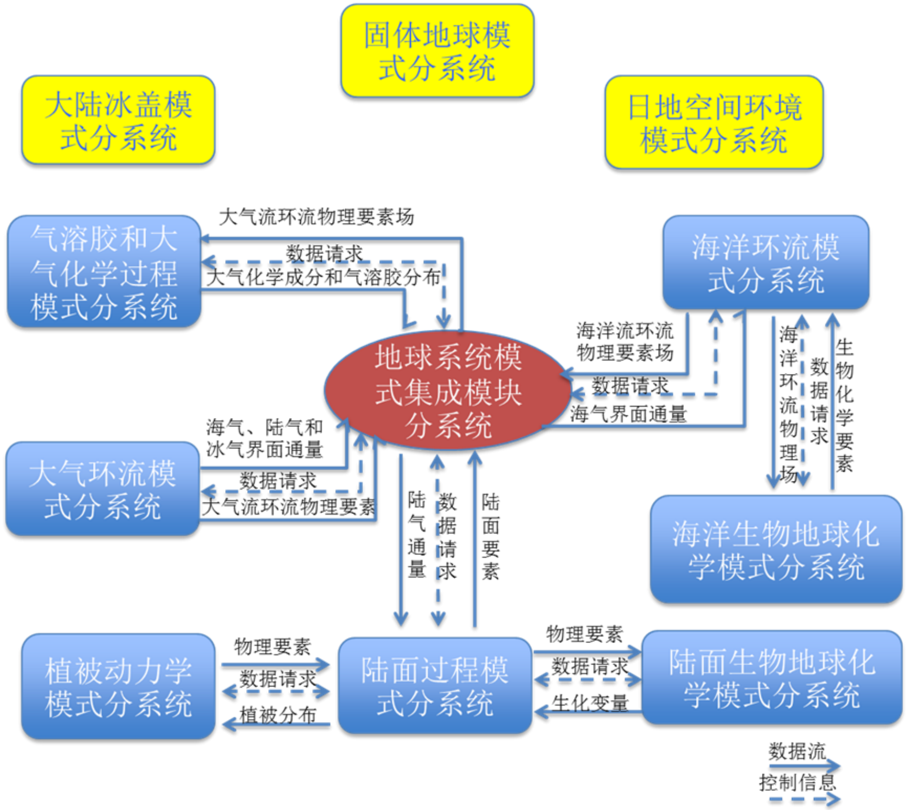
地球系统数值模拟装置项目
陆面过程模式分系统培训

培训人：魏楠

中山大学 大气科学学院

2022年5月27日

1. 分系统介绍



固体地球模式分系统

大陆冰盖模式分系统

日地空间环境模式分系统

气溶胶和大气化学过程模式分系统

大气流环流物理要素场

数据请求

大气化学成分和气溶胶分布

海洋环流模式分系统

海洋流环流物理要素场

数据请求

海气界面通量

海洋环流物理场

数据请求

生物化学要素

地球系统模式集成模块分系统

海气、陆气和冰气界面通量

数据请求

大气流环流物理要素

大气环流模式分系统

陆气通量

数据请求

陆面要素

海洋生物地球化学模式分系统

植被动力学模式分系统

物理要素

数据请求

植被分布

陆面过程模式分系统

物理要素

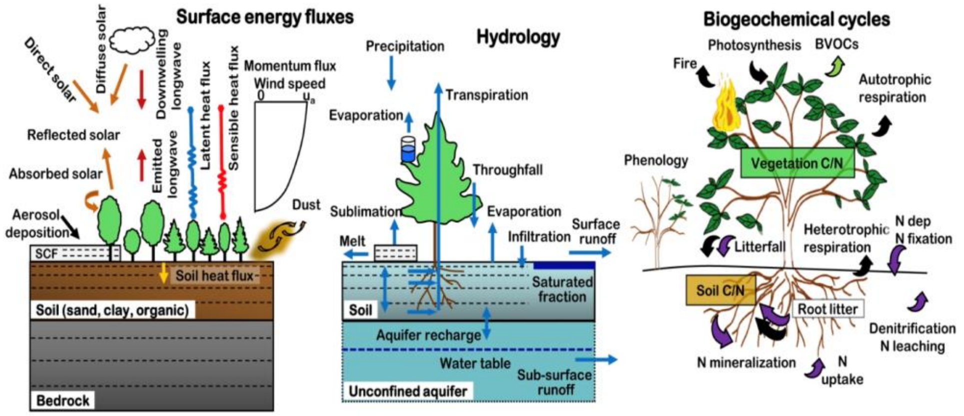
数据请求

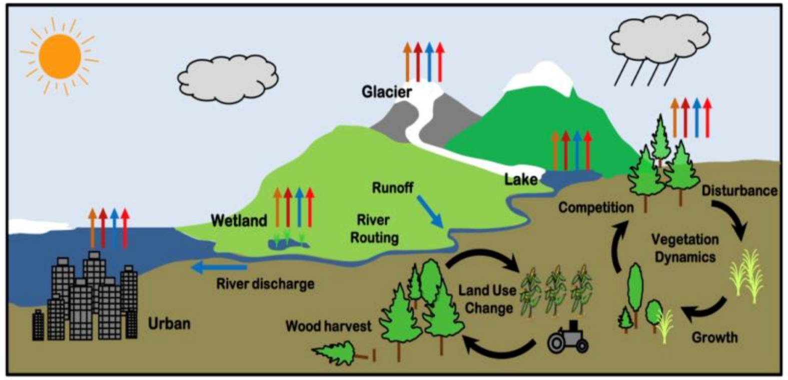
生化变量

陆面生物地球化学模式分系统

数据流

控制信息

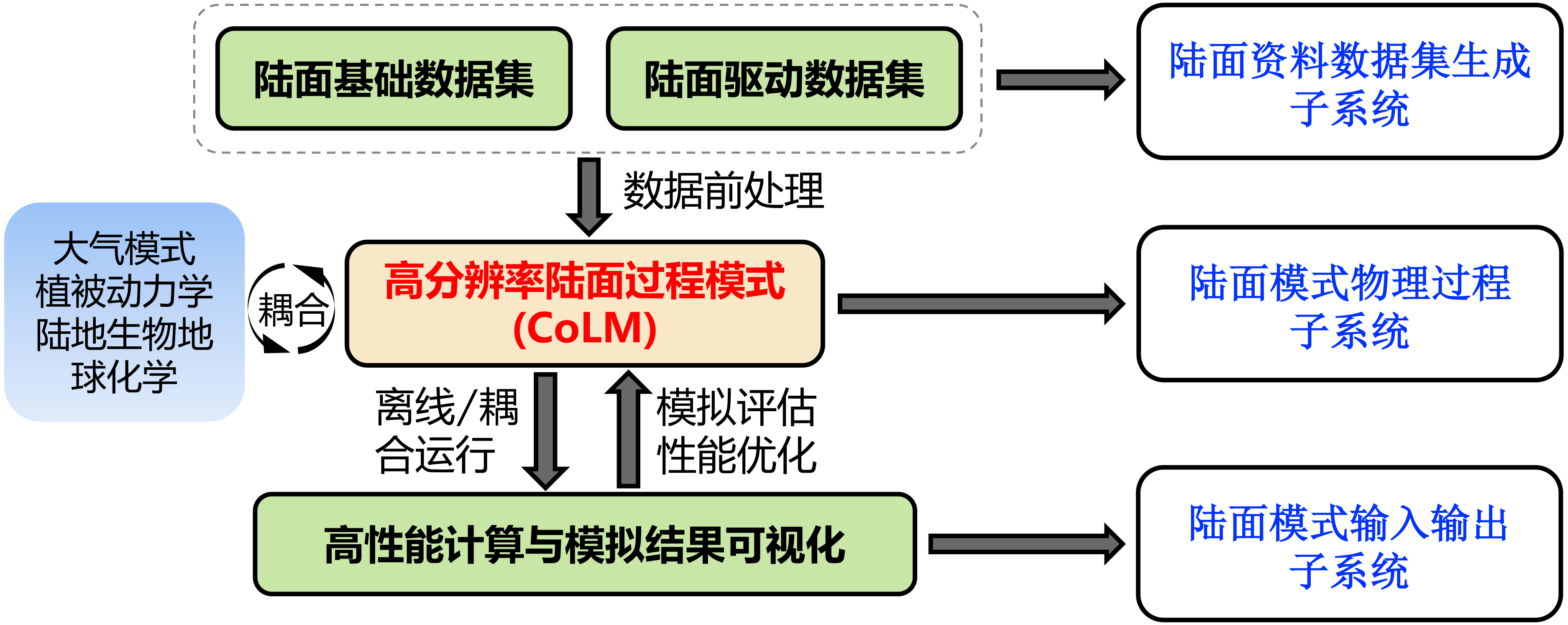




陆面过程模式是天气/气候/地球系统模式的**重要组成部分**，是陆面过程机理研究的**重要手段**，也是全球气象/水文/生态的精细化预报的**核心技术**，为大气模式分系统提供**下边界条件**。

1. 分系统介绍

功能：形成规范化、模块化、包含多过程之间的相互作用、可稳定模拟土壤温湿度、积雪、河川径流、植物生产力、陆气交互通量等陆面基本状态变量的陆面模式系统，同时为大气和海洋模式提供边界条件。



陆面基础数据集

陆面驱动数据集

陆面资料数据集生成子系统

数据前处理

大气模式
植被动力学
陆地生物地球化学

耦合

高分辨率陆面过程模式 (CoLM)

陆面模式物理过程子系统

离线/耦合运行

模拟评估性能优化

高性能计算与模拟结果可视化

陆面模式输入输出子系统

1. 分系统介绍

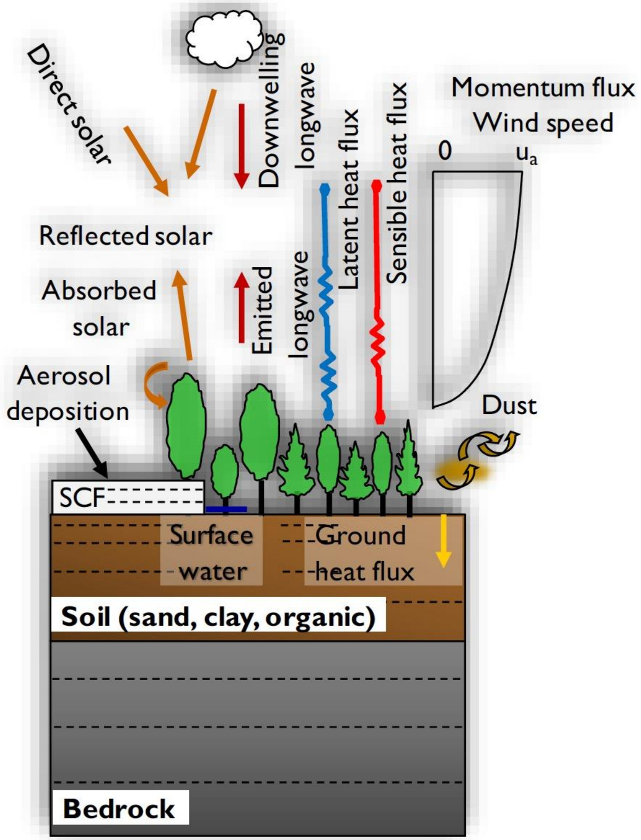
具备的功能与性能：

- 耦合版本的水平分辨率与大气环流模式相同，有多种水平分辨率可选；
- 离线版本的全球水平分辨率约5KM（中国区域1KM），垂直35层，10000核时的离线运行速度达到5模式年/天，并行效率达到30%；
- 模式功能较为完备，包括地表资料数据集生成、陆面模式物理过程和陆面模式输入输出子系统；
- 模式性能较为优良，对东亚区域的模拟效果优于或不差于国际主流模式。

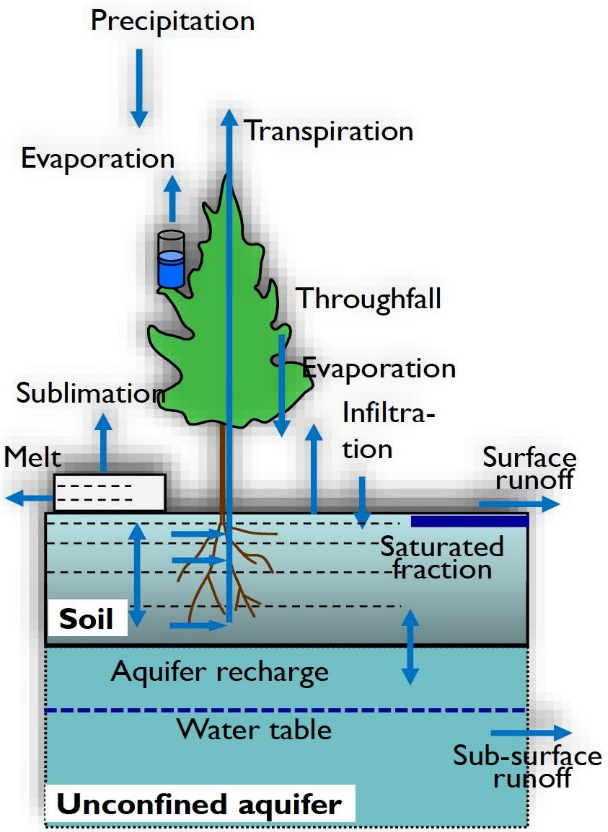
2. 模式原理

通用陆面过程模式CoLM：模拟地表主要生物地球物理过程

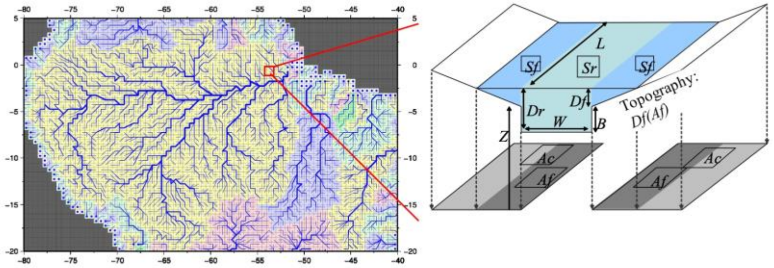
Surface energy fluxes



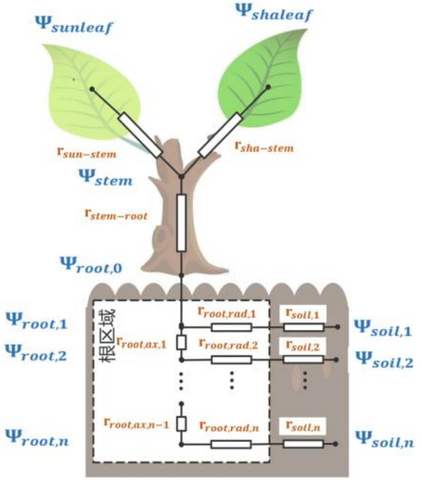
Hydrology



CaMa-Flood (**Ca**tchment-based **Ma**cro-scale **Flood**plain model)
- Distributed river routing model using River Network Map



Yamazaki, Kim et al, 2011，2020



2. 模式原理

陆面过程模式的大气驱动：模式向前运转的动力

传统意义的有：

- 风(u,v)
- 压(P)
- 温(T)
- 湿(q)
- 降雨(p)
- 辐射(sw, lw)

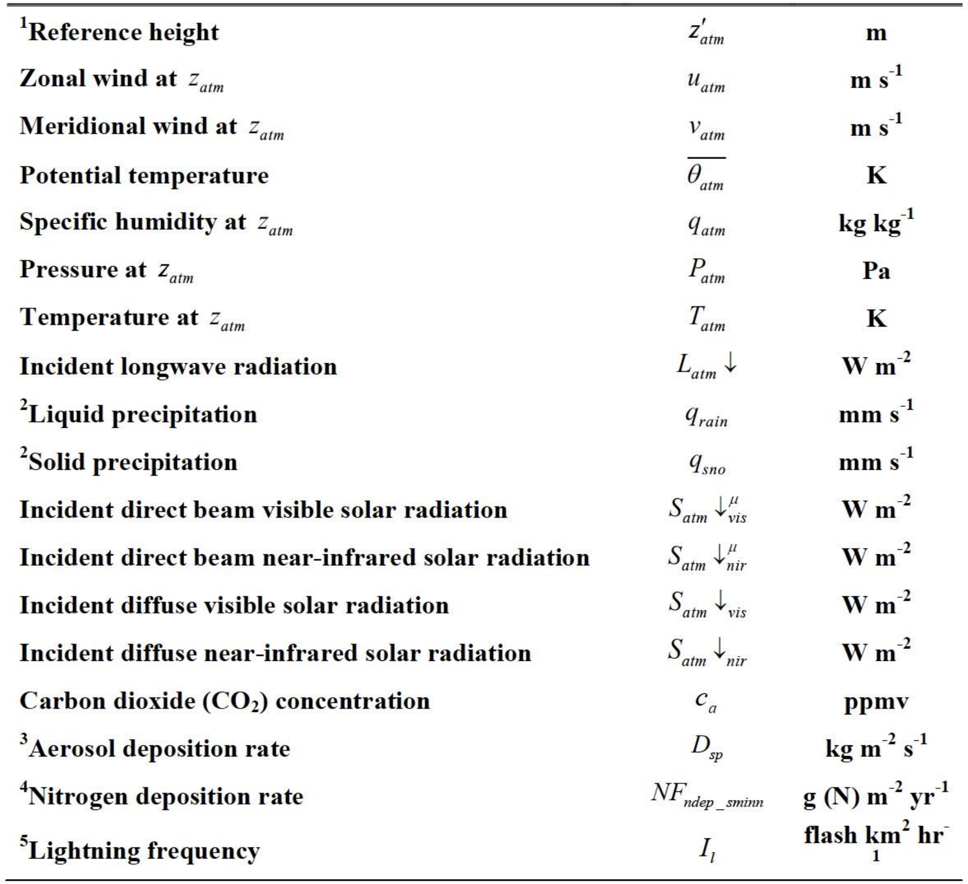
以上是在给定参考(观测)高度上的值

CO_2浓度

Aerosol/$N_{deposition\_rate}$

…

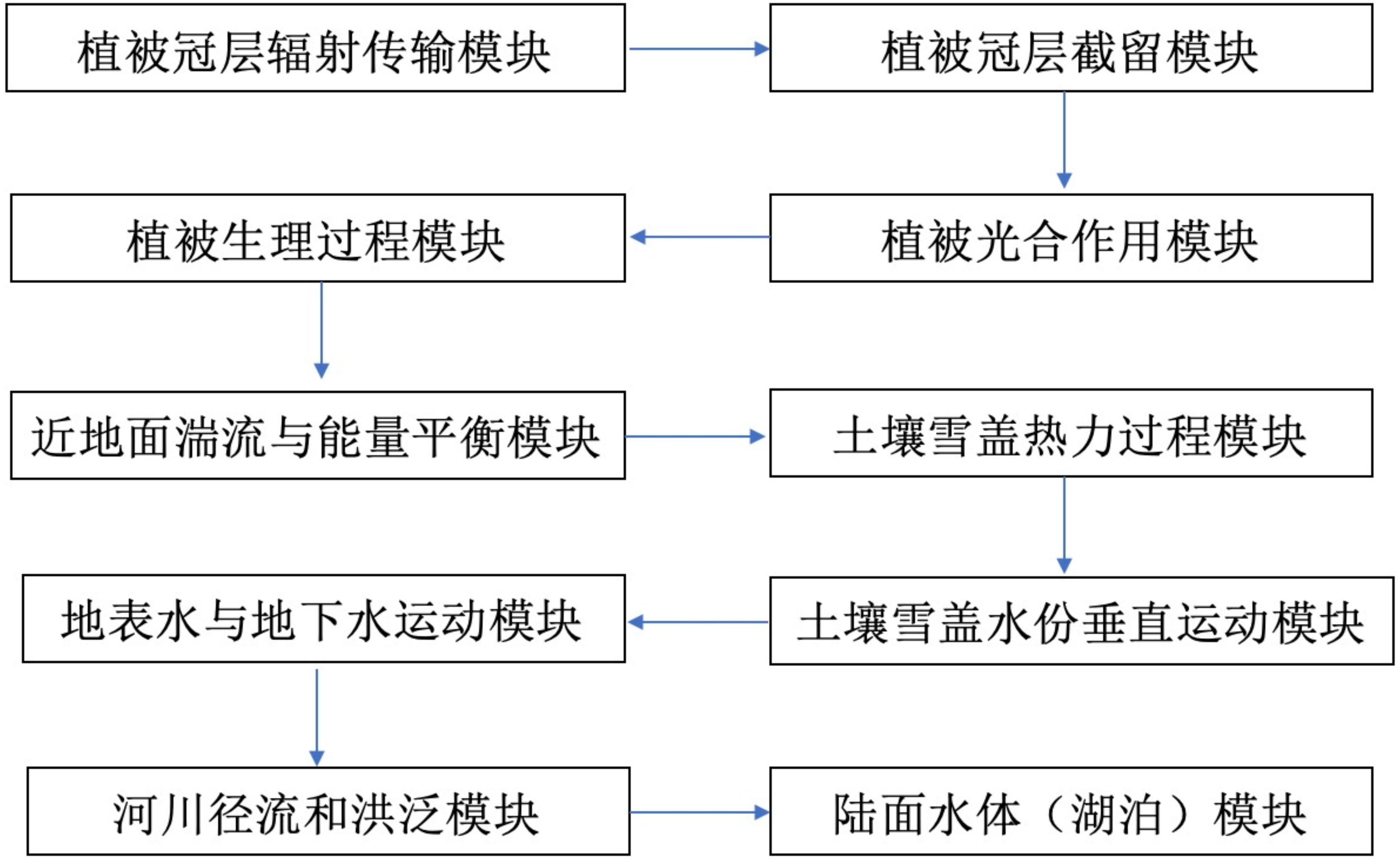
例：CLM陆面模式强迫输入变量—CLM5.0技术报告



[1]Reference height	z'_{atm}	m
Zonal wind at z_{atm}	u_{atm}	$m\ s^{-1}$
Meridional wind at z_{atm}	v_{atm}	$m\ s^{-1}$
Potential temperature	$\overline{\theta_{atm}}$	K
Specific humidity at z_{atm}	q_{atm}	$kg\ kg^{-1}$
Pressure at z_{atm}	P_{atm}	Pa
Temperature at z_{atm}	T_{atm}	K
Incident longwave radiation	$L_{atm}\downarrow$	$W\ m^{-2}$
[2]Liquid precipitation	q_{rain}	$mm\ s^{-1}$
[2]Solid precipitation	q_{sno}	$mm\ s^{-1}$
Incident direct beam visible solar radiation	$S_{atm}\downarrow^{\mu}_{vis}$	$W\ m^{-2}$
Incident direct beam near-infrared solar radiation	$S_{atm}\downarrow^{\mu}_{nir}$	$W\ m^{-2}$
Incident diffuse visible solar radiation	$S_{atm}\downarrow_{vis}$	$W\ m^{-2}$
Incident diffuse near-infrared solar radiation	$S_{atm}\downarrow_{nir}$	$W\ m^{-2}$
Carbon dioxide (CO_2) concentration	c_a	ppmv
[3]Aerosol deposition rate	D_{sp}	$kg\ m^{-2}\ s^{-1}$
[4]Nitrogen deposition rate	NF_{ndep_sminn}	$g\ (N)\ m^{-2}\ yr^{-1}$
[5]Lightning frequency	I_l	$flash\ km^{2}\ hr^{-1}$

2. 模式原理

通用陆面过程模式CoLM：主要物理模块



植被冠层辐射传输模块

植被冠层截留模块

植被光合作用模块

植被生理过程模块

近地面湍流与能量平衡模块

土壤雪盖热力过程模块

土壤雪盖水份垂直运动模块

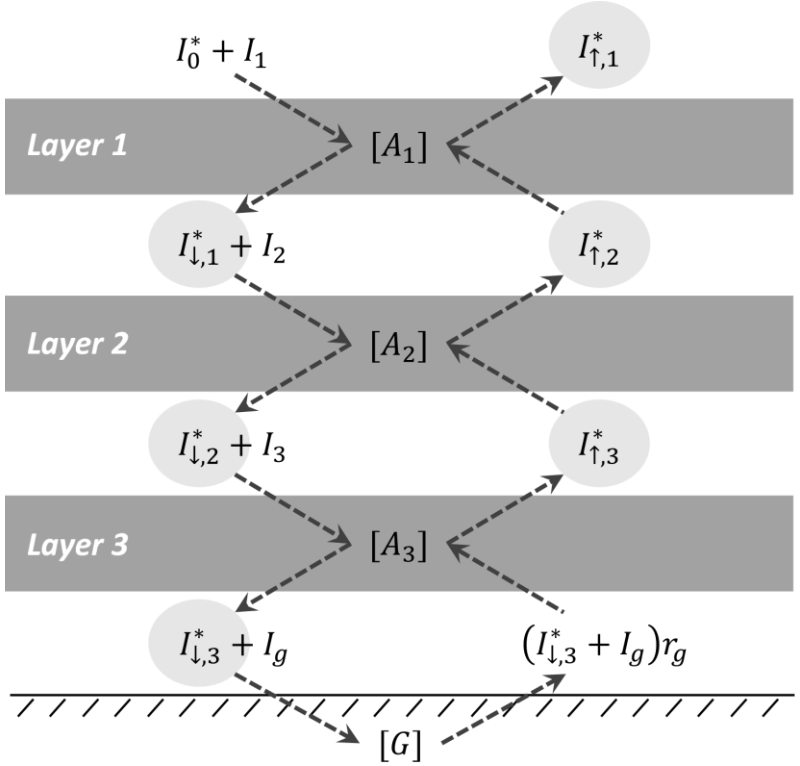
地表水与地下水运动模块

河川径流和洪泛模块

陆面水体（湖泊）模块

2. 模式原理

通用陆面过程模式CoLM：植被冠层辐射传输模块



$$I_{\uparrow,1}^* = I_1\alpha_1 + I_{\uparrow,2}^* S_1^* T_{i,1}^* + I_{\uparrow,2}^*\left(1-S_1^*\right) + I_0^* S_1^* \alpha_1^*,$$

$$I_{\downarrow,1}^* = I_1 T_{i,1} + I_0^* S_1^* T_{i,1}^* + I_0^*\left(1-S_1^*\right) + I_{\uparrow,2}^* S_1^* \alpha_1^*,$$

$$I_{\uparrow,2}^* = I_2\alpha_2 + I_{\uparrow,3}^* S_2^* T_{i,2}^* + I_{\uparrow,3}^*\left(1-S_2^*\right) + I_{\downarrow,1}^* S_2^* \alpha_2^*,$$

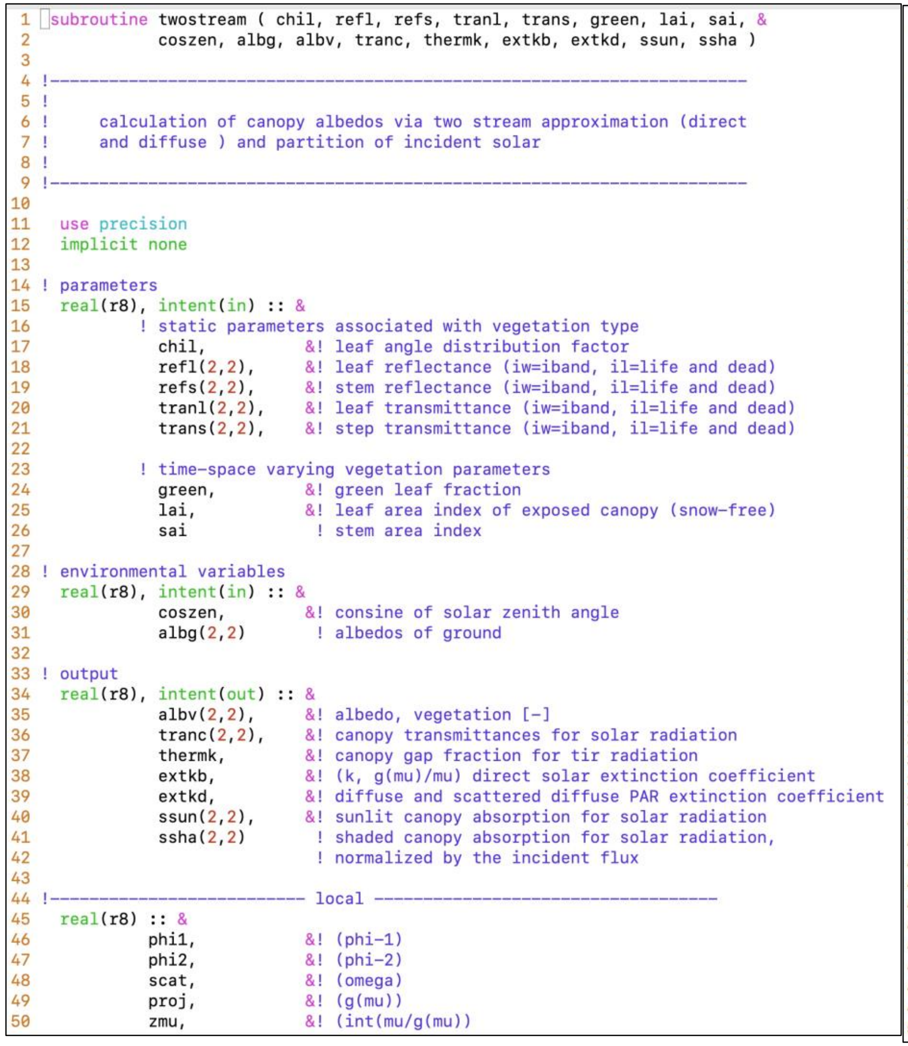
$$I_{\downarrow,2}^* = I_2 T_{i,2} + I_{\downarrow,1}^* S_2^* T_{i,2}^* + I_{\downarrow,1}^*\left(1-S_2^*\right) + I_{\uparrow,3}^* S_2^* \alpha_2^*,$$

$$I_{\uparrow,3}^* = I_3\alpha_3 + \left(I_{\downarrow,3}^* + I_g\right) r_g S_3^* T_{i,3}^* + \left(I_{\downarrow,3}^* + I_g\right) r_g \left(1-S_3^*\right) + I_{\downarrow,2}^* S_3^* \alpha_3^*$$

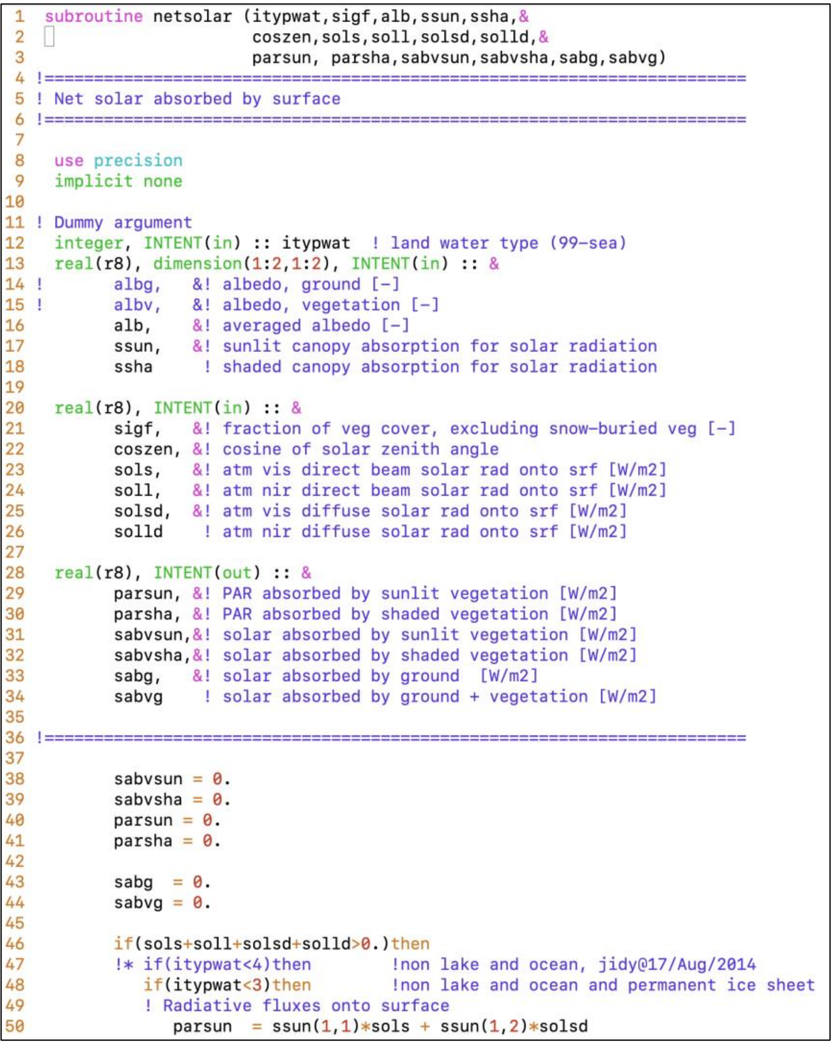
求解线性方程组计算植被冠层和地表的辐射通量，采用二流近似的方法求解植被反照率。

2. 模式原理

通用陆面过程模式CoLM：植被冠层辐射传输模块



```fortran
subroutine twostream ( chil, refl, refs, tranl, trans, green, lai, sai, &
           coszen, albg, albv, tranc, thermk, extkb, extkd, ssun, ssha )

!-----------------------------------------------------------------------
!
!     calculation of canopy albedos via two stream approximation (direct
!     and diffuse ) and partition of incident solar
!
!-----------------------------------------------------------------------

  use precision
  implicit none

! parameters
  real(r8), intent(in) :: &
          ! static parameters associated with vegetation type
            chil,          &! leaf angle distribution factor
            refl(2,2),     &! leaf reflectance (iw=iband, il=life and dead)
            refs(2,2),     &! stem reflectance (iw=iband, il=life and dead)
            tranl(2,2),    &! leaf transmittance (iw=iband, il=life and dead)
            trans(2,2),    &! step transmittance (iw=iband, il=life and dead)

          ! time-space varying vegetation parameters
            green,         &! green leaf fraction
            lai,           &! leaf area index of exposed canopy (snow-free)
            sai             ! stem area index

! environmental variables
  real(r8), intent(in) :: &
            coszen,        &! consine of solar zenith angle
            albg(2,2)       ! albedos of ground

! output
  real(r8), intent(out) :: &
            albv(2,2),     &! albedo, vegetation [-]
            tranc(2,2),    &! canopy transmittances for solar radiation
            thermk,        &! canopy gap fraction for tir radiation
            extkb,         &! (k, g(mu)/mu) direct solar extinction coefficient
            extkd,         &! diffuse and scattered diffuse PAR extinction coefficient
            ssun(2,2),     &! sunlit canopy absorption for solar radiation
            ssha(2,2)       ! shaded canopy absorption for solar radiation,
                            ! normalized by the incident flux

!-------------------------- local -----------------------------------
  real(r8) :: &
           phi1,           &! (phi-1)
           phi2,           &! (phi-2)
           scat,           &! (omega)
           proj,           &! (g(mu))
           zmu,            &! (int(mu/g(mu))
```

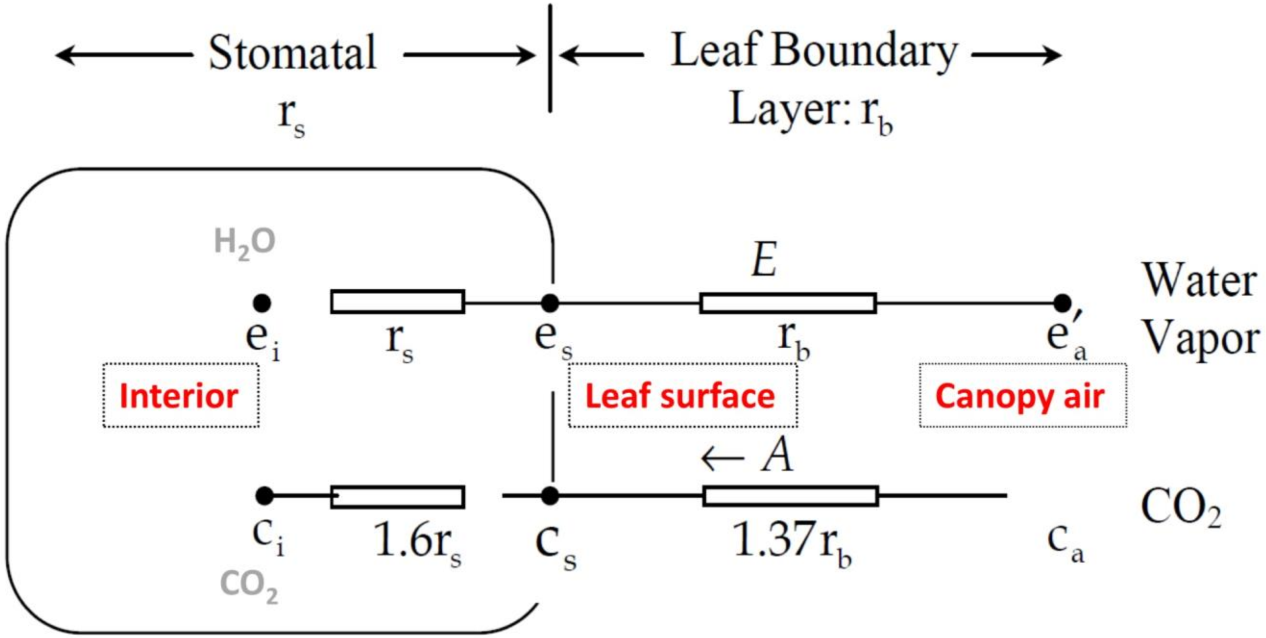


```fortran
subroutine netsolar (itypwat,sigf,alb,ssun,ssha,&
                     coszen,sols,soll,solsd,solld,&
                     parsun, parsha,sabvsun,sabvsha,sabg,sabvg)
!=======================================================================
! Net solar absorbed by surface
!=======================================================================

  use precision
  implicit none

! Dummy argument
  integer, INTENT(in) :: itypwat  ! land water type (99-sea)
  real(r8), dimension(1:2,1:2), INTENT(in) :: &
!       albg,   &! albedo, ground [-]
!       albv,   &! albedo, vegetation [-]
        alb,    &! averaged albedo [-]
        ssun,   &! sunlit canopy absorption for solar radiation
        ssha     ! shaded canopy absorption for solar radiation

  real(r8), INTENT(in) :: &
        sigf,   &! fraction of veg cover, excluding snow-buried veg [-]
        coszen, &! cosine of solar zenith angle
        sols,   &! atm vis direct beam solar rad onto srf [W/m2]
        soll,   &! atm nir direct beam solar rad onto srf [W/m2]
        solsd,  &! atm vis diffuse solar rad onto srf [W/m2]
        solld    ! atm nir diffuse solar rad onto srf [W/m2]

  real(r8), INTENT(out) :: &
        parsun, &! PAR absorbed by sunlit vegetation [W/m2]
        parsha, &! PAR absorbed by shaded vegetation [W/m2]
        sabvsun,&! solar absorbed by sunlit vegetation [W/m2]
        sabvsha,&! solar absorbed by shaded vegetation [W/m2]
        sabg,   &! solar absorbed by ground  [W/m2]
        sabvg    ! solar absorbed by ground + vegetation [W/m2]

!=======================================================================

        sabvsun = 0.
        sabvsha = 0.
        parsun = 0.
        parsha = 0.

        sabg  = 0.
        sabvg = 0.

        if(sols+soll+solsd+solld>0.)then
        !* if(itypwat<4)then        !non lake and ocean, jidy@17/Aug/2014
           if(itypwat<3)then        !non lake and ocean and permanent ice sheet
           ! Radiative fluxes onto surface
              parsun  = ssun(1,1)*sols + ssun(1,2)*solsd
```

2. 模式原理

通用陆面过程模式CoLM：植被光合作用与能量平衡模块

气孔阻抗模型，并将光合作用速率与气孔导度之间的关系进行参数化



H_2O and CO_2 exchanges with external environment:

$E_{tr} \propto g_l(e_s - e_a) = g_s(e_i - e_s)$ H_2O flux (kg m^{-2} s^{-1})

$A_n = \frac{c_a - c_s}{p}\frac{g_l}{1.37} = \frac{c_s - c_i}{p}\frac{g_s}{1.6}$ CO_2 flux (mol m^{-2} s^{-1})

g_s = leaf stomatal conductance (mol m^{-2} s^{-1}); $g_s = 1/r_s$

g_l = leaf boundary conductance (mol m^{-2} s^{-1}); $g_l = 1/r_b$

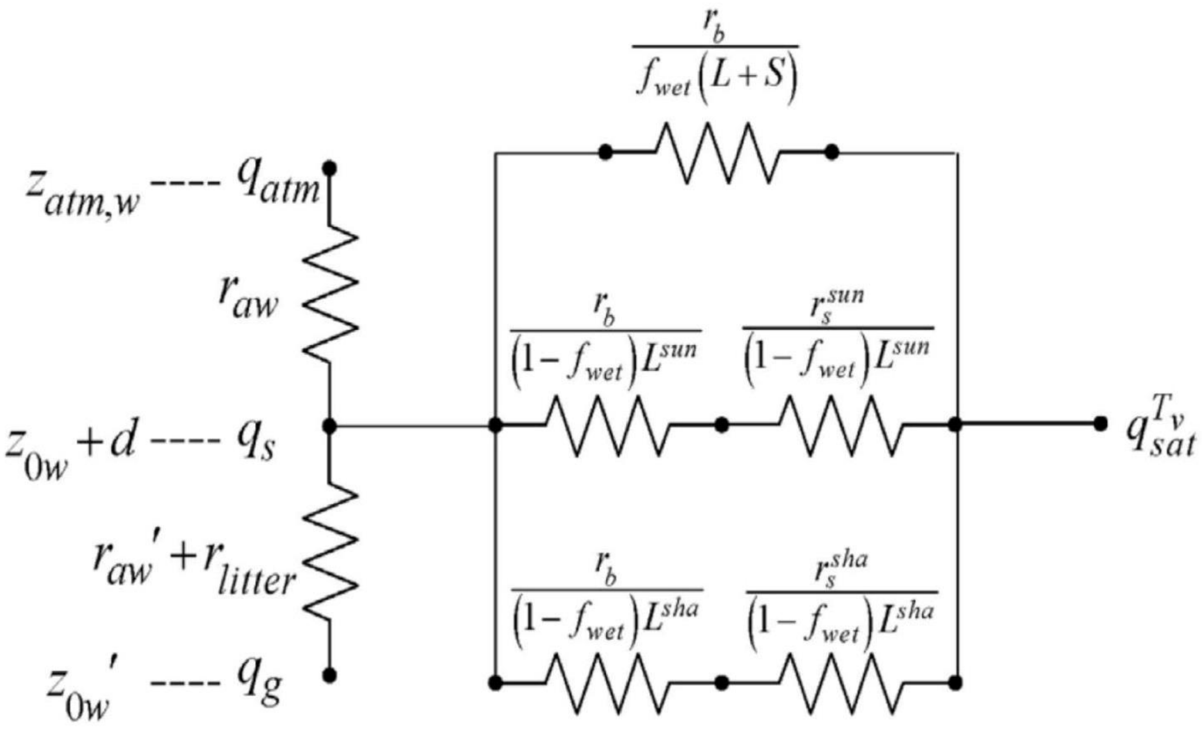
An empirical model for photosynthesis and stomatal conductance (Ball et al., 1987)

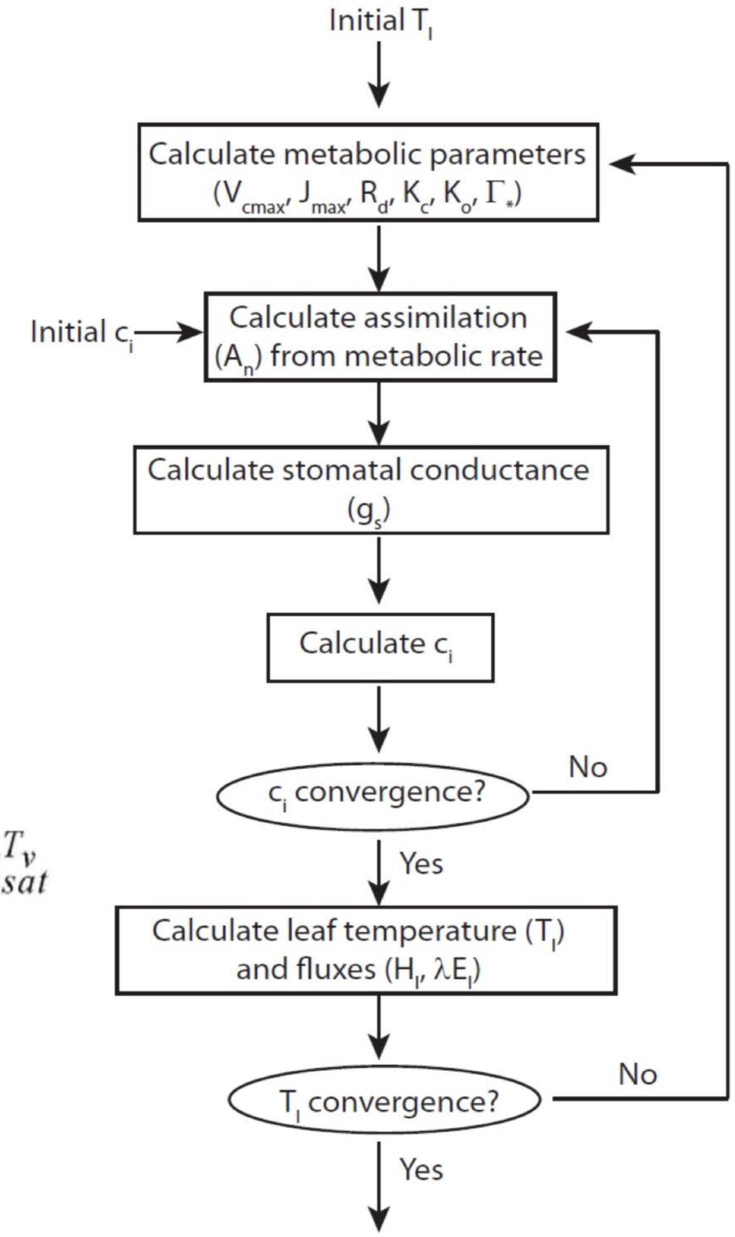
$$g_s = m\frac{A_n}{c_s}h_s p + b$$

2. 模式原理

把光合作用过程与生物物理过程结合在一起

$$-\vec{S}_v + \vec{L}_v\left(T_v\right) + H_v\left(T_v\right) + \lambda E_v\left(T_v\right) = 0$$




Initial T_l

Calculate metabolic parameters (V_{cmax}, J_{max}, R_d, K_c, K_o, Γ_*)

Initial c_i

Calculate assimilation (A_n) from metabolic rate

Calculate stomatal conductance (g_s)

Calculate c_i

c_i convergence?

No

Yes

Calculate leaf temperature (T_l) and fluxes (H_l, λE_l)

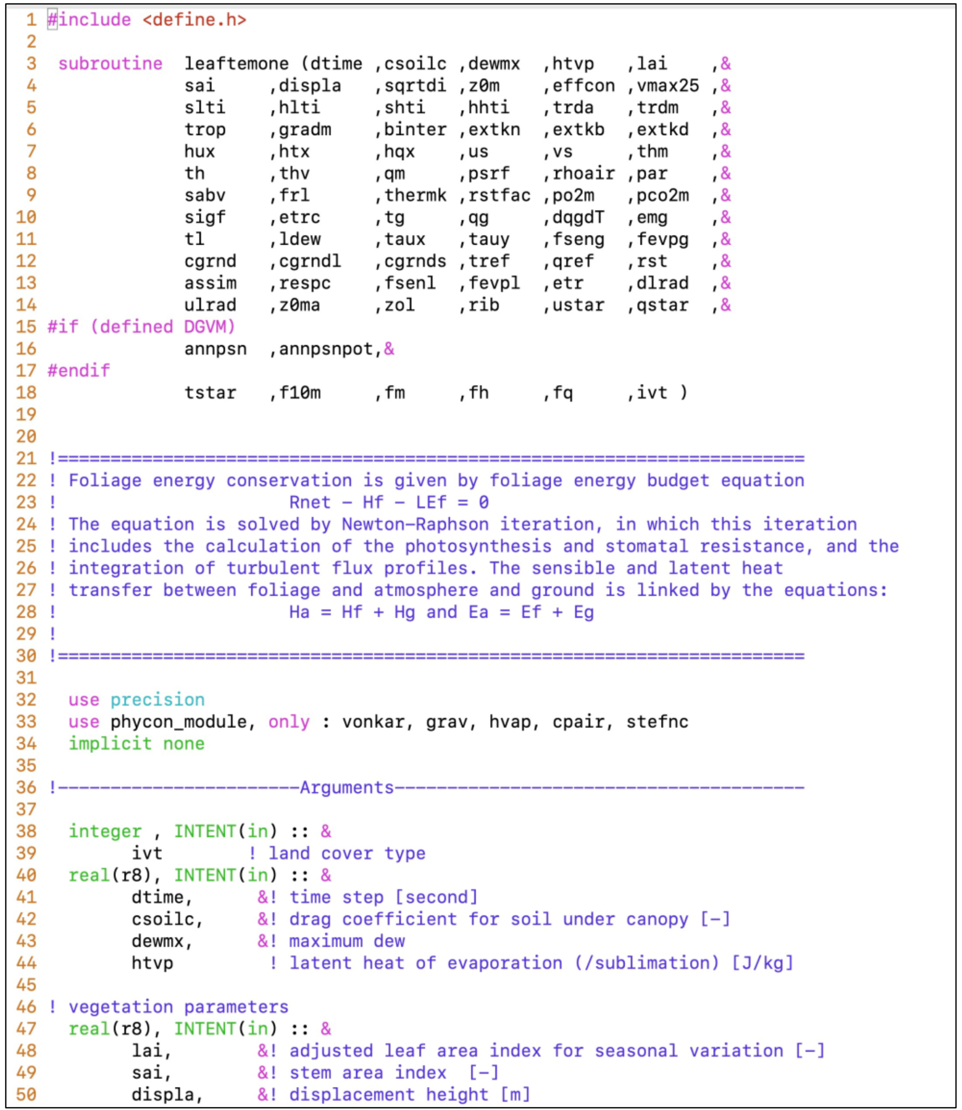
T_l convergence?

No

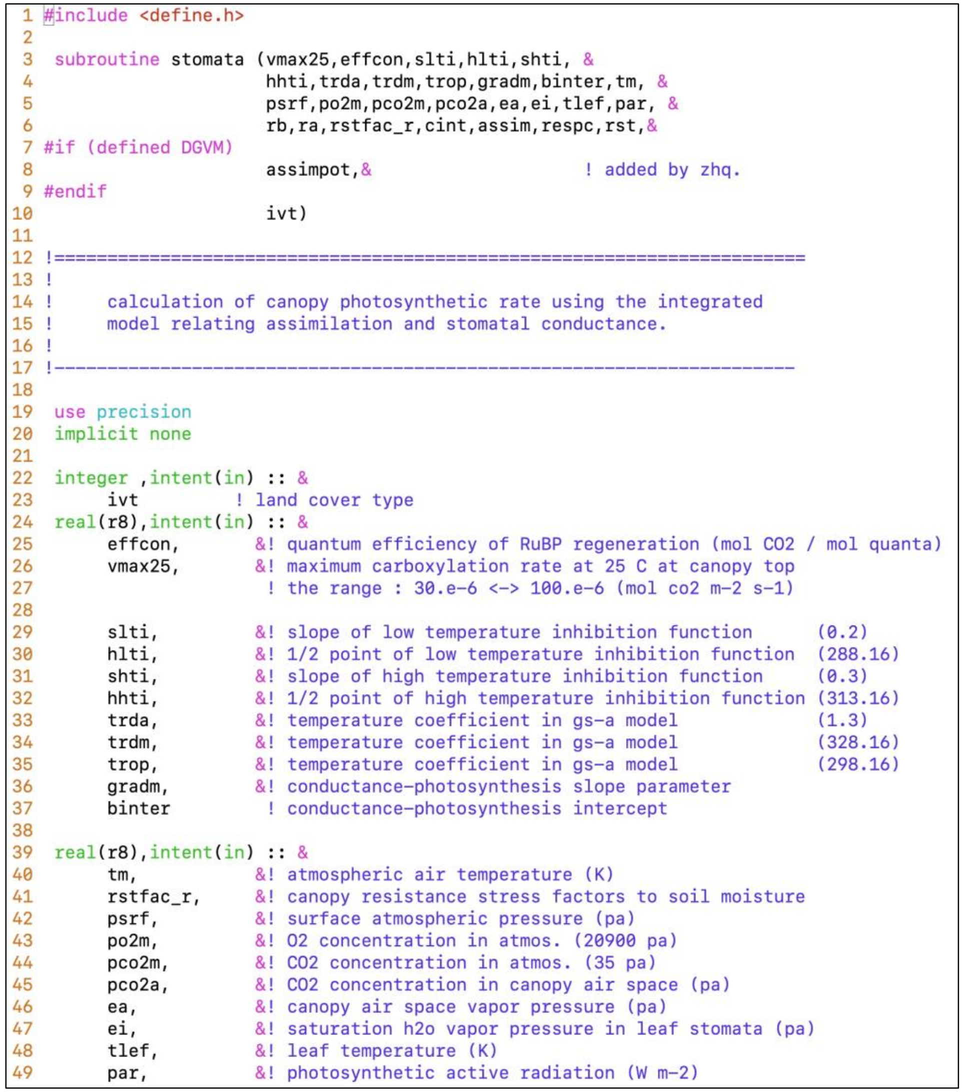
Yes

2. 模式原理

通用陆面过程模式CoLM：植被光合作用与能量平衡模块



```fortran
#include <define.h>

 subroutine  leaftemone (dtime ,csoilc ,dewmx  ,htvp   ,lai    ,&
             sai     ,displa   ,sqrtdi ,z0m    ,effcon ,vmax25 ,&
             slti    ,hlti     ,shti   ,hhti   ,trda   ,trdm   ,&
             trop    ,gradm    ,binter ,extkn  ,extkb  ,extkd  ,&
             hux     ,htx      ,hqx    ,us     ,vs     ,thm    ,&
             th      ,thv      ,qm     ,psrf   ,rhoair ,par    ,&
             sabv    ,frl      ,thermk ,rstfac ,po2m   ,pco2m  ,&
             sigf    ,etrc     ,tg     ,qg     ,dqgdT  ,emg    ,&
             tl      ,ldew     ,taux   ,tauy   ,fseng  ,fevpg  ,&
             cgrnd   ,cgrndl   ,cgrnds ,tref   ,qref   ,rst    ,&
             assim   ,respc    ,fsenl  ,fevpl  ,etr    ,dlrad  ,&
             ulrad   ,z0ma     ,zol    ,rib    ,ustar  ,qstar  ,&
#if (defined DGVM)
             annpsn  ,annpsnpot,&
#endif
             tstar   ,f10m     ,fm     ,fh     ,fq     ,ivt )


!=======================================================================
! Foliage energy conservation is given by foliage energy budget equation
!                      Rnet - Hf - LEf = 0
! The equation is solved by Newton-Raphson iteration, in which this iteration
! includes the calculation of the photosynthesis and stomatal resistance, and the
! integration of turbulent flux profiles. The sensible and latent heat
! transfer between foliage and atmosphere and ground is linked by the equations:
!                      Ha = Hf + Hg and Ea = Ef + Eg
!
!=======================================================================

  use precision
  use phycon_module, only : vonkar, grav, hvap, cpair, stefnc
  implicit none

!-----------------------Arguments---------------------------------------

  integer , INTENT(in) :: &
        ivt          ! land cover type
  real(r8), INTENT(in) :: &
        dtime,      &! time step [second]
        csoilc,     &! drag coefficient for soil under canopy [-]
        dewmx,      &! maximum dew
        htvp         ! latent heat of evaporation (/sublimation) [J/kg]

! vegetation parameters
  real(r8), INTENT(in) :: &
        lai,        &! adjusted leaf area index for seasonal variation [-]
        sai,        &! stem area index  [-]
        displa,     &! displacement height [m]
```



```fortran
#include <define.h>

 subroutine stomata (vmax25,effcon,slti,hlti,shti, &
                     hhti,trda,trdm,trop,gradm,binter,tm, &
                     psrf,po2m,pco2m,pco2a,ea,ei,tlef,par, &
                     rb,ra,rstfac_r,cint,assim,respc,rst,&
#if (defined DGVM)
                     assimpot,&                    ! added by zhq.
#endif
                     ivt)

!=======================================================================
!
!     calculation of canopy photosynthetic rate using the integrated
!     model relating assimilation and stomatal conductance.
!
!----------------------------------------------------------------------

 use precision
 implicit none

 integer ,intent(in) :: &
      ivt           ! land cover type
 real(r8),intent(in) :: &
      effcon,       &! quantum efficiency of RuBP regeneration (mol CO2 / mol quanta)
      vmax25,       &! maximum carboxylation rate at 25 C at canopy top
                     ! the range : 30.e-6 <-> 100.e-6 (mol co2 m-2 s-1)

      slti,         &! slope of low temperature inhibition function      (0.2)
      hlti,         &! 1/2 point of low temperature inhibition function  (288.16)
      shti,         &! slope of high temperature inhibition function     (0.3)
      hhti,         &! 1/2 point of high temperature inhibition function (313.16)
      trda,         &! temperature coefficient in gs-a model             (1.3)
      trdm,         &! temperature coefficient in gs-a model             (328.16)
      trop,         &! temperature coefficient in gs-a model             (298.16)
      gradm,        &! conductance-photosynthesis slope parameter
      binter         ! conductance-photosynthesis intercept

 real(r8),intent(in) :: &
      tm,           &! atmospheric air temperature (K)
      rstfac_r,     &! canopy resistance stress factors to soil moisture
      psrf,         &! surface atmospheric pressure (pa)
      po2m,         &! O2 concentration in atmos. (20900 pa)
      pco2m,        &! CO2 concentration in atmos. (35 pa)
      pco2a,        &! CO2 concentration in canopy air space (pa)
      ea,           &! canopy air space vapor pressure (pa)
      ei,           &! saturation h2o vapor pressure in leaf stomata (pa)
      tlef,         &! leaf temperature (K)
      par,          &! photosynthetic active radiation (W m-2)
```

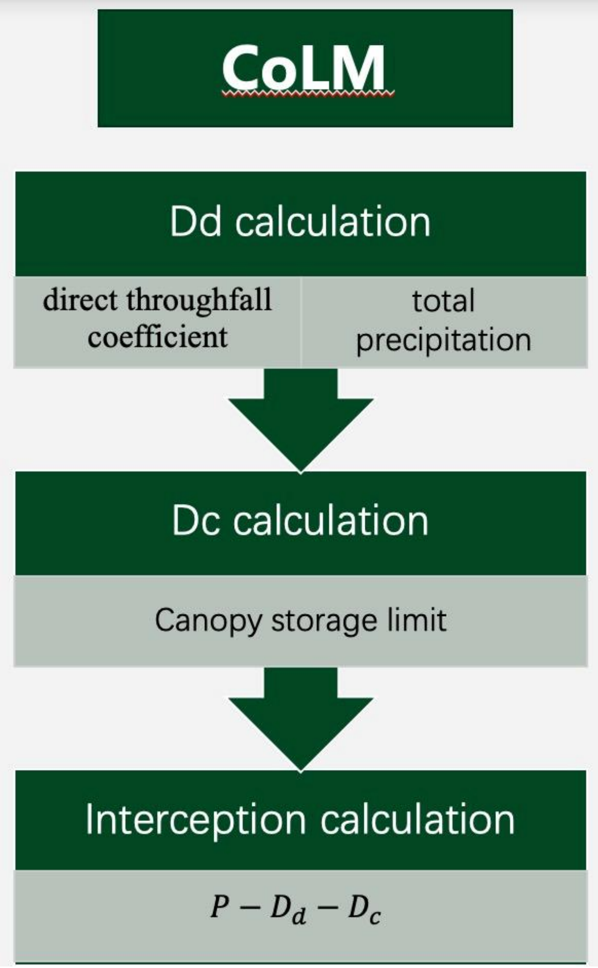
2. 模式原理

通用陆面过程模式CoLM：植被冠层截留模块

求解冠层截留水量平衡方程

$$\frac{\partial M_{cw}}{\partial t} = P - D_d - D_c - E_{ci}/\rho_w$$

D_d （throughfall）是穿透雨量强度（m s^{-1}）； D_c （stemflow）代表树干径流速率（m s^{-1}）；以及 E_{ci} （canopy evaporation）代表叶表面存储冠层截留所产生的蒸发速率（kg m^{-2} s^{-1}）； $\frac{\partial M_{cw}}{\partial t}$ 指的是冠层水的改变量（大于0），即冠层截留量（canopy interception）



CoLM

Dd calculation

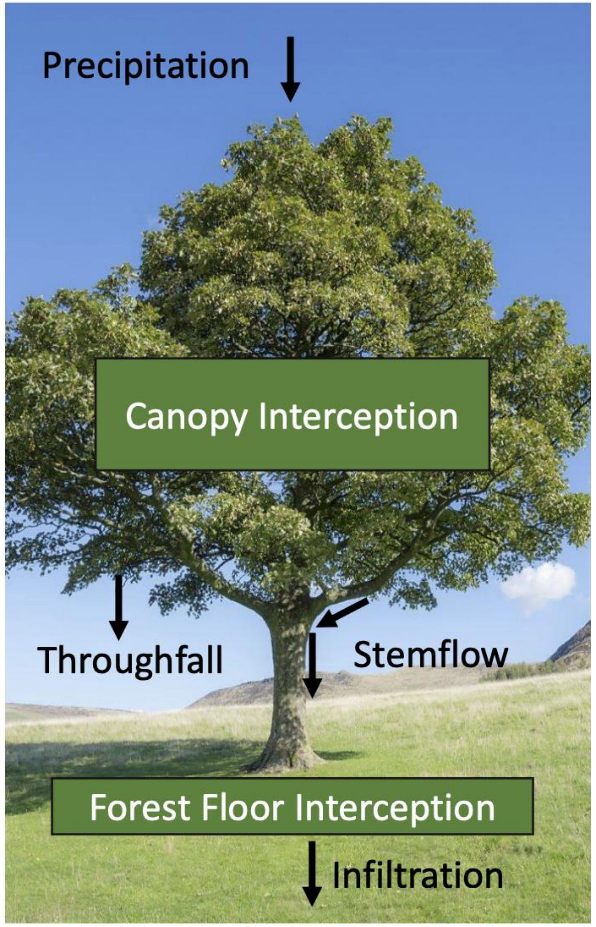
direct throughfall coefficient	total precipitation

Dc calculation

Canopy storage limit

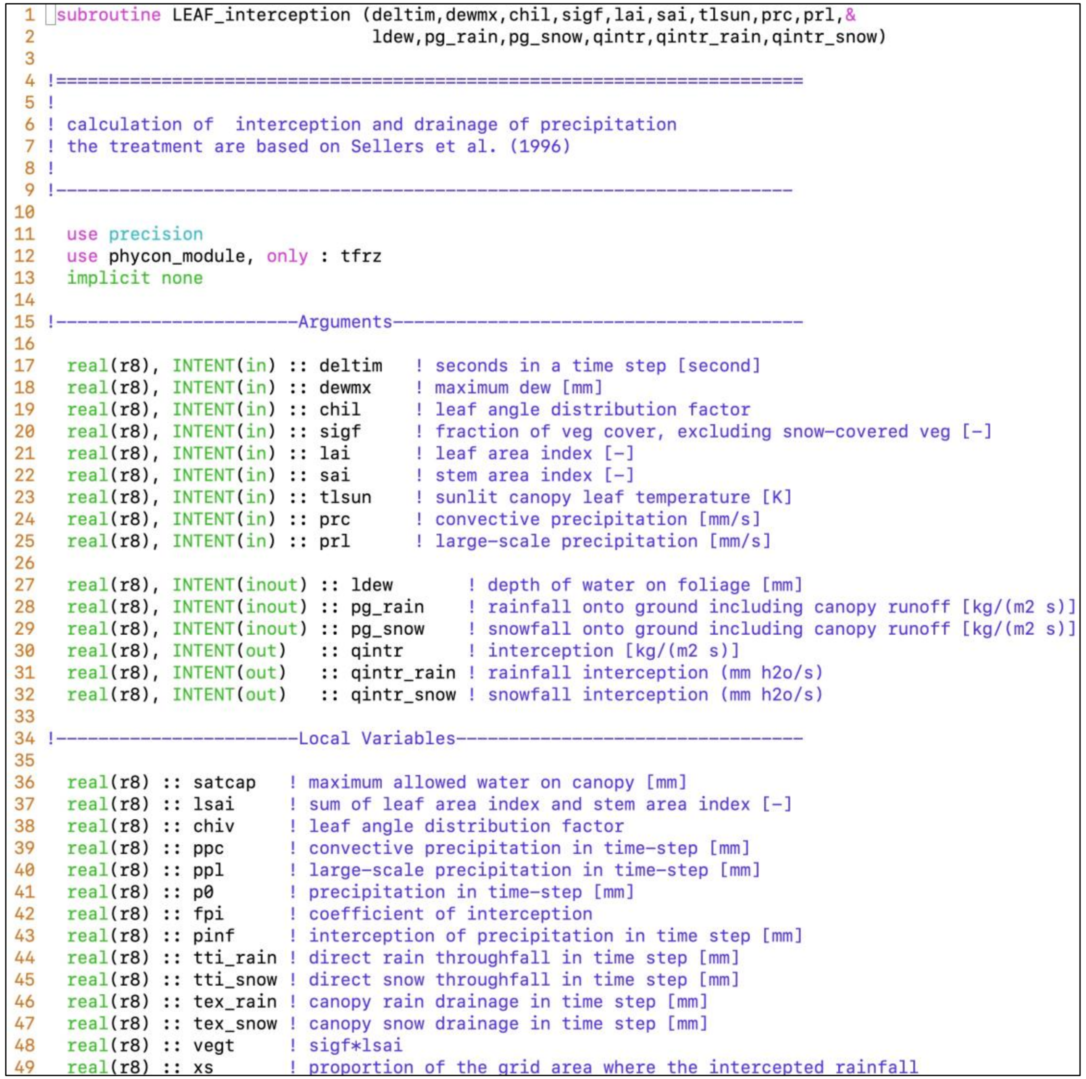
Interception calculation

$P - D_d - D_c$



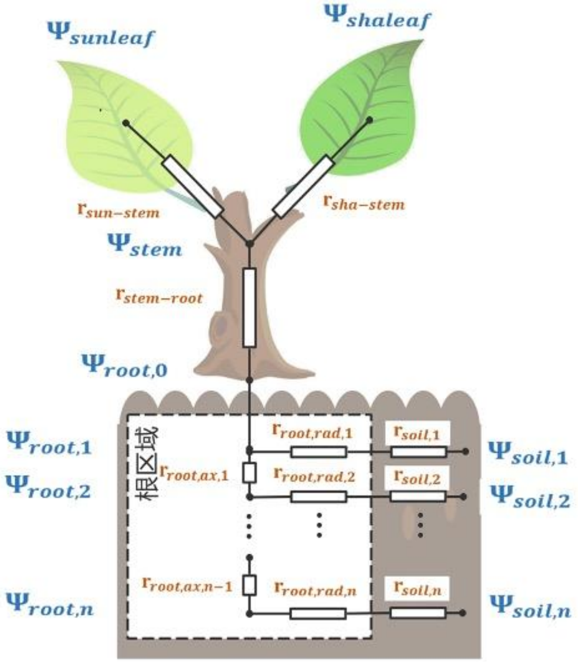
Shuttleworth (2012)

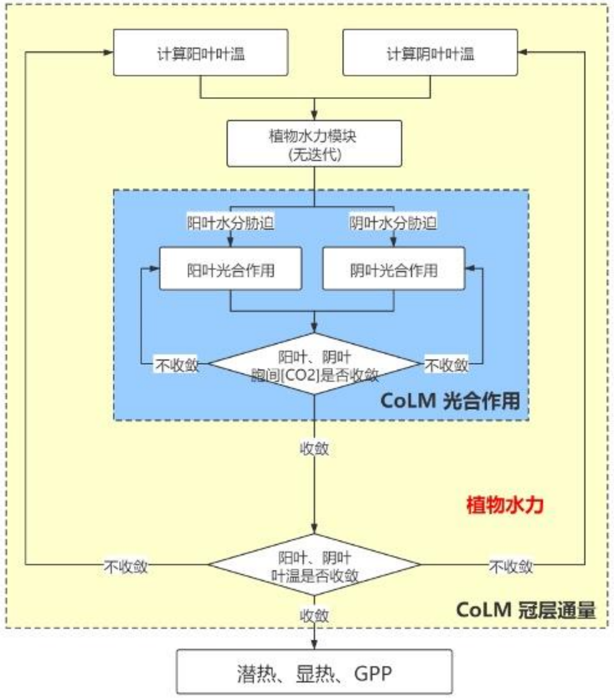
通用陆面过程模式CoLM：植被冠层截留模块



```fortran
subroutine LEAF_interception (deltim,dewmx,chil,sigf,lai,sai,tlsun,prc,prl,&
                              ldew,pg_rain,pg_snow,qintr,qintr_rain,qintr_snow)

!=======================================================================
!
! calculation of  interception and drainage of precipitation
! the treatment are based on Sellers et al. (1996)
!
!-----------------------------------------------------------------------

  use precision
  use phycon_module, only : tfrz
  implicit none

!-----------------------Arguments---------------------------------------

  real(r8), INTENT(in) :: deltim    ! seconds in a time step [second]
  real(r8), INTENT(in) :: dewmx     ! maximum dew [mm]
  real(r8), INTENT(in) :: chil      ! leaf angle distribution factor
  real(r8), INTENT(in) :: sigf      ! fraction of veg cover, excluding snow-covered veg [-]
  real(r8), INTENT(in) :: lai       ! leaf area index [-]
  real(r8), INTENT(in) :: sai       ! stem area index [-]
  real(r8), INTENT(in) :: tlsun     ! sunlit canopy leaf temperature [K]
  real(r8), INTENT(in) :: prc       ! convective precipitation [mm/s]
  real(r8), INTENT(in) :: prl       ! large-scale precipitation [mm/s]

  real(r8), INTENT(inout) :: ldew       ! depth of water on foliage [mm]
  real(r8), INTENT(inout) :: pg_rain    ! rainfall onto ground including canopy runoff [kg/(m2 s)]
  real(r8), INTENT(inout) :: pg_snow    ! snowfall onto ground including canopy runoff [kg/(m2 s)]
  real(r8), INTENT(out)   :: qintr      ! interception [kg/(m2 s)]
  real(r8), INTENT(out)   :: qintr_rain ! rainfall interception (mm h2o/s)
  real(r8), INTENT(out)   :: qintr_snow ! snowfall interception (mm h2o/s)

!-----------------------Local Variables---------------------------------

  real(r8) :: satcap   ! maximum allowed water on canopy [mm]
  real(r8) :: lsai     ! sum of leaf area index and stem area index [-]
  real(r8) :: chiv     ! leaf angle distribution factor
  real(r8) :: ppc      ! convective precipitation in time-step [mm]
  real(r8) :: ppl      ! large-scale precipitation in time-step [mm]
  real(r8) :: p0       ! precipitation in time-step [mm]
  real(r8) :: fpi      ! coefficient of interception
  real(r8) :: pinf     ! interception of precipitation in time step [mm]
  real(r8) :: tti_rain ! direct rain throughfall in time step [mm]
  real(r8) :: tti_snow ! direct snow throughfall in time step [mm]
  real(r8) :: tex_rain ! canopy rain drainage in time step [mm]
  real(r8) :: tex_snow ! canopy snow drainage in time step [mm]
  real(r8) :: vegt     ! sigf*lsai
  real(r8) :: xs       ! proportion of the grid area where the intercepted rainfall
```

通用陆面过程模式CoLM：植被水力模块

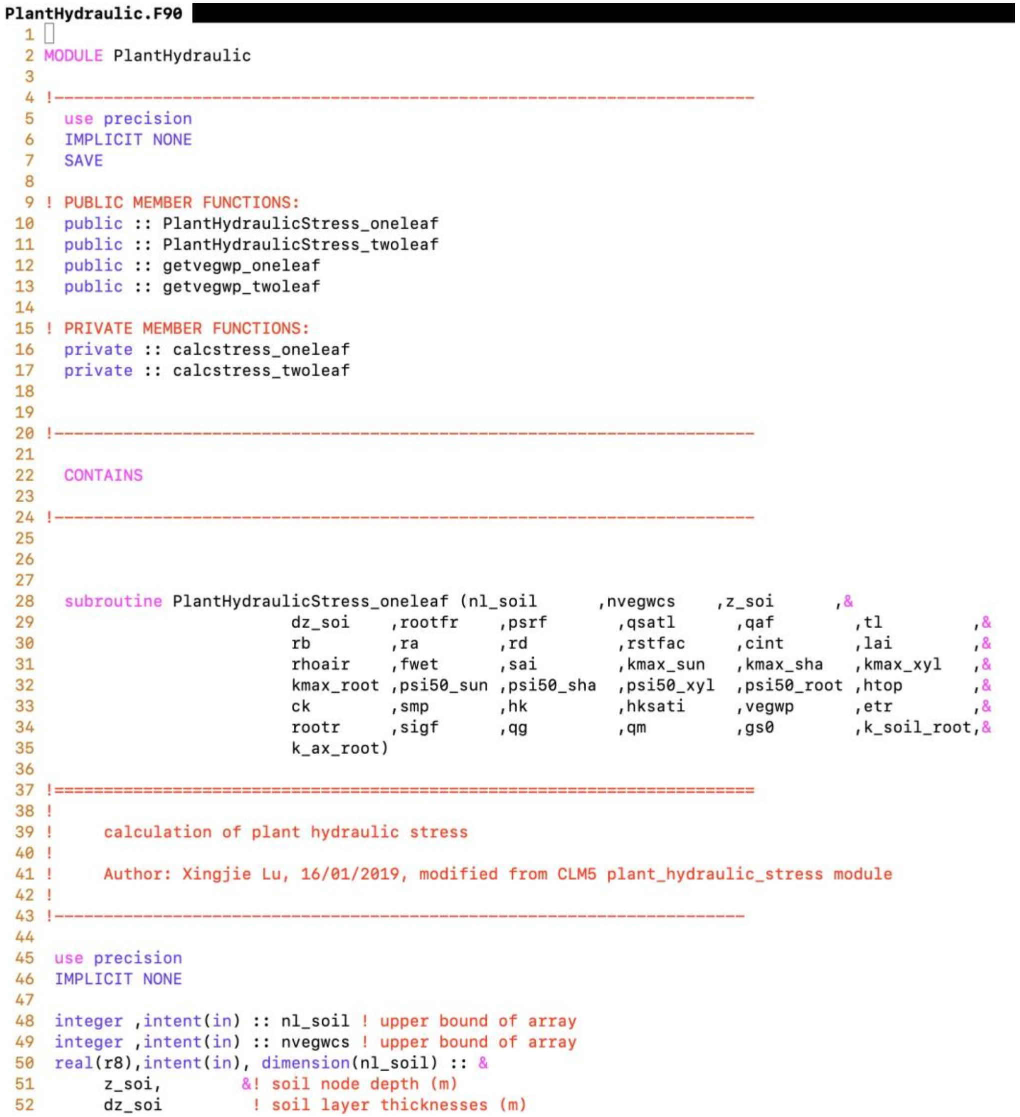




	CoLM	CoLM+PHS
影响气孔导度的水分胁迫	土壤湿度的函数	叶片水势的函数
根吸水	根比例的函数	水力重分配方案

2. 模式原理

通用陆面过程模式CoLM：植被水力模块

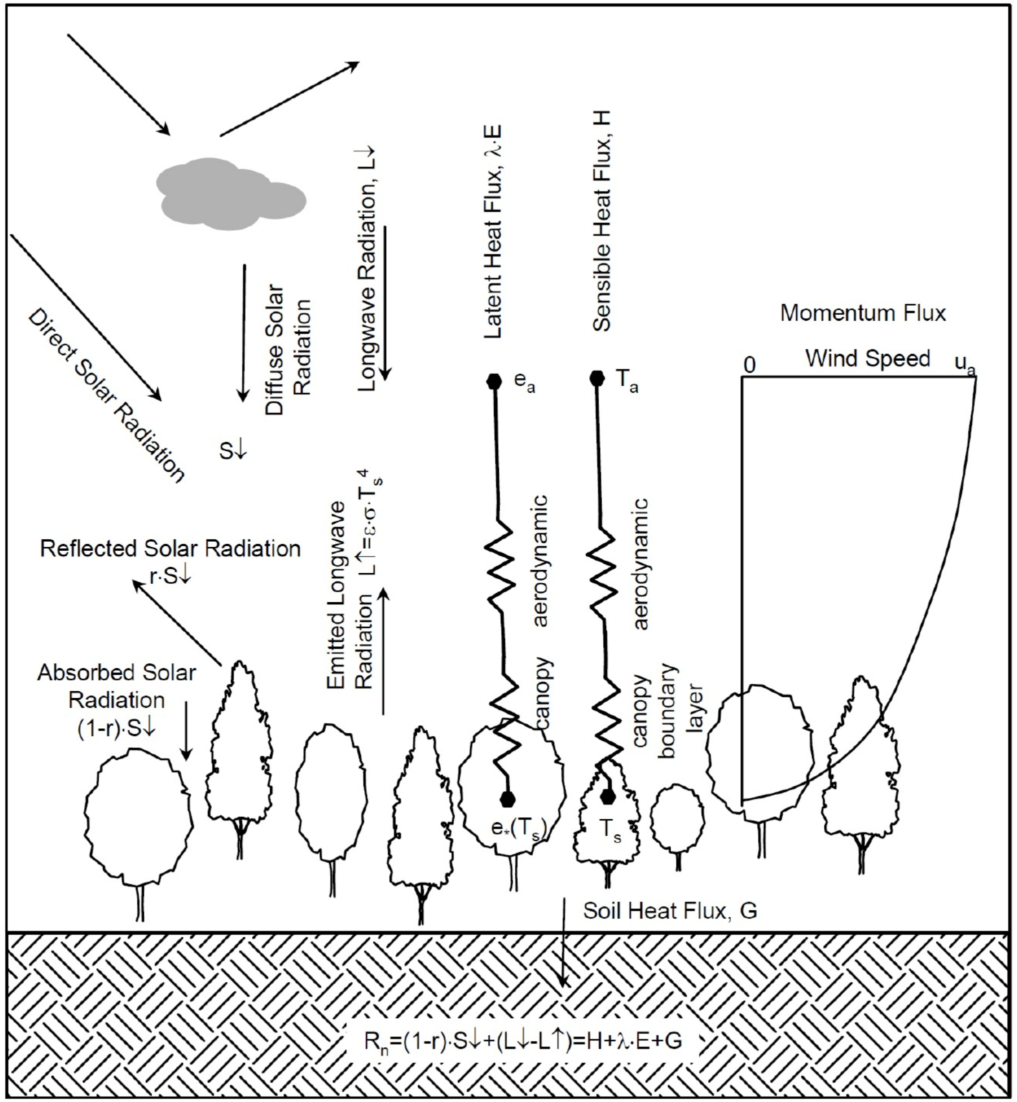


PlantHydraulic.F90 [redacted]

```fortran

MODULE PlantHydraulic

!-----------------------------------------------------------------------
  use precision
  IMPLICIT NONE
  SAVE

! PUBLIC MEMBER FUNCTIONS:
  public :: PlantHydraulicStress_oneleaf
  public :: PlantHydraulicStress_twoleaf
  public :: getvegwp_oneleaf
  public :: getvegwp_twoleaf

! PRIVATE MEMBER FUNCTIONS:
  private :: calcstress_oneleaf
  private :: calcstress_twoleaf


!-----------------------------------------------------------------------

  CONTAINS

!-----------------------------------------------------------------------



  subroutine PlantHydraulicStress_oneleaf (nl_soil      ,nvegwcs    ,z_soi      ,&
                         dz_soi    ,rootfr    ,psrf        ,qsatl      ,qaf        ,tl          ,&
                         rb        ,ra        ,rd          ,rstfac     ,cint       ,lai         ,&
                         rhoair    ,fwet      ,sai         ,kmax_sun   ,kmax_sha   ,kmax_xyl    ,&
                         kmax_root ,psi50_sun ,psi50_sha   ,psi50_xyl  ,psi50_root ,htop        ,&
                         ck        ,smp       ,hk          ,hksati     ,vegwp      ,etr         ,&
                         rootr     ,sigf      ,qg          ,qm         ,gs0        ,k_soil_root,&
                         k_ax_root)

!=======================================================================
!
!     calculation of plant hydraulic stress
!
!     Author: Xingjie Lu, 16/01/2019, modified from CLM5 plant_hydraulic_stress module
!
!----------------------------------------------------------------------

 use precision
 IMPLICIT NONE

 integer ,intent(in) :: nl_soil ! upper bound of array
 integer ,intent(in) :: nvegwcs ! upper bound of array
 real(r8),intent(in), dimension(nl_soil) :: &
      z_soi,        &! soil node depth (m)
      dz_soi         ! soil layer thicknesses (m)
```

2. 模式原理

通用陆面过程模式CoLM：近地面能量平衡模块

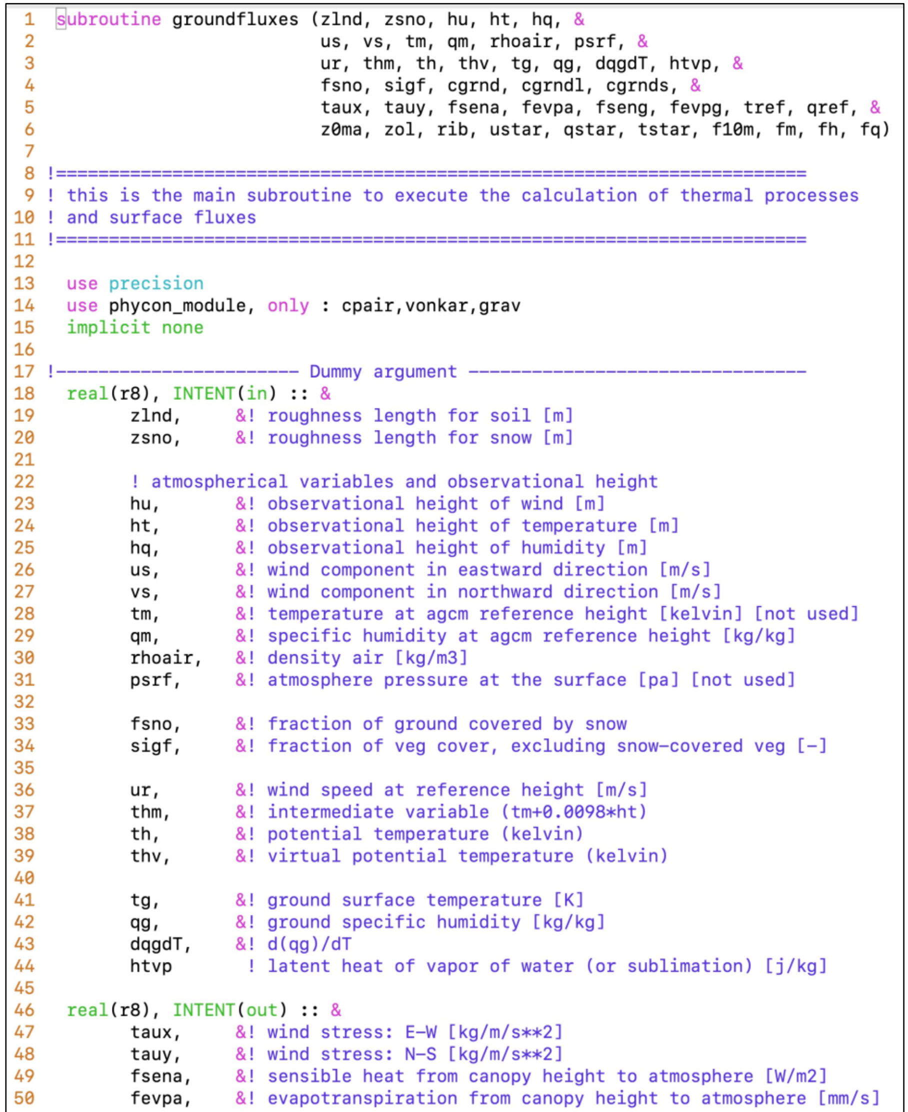


Surface energy balance (and other energy balances, e.g. in canopy, snow, soil)

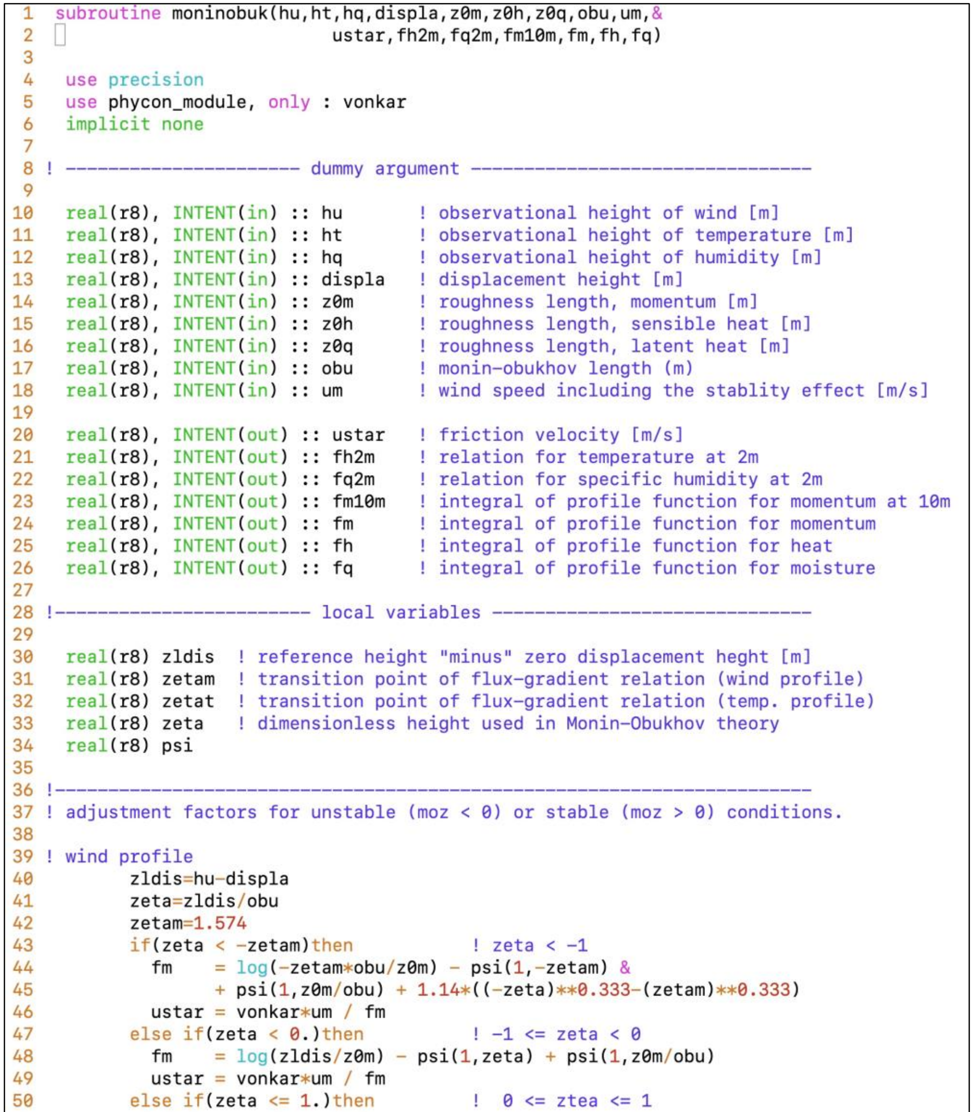
- $S^{\downarrow} + L^{\downarrow} = S^{\uparrow} + L^{\uparrow} + \lambda E + H + G$
  - $S^{\downarrow}$, $S^{\uparrow}$ are down(up)welling solar radiation
  - $L^{\downarrow}$, $L^{\uparrow}$ are down(up)welling longwave radiation
  - λ is latent heat of vaporization, E is evaporation
  - H is sensible heat flux
  - G is ground heat flux

2. 模式原理

通用陆面过程模式CoLM：近地面能量平衡模块



```fortran
subroutine groundfluxes (zlnd, zsno, hu, ht, hq, &
                         us, vs, tm, qm, rhoair, psrf, &
                         ur, thm, th, thv, tg, qg, dqgdT, htvp, &
                         fsno, sigf, cgrnd, cgrndl, cgrnds, &
                         taux, tauy, fsena, fevpa, fseng, fevpg, tref, qref, &
                         z0ma, zol, rib, ustar, qstar, tstar, f10m, fm, fh, fq)

!========================================================================
! this is the main subroutine to execute the calculation of thermal processes
! and surface fluxes
!========================================================================

 use precision
 use phycon_module, only : cpair,vonkar,grav
 implicit none

!----------------------- Dummy argument --------------------------------
 real(r8), INTENT(in) :: &
       zlnd,     &! roughness length for soil [m]
       zsno,     &! roughness length for snow [m]

       ! atmospherical variables and observational height
       hu,       &! observational height of wind [m]
       ht,       &! observational height of temperature [m]
       hq,       &! observational height of humidity [m]
       us,       &! wind component in eastward direction [m/s]
       vs,       &! wind component in northward direction [m/s]
       tm,       &! temperature at agcm reference height [kelvin] [not used]
       qm,       &! specific humidity at agcm reference height [kg/kg]
       rhoair,   &! density air [kg/m3]
       psrf,     &! atmosphere pressure at the surface [pa] [not used]

       fsno,     &! fraction of ground covered by snow
       sigf,     &! fraction of veg cover, excluding snow-covered veg [-]

       ur,       &! wind speed at reference height [m/s]
       thm,      &! intermediate variable (tm+0.0098*ht)
       th,       &! potential temperature (kelvin)
       thv,      &! virtual potential temperature (kelvin)

       tg,       &! ground surface temperature [K]
       qg,       &! ground specific humidity [kg/kg]
       dqgdT,    &! d(qg)/dT
       htvp       ! latent heat of vapor of water (or sublimation) [j/kg]

 real(r8), INTENT(out) :: &
       taux,     &! wind stress: E-W [kg/m/s**2]
       tauy,     &! wind stress: N-S [kg/m/s**2]
       fsena,    &! sensible heat from canopy height to atmosphere [W/m2]
       fevpa,    &! evapotranspiration from canopy height to atmosphere [mm/s]
```



```fortran
subroutine moninobuk(hu,ht,hq,displa,z0m,z0h,z0q,obu,um,&
                          ustar,fh2m,fq2m,fm10m,fm,fh,fq)

 use precision
 use phycon_module, only : vonkar
 implicit none

! ---------------------- dummy argument --------------------------------

 real(r8), INTENT(in) :: hu       ! observational height of wind [m]
 real(r8), INTENT(in) :: ht       ! observational height of temperature [m]
 real(r8), INTENT(in) :: hq       ! observational height of humidity [m]
 real(r8), INTENT(in) :: displa   ! displacement height [m]
 real(r8), INTENT(in) :: z0m      ! roughness length, momentum [m]
 real(r8), INTENT(in) :: z0h      ! roughness length, sensible heat [m]
 real(r8), INTENT(in) :: z0q      ! roughness length, latent heat [m]
 real(r8), INTENT(in) :: obu      ! monin-obukhov length (m)
 real(r8), INTENT(in) :: um       ! wind speed including the stablity effect [m/s]

 real(r8), INTENT(out) :: ustar   ! friction velocity [m/s]
 real(r8), INTENT(out) :: fh2m    ! relation for temperature at 2m
 real(r8), INTENT(out) :: fq2m    ! relation for specific humidity at 2m
 real(r8), INTENT(out) :: fm10m   ! integral of profile function for momentum at 10m
 real(r8), INTENT(out) :: fm      ! integral of profile function for momentum
 real(r8), INTENT(out) :: fh      ! integral of profile function for heat
 real(r8), INTENT(out) :: fq      ! integral of profile function for moisture

!------------------------ local variables ------------------------------

 real(r8) zldis  ! reference height "minus" zero displacement heght [m]
 real(r8) zetam  ! transition point of flux-gradient relation (wind profile)
 real(r8) zetat  ! transition point of flux-gradient relation (temp. profile)
 real(r8) zeta   ! dimensionless height used in Monin-Obukhov theory
 real(r8) psi

!-----------------------------------------------------------------------
! adjustment factors for unstable (moz < 0) or stable (moz > 0) conditions.

! wind profile
       zldis=hu-displa
       zeta=zldis/obu
       zetam=1.574
       if(zeta < -zetam)then           ! zeta < -1
         fm    = log(-zetam*obu/z0m) - psi(1,-zetam) &
               + psi(1,z0m/obu) + 1.14*((-zeta)**0.333-(zetam)**0.333)
         ustar = vonkar*um / fm
       else if(zeta < 0.)then          ! -1 <= zeta < 0
         fm    = log(zldis/z0m) - psi(1,zeta) + psi(1,z0m/obu)
         ustar = vonkar*um / fm
       else if(zeta <= 1.)then         !  0 <= ztea <= 1
```

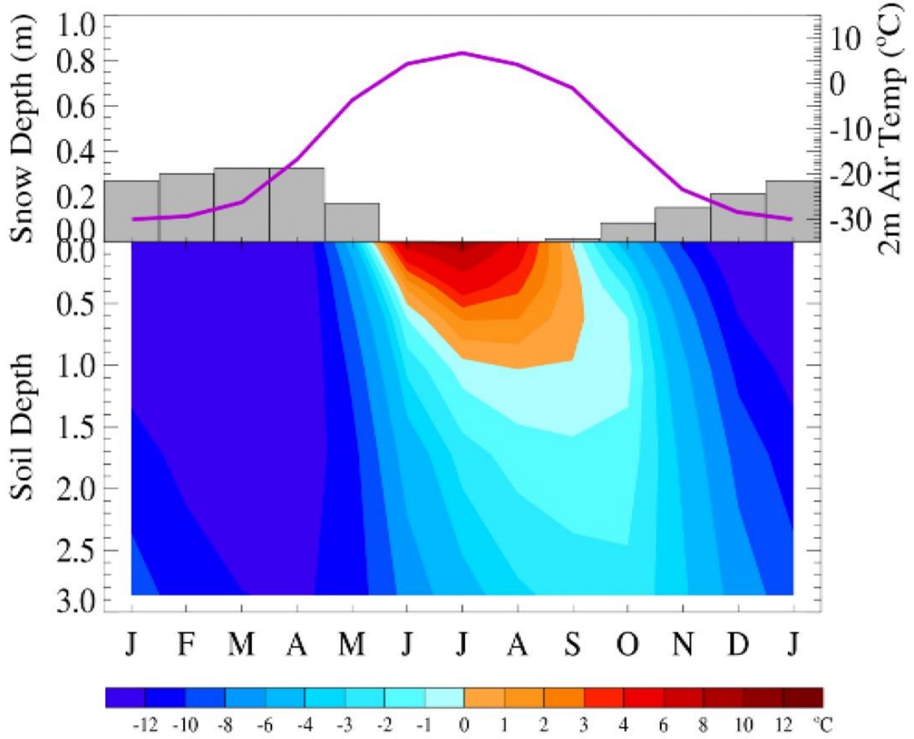
2. 模式原理

通用陆面过程模式CoLM：土壤雪盖热力过程模块

Solve the heat diffusion equation for multi-layer snow and soil model

$$C_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(K \frac{\partial T}{\partial z} \right)$$

- where C_p (heat capacity) and K (thermal conductivity) are functions of:
  - temperature
  - total soil moisture
  - soil texture
  - ice/liquid content



2. 模式原理

通用陆面过程模式CoLM：土壤雪盖热力过程模块

```
#include <define.h>
 subroutine groundtem (itypwat,lb,nl_soil,dtime,n_pft,num_filterp,filterp,wt_patch, &
                       wt_column,capr,cnfac,csol,porsl,dkmg,dkdry,dksatu, &
                       sigf,dz,z,zi,tss,
                       frl,dlrad,sabg,fs
#if (defined FHNP) && (defined FTF)
                       frostdp, thawdp,
                       frost_day, thaw_
#endif
                       imelt,sm,xmf,fact
! Snow and soil temperatures
! o The volumetric heat capacity is calc
!   in terms of the volumetric fraction
! o The thermal conductivity of soil is
!   the algorithm of Johansen (as report
!   the formulation used in SNTHERM (Jo
! o Boundary conditions:
!   F = Rnet - Hg - LEg (top),  F= 0 (ba
! o Soil / snow temperature is predicted
!   in 10 soil layers and up to 5 snow
!   The thermal conductivities at the in
!   (j, j+1) are derived from an assumpt
!   is equal to that from the node j to
!   interface to the node j+1. The equat
!   method and resulted in a tridiagonal
!
!=====================================

  use precision
  use phycon_module, only : stefnc,tfrz
!use debug
  implicit none
  integer, INTENT(in) :: lb
  integer, INTENT(in) :: nl_soil
  integer, INTENT(in) :: n_pft
  integer, INTENT(in) :: itypwat

  integer, INTENT(in) :: num_filterp
  integer, INTENT(in) :: filterp(n_pft)
  real(r8), INTENT(in) :: wt_patch(n_pft
  real(r8), INTENT(in) :: wt_column

  real(r8), INTENT(in) :: dtime    !mode
  real(r8), INTENT(in) :: capr     !tuni
  real(r8), INTENT(in) :: cnfac    !Cran

  real(r8), INTENT(in) :: csol(1:nl_soi
  real(r8), INTENT(in) :: porsl(1:nl_so
  real(r8), INTENT(in) :: dkmg(1:nl_soi
  real(r8), INTENT(in) :: dkdry(1:nl_so
```

```
subroutine meltf (lb,nl_soil,deltim, &
                  fact,brr,hs,dhsdT, &
                  t_soisno_bef,t_soisno,wliq_soisno,wice_soisno,imelt, &
                  scv,sn

!------------------------
!
! calculation of the phas
!
! (1) check the condition
!     i.e., the layer tem
!     and the ice mass is
!     or layer temperatur
!     and the liquid wate
! (2) assess the rate of
!     after setting the l
! (3) re-adjust the ice a
!
!------------------------

  use precision
  use phycon_module, only
  IMPLICIT NONE

!------------------------

   integer, INTENT(in) ::
   integer, INTENT(in) ::
  real(r8), INTENT(in) ::
  real(r8), INTENT(in) ::
  real(r8), INTENT(in) ::
  real(r8), INTENT(in) ::
  real(r8), INTENT(in) ::
  real(r8), INTENT(in) ::

  real(r8), INTENT(inout)
  real(r8), INTENT(inout)
  real(r8), INTENT(inout)
  real(r8), INTENT(inout)
  real(r8), INTENT(inout)

  real(r8), INTENT(out) :
  real(r8), INTENT(out) :
   integer, INTENT(out) :

! Local
  real(r8) :: hm(lb:nl_so
  real(r8) :: xm(lb:nl_so
  real(r8) :: heatr
  real(r8) :: temp1
```

```
 subroutine hCapacity (itypwat,lb,nl_soil,csol,porsl,wice,wliq,scv,dz,

!-------------------------------------------------------------------------
! calculation of heat capacities of snow / soil layers
! the volumetric heat capacity is calculated as a linear combination
! in terms of the volumetric fraction of the constituent phases.
!-------------------------------------------------------------------------

  use precision
  use phycon_module, only : cpice,cpliq
  implicit none

  integer, INTENT(in) :: lb               ! lower bound of array
  integer, INTENT(in) :: nl_soil          ! upper bound of array
  integer, INTENT(in) :: itypwat          ! land water type (0=soil, 1=urba
  real(r8), INTENT(in) :: csol(1:nl_soil)  ! heat capacity of soil soil
  real(r8), INTENT(in) :: porsl(1:nl_soil) ! soil porosity
  real(r8), INTENT(in) :: wice(lb:nl_soil) ! ice lens [kg/m2]
  real(r8), INTENT(in) :: wliq(lb:nl_soil) ! liqui water [kg/m2]
  real(r8), INTENT(in) :: dz(lb:nl_soil)   ! layer thickiness [m]
  real(r8), INTENT(in) :: scv              ! snow water equivalent [mm]
  real(r8), INTENT(out) :: cv(lb:nl_soil)  ! heat capacity [J/(m2 K)]

!-------------------------------------------------------------------------
! Soil heat capacity, which from de Vires (1963)

      if(itypwat<=1 .or. itypwat==4 .or. itypwat==5 )then ! soil ground
         cv(1:) = csol(1:)*(1.-porsl(1:))*dz(1:) + wice(1:)*cpice + wli
      else                          ! wetland or glacier
         cv(1:) = wice(1:)*cpice + wliq(1:)*cpliq
      endif
      if(lb==1 .AND. scv>0.) cv(1) = cv(1) + cpice*scv

! Snow heat capacity
      if(lb<1)then
        cv(:0) = cpliq*wliq(:0) + cpice*wice(:0)
      endif

 end subroutine hCapacity
```

```
#include <define.h>
#if (defined FHNP) && (defined FTF)
 subroutine hConductivity (itypwat,lb,nl_soil,&
                           dkmg,dkdry,dksatu,porsl,dz,z,zi,tss,wice,wliq,tk,thkhc)
#else
subroutine hConductivity (itypwat,lb,nl_soil,&
                          dkmg,dkdry,dksatu,porsl,dz,z,zi,tss,wice,wliq,tk)
#endif

!-----------------------------------------------------------------------
! calculation of thermal conductivities of snow / soil layers
! The thermal conductivity of soil is computed from
! the algorithm of Johansen (as reported by Farouki 1981), and of snow is from
! the formulation used in SNTHERM (Jordan 1991).
!
! The thermal conductivities at the interfaces between two neighbor layers
! (j, j+1) are derived from an assumption that the flux across the interface
! is equal to that from the node j to the interface and the flux from the
! interface to the node j+1.
!-----------------------------------------------------------------------

  use precision
  use phycon_module, only : denh2o,denice,tfrz,tkwat,tkice,tkair
  implicit none

  integer, INTENT(in) :: lb              ! lower bound of array
  integer, INTENT(in) :: nl_soil         ! upper bound of array
  integer, INTENT(in) :: itypwat         ! land water type (0=soil, 1=urban, 2=wetland,
                                         ! 3=land ice, 4=deep lake, 5=shallow lake)
  real(r8), INTENT(in) ::   dkmg(1:nl_soil)  ! dkm**dmvol, where dkm is the mineral cond.
  real(r8), INTENT(in) ::  dkdry(1:nl_soil)  ! thermal conductivity for dry soil [W/m-K]
  real(r8), INTENT(in) :: dksatu(1:nl_soil)  ! Thermal conductivity of saturated soil [W/m-K]
  real(r8), INTENT(in) ::  porsl(1:nl_soil)  ! fractional volume between soil grains=1.-dmvol
  real(r8), INTENT(in) ::   dz(lb:nl_soil)   ! layer thickiness [m]
  real(r8), INTENT(in) ::    z(lb:nl_soil)   ! node depth [m]
  real(r8), INTENT(in) ::   zi(lb-1:nl_soil) ! interface depth [m]
  real(r8), INTENT(in) ::  tss(lb:nl_soil)   ! Nodal temperature [K]
  real(r8), INTENT(in) :: wice(lb:nl_soil)   ! ice lens [kg/m2]
  real(r8), INTENT(in) :: wliq(lb:nl_soil)   ! liqui water [kg/m2]
  real(r8), INTENT(out) :: tk(lb:nl_soil)    ! thermal conductivity [W/(m K)]
!====liruichao add====
#if (defined FHNP) && (defined FTF)
  real(r8), INTENT(out) :: thkhc(lb:nl_soil)
#endif
!====liruichao add====
! local
  real(r8) rhosnow      ! partitial density of water (ice + liquid)
  real(r8) dksat        ! thermal conductivity for saturated soil (j/(k s m))
  real(r8) dke          ! kersten number
  real(r8) fl           ! fraction of liquid or unfrozen water to total water
```

通用陆面过程模式CoLM：土壤雪盖垂直水分运动模块

- 土壤水的状态和垂直运动主要取决于地表入渗、地下径流、土水势的梯度、重力和植被根部对水的吸收等过程。模式中仅考虑土壤水在垂直方向上的运动。
- 在非饱和区域，土壤水通量用**Buckingham-Darcy定律**来描述，

$$q=-K\frac{\partial}{\partial z}(\psi-z)$$

其中，z为土壤深度，取土壤表面为0，垂直向下为正方向，ψ为土壤水势，K为土壤导水率。

- 根据质量守恒定律，可得一维土壤水的垂直运动方程（**Richards方程**），

$$\frac{\partial\theta}{\partial t}=-\frac{\partial q}{\partial z}-Q$$

其中Q为土壤水的汇，主要为由于根系吸水作用（蒸腾）而减少的水分。

通用陆面过程模式CoLM：土壤雪盖垂直水分运动模块

- 土壤导水率K和土水势ψ与土壤含水量和土壤质地有关，基于

Clapp and Hornberger (1978) 的工作，

$$\psi = \psi_s \left(\frac{\theta}{\theta_{sat}}\right)^{-b}, \qquad K = K_{sat}\left(\frac{\theta}{\theta_{sat}}\right)^{2b+3}$$

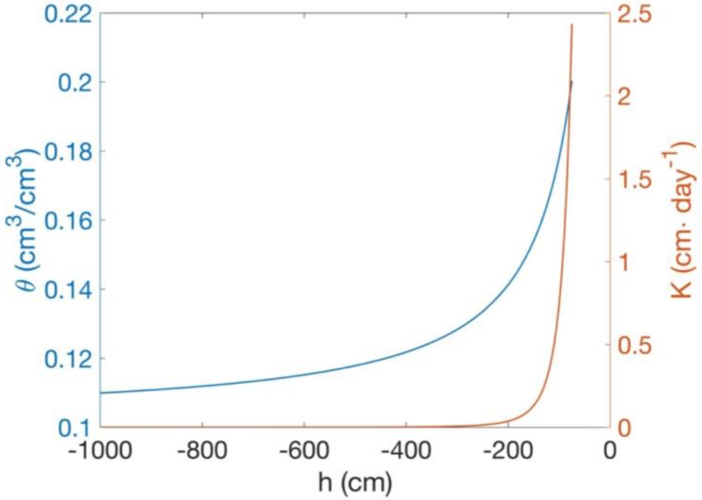
其中ψ_s为饱和土水势，θ_{sat}为饱和土壤水体积含量，K_{sat}为饱和导水率，θ为土壤水体积含量，b为土壤孔隙大小分布指数的倒数，参数$\psi_s, \theta_{sat}, K_{sat}, b$均依赖于土壤质地。

van Genuchten & Mualem 公式：

$$m = 1 - 1/n$$

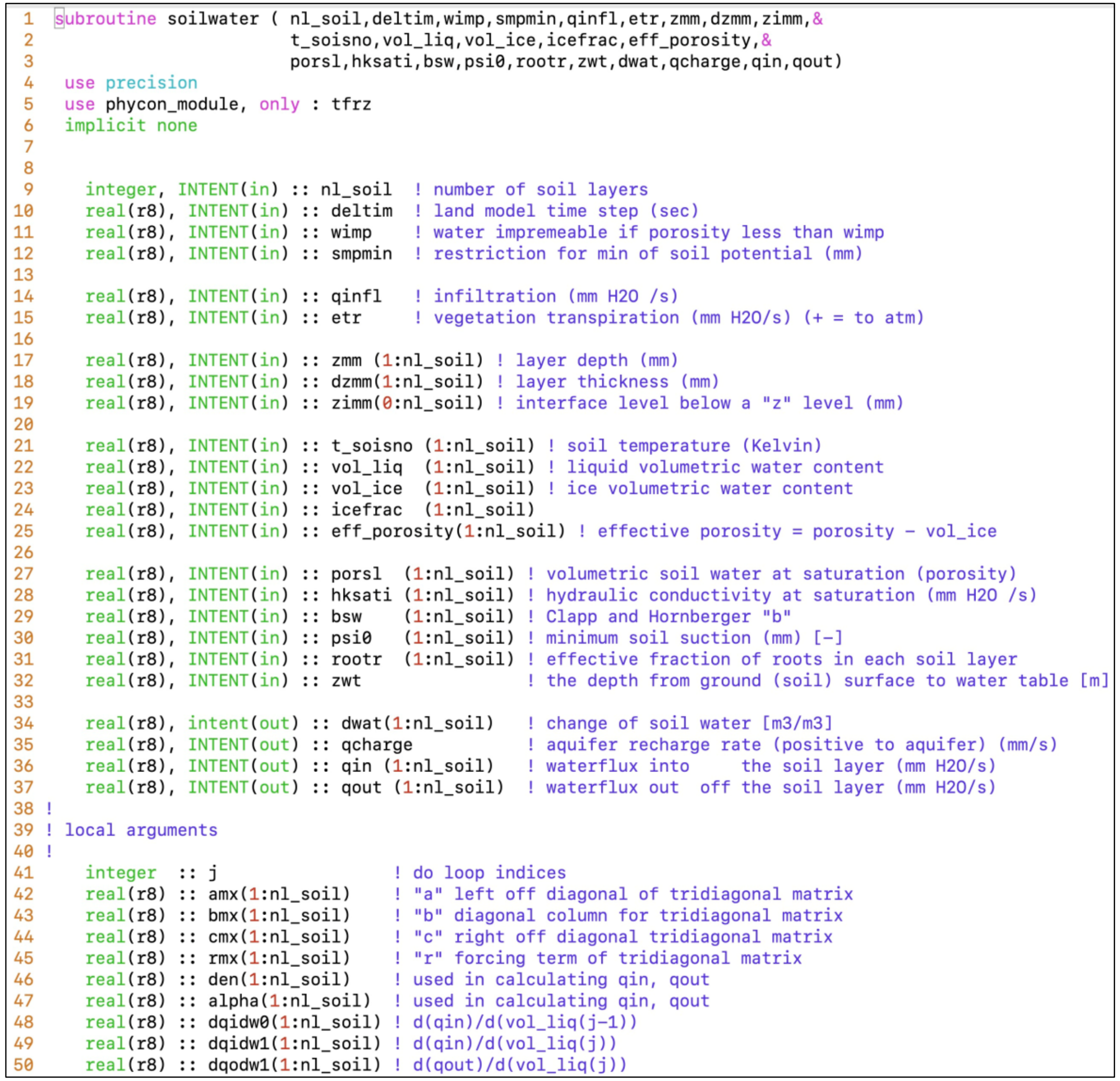
$$\Theta = \frac{\theta - \theta_r}{\theta_s - \theta_r} = \left[\frac{1}{1 + (\alpha h)^n}\right]^m$$

$$k = k_s \Theta^L \left[1 - \left(1 - \Theta^{1/m}\right)^m\right]^2$$

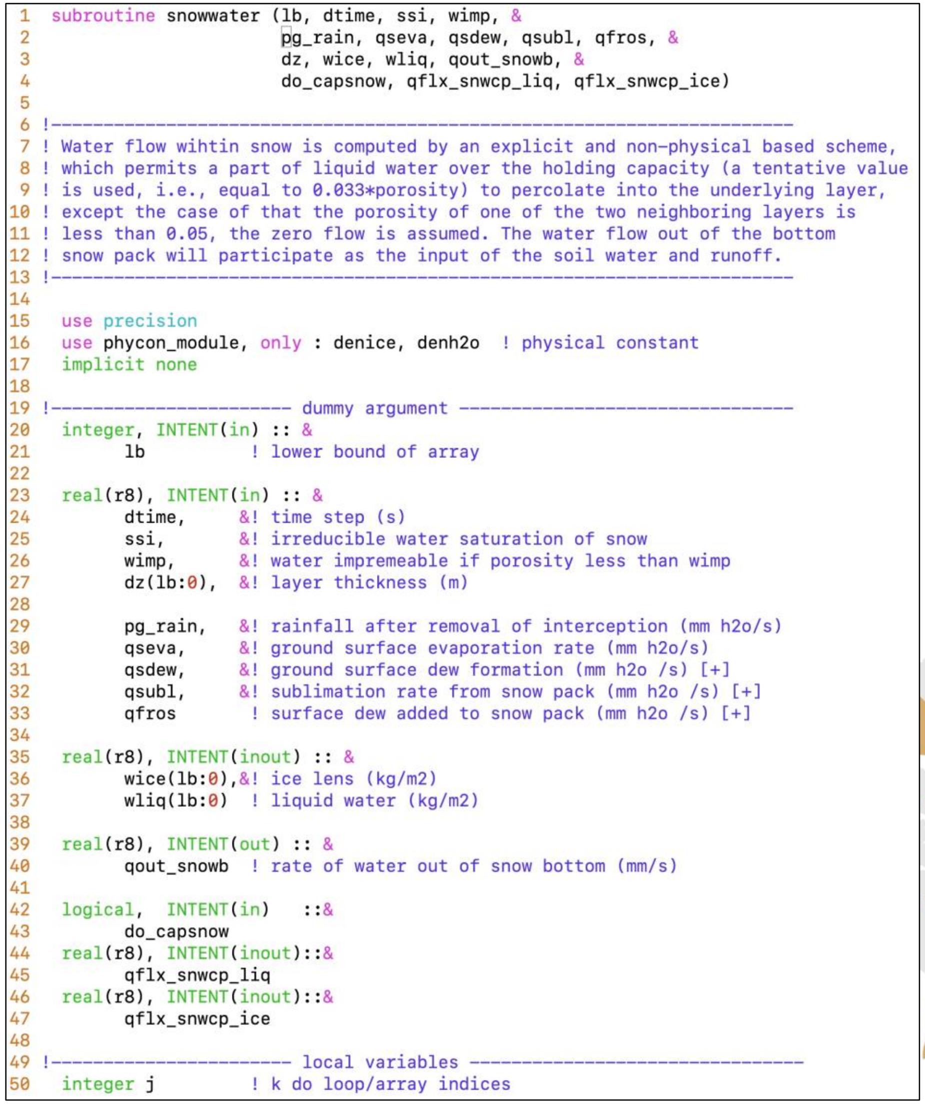


2. 模式原理

通用陆面过程模式CoLM：土壤雪盖垂直水分运动模块



```fortran
subroutine soilwater ( nl_soil,deltim,wimp,smpmin,qinfl,etr,zmm,dzmm,zimm,&
                       t_soisno,vol_liq,vol_ice,icefrac,eff_porosity,&
                       porsl,hksati,bsw,psi0,rootr,zwt,dwat,qcharge,qin,qout)
 use precision
 use phycon_module, only : tfrz
 implicit none


   integer, INTENT(in) :: nl_soil  ! number of soil layers
   real(r8), INTENT(in) :: deltim  ! land model time step (sec)
   real(r8), INTENT(in) :: wimp    ! water impremeable if porosity less than wimp
   real(r8), INTENT(in) :: smpmin  ! restriction for min of soil potential (mm)

   real(r8), INTENT(in) :: qinfl   ! infiltration (mm H2O /s)
   real(r8), INTENT(in) :: etr     ! vegetation transpiration (mm H2O/s) (+ = to atm)

   real(r8), INTENT(in) :: zmm (1:nl_soil) ! layer depth (mm)
   real(r8), INTENT(in) :: dzmm(1:nl_soil) ! layer thickness (mm)
   real(r8), INTENT(in) :: zimm(0:nl_soil) ! interface level below a "z" level (mm)

   real(r8), INTENT(in) :: t_soisno (1:nl_soil) ! soil temperature (Kelvin)
   real(r8), INTENT(in) :: vol_liq  (1:nl_soil) ! liquid volumetric water content
   real(r8), INTENT(in) :: vol_ice  (1:nl_soil) ! ice volumetric water content
   real(r8), INTENT(in) :: icefrac  (1:nl_soil)
   real(r8), INTENT(in) :: eff_porosity(1:nl_soil) ! effective porosity = porosity - vol_ice

   real(r8), INTENT(in) :: porsl  (1:nl_soil) ! volumetric soil water at saturation (porosity)
   real(r8), INTENT(in) :: hksati (1:nl_soil) ! hydraulic conductivity at saturation (mm H2O /s)
   real(r8), INTENT(in) :: bsw    (1:nl_soil) ! Clapp and Hornberger "b"
   real(r8), INTENT(in) :: psi0   (1:nl_soil) ! minimum soil suction (mm) [-]
   real(r8), INTENT(in) :: rootr  (1:nl_soil) ! effective fraction of roots in each soil layer
   real(r8), INTENT(in) :: zwt                ! the depth from ground (soil) surface to water table [m]

   real(r8), intent(out) :: dwat(1:nl_soil)   ! change of soil water [m3/m3]
   real(r8), INTENT(out) :: qcharge           ! aquifer recharge rate (positive to aquifer) (mm/s)
   real(r8), INTENT(out) :: qin (1:nl_soil)   ! waterflux into     the soil layer (mm H2O/s)
   real(r8), INTENT(out) :: qout (1:nl_soil)  ! waterflux out  off the soil layer (mm H2O/s)
!
! local arguments
!
   integer  :: j                 ! do loop indices
   real(r8) :: amx(1:nl_soil)    ! "a" left off diagonal of tridiagonal matrix
   real(r8) :: bmx(1:nl_soil)    ! "b" diagonal column for tridiagonal matrix
   real(r8) :: cmx(1:nl_soil)    ! "c" right off diagonal tridiagonal matrix
   real(r8) :: rmx(1:nl_soil)    ! "r" forcing term of tridiagonal matrix
   real(r8) :: den(1:nl_soil)    ! used in calculating qin, qout
   real(r8) :: alpha(1:nl_soil)  ! used in calculating qin, qout
   real(r8) :: dqidw0(1:nl_soil) ! d(qin)/d(vol_liq(j-1))
   real(r8) :: dqidw1(1:nl_soil) ! d(qin)/d(vol_liq(j))
   real(r8) :: dqodw1(1:nl_soil) ! d(qout)/d(vol_liq(j))
```



```fortran
subroutine snowwater (lb, dtime, ssi, wimp, &
                      pg_rain, qseva, qsdew, qsubl, qfros, &
                      dz, wice, wliq, qout_snowb, &
                      do_capsnow, qflx_snwcp_liq, qflx_snwcp_ice)

!-----------------------------------------------------------------------
! Water flow wihtin snow is computed by an explicit and non-physical based scheme,
! which permits a part of liquid water over the holding capacity (a tentative value
! is used, i.e., equal to 0.033*porosity) to percolate into the underlying layer,
! except the case of that the porosity of one of the two neighboring layers is
! less than 0.05, the zero flow is assumed. The water flow out of the bottom
! snow pack will participate as the input of the soil water and runoff.
!-----------------------------------------------------------------------

 use precision
 use phycon_module, only : denice, denh2o  ! physical constant
 implicit none

!----------------------- dummy argument --------------------------------
 integer, INTENT(in) :: &
       lb            ! lower bound of array

 real(r8), INTENT(in) :: &
       dtime,      &! time step (s)
       ssi,        &! irreducible water saturation of snow
       wimp,       &! water impremeable if porosity less than wimp
       dz(lb:0),   &! layer thickness (m)

       pg_rain,    &! rainfall after removal of interception (mm h2o/s)
       qseva,      &! ground surface evaporation rate (mm h2o/s)
       qsdew,      &! ground surface dew formation (mm h2o /s) [+]
       qsubl,      &! sublimation rate from snow pack (mm h2o /s) [+]
       qfros        ! surface dew added to snow pack (mm h2o /s) [+]

 real(r8), INTENT(inout) :: &
       wice(lb:0),&! ice lens (kg/m2)
       wliq(lb:0)  ! liquid water (kg/m2)

 real(r8), INTENT(out) :: &
       qout_snowb  ! rate of water out of snow bottom (mm/s)

 logical,  INTENT(in)   ::&
       do_capsnow
 real(r8), INTENT(inout)::&
       qflx_snwcp_liq
 real(r8), INTENT(inout)::&
       qflx_snwcp_ice

!----------------------- local variables --------------------------------
 integer j           ! k do loop/array indices
```

2. 模式原理

通用陆面过程模式CoLM：地表水与地下水运动模块

- 地表径流的参数化方案考虑了地形、地下水位、降水和入渗速度等因素；
- 一个模式网格内，饱和区域的地表水全部转化为径流流走（蓄满产流），非饱和区域的地表水，部分入渗到土壤中，剩余部分转化为径流（超渗产流），总的地表径流为：

$$\mathrm{r}_{\mathrm{surface}} = f_{sat} * G_{water} + (1 - f_{sat}) * (G_{water} - q_{inmax})$$

- 地下水的运动包含土壤水对地下水的补给和由地形起伏引起的地下径流。

2. 模式原理

通用陆面过程模式CoLM：地表水与地下水运动模块

```fortran
subroutine surfacerunoff (nl_soil,wtfact,wimp,bsw,porsl,phi0,hksati,&
                          z,dz,zi,eff_porosity,icefrac,zwt,gwat,rsur)
!=======================================================================
! the original code was provide by Robert E. Dickinson based on following clues:
! a water table level determination level added including highland and
! lowland levels and fractional area of wetland (water table above the surface.
! Runoff is parametrized from the lowlands in terms of precip incident on
! wet areas and a base flow, where these are estimated using ideas from TOPMODEL.
!=======================================================================
 use precision
 implicit none
!-----------------------Arguments---------------------------------------

 integer, INTENT(in) :: nl_soil   ! number of soil layers
 real(r8), INTENT(in) :: &
       wtfact,        &! fraction of model area with high water table
       wimp,          &! water impremeable if porosity less than wimp
      bsw(1:nl_soil), &! Clapp-Hornberger "B"
    porsl(1:nl_soil), &! saturated volumetric soil water content(porosity)
     phi0(1:nl_soil), &! saturated soil suction (mm)
   hksati(1:nl_soil), &! hydraulic conductivity at saturation (mm h2o/s)
        z(1:nl_soil), &! layer depth (m)
       dz(1:nl_soil), &! layer thickness (m)
       zi(0:nl_soil), &! interface level below a "z" level (m)
eff_porosity(1:nl_soil), &! effective porosity = porosity - vol_ice
  icefrac(1:nl_soil), &! ice fraction
       gwat         , &! net water input from top
       zwt              ! the depth from ground (soil) surface to water table [m]

 real(r8), INTENT(out) :: rsur    ! surface runoff (mm h2o/s)

!-----------------------Local Variables---------------------------------

 real(r8) qinmax        ! maximum infiltration capability
 real(r8) fsat          ! fractional area with water table at surface
 real(r8), parameter :: fff = 0.5   ! runoff decay factor (m-1)

!-----------------------End Variable List-------------------------------

!  fraction of saturated area
     fsat = wtfact*min(1.0,exp(-0.5*fff*zwt))

! Maximum infiltration capacity
     qinmax = minval(10.**(-6.0*icefrac(1:3))*hksati(1:3))
     if(eff_porosity(1)<wimp) qinmax = 0.

! Surface runoff
     rsur = fsat*max(0.0,gwat) + (1.-fsat)*max(0.,gwat-qinmax)

end subroutine surfacerunoff
```

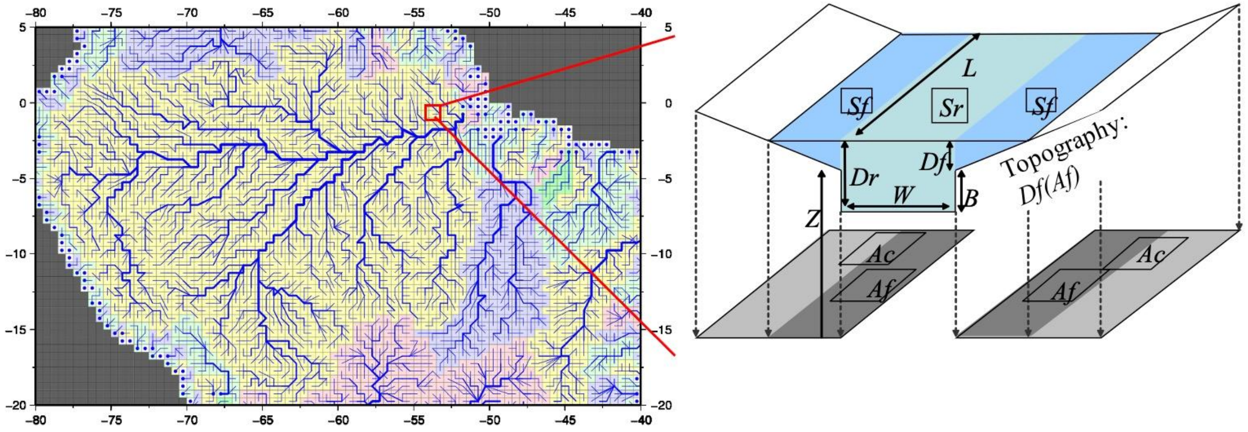
```fortran
    subroutine subsurfacerunoff(nl_soil,deltim,pondmx,&
                              eff_porosity,icefrac,&
                              dz_soisno,zi_soisno,wice_soisno,wliq_soisno,&
                              porsl,psi0,bsw,zwt,wa,&
                              qcharge,rsubst)

! --------------------------------------------------------------------------

    use precision
!
! ARGUMENTS:
    IMPLICIT NONE

    integer, INTENT(in) :: nl_soil        !
    real(r8), INTENT(in) :: deltim        ! land model time step (sec)
    real(r8), INTENT(in) :: pondmx        !

    real(r8), INTENT(in) :: eff_porosity(1:nl_soil) ! effective porosity = porosity - vol_ice
    real(r8), INTENT(in) :: icefrac(1:nl_soil)      ! ice fraction (-)

    real(r8), INTENT(in) :: dz_soisno  (1:nl_soil)  ! layer depth (m)
    real(r8), INTENT(in) :: zi_soisno  (0:nl_soil)  ! interface level below a "z" level (m)
    real(r8), INTENT(inout) :: wice_soisno(1:nl_soil)  ! ice lens (kg/m2)
    real(r8), INTENT(inout) :: wliq_soisno(1:nl_soil)  ! liquid water (kg/m2)

    real(r8), INTENT(in) :: porsl(1:nl_soil)         ! volumetric soil water at saturation (porosity)
    real(r8), INTENT(in) :: psi0(1:nl_soil)          ! minimum soil suction (mm) [-]
    real(r8), INTENT(in) :: bsw(1:nl_soil)           ! Clapp and Hornberger "b"

    real(r8), INTENT(inout) :: zwt        ! the depth from ground (soil) surface to water table [m]
    real(r8), INTENT(inout) :: wa         ! water in the unconfined aquifer (mm)
    real(r8), INTENT(in)    :: qcharge    ! aquifer recharge rate (positive to aquifer) (mm/s)
    real(r8), INTENT(out)   :: rsubst     ! drainage drainage (positive = out of soil column) (mm H2O /s)

!
! LOCAL ARGUMENTS
!

    integer  :: j                     ! indices
    integer  :: jwt                   ! index of the soil layer right above the water table (-)
    real(r8) :: xs                    ! water needed to bring soil moisture to watmin (mm)
    real(r8) :: dzmm(1:nl_soil)   ! layer thickness (mm)
    real(r8) :: xsi                   ! excess soil water above saturation at layer i (mm)
    real(r8) :: xsia                  ! available pore space at layer i (mm)
    real(r8) :: xs1                   ! excess soil water above saturation at layer 1 (mm)
    real(r8) :: ws                    ! summation of pore space of layers below water table (mm)
    real(r8) :: s_node                ! soil wetness (-)
    real(r8) :: available_wliq_soisno      ! available soil liquid water in a layer
    real(r8) :: qcharge_tot           !
```

2. 模式原理

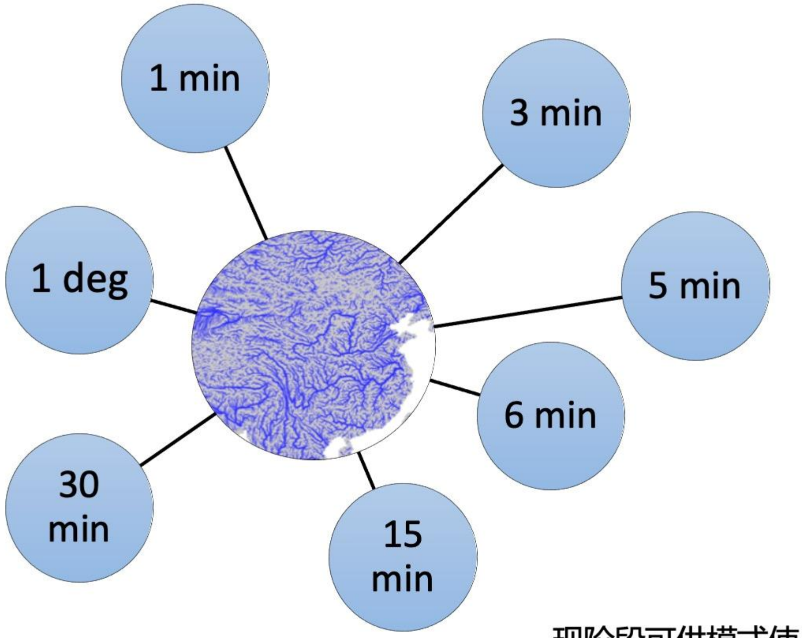
通用陆面过程模式CoLM：河道径流模块CaMa_Flood

CaMa-Flood (**Ca**tchment-based **Ma**cro-scale **Flood**plain model)

- Distributed river routing model using River Network Map



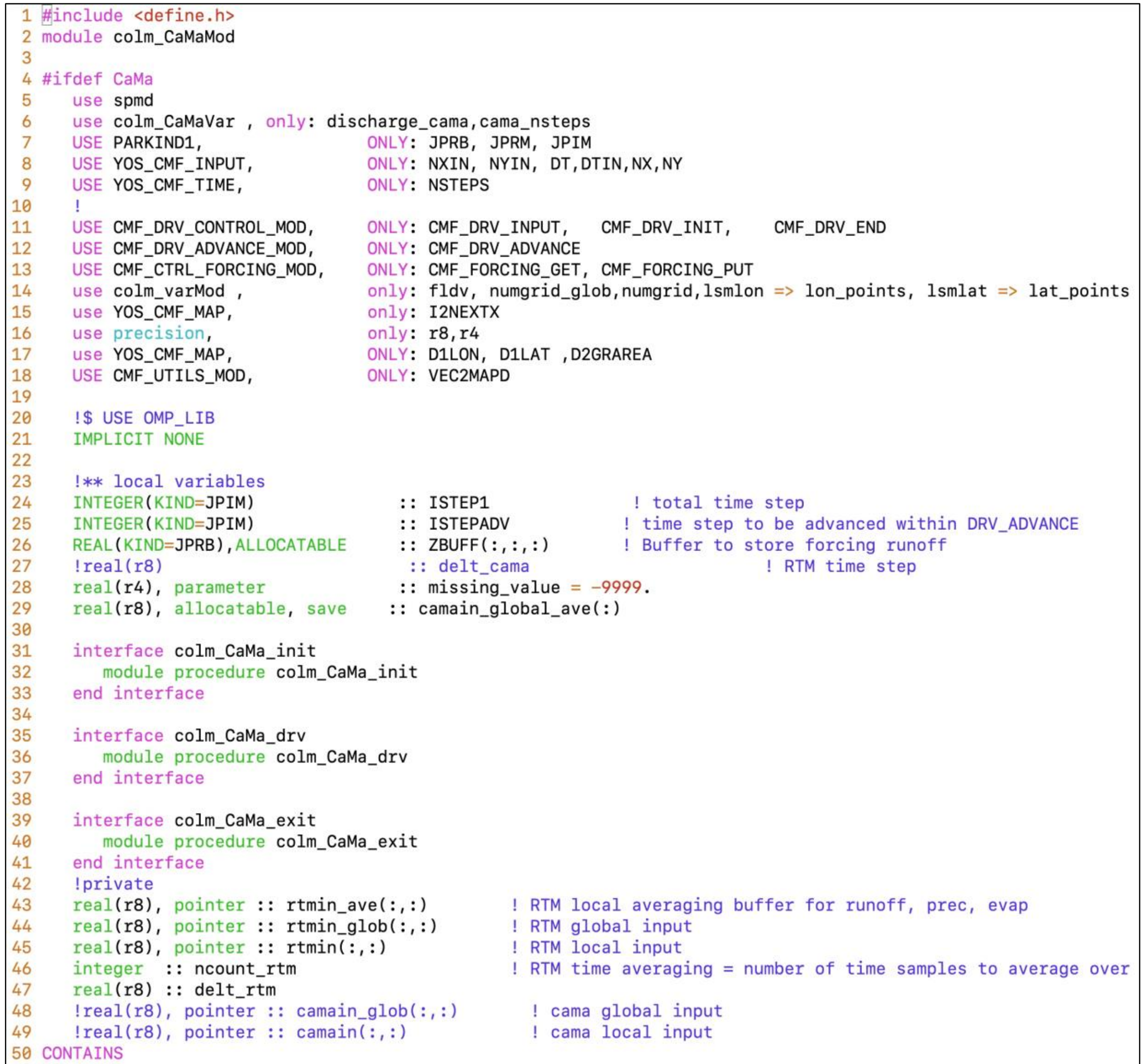
Yamazaki, Kim et al, 2011，2020



现阶段可供模式使用的分辨率共用7种

2. 模式原理

通用陆面过程模式CoLM：河道径流模块CaMa_Flood



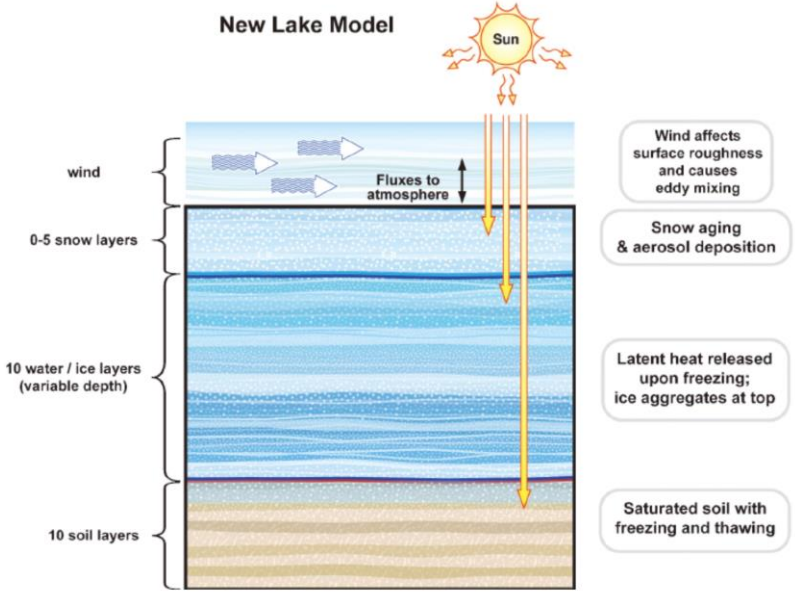
```
#include <define.h>
module colm_CaMaMod

#ifdef CaMa
   use spmd
   use colm_CaMaVar , only: discharge_cama,cama_nsteps
   USE PARKIND1,                ONLY: JPRB, JPRM, JPIM
   USE YOS_CMF_INPUT,           ONLY: NXIN, NYIN, DT,DTIN,NX,NY
   USE YOS_CMF_TIME,            ONLY: NSTEPS
   !
   USE CMF_DRV_CONTROL_MOD,     ONLY: CMF_DRV_INPUT,   CMF_DRV_INIT,    CMF_DRV_END
   USE CMF_DRV_ADVANCE_MOD,     ONLY: CMF_DRV_ADVANCE
   USE CMF_CTRL_FORCING_MOD,    ONLY: CMF_FORCING_GET, CMF_FORCING_PUT
   use colm_varMod ,            only: fldv, numgrid_glob,numgrid,lsmlon => lon_points, lsmlat => lat_points
   use YOS_CMF_MAP,             only: I2NEXTX
   use precision,               only: r8,r4
   use YOS_CMF_MAP,             ONLY: D1LON, D1LAT ,D2GRAREA
   USE CMF_UTILS_MOD,           ONLY: VEC2MAPD

   !$ USE OMP_LIB
   IMPLICIT NONE

   !** local variables
   INTEGER(KIND=JPIM)              :: ISTEP1              ! total time step
   INTEGER(KIND=JPIM)              :: ISTEPADV           ! time step to be advanced within DRV_ADVANCE
   REAL(KIND=JPRB),ALLOCATABLE     :: ZBUFF(:,:,:)       ! Buffer to store forcing runoff
   !real(r8)                        :: delt_cama                       ! RTM time step
   real(r4), parameter             :: missing_value = -9999.
   real(r8), allocatable, save    :: camain_global_ave(:)

   interface colm_CaMa_init
      module procedure colm_CaMa_init
   end interface

   interface colm_CaMa_drv
      module procedure colm_CaMa_drv
   end interface

   interface colm_CaMa_exit
      module procedure colm_CaMa_exit
   end interface
   !private
   real(r8), pointer :: rtmin_ave(:,:)        ! RTM local averaging buffer for runoff, prec, evap
   real(r8), pointer :: rtmin_glob(:,:)       ! RTM global input
   real(r8), pointer :: rtmin(:,:)            ! RTM local input
   integer  :: ncount_rtm                     ! RTM time averaging = number of time samples to average over
   real(r8) :: delt_rtm
   !real(r8), pointer :: camain_glob(:,:)       ! cama global input
   !real(r8), pointer :: camain(:,:)            ! cama local input
CONTAINS
```

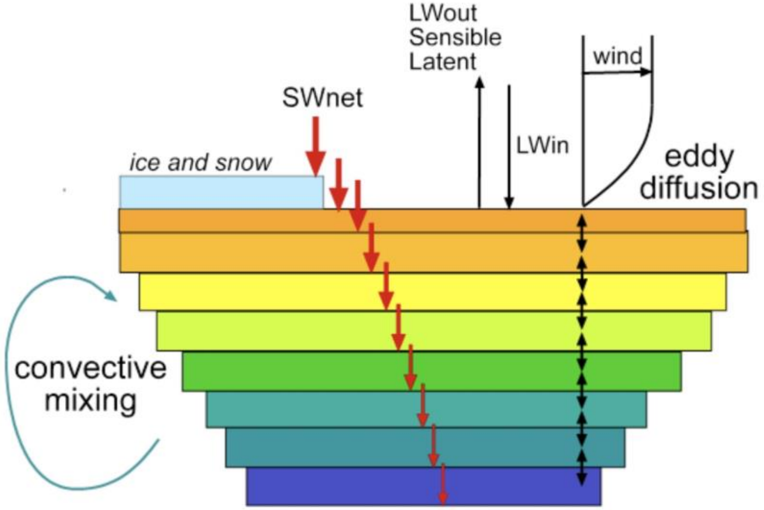
2. 模式原理

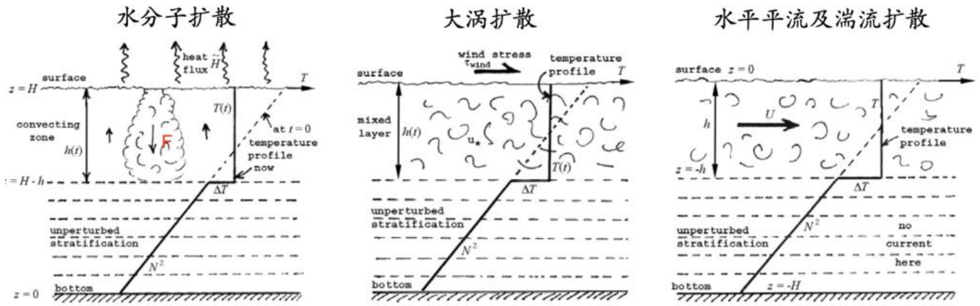
通用陆面过程模式CoLM：湖泊模块

主要特点：

- 降水与湖泊表面的能量交换及积雪过程
- 湖水相变及三种湖水垂直混合过程
- 湖泊的短波辐射吸收特性参数化方案
- 雪层、湖泊层、淤泥层作为一个整体同时求解温度

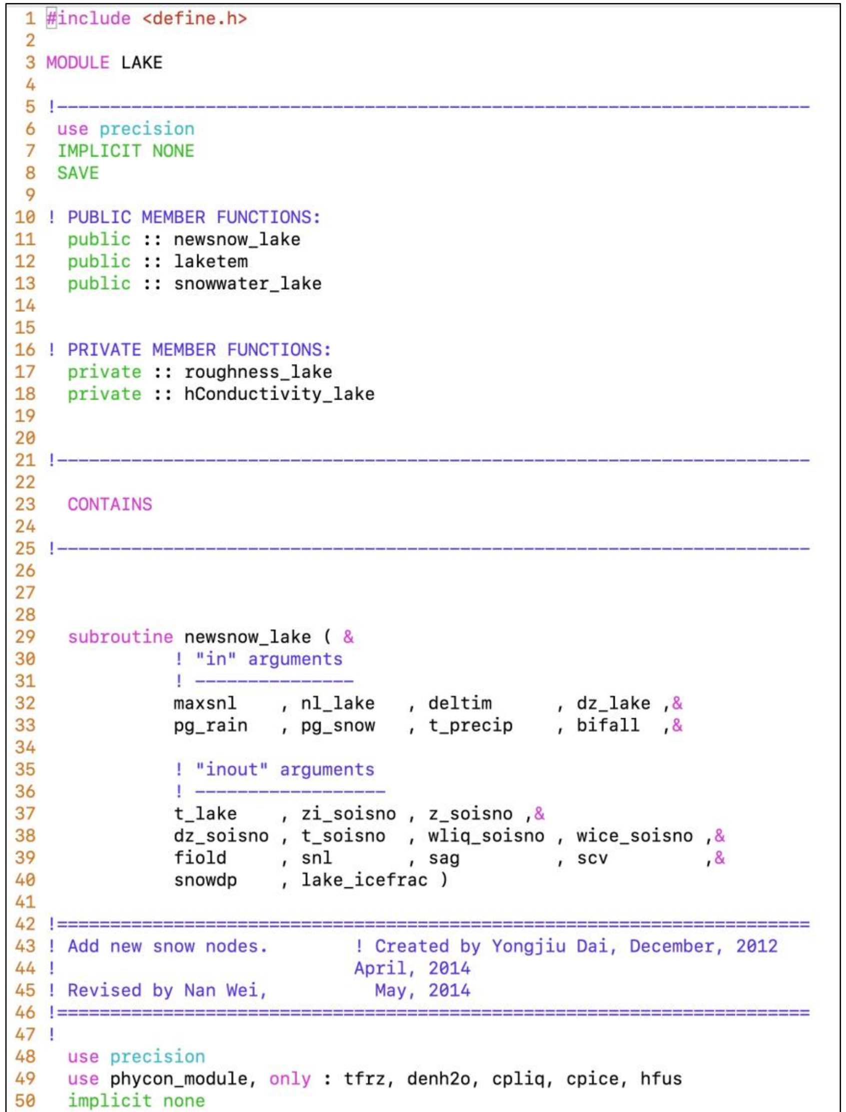




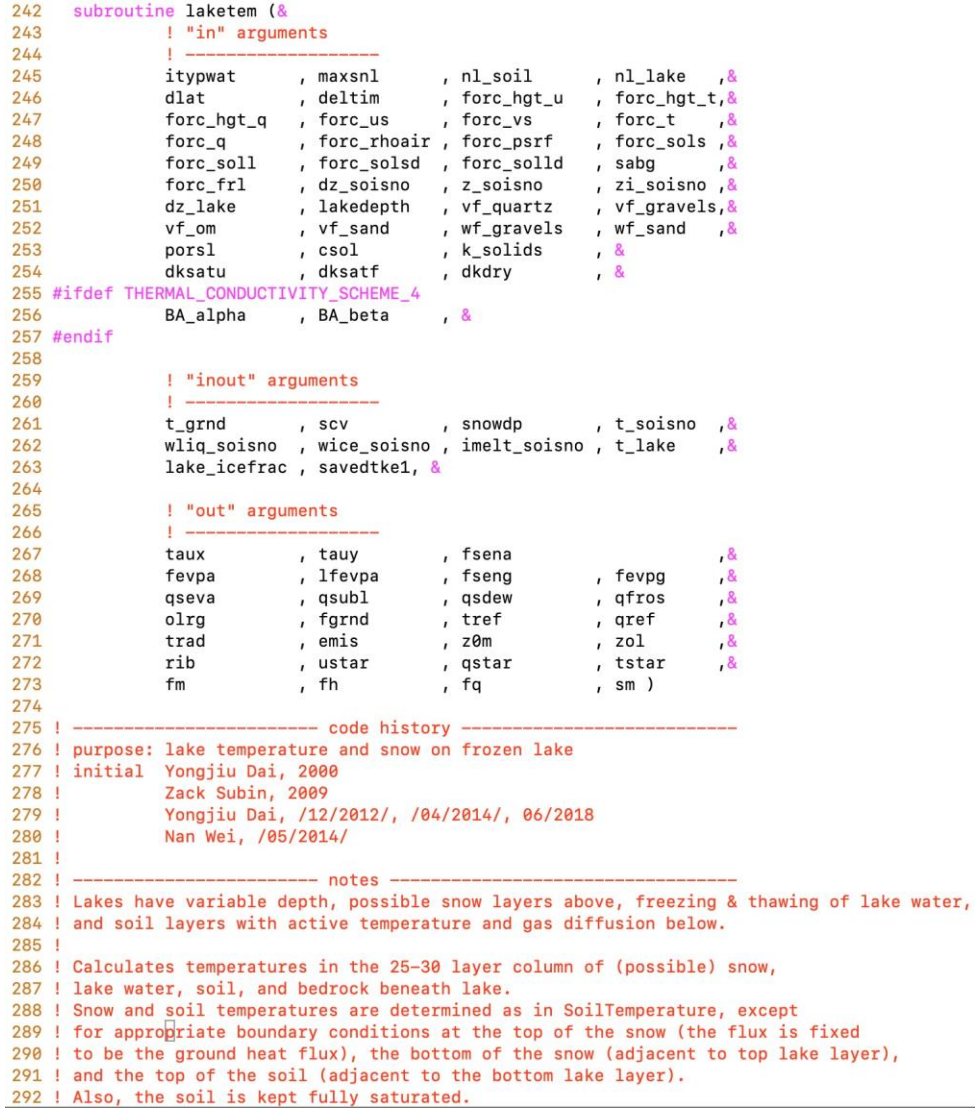


2. 模式原理

通用陆面过程模式CoLM：湖泊模块



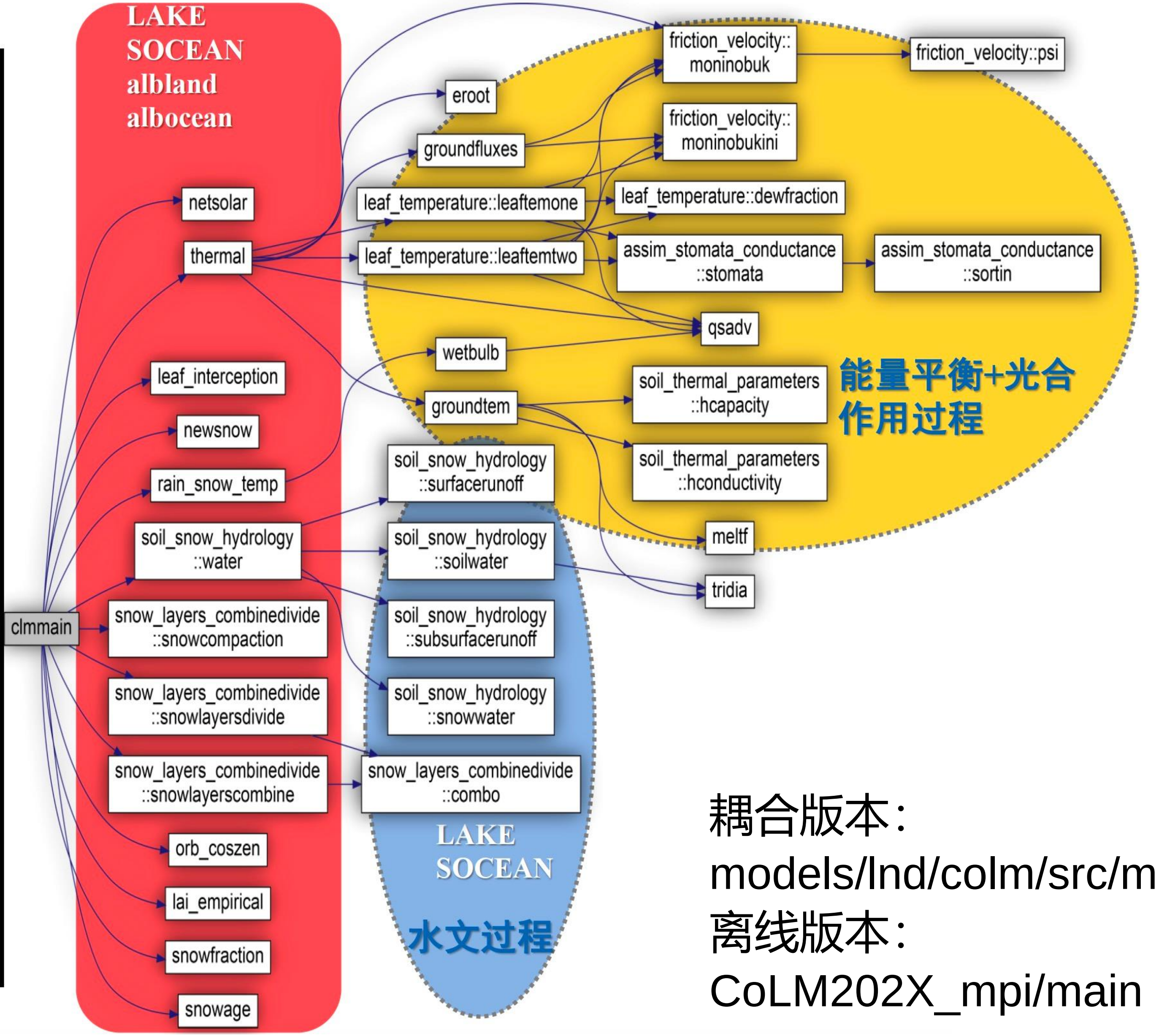
```fortran
#include <define.h>

MODULE LAKE

!-----------------------------------------------------------------------
 use precision
 IMPLICIT NONE
 SAVE

! PUBLIC MEMBER FUNCTIONS:
  public :: newsnow_lake
  public :: laketem
  public :: snowwater_lake


! PRIVATE MEMBER FUNCTIONS:
  private :: roughness_lake
  private :: hConductivity_lake


!-----------------------------------------------------------------------

  CONTAINS

!-----------------------------------------------------------------------



  subroutine newsnow_lake ( &
             ! "in" arguments
             ! ---------------
             maxsnl     , nl_lake    , deltim       , dz_lake ,&
             pg_rain    , pg_snow    , t_precip     , bifall  ,&

             ! "inout" arguments
             ! -----------------
             t_lake     , zi_soisno , z_soisno ,&
             dz_soisno , t_soisno   , wliq_soisno , wice_soisno ,&
             fiold      , snl        , sag          , scv         ,&
             snowdp     , lake_icefrac )

!=======================================================================
! Add new snow nodes.         ! Created by Yongjiu Dai, December, 2012
!                               April, 2014
! Revised by Nan Wei,            May, 2014
!=======================================================================
!
  use precision
  use phycon_module, only : tfrz, denh2o, cpliq, cpice, hfus
  implicit none
```



```fortran
  subroutine laketem (&
           ! "in" arguments
           ! -------------------
           itypwat      , maxsnl      , nl_soil      , nl_lake   ,&
           dlat         , deltim      , forc_hgt_u   , forc_hgt_t,&
           forc_hgt_q   , forc_us     , forc_vs      , forc_t    ,&
           forc_q       , forc_rhoair , forc_psrf    , forc_sols ,&
           forc_soll    , forc_solsd  , forc_solld   , sabg      ,&
           forc_frl     , dz_soisno   , z_soisno     , zi_soisno ,&
           dz_lake      , lakedepth   , vf_quartz    , vf_gravels,&
           vf_om        , vf_sand     , wf_gravels   , wf_sand   ,&
           porsl        , csol        , k_solids     , &
           dksatu       , dksatf      , dkdry        , &
#ifdef THERMAL_CONDUCTIVITY_SCHEME_4
           BA_alpha     , BA_beta     , &
#endif

           ! "inout" arguments
           ! -------------------
           t_grnd       , scv         , snowdp       , t_soisno  ,&
           wliq_soisno  , wice_soisno , imelt_soisno , t_lake    ,&
           lake_icefrac , savedtke1, &

           ! "out" arguments
           ! -------------------
           taux         , tauy        , fsena                    ,&
           fevpa        , lfevpa      , fseng        , fevpg     ,&
           qseva        , qsubl       , qsdew        , qfros     ,&
           olrg         , fgrnd       , tref         , qref      ,&
           trad         , emis        , z0m          , zol       ,&
           rib          , ustar       , qstar        , tstar     ,&
           fm           , fh          , fq           , sm )

! ------------------------ code history ---------------------------
! purpose: lake temperature and snow on frozen lake
! initial  Yongjiu Dai, 2000
!          Zack Subin, 2009
!          Yongjiu Dai, /12/2012/, /04/2014/, 06/2018
!          Nan Wei, /05/2014/
!
! ------------------------ notes ----------------------------------
! Lakes have variable depth, possible snow layers above, freezing & thawing of lake water,
! and soil layers with active temperature and gas diffusion below.
!
! Calculates temperatures in the 25-30 layer column of (possible) snow,
! lake water, soil, and bedrock beneath lake.
! Snow and soil temperatures are determined as in SoilTemperature, except
! for appropriate boundary conditions at the top of the snow (the flux is fixed
! to be the ground heat flux), the bottom of the snow (adjacent to top lake layer),
! and the top of the soil (adjacent to the bottom lake layer).
! Also, the soil is kept fully saturated.
```

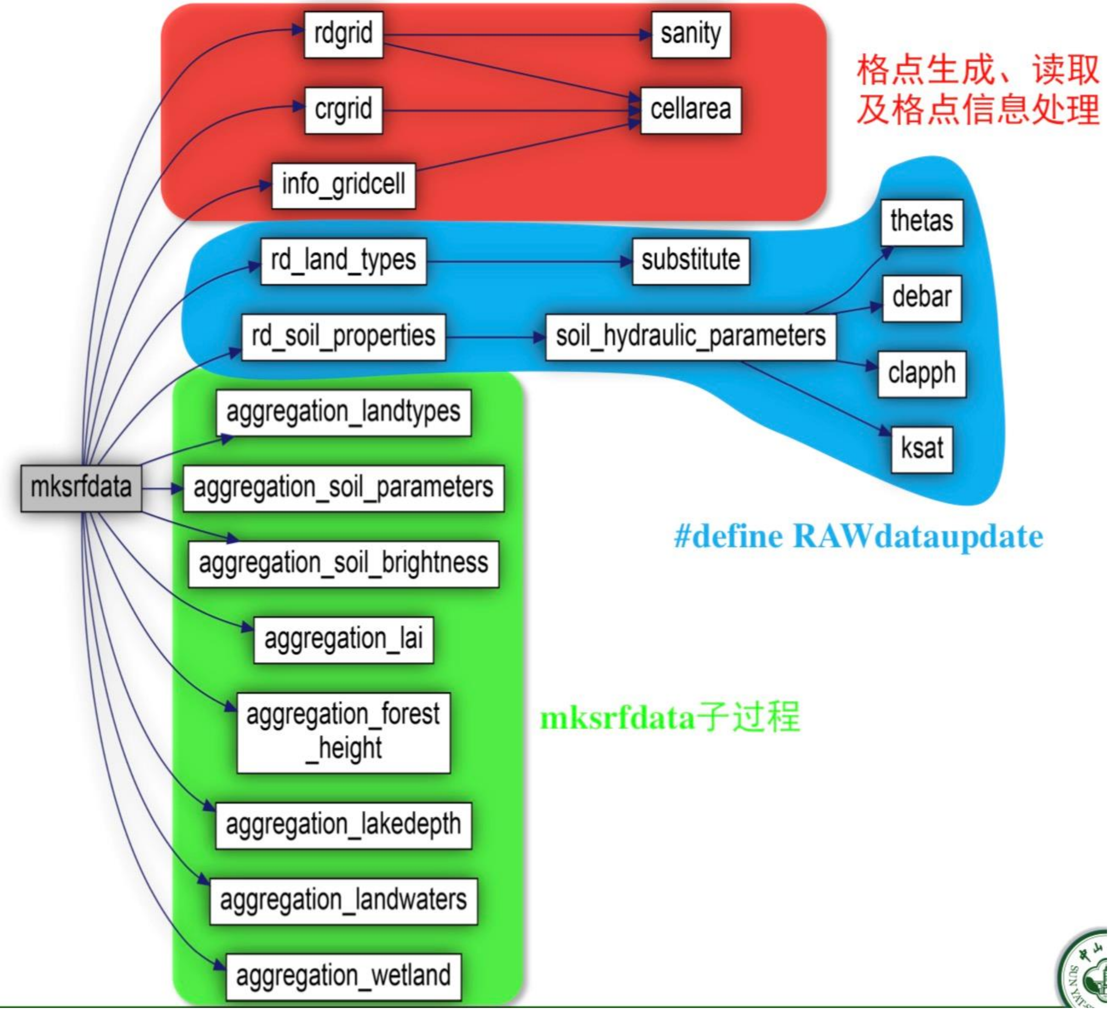
3. 程序代码结构（物理过程主程序）

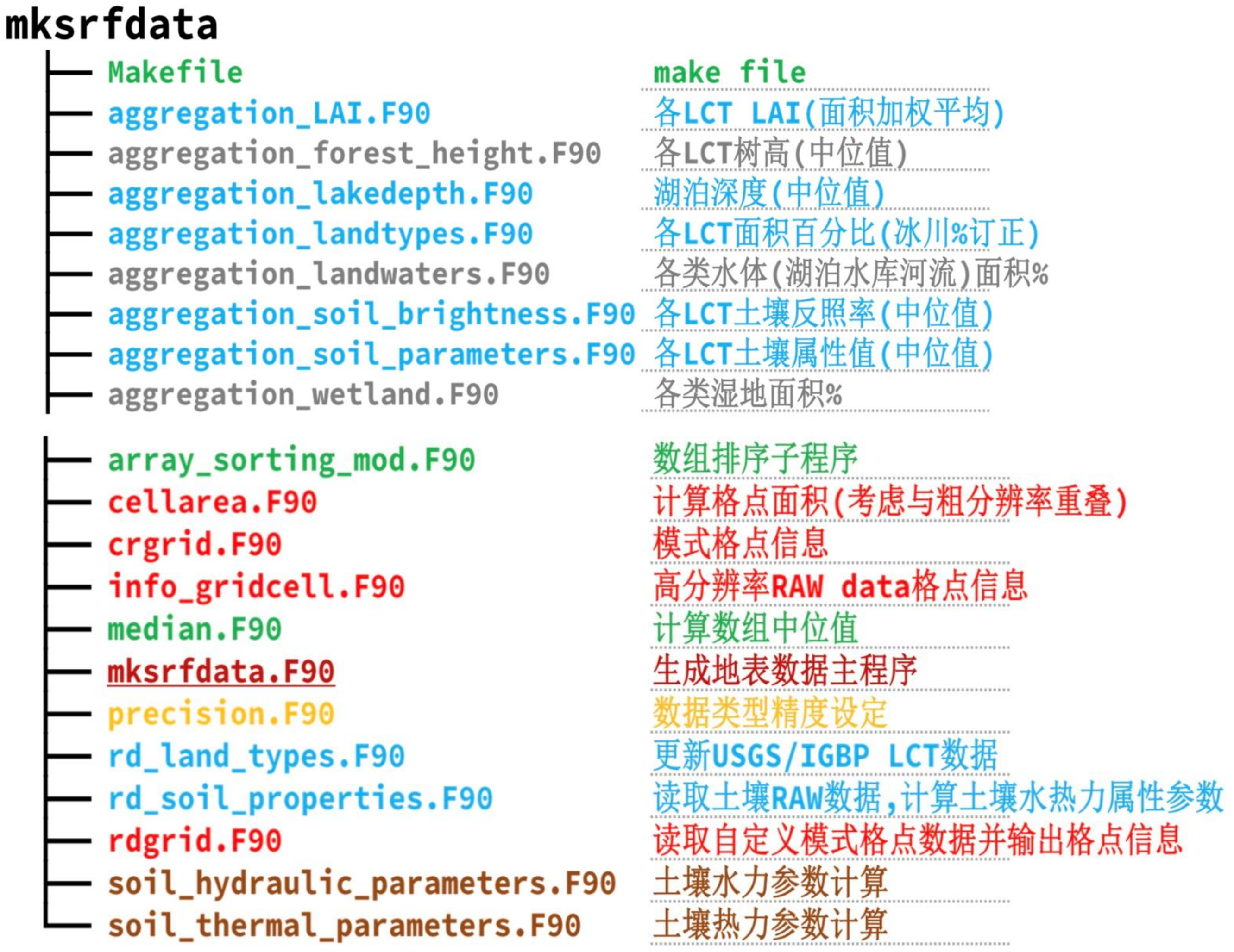
```
netsolar
rain_snow_temp

LEAF_interception
newsnow
THERMAL
WATER
snowcompaction
snowlayerscombine
snowlayersdivide
snowage

newsnow_lake
laketem
snowwater_lake

SOCEAN

orb_coszen
EcoModel (LAI_empirical)
snowfraction
albland
albocean
```



耦合版本：
models/lnd/colm/src/mainc
离线版本：
CoLM202X_mpi/main

3. 程序代码结构（生成与模式分辨率相匹配的地表资料）



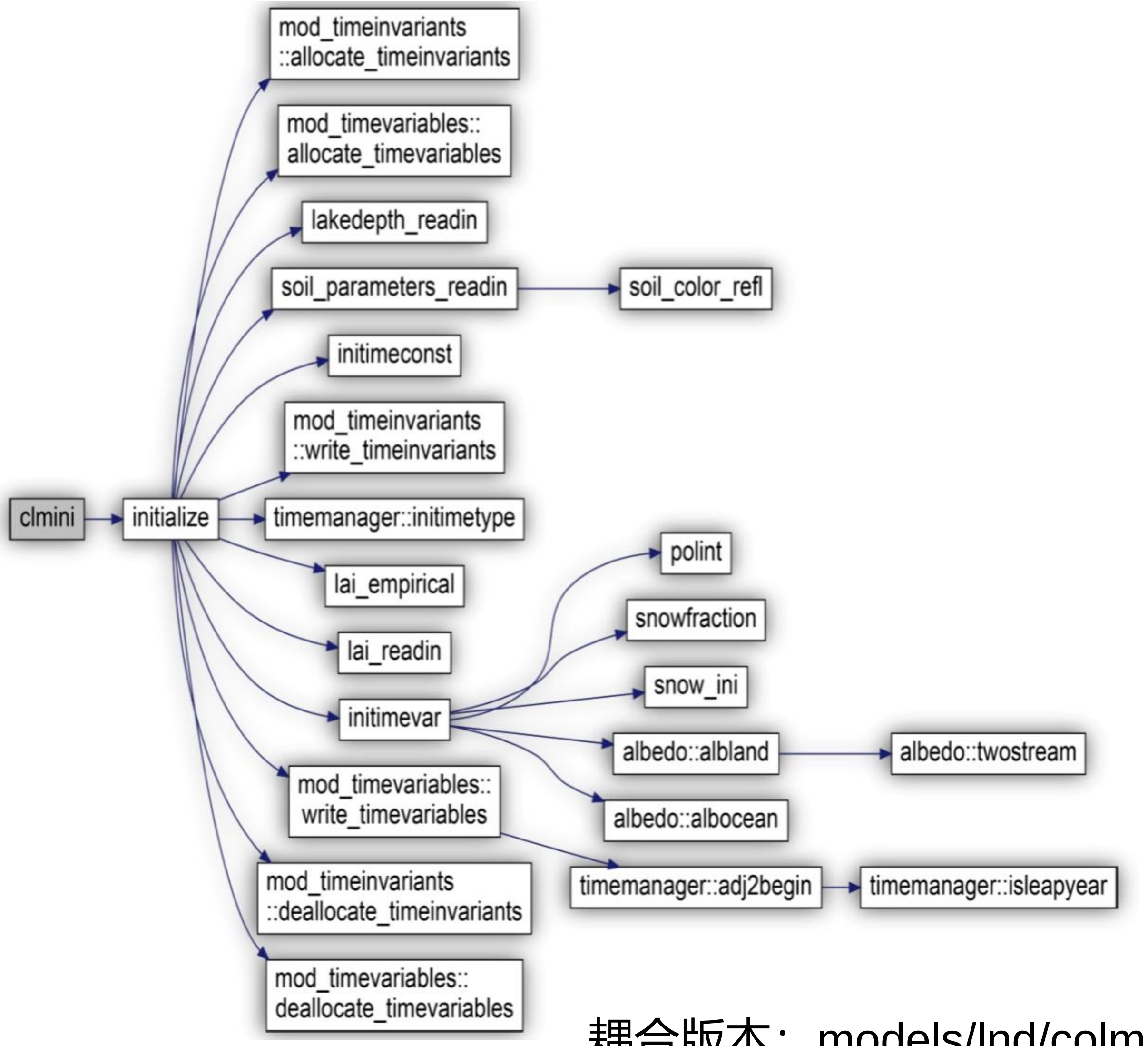


mksrfdata

Makefile	make file
aggregation_LAI.F90	各LCT LAI(面积加权平均)
aggregation_forest_height.F90	各LCT树高(中位值)
aggregation_lakedepth.F90	湖泊深度(中位值)
aggregation_landtypes.F90	各LCT面积百分比(冰川%订正)
aggregation_landwaters.F90	各类水体(湖泊水库河流)面积%
aggregation_soil_brightness.F90	各LCT土壤反照率(中位值)
aggregation_soil_parameters.F90	各LCT土壤属性值(中位值)
aggregation_wetland.F90	各类湿地面积%
array_sorting_mod.F90	数组排序子程序
cellarea.F90	计算格点面积(考虑与粗分辨率重叠)
crgrid.F90	模式格点信息
info_gridcell.F90	高分辨率RAW data格点信息
median.F90	计算数组中位值
mksrfdata.F90	生成地表数据主程序
precision.F90	数据类型精度设定
rd_land_types.F90	更新USGS/IGBP LCT数据
rd_soil_properties.F90	读取土壤RAW数据,计算土壤水热力属性参数
rdgrid.F90	读取自定义模式格点数据并输出格点信息
soil_hydraulic_parameters.F90	土壤水力参数计算
soil_thermal_parameters.F90	土壤热力参数计算

离线版本：CoLM202X_mpi/mksrfdata

3. 程序代码结构（陆面模式初始化）



mkinidata

ALBEDO.F90	地表反照率计算
CLMINI.F90	初始化主程序
IniTimeConst.F90	初始化模式常量
IniTimeVar.F90	初始化模式变量
LAI_readin.F90	读取LAI数据
MOD_TimeInvariants.F90	常量数据定义、输出、读取
MOD_TimeVariables.F90	变量数据定义、输出、读取
Makefile	make file
PhysicalConstants.F90	物理常数
initialize.F90	初始化主程序
lai_empirical.F90	LAI/SAI经验计算方法
lakedepth_readin.F90	湖泊深度数据读取及分层
orb_coszen.F90	计算太阳天顶角
precision.F90	同mksrfdata目录文件
snowfraction.F90	计算地表雪面积覆盖比例
soil_color_refl.F90	土壤反照率经验计算
soil_parameters_readin.F90	土壤属性数据读取
timemanager.F90	时间相关操作函数工具

耦合版本：models/lnd/colm/src/mainc

离线版本：CoLM202X_mpi/mkinidata

4. 地表数据

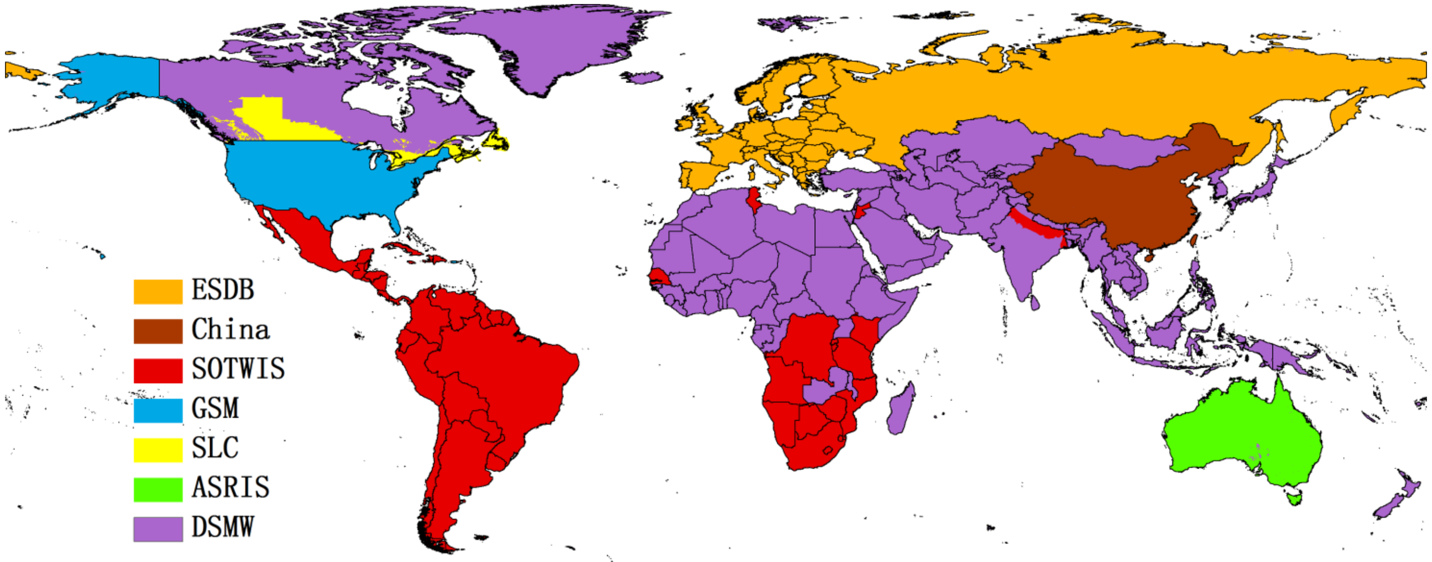
陆面模式基础数据集（全球）— 已提前生成

数据集	现使用	分辨率(米)	生成/引入
土壤属性	中山大学	1000	生成
土地覆盖类型	USGS MODIS	1000	引入
陆冰	GLIMS3.2	1000	引入
湖泊/湿地	FLAKE WWF	1000	引入
叶面积指数	中山大学	500	生成
地形	SRTM	90	引入
河川河道	SRTM	90	引入

4. 地表数据

土壤属性数据集(GSDE)

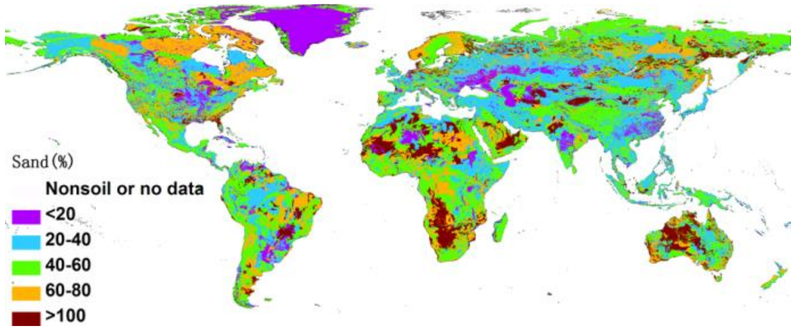
- ✓ 融合了可获得的多源土壤数据，形成了最为详尽的世界土壤数据集，共34个土壤属性（土壤颜色、质地、有机质、容重、砾石含量、土壤pH值，碳、氮、磷、硫、钾、可交换阳离子含量等）供地球系统模式直接使用。
- ✓ 空间上采用土壤连接法，依据土壤类别、质地类别、土壤剖面个数与土壤图斑之间距离等信息将土壤图和土壤剖面连接起来



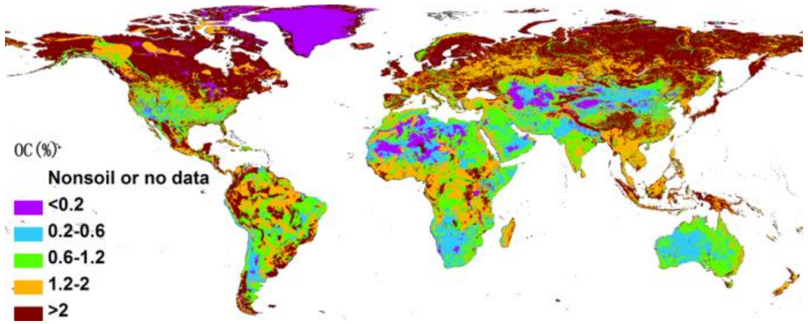
4. 地表数据

土壤属性数据集(GSDE)

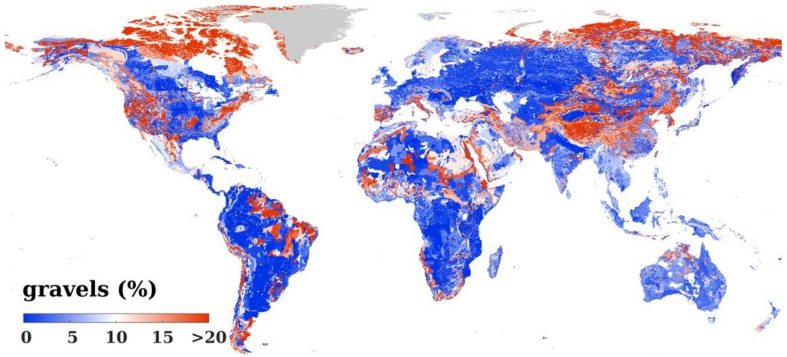
矿物质含量



有机质含量



砾石含量



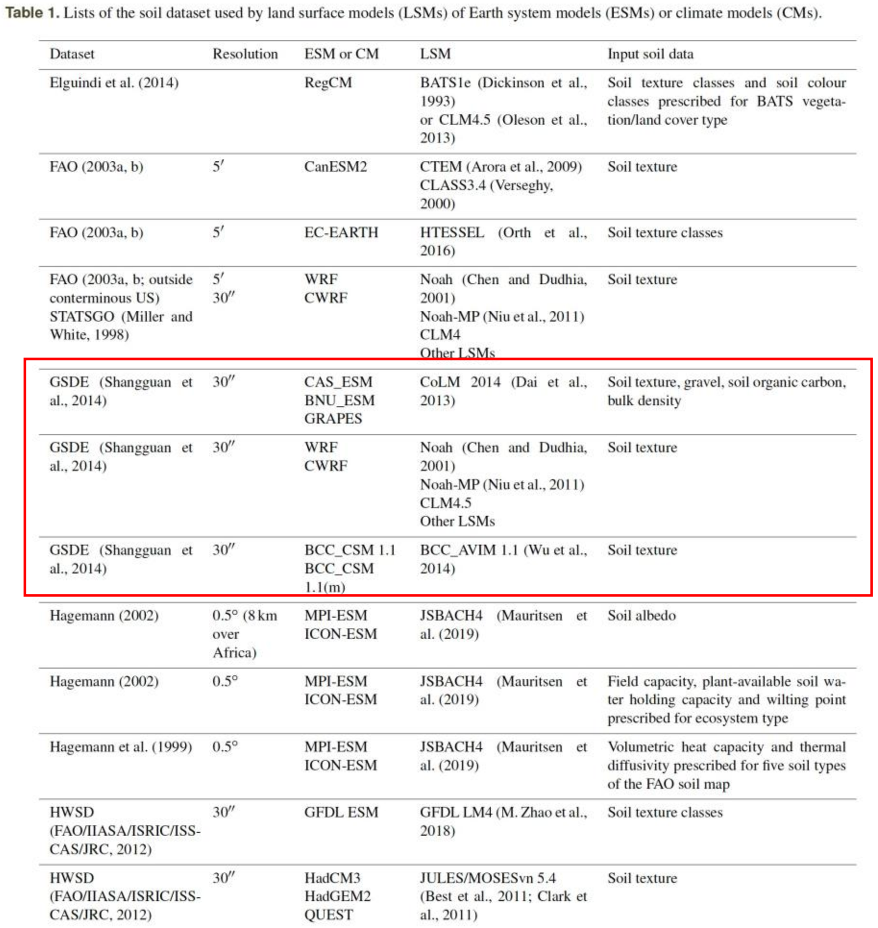


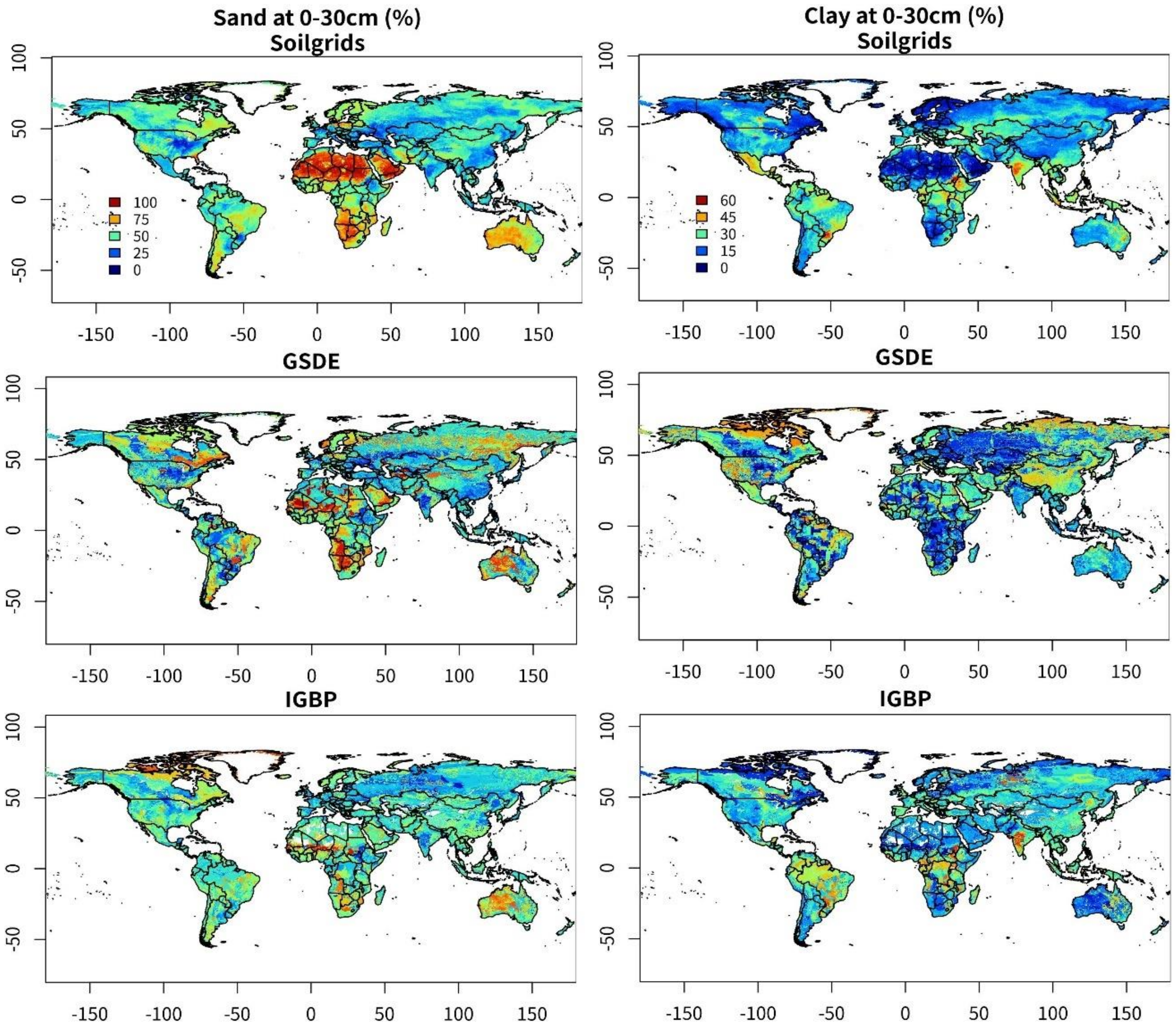
Table 1. Lists of the soil dataset used by land surface models (LSMs) of Earth system models (ESMs) or climate models (CMs).

Dataset	Resolution	ESM or CM	LSM	Input soil data
Elguindi et al. (2014)		RegCM	BATS1e (Dickinson et al., 1993) or CLM4.5 (Oleson et al., 2013)	Soil texture classes and soil colour classes prescribed for BATS vegetation/land cover type
FAO (2003a, b)	5′	CanESM2	CTEM (Arora et al., 2009) CLASS3.4 (Verseghy, 2000)	Soil texture
FAO (2003a, b)	5′	EC-EARTH	HTESSEL (Orth et al., 2016)	Soil texture classes
FAO (2003a, b; outside conterminous US) STATSGO (Miller and White, 1998)	5′ 30″	WRF CWRF	Noah (Chen and Dudhia, 2001) Noah-MP (Niu et al., 2011) CLM4 Other LSMs	Soil texture
GSDE (Shangguan et al., 2014)	30″	CAS_ESM BNU_ESM GRAPES	CoLM 2014 (Dai et al., 2013)	Soil texture, gravel, soil organic carbon, bulk density
GSDE (Shangguan et al., 2014)	30″	WRF CWRF	Noah (Chen and Dudhia, 2001) Noah-MP (Niu et al., 2011) CLM4.5 Other LSMs	Soil texture
GSDE (Shangguan et al., 2014)	30″	BCC_CSM 1.1 BCC_CSM 1.1(m)	BCC_AVIM 1.1 (Wu et al., 2014)	Soil texture
Hagemann (2002)	0.5° (8 km over Africa)	MPI-ESM ICON-ESM	JSBACH4 (Mauritsen et al. (2019)	Soil albedo
Hagemann (2002)	0.5°	MPI-ESM ICON-ESM	JSBACH4 (Mauritsen et al. (2019)	Field capacity, plant-available soil water holding capacity and wilting point prescribed for ecosystem type
Hagemann et al. (1999)	0.5°	MPI-ESM ICON-ESM	JSBACH4 (Mauritsen et al. (2019)	Volumetric heat capacity and thermal diffusivity prescribed for five soil types of the FAO soil map
HWSD (FAO/IIASA/ISRIC/ISS-CAS/JRC, 2012)	30″	GFDL ESM	GFDL LM4 (M. Zhao et al., 2018)	Soil texture classes
HWSD (FAO/IIASA/ISRIC/ISS-CAS/JRC, 2012)	30″	HadCM3 HadGEM2 QUEST	JULES/MOSESvn 5.4 (Best et al., 2011; Clark et al., 2011)	Soil texture

Shangguan et al., 2014, JAMES

4. 地表数据

土壤属性数据集(GSDE)



- ❖ 对最新的土壤属性数据进行比较，发现他们之间的差异较大，这必然带来模拟不确定性
- ❖ 通过与近万个站点观测数据进行验证，精度最高的两套数据分别是GSDE和SoilGrids

Dai et al., 2020, Soil

4. 地表数据

全球1 km 土壤水热特征参数集

陆面模式土壤水热传输过程

- ❖ 从大尺度到小尺度模拟**非饱和水分运移**，都需要求解Richards方程, 1D形式如下：

$$\frac{\partial\theta}{\partial t}=\frac{\partial}{\partial z}\left[K(\theta)\left(\frac{\partial h(\theta)}{\partial z}-1\right)\right]-S(\theta)$$

$$\xrightarrow[K(\theta)]{h(\theta)}$$

- ❖ 模拟**热传导过程**，求解如下热传导方程：

$$c\frac{\partial T}{\partial t}=\frac{\partial}{\partial z}\left[\tau\frac{\partial T}{\partial z}\right]+S$$

- ❖ 土壤水分特征曲线方程：

1) Campbell模型 (Campbell 1974):

$$\psi=\psi_s\left(\frac{\theta}{\theta_s}\right)^{-1/\lambda},K(\theta)=K_s\left(\frac{\theta}{\theta_s}\right)^{3+2/\lambda}$$

2) VGM模型 (van Genuchten, 1980; Mualem, 1976):

$$\Theta=\frac{\theta-\theta_r}{\theta_s-\theta_r}=\left[1+(\alpha\psi)^n\right]^{\frac{1}{n}-1}$$

$$K(\Theta)=K_s\Theta^L\left(1-\left(1-\Theta^{1/(1-1/n)}\right)^{1-1/n}\right)^2$$

Parameters to be determined: ψ_s λ θ_r θ_s α n K_s c τ

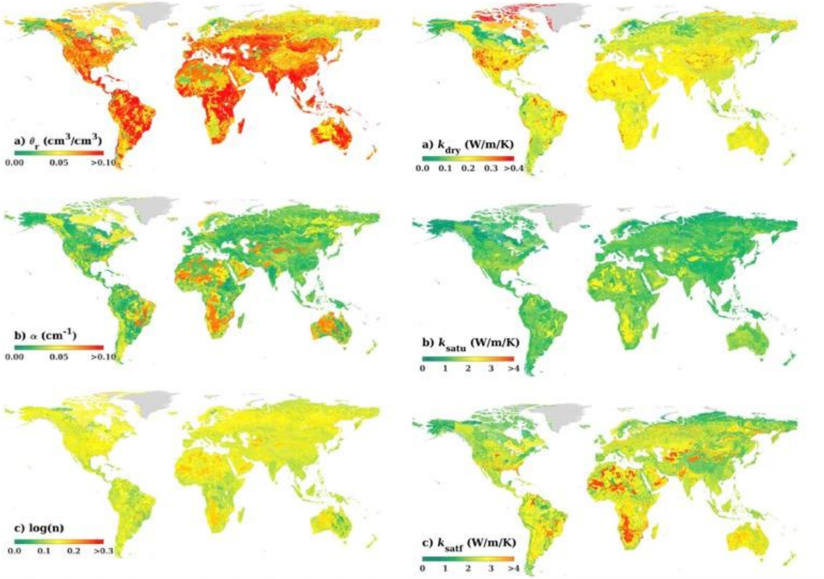
❖ **土壤基本性质**
- ✓ 粒径成分
- ✓ 孔隙度
- ✓ 容重
- ✓ 有机碳含量

改进拟合多种土壤水热参数计算方案

土壤水热参数最优估计

全新的土壤水热参数集合预报方法

Category	Symbol	Description	Units	Variable name
Basic soil properties	ν_{gravel}	volumetric fraction of gravels	cm^3/cm^3	vf_gravels_s_l
	ν_{SOM}	volumetric fraction of organic matter	cm^3/cm^3	vf_om_s_l
	ν_{sand}	volumetric fraction of sand	cm^3/cm^3	vf_sand_s_l
	ν_{silt}	volumetric fraction of silt	cm^3/cm^3	vf_silt_s_l
	ν_{clay}	volumetric fraction of clay	cm^3/cm^3	vf_clay_s_l
	ν_{quartz}	volumetric fraction of quartz within mineral soils	cm^3/cm^3	vf_quartz _s_l
Soil hydraulic properties in both Campbell and VG models	θ_s	saturated water content	cm^3/cm^3	theta_s_l
	K_s	saturated hydraulic conductivity	cm/day	k_s_l
Soil hydraulic properties in the Campbell model	ψ_s	saturated suction	cm	psi_s_l
	λ	pore size distribution index	dimensionless	lambda_l
Soil hydraulic properties in the VG model	θ_r	residual moisture content	cm^3/cm^3	VGM_theta_r_l
	α	a parameter that corresponds approximately to the inverse of the air-entry value	cm^{-1}	VGM_alpha_l
	n	a shape parameter	dimensionless	VGM_n_l
Soil thermal properties	c_s	volumetric heat capacity of soil solids in a unit soil volume	J · m^{-1} · K^{-1}	csol_l
	k_{satu}	thermal conductivity of unfrozen saturated soils	W · m^{-1} · K–^{1}	ksat_u_l
	k_{satf}	thermal conductivity of frozen saturated soils	W · m^{-1} · K^{-1}	ksat_f_l
	k_{dry}	thermal conductivity of dry soils	W · m^{-1} · K^{-1}	kdry_l



参数最全，分辨率最高的全球土壤参数数据集，可直接用于CLM, CoLM, Noah-LSM, JULES等国际主要陆面模式

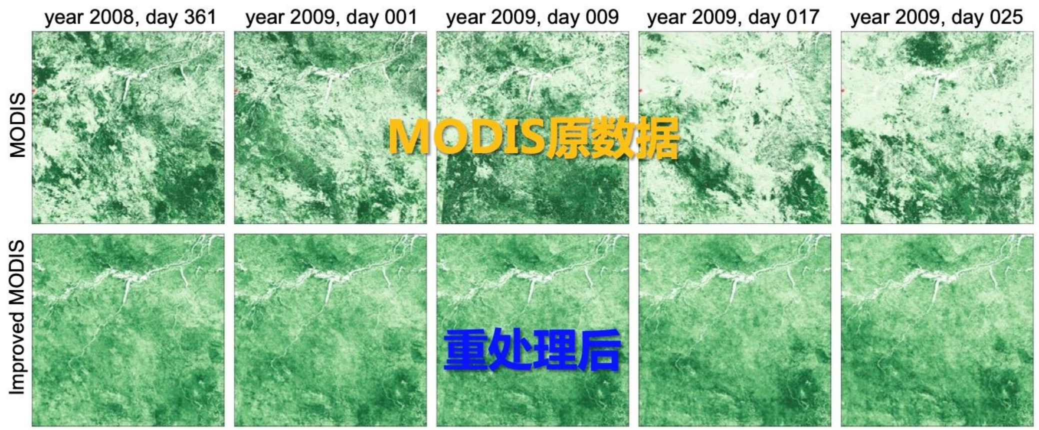
Dai et al., 2020, JAMES

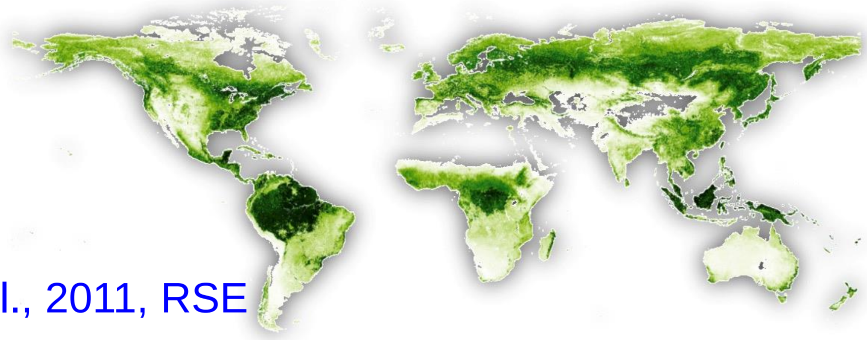
4. 地表数据

长时间序列全球500m叶面积指数数据研制

- 构建了相互协调的滤波算法和订正方法，解决了MODIS原数据的时空不连续和不一致性问题，建立了全球500m LAI（2000-2019年）数据集。

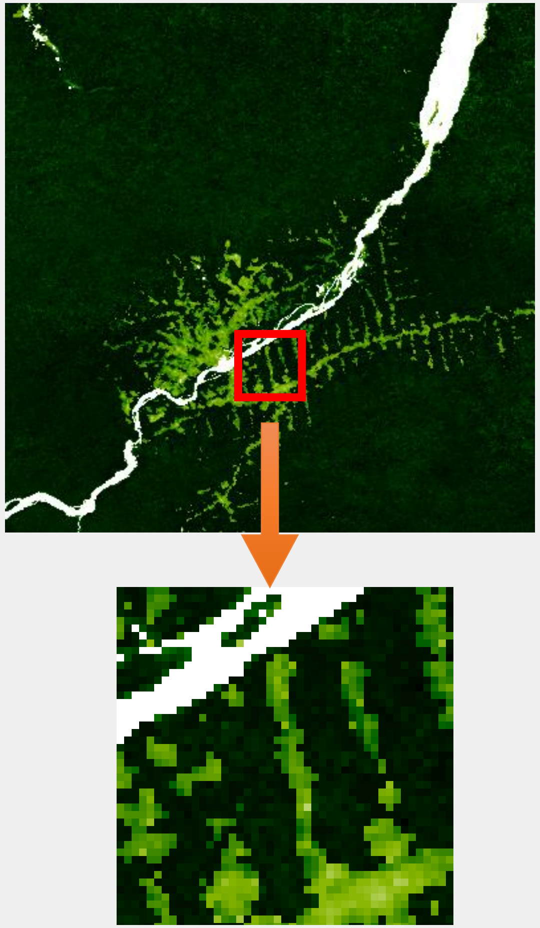
处理结果：时空比较





Yuan et al., 2011, RSE

时空更连续
细节更丰富



4. 地表数据

通过运行离线版本陆面过程模式mksrfdata生成与模式分辨率相匹配的地表资料

```
&nl_colm

   DEF_CASE_NAME = 'global_grid_cruncep_usgs'

   DEF_domain%edges = -90.0
   DEF_domain%edgen = 90.0
   DEF_domain%edgew = -180.0
   DEF_domain%edgee = 180.0

   DEF_nx_blocks = 30
   DEF_ny_blocks = 30
   DEF_PIO_groupsize = 6

   DEF_simulation_time%greenwich     = .TRUE.
   DEF_simulation_time%start_year    = 1970
   DEF_simulation_time%start_month   = 1
   DEF_simulation_time%start_day     = 1
   DEF_simulation_time%start_sec     = 0
   DEF_simulation_time%end_year      = 2016
   DEF_simulation_time%end_month     = 12
   DEF_simulation_time%end_day       = 31
   DEF_simulation_time%end_sec       = 86400
   DEF_simulation_time%spinup_year   = 1980
   DEF_simulation_time%spinup_month  = 1
   DEF_simulation_time%spinup_day    = 1
   DEF_simulation_time%spinup_sec    = 0
   DEF_simulation_time%timestep      = 1800.

   DEF_dir_rawdata  = '/data/sysu_daiyj_01/data_mpi/CLMrawdata_usgs/'
   DEF_dir_output   = '/data/sysu_daiyj_01/cases_mpi'

   ! ----- land units and land sets -----
   ! for GRIDBASED
   DEF_file_landgrid = '/tera02/wein/data_mpi/landdata/landmask_usgs_0.5

   ! for CATCHMENT
   ! DEF_dir_hydrodata = '/tera02/zhangsp/hillslope/output/columbia'
   ! DEF_max_hband = 25
```

```
#!/bin/bash
#SBATCH -J global_usgs_CB
#SBATCH -p normal
#SBATCH -N 64
#SBATCH -n 3840
#SBATCH --ntasks-per-node=60
#SBATCH --mem=220G
#SBATCH -o colm.o%j
#SBATCH -e colm.e%j
#SBATCH --exclusive
#SBATCH -t 7-24:00

module load mathlib/netcdf/intel/4.4.1

export I_MPI_FABRICS=shm:dapl
export I_MPI_DAPL_UD=1
export I_MPI_DAPL_UD_RDMA_MIXED=1
export I_MPI_LARGE_SCALE_THRESHOLD=8192
export I_MPI_DAPL_UD_ACK_SEND_POOL_SIZE=8704
export I_MPI_DAPL_UD_ACK_RECV_POOL_SIZE=8704
export I_MPI_DAPL_UD_RNDV_EP_NUM=2

export DAPL_UCM_REP_TIME=8000 #  REQUEST timer, waiting for REPLY in millisecs
export DAPL_UCM_RTU_TIME=8000 #  REPLY timer, waiting for RTU in millisecs
export DAPL_UCM_RETRY=10 #  REQUEST and REPLY retries
export DAPL_UCM_CQ_SIZE=2000
export DAPL_UCM_QP_SIZE=2000

export DAPL_UCM_DREQ_RETRY=4 #default == 1
export DAPL_UCM_DREP_TIME=200 #default == 200ms
export DAPL_UCM_WAIT_TIME=10000 #default == 60000ms

ulimit -s unlimited
srun --mpi=pmi2 --cpu-bind=cores ../mksrfdata/mksrfdata.x input_global_grid_cruncep_usgs.nml
```

5. 模式安装

耦合模式代码路径：**/data/sysu_daiyj_01/cas_esm，**

支持传统分辨率与新增140公里和25公里分辨率

(1) 新建算例。在模式scripts路径下，运行如下指令：./create_newcase -case xxx(算例名称) -compset I_2000_X -mach generic_linux_intel -res fd025_licom03 -scratchroot xxx(算例编译运行路径) -din_loc_root_csmdata xxx(模式输入数据路径) -max_tasks_per_node xxx （目标机每个节点最大可运行的任务数）；

(2) 进入算例路径，更改相关环境变量设置。如在env_run.xml更改运行时长，在env_mach_pes.xml更改每个分系统进行计算所用的CPU进程数和线程数等。注意：若运行河道径流模式CaMa_Flood，因CaMa_Flood仅支持OPENMP并行，所以可将陆面模式运行时每个节点调用的线程数调高以加速CaMa_Flood运行;

(3) 根据机器具体配置修改机器配置文件Macros.generic_linux_intel（可参考/data/sysu_daiyj_01/cas_esm/scripts/test_amip_coupling/Macros.generic_linux_intel进行配置）；

(4) 运行./configure –case;

5. 模式安装

耦合模式代码路径: **/data/sysu_daiyj_01/cas_esm，**

支持传统分辨率与新增140公里和25公里分辨率

(5) 进入Buildconf路径，设置脚本colm.buildexe.csh和colm.buildnml.csh。其中colm.buildexe.csh主要通过宏定义控制不同模块的开关，如运行CaMa_Flood则需在宏定义中修改为#define CaMa；colm.buildnml.csh主要设置陆面模式运行需要的namelist，如通过fsrf、flai和frivinp_cama指定地表资料数据、LAI数据和CaMa_Flood运行所用的河网数据，fini和fsbc为模式的初始状态数据，第一次运行时需置空；

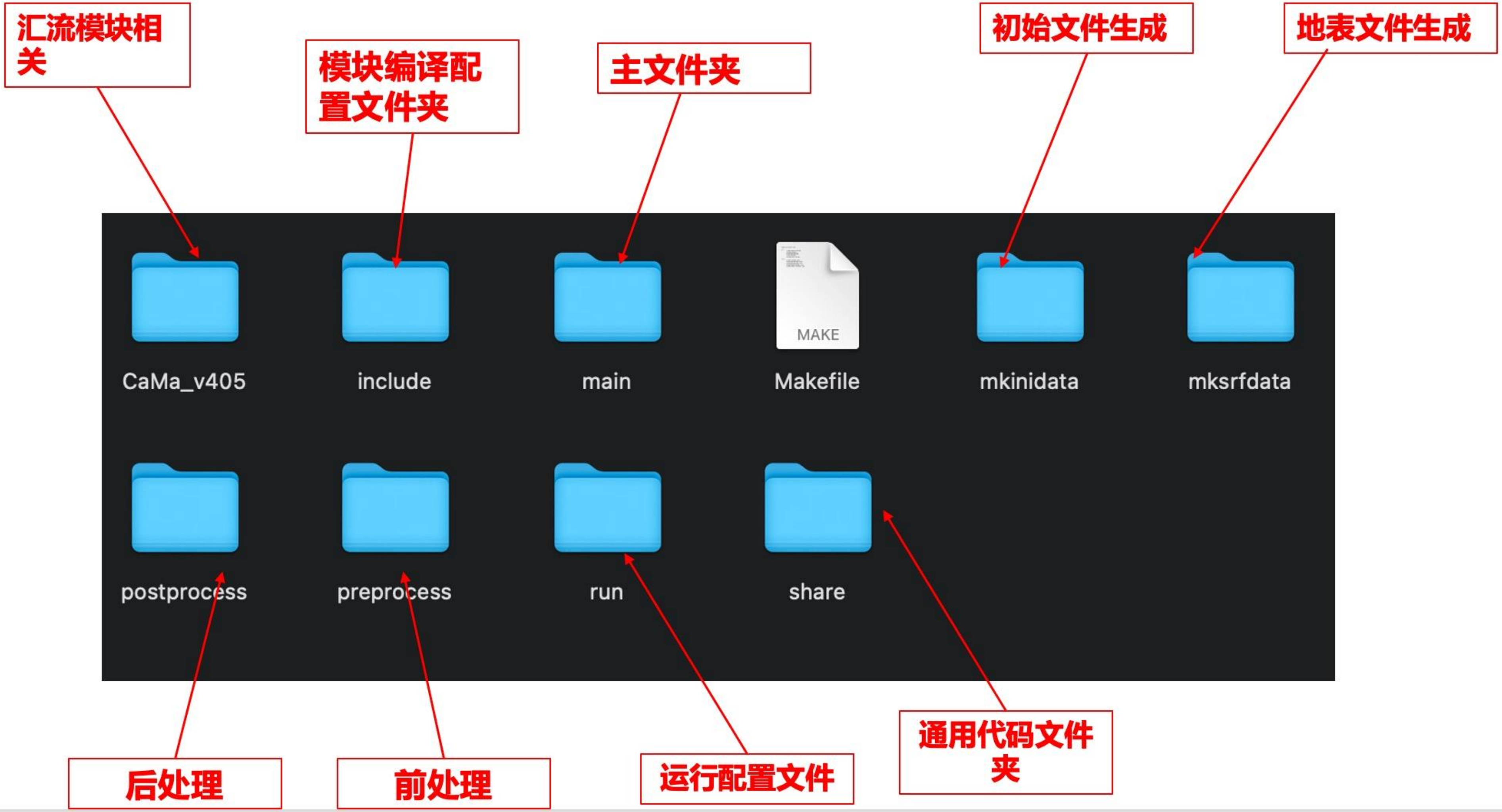
注： CaMa_Flood运行分辨率为0.25度，其河网数据放置于
/data/sysu_daiyj_01/cas_esm/models/lnd/colm/data
glb_15min-cas-esm_128x256和glb_15min-cas-esm_0.23x0.31

(6) 编译算例：运行*.build文件；

(7) 根据目标机提交作业方式配置作业提交脚本，并通过sbatch提交作业。作业提交脚本可参考地球模拟器/public/software路径中给出的样例脚本。

5. 模式安装

离线（全球1km）模式代码路径: **/data/sysu_daiyj_01/CoLM202X_mpi**



汇流模块相关

模块编译配置文件夹

主文件夹

初始文件生成

地表文件生成

后处理

前处理

运行配置文件

通用代码文件夹

5. 模式安装

离线（全球1km）模式代码路径: **/data/sysu_daiyj_01/CoLM202X_mpi**

修改软件配置（include 文件夹）

编译程序

修改namelist

制作地表文件（运行mksrfdata）

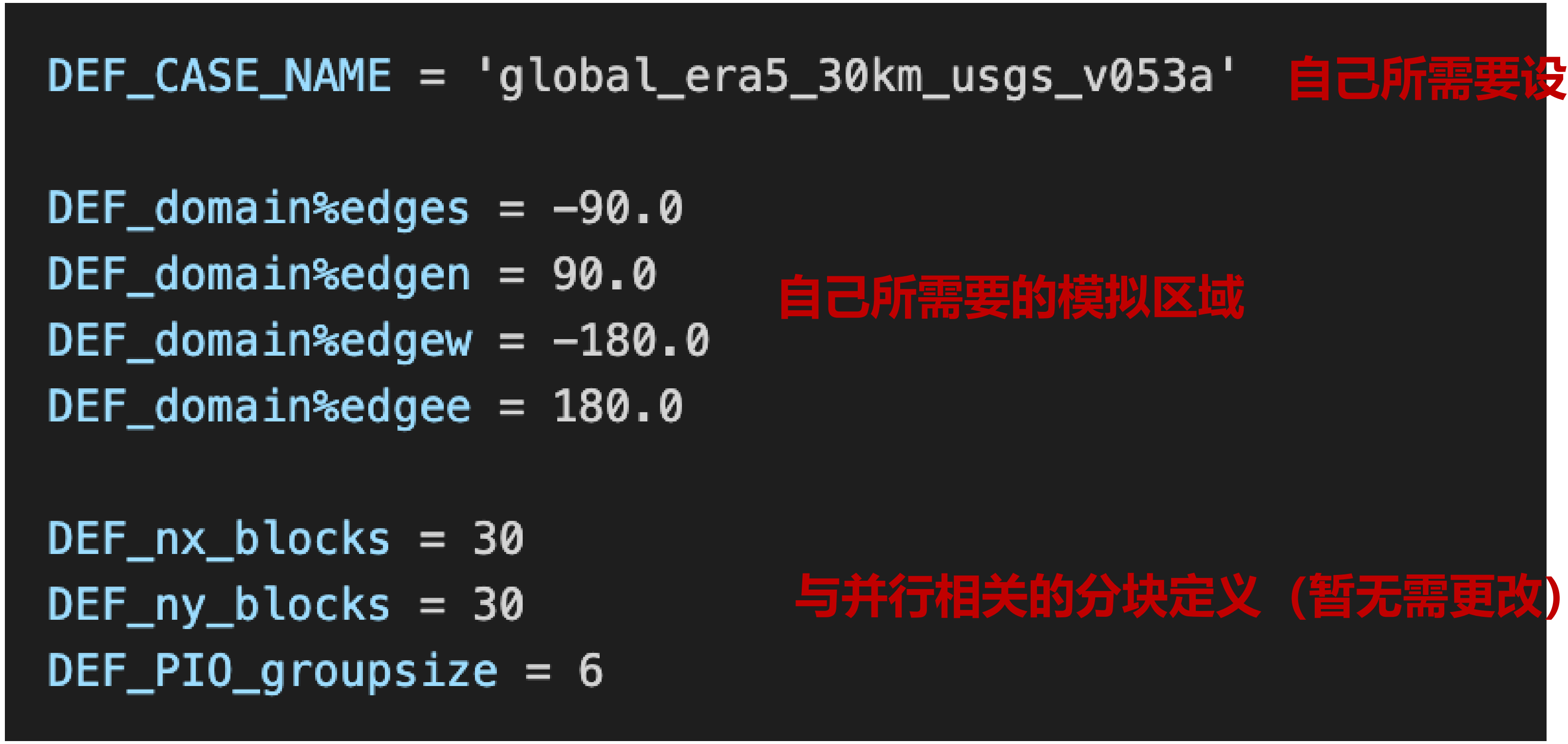
制作初始文件(运行mkinidata)

运行模式(运行main)

5. 模式安装

离线（全球/km）模式代码路径: /data/sysu_daiyj_01/CoLM202X_mpi

Namelist含义：



```
DEF_CASE_NAME = 'global_era5_30km_usgs_v053a'

DEF_domain%edges = -90.0
DEF_domain%edgen = 90.0
DEF_domain%edgew = -180.0
DEF_domain%edgee = 180.0

DEF_nx_blocks = 30
DEF_ny_blocks = 30
DEF_PIO_groupsize = 6
```

自己所需要设定的模拟的实验名

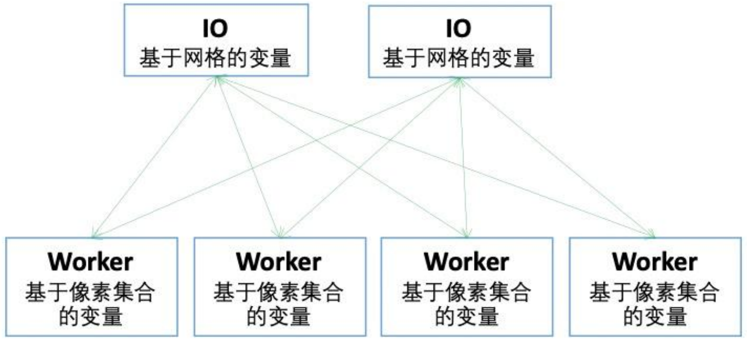
自己所需要的模拟区域

与并行相关的分块定义（暂无需更改）

5. 模式安装

离线（全球1km）模式版本并行框架

进程拓扑结构



IO
基于网格的变量

IO
基于网格的变量

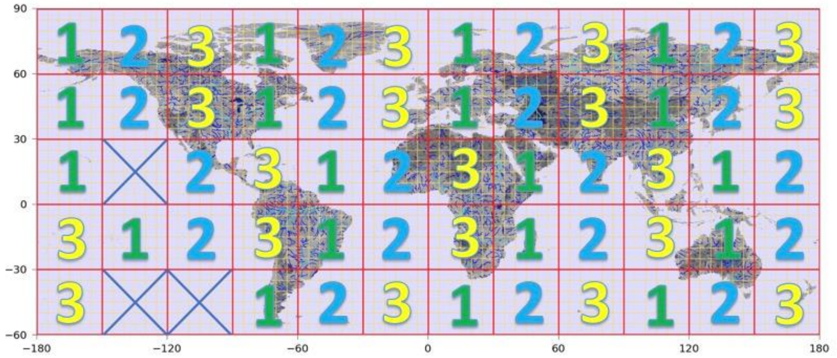
Worker
基于像素集合的变量

Worker
基于像素集合的变量

Worker
基于像素集合的变量

Worker
基于像素集合的变量

数据读写区域划分



任务分配及流程

IO节点：读取驱动 → 数据交换 → 输出数据 → 读取驱动 → 数据交换 → 输出数据 → 读取驱动 →

↑↓ ↑↓

Woker节点：数据交换 → 模型过程 → 数据交换 → 模型过程 →

通过

① 去中心化的进程拓扑结构
② 按块轮循的网格数据读写方式
③ 不同类型数据和模块的进程区分

实现了

① 更均衡的负载
② 更有效的内存利用
③ 更少的进程等待
④ 更快的读写
⑤ 更简单的耦合
⑥ 更大的数据量

5. 模式安装

离线(全球/km)模式代码路径: **/data/sysu_daiyj_01/CoLM202X_mpi**

Namelist含义：

```
DEF_simulation_time%greenwich     = .TRUE.
DEF_simulation_time%start_year    = 2001
DEF_simulation_time%start_month   = 1
DEF_simulation_time%start_day     = 1
DEF_simulation_time%start_sec     = 0
DEF_simulation_time%end_year      = 2001
DEF_simulation_time%end_month     = 12
DEF_simulation_time%end_day       = 31
DEF_simulation_time%end_sec       = 3600.
DEF_simulation_time%spinup_year   = 1979
DEF_simulation_time%spinup_month  = 1
DEF_simulation_time%spinup_day    = 1
DEF_simulation_time%spinup_sec    = 0
DEF_simulation_time%timestep      = 3600.
```

是否UTC（暂无需更改)

模拟起始时间

模拟结束时间

SpinUp的截止时间

模拟的时间步长

5. 模式安装

离线（全球1km）模式代码路径: **/data/sysu_daiyj_01/CoLM202X_mpi**

Namelist含义：

```
DEF_dir_rawdata  = '/tera04/zhwei/colm/data/CLMrawdata_usgs_v0.53/'
DEF_dir_output   = '../../cases/'

! ----- land units and land sets -----
! for GRIDBASED
DEF_file_landgrid = '/tera04/zhwei/colm/data/landdata/nc/landmask_usgs_30km.nc'
! for CATCHMENT
! DEF_dir_hydrodata = '/tera02/zhangsp/hillslope/output/columbia'
! DEF_max_hband = 25

! LAI setting
DEF_LAI_TRUE = .False.
```

地表基础数据的路径

landmask数据的路径

是否使用观测的叶面积指数的数据

5. 模式安装

离线(全球1km)模式代码路径: **/data/sysu_daiyj_01/CoLM202X_mpi**

Namelist含义：

```
! ------ history ----
DEF_nlon_hist = 1440
DEF_nlat_hist = 720
DEF_WRST_FREQ = 'YEARLY' ! write restart file frequency: HOURLY/DAILY/MONTHLY/YEARLY
DEF_HIST_FREQ = 'HOURLY' ! write history file frequency: HOURLY/DAILY/MONTHLY/YEARLY
DEF_HIST_groupby = 'DAY' ! history in one file: DAY/MONTH/YEAR
DEF_HIST_mode = 'one' ! history in one or block
DEF_REST_COMPRESS_LEVEL = 1
DEF_HIST_COMPRESS_LEVEL = 1
```

输出分辨率

输出文件的频率

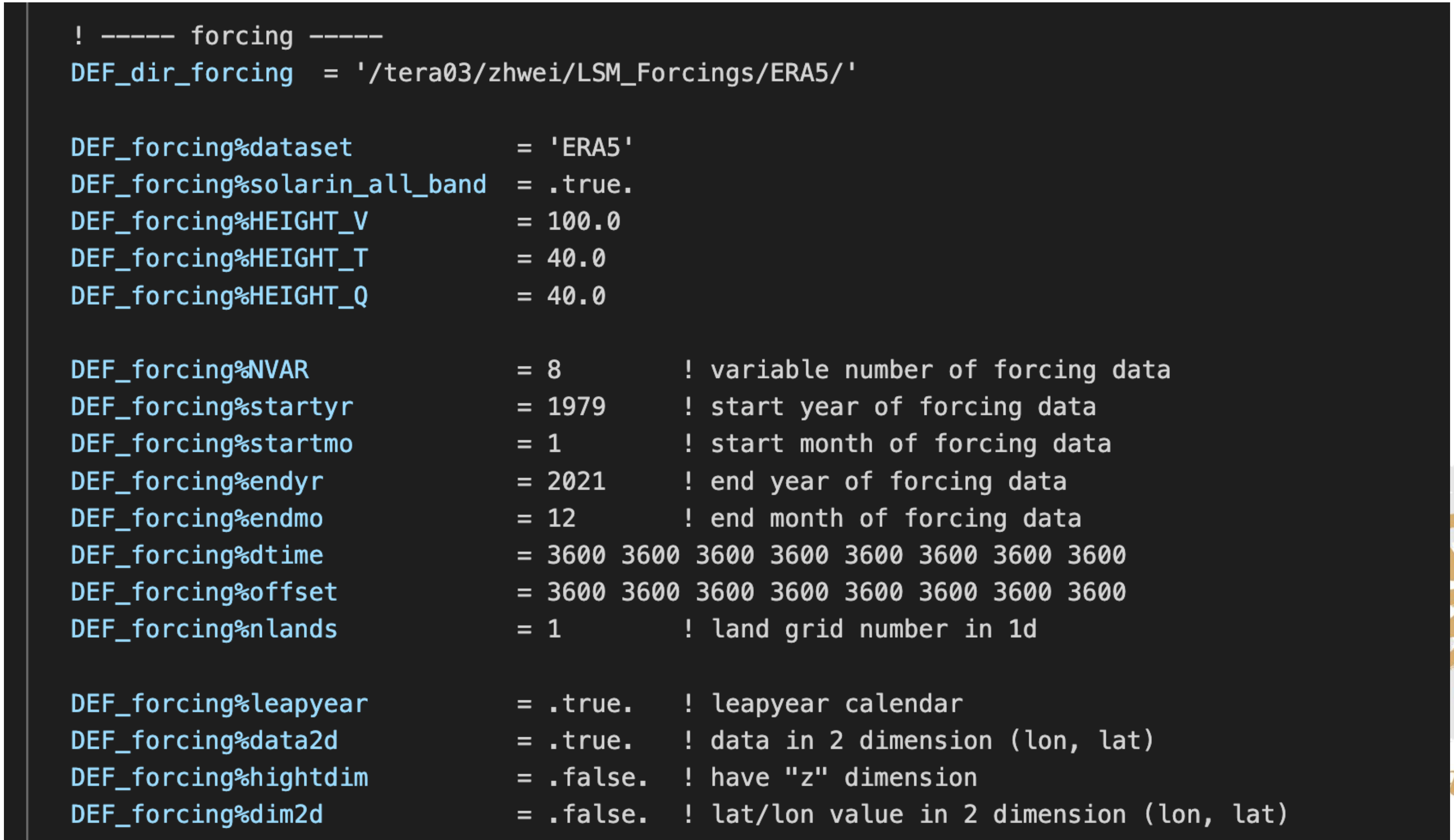
分块输出还是输出为一个文件

是否压缩

5. 模式安装

离线（全球/km）模式代码路径：**/data/sysu_daiyj_01/CoLM202X_mpi**

Namelist含义：

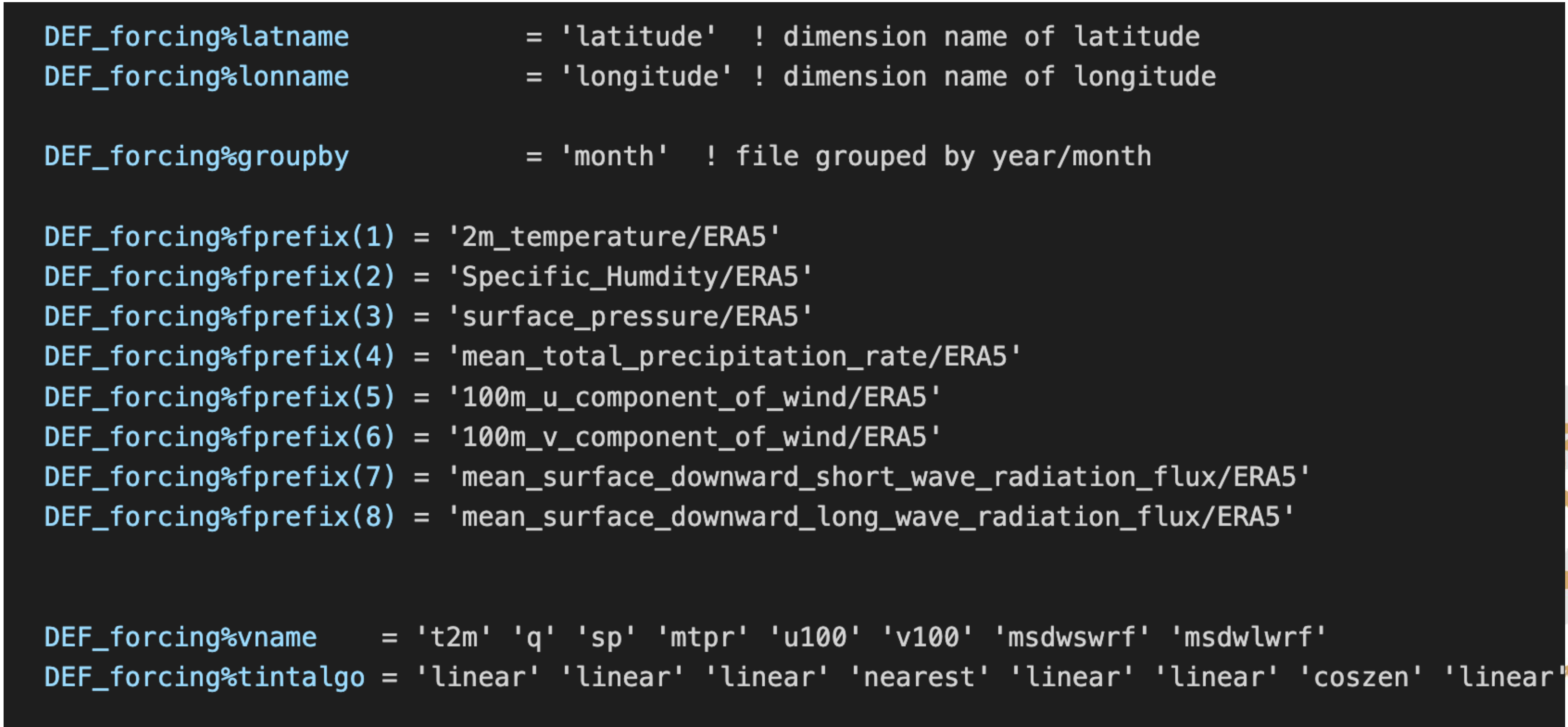


```
! ----- forcing -----
DEF_dir_forcing  = '/tera03/zhwei/LSM_Forcings/ERA5/'

DEF_forcing%dataset              = 'ERA5'
DEF_forcing%solarin_all_band     = .true.
DEF_forcing%HEIGHT_V             = 100.0
DEF_forcing%HEIGHT_T             = 40.0
DEF_forcing%HEIGHT_Q             = 40.0

DEF_forcing%NVAR                 = 8        ! variable number of forcing data
DEF_forcing%startyr              = 1979     ! start year of forcing data
DEF_forcing%startmo              = 1        ! start month of forcing data
DEF_forcing%endyr                = 2021     ! end year of forcing data
DEF_forcing%endmo                = 12       ! end month of forcing data
DEF_forcing%dtime                = 3600 3600 3600 3600 3600 3600 3600 3600
DEF_forcing%offset               = 3600 3600 3600 3600 3600 3600 3600 3600
DEF_forcing%nlands               = 1        ! land grid number in 1d

DEF_forcing%leapyear             = .true.   ! leapyear calendar
DEF_forcing%data2d               = .true.   ! data in 2 dimension (lon, lat)
DEF_forcing%hightdim             = .false.  ! have "z" dimension
DEF_forcing%dim2d                = .false.  ! lat/lon value in 2 dimension (lon, lat)
```

5. 模式安装

离线(全球1km)模式代码路径: /data/sysu_daiyj_01/CoLM202X_mpi

Namelist含义：



```
DEF_forcing%latname           = 'latitude'  ! dimension name of latitude
DEF_forcing%lonname           = 'longitude' ! dimension name of longitude

DEF_forcing%groupby           = 'month'  ! file grouped by year/month

DEF_forcing%fprefix(1) = '2m_temperature/ERA5'
DEF_forcing%fprefix(2) = 'Specific_Humdity/ERA5'
DEF_forcing%fprefix(3) = 'surface_pressure/ERA5'
DEF_forcing%fprefix(4) = 'mean_total_precipitation_rate/ERA5'
DEF_forcing%fprefix(5) = '100m_u_component_of_wind/ERA5'
DEF_forcing%fprefix(6) = '100m_v_component_of_wind/ERA5'
DEF_forcing%fprefix(7) = 'mean_surface_downward_short_wave_radiation_flux/ERA5'
DEF_forcing%fprefix(8) = 'mean_surface_downward_long_wave_radiation_flux/ERA5'


DEF_forcing%vname    = 't2m' 'q' 'sp' 'mtpr' 'u100' 'v100' 'msdwswrf' 'msdwlwrf'
DEF_forcing%tintalgo = 'linear' 'linear' 'linear' 'nearest' 'linear' 'linear' 'coszen' 'linear'
```

5. 模式安装

离线（全球1km）模式代码路径：**/data/sysu_daiyj_01/CoLM202X_mpi**

Namelist含义：

```
! ----- history -----
DEF_hist_vars%xy_us         = .true.
DEF_hist_vars%xy_vs         = .true.
DEF_hist_vars%xy_t          = .true.
DEF_hist_vars%xy_q          = .true.
DEF_hist_vars%xy_prc        = .true.
DEF_hist_vars%xy_prl        = .true.
DEF_hist_vars%xy_pbot       = .true.
DEF_hist_vars%xy_frl        = .true.
DEF_hist_vars%xy_solarin    = .true.
DEF_hist_vars%xy_rain       = .true.
DEF_hist_vars%xy_snow       = .true.
DEF_hist_vars%taux          = .true.
DEF_hist_vars%tauy          = .true.
DEF_hist_vars%fsena         = .true.
DEF_hist_vars%lfevpa        = .true.
DEF_hist_vars%fevpa         = .true.
DEF_hist_vars%fsenl         = .true.
```

指定哪些变量输出

5. 模式安装

离线（全球1km）模式代码路径：**/data/sysu_daiyj_01/CoLM202X_mpi**

模拟器运行：

```
#!/bin/bash
#SBATCH -J global_usgs_CB
#SBATCH -p normal
#SBATCH -N 64
#SBATCH -n 3840
#SBATCH --ntasks-per-node=60
#SBATCH --mem=220G
#SBATCH -o colm.o%j
#SBATCH -e colm.e%j
#SBATCH --exclusive
#SBATCH -t 7-24:00

module load mathlib/netcdf/intel/4.4.1

export I_MPI_FABRICS=shm:dapl
export I_MPI_DAPL_UD=1
export I_MPI_DAPL_UD_RDMA_MIXED=1
export I_MPI_LARGE_SCALE_THRESHOLD=8192
export I_MPI_DAPL_UD_ACK_SEND_POOL_SIZE=8704
export I_MPI_DAPL_UD_ACK_RECV_POOL_SIZE=8704
export I_MPI_DAPL_UD_RNDV_EP_NUM=2

export DAPL_UCM_REP_TIME=8000 #  REQUEST timer, waiting for REPLY in millisecs
export DAPL_UCM_RTU_TIME=8000 #  REPLY timer, waiting for RTU in millisecs
export DAPL_UCM_RETRY=10 #  REQUEST and REPLY retries
export DAPL_UCM_CQ_SIZE=2000
export DAPL_UCM_QP_SIZE=2000

export DAPL_UCM_DREQ_RETRY=4 #default == 1
export DAPL_UCM_DREP_TIME=200 #default == 200ms
export DAPL_UCM_WAIT_TIME=10000 #default == 60000ms

ulimit -s unlimited
srun --mpi=pmi2 --cpu-bind=cores ../mksrfdata/mksrfdata.x input_global_grid_cruncep_usgs.nml
srun --mpi=pmi2 --cpu-bind=cores ../mkinidata/initial.x input_global_grid_cruncep_usgs.nml
srun --mpi=pmi2 --cpu-bind=cores ../main/clm.x input_global_grid_cruncep_usgs.nml
```

6. 模式评估诊断包ILAMB与评估演示

ILAMB的打分原理：基于基准数据，计算模式结果的各个误差指标，换算为分数。

各个误差指标分数加权平均得到总体分数$S_{overall}$：

$$S_{overall} = \frac{S_{bias} + 2S_{rmse} + S_{phase} + S_{iav} + S_{dist}}{1 + 2 + 1 + 1 + 1}$$

S_{bias}：偏差得分

S_{rmse}：均方根误差得分

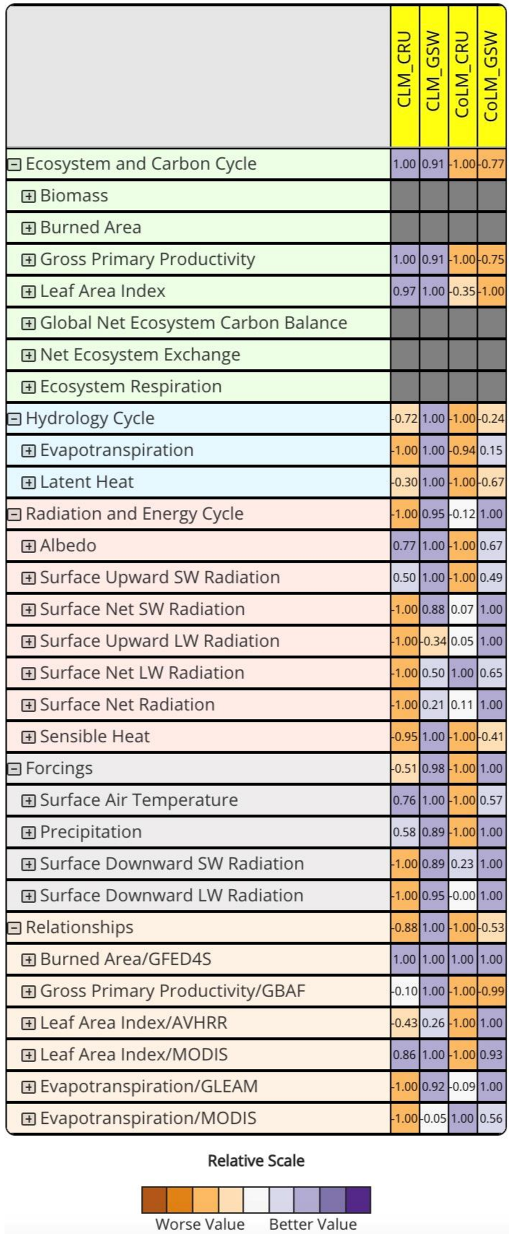
S_{phase}：位相偏移得分

S_{iav}：年际变异得分

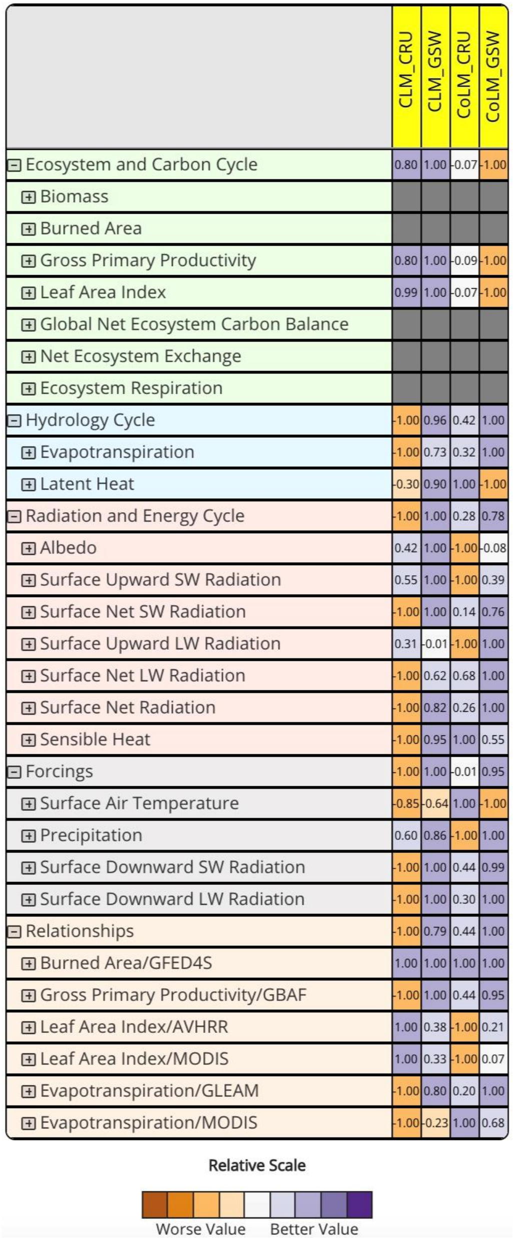
S_{dist}：空间分布得分

模拟试验	模式名称	大气强迫
CoLM_CRU	CoLM	CRUNCEP_v7
CoLM_GSW	CoLM	GSWP3_v1
CLM_CRU	CLM5	CRUNCEP_v7
CLM_GSW	CLM5	GSWP3_v1

6. 模式评估诊断包ILAMB与评估演示



	CLM_CRU	CLM_GSW	CoLM_CRU	CoLM_GSW
⊟ Ecosystem and Carbon Cycle	1.00	0.91	-1.00	-0.77
⊞ Biomass				
⊞ Burned Area				
⊞ Gross Primary Productivity	1.00	0.91	-1.00	-0.75
⊞ Leaf Area Index	0.97	1.00	-0.35	-1.00
⊞ Global Net Ecosystem Carbon Balance				
⊞ Net Ecosystem Exchange				
⊞ Ecosystem Respiration				
⊟ Hydrology Cycle	-0.72	1.00	-1.00	-0.24
⊞ Evapotranspiration	-1.00	1.00	-0.94	0.15
⊞ Latent Heat	-0.30	1.00	-1.00	-0.67
⊟ Radiation and Energy Cycle	-1.00	0.95	-0.12	1.00
⊞ Albedo	0.77	1.00	-1.00	0.67
⊞ Surface Upward SW Radiation	0.50	1.00	-1.00	0.49
⊞ Surface Net SW Radiation	-1.00	0.88	0.07	1.00
⊞ Surface Upward LW Radiation	-1.00	-0.34	0.05	1.00
⊞ Surface Net LW Radiation	-1.00	0.50	1.00	0.65
⊞ Surface Net Radiation	-1.00	0.21	0.11	1.00
⊞ Sensible Heat	-0.95	1.00	-1.00	-0.41
⊟ Forcings	-0.51	0.98	-1.00	1.00
⊞ Surface Air Temperature	0.76	1.00	-1.00	0.57
⊞ Precipitation	0.58	0.89	-1.00	1.00
⊞ Surface Downward SW Radiation	-1.00	0.89	0.23	1.00
⊞ Surface Downward LW Radiation	-1.00	0.95	-0.00	1.00
⊟ Relationships	-0.88	1.00	-1.00	-0.53
⊞ Burned Area/GFED4S	1.00	1.00	1.00	1.00
⊞ Gross Primary Productivity/GBAF	-0.10	1.00	-1.00	-0.99
⊞ Leaf Area Index/AVHRR	-0.43	0.26	-1.00	1.00
⊞ Leaf Area Index/MODIS	0.86	1.00	-1.00	0.93
⊞ Evapotranspiration/GLEAM	-1.00	0.92	-0.09	1.00
⊞ Evapotranspiration/MODIS	-1.00	-0.05	1.00	0.56

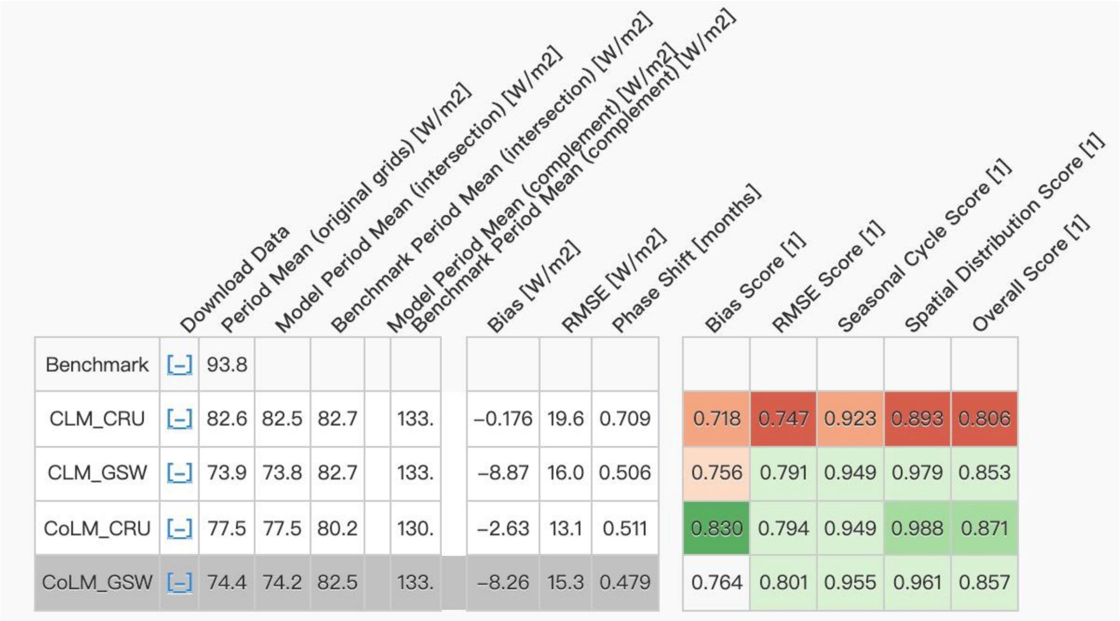


	CLM_CRU	CLM_GSW	CoLM_CRU	CoLM_GSW
⊟ Ecosystem and Carbon Cycle	0.80	1.00	-0.07	-1.00
⊞ Biomass				
⊞ Burned Area				
⊞ Gross Primary Productivity	0.80	1.00	-0.09	-1.00
⊞ Leaf Area Index	0.99	1.00	-0.07	-1.00
⊞ Global Net Ecosystem Carbon Balance				
⊞ Net Ecosystem Exchange				
⊞ Ecosystem Respiration				
⊟ Hydrology Cycle	-1.00	0.96	0.42	1.00
⊞ Evapotranspiration	-1.00	0.73	0.32	1.00
⊞ Latent Heat	-0.30	0.90	1.00	-1.00
⊟ Radiation and Energy Cycle	-1.00	1.00	0.28	0.78
⊞ Albedo	0.42	1.00	-1.00	-0.08
⊞ Surface Upward SW Radiation	0.55	1.00	-1.00	0.39
⊞ Surface Net SW Radiation	-1.00	1.00	0.14	0.76
⊞ Surface Upward LW Radiation	0.31	-0.01	-1.00	1.00
⊞ Surface Net LW Radiation	-1.00	0.62	0.68	1.00
⊞ Surface Net Radiation	-1.00	0.82	0.26	1.00
⊞ Sensible Heat	-1.00	0.95	1.00	0.55
⊟ Forcings	-1.00	1.00	-0.01	0.95
⊞ Surface Air Temperature	-0.85	-0.64	1.00	-1.00
⊞ Precipitation	0.60	0.86	-1.00	1.00
⊞ Surface Downward SW Radiation	-1.00	1.00	0.44	0.99
⊞ Surface Downward LW Radiation	-1.00	1.00	0.30	1.00
⊟ Relationships	-1.00	0.79	0.44	1.00
⊞ Burned Area/GFED4S	1.00	1.00	1.00	1.00
⊞ Gross Primary Productivity/GBAF	-1.00	1.00	0.44	0.95
⊞ Leaf Area Index/AVHRR	1.00	0.38	-1.00	0.21
⊞ Leaf Area Index/MODIS	1.00	0.33	-1.00	0.07
⊞ Evapotranspiration/GLEAM	-1.00	0.80	0.20	1.00
⊞ Evapotranspiration/MODIS	-1.00	-0.23	1.00	0.68

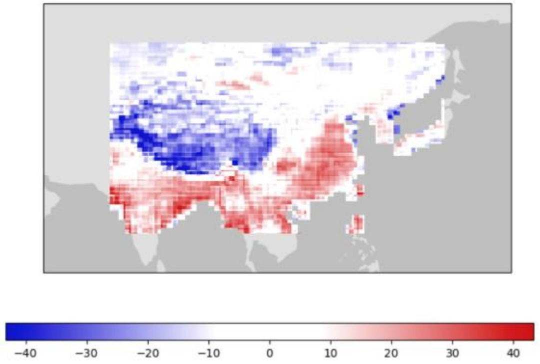
总体分数表明：针对陆面物理过程，CoLM与CLM5具有等价的模拟性能；针对中国区域的水文和辐射过程模拟CoLM优于CLM5 。

6. 模式评估诊断包ILAMB与评估演示

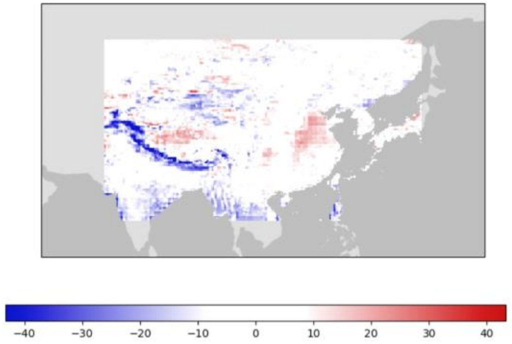
地表净辐射模拟状况(基于CERES)



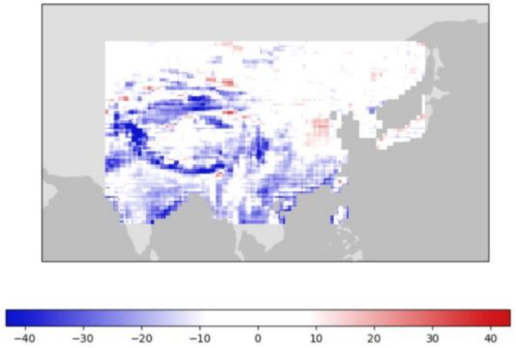
	Download Data	Period Mean (original grids) [W/m2]	Model Period Mean (intersection) [W/m2]	Benchmark Period Mean (intersection) [W/m2]	Model Period Mean (complement) [W/m2]	Benchmark Period Mean (complement) [W/m2]	Bias [W/m2]	RMSE [W/m2]	Phase Shift [months]	Bias Score [1]	RMSE Score [1]	Seasonal Cycle Score [1]	Spatial Distribution Score [1]	Overall Score [1]
Benchmark	[-]	93.8												
CLM_CRU	[-]	82.6	82.5	82.7		133.	−0.176	19.6	0.709	0.718	0.747	0.923	0.893	0.806
CLM_GSW	[-]	73.9	73.8	82.7		133.	−8.87	16.0	0.506	0.756	0.791	0.949	0.979	0.853
CoLM_CRU	[-]	77.5	77.5	80.2		130.	−2.63	13.1	0.511	0.830	0.794	0.949	0.988	0.871
CoLM_GSW	[-]	74.4	74.2	82.5		133.	−8.26	15.3	0.479	0.764	0.801	0.955	0.961	0.857



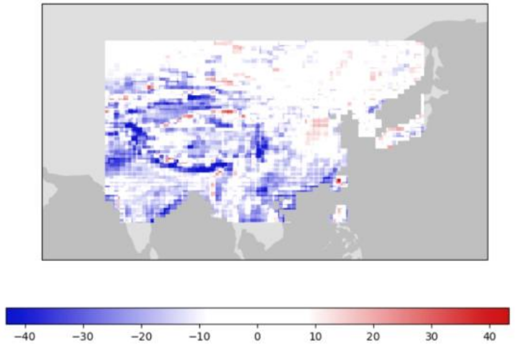
CLM5_CRU



CoLM_CRU



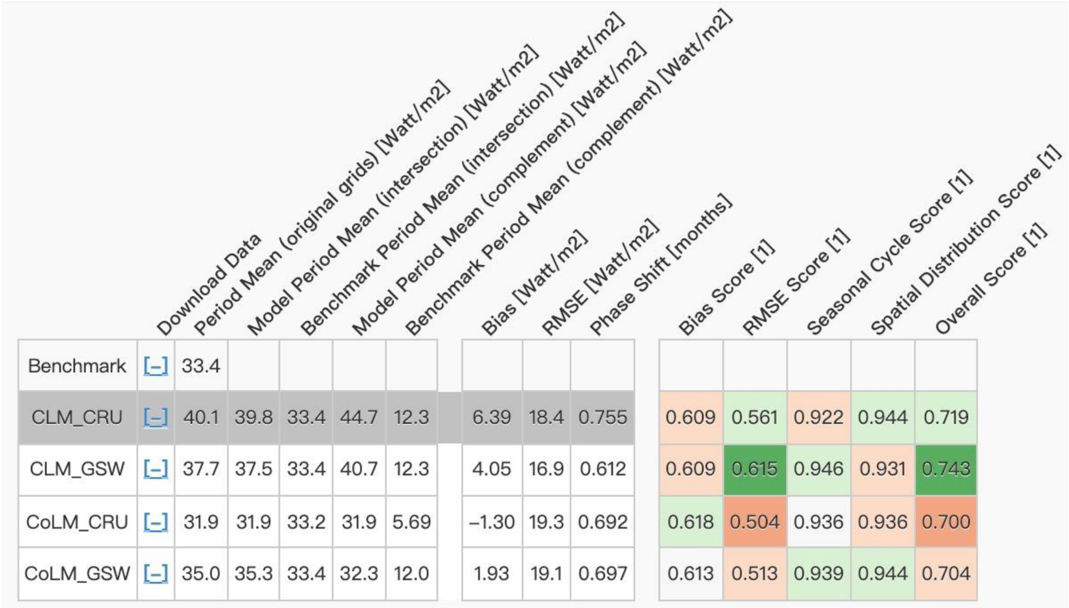
CoLM_GSW



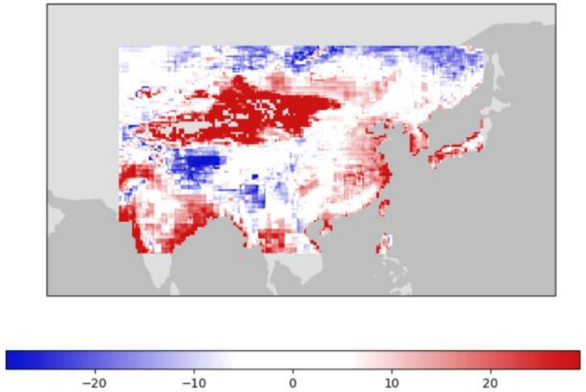
CLM5_GSW

6. 模式评估诊断包ILAMB与评估演示

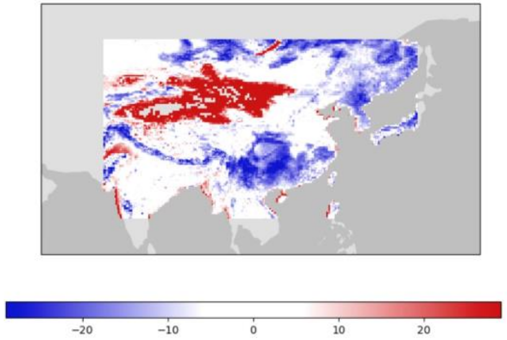
感热通量模拟状况（基于GBAF）



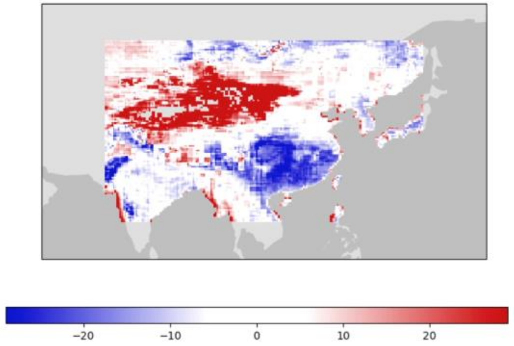
	Download Data	Period Mean (original grids) [Watt/m2]	Model Period Mean (intersection) [Watt/m2]	Benchmark Period Mean (intersection) [Watt/m2]	Model Period Mean (complement) [Watt/m2]	Benchmark Period Mean (complement) [Watt/m2]	Bias [Watt/m2]	RMSE [Watt/m2]	Phase Shift [months]	Bias Score [1]	RMSE Score [1]	Seasonal Cycle Score [1]	Spatial Distribution Score [1]	Overall Score [1]
Benchmark	[-]	33.4												
CLM_CRU	[-]	40.1	39.8	33.4	44.7	12.3	6.39	18.4	0.755	0.609	0.561	0.922	0.944	0.719
CLM_GSW	[-]	37.7	37.5	33.4	40.7	12.3	4.05	16.9	0.612	0.609	0.615	0.946	0.931	0.743
CoLM_CRU	[-]	31.9	31.9	33.2	31.9	5.69	−1.30	19.3	0.692	0.618	0.504	0.936	0.936	0.700
CoLM_GSW	[-]	35.0	35.3	33.4	32.3	12.0	1.93	19.1	0.697	0.613	0.513	0.939	0.944	0.704



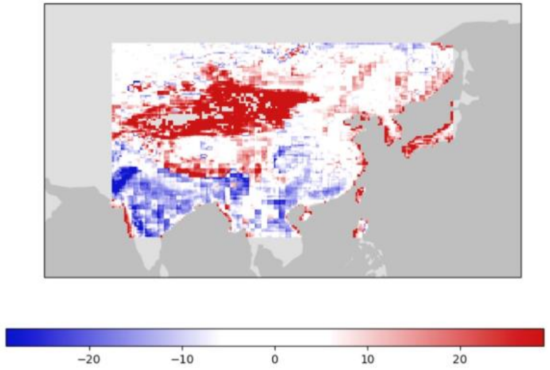
CLM5_CRU



CoLM_CRU



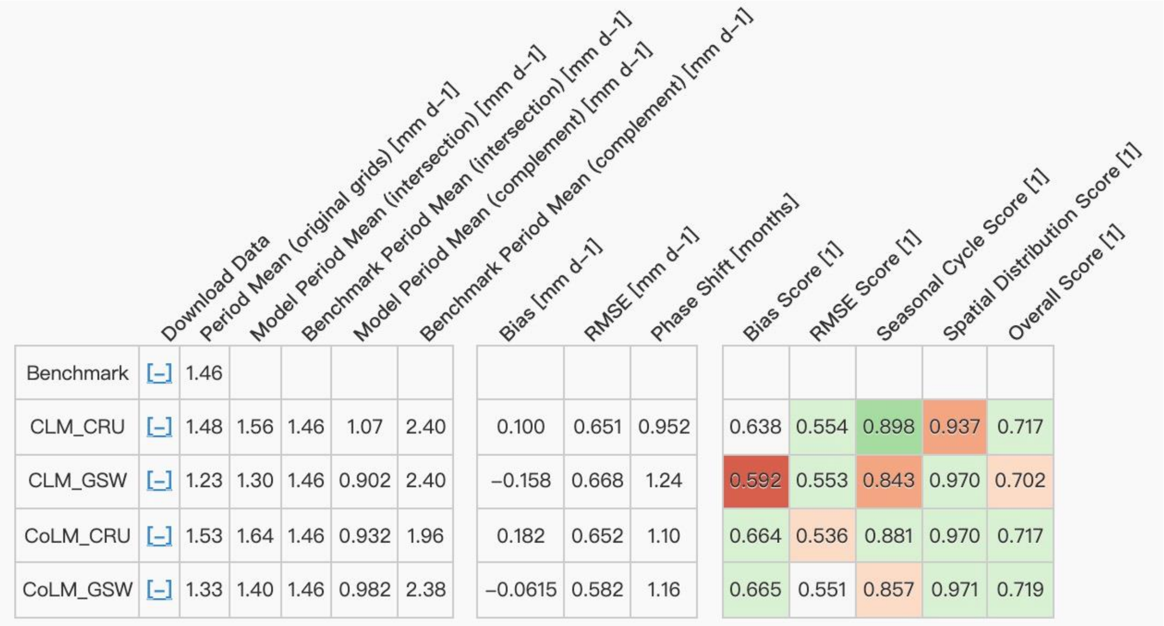
CoLM_GSW



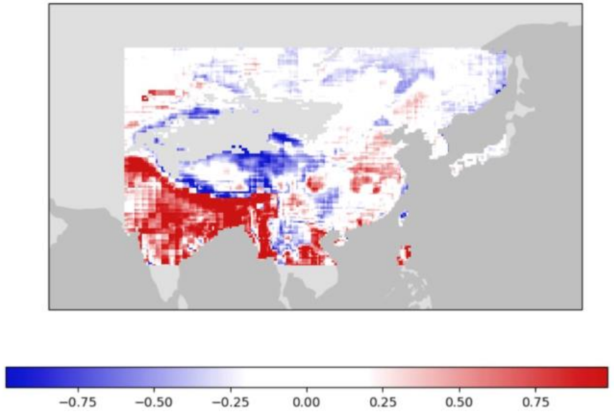
CLM5_GSW

6. 模式评估诊断包ILAMB与评估演示

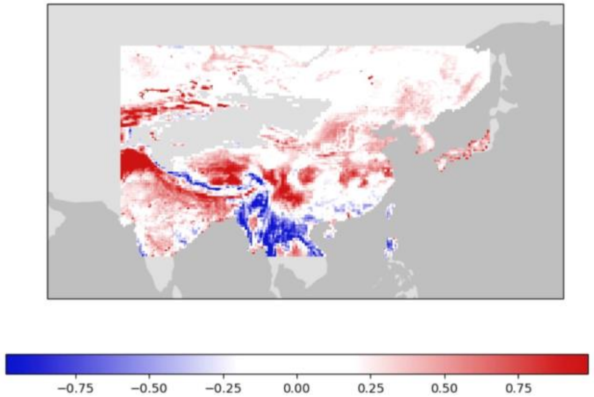
蒸散发模拟状况（基于MODIS）



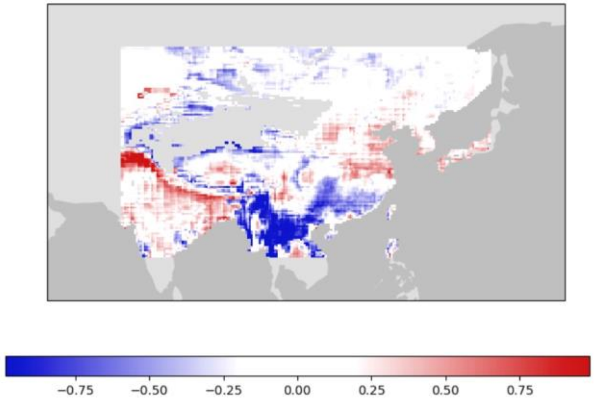
	Download Data	Period Mean (original grids) [mm d-1]	Model Period Mean (intersection) [mm d-1]	Benchmark Period Mean (intersection) [mm d-1]	Model Period Mean (complement) [mm d-1]	Benchmark Period Mean (complement) [mm d-1]	Bias [mm d-1]	RMSE [mm d-1]	Phase Shift [months]	Bias Score [1]	RMSE Score [1]	Seasonal Cycle Score [1]	Spatial Distribution Score [1]	Overall Score [1]
Benchmark	[-]	1.46												
CLM_CRU	[-]	1.48	1.56	1.46	1.07	2.40	0.100	0.651	0.952	0.638	0.554	0.898	0.937	0.717
CLM_GSW	[-]	1.23	1.30	1.46	0.902	2.40	−0.158	0.668	1.24	0.592	0.553	0.843	0.970	0.702
CoLM_CRU	[-]	1.53	1.64	1.46	0.932	1.96	0.182	0.652	1.10	0.664	0.536	0.881	0.970	0.717
CoLM_GSW	[-]	1.33	1.40	1.46	0.982	2.38	−0.0615	0.582	1.16	0.665	0.551	0.857	0.971	0.719



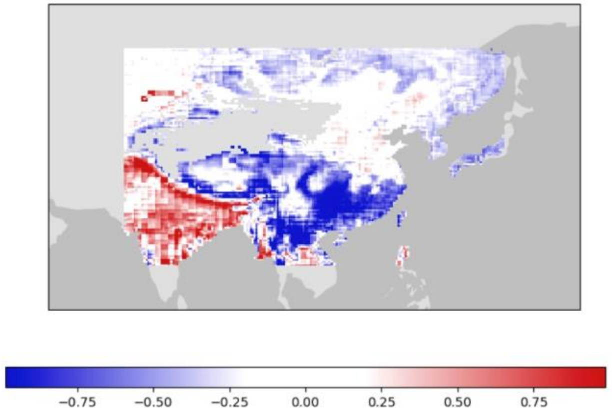
CLM5_CRU



CoLM_CRU



CoLM_GSW



CLM5_GSW

谢谢