# WRF\_v3.1程序调用结构

## WRF.f90

编译最耗时的是module\_comm\_dm.f90，最好不要修改了。

WRF的模型代码大致可分为几个部分：

* 模型主框架：
* 网格、数值方法
* 物理模块(mp, bl, sf, pbl, ra, cu, fire, chem...)
* 中间层(mediation layer)
* 数据同化(fdda\_obs)
* 并行计算(RSL Lite)
* 输入输出(io\_int, io\_nc, io\_grib1)

WRF.f90: use module\_wrf\_top ->wrf\_init() wrf\_run() wrf\_finalize()

WRF.f90主程序，调用wrf\_run() -> call integrate(head\_grid, ) (核心子程序， recursive subroutine) -> call solve\_interface (interface subroutines), 包括solve\_em()

solve\_em(grid, config\_flags,...) -> ! 求解方程的核心子程序，调用以下程序：

rk\_step\_prep

RK LOOP: IF (rk\_step == 1) THEN ! 程序初始化

-> get\_ijk\_from\_grid( ) ! 获取网格节点编号(ijk)

! RK的第一部分考虑了很多物理过程，包括降雨、辐射、表面模型、行星边界层模型、积云参数化、SCM、数据同化

-> first\_rk\_step\_part1( )-> { init\_zero\_tendency,

phy\_prep ! in module\_big\_step\_utilities\_em 物理过程计算的准备(初始化)

pre\_radiation\_driver-> radconst、toposhad\_init、toposhad

radiation\_driver -> SELECT CASE(lw\_physics)\ SELECT CASE(sw\_physics)

surface\_driver -> Land surface model

pbl\_driver

sfire\_driver\_em ! 火灾计算模块

cumulus\_driver

force\_scm

fddagd\_driver

}

! RK的第2步考虑了边界条件和数值格式

-> first\_rk\_step\_part2( ) ->{

calculate\_phy\_tend

set\_physical\_bc3d

phy\_bc

update\_phy\_ten

tke\_rhs

vertical\_diffusion\_2

horizontal\_diffusion\_2

}

ENDIF

rk\_tendency

call advance\_ppt( )

**！ Micro-physics process**

**IF** (config\_flags%mp\_physics /= 0) **then**

moist\_physics\_prep\_em

call microphysics\_driver

! micro-physics process。。。。

calc\_p\_rho\_phi

call diagnose\_w ! 静水压力假设时，开启

call pxft ! 极地区域的fft计算

！ 设置物理过程的边界条件

call diagnostic\_output\_calc ! 输出结果的诊断

ENDIF

## Surface Layer Model

在RK迭代过程中，调用相应的参数化模块：

surface\_driver() -> {

sst\_skin\_update

tmnupdate

! Surface Layer

**SELECT CASE**(sf\_sfclay\_physics)

**CASE** (SFCLAYSCHEME)

**CASE** (PXSFCSCHEME) ! Pleim-Xiu Land surface model

**CASE**(MYNNSFCSCHEME)

**END SELECT** sfclay\_select

## module\_surface\_driver:

! Land surface model

sfc\_select: **SELECT CASE**(sf\_surface\_physics)

**CASE** (SLABSCHEME) ! WRFV3.1仅在此实施了海洋混合层计算

call SLAB() -> SLAB1D() -> call OCEANML

**CASE** (LSMSCHEME)

call lsm() -> 考虑了周到的下垫面条件

**CASE** (RUCLSMSCHEME)

call LSMRUC()

**CASE** (PXLSMSCHEME)

call PXLSM()

}

lsm() ->{

SFLX

! 城市下垫面条件

**IF** (SF\_URBAN\_PHYSICS == 1 ) **THEN call urban()**

**IF** (SF\_URBAN\_PHYSICS == 2) **THEN call BEP() ! new in v3.1**

**IF** (SF\_URBAN\_PHYSICS == 3) **THEN call BEP\_BEM() ! new in v3.2**

}

## module\_pbl\_driver:

行星边界层模块

pbl\_driver() -> {

SELECT CASE(sf\_urban\_physics): BEPSCHEME

SELECT CASE(sf\_sfclay\_physics): MYJSCSCHIEME

pbl\_select: **SELECT CASE**(bl\_pbl\_physics)

**CASE** (YSUSCHEME)

call ysu()

**CASE** (MRFSCHEME)

call mrf()

**CASE** (GFSSCHEME)

call bl\_gfs()

**CASE** (MYJPBLSCHEME)

call myjpbl() or call myjurb

**CASE** (QNSEPBLSCHEME)

**CALL** qnsepbl

**CASE** (ACMPBLSCHEME)

**CALL** ACMPBL

**CASE** (MYNNPBLSCHEME2, MYNNPBLSCHEME3)

**CALL** mynn\_bl\_driver

**CASE** (BOULACSCHEME)

**CALL** BOULAC

END SELECT

call gwdo () ! Gravity wave drag option

call diff3d() !

*Diffusion in WRF is categorized under two parameters, the diffusion option and the K option. The diffusion option selects how the derivatives used in diffusion are calculated, and the K option selects how the K coefficients are calculated. Note that when a PBL option is selected, vertical diffusion is done by the PBL scheme, and not by the diffusion scheme. In Version 3, vertical diffusion is also linked to the surface fluxes.*

}

cumulus\_driver ->{

}

rk\_step\_prep –>{

calculate\_full

calc\_mu\_uv

couple\_momentum

calc\_ww\_cp

calc\_cq

calc\_alt

calc\_php

}

rk\_tendency->{

zero\_tend

advect\_u

advect\_v

**IF** (non\_hydrostatic) advect\_w

advect\_scalar

rhs\_ph

horizontal\_pressure\_gradient

**IF** (non\_hydrostatic) pg\_buoy\_w

w\_damp

}

## WRF计算输入输出

输入

module\_config.f90:

initial\_config->读取namelist.input，还有：namelist.output

! namelist.input一共有11个部分

input\_inname = "wrfinput\_d<domain>"

input\_outname = "wrfinput\_d<domain>"

bdy\_inname = "wrfbdy\_d<domain>"

bdy\_outname = "wrfbdy\_d<domain>"

rst\_inname = "wrfrst\_d<domain>\_<date>"

rst\_outname = "wrfrst\_d<domain>\_<date>"

! 所有网格都使用最后一个网格的micro-physics过程

**DO** i = 1, max\_dom

mp\_physics(i) = mp\_physics(max\_dom)

**ENDDO**

! 设置为读入namelist的控制参数

model\_config\_rec % run\_days = run\_days

输出

WRF使用三种输出格式：Internal IO (MM5格式), netCDF, GRIB1，最常用的是netCDF格式。

output\_在mediation\_integrte.f90中

recursive subroutine integrate() -> {

med\_before\_solve\_io -> med\_hist\_out -> output\_history –> output\_wrf

!WRF输出

solve\_interface ! 核心计算，时间层推进

med\_after\_solve\_io ! -> calc\_ts

med\_last\_solve\_io !

}

WRF的netcDF文件输出在wrf\_io.f90 ->

output\_wrf-> wrf\_histout (很多输出子程序，如auxhist1in~auxhist11in、auxhist1out~auxhist11out)->

ext\_ncd\_write\_field-> FieldIO ->

ext\_ncd\_IntFieldIO

ext\_ncd\_RealFieldIO

ext\_ncd\_DoubleFieldIO

ext\_ncd\_LogicalFieldIO

module\_io.f90 :

wrf\_ext\_write\_field -> wrf\_write\_field -> wrf\_write\_field1 -> collect\_fld\_and\_call\_pkg (ext\_ncd\_write\_field, )

quilt -> ext\_ncd\_write\_field

write\_outbuf -> ext\_ncd\_write\_field

mpi\_init()在WRF中有3处：

init\_modules->ini\_module\_dm:

init\_module\_wrf\_quilt

split\_communicator

mpi\_finalize()调用了2次：

wrf\_shutdown->wrf\_dm\_shutdown: call mpi\_finalize

quilt-> if(obufsize<0) call mpi\_finalize()

## 没有用处的几个子程序：

init\_modules\_em

module\_solvedebug\_em

med\_calc\_model\_time

dummy\_interp\_em

module\_data\_gocart\_dust

## V3.1和V3.2的区别

V3.2相比较V3.1，有较大改变。

v3.2在noah\_lsm中增强了城市下垫面的计算。

需要修改sf\_slab surface\_driver, 使每个sf模型都能计算oml

海洋混合层在bl\_init和module\_surface\_driver中，使用USE module\_sf\_oml

**Surface Layer (sf\_sfclay\_physics)**

f. *iz0tlnd* = 1 (for *sf\_sfclay\_physics* = 1 or 2), Chen-Zhang thermal roughness length over land, which depends on vegetation height, 0 = original thermal roughness length in each sfclay option. New in Version 3.2.

**3.3 Urban Surface (sf\_urban\_physics – replacing old switch ucmcall)**

c. BEM (3). Building Energy Model. Adds to BEP, building energy budget with heating and cooling systems. Works with same options as BEP. New in Version 3.2.

## 问题：

输出模块有很多种，如auxhist1in~auxhist11in

auxhist1out~auxhist11out，都是什么格式呢？

quilt?

input\_aux\_model\_input1？

F90文件是怎么生成的？