

WRF MANUAL

WRF 安装运行入门指南

(WPS WRF_SI WRFV2.2 NCARG GrADS)

2007.04.10

aioply 编辑整理

WRF 安装运行入门指南

目录

前言	2
1. WRF 模式简介	3
2. 准备工作 (SSH 和 NetCDF)	5
3. WPS+WRFV2.2 安装运行简介	7
3.0. ~收集数据~	7
3.1. ~安装前奏~	9
~ WPS ~	
3.2. ~安装 WPS~	10
3.3. ~运行 WPS~	12
~ WRFV2.2 ~	
3.4. ~安装 WRFV2.2~	17
3.5. ~运行 WRFV2.2~	21
4. WRF_SI+WRFV2.2 安装运行简介	24
4.0. ~收集数据~	24
~ WRF_SI ~	
4.1. ~安装 WRF_SI~	24
4.1.1. ~定义环境变量~	24
4.1.2. ~安装 WRF_SI~	27
4.1.3. ~使用 WRFSI 的 GUI~	27
4.2. ~运行 WRF_SI~	28
STEP1: Localize model domain and create static files	28
STEP2: DeGrib GRIB files	29
STEP3: Interpolate meteorological data	31
~ WRFV2.2 ~	
4.3. ~安装 WRFV2.2~	33
4.4. ~运行 WRFV2.2~	36
5. 安装运行 WRF2GrADS	39
6. 在 UNIX 下安装 GrADS	43
7. 利用其它数据的练习 (ds083.2)	45
附录1: 安装NCAR Graphic	48
附录2: 关于WRF_SI2.0中wrfsi.nl的参数配置说明 (中文版)	50
附录3: 关于WRFV2.2中namelist.input的参数配置说明 (中文版)	55
附录4: 一些简单的UNIX命令	66

前言

连我自己也没想到，还会接着 WRF 版本的更新，不自量力地整理出第三版入门指南。但我想这将是自己能整理出来的最后一个版本了。曾经在发布第二版时承诺要增加 WRF 的 `namelist.input` 和 WRF2GrADS 的 `control_file` 文件的翻译以及 NCARG 和 WRF-3DVAR 的安装运行入门等相关内容。可惜自己的能力和精力有限，`control_file` 和 WRF-3DVAR 的工作没有做完，十分抱歉。不过有兴趣的人可以接着做。

由于第二版和第三版的初衷一样，就把第二版的前言稍作修改就又再贴上了。

我是 WRF 的初学者。

在自己刚接触 WRF 时亦曾在 Google 上搜索过 WRF 中文手册之类的东西，但没有任何收获。WRF 主页提供的教程虽然详细，但对于不熟悉 UNIX 系统和头一次接触气象模式的人来说，几乎就是无从下手。于是师姐 RODA 把自己当时的自学笔记借给我。我十分感谢她的帮助。后来我在自学的基础上，又加以补充和整理，编辑成了《WRF 安装运行入门指南》（WRF_SI+WRF 版本，2006.10.28 成稿）。随后，自己又在运转 WRF 的过程中积累经验，又赶上 WPS 的发放，就一并整理了出来，分享给大家。希望这本指南能对 WRF 初学者有一定的帮助。也许还有很多和我一样从没接触过 UNIX 系统的人，我也尽量把安装过程和命令文的输入方法写得详细。希望任何 WRF 的初学者们都能顺利地看懂这本指南，并能顺利地安装并运行起 WRF 模式。

同时，限于整理者的水平，在本指南中不仅用词十分简陋，而且对许多专用术语也未能正确翻译和使用，希望大家在使用的时候，请以 WRF 主页的 [tutorial](#) 为主，把本指南作为参考来用。同时强烈建议大家再安装运行 WRF 的时候，把自己做过的内容、遇到的错误等信息详细记录下来，不仅有利于以后的复习，也方便错误的查找。不敢期待这本指南能会有多大的用途。但是，我想多一些基础性的教程，就会多一些感兴趣的人，多一些研究者。特别是在论坛上交流讨论的时候，大家就不会再把时间浪费在一些初级问题上，更多的是挖掘它的内涵。当然，这些东西只靠几个人的经验和能力是远远不够的，需要大家的支持。为了方便更多的人学习 WRF，我希望大家能把自己在转 WRF 时的经验和遇到的问题及解决办法介绍出来，整理后和指南一并贴出。如果能得到大家的响应，我想这本指南会帮助更多的人学习 WRF 模式。

根据使用的计算机的软硬件的差别，在编译的过程当中不会一帆风顺；编译通不过的原因也多种多样。特别是在运行的初期阶段，个人的错误操作原因为多。遇到了问题时不要焦急、也不要气馁，在自己寻求答案未果的情况下，多到动力论坛和 wrf forum 里和大家交流交流。

在使用指南的过程中，如果你认为当中有翻译不恰当，用词有错误，或者是有任何意见或建议的话，敬请来信告知 aioply@163.com。谢谢。

在整理本指南的时候，得到了动力论坛（LASG）[『资料与数据处理』](#) 版主 `ustcsunl` 的大力支持；以及动力论坛上的网友 `tzhang`、`tanghao` 和 `穹山` 提供的参考资料；在模式及其相关软件的学习过程当中 `windrisingdl` 的无私的帮助。`namelist.input` 的翻译也是多亏有 `tanghao` 和 `windrisingdl` 的支持和帮助才得以完成大部分内容。当然，`donglipl`、`lepy`、`yuhuaying`、`zhucoffee` 等坛友提供了宝贵的参考意见和建议，特别还有 `light`、`distance4lee` 提供的修改意见，对本指南的最后成形起了很大的作用，在此一并表示感谢。

注：在本指南的第 3 部分和第 4 部分分别记述了 WPS+WRFV2.2 和 WRF_SI+WRFV2.2 的安装运行方法，实际使用时可任选其一（WRF 的开发者们推荐使用前者）。

指南中以绿色书写的部分为 UNIX 的命令文，蓝色为链接部分。

1. <WRF 模式简介>

[Weather Research and Forecasting Model\(WRF\)](#)被誉为是次世代的中尺度天气预报模式。二战后，由于计算机技术的迅猛发展，气象预报技术也随之突飞猛进。短短的几十年里，世界各地的气象研究机关开发出了各自的相对独立的气象模式。这些模式之间缺少互换性，对科研及业务上的交流极其不便。从上世纪 90 年代后半开始，美国对这种乱立的模式状况进行反省。最后由美国环境预测中心（NCEP），美国国家大气研究中心（NCAR）等美国的科研机构为中心开始着手开发一种统一的气象模式。终于于 2000 年开发出了 WRF 模式。同时，为使研究成果能够迅速地应用到现实的天气预报当中去，WRF 模式分为 ARW(the Advanced Research WRF)和 NMM(the Nonhydrostatic Mesoscale Model)两种，即研究用和业务用两种形式，分别由 NCEP 和 NCAR 管理维持着。本指南中使用的是前者 WRF ARW。

WRF 模式为完全可压缩以及非静力模式，采用 F90 语言编写。水平方向采用 Arakawa C(荒川 C)网格点，垂直方向则采用地形跟随质量坐标。WRF 模式在时间积分方面采用三阶或者四阶的 Runge-Kutta 算法。WRF 模式不仅可以用于真实天气的个案模拟，也可以用其包含的模块组作为基本物理过程探讨的理论根据。WRF 的开发组是这样介绍 WRF 模式的特点的：

The WRF model is a fully compressible, nonhydrostatic model (with a hydrostatic option). Its vertical coordinate is a terrain-following hydrostatic pressure coordinate. The grid staggering is the Arakawa C-grid. The model uses the Runge-Kutta 2nd and 3rd order time integration schemes and 2nd to 6th order advection schemes in both horizontal and vertical directions. It uses a time-split small step for acoustic and gravity-wave modes. The dynamics conserves scalar variables.

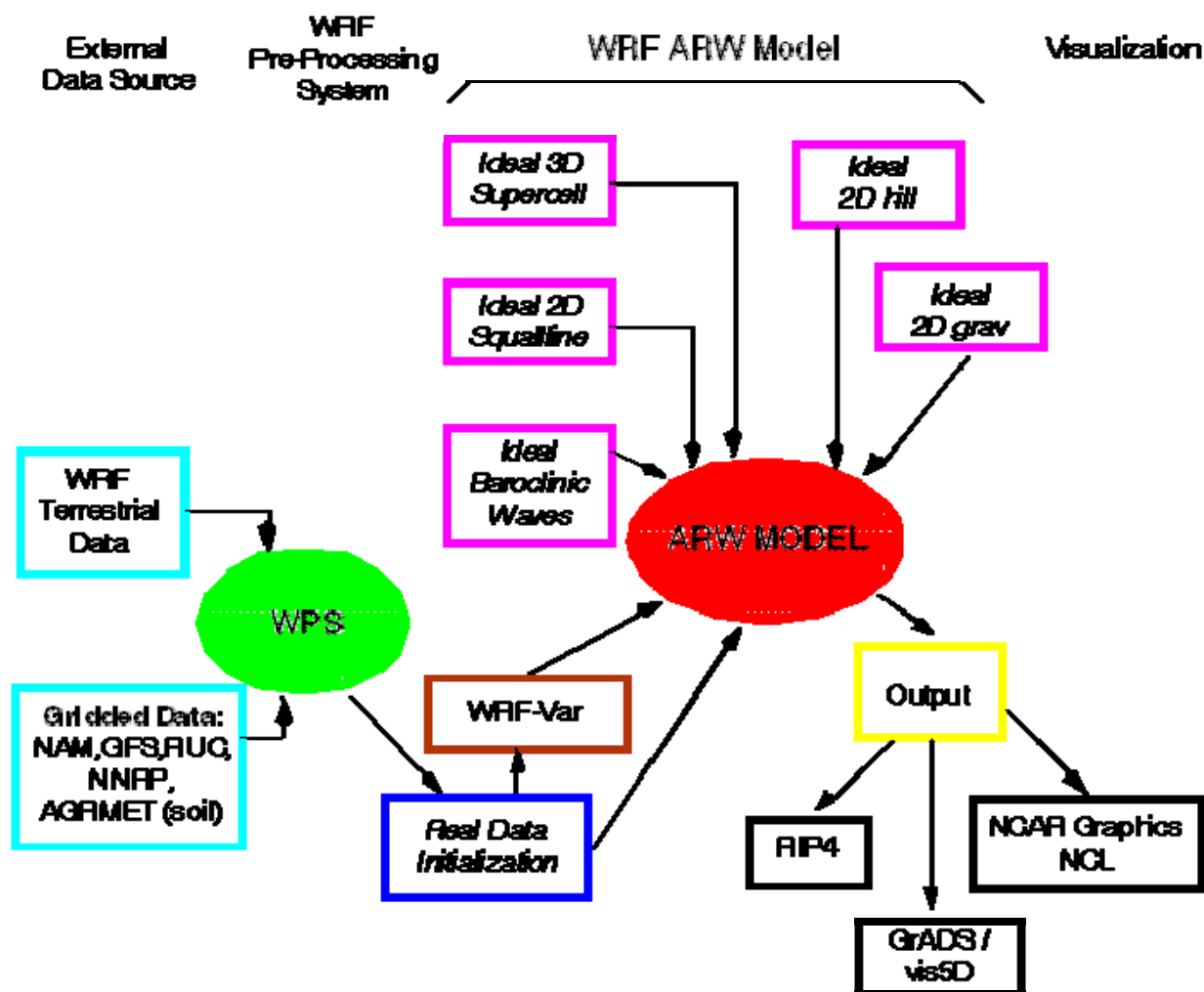
- *Real-data and idealized simulations*
- *Various lateral boundary condition options for real-data and idealized simulations*
- *Full physics options*
- *Non-hydrostatic and hydrostatic (runtime option)*
- *One-way, two-way nesting and moving nest*
- *Three-dimensional analysis nudging*
- *Observation nudging*
- *Applications ranging from meters to thousands of kilometers*

另外模式的输出及其后的分析承接前一代 MM5 的系统，透过 RIP、NCAR Graphic、Vis5D 以及 GRADS 等绘图软件绘制各种气象场。

WRF 的最新版本是 2006 年的圣诞节前 12 月 22 日推出的 Ver2.2。这一版本里，在修补了前一版本的许多错误之上，新增了许多模块。不仅推出了 WRF 的前处理 WRFSI 的进化版 WPS，作为过渡还仍然保留了 WRF 本体和 WRFSI 的衔接。

WRF模式的流程示意图如下:

WRF ARW Modeling System Flow Chart (for WRFV2)



出处: User's Guide for Advanced Research WRF (ARW) Modeling System Version 2.2

2. <准备工作>

注:本文所记述的安装过程仅是在编译器等软件安装完备的条件下进行的。限于笔者的知识水平有限,对这些基本环境的设定不能进行意义描述,十分抱歉。

设定 SSH

要进行正规的模拟气象的运算,就必须要有 Super-Computer。如果没有条件,可直接进入安装部分。不需要连接 Super-Computer 的人可直接进入下一步安装过程。

<首先,把自己的电脑和 supercomputer 进行连接>

- ① 我们使用[SSHSecureShellClient-3.2.9](#) 共享软件。从网上可以下载得到。好像目前只有英语版的。没关系,既然要搞天气预报了,这点简单的英语应该不算难的。



- ② 安装 SSHSecureShellClient。在桌面上会出现两个图标。点击白色的。Edit—> Setting—> ProfileSetting—> connection。在里面填上自己的 Host, User, Port。点击 OK。软件会记住你的 Host, User, Port, 以后每回只需输入密码即可使用。然后点击 Quick connect。会出现一个小窗口。

里面记载了自己的 Host Name, User Name, Port Number, Authentication Method 信息, 在 Authentication Method 处选择 Password, 点击 Connect。然后输入密码, 如果成功的话就应该会和 supercomputer 连接上了。

还有一个黄色的图标 SSH Secure File Transfer Client, 此程序可用于自己当前计算机和 supercomputer 之间的文件传输交换, 支持直接拖拽文件。

第一次和 supercomputer 连接上后, 马上进行密码变更。操作顺序如下:

➤ passwd

Changing password for user user001

Enter login(LDAP) password: (旧密码)

New UNIX password: (新密码)

Retype new UNIX password: (新密码)

LDAP password information changed for user001

Passwd: all authentication tokens updated successfully.

Software requirement

重要！ 首先检查自己是否已经具备了运行 WRF 的工作环境和机器配置条件。

- Fortran 90 or 95 and c compiler
- perl 5.04 or better
- If MPI and OpenMP compilation is desired, it requires MPI or OpenMP libraries
- WRF I/O API supports netCDF, PHD5, GriB 1 and GriB 2 formats, hence one of these libraries needs to be available on the computer where you compile and run WRF

Bash 和 Csh

安装之前，一定要弄清楚自己所使用的计算机的shell。一般输入下面的命令就可以得到答案。

➤ **echo \$SHELL**

如果答案是csh，就请参考[Online tutorial](#)里的命令文的输入方法。

本指南使用的是 bash。如何区别这两种 shell 的写法，请另行找书参考。

安装NetCDF

安装运行 WRF 模式之前，必须要安装 NetCDF。可以通过下面的网址下载。

<http://www.unidata.ucar.edu/software/netcdf/index.html>

使用 gunzip 和 tar 命令对文件进行解压，展开。

➤ **tar -xvf netcdf-3.6.1.tar**

➤ **cd netcdf-3.6.1/src**

➤ **./configure --prefix=/home/user/you/**（安装在指定的目录下）

检查/netcdf-3.6.1/src 目录下的 macros.make 文件。

➤ **vi macros.make**

（在此处，使用 vi 命令。UNIX 的初学者注意：用 vi 命令可以编辑文件，不保存退出时按顺序按下 <Esc> : q ! <shift+1>键，即可退出；若要保存后退出可按 <Esc> : w q 共四键）

注意到 INSTALL 行。如果此行是 **INSTALL = /usr/bin/install -c** 即可；如果不是，要按此修正。

➤ **make check**

➤ **make install** (UNIX 下安装软件的一般步骤: 1. ./configure 2. make 3. make install)

此时，要确认在 prefix 的地方（这回是/home/user/you/）会有 bin, lib, include 和 man 目录生成。

设定 NetCDF 的环境变量：

➤ **NETCDF=/home/user/you;export NETCDF**

2007 年 3 月初，推出了 NetCDF-3.6.2 版本。使用最新版或者前一版都可以安装运行 WRF Model。

同时，为方便使用，我们可以将某些环境变量登录到 .bashrc 里。例如上面的 NetCDF 的环境变量。

注意：慎重修改.bashrc 文件!!!

1) 在/home/user/you/目录下输入

➤ **vi .bashrc**

2) 输入 NetCDF 的环境变量。键入<Esc> : w q 保存退出。

3) 键入如下命令即可定义环境变量：

➤ **source ~/.bashrc**

3. <WPS+WRFV2.2安装运行简介>

请参照网页<http://www.mmm.ucar.edu/wrf/OnLineTutorial/index.htm>, 内有详细的Online Tutorial。本章的内容是参照上述网页的内容进行翻译, 并结合自己在操作过程中遇到的困难进行归纳整理而完成的。其中, 本章和第 4 章的WRFV2.2 部分基本一致 (只有ln处稍有不同), 重复书写难免会有些累赘。但为了保持WPS和WRF_SI各自运行的连贯性, 请读者在使用时注意!

3.0. ~收集数据~

Get Source Code

下载WRF模式的源码。在下载之前要认真阅读[此页](#) 的内容, 并从中可以下载到WRF的Source_Codes。第一次使用要[注册](#); 已经注册过的要[登陆](#)。

把所需要的, 最新的 Source_Codes 收集在一起, 分类放到同一个目录下, 比如:

在/home/user/you/下建立一个名为 Source_Codes_and_Graphics_Software 的源码存放区。例如:

➤ **mkdir Source_Codes_and_Graphics_Software**

并且为以防万一, 把收集到的 Source_Codes 刻成 CD 或者 DVD, 作为备份。

Program Flow

根据自己的研究需要, 下载下列程序:

- 仅需要模拟理想状态问题:

WRF-ARW Model + PostProcessing

- 需要模拟实际问题:

WPS + WRF-ARW Model + PostProcessing

- 模拟加入影响变动值后的实际问题:

WPS + WRF-Var + WRF-ARW Model + PostProcessing

(如果想使用WRF-Var , 还要另外学习其使用方法[WRF-Var Online Tutorial](#))

Documentation

[Users' Guide](#)

User's Guide 里包含了全部的 WRF OnLine Tutorial。并且, User's Guide 是每半年更新一次, 为更好的使用 WRF ARW model, 请使用最新版的。在运行模式之前, 下载一份 User's Guide 作为指导教程。

[WRF ARW Technical Note](#)

PDF 文件。在此 Note 里包含有以下内容:

- ARW model 的方程式, discretization, 初期设定, nesting 的概要
- 模式中利用可能的 Physical Options 的概要
- WRF-Var 的概要

[Bi-Annual Tutorial Presentation](#)

开发人员编写的 PowerPoint, 可以算是 WRF 模式讲解的精华。强烈建议仔细, 反复阅读。在转 WRF 的过

程中，不同时期，怀着不同目的时，会有不同的理解和收获。对初次学习 WRF 模式的人会有很大的帮助。

[WRF-Var](#)

有关 WRF-Var 的解释说明和 WRF-Var OnLine Tutorial 的连接。

[WRFSI](#)

在此页，不仅有 WRFSI 的说明，还可以下载一些必要相关的软件。

- ★ 从WRF的主页（[WRF ARW Users Pages](#)）上可以得到更多的情报和一些很有用的解释说明。

Case Study

在此，以 Hurricane Katrina (August 28, 2005) 为例进行 WRF test run 练习。

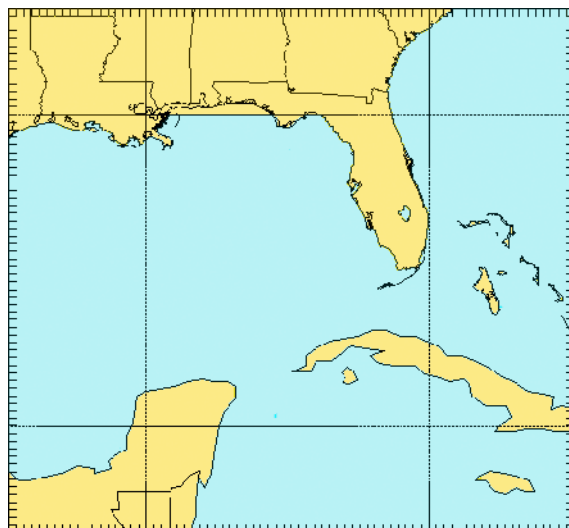
为了 Case Study 和下载练习所需的数据，必须准备足够的硬盘空间。然后建立 working directory 工作域。我自己的是建立在/home/user/you/下的。

例如你可以输入: (输入命令文时严格区分字母大小写)

- **mkdir WRF** (建立目录)
- **cd WRF** (进入目录)
- (返回上一目录的命令是 **cd ..**)

在[global AVN data](#) 处下载所练习用的气象数据。

本次的 case study 的领域如右图所示。



WRF ARW 模式里，主要流程如下图所示：



WPS 是和 WRFV2.2 一同被推出的，可以看成是 WRF_SI 的进化版。

3.1. ～安装前奏～

Get Source Code

下载[WPS](#)和[WRFV2.2](#)的source code。。

建立一个名为WRF的工作目录（如果在[上一页](#)已经建立了的话，此处就没有必要再建立了）。

（比如说我是在/home/user/you/里建立的）。

➤ **mkdir WRF**

Unpack the Code

把下载到的 WPS.TAR.gz 和 WRFV2.TAR.gz 文件复制到新建的 WRF 目录下。然后进行解压和展开工作。因为版本可能会随时被更新，请保持下载最新版学习使用。

复制：

➤ **cp WPS.TAR.gz /home/user/you/WRF/**（可参考 **cp** 命令的使用方法）

➤ **cp WRFV2.TAR.gz /home/user/you/WRF/**

解压：

➤ **gunzip WPS.TAR.gz**

➤ **gunzip WRFV2.TAR.gz** （对文件进行解压）

打开 TAR file：

➤ **tar -xvf WPS.TAR**

➤ **tar -xvf WRFV2.TAR** （不要忘记 **-xvf** ）

于是，会有 WPS/和 WRFV2/目录被做成。

Compile WRFV2.2 First

安装WPS有点麻烦，因为安装WPS时要用到WRFV2.2 里的几个文件库。所以在安装WPS之前还要先安装好WRFV2.2。详细的安装过程请参照本指南的[3.4.节](#)安装WRFV2.2。这里，只为安装WPS记述安装过程的几步命令文。

➤ **cd /home/user/you/WRF/WRFV2**

➤ **./configure**

➤ **1** （如果考虑到了并行计算、嵌套等问题，请选择 **3**。详见后文）

➤ **./compile em_real >& compile.log**

如果在 WRFV2/main/目录下生成了 real.exe 和 wrf.exe 等可执行文件，则表明 WRFV2.2 编译成功。下一步可以进行安装 WPS 了。

在运行WPS时，使用NCARG可以随时把各步骤的设定信息以图像的形式表现出来，方便大家随时进行确认和更改。当然，是否安装NCARG和成功与否，与最后的WRF模式的运行无关。NCARG软件可以作为选择安装项。具体的安装方法请参考[附录 1](#)。

3.2. ～安装 WPS～

Examine the Source Code

进入到/home/user/you/WRF/WPS/目录下

➤ **cd WPS**

检查目录中的内容。

➤ **ls -all** (**ls -a** 亦可)

会有如下内容表示:

Inside this directory, you will find the following files and directories:

```
-rw-r--r-- 1 you user 5074 Sep 15 13:05 README
-rw-r--r-- 1 you user 13 Nov 14 14:49 VERSION
drwxr-xr-x 2 you user 32768 Nov 14 14:48 arch
-rwxr-xr-x 1 you user 1672 Sep 08 18:50 clean
-rwxr-xr-x 1 you user 3349 Sep 12 11:11 compile
-rwxr-xr-x 1 you user 4257 Jul 19 13:47 configure
drwxr-xr-x 5 you user 32768 Nov 14 14:48 geogrid
-rwxr-xr-x 1 you user 1138 Aug 03 10:09 link_grib.csh
drwxr-xr-x 4 you user 32768 Nov 14 14:48 metgrid
-rw-r--r-- 1 you user 1638 Oct 30 11:54 namelist.wps
drwxr-xr-x 7 you user 32768 Nov 14 14:48 test_suite
drwxr-xr-x 4 you user 32768 Nov 14 14:48 ungrib
drwxr-xr-x 3 you user 32768 Nov 14 14:48 util
```

README 文件里有与程序安装和运行有关的有用情报。运行之前最好阅读一次。

WPS 目录下的文件分类:

Source code directories:

geogrid/	Directory containing code to create the static data
metgrid/	Directory containing code to create input to WRFV2
ungrib/	Directory containing code to unpack GRIB data
util/	Directory containing some utilities

Scripts:

clean	Script to clean created files, executables
compile	Script for compiling WPS code
configure	Script to configure the configure.wps file for compile
link_grib.csh	Script to link GRIB files
arch/	Directory where compile options are gathered
namelist.wps	WPS namelist

Configure WPS

键入：

➤ **./configure**

此时，屏幕上会显示出计算机所支持的平台。

```
you@tgg075004:/home/user/you/WRF/WPS> ./configure
Will use NETCDF in dir: /home/user/you/
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...

-----

Please select from among the following supported platforms.

1.  PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, serial, NO GRIB2
2.  PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, serial
3.  PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, DM parallel, NO GRIB2
4.  PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher, DM parallel
5.  PC Linux x86_64 (IA64 and Opteron), PathScale compiler 2.1 or higher, serial, NO GRIB2
6.  PC Linux x86_64 (IA64 and Opteron), PathScale compiler 2.1 or higher, DM parallel, NO GRIB2

Enter selection [1-6] :
```

（由于计算机的差异，有可能会显示出其他的平台选项）

作为练习，我们选择最为基本的（serial）。然后键入：

➤ **1**

如果需要使用 MPI 并行运行的话，就要选择 **3** 或者 **4**。其区别应该可以一目了然吧。

然后会出现以下的对话：

```
-----
Configuration successful. To build the WPS, type: compile
-----
```

这时将会有 configure.wps 文件被做成。如果有必要，对文件 compile options/paths 也要进行编辑。

Compile WPS

然后开始编译模块。

➤ **./compile >& compile.log**

如果编辑成功，会有以下的可执行文件被做成！

```
lrwxrwxrwx 1 you user 23 Mar 9 13:01 geogrid.exe -> geogrid/src/geogrid.exe*
lrwxrwxrwx 1 you user 23 Mar 9 13:02 metgrid.exe -> metgrid/src/metgrid.exe*
lrwxrwxrwx 1 you user 21 Mar 9 13:01 ungrib.exe -> ungrib/src/ungrib.exe*
```

如果编辑失败了，请确认[compile.log](#)里的错误信息。

```
geogrid.exe  Generate static data
metgrid.exe  Generate input data for WRFV2
ungrib.exe   Unpack GRIB data
```

另外，还有一些有用的工具会被链接到 **util** 目录下。

```
lrwxrwxrwx 1 you user 16 Mar  9 17:46 avg_tsfc.exe -> src/avg_tsfc.exe*
lrwxrwxrwx 1 you user 25 Mar  9 17:46 g1print.exe -> ../ungrib/src/g1print.exe*
lrwxrwxrwx 1 you user 25 Mar  9 17:46 g2print.exe -> ../ungrib/src/g2print.exe*
lrwxrwxrwx 1 you user 16 Mar  9 17:46 mod_levs.exe -> src/mod_levs.exe*
lrwxrwxrwx 1 you user 15 Mar  9 17:46 plotfmt.exe -> src/plotfmt.exe*
lrwxrwxrwx 1 you user 17 Mar  9 17:46 plotgrids.exe -> src/plotgrids.exe*
lrwxrwxrwx 1 you user 23 Mar  9 17:46 rd_intermediate.exe -> src/rd_intermediate.exe*
```

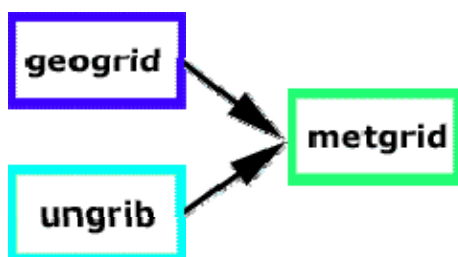
avg_tsfc.exe	Computes daily mean surface temperature from intermediate files. <i>Recommended for using with the 5-layer soil model (sf_surface_physics = 1) in WRF</i>
g1print.exe	List the contents of a GRIB1 file
g2print.exe	List the contents of a GRIB2 file
mod_levs.exe	Remove superfluous levels from 3-d fields in intermediate files
plotfmt.exe	Plot intermediate files (<i>dependent on NCAR Graphics - if you don't have these libraries, plotfmt.exe will not compile correctly</i>)
plotgrids.exe	Generate domain graphics. An excellent tool to configure domains before running geogrid.exe (<i>dependent on NCAR Graphics - if you don't have these libraries, plotgrids.exe will not compile correctly</i>)
rd_intermediate.exe	Read intermediate files

如果还未安装 NCARG 或者安装失败，则不会出现 plotfmt.exe 和 plotgrids.exe 两个可执行文件。这两个文件的有无并不会影响到 WRF 的正常运行。

You are now ready to run the WPS.

3.3. ～运行 WPS～

WPS 用来建立 WRFV2 的输入文件。其流程如下图所示：



geogrid和**ungrib** 属并列关系，运行不分先后。

geogrid 建立“静态的”地面数据。

ungrib 解压GRIB 气象数据，并归纳成一个 [intermediate](#) 文件格式。

metgrid 把气象数据水平插入模式领域内。

metgrid的输出文件将被用作WRFV2.2 的输入文件。

Get Terrestrial Input Data

同样，在 WRF 目录里建立 WPS_GEOG 目录，用来盛放模式用的地表面静态数据。

下载 WPS_GEOG 数据，将模式所需的地表数据解压。这些数据分两类，一类是 general input files；另一类数据是根据解像度分为 10m、5m、2m 和 30s（即大约分别为 110km，55km，20km 和 1km）共 4 种。如果不确定要使用哪一种，可以把这些数据全部保存起来，以后备用。

```
-rw-r--r--  1 you user  190945280  Mar 10 14:10 geog_10m.tar.gz
-rw-r--r--  1 you user  4403015680  Mar 10 14:11 geog_2m.tar.gz
-rw-r--r--  1 you user  4686346240  Mar 10 14:14 geog_30s.tar.gz
-rw-r--r--  1 you user  725934080  Mar 10 14:19 geog_5m.tar.gz
-rw-r--r--  1 you user   76021760  Mar 10 14:31 geog_general.tar.gz
```

在 WRF/WPS_GEOG/目录下：使用 **gunzip** 和 **tar** 命令进行解压。

解压后，会自动生成名为 geog/的目录。在 geog/目录里包含有这 4 种解像度的目录，每个目录里都有介绍数据的 index 文件。

Run geogrid.exe

编辑 [namelist.wps](#) 文件，作以下领域设定的修改。为方便初学者学习，未考虑嵌套问题。

```
max_dom = 1,
e_we = 75,
e_sn = 70,
map_proj = 'mercator',
ref_lat = 25.00,
ref_lon = -85.00,
truelat1 = 0.0,
truelat2 = 0.0,
stand_lon = -85.00,
geog_data_path = '/home/user/you/WRF/WPS_GEOG/geog'
```

我们可以使用 [NCAR Graphic](#) 来查看我们的领域设定。具体方法请参考 WRF 网站的 Tutorial。

如果满意，继续运行 geogrid.exe，生成静态数据。

➤ **./geogrid.exe**

如果使用并行（4CPUs）计算的话，

➤ **mpirun -np 4 geogrid.exe**

在运行过程当中，会出现如下的画面信息：

```
Parsed 11 entries in GEOGRID.TBL
Processing domain 1 of 1
  Processing XLAT and XLONG
  Processing MAPFAC
  Processing F and E
  Processing ROTANG
```

```

Processing LANDUSEF
Calculating landmask from LANDUSEF (WATER = 16)
Processing HGT_M
Processing HGT_U
Processing HGT_V
Processing SOILTEMP
Processing SOILCTOP
Processing SOILCAT
Processing SOILCBOT
Processing ALBEDO12M
Processing GREENFRAC
Processing SNOALB
Processing SLOPECAT
Processing SLOPECAT
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!   Successful completion of geogrid. !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

最后如果出现了"*Successful completion of geogrid*", 即表明运行成功。同时也会出现包含运行过程的 geogrid.log 文件。

出现成功信息后, 确认是否有静态的数据生成:

```
-rw-r--r--  1 you user  2133896 Mar 10 15:01 geo_em.d01.nc
```

这里生成的.nc文件是NetCDF格式的文件。如果安装了[graphical tool](#), 我们还可以将这些数据进行可视化处理。

Run ungrid.exe

整个练习用的数据可以计算 72 小时(网上的 Tutorial 里只进行了最先的 24 小时的计算)。所以 namelist.wps 里的设定如下:

```

start_date = '2005-08-28_00:00:00',
end_date = '2005-08-31_00:00:00',
interval_seconds = 21600

```

[下载](#) 客观分析数据, 并单独放到一个目录。最好保持各种不同的计算数据放到各自的目录里, 例如放到 /home/user/you/WRF/DATA/Katrina/ 目录下。这些数据是 global GFS/AVN GRIB1 data, available every 6 hours, from 2005082800 to 2005083100。然后使用 [gunzip](#) 和 [tar](#) 来解压。

使用 [link_grib.csh](#) 文件把这些数据链接到 WPS 目录下。

例如, 把这些数据解压后放到了 /home/user/you/WRF/DATA/Katrina/ 目录下, 其链接方法如下:

```
➤ ./link_grib.csh /home/user/you/WRF/DATA/Katrina/avn_050828
```

然后, 我们会看到 WPS 目录里多了以下文件:


```
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAA -> /home/user/you/WRF/DATA/avn_050828_00_00
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAB -> /home/user/you/WRF/DATA/avn_050828_00_06
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAC -> /home/user/you/WRF/DATA/avn_050828_00_12
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAD -> /home/user/you/WRF/DATA/avn_050828_00_18
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAE -> /home/user/you/WRF/DATA/avn_050828_00_24
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAF -> /home/user/you/WRF/DATA/avn_050828_00_30
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAG -> /home/user/you/WRF/DATA/avn_050828_00_36
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAH -> /home/user/you/WRF/DATA/avn_050828_00_49
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAI -> /home/user/you/WRF/DATA/avn_050828_00_48
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAJ -> /home/user/you/WRF/DATA/avn_050828_00_54
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAK -> /home/user/you/WRF/DATA/avn_050828_00_60
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAL -> /home/user/you/WRF/DATA/avn_050828_00_66
lrwxrwxrwx 1 you user 49 Mar 12 13:02 GRIBFILE.AAM -> /home/user/you/WRF/DATA/avn_050828_00_72
```

把 **Vtable** 链接到当前目录 (WRF/WPS/) 下。因为我们使用的数据是 *GFS/AVN data*, 所以使用 *GFS Vtable*。连接方法如下:

➤ **ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable**

结果是生成下面的文件:

```
lrwxrwxrwx 1 you user 33 Mar 12 18:14 Vtable -> ungrib/Variable_Tables/Vtable.GFS
```

运行 **ungrib.exe**, 并保留 [log 文件](#)。

➤ **./ungrib.exe >& ungrib.log**

最好不要在这里使用并行。

最后屏幕里会出现下面的信息表示运行成功:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Successful completion of ungrib. !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

这时目录里会出现以下几个文件:

```
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-28_00
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-28_06
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-28_12
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-28_18
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-29_00
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-29_06
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-29_12
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-29_18
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-30_00
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-30_06
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-30_12
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-30_18
-rw-r--r-- 1 you user 38869928 Mar 13 15:20 FILE:2005-08-31_00
```

The [log file](#) created during this step is very important. It contains information regarding the fields which were found in the input file and on which level these fields are available.

在这里, 我们也同样可以使用 [NCAR Graphic](#) 来查看我们的领域设定。具体方法请参考 WRF 网站的 [Tutorial](#)。

Run metgrid.exe

最后一部最简单，只要输入下面的命令即可：

➤ **./metgrid.exe**

如果使用并行（4CPUs）计算的话，

➤ **mpirun -np 4 metgrid.exe**

运行结束后屏幕出现这样的信息：

```
Processing domain 1 of 1.
Processing 2005-08-28_00:00:00
./FILE
Processing 2005-08-28_06:00:00
./FILE
...（此处省略若干行）
Processing 2005-08-30_18:00:00
./FILE
Processing 2005-08-31_00:00:00
./FILE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Successful completion of metgrid. !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

同时在目录里还会生成下面的几个 NetCDF 文件：

```
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-28_00:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-28_06:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-28_12:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-28_18:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:28 met_em.d01.2005-08-29_00:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-29_06:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-29_12:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-29_18:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:28 met_em.d01.2005-08-30_00:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-30_06:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-30_12:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:27 met_em.d01.2005-08-30_18:00:00.nc
-rw-r--r-- 1 you user 5932836 Mar 13 15:28 met_em.d01.2005-08-31_00:00:00.nc
```

这些文件都是 NetCDF 格式，可以使用 ncdump 工具来查看。

➤ **ncdump -h met_em.d01.2005-08-28_00:00:00.nc**

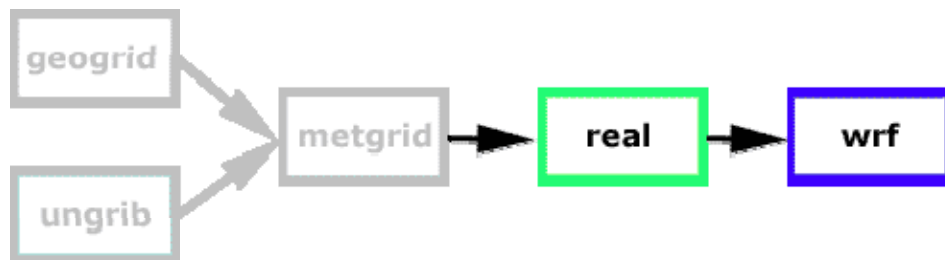
我们还可以查以图片的方式来查看这些数据。

恭喜你，到这里你已经成功了一半，下一步要运行 WRFV2.2 的本体计算部分了。

You are now ready to run the WRFV2.

3.4. ～安装 WRFV2.2～

这里，如果是运行 real case，流程示意图如下：



如果是运行 ideal case，最后部分的流程示意图如下：



ideal case 和 real case 的运行方法大同小异，在本指南里不对 ideal case 的安装运行进行详细描述。

安装之前要仔细设计好自己要做的实验，选好实验 case 和运行条件。例如是 real 还是 ideal；是 single 还是 palalell；是否需要 nesting 等等。因为每修改一次运行条件都要重新进行编译，这样就会浪费掉许多时间。

Examine the Source Code

在 WRF/目录下...

➤ **cd WRFV2**

检查目录中的内容。

➤ **ls -all** (**ls -a** 亦可)

会有如下内容表示：

```

-rw-r--r--  1 you user 16421 Oct 21  2005 Makefile
-rw-r--r--  1 you user 7250 Nov  9  2005 README
-rw-r--r--  1 you user  7238 Oct  5  2005 README.NMM
-rw-r--r--  1 you user  2548 May 18  2004 README_test_cases
drwxr-xr-x  2 you user  4096 Apr 11 16:00 Registry
drwxr-xr-x  2 you user  4096 Nov  9  2005 arch
drwxr-xr-x  2 you user  4096 Nov  9  2005 chem
-rwxr-xr-x  1 you user  1078 May 24  2005 clean
-rwxr-xr-x  1 you user  6913 Oct 21  2005 compile
-rwxr-xr-x  1 you user 10636 May  7  2005 configure
drwxr-xr-x  2 you user  4096 Apr 20 20:01 dyn_em
drwxr-xr-x  2 you user  4096 Nov  9  2005 dyn_exp
drwxr-xr-x  2 you user  4096 Nov  9  2005 dyn_nmm
drwxr-xr-x 12 you user  4096 Nov  9  2005 external
drwxr-xr-x  2 you user  4096 Apr 11 16:02 frame
  
```

```

drwxr-xr-x  2 you user  4096 Mar  14 16:44 inc
drwxr-xr-x  2 you user  4096 Mar  14 16:44 main
drwxr-xr-x  2 you user  4096 Apr 11 16:13 phys
drwxr-xr-x  2 you user  4096 Mar  14 16:44 run
drwxr-xr-x  2 you user  8192 Mar  14 16:44 share
drwxr-xr-x 11 you user  4096 Nov  9  2005 test
drwxr-xr-x  3 you user  4096 Apr 11 16:01 tools

```

README 文件里有与程序安装和运行有关的有用情报。运行之前最好阅读一次。

WRFV2 目录下的文件分类：

Source code directories:

```

chem/      Directory containing modules for chemistry (not currently supported)
dyn_em/    Directory containing modules for dynamics in WRF ARW core
dyn_nmm/   Directory containing modules for dynamics in WRF NMM core
dyn_exp/   Directory for a 'toy' dynamical core
external/  Directory containing external packages,  such as those for IO,  time keeping and MPI
frame/     Directory containing modules for WRF framework
inc/       Directory containing include files
main/      Directory for main routines,  such as wrf.F,  and all executables after install
phys/      Directory for all physics modules
share/     Directory containing mostly modules for WRF mediation layer and WRF I/O
tools/     Directory containing tools

```

Scripts:

```

clean      Script to clean created files,  executables
compile    Script for compiling WRF code
configure  Script to configure the configure.wrf file for compile
Makefile   Top-level makefile
Registry/  Directory for WRF Registry file
arch/      Directory where compile options are gathered
run/       Directory where one may run WRF
test/      Directory that contains 7 test case directories,  may be used to run WRF

```

Environment Variable-NetCDF

对 NetCDF 环境变量进行定义：

```
> NETCDF=/home/user/you;export NETCDF
```

Configure WRFV2

键入：

```
> ./configure
```

在此，会显示出计算机所支持的平台。

```

checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home/user/you/netcdf-3.6.1
PHDF5 not set in environment. Will configure WRF for use without.
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...

-----

Please select from among the following supported platforms.

1.  PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher  (Single-threaded, no nesting)
2.  PC Linux x86_64 (IA64 and Opteron), PGI 5.2 or higher, DM-Parallel  (RSL, MPICH, Allows
nesting)
3.  PC Linux x86_64 (IA64 and Opteron), PGI 5.2 or higher DM-Parallel  (RSL_LITE, MPICH,
Allows nesting, No periodic LBCs)
4.  PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher  (Single-threaded, RSL, Allows
nesting)
5.  AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (single-threaded, no nesting)
6.  AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler  (single threaded, allows nesting
using RSL without MPI)
7.  AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (OpenMP)
8.  AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler SM-Parallel (OpenMP, allows
nesting using RSL without MPI)
9.  AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+icc compiler  DM-Parallel (RSL, MPICH,
allows nesting)
10. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+gcc compiler  DM-Parallel (RSL, MPICH,
allows nesting)
11.  PC Linux x86_64 (IA64 and Opteron), PathScale 2.1 or higher (Single-threaded, no nesting)
12.  PC Linux x86_64 (IA64 and Opteron), PathScale 2.1 or higher DM-Parallel  (RSL_LITE,
PathScale MPICH, No periodic LBCs)
13.  Cray XT3 Catamount/Linux x86_64 (Opteron), PGI 5.2 or higher DM-Parallel (RSL_LITE,
MPICH, Allows nesting, Periodic in X only)

Enter selection [1-13] :

```

（根据计算机的差异，也许会显示出更多的平台选项）

作为练习，我们选择最为基本的（single-threaded, no nesting）。然后键入：

➤ 1

注意：如果需要进行并行计算或者嵌套，要注意括号里的说明项。例如，可以先选择 **3**。在此步骤，将有 configure.wrf 文件被做成。如果有必要，对此文件的 compile options/paths 也要进行编辑。

Compile WRFV2

➤ ./compile

然后屏幕出现：

```
Usage:
  compile wrf           compile wrf in run dir (NOTE: no real.exe, ndown.exe, or ideal.exe generated)
  or choose a test case (see README_test_cases for details) :
    compile em_b_wave
    compile em_esmf_exp
    compile em_grav2d_x
    compile em_hill2d_x
    compile em_quarter_ss
    compile em_real
    compile em_squall2d_x
    compile em_squall2d_y
    compile exp_real
    compile nmm_real
  compile -h           help message
```

因为要运行 WRF ARW real data case，所以选择 em_real。

然后开始编译模块。

➤ ./compile em_real >& compile.log

运行会稍微花一点时间。（如果在./configure 处选择了 3，你会有将近 30 分钟左右的时间来喝杯咖啡）

如果在进行过程当中自己弄出错了，或者要更改 configure、compile 等的选项，可以用“clean -a”的命令来清理所有的 built files, including configure.wrf。

➤ ./clean -a

然后开始重新编译 WRFV2.2。

编译结束后，打开[compile.log](#)确认是否有错误信息存在。如果编辑成功，在main/目录下会有以下的文件被做成！

```
-rwxr-xr-x  1 you user  13612035 Mar 14 15:11 ndown.exe
-rwxr-xr-x  1 you user  13237353 Mar 14 15:11 nup.exe
-rwxr-xr-x  1 you user  10909533 Mar 14 15:11 real.exe
-rwxr-xr-x  1 you user  13237353 Mar 14 15:11 wrf.exe
```

ndown.exe 用于 one-way nesting 。

nup.exe 用在 WRF_3DVAR。

real.exe 用于 real data cases 的初期化

wrf.exe 用于 WRF model integration

这些文件被从 main/目录链接到了 run/, test/em_real/目录下。

（如果运行其它 idealized cases，方法一样。在./compile em_real >& compile.log 处选择相应的文件选项。不过之后会生成 ideal.exe 可执行文件）

3.5. ~运行 WRFV2.2~

Edit namelist.input

到 run/ 或 test/em_real 目录下运行 WRFV2.2。在这里，我选择后者。

➤ **cd test/em_real**

（如果在 em_real 下不能正常运行 real.exe 的话，可以到 run 目录下试试）

然后键入 **ls -l** 后，目录里包含有下面这些文件：

```
lrwxrwxrwx 1 you user 22 Mar 14 17:49 CAM_ABS_DATA -> ../../run/CAM_ABS_DATA
lrwxrwxrwx 1 you user 25 Mar 14 17:49 CAM_AEROPT_DATA -> ../../run/CAM_AEROPT_DATA
lrwxrwxrwx 1 you user 23 Mar 14 17:49 ETAMPNEW_DATA -> ../../run/ETAMPNEW_DATA
lrwxrwxrwx 1 you user 21 Mar 14 17:49 GENPARM.TBL -> ../../run/GENPARM.TBL
lrwxrwxrwx 1 you user 21 Mar 14 17:49 LANDUSE.TBL -> ../../run/LANDUSE.TBL
lrwxrwxrwx 1 you user 25 Mar 14 17:49 README.namelist -> ../../run/README.namelist
-rw-r--r-- 1 you user 5166 Dec 20 10:25 README.obs_fdda
lrwxrwxrwx 1 you user 19 Mar 14 17:49 RRTM_DATA -> ../../run/RRTM_DATA
lrwxrwxrwx 1 you user 22 Mar 14 17:49 SOILPARM.TBL -> ../../run/SOILPARM.TBL
lrwxrwxrwx 1 you user 21 Mar 14 17:49 VEGPARM.TBL -> ../../run/VEGPARM.TBL
lrwxrwxrwx 1 you user 22 Mar 14 17:49 grib2map.tbl -> ../../run/grib2map.tbl
lrwxrwxrwx 1 you user 21 Mar 14 17:49 gribmap.txt -> ../../run/gribmap.txt
-rw-r--r-- 1 you user 1762 Feb 19 2005 landFileNames
-rwxr-xr-x 1 you user 4663 Dec 19 04:13 namelist.input
-rw-r--r-- 1 you user 5647 Dec 15 06:28 namelist.input.chem
-rwxr-xr-x 1 you user 5732 Dec 19 04:13 namelist.input.grid_fdda
-rwxr-xr-x 1 you user 5902 Dec 19 04:13 namelist.input.jan00
-rwxr-xr-x 1 you user 5901 Dec 15 06:28 namelist.input.jun01
-rwxr-xr-x 1 you user 5655 Dec 19 04:13 namelist.input.obs_fdda
-rwxr-xr-x 1 you user 4804 Dec 19 04:13 namelist.input.si
-rwxr-xr-x 1 you user 6108 Dec 19 04:13 namelist.input.wps
lrwxrwxrwx 1 you user 20 Mar 14 17:49 ndown.exe -> ../../main/ndown.exe
lrwxrwxrwx 1 you user 18 Mar 14 17:49 nup.exe -> ../../main/nup.exe
lrwxrwxrwx 1 you user 25 Mar 14 17:49 ozone.formatted -> ../../run/ozone.formatted
lrwxrwxrwx 1 you user 29 Mar 14 17:49 ozone_lat.formatted -> ../../run/ozone_lat.formatted
lrwxrwxrwx 1 you user 30 Mar 14 17:49 ozone_plev.formatted -> ../../run/ozone_plev.formatted
lrwxrwxrwx 1 you user 19 Mar 14 17:49 real.exe -> ../../main/real.exe
-rw-r--r-- 1 you user 10240 Dec 9 16:40 run_1way.tar
-rw-r--r-- 1 you user 20480 Jan 20 2006 run_2way.tar
-rw-r--r-- 1 you user 10240 Jan 20 2006 run_restart.tar
lrwxrwxrwx 1 you user 17 Mar 14 17:49 tr49t67 -> ../../run/tr49t67
lrwxrwxrwx 1 you user 17 Mar 14 17:49 tr49t85 -> ../../run/tr49t85
lrwxrwxrwx 1 you user 17 Mar 14 17:49 tr67t85 -> ../../run/tr67t85
```



```
lrwxrwxrwx 1 you user 25 Mar 14 17:49 urban_param.tbl -> ../../run/urban_param.tbl
lrwxrwxrwx 1 you user 18 Mar 14 17:49 wrf.exe -> ../../main/wrf.exe
```

新生成的文件的用途和说明，如下表归纳所示：

ETAMPNEW_DATA	Variables for the Ferrier (new Eta) microphysics scheme, (binary file).
CAM_ABS_DATA CAM_AEROPT_DATA ozone.formatted ozone_lat.formatted ozone_plev.formatted	CAM radiation variables (binary files). Details available in phys/module_ra_cam.F
RRTM_DATA	Data for the 16 longwave spectral bands used in RRTM (binary file). Details available in phys/module_ra_rrtm.F
tr49t67 tr49t85 tr67t85	Data for the GFDL radiation scheme (binary files). Details available in phys/module_ra_gfdleta.F
LANDUSE.TBL	Land surface parameters required by the LSM models. The file is an ASCII table. Details available in <i>phys/module_physics_init.F</i>
GENPARM.TBL SOILPARM.TBL VEGPARM.TBL urban_param.tbl	Soil and vegetation parameters required by the Noah LSM model. The files are ASCII tables. Details available in phys/module_sf_noahlsn.F and phys/module_sf_urban.F
grib2map.tbl gribmap.txt	Tables required for creating GRIB1 and GRIB2 output.
namelist.input namelist.input.jan00 namelist.input.jun01 namelist.input.wps namelist.input.SI	Sample namelists to set up and run real.exe and wrf.exe namelist.input will always be used by real.exe and wrf.exe .jan00 and .jun01 have been set up for our default test cases .wps and .SI are samples of different setups required depending on the preprocessor used. Details of all the namelist options are available in the README.namelist file.

针对本次的OnLine Tutorial case，编辑[namelist.input](#)。归纳一下，主要修改以下几点：

Start date: 2005082800

End date: 2005083100

Interval: 21600 sec (6 hours)

Grid points in EW direction: 75

Grid points in SN direction: 70

Number of vertical levels: 31

Grid distance: 30km

namelist.input 文件被用在了 real.exe 和 wrf.exe 可执行文件的运行等处。

Link

把在WPS处做成的[met_em.d01_0508* files](#)链接到WRFV2/run/ 或者WRFV2/test/em_real/ 目录下。

➤ `ln -sf /home/user/you/WRF/WPS/met_em.d01.2005-08* .`

使用方法可以参考 `ln` 命令

Run real.exe

进行 single processor 运行。例如本次的演习。

➤ `./real.exe`

如果是 DM(distributed memory) parallel systems, 就会需要 `mpirun` command。例如, 如果是 Linux cluster, 实行 4CPUs 的 MPI code 的命令就是:

➤ `mpirun -np 4 real.exe`

可以根据自己的条件选择 CPU 的个数。同时不要忘记, 在 `./configure` 处要选择与之相对应的平台。

在键入 `./real.exe` 后, 最好不要打扰计算机的运行。如果运行成功的话, 最后会有 `wrfbdy_d01` 和 `wrfinput_d01` 文件生成。

Check Your rsl Files

如果进行了并行计算, 运行完 `real.exe` 或者 `wrf.exe` 之后, 要检查 `rsl.out.*` 和 `rsl.error.*` (对于每个 processor, 都会出现 `rsl.out` 和 `rsl.error`)。在 `rsl.out.0000` 和 `rsl.error.0000` 里包含了重要的情报。如果运行失败, error message 有可能存在于这些文件当中的某一个文件里。同时, 在 `namelist.input` 里有 `debug_level` 一项。这一项对应的有效数值为 0, 50, 100, 200, 300 等, 数值越大, 在 `rsl.out.*` 和 `rsl.error.*` 里输出的信息就越详细, 越有利于寻找错误的根源。如果多次运行都不能成功, 请考虑使用此项。

使用 `head -n 30 rsl.out.*` 和 `tail -n 30 rsl.out.*` 命令来查看 `rsl` 的文件头 30 行和文件尾 30 行。

如果运行 `real.exe` 成功, 则在 `rsl.out.0000` 的最后会有提示:

SUCCESS COMPLETE REAL_EM INIT

因为 `wrf.exe` 也会生成同名的 log 文件, 所以把上面的 `rsl.out.*` 和 `rsl.error.*` 移动到其他目录, 或者删掉。

Run wrf.exe

➤ `./wrf.exe`

如果是 single processor machine, 输入 `./wrf.exe`。

如果是 DM(distributed memory) parallel systems, 会有需要 `mpirun` command 的场合。例如, 如果是 Linux cluster, 实行 4CPUs 的 MPI code 的命令就是:

➤ `mpirun -np 4 wrf.exe`

如果全部进行顺利的话, 新的 `wrfout_d01` file 将会被做成。

从 2005082800 到 2005083100 的 72 小时份的文件将被做成。用 `ncdump` 可以查看 `wrfout_d01` file 里的信息。

➤ `ncdump -h wrfout_d01_2005-08-28_00:00:00` [参照](#)

wrf: SUCCESS COMPLETE WRF

4. <WRF_SI+WRFV2.2 的安装运行>

请参照网页<http://www.mmm.ucar.edu/wrf/OnLineTutorial/WRFSL/>，内有详细的Online Tutorial。

本章的内容是参照上述网页的内容进行翻译，并结合自己在操作过程中遇到的困难进行归纳整理完成的。

4.0. ～收集数据～

Get Source Code

和WPS+为WRFV2.2一样，把所需要的，最新的Source_Codes收集在一起，分类放到同一个目录下。

下载WRF_SI和WRFV2.2的source code。

在/home/user/you/目录下建立一个名为 Source_Codes_and_Graphics_Software 的源码存放区。例如：

➤ **mkdir Source_Codes_and_Graphics_Software**

并且为以防万一，把收集到的 Source_Codes 刻成 CD 或者 DVD，作为备份。

Unpack the Code

在/home/user/you/目录下建立一个名为 WRF 的工作目录：

➤ **mkdir WRF**

把下载到的 wrfsi_v2.1.2.tar.gz 和 WRFV2.TAR.gz 文件复制到新建的 WRF 目录下。然后解压和展开。

➤ **cp wrfsi_v2.1.2.tar.gz /home/user/you/WRF/** （可参考 cp 命令的使用方法）

➤ **cp WRFV2.TAR.gz /home/user/you/WRF/**

解压：

➤ **gunzip wrfsi_v2.1.2.tar.gz** （对文件进行解压）

➤ **gunzip WRFV2.TAR.gz**

打开 TAR file。

➤ **tar -xvf wrfsi_v2.1.2.tar** （不要忘记 -xvf ）

➤ **tar -xvf WRFV2.TAR**

于是，会有 wrfsi/和 WRFV2/目录被做成。

4.1. ～安装 WRF_SI～

4.1.1. ～定义环境变量～

Examine the Source Code

在/home/user/you/WRF/目录下

➤ **cd wrfsi**

检查目录中的内容。

➤ **ls -all** （ls -a 亦可）

会有如下内容表示：

```
total 156
drwxr-xr-x  9 you user  4096  Mar 10 08:08  .
drwxr-xr-x  3 you user  4096  May 25 19:04  ..
```

```

-rw-r--r--    1 you user  15975  Jun 19  2004  CHANGES
-rw-r--r--    1 you user  11166  Mar 12  2003  HOW_TO_RUN.txt
-rw-r--r--    1 you user   4405  Jan 18  2003  INSTALL
-rw-r--r--    1 you user   4101  Dec 11  2003  Makefile
-rw-r--r--    1 you user  30421  Jan 28  2005  README
-rw-r--r--    1 you user  12661  Mar  8  2005  README.wrf_si_nl
drwxr-xr-x    7 you user   4096  Aug 23  2005  data
drwxr-xr-x    2 you user   4096  Mar  6 07:18  etc
drwxr-xr-x    7 you user   4096  Aug  3  2005  extdata
drwxr-xr-x    3 you user   4096  Aug  3  2005  graphics
drwxr-xr-x    5 you user   4096  Mar 10 08:11  gui
-rwxr-xr-x    1 you user  27153  Mar 10 08:08  install_wrf_si.pl
drwxr-xr-x   12 you user   4096  Aug  3  2005  src
drwxr-xr-x    3 you user   4096  Aug 23  2005  util

```

在 CHANGES 文件里写入了最近更新的全部 code。

HOW_TO_RUN.txt · INSTALL · README · README.wrf_si_nl 文件里有与程序安装和运行有关的有用情报。运行之前最好阅读一次。install_wrf_si.pl 是用来编译 WRF_SI 或安装 perl script 用的。

Set Environment Variables

安装和运行 WRF_SI 之前要先记述环境变量。

无论是定义环境变量，还是使用初期值，在/home/user/you/WRF/wrf_si/中的 config_paths 文件中都可以找得到。可以用 cat 和 vi 命令：**cat config_paths** 进行阅览，**vi config_paths** 进行修改。

正确定义环境变量，对正确的安装和运行 WRF_SI 程序来说很重要。

注意：在这里没有对包含有 Input GriB 数据文件的文件夹进行环境变量的定义。这个定义将在使用 [namelist file that the grib_prep](#) 时输入。

NetCDF

安装 WRFV2 之前，要对 NetCDF 的环境变量进行定义。

➤ **NETCDF=/home/user/you;export NETCDF**

SOURCE_ROOT

定义 source root directory 路径。

➤ **SOURCE_ROOT=/home/user/you/WRF/wrf_si;export SOURCE_ROOT**

INSTALLROOT

定义 install root directory 路径。

➤ **INSTALLROOT=/home/user/you/WRF/wrf_si;export INSTALLROOT**

EXR_DATAROOT

定义 degribed （媒体）的 directory 路径。

➤ **EXT_DATAROOT=/home/user/you/WRF/wrfsi/extdata;export EXT_DATAROOT**

TEMPLATES

放置 templates directory 的地方。这个目录里包含有为修正自己构筑的 case 时所需要的 SI namelist 文件。仅和运行 SI（不使用 GUI）时相关联。

➤ **TEMPLATES=/home/user/you/WRF/wrfsi/templates;export TEMPLATES**

DATAROOT & MOAD_DATAROOT

DARAROOT 可以把含有复数的 sundirectory（MOAD_DATAROOT）放置在最上部的 directory。如果那里仅有一个 MOAD_DATAROOT 时，MOAD_DATAROOT 可以和 DATAROOT directory 相同。

➤ **DATAROOT=/home/user/you/WRF/wrfsi/domains;export DATAROOT**

但是，不可设定为 wrfsi/data directory。

MOAD_DATAROOT 是用来运行个别案例的目录。对于此环境变量，并没有设定其初期值。通常是如下定义：（现在并不需要输入下面的语句，仅作参考用。对此环境变量的指定还会在后面进行描述。）

➤ **MOAD_DATAROOT=/home/user/you/WRF/wrfsi/domains/case-name;export MOAD_DATAROOT**

GEOG_DATAROOT

此环境变量可以指定自己的 terrestrial input data（terrain, landuse, etc.）的目录。

总之，在此处（即为 GEOG），必须事先放入最初下载的 [Source Codes](#)。一共 21 个 gz 文件。这些数据可以从 [WRF SI](#) 处下载得到。把数据放到 /WRF/wrfsi/extdata/GEOG 内，然后用前面用过的 **gunzip** 和 **tar** 命令逐个进行解冻。定义：

➤ **GEOG_DATAROOT=/home/user/you/WRF/wrfsi/extdata/GEOG;export GEOG_DATAROOT**

NCL_COMMAND

如果要使用 GUI，我们还要进行环境变量的设定。使用 GUI，可以制作出 terrestrial input data 的画像。

➤ **NCL_COMMAND=/home/user/you/ncl;export NCL_COMMAND**

为方便使用，我们可以将这些环境变量也登录到 **.bashrc** 里。

1) 在 /home/user/you/ 目录下输入

➤ **vi .bashrc**

2) 输入上面除了 **MOAD_DATAROOT** 以外的 8 个环境变量。其中，NetCDF 在上面已经登录过。然后键入 **<Esc>:wq** 保存退出。

3) 键入下面命令即可定义环境变量：

➤ **source ~/.bashrc**

4.1.2. ~安装 WRF_SI~

Install WRF_SI

➤ `perl install_wrfsi.pl`

在画面上会出现下面的对话：

```
Routine: Install_wrfsi
Path to perl: /usr/bin/perl
--path_to_netcdf not specified, attempting to determine...
netCDF path found from environment variable.
Do you want to install the WRF SI graphical user interface? [y|n]:
```

在此处，如果选择“n”的话，只会安装 WRF_SI。选择“y”，会一起安装 GUI。

我们在此选择“y”。（例：如下表示。[“n”](#)，[“y”](#)）

➤ `y`

如果安装成功，在下面的目录里会有下面几个可执行文件生成。

➤ `cd /home/user/you/WRF/wrfsi/bin`

➤ `ls -l`

```
-rwxr-xr-x 1 you user 870534 Jul 12 16:01 grib_prep.exe
-rwxr-xr-x 1 you user 1901081 Jul 12 16:02 gridgen_model.exe
-rwxr-xr-x 1 you user 1760763 Jul 12 16:01 hinterp.exe
-rwxr-xr-x 1 you user 1647080 Jul 12 16:02 siscan
-rwxr-xr-x 1 you user 1754324 Jul 12 16:02 staticpost.exe
-rwxr-xr-x 1 you user 2018603 Jul 12 16:01 vinterp.exe
```

还有，检查一下在下面的目录里是否有 makefile 生成。

➤ `cd /home/user/you/WRF/wrfsi/src/include`

makefile_alpha.inc.in

* 如果install 失败了… … 在ARW OnLine Tutorial里登载了一些[error解决办法](#)，请参考使用。

4.1.3. ~使用 WRF_SI 的 GUI~

Run WRF_SI with GUI

如果想运行GUI，会使用到X-window软件。我使用的是名为EXODUS的软件，需要通过SSH软件和超算进行连接。再有，这里的记述的方法也许仅适用于我所使用的超算，仅供参考。[WRF_SI web site](#)上的方法自己没有研究过，在这里不能详细写出，请原谅。在输入下面语句之前要先打开X-window。

- `ssh -Y login.wrftest.ac.cn` （自己的 **host name**）
- `export DISPLAY=192.168.111.111:0.0` （自己的 PC 的 **IP 地址**）
- `LANG=C`
- `export LANG`
- `qmon &`

（到这里是连接 UNIX 系统和 X-window 的命令语句，在使用画图软件 GrADS 时也会用到）

然后打开 WRF_SI 的 GUI 系统:

```
> cd /home/user/you/WRF/wrfsti
> ./wrf_tools
```

在画面上会有GUI显现。详细信息请参照[WRF SI web site](#)。

(以上的操作方法有可能会和其它软件不同。例如, x-win, x-winPro 等软件就不需要什么特殊的设定。跳过 4.1.3 节, 仍可继续运行 WRFV2。)

4.2. ~运行 WRF_SI~

WRF_SI 里主要有三个步骤, 并且每一步都需要对相应的 namelist 文件进行编辑:

STEP1: 设定领域	——>	编辑 wrfsti.nl 的 1、2 部分
STEP2: 指定气象数据	——>	编辑 grib_prep.nl 文件
STEP3: 向领域内插入气象数据	——>	编辑 wrfsti.nl 的 3、4 部分

Run WRF_SI

STEP1: Localize model domain and create static files

在此处设定对象领域, 建立为运行 WRF 所必须的静态目录。到此阶段, 各模式领域的设定仅一次即可。

Localization 所必需的 script: /home/user/you/WRF/wrfsti/etc/window_domain_rt.pl

Namelist: wrfsti.nl

此处把存放模拟用的数据的目录命名为 OnlineTut 。在/WRF/wrfsti/下:

```
> cd domains
> mkdir OnlineTut
```

设定环境变量 (在上面[Set Environment Variables](#)处描述过)

```
> MOAD_DATAROOT=/home/user/you/WRF/wrfsti/domains/OnlineTut;export MOAD_DATAROOT
```

做成 TEMPLATE。

```
> cd /home/user/you/WRF/wrfsti/templates
> cp -r default OnlineTut
> chmod -R u+w OnlineTut
> cd OnlineTut
```

Edit wrfsti.nl

wrfsti.nl 文件被用于 step1 (localization) 和 step3 (interpolation of data)。在[wrfsti.nl](#)网页中: 红色部分与 step1, 紫色部分与 step3 相关联。关于 namelist 中变量的详细内容请参照[README.wrfsti.nl](#), [附录 2](#)的中文版也许也会有很大帮助。

现在, 我们开始编译 wrfsti.nl 文件。建议两部分一起编辑。

进入 /home/user/you/WRF/wrfsti/templates/ 下的 OnlineTut 目录。使用 vi 命令对 wrfsti.nl 进行编辑。(<Esc> 变换输入形式; x 键可删除字符; i 或 a 键可插入字符)

1. 通过编辑 [hgridspec](#) 来设定所使用的模式。我们将模拟 Hurricane Katrina。

2. 在此, 关键是要核对通向 terrestrial data 的 path 及其它的环境变量是否被正确设定以这两个方面内容。还要检

查确认 [sfcfiles](#) 的内容。

3. [interp_control](#) 部分，设定内插 data。对于本次的模拟练习，要检查确认 INIT_ROOT 和 LBC_ROOT 是否被正确定义。使用 AVN data 时，这些变量都需要被设定。如果使用不同的模式，变更 sigma level 时，在此步骤下还要变更 LEVELS。对此次的数值模拟，我们使用的初期值为 31 levels。

4. 检查 [si_path](#) 是否被定义在 intermediate files (/home/user/you/WRF/wrfsi/extdata/) 处。

通过对模式构成的编辑，调整模式领域的适用准备。

➤ `cd /home/user/you/WRF/wrfsi`

➤ `./etc/window_domain_rt.pl -w wrfsi -t /home/user/you/WRF/wrfsi/templates/OnlineTut`

运行后，有 [这样的画面](#) 出现。中间会停顿十几秒，最后表示的是设定完成的提示信息：

```
*****
*****  Domain Localization complete  *****
*****
```

在 log file ([/home/user/you/WRF/wrfsi/domains/OnlineTut/log/localize_domain.log.date](#)) 处可以查看运行是否成功。(date 是刚才的运行时间)

如果 static file 被做成了的话，就表示运行成功了。

➤ `/home/user/you/WRF/wrfsi/domains/OnlineTut/static/static.wrfsi.d01`

可以使用 `ncdump -h /home/user/you/WRF/wrfsi/domains/OnlineTut/static/static.wrfsi.d01`，来浏览有什么内容被写进了文件。如果这个文件被成功地建立了，就可以向输入 GRIB 的 deGrib 移动了。

STEP2: deGrib GRIB files

对所有的 GRIB 数据文件都需要进行 deGrib。

Script needs to deGrib GRIB file: [/home/user/you/WRF/wrfsi/etc/grib_prep.pl](#)

Namelist: [/home/user/you/WRF/wrfsi/extdata/static/grib_prep.nl](#)

运行自己所需要的数值模拟，必须有相对应的 GRIB 数据文件。作为本次的模拟练习，我们可以从网上下载到 [AVN test data for the Katrina case](#)。

Katrina case

- Global AVN/GFS (90.0 to -90.0 by 1.0 latitude & 0.0 to 360.0 by 1.0 longitude)
- [List of field](#) available in these files
- Start date: 2005082800
- End date: 2005083100
- Frequency: 6 hourly
-

把上面下载到的文件放入其专用的文件夹里，例如创建 `../WRF/wrfsi/case-name/`。(推荐把各种模拟用的气象数据资料放到各自专用的文件夹里，每次模拟都在各自的文件夹里运行)

在 /home/user/you/WRF/wrfsi/ 下，建立一个名为 Katrina 的目录，放入下载到的 GRIB 数据文件。

➤ `mkdir Katrina`

把下载到的 Katrina_AVN_input.tar.gz 文件复制到 Katrina 目录里。然后解压。

➤ `tar xvfz Katrina_AVN_input.tar.gz`

Edit grib_prep.nl

在/home/user/you/WRF/wrfsi/extdata/static/目录下，使用 **vi** 编辑 grib_prep.nl 文件。

编辑 **gpinput_defs** 部分的时候，要注意上下文之间的位置关系，各种SRC* 都是上下相互对应的。

当有AVN data的时候，对**3th path in the SRCPATH**变量必须定义到含有GRIB input file的directory。

具体的编辑方法可参考下面的例子（已编辑完）：

```
&filetimespec
  START_YEAR = 2005
  START_MONTH = 08
  START_DAY = 28
  START_HOUR = 00
  START_MINUTE = 00
  START_SECOND = 00
  END_YEAR = 2005
  END_MONTH = 08
  END_DAY = 31
  END_HOUR = 00
  END_MINUTE = 00
  END_SECOND = 00
  INTERVAL = 21600
/

&gpinput_defs
  SRCNAME = 'ETA', 'GFS', 'AVN', 'RUCH', 'NNRP', 'NNRPSFC', 'SST'
  SRCVTAB = 'ETA', 'GFS', 'AVN', 'RUCH', 'NNRPSFC', 'NNRPSFC', 'SST'
  SRCPATH = '/public/data/grids/eta/40km_eta212_isobaric/grib',
            '/public/data/grids/gfs/0p5deg/grib',
            '/home/user/you/WRF/wrfsi/Katrina',    (刚才的解压目录)
            '/rt0/rucdev/nrelwind/run/maps_fcst',
            '/path/to/nnrp/grib',
            '/path/to/nnrp/sfc/grib',
            '/public/data/grids/ncep/sst/grib'

  SRCCYCLE = 6, 6, 6, 6, 12, 12, 24
  SRCDELAY = 3, 4, 4, 3, 0, 0, 12
/
```

键入<Esc> : w q 保存退出。准备运行 script。

➤ **cd /home/user/you/WRF/wrfsi**

➤ **./etc/grib_prep.pl -s 2005082800 -l 72 -t 6 AVN** （1 为小写的 L）

（特别要注意选项的含义：-s=start time, -l=forecast length, -t=interval）

运行结束后，检查/home/user/you/WRF/wrfsi/extdata/log/gp_AVN.200508280.log 文件。

如果成功了，将在/home/user/you/WRF/wrfsi/extdata/extprd 里有生成 intermediate files 文件。

（一般情况下，如果没有对上面的文件进行正确编辑，则不会产生下面的文件。）

```
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-28_00
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-28_06
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-28_12
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-28_18
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-29_00
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-29_06
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-29_12
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-29_18
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-30_00
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-30_06
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-30_12
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-30_18
-rw-rw-rw- 1 you user 38867544 Mar 20 12:22 AVN:2005-08-31_00
```

此处成功了的话，就表示准备好了向在 step1（localization）处做成的 WRF model 领域内插数据的工作。

STEP3: Interpolate meteorological data

在此处，向模式领域里插入气象数据。将会运行 hinterp 和 vinterp 部分。

Script needed: /home/user/you/WRF/wrfsi/etc/wrfprep.pl

Namelist: wrfsi.nl

- * 如果在 step1 里，已经对 wrfsi.nl 编辑过了，此处就没有必要再编辑了。
- * 如果，在运行 wrfprel.pl 之前，想要更改 [inter control](#) 或 [si_paths](#) 部分，对
/home/user/you/WRF/wrfsi/domains/OnlineTut/static/wrfsi.nl 也要编辑。不是复本
/home/user/you/WRF/wrfsi/templates/OnlineTut/wrfsi.nl。

准备运行 script。

```
> cd /home/user/you/WRF/wrfsi
> ./etc/wrfprep.pl -s 2005082800 -f 72 -t 6
```

这是 wrfsi.nl 里的时间设定部分（filetimespec），可以用命令文进行覆盖。（特别要注意选项的含义：-s=start time，-f=forecast length，-t=interval）

在运行当中，你会看到如下画面。

```
Routine: wrfprep.pl
INSTALLROOT = /home/user/you/WRF/wrfsi
MOAD_DATAROOT = /home/user/you/WRF/wrfsi/domains/OnlineTut
Start time: 2005/08/28 00:00:00
End time:    2005/08/31 00:00:00
CYCLE.0524000000072
ictime = 2005-08-28_00
lbctime = 2005-08-28_06
```

```

lbctime = 2005-08-28_12
lbctime = 2005-08-28_18
lbctime = 2005-08-29_00
lbctime = 2005-08-29_06
lbctime = 2005-08-29_12
lbctime = 2005-08-29_18
lbctime = 2005-08-30_00
lbctime = 2005-08-30_06
lbctime = 2005-08-30_12
lbctime = 2005-08-30_18
lbctime = 2005-08-31_00

/home/user/you/WRF/wrfpsi/domains/OnlineTut/siprd/hinterp.d01.2005-08-31_00:00:00
/home/user/you/WRF/wrfpsi/domains/OnlineTut/siprd/wrf_real_input_em.d01.2005-08-31_00:00:00

```

运行结束后，检查一下/home/user/you/WRF/wrfpsi/domains/OnlineTut/log 里的 log file。

有以下的文件出现即可。

```

2005082800.hinterp
2005082800.vinterp
2005082800.wrfprep

```

成功了的话，在/home/user/you/WRF/wrfpsi/domains/OnlineTut/siprd 里将有下面几个文件生成。

```

-rw-r--r--  1 you user      185 Mar  21 15:13 CYCLE.0524000000072
-rw-r--r--  1 you user 2821104 Mar  21 15:13 hinterp.d01.2005-08-28_00:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:13 hinterp.d01.2005-08-28_06:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:13 hinterp.d01.2005-08-28_12:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-28_18:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-29_00:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-29_06:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-29_12:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-29_18:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-30_00:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-30_06:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-30_12:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-30_18:00:00
-rw-r--r--  1 you user 2821104 Mar  21 15:14 hinterp.d01.2005-08-31_00:00:00
-rw-r--r--  1 you user      368 Mar  21 15:14 hinterp.global.metadata
-rw-r--r--  1 you user 4338960 Mar  21 15:14 wrf_real_input_em.d01.2005-08-28_00:00:00
-rw-r--r--  1 you user 4338576 Mar  21 15:14 wrf_real_input_em.d01.2005-08-28_06:00:00
-rw-r--r--  1 you user 4338576 Mar  21 15:14 wrf_real_input_em.d01.2005-08-28_12:00:00
-rw-r--r--  1 you user 4338576 Mar  21 15:14 wrf_real_input_em.d01.2005-08-28_18:00:00

```

```

-rw-r--r-- 1 you user 4338576 Mar 21 15:14 wrf_real_input_em.d01.2005-08-29_00:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:14 wrf_real_input_em.d01.2005-08-29_06:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:14 wrf_real_input_em.d01.2005-08-29_12:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:14 wrf_real_input_em.d01.2005-08-29_18:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:14 wrf_real_input_em.d01.2005-08-30_00:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:14 wrf_real_input_em.d01.2005-08-30_06:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:13 wrf_real_input_em.d01.2005-08-30_12:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:13 wrf_real_input_em.d01.2005-08-30_18:00:00
-rw-r--r-- 1 you user 4338576 Mar 21 15:13 wrf_real_input_em.d01.2005-08-31_00:00:00

```

到此处你也成功了的话，我们就已经做好运行 WRF ARM 模式的准备好了。

4.3. ~安装 WRFV2.2~

Examine the Source Code

➤ **cd WRFV2**

在这个目录下存在有以下的文件和目录：

```

-rw-r--r-- 1 you user 16421 Oct 21 2005 Makefile
-rw-r--r-- 1 you user 7250 Nov 9 2005 README
-rw-r--r-- 1 you user 7238 Oct 5 2005 README.NMM
-rw-r--r-- 1 you user 2548 May 18 2004 README_test_cases
drwxr-xr-x 2 you user 4096 Apr 11 16:00 Registry
drwxr-xr-x 2 you user 4096 Nov 9 2005 arch
drwxr-xr-x 2 you user 4096 Nov 9 2005 chem
-rwxr-xr-x 1 you user 1078 May 24 2005 clean
-rwxr-xr-x 1 you user 6913 Oct 21 2005 compile
-rwxr-xr-x 1 you user 10636 May 7 2005 configure
drwxr-xr-x 2 you user 4096 Apr 20 20:01 dyn_em
drwxr-xr-x 2 you user 4096 Nov 9 2005 dyn_exp
drwxr-xr-x 2 you user 4096 Nov 9 2005 dyn_nmm
drwxr-xr-x 12 you user 4096 Nov 9 2005 external
drwxr-xr-x 2 you user 4096 Apr 11 16:02 frame
drwxr-xr-x 2 you user 4096 Mar 21 16:44 inc
drwxr-xr-x 2 you user 4096 Mar 21 16:44 main
drwxr-xr-x 2 you user 4096 Apr 11 16:13 phys
drwxr-xr-x 2 you user 4096 Mar 21 16:44 run
drwxr-xr-x 2 you user 8192 Mar 21 16:44 share
drwxr-xr-x 11 you user 4096 Nov 9 2005 test
drwxr-xr-x 3 you user 4096 Apr 11 16:01 tools

```

在 README file 里，写有关于 code 的有用信息和模式运行的设定方法。

WRFV2 目录下的文件分类:

Source code directories:

chem/	Directory containing modules for chemistry (not currently supported)
dyn_em/	Directory containing modules for dynamics in WRF ARW core
dyn_nmm/	Directory containing modules for dynamics in WRF NMM core
dyn_exp/	Directory for a 'toy' dynamical core
external/	Directory containing external packages, such as those for IO, time keeping and MPI
frame/	Directory containing modules for WRF framework
inc/	Directory containing include files
main/	Directory for main routines, such as wrf.F, and all executables after install
phys/	Directory for all physics modules
share/	Directory containing mostly modules for WRF mediation layer and WRF I/O
tools/	Directory containing tools

Scripts:

clean	Script to clean created files, executables
compile	Script for compiling WRF code
configure	Script to configure the configure.wrf file for compile
Makefile	Top-level makefile
Registry/	Directory for WRF Registry file
arch/	Directory where compile options are gathered
run/	Directory where one may run WRF
test/	Directory that contains 7 test case directories, may be used to run WRF

Environment Variable-NETCDF

定义 NetCDF 环境变量:

➤ **NETCDF=/home/user/you;export NETCDF**

Configure WRFV2

键入:

➤ **./configure**

在此, 会显示出计算机所支持的平台。

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home/user/you/netcdf-3.6.1
PHDF5 not set in environment. Will configure WRF for use without.
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
-----
Please select from among the following supported platforms.
```

1. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher (Single-threaded, no nesting)

2. PC Linux x86_64 (IA64 and Opteron), PGI 5.2 or higher, DM-Parallel (RSL, MPICH, Allows nesting)
3. PC Linux x86_64 (IA64 and Opteron), PGI 5.2 or higher DM-Parallel (RSL_LITE, MPICH, Allows nesting, No periodic LBCs)
4. PC Linux x86_64 (IA64 and Opteron), PGI compiler 5.2 or higher (Single-threaded, RSL, Allows nesting)
5. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (single-threaded, no nesting)
6. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (single threaded, allows nesting using RSL without MPI)
7. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (OpenMP)
8. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler SM-Parallel (OpenMP, allows nesting using RSL without MPI)
9. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+icc compiler DM-Parallel (RSL, MPICH, allows nesting)
10. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+gcc compiler DM-Parallel (RSL, MPICH, allows nesting)
11. PC Linux x86_64 (IA64 and Opteron), PathScale 2.1 or higher (Single-threaded, no nesting)
12. PC Linux x86_64 (IA64 and Opteron), PathScale 2.1 or higher DM-Parallel (RSL_LITE, PathScale MPICH, No periodic LBCs)
13. Cray XT3 Catamount/Linux x86_64 (Opteron), PGI 5.2 or higher DM-Parallel (RSL_LITE, MPICH, Allows nesting, Periodic in X only)

Enter selection [1-13] :

(根据计算机的差异, 有可能会显示出更多的平台选项)

作为练习, 我们选择最为基本的 (single-threaded, no nesting)。然后键入:

➤ 1

如果要进行并行计算或者嵌套的话, 就要选择 3。

在此步骤, 将有 configure.wrf 文件被做成。如果有必要, 对此文件的 compile options/paths 也要进行编辑。

Compile WRFV2

➤ ./compile

然后屏幕出现:

Usage:

compile wrf compile wrf in run dir (NOTE: no real.exe, ndown.exe, or ideal.exe generated)

or choose a test case (see README_test_cases for details) :

compile em_b_wave

compile em_grav2d_x

compile em_hill2d_x

compile em_quarter_ss


```

compile em_real
compile em_squall2d_x
compile em_squall2d_y
compile exp_real
compile nmm_real

compile -h          help message

```

因为要对 WRF ARW real data case 进行编辑，所以选择 `em_real`。

然后开始编译模块。

➤ `./compile em_real >& compile.log`

运行会稍微花一点时间。（如果在 `./configure` 处选择了 **3**，这时你会有将近 30 分钟左右的时间来喝杯咖啡。）如果在进行过程当中自己弄出错了，或者要更改 `configure`、`compile` 等的选项，可以用 “`clean -a`” 的命令来清理所有的 built files, including `configure.wrf`。

➤ `./clean -a`

然后再重新编译 WRFV2.2。

编译结束后，打开 [compile.log](#) 确认是否有错误信息存在。如果编辑成功，在 `main/` 目录下会有以下的文件被做成！

```

-rwxr-xr-x  1 you user 13612035 Mar 22 15:11 ndown.exe
-rwxr-xr-x  1 you user 13237353 Mar 22 15:11 nup.exe
-rwxr-xr-x  1 you user 10909533 Mar 22 15:11 real.exe
-rwxr-xr-x  1 you user 13237353 Mar 22 15:11 wrf.exe

```

`ndown.exe` 用于 one-way nesting 。

`nup.exe` 用在 WRF_3DVAR。

`real.exe` 用于 real data cases 的初期化

`wrf.exe` 用于 WRF model integration

这些文件被从 `main/` 目录链接到了 `run/`，`test/em_real/` 目录下。

（如果运行其它 idealized cases，方法也一样。在 `./compile em_real >& compile.log` 处选择相应的文件选项。不过做后会生成 `ideal.exe`）

4.4. ~运行 WRFV2.2~

Edit namelist.input

移动到 `run/` 或 `test/em_real` 目录下。无论哪个都可以。在这里，我们选择后者。

➤ `cd test/em_real`

（如果在 `em_real` 下不能正常运行 `real.exe` 的话，可以到 `run` 目录下试试）

在此处会有以下文件生成。

```

lrwxrwxrwx  1 you user      23 Mar 22 15:11 ETAMPNEW_DATA -> ../../run/ETAMPNEW_DATA
lrwxrwxrwx  1 you user     21 Mar 22 15:11 GENPARAM.TBL -> ../../run/GENPARAM.TBL

```

```

lrwxrwxrwx    1 you user      21 Mar 22 15:11 LANDUSE.TBL -> ../../run/LANDUSE.TBL
lrwxrwxrwx    1 you user      25 Mar 22 15:11 README.namelist -> ../../run/README.namelist
lrwxrwxrwx    1 you user      19 Mar 22 15:11 RRTM_DATA -> ../../run/RRTM_DATA
lrwxrwxrwx    1 you user      22 Mar 22 15:11 SOILPARM.TBL -> ../../run/SOILPARM.TBL
lrwxrwxrwx    1 you user      21 Mar 22 15:11 VEGPARM.TBL -> ../../run/VEGPARM.TBL
lrwxrwxrwx    1 you user      21 Mar 22 15:11 gribmap.txt -> ../../run/gribmap.txt
-rw-r--r--    1 you user      1762 Feb 19   2005 landFilenames
-rw-r--r--    1 you user      5485 Mar  21 17:03 namelist.input
-rwxr-xr-x    1 you user      5489 Jun 20   2005 namelist.input.jan00
-rwxr-xr-x    1 you user      5488 Jun 20   2005 namelist.input.jun01
lrwxrwxrwx    1 you user      20 Mar 22 15:11 ndown.exe -> ../../main/ndown.exe
lrwxrwxrwx    1 you user      19 Mar 22 15:11 real.exe -> ../../main/real.exe
-rw-r--r--    1 you user     10240 Sep 17   2004 run_1way.tar
-rw-r--r--    1 you user     10240 Oct 18   2005 run_restart.tar
lrwxrwxrwx    1 you user      17 Mar 22 15:11 tr49t67 -> ../../run/tr49t67
lrwxrwxrwx    1 you user      17 Mar 22 15:11 tr49t85 -> ../../run/tr49t85
lrwxrwxrwx    1 you user      17 Mar 22 15:11 tr67t85 -> ../../run/tr67t85
lrwxrwxrwx    1 you user      18 Mar 22 15:11 wrf.exe -> ../../main/wrf.exe

```

这些文件的用途见上一章的[运行WRFV2.2](#)部分。

针对本次的OnLine Tutorial case，编辑[namelist.input](#)。

归纳一下，主要修改以下几点。

```

Start date: 2005082800
End date: 2005083100
Interval: 21600 sec (6 hours)
Grid points in EW direction: 75
Grid points in SN direction: 70
Number of vertical levels: 31
Grid distance: 30km

```

同样，namelist.input 也被用在了 real.exe 和 wrf.exe 可执行文件处（应该也是被链接过去的吧）。

Link

把在WRF_SI处做成的[wrf_real_input* files](#)链接到WRFV2/test/em_run/ 目录下。

➤ `ln -sf /home/user/you/WRF/wrfsi/domains/OnlineTut/siprd/wrf_real_input_em.d01.2005-08-*` .

使用方法可以参考 `ln` 命令

Run real.exe

进行 single processor 运行。

➤ `./real.exe`

如果是 DM(distributed memory) parallel systems，就会需要 mpirun command 。例如，对于 Linux cluster，

要运行 4 个 CPU 的 MPI code 的命令文就是：

➤ **mpirun -np 4 real.exe**

可以根据自己的条件选择 CPU 的个数。同时不要忘记，在 **./configure** 处要选择对应的平台。

在键入 **./real.exe** 后，最好不要打扰计算机的运行。如果运行成功的话，最后会有 **wrfbdy_d01** 和 **wrfinput_d01** 文件生成。

Check your rsl file

确认 **rsl.out.*** 和 **rsl.error.***（对于每个 processor，都会出现 **rsl.out** 和 **rsl.error**）。在 **rsl.out.0000** 和 **rsl.error.0000** 里包含了重要的情报。如果运行失败，**error message** 有可能存在于这些文件当中的某一个文件里。同时，在 **namelist.input** 里有 **debug_level** 一项。这一项对应的数值为 0, 50, 100, 200, 300 等，数值越大，在 **rsl.out.*** 和 **rsl.error.*** 里输出的信息越详细，有利于寻找错误的根源。如果运行不能成功，请考虑使用此项。

使用 **head -n 30 rsl.out.*** 和 **tail -n 30 rsl.out.*** 命令来查看 **rsl** 的文件头 30 行和文件尾 30 行。

如果运行 **real.exe** 成功，则在 **rsl.out.0000** 的最后会有提示：

SUCCESS COMPLETE REAL_EM INIT

因为 **wrf.exe** 也会生成同名的 **log** 文件，所以把上面的 **rsl.out.*** 和 **rsl.error.*** 移动到其他目录，或者删掉。

Run wrf.exe

进行 single processor 运行。

➤ **./wrf.exe**

如果是 DM(distributed memory) parallel systems，会有需要 **mpirun command** 的场合。例如，如果是 Linux cluster，实行 4CPUs 的 MPI code 的命令就是：

➤ **mpirun -np 4 wrf.exe**

如果全部进行顺利的话，会有 **wrfout_d01_file** 被做成。

从 2005082800 到 2005083100 的 72 小时份的文件将被做成。用 **ncdump** 可以查看 **wrfout_d01 file** 里的信息。

➤ **ncdump -h wrfout_d01_2005-08-28_00:00:00** [参照](#)

wrf: SUCCESS COMPLETE WRF

5.<安装使用 WRF2GrADS>

在 WRF 的 tool 里直接介绍了四种气象作图的工具，并为他们提供了相应的文件格式的转换工具。

它们是：

- NCL
- RIP4
- WRF-to-Grads
- WRF-to-vis5d

对于这四种制图软件，WRF有专门的讲解PowerPoint。如果想知道的更多，可以下载[WRF Post-Processing](#)。

其中 Grads 的历史悠久，功能强大，使用也不算麻烦。并且，在一些有实力的论坛上对 GrADS 的讨论也比较多，有什么不明白的问题可以从论坛里得到一定的帮助。作为练习，把 WRF 的 output 数据转化成可用于 GrADS 的数据格式。

Install wrf2grads

具体方法可参照[WRF2GrADS](#)的相关网页。

从已经下载的[Source Codes](#)中复制wrf2grads.tar.gz 文件到/home/user/you/WRF下，用gunzip 和 tar命令进行解压。

- `gunzip wrf2grads.tar.gz`
- `tar -xvf wrf2grads.tar`

在/home/user/you/目录下会产生名为 WRF2GrADS 的目录。主要有以下文件：

```
Makefile
README
control_file
control_file_height
control_file_pressure
module_wrf_to_grads_netcdf.F
module_wrf_to_grads_util.F
wrf_to_grads.F
```

Edit Makefile & control_file

在使用WRF2GrADS之前，一定要仔细阅读其自带的[README](#)文件。

- `cd WRF2GrADS`

我们需要通过编辑Makefile来选择一款适合于自己计算机的描述。

- `vi Makefile`

比如，适合我使用的计算机的描述是：

```
.....
# linux flags (PGI)

#LIBNETCDF = -L/usr/local/netcdf/lib -lnetcdf -lm
#INCLUDE = -I/usr/local/netcdf/include -I./
#FC = pgf90
#FCFLAGS = -g -C -Mfree
#FCFLAGS = -fast -Mfree
#CPP = /usr/bin/cpp
#CPPFLAGS = -I. -C -traditional -DRECL4
.....
（我们可以根据“FC = ”的性质来选择适合我们计算机的描述）
```

然后对 Makefile 进行编辑(红色部分)：

```
.....
# linux flags (PGI)

LIBNETCDF = -L/home/user/you/lib -lnetcdf -lm
INCLUDE = -I/home/user/you/include -I./
FC = pgf90
FCFLAGS = -g -C -Mfree
FCFLAGS = -fast -Mfree
CPP = /usr/bin/cpp
CPPFLAGS = -I. -C -traditional -DRECL4
.....
```

注意：一定要去掉描述语句前的 # 字符。然后<Esc>:wq 保存退出。

然后键入：

➤ **make**

在目录下会自动生成名为 wrf_to_grads 的可执行文件。

通过对 control_file 文件的编辑，使 wrf2grads 可以生成时间相对应的 output 文件。生成的文件为可使用在 GrADS 里的.ctl 和.dat 文件。例如：

➤ **vi control_file**

然后，主要对以下信息进行编辑：

- set times to be processed
- set variables to be processed
- define the input file
- specify if the input is real/ideal/static data
- set levels to interpolate too

例如，对于我自己的文件，编辑部分的 control_file 的如下：

```

7                ! number of times to put in GrADS file, negative means ignore the times
2005-08-28_00:00:00
2005-08-28_12:00:00
2005-08-29_00:00:00
2005-08-29_12:00:00
2005-08-30_00:00:00
2005-08-30_12:00:00
2005-08-31_00:00:00
end_of_time_list

                ! 3D variable list for GrADS file
                ! indent one space to skip

.....
/ DATA/real/wrfinput_d01

/home/user/you/WRF/WRFV2/run/wrfout_d01_2005-08-28_00:00:00

/ DATA/b_wave/wrfout_d01_0001-01-01_00:00:00
/ DATA/grav2d_x/wrfout_d01_0001-01-01_00:00:00
.....

```

编辑好后键入<Esc> : w q 保存退出。

注意：

- ①. 在这里编辑的第一个数字与下面的列出的时间相对应。在 GrADS 里，这个数字被用在了 t 时次。例如：我们在 GrADS 里，可使用命令：set t 1 7。当数字少于下面给出的时间列的时候，output 文件里只会生成前几个时间列的数据；当数字为负的时候，会忽略数字，并对后面所有列出的时间进行处理。
- ②. 最后，前往不能忘记要加上 end_of_time_list 作为结尾。
- ③. 在 control_file 的编辑过程中不可多加空行，不然会产生 error。
- ④. 暂时未对 input data , set levels 等项进行编辑。

Run wrf2grads

开始运行转换。

➤ `./wrf_to_grads control_file wrfout_d01_2005-08-28_00:00:00 -v`

直接运行 `wrf_to_grads` 和 `control_file` 文件。在这里，`wrfout_d01_2005-08-28_00:00:00` 是转换后生成文件的文件名，可以自行定义任意文件名；最后的 `-v` 是表示属性。

如果顺利，格式转换完后会出现如下的信息：

```
... ..
writing out variable, time          7          3
time 2005-08-30_00:00:00, output variable Z
getting data for HGT
writing out variable, time          8          3
time 2005-08-30_00:00:00, output variable HGT

-----
                Gracefull STOP
-----
```

在当目录下会生成 `****.ctl` 和 `****.dat` 文件。例如：

`wrfout_d01_2005-08-28_00:00:00.ctl`

`wrfout_d01_2005-08-28_00:00:00.dat`

这两个文件 `ctl` 和 `dat` 将会在 GrADS 画图软件中使用。

6.<在 UNIX 系统下安装 GrADS>

[GrADS](#)是一款功能强大的气象绘图工具。关于GrADS的中文版的使用手册，可以到[LASG](#)（大气科学和地球流体力学数值模拟国家重点实验室）的动力论坛[专业绘图软件](#)去下载。里面还介绍了在WINDOWS下的安装方法。关于[GrADS](#)软件，可以到其主页下载。对于不同的平台，提供了相应的版本。我是在UNIX系统下安装的GrADS，在这里只是简单的描述UNIX系统下的安装过程。如果只需要source_code ,可以使用[grads-src-1.8sl1](#) 或 [grads-src-1.9b4](#) 。我使用的是后者。

在Windows下的安装方法请参考[LASG](#)编的[《GrADS实用手册》](#)。

Install GrADS

安装之前，定义 NetCDF 的环境变量。

➤ **NETCDF=/home/user/you;export NETCDF**

把 grads-src-1.9b4 下载到/home/user/you/目录下，用 gunzip 和 tar 解冻。会生成一个名为 grads-1.9b4 的文件夹。进入文件夹，键入以下命令进行软件的安装。

➤ **./configure**

➤ **make**

➤ **make install**

安装完毕。

Set Environment Variables

使用 Grads 前要进行环境变量的定义。对 GADDIR, GASCRP, GAUDFT, PATH 四个环境变量进行定义。如下：

➤ **GADDIR=/home/user/you/grads-1.9b4/data;export GADDIR**

➤ **GASCRP=/home/user/you/grads-1.9b4;export GASCRP**

➤ **GAUDFT=/home/user/you/grads-1.9b4/data;export GAUDFT**

➤ **PATH=\$PATH:\$GADDIR;export PATH**

这些环境变量同样可以加入上面的隐藏文件[.bashrc](#)中。

Open GUI

因为要使用GUI，可以先运行[前面](#)讲过的方法打开GUI。

Run GrADS

然后进入存放有 wrfout_d01_2005-08-28_00:00:00.ctl 和 wrfout_d01_2005-08-28_00:00:00.dat 的文件夹，即生成 ctl 和 dat 文件的目录 WRF2GrADS。

➤ **cd /home/user/you/WRF2GrADS**

这里的.ctl和.dat文件即是由[WRF2GrADS](#)生成的文件。在此目录下，键入如下命令来打开grads。

➤ **/home/user/you/grads-1.9b4/bin/gradsc**

这时有一个窗口被打开，这就是我们要进行查看绘图的窗口。同时还会出现如下的对话：

Grid Analysis and Display System (GrADS) Version 1.9b4

Copyright (c) 1988-2005 by Brian Doty and IGES

Center for Ocean-Land-Atmosphere Studies (COLA)

Institute for Global Environment and Society (IGES)

GrADS comes with ABSOLUTELY NO WARRANTY

See file COPYRIGHT for more information

Config: v1.9b4 32-bit little-endian lats

Issue 'q config' command for more information.

Landscape mode? (no for portrait):

键入 **y** 和回车或者直接回车选择默认，就会直接进入 GrADS 的对话模式。

.....

Landscape mode? (no for portrait): **y**

GX Package Initialization: Size = 11 8.5

ga>

在提示符 **ga>** 后输 GrADS 命令文，就可操作 GrADS 进行制图了。

关于GrADS的具体使用方法，可以参考LASG编的[《GrADS实用手册》](#)。

7.<利用其它数据的练习>

通过对《WRF V2 安装运行入门指南》的前面的 6 节的学习，大家应该都了解了 WRF 的基本运行步骤了。这时，你也一定想通过运行一些其它领域和时间的案例，来巩固和验证自己对 WRF 的理解和运用。我们可以从 WRF 推荐的几个网页上下载一些数据来使用。

Download Dataset

- * 静态数据：可以使用已经装好的 Katrina 的那款；
- * 气象数据：WRF_SI和WPS的输入数据的格式有grid1 和grid2。我们可以到NCAR的[dss](#)处下载[ds083.2](#)数据来使用。ds083.2 是NECP的客观分析数据FNL形式。在其下载页面上仅登有最近一年的数据可供免费下载。这些数据的文件名格式是fnl_YYMMDD_hh_mm，即 fnl_年年月月日日_时时_分分。注意：MS的IE下载到的这些数据通常会被追加成 .txt 的文本格式。不要轻易打开，去掉后面的.txt一样可用。个人经验，使用Opera也许下载会更方便。这些气象数据都是全球范围的。除此之外，还有其他现成的数据可以使用。参考WRF的[网页](#)。

～ WPS+WRFV2.2 ～

WPS

Run WPS

在WRF/WPS/目录下，编辑[namelist.wps](#)。

➤ `./geogrid.exe`

并行计算的话

➤ `mpirun -np 4 geogrid.exe`

➤ `./link_grib.csh /home/user/you/WRF/DATA/TESTDATA/fnl_0610`

➤ `ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable`

➤ `./ungrib.exe >& ungrib.log`

➤ `./metgrid.exe`

并行计算的话

➤ `mpirun -np 4 metgrid.exe`

WRFV2.2

在/WRF/WRFV2/目录下

➤ `cd test/em_real` or `cd run`

编辑[namelist.input](#)

➤ `ln -sf /home/user/you/WRF/WPS/met_em.d01.2005-08*.`

➤ `./real.exe`

并行计算的话...

➤ `mpirun -np 4 real.exe`

➤ `./wrf.exe`

并行计算的话...

➤ `mpirun -np 4 wrf.exe`

~ WRF_SI+WRFV2.2 ~

WRF_SI

Set Environment Variables

定义环境变量 可使用[source ~/.bashrc](#) 命令。

Run WRF_SI

STEP1

在/WRF/wrfsi/目录下：

- `cd domains`
- `mkdir TestOne`
- `MOAD_DATAROOT=/home/user/you/WRF/wrfsi/domains/TestOne;export MOAD_DATAROOT`
- `cd /home/user/you/WRF/wrfsi/templates`
- `cp -r default TestOne`
- `chmod -R u+w TestOne`
- `cd TestOne`

编辑[wrfsi.nl](#)的 1 和 2 部分

- `cd /home/user/you/WRF/wrfsi`
- `./etc/window_domain_rt.pl -w wrfsi -t /home/user/you/WRF/wrfsi/templates/TestOne`

STEP2

新建一个专门存放 TestOne 用的气象数据文件夹。比如说，在/WRF/wrfsi/下建立一个名为 FTestone 的目录。把要处理的数据放入 FTestone 目录下。同时要注意：①文件名格式；②数据时间上的连续；③不要放入无关文件。比如说，我在 FTestone 目录里存放了 fnl_061020_12_00~fnl_061022_00_00 共 6 个数据文件。

编辑[grib_prep.nl](#) 硬性指定数据文件为“AVN”数据。

- `cd /home/user/you/WRF/wrfsi`
- `./etc/grib_prep.pl -s 2006102012 -l 36 -t 6 AVN`

STEP3

编辑[wrfsi.nl](#)的 3 和 4 部分：特别注意下面两个变量的设定为“AVN”：

`INIT_ROOT = 'AVN'`

`LBC_ROOT = 'AVN'`

- `cd /home/user/you/WRF/wrfsi`
- `./etc/wrfprep.pl -s 2006102012 -f 36 -t 6`

WRF V2.2

在/WRF/WRFV2/目录下，

- `cd test/em_real` or `cd run`

编辑[namelist.input](#)

- `ln -sf /home/user/you/WRF/wrfsi/domains/TestOne/siprd/wrf_real_em.d01.2006-10-* .`
- `./real.exe`

并行计算的话...

- `mpirun -np 4 real.exe`

```
> ./wrf.exe
```

并行计算的话...

```
> mpirun -np 4 wrf.exe
```

WRF2GrADS

在/home/user/you/WRF/目录下

```
> cd WRF2GrADS
```

[编辑control_file](#)

然后运行 wrf_to_grads。

```
> ./wrf_to_grads control_file TestOne -v
```

GrADS

Set Environment Variables

[定义环境变量](#) 如果把环境变量一并写入.bashrc里的话,可以省去这一步骤。

在这里要[运行GUI](#), 所以要事先打开X-window。

```
> cd /home/user/you/WRF2GrADS
```

```
> /home/user/you/grads-1.9b4/bin/gradsc
```

进入 GrADS 软件的操作

```
> open TestOne.ctl
```

```
....
```

附录 1: 安装 NCARG (NCAR Graphic)

NCARG 是 NCAR Graphic, 由 NCAR 开发的视图工具。与 GrADS 相比, NCARG 的应用并不广泛。但作为 NCAR 的产品, 被积极地用在了 WPS 的领域查看和设计等中途工作, 而且对于有些工作也可以使用 gradsnc 来代替执行。但 NCAR 既然提到了使用 NCARG, 就姑且把 NCARG 的安装方法一并记述下来, 以供日后参考使用考。

NCARG 的下载方法和详细的安装方法, 请参考其[主页](#)。

Install NCARG

把 NCARG 下载到 WRF/目录下并解压。

➤ `gunzip ncarg-4.4.1.src.tar.gz`

➤ `tar -xvf ncarg-4.4.1.src.tar`

这时, WRF/目录下会生成名为 ncarg-4.4.1 的目录。然后定义环境变量。

➤ `NCARG=/home/user/you/WRF/ncarg-4.4.1;export NCARG`

➤ `cd ncarg-4.4.1`

➤ `./Configure -v`

如果顺利, 屏幕开始出现一些对话, 你会被问到几个问题。这时你必须要清楚自己使用的计算机的环境设定状况: 安装 NCARG 时的默认设置是 /usr/local/ncarg/lib 等等, 这些都必须更改为 NCARG 的路径 (例如, 可更改为 /home/user/you/WRF/ncarg-4.4.1/lib)。其它环境设定状况例如: 使用的 X11 软件是 32-bit 还是 64-bit (X11 的开发等原因, 一般有 /usr/X11R6/lib (32bit 环境) 和 /usr/X11R6/lib64 (64bit 环境))、X11 的路径等等。也会被问到是否安装 HDF, 选 no 和 yes 均可。关于其他问题, 请自己仔细阅读判断并回答。

环境设定结束后, 安装之前我们仍然可以最后确定一下我们的选择和设定:

➤ `make Info`

然后屏幕中出现下面的信息:

NCAR Graphics - Version 4.4.1 Installation Configuration	
System File	LINUX
Binary Install Directory	/home/user/you/WRF/ncarg-4.4.1/bin
Library Install Directory	/home/user/you/WRF/ncarg-4.4.1/lib
Include Install Directory	/home/user/you/WRF/ncarg-4.4.1/include
Manpage Install Directory	/home/user/you/WRF/ncarg-4.4.1/man
Config Install Directory	/home/user/you/WRF/ncarg-4.4.1/lib/ncarg/config
Data Base Directory	/home/user/you/WRF/ncarg-4.4.1/lib/ncarg/database
Programmer Doc Dir	/home/user/you/WRF/ncarg-4.4.1/lib/ncarg/doc
Reloc Obj. Directory	/home/user/you/WRF/ncarg-4.4.1/lib/ncarg/robj
Examples Directory	/home/user/you/WRF/ncarg-4.4.1/lib/ncarg/examples
Tutorial Directory	/home/user/you/WRF/ncarg-4.4.1/lib/ncarg/tutorial
Test Directory	/home/user/you/WRF/ncarg-4.4.1/lib/ncarg/tests

```

X App. Def. Directory  /home/user/you/WRF/ncarg-4.4.1/lib/ncarg/xapp
f77 Compiler           pgf95
f77 Flags              -O
C Compiler             gcc
cc Flags               -ansi -O -I./include -I/usr/X11R6/include -DSYSV -D_POSIX_SOURCE
-D_XOPEN_SOURCE -DByteSwapped -DNeedFuncProto

```

对话结束后，输入 **make Everything** 进行 NCARG 软件安装。

➤ **make Everything >& make-output &**

安装过程是在后台进行的。我们可以使用 **tail** 命令来随时可以查看日志文件 **make-output** 的输出内容。

➤ **tail -f make-output**

如果中途出现了错误信息，请参考[Restarting the installation](#)，再进行reinstall。

安装结束后，定义环境变量：

➤ **NCARG_ROOT=/home/user/you/WRF/ncarg-4.4.1;export NCARG_ROOT**

➤ **PATH=\$NCARG_ROOT/bin:\$PATH**

➤ **MANPATH=\$NCARG_ROOT/man:\$MANPATH**

Check NCARG

校验NCARG的安装是否成功。NCARG里自带有检验安装的程序。这时要打开X-window，并对相应的[GUI环境](#)进行设定。然后按下面的方法运行。

➤ **ncargex cpex08**

NCAR Graphics Fortran Example <cpex08>

Copying cpex08.f

Copying cpexcc.f

Compiling and linking...

pgf95 -O -o cpex08 cpexcc.f cpex08.f -L/home/user/you/WRF/ncarg-4.4.1/lib -L/usr/X11R6/lib64

-lncarg -lncarg_gks -lncarg_c -lX11 -lXext

cpexcc.f:

cpex08.f:

Executing <cpex08>...

PLOT TITLE WAS EXAMPLE 8

INTEGER WORKSPACE USED 120

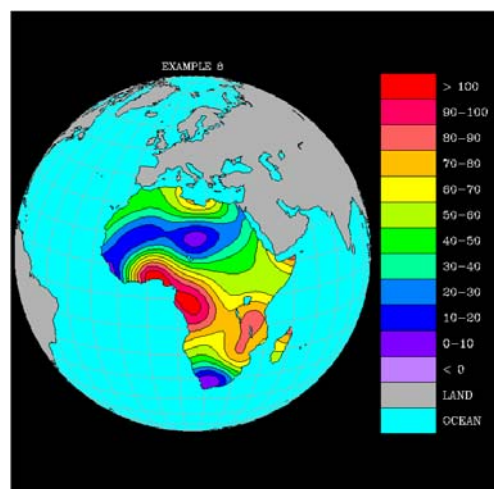
REAL WORKSPACE USED 400

AREA MAP SPACE USED 93250

FORTRAN STOP

Metafile file is named cpex08.ncgm.

➤ **ctrans -d X11 cpex08.ncgm**



然后，屏幕中会出现右图。

关于如何在WPS中使用NCARG，请参考WPS的[网页](#)。

附录 2:在WRF_SI阶段,对wrfsi.nl的设定极为重要。这是我在网上(LASG的动力论坛)收集到的有关wrfsi.nl的参数配置的中文说明。大家也可到[原地址](#)下载。

关于 WRF2.0 中 wrfsi.nl 的参数配置说明 (中文版)

(这篇文章已经被阅读了 n 次) 时间: 2004/06/24 02:53pm 来源: meteogg

[这个帖子最后由 meteogg 在 2004/06/24 02:54pm 编辑]

折腾了一天,终于把它整理成中文,希望能对使用者有帮助。另外有解释的不合适的地方,请提意见。全文如下:

A. PROJECT_ID Section

该部分为说明部分。该部分的设置不影响 WRF_SI 的运行,但是便于在输出元数据中说明运行。

1. SIMULATION_NAME: 指定本次运行实验的名称。
2. USER_DESC: 描述运行者。

B. FILETIMESEPC Section

该部分确定 SI 要处理的数据的起始和终止时间。注意: 如果要用 wrfprep.pl 运行 hinterp/vinterp (我们建议您必须.....), 脚本会自动为您编辑这些值。时间为世界时。

1. START_YEAR: 四位数表示起始年。
2. START_MONTH: 两位数表示起始月份。
3. START_DAY: 两位数表示起始日期。
4. START_MINUTE: 两位数表示起始的分钟数。
5. START_HOUR: 两位数表示起始的小时数。
6. START_SECOND: 两位数表示起始的秒数。
- 7-12. END_YEAR-END_SECOND: 和 1-5 一样, 但是为结束时间。
13. INTERVAL: 输出的时间间隔, 单位为秒。

C. HGRIDSPEC section

该部分定义 WRF 的水平格点。

1. NUM_DOMAINS: 区域数。如果不嵌套, 则定义为 1。
2. XDIM/YDIM: MOAD (母区域) 的格点数。尽管区域数大于 1, 即有嵌套时, 在 wrfsi.nl 中也仅设置一个母区域的 X 和 Y 方向上的格点, 因为嵌套区域的 xdim 和 ydim 根据其它参数自动计算 (见下面的设置 5)。

注意：WRF_SI 将底图定义为“非交错”网格。运用 Arakawa-C 交错格点假设所有 3 维变量（U， V， 和质量）关于这些点是交错格点。对于定义的非交错格点，U 格点向上交错了 0.5 个格点，V 格点向右交错了 0.5 个格点，质量网格分别向上向右交错了 0.5 个格点。为了便于说明，下面给出一个(XDIM, YDIM) = (4, 4)的例子：

```

+ V + V + V +
U T U T U T U
+ V + V + V +
U T U T U T U
+ V + V + V +
U T U T U T U
+ V + V + V +

```

(+) 为根据 namelist 中的参数定义的点。(T)为由 WRF 预报模式提供和输出的质量变量的格点位置。(U)点为由 WRF 模式提供和输出的 U 动量变量的格点位置。(V)点为由 WRF 模式提供和输出的 V 动量变量的格点位置。这样，如果 WRF_SI 配置使用维数(XDIM, YDIM)，则模式输出如下：

```

(XDIM-1, YDIM-1)维的质量变量
(XDIM, YDIM-1)维的 U 动量
(XDIM-1, YDIM)维的 V 动量

```

3. PARENT_ID: 嵌套区域的母区域的标号。注意 MOAD 本身没有母区域，因此 PARENT_ID 的第一列总是设为 1。第二列必须等于 1。总列数必须等于 NUM_DOMAINS。

4. RATIO_TO_PARENT: 该整数指定每个子区域和它们母区域的格距比。典型值为 2 和 3，但是也可以更大一些。该参数的列数必须等于 NUM_DOMAINS。

5. DOMAIN_ORIGIN_[LLI/LLJ/URI/URJ]: 这四个参数定义每个子区域在其母区域中西南（左下[LL]）和东北（右上[UR]）角的 ij 方向的格点位置。对于 MOAD 而言，SW = 1, 1 ; NE = xdim, ydim。则子区域的 xdim, ydim 值可由下式计算（SI 中自动计算）：

```

xdim = (uri-lli)*ratio_to_parent+1,
ydim = (urj-llj)*ratio_to_parent+1.

```

6. MAP_PROJ_NAME: 该字符串指定地图投影的方式。有效选项如下：

```

"polar" -> 极射投影
"lambert" -> 兰伯托等角投影（正割和正切）
"mercator" -> 麦卡托

```


7. MOAD_KNOWN_LAT/MOAD_KNOWN_LON: 网格中心点的经纬度。单位为度, 正的纬度表示北半球, 负值表示南半球。纬度在-90 和 90 之间, 经度在-180 到 180 之间。

8. MOAD_STAND_LATS: 两个实数表示标准纬度(此处格距不变)。必须在-90 和 90 之间, 根据投影选取该参数的值:

极射投影: 第一个参数为格距不变的纬度, 可设为中心纬度。第二个参数南/北半球为+/-90.。

兰勃托投影: 两个值均与中心纬度正负号相同。对于正切投影, 两个参数设为同一个值(通常设为中心纬度)。而对于正割投影, 两个参数必须为不同值。

麦卡托投影: 第一个参数为格距不变的纬度(通常为中心纬度)。第二个参数没有用到。

9. MOAD_STAND_LONS: 该参数指定于 y 坐标平行的经度(-180->180)。通常设为中心经度。

10. MOAD_DELTA_X/MOAD_DELTA_Y: 该浮点数分别指定东西和南北向的格距, 单位为米。目前, 两个值必须相同。嵌套子区域的格距由 SI 和 GUI 自动计算(见文件 src/lib/get_wrf_si_config.f 中的 function grid_spacing_wrf_m(nest)):

```
grid_spacing_wrf_m = moad_delta_x
i = nest:
do while (i.ne.1)
  grid_spacing_wrf_m = grid_spacing_wrf_m/ratio_to_parent(i)
  i = parent_id(i)
enddo
```

11. SILAVWT_PARM_WRF (见 README_toptwvl_silavwt in src/grid):

12. TOPTWVL_PARM_WRF (见 README_toptwvl_silavwt in src/grid):

D. SFCFILES Section

该部分指定静态数据的路径, 这些静态的全球地理数据由下面 ftp 网址获取:

ftp://aftp.fsl.noaa.gov/divisions/frd-laps/WRFSI/Geog_Data

每个数据集必须有它自己的子目录!

1. TOPO_30S: 源自 USGS 的 30 秒地形高度数据的路径。

2. PCTLAND_10M: 10 分的土地分类资料的路径。

3. LANDUSE_30S: 30 秒 USGS 24 种土壤利用类型资料的路径。
4. SOILTYPE_TOP_30S: 30 秒 FAO 上层 16 种土壤类型资料的路径。
5. SOILTYPE_BOT_30S: 和(4)相同, 但是为底层。
6. GREENFRAC: 植被分类数据的路径。
7. SOILTEMP_1DEG: 1 度年平均深层土壤温度数据的路径。
8. ALBEDO_NCEP: 反照率的月气候数据的路径。(归一为局地天顶)。
9. SSTEMP: 未使用, 预留给气候的 SST 数据。

E. INTERP_CONTROL section

该部分控制输入的格点资料的水平和垂直插值。

1. NUM_ACTIVE_SUBNESTS: 如果 C 部分 hgridspec 中 NUM_DOMAINS > 1, (即在母区域 MOAD 中实行的嵌套网格), 该参数表示多少个子区域进行水平和垂直插值来产生初始条件。如果仅处理 MOAD, 则 NUM_ACTIVE_SUBNESTS 必须设为 0。 NUM_ACTIVE_SUBNESTS 设定 ACTIVE_SUBNESTS 参数的维数, 必须在 0 和 sets NUM_DOMAINS-1 (含)之间。
2. ACTIVE_SUBNESTS: 如果 NUM_ACTIVE_SUBNESTS >=1, 该参数为对应于 NUM_ACTIVE_SUBNESTS 参数的一系列整数, 每一个整数为预生成初始条件的子区域的 ID 号。每个数值互不相同, 且必须 >=2 以及 <=NUM_DOMAINS。当仅处理 MOAD 区时(NUM_ACTIVE_SUBNEST = 0), 该参数没有用。该参数与 NUM_ACTIVE_SUBNESTS 结合起来, 可以设定多个同级子区域, 但是在实际运行 WRF 的初始化时仅选取一个子区域来处理。
3. PTOPIA: 指定模式顶, 单位为 Pa。默认值为 5000Pa。
4. HINTERP_METHOD: 整数, 指定大气变量的插值方法。编号:

0: 最近邻方法 (不推荐使用)
1: 4 位双线性插值 (如果输入和输出数据的分辨率相同, 使用该选项)
2: 16 位
5. LSM_HINTERP_METHOD: 整数, 指定地貌数据场所使用的插值方法。编号与上面的相同。建议的默认值为 0 或 1。注意: 如果想使用背景资料的土地利用和土地分类, 该参数必须设为 0, 而且必须由输入数据集中获取"VEGCAT" 和 "SOILCAT", 其中 VEGCAT 为主要的土壤利用分类 (USGS 为 24 种) 的 2 维数组, SOILCAT 为 FAO 主要土壤类型的 2 维数组。

6. NUM_INIT_TIMES: 整数, 目前设为 1。控制输出时间数来使用由"INIT_ROOT" 和 "LSM_ROOT" 指定的前缀文件。将来可以支持分析"nudging"。在 1: NUM_INIT_TIMES 这段时间内, 使用由 INIT_ROOT/LSM_ROOT 指定的数据, 然后剩下的时段转而使用由"LBC_ROOT"指定的数据。如果设为 0, 所有的数据都来自 LBC_ROOT 和 CONSTANTS_FULL_NAME。设为 1 时, 与侧边界条件相比, 模式可以用不同来源的资料进行初始化得到初始场和陆面信息。

7. INIT_ROOT: 在 1:NUM_INIT_TIMES 时段内所使用数据的前缀。Wrfprep.pl 脚本会在 ANALPATH(见 SI_PATHS 部分) 中访问带有该前缀和相应时间后缀的文件。该参数仅当 NUM_INIT_TIMES > 0 时有用。

9. LBC_ROOT: 侧边界条件的时段所使用的数据文件的前缀。wrfprep.pl 脚本会连接"LBCPATH"目录下的所有带有该前缀和有效时间后缀的文件。

10. LSM_ROOT: 对于每一个 NUM_INIT_TIME (当 NUM_INIT_TIMES >0), wrfprep.pl 脚本会连接在 LSMPATH 下相应时间的带有该前缀的一个文件。该参数用来指定除 INIT_ROOT 以外的 NOAH LSM 方案中的输入数据。

11. CONSTANTS_FULL_NAME: 指定在"CONSTANTS_PATH"中搜寻的一系列文件名。搜寻到的文件中的数据将在每个输出时间中使用, 并且在 LSM_ROOT/INIT_ROOT/LBC_ROOT 文件中作备份。

12. VERBOSE_LOG: 逻辑型参数。设为 true 时在出现错误时提供更多的记录信息。

13. OUTPUT_COORD: 指定输出数据在何种垂直坐标上。有四种字符串选项:

'ETAP': 输入数据在由 LEVELS 参数指定的 eta 面上转化为 WRF 的质量形式的输出数据。

'NMMH': 输入数据在由 LEVELS 参数指定的 NMM 面上转化为 WRF 的 NCEP NMM 形式的输出数据。

'ZETA': 不再支持该坐标, 选此不能正常运行!

14. LEVELS: 按大气中的上升顺序的 WRF 模式中使用的垂直层次列表。如果 OUTPUT_COORD 设为 'ZETA', 这些数值从 0 变化到模式顶, 单位为米。如果 OUTPUT_COORD 为 "ETAP", 该参数由 1.0 变化到 0.0。

F. SI_PATHS Section

指定 grib_prep 输出数据文件的路径。多数情况下所有这些均设为同一路径 (\$EXT_DATAROOT/extprd)。

1. ANALPATH: 当 NUM_INIT_TIMES > 0 时文件名带有 INIT_ROOT 前缀的文件的的路径。
2. LBCPATH: 对于所有大于 NUM_INIT_TIMES 的时间段, 文件名带有 LBC_ROOT 前缀的文件的的路径。
3. LSMPATH: 对于所有 0:NUM_INIT_TIMES 的时间段内文件名带有 LSM_ROOT 前缀的文件所在路径。
4. CONSTANTS_PATH: 对于所有时间段文件名为 CONSTANTS_FULL_NAME 中所包括的文件的的路径。

附录 3:在 WRF 本体计算里, `namelist.input` 的设定最重要。这里记述了大部分的运行设置情报。这些参数变量的翻译是在 **tanghao** (动力论坛) 提供的 WRFV2.0 版本的 `namelist.input` 的翻译基础之上做了一些补充。同时也十分感谢 **windrisingdl** 作的补充工作。限于个人的精力, 全部的翻译工作没有完成。

`namelist` 中参数变量的取值

变量名	取值	描述
&time_control		时间控制
<code>run_days</code>	1	运行时间 (天)
<code>run_hours</code>	0	运行时间 (小时)
	注意: 如果模式积分时间大于 1 天, 则可同时设置 <code>run_days</code> 和 <code>_run_hours</code> , 也可设置 <code>run_hours</code> 一个参数。比如: 模式运行的总时间长度为 36 小时, 则可设置 <code>run_days=1</code> , 且 <code>run_hours=12</code> , 或者设置 <code>run_days=0</code> , 且 <code>run_hours=36</code> 。	
<code>run_minutes</code>	0	运行时间 (分钟)
<code>run_seconds</code>	0	运行时间 (秒)
<code>start_year(max_dom)</code>	2001	四位数字表示的起始年份
<code>start_month(max_dom)</code>	06	两位数字(01-12)表示的起始月份
<code>start_day(max_dom)</code>	11	两位数字(01-31)表示的起始日数
<code>start_hour(max_dom)</code>	12	两位数字(00-23)表示的起始小时数
<code>start_minute(max_dom)</code>	00	两位数字(00-59)表示的起始分钟数
<code>start_second (max_dom)</code>	00	两位数字(00-59)表示的起始秒数
<code>end_year(max_dom)</code>	2001	四位数字表示的终止年份
<code>end_month(max_dom)</code>	06	两位数字(01-12)表示的终止月份
<code>end_day(max_dom)</code>	12	两位数字(01-31)表示的终止日数
<code>end_hour(max_dom)</code>	12	两位数字(00-23)表示的终止小时数
<code>end_minute(max_dom)</code>	00	两位数字(00-59)表示的终止分钟数
<code>end_second (max_dom)</code>	00	两位数字(00-59)表示的终止秒数
	说明: 模式的积分是用起止时间设置来控制。 <code>real.exe</code> 的起止也是用起止时间参数来设定的。模式的积分时间可以用 <code>run_days</code> 、 <code>run_hours</code> 等来控制, 也可以用 <code>end_year</code> 、 <code>end_month</code> 等来控制。但前者 <code>run_days</code> 等优先于后者 <code>end_year</code> 等。而在 <code>real.exe</code> 中只用 <code>end_year</code> 等来控制结束时间信息。	
<code>interval_seconds</code>	21600	前处理程序的两次分析时间之间的时间间隔, 以秒为单位。也即模式的实时输入数据的时间间隔, 一般为输入边界条件的文件的时间间隔。
<code>input_from_file (max_dom)</code>	.true / .false	嵌套初始场输入选项。嵌套时, 指定嵌套网格是否用不同的初始场文件。
<code>fine_inpur_stream (max_dom)</code>	0 2	selected fields from nest input

history_interval(max_dom)	60	指定模式结果输出的时间间隔，以分钟为单位。
history_interval_mo(max_dom)	1	指定模式结果输出的时间间隔，以月数为单位。
history_interval_d(max_dom)	1	指定模式结果输出的时间间隔，以日数为单位。
history_interval_h(max_dom)	1	指定模式结果输出的时间间隔，以小时为单位。
history_interval_m(max_dom)	1	指定模式结果输出的时间间隔，以分钟为单位。
history_interval_s(max_dom)	1	指定模式结果输出的时间间隔，以秒数为单位。
frame_per_outfile(max_dom)	1	output times per history output file, used to split output files into smaller pieces.
restart	.true / .false	是否进行重行启动
restart_interval	1440	指定模式结果输出的时间间隔，以分钟为单位。
auxinput1_inname	"met_em.d<domain>_<date>" "wrf_real_input_em.d<domain>_<date>"	
io_form_history	2	2 = NetCDF
io_form_restart	2	指定模式断点重启输出的格式, 2 为 netCDF 格式
io_form_input	2	2 = NetCDF
io_form_boundary		指定模式边界条件数据的格式
	1	二进制格式
	2	NetCDF 格式
	4	PHD5 格式
	5	GRIB1 格式
debug_level	0	此选项指定模式运行时的调试信息输出等级。取值可为 0,50,100,200,300，数值越大，调试信息输出就越多，默认值为 0。
auxhist2_outname	"rainfall"	指定模式加密输出文件的文件名，缺省时取值为 "auxhist2_d_”。另外，需要指出的是，加密输出变量需要修改注册表文件 Registry.EM。
auxhist2_interval	10	此参数指定模式加密结果输出的时间间隔，以分钟为单位。
io_form_auxhist2	2	指定模式加密输出文件的格式, 2 为 NetCDF 格式。
nocolons	.true / .false	
write_input	.true / .false	输出的初始场格式写成与模式结果文件格式一致
inputout_interval	180	输出的初始场文件的时间间隔，以分钟为单位
input_outname	"wrf_3dv_input_d<domain>_<date>"	
inputout_begin_y	0	四位数字表示输出 3DVAR 数据开始年份。
inputout_begin_mo	0	两位数字表示输出 3DVAR 数据开始月份。
inputout_begin_d	0	两位数字表示输出 3DVAR 数据开始日期。
inputout_begin_h	3	两位数字表示输出 3DVAR 数据开始时次。
inputout_begin_m	0	两位数字表示输出 3DVAR 数据开始分钟数。
inputout_begin_s	0	两位数字表示输出 3DVAR 数据开始秒数。
inputout_end_y	0	四位数字表示输出 3DVAR 数据终止年份。

inputout_end_mo	0	两位数字表示输出 3DVAR 数据终止月份。
inputout_end_d	0	两位数字表示输出 3DVAR 数据终止日期。
inputout_end_h	12	两位数字表示输出 3DVAR 数据终止时次。
inputout_end_m	0	两位数字表示输出 3DVAR 数据终止分钟数。
inputout_end_s	0	两位数字表示输出 3DVAR 数据终止秒数。

以上示范设置表明输入格式数据是从 3 小时后开始输出，直到 12 小时，时间间隔是 180 分钟，即每 3 小时输出一次。

变量名	取值	描述
&domains		
time_step	60	区域定义：尺度、嵌套参数 积分的时间步长，为整型数，单位为秒，在真实大气中推荐值为 dx 公里数的 6 倍
time_step_fract_num	0	实数型时间步长的分子部分
time_step_fract_den	1	实数型时间步长的分母部分
说明：如果想以 60.3 秒作为积分时间步长，那么可以设置 time_step=60, time_step_fract_num=3, 并且设置 time_step_fract_den=10。其中 time_step 对应与时间步长的整数部分，time_step_fract_num/time_step_fract_den 对应于时间步长的小数部分。		
max_dom	1	计算区域个数。计算区域默认值为 1，如果使用嵌套功能，则 max_dom 大于 1。
s_we(max_dom)	1	x 方向(西-东方向)的起始格点值 (通常为 1)
e_we(max_dom)	91	x 方向(西-东方向)的终止格点值 (通常为 x 方向的格点数)
s_sn (max_dom)	1	y 方向(南-北方向)的起始格点值 (通常为 1)
e_sn (max_dom)	82	y 方向(南-北方向)的终止格点值 (通常为 y 方向的格点数)
s_vert (max_dom)	1	z 方向(垂直方向)的起始格点值
e_vert (max_dom)	28	z 方向(垂直方向)的终止格点值，即全垂直 eta 层的总层数。垂直层数在各嵌套网格中必须保持一致。
num_metgrid_levels	40	number of vertical levels of 3d meteorological fields coming from WPS metgrid program
eta_levels	1.0,0.99,...0.0	
force_sfc_in_vinterp	1	
p_top_requested	1	
lagrange_type	1	
lowest_lev_from_sfc	.true / .false	
dx (max_dom)	10000	指定 x 方向的格距（单位为米）。在真实大气方案中，此参数值必须与输入数据中的 x 方向格距一致。
dy (max_dom)	10000	指定 y 方向的格距（单位为米）。在真实大气方案中，此参数值必须与输入数据中的 y 方向格距一致。
ztop (max_dom)	19000	指定模式顶的高度。在质量坐标动力框架中，此高度值仅用于理想实验方案。

grid_id (max_dom)	1	计算区域的编号，一般是从 1 开始。
parent_id (max_dom)	0	嵌套网格的上一级网格（母网格）的编号，一般是从 0 开始。
i_parent_start (max_dom)	0	嵌套网格的左下角（LLC）在上一级网格（母网格）中 x 方向的起始位置
j_parent_start (max_dom)	0	嵌套网格的左下角（LLC）在上一级网格（母网格）中 y 方向的起始位置
parent_grid_ratio (max_dom)	1	嵌套时，母网格相对于嵌套网格的水平网格比例。在真实大气方案中，此比例必须为奇数；在理想大气方案中，如果将反馈选项 feedback 设置为 0 的话，则此比例也可以为偶数
parent_time_step_ratio (max_dom)	1	嵌套时，母网格相对于嵌套网格的时间步长比例。
feedback	1	嵌套时，嵌套网格向母网格得反馈作用。设置为 0 时，无反馈作用。而反馈作用也只有母网格和子网格的网格比例(parent_grid_ratio)为奇数时才起作用。
smooth_option		向上一级网格（母网格）反馈的平滑选项，只有设置了反馈选项为 1 时才起作用的。
	0	不平滑
	1	1-2-1 平滑
	2	smoothing-desmoothing
num_moves	2	移动嵌套网格总移动次数。
move_id	2	每一次移动嵌套网格区域编号列表。
move_interval	60,120	每一次移动的启动时间列表，单位为分钟，自模式积分起始时刻算起。
move_cd_x	1,-1	在 i 方向(即东西方向)每一次相对于父网格移动格点数。
move_cd_y	-1,1	在 j 方向(即南北方向)每一次相对于父网格移动格点数。
vortex_interval	15	经过多长时间计算一次涡旋的位置，单位为分钟
max_vortex_interval	40	涡旋的最大移动速度，用于计算新涡旋位置的搜索半径。
corral_dist	8	移动嵌套网格靠近粗网格边界允许的最大网格单元数，此参数也就是规定了移动网格靠近粗网格允许的最大距离。

变量名

取值

描述

&physics

物理方案

说明：虽然不同的嵌套网格可以使用不同的物理方案，但必须注意各种方案的使用条件和范围。

mp_physics (max_dom)

设置微物理过程方案，默认值为 0。

0

不采用微物理过程方案

1

Kessler 方案 (暖雨方案)

	2	Lin 等的方案 (水汽、雨、雪、云水、冰、冰雹)
	3	WSM 3 类简单冰方案
	4	WSM 5 类方案
	5	Ferrier(new Eta)微物理方案(水汽、云水)
	6	WSM 6 类冰雹方案
	8	新 Thompson 的冰雹方案
	98	NCEP 3 类简冰方案 (水汽、云/冰和雨/雪)
	99	NCEP 5 类方案(水汽、雨、雪、云水和冰) (将放弃)
mp_zero_out		选用微物理过程时, 保证 $Q_v \geq 0$, 以及当其他一些水汽变量小于临界值时, 将其设置为 0。
	0	表示不控制
	1	除了 Q_v 外, 所有的其他水汽变量当其小于临界值时, 则设置为 0
	2	确保 $Q_v \geq 0$, 并且所有的其他水汽变量当其小于临界值时, 则设置为 0。
mp_zero_out_thresh	1.e-8	水汽变量 (Q_v 除外) 的临界值, 低于此值时, 则设置为 0 (kg/kg)。
ra_lw_physics (max_dom)		此选项指定长波辐射方案, 默认值为 0。
	0	不采用长波辐射方案
	1	rrtm 方案
	99	GFDL (Eta) 长波方案 (semi-supported)
ra_sw_physics (max_dom)		此选项指定短波辐射方案, 默认值为 0。
	0	不采用短波辐射方案
	1	Dudhia 方案
	2	Goddard 短波方案
	99	GFDL (Eta) 短波方案 (semi-supported)
radt (max_dom)	30	此参数指定调用辐散物理方案的时间间隔, 默认值为 0, 单位为分钟。建议与 dx 的公里数取同样的值。
cam_abs_freq_s	21600	CAM clearsky longwave absorption calculation frequency (recommended minimum value to speed scheme up)
levsiz	59	for CAM radiation input ozone levels
paerlev	29	for CAM radiation input aerosol levels
cam_abs_dim1	4	for CAM absorption save array
cam_abs_dim2	same as e_vert	for CAM 2nd absorption save array
sf_sfclay_physics (max_dom)		此选项指定近地面层(surface-layer)方案, 默认值为 0。
	0	不采用近地面层方案
	1	Monin-Obukhov 方案
	2	MYJ Monin-Obukhov 方案 (仅用于 MYJ 边界层方案)
sf_surface_physics (max_dom)		此选项指定陆面过程方案, 默认值为 0。
	0	不采用陆面过程方案

	1	热量扩散方案
	2	Noah 陆面过程方案
	3	RUC 陆面过程方案
bl_pbl_physics (max_dom)		此选项指定边界层方案，默认值为 0
	0	不采用边界层方案
	1	YSU 方案
	2	Eta Mellor-Yamada-Janjic TKE(湍流动能) 方案
	3	NCEP Global Forecast System 方案
	99	MRF 方案（将放弃）
bldt (max_dom)	0	此参数指定调用边界层物理方案的时间间隔，默认值为 0，单位为分钟。此参数指定调用边界层物理方案的时间间隔，默认值为 0，单位为分钟。0 (推荐值)表示每一个时间步长都调用边界层物理方案。
cu_physics (max_dom)		此选项指定积云参数化方案，默认值为 0。
	0	不采用积云参数化方案
	1	浅对流 Kain-Fritsch (new Eta)方案
	2	Betts-Miller-Janjic 方案
	3	Grell-Devenyi 集合方案
	4	Simplified Arakawa-Schubert 方案
	99	老 Kain-Fritsch 方案
cudt (max_dom)	0	积云参数化方案的调用时间间隔，默认值为 0，单位为分钟。一般的积云参数化方案是每一步都要调用，但如果是用 Kain-Fritsch 方案(cu_physics=1)，则可以设 cudt=5。
isfflx	1	在选用扰动边界层和陆面物理过程时(sf_sfclay_physics = 1)是否考虑地面热量和水汽通量，默认值为 1。
	0	不考虑地面通量
	1	考虑地面通量
ifsnow	0	是否考虑雪盖效应。考虑雪盖效应时，必须要有雪盖输入场。默认值为 0，只有在利用扰动边界层 PBL 预报土壤温度时才有效，即 sf_surface_physics = 1。
	0	不考虑雪盖效应
	1	考虑雪盖效应
icloud	1	辐射光学厚度中是否考虑云的影响，默认值为 1。仅当 ra_sw_physics = 1 和 ra_lw_physics = 1 时有效。
	0	不考虑云的影响
	1	考虑云的影响
swrat_scatt	1	scattering tuning parameter (default 1. is 1.e-5 m2/kg)
surface_input_source		土地利用类型和土壤类型数据的来源格式，默认值为 1。
	1	SI/gridgen（由 SI 的 gridgen_model.exe 程序产生）
	2	其他模式产生的 GRIB 码数据(VEGCAT/SOILCAT 数据

		都在由 SI 产生的 wrf_real_input_em 文件中)
num_soil_layers		指定陆面模式中的土壤层数，默认值为 5
	5	热量扩散方案
	4	Noah 陆面过程方案
	6	RUC 陆面过程方案
maxiens	1	默认值为 1，仅用于积云参数化方案中的 Grell-Devenyi 集合方案
maxens	3	默认值为 3，仅用于积云参数化方案中的 Grell-Devenyi 集合方案
maxens2	3	默认值为 3，仅用于积云参数化方案中的 Grell-Devenyi 集合方案
maxens3	16	默认值为 16，仅用于积云参数化方案中的 Grell-Devenyi 集合方案
ensdim	144	默认值为 144，仅用于积云参数化方案中的 Grell-Devenyi 集合方案

以上这些用于 Grell-Devenyi 方案的默认值，都是推荐使用的数值，请谨慎修改。

seaice_threshold	271	海冰温度临界值。当 TSK 小于此临界值时，如果模式格点是水体，陆面过程选用 5 层的 SLAB 方案，则将此模式格点设置为陆地，且为永久性冰体；如果模式格点是水体，陆面过程选用 Noah 方案，则将此模式格点设置为陆地，且为永久性冰体，并将设置 0~3 米的 TEMPS，以及设置 SMOIS 和 SH2O。
sst_update	0	时变海温控制参数。0 表示不用，1 表示使用。如果选择使用时变海温，则 real.exe 会从 wrflowinp_d01 文件中读取 SST 和 VEGFRA 数据，wrf.exe 则会以更新边条件数据相同的时间间隔来更新这些数据。要使用此功能，则在参数列表文件 namelist.input 的时间控制区还必须包含 auxinput5_interval, auxinput5_end_h, 和 auxinput5_inname = "wrflowinp_d<domain>"。
	1	

变量名	取值	描述
&fdda		
grid_fdda (max_dom)	1	grid-nudging fdda on (=0 off) for each domain
gfdda_inname	"wrffdda_d<domain>"	

变量名	取值	描述
&dynamics		
		扩散、抑制、平流
dyn_opt	2	模式框架配置选项，欧拉质量坐标
rk_ord		Runge-Kutta 时间积分方案阶数，默认值为 3

	2	Runge-Kutta 二阶
	3	Runge-Kutta 三阶 (推荐)
diff_opt		湍流和混合作用选项, 默认值为 0
	0	没有湍流或者显式空间数值滤波(km_opt 将被忽略)
	1	老扩散方案, 计算坐标面上二阶扩散项。如果没有指定 PBL 选项, 则用 kvdif 选项当作垂直扩散系数。通常用于 km_opt=1 或者 4。(在真实大气方案的水平分辨率格距小于 10km 时, 推荐使用 1)
	2	新扩散方案, 计算物理空间(x,y,z)中的混合作用项(应力形式)。用 km_opt 来指明湍流参数化过程。
km_opt		湍涡系数选项, 默认值为 1
	1	固定不变 (用参数配置第三部分的 khdif, kvdif 参数值)。与 diff_opt=1 的区别在于 km_opt 的水平扩散作用不在模式的 zeta 面上。因此, 只有在没有地形的情况下, 这两个选项的作用才是相同的。
	2	1.5 阶 TKE(湍流动能)闭合 (3D)
	3	Smagorinsky 一阶闭合。2 和 3 在水平格距大于 2km 时, 不推荐使用。
	4	水平 Smagorinsky 一阶闭合。4 在水平格距小于 10km 时, 推荐使用。
diff_6th_opt	0	6th-order numerical diffusion
	0	no 6th-order diffusion (default)
	1	6th-order numerical diffusion
	2	6th-order numerical diffusion but prohibit up-gradient diffusion
diff_7th_factor	0.12	6th-order numerical diffusion non-dimensional rate (max value 1.0 corresponds to complete removal of 2dx wave in one timestep)
damp_opt		顶层抽吸作用标志选项 (当 diff_opt = 1 时, 此选项失效), 默认值为。同时, 必需在参数配置第三部分设置 zdamp 和 dampcoef 参数。
	0	无抽吸作用
	1	有抽吸作用
	2	with Rayleigh damping (dampcoef inverse time scale [1/s] e.g. .003; not for real-data cases)
w_damping		垂直速度拟制标志选项 (用于实际业务) 默认值为 0。
	0	无抑制作用
	1	有抑制作用
zdamp (max_dom)	5000	此参数设定模式顶部的抽吸厚度。推荐值为 5000 米, (仅用于理想大气)
dampcoef (max_dom)	0	指定抽吸系数(仅用于理想大气)
base_temp	290	基本海平面温度 (仅用于真实大气、质量坐标)

base_pres	100000	基本海平面气压，请不要改变推荐值
base_lapse	50	基本温度垂直递减率（仅用于真实大气、质量坐标），请不要改变推荐值
khdif (max_dom)	0	水平扩散系数(单位为 m^2/s)，默认值为 0。使用此参数时，必须设定选项 <code>diff_opt = 1</code> 或者 <code>km_opt = 1</code> 。
kvdif (max_dom)	0	此参数设定垂直扩散系数(单位为 m^2/s)，默认值为 0。使用此参数时，必须设定选项 <code>diff_opt = 1</code> 或者 <code>km_opt = 1</code> 。
smdiv(max_dom)	0.1	辐散抽吸(系数) (通常取为 0.1)，默认值为 0。此参数在时间分裂 RK 方案中用于选择性地消除声波。
emdiv (max_dom)	0.01	额外模态滤波系数，默认值为 0.01。（仅用于真实大气、质量坐标）
epssm (max_dom)	0.1	此参数指定垂直声波的离心时间(time off-centering)，默认值为 0.1。
non_hydrostatic(max_dom)	.true.	模式动力框架参数，指定模式动力框架是否是非静力模式，.true.为非静力，.false.为静力，默认为.False。
pert_coriolis (max_dom)	.false	科氏参数，仅影响扰动风场（适用于理想方案）
mix_full_fields	.false	与 <code>diff_opt = 2</code> 配合使用。除高分辨率的理想模拟外，推荐取值为".true."，但 <code>damp_opt</code> 不能同时为 1。当取".false."时，表示混合前扣除 1 维的静态廓线(base-state profile)。
h_mom_adv_order (max_dom)	5	此选项指定水平动量平流的阶数，默认值为 3。(例如 5=5 阶，等等)，有效值为 2~6，推荐值为 5。
v_mom_adv_order (max_dom)	3	此选项指定垂直动量平流的阶数，默认值为 3。有效值为 2~6，推荐值为 3。
h_sca_adv_order (max_dom)	5	此选项指定水平标量(scalar)平流的阶数，默认值为 3。有效值为 2~6，推荐值为 5。
v_sca_adv_order (max_dom)	3	此选项指定垂直标量平流的阶数，默认值为 3。有效值为 2~6，推荐值为 3。
time_step_sound (max_dom)	4	每一时间步长中声波的步数(sound steps)。通常为 4，默认值为 10。如果时间步长远大于 $6 \times dx$ （公里），则需增加声波步数。
pd_moist	.false	positive definite advection of moisture
pd_scalar	.false	positive definite advection of scalars
pd_tke	.false	positive definite advection of chem variables
pd_chem	.false	positive definite advection of tke
tke_drag_coefficient (max_dom)	0	surface drag coefficient (Cd, dimensionless) for <code>diff_opt=2</code> only
tke_heat_flux (max_dom)	0	surface thermal flux ($H/(\rho \cdot c_p)$, K m/s) for <code>diff_opt=2</code> only

变量名	取值	描述
&bdy_control		边界条件控制
spec_bdy_width	5	边界过渡的格点总行数，默认值为 5。此参数只用于真实大气方案。参数的大小至少为 spec_zone 和 relax_zone 的和。
spec_zone	1	指定区域(specified zone)的格点数，默认值为 1。指定边条件时起作用，此参数只用于真实大气方案。
relax_zone	4	指定松弛区域的格点数，默认值为 4。指定边条件时起作用，此参数只用于真实大气方案。
specified (max_dom)	.false.	是否使用特定边条件，逻辑型，默认值为 .false.。特定边条件选项只用于真实大气方案的数值模拟中，并且要求多个时次的边条件数据(文件 wrfbdy)。
periodic_x (max_dom)	.false.	在 x 方向是否使用周期性边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
symmetric_xs (max_dom)	.false.	在 x 方向的起始点(西边界)是否使用对称性边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
symmetric_xe (max_dom)	.false.	在 x 方向的终止点(东边界)是否使用对称性边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
open_xs (max_dom)	.false.	在 x 方向的起始点(西边界)是否使用自由边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
open_xe (max_dom)	.false.	在 x 方向的终止点(东边界)是否使用自由边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
periodic_y (max_dom)	.false.	在 y 方向是否使用周期性边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
symmetric_ys (max_dom)	.false.	在 y 方向的起始点(南边界)是否使用对称性边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
symmetric_ye (max_dom)	.false.	在 y 方向的终止点(北边界)是否使用对称性边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
open_ys (max_dom)	.false.	在 y 方向的起始点(南边界)是否使用自由边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
open_ye (max_dom)	.false.	在 y 方向的终止点(北边界)是否使用自由边界条件。逻辑型，默认值为 .false.。通常只用于理想大气试验方案。
nested (max_dom)	.false.	此选项设定嵌套边界条件。逻辑型，默认值为 .false.。

变量名	取值	描述
&namelist_quilt		参数列表的这一部分用于控制 MPI 异步通讯形式的输入/输出。
nio_tasks_per_group		此参数指定模式需要多少个 I/O 处理器：
	0	不要求单独的 I/O 处理器。
	n	如果 n>0, 表明需要 n 个 I/O 处理器。
nio_groups	1	设置为 1，目前为预留参数，请勿改动。

tile_sz_x	0	在共享式内存进程中指定 x 方向计算的格点数，默认值为 0。如果指定了 numtiles，则不需要此参数。
tile_sz_y	0	在共享式内存进程中指定 y 方向计算的格点数，默认值为 0。如果指定了 numtiles，则不需要此参数。
numtiles	1	此参数在共享式内存进程中指定每个内存块中的内存片数，默认值为 1。(或者是指定上面 tile_sz_x 和 tile_sz_y 两个参数)
nproc_x	-1	区域分解时，指定 x 方向上的上的线程数，默认值为 -1。
nproc_y	-1	区域分解时，指定 x 方向上的上的线程数，默认值为 -1。
	-1	程序自动分解
	>1	用于分解的数目。

附录 4: 一些简单的 UNIX 命令:

① cd 命令

`cd directory` 改变工作目录

② ls 命令

`-a` 显示目录下所有子目录与文件(包括隐藏文件)

`-l` 显示文件的详细信息

③ cp 命令

➤ `cp -r default OnlineTut`

`-r` 递归复制源目录下所有的子目录和文件

④ chmod 命令

➤ `chmod -R u+w OnlineTut`

`chmod` : `chang mode` 改变文件或目录的访问权限

`-R` 以递归的方式逐个对当前目录下的所有档案与子目录进行相同的权限变更

`+` 增加权限

`-` 取消权限

`u` 表示该档案的拥有者

`w` 表示可写入权

⑤ vi 命令

➤ `vi file_name` 开始编辑或者创建一个文件

编辑命令: `<Esc>` 模式切换

`x` 删除光标所在文字

`dd` 删除光标所在行

`a` 在光标后新增文字

`i` 在光标前新增文字

`o` 在光标下方新增一行

`O` 在光标上方新增一行

`:wq` 以原档案名保存并退出

`:q!` 不保存文档强制退出

⑥ rm 命令

`rm file_name` 删除文件

`rm -r deirectory_name` 递归删除全部目录和子目录

⑦ mkdir 命令

`mkdir directory_name` 创建新目录

⑧ rmdir 命令

`rmdir directory_name` 删除空目录

⑨ unzip 命令

➤ `gunzip xxxxx.tar.gz` 解压缩。

⑩ tar 命令

- **tar -xvf xxxxx.tar** 解压文件
- x 从档案文件中释放文件
- v 详细报告 tar 处理的文件信息
- f 使用档案文件或设备，这个选项通常是必选的

可以和上面的⑨合写成 tar xvfz xxxxx.tar.gz

⑪ clear 命令

清除屏幕上的信息

⑫ pwd 命令

- **pwd** 显示出当前工作目录的绝对路径

⑬ <Ctrl> + C

强制放弃正在执行的任务

⑭ exit 命令

退出 UNIX 系统（包括退出 SSH）