



INSTITUTO DE CONTROLE DO ESPAÇO AÉREO

MANUAL DE INSTALAÇÃO, COMPILAÇÃO E EXECUÇÃO DO MODELO DE MESOESCALA WRF NO ICEA (VERSÃO 3.4.1)

Miguel Ângelo Vargas de Carvalho (PBCA/ICEA)

Milton Luiz Abrunhosa (FSDTP/ICEA)

Jonas Ricardo Oliveira Lopes (PBCA/ICEA)

Projeto de pesquisa coordenado pelo Grupo de Trabalho do Programa de Modelagem
Numérica de Tempo (PMNT)

São José dos Campos – SP
PBCA/ICEA
2013

SUMÁRIO

	Pág.
1 INTRODUÇÃO.....	5
1.1 Objetivo	6
2 DESCRIÇÃO DO MODELO WRF	7
3 INSTALAÇÃO, COMPILAÇÃO E EXECUÇÃO DOS SUBSISTEMAS DO WRF	9
3.1 Preparação inicial.....	9
3.2 Bibliotecas opcionais para o GRIB2.....	10
3.2 Instalação do NetCDF.....	11
3.3 Instalação do WRF.....	12
3.4.1 Instalação do subsistema WRFV3	12
3.4.2 Instalação do subsistema WPS.....	14
3.4.3 Instalação do subsistema ARWpost.....	16
3.5 Dados geográficos e de terreno.....	17
3.6 Instalação do pacote gráfico GrADS	17
4 CONFIGURAÇÕES INICIAIS DO WRF	19
5 EXECUTANDO O WRF	23
5.1. Execução do WPS.....	23
5.1.1 Execução GEOGRID	24
5.1.2 Execução UNGRIB	25
5.1.3 Execução METGRID	27
5.2 Execução do Real e WRF.....	28
5.3 Execução do ARWpost	30
6 CONSIDERAÇÕES FINAIS.....	33
7 AGRADECIMENTOS.....	34
REFERÊNCIAS BIBLIOGRÁFICAS	35
APÊNDICE A – NAMELISTS GRADE SUDESTE.....	37
APÊNDICE B – NAMELISTS GRADE NORDESTE.....	41
APÊNDICE C – NAMELISTS GRADE NORTE.....	45
APÊNDICE D – NAMELISTS GRADE ALCÂNTARA.....	49

1 INTRODUÇÃO

Para o Comando da Aeronáutica, a previsão de fenômenos meteorológicos é importante por causa do impacto nas atividades aeronáuticas e aeroespaciais, tais como gerenciamento do fluxo de tráfego aéreo e lançamento de foguetes. No Brasil, o responsável pelas informações e prognósticos meteorológicos para a aviação é o Departamento de Controle do Espaço Aéreo (DECEA). Informações e prognósticos meteorológicos confiáveis são ferramentas essenciais que possibilitam a segurança das operações aéreas, o conforto dos passageiros e estabelecimento de rotas mais rápidas e econômicas para as aeronaves. Por intermédio da Portaria DECEA 009/SDAD, de 06 de outubro de 2008, e reestruturado pela Portaria DECEA 033/SDOP, de 12 de março de 2013, foi constituído o Grupo de Trabalho (GT) do Programa de Modelagem Numérica do Tempo (PMNT). Os participantes do GT são oriundos de diferentes organizações do Sistema de Controle do Espaço Aéreo Brasileiro (SISCEAB), da especialidade de Meteorologia, na sua maioria com curso de pós-graduação (nível Doutorado e Mestrado), ou com conhecimentos específicos da área da Tecnologia da Informação (TI). O PMNT visa proporcionar prognósticos em alta resolução para áreas de maior fluxo da navegação aérea nacional, o que possibilita um aumento do detalhamento das informações meteorológicas e da qualidade da previsão do tempo para o SISCEAB. Em 2002, foram iniciadas, tanto para fins operacionais, quanto para pesquisa, as primeiras simulações com modelos de Previsão Numérica de Tempo (PNT) na Força Aérea Brasileira (FAB). O modelo de PNT inicialmente escolhido foi a 5ª Geração do Modelo de Mesoescala (MM5). Atualmente, o MM5 tem sido usado pelas seguintes instituições: Instituto de Aeronáutica e Espaço (IAE), Instituto de Controle do Espaço Aéreo (ICEA) e Centro Nacional de Meteorologia Aeronáutica (CNMA). Os prognósticos gerados pelo MM5 estão disponibilizados na homepage do ICEA e da Rede de Meteorologia do Comando da Aeronáutica (REDEMET). Com o contínuo aumento na capacidade de processamento dos computadores e de conectividade, tornou-se possível o emprego de modelos numéricos de previsão de tempo cada vez mais sofisticados. Assim, em 2011, iniciou-se a instalação do *Weather Research and Forecasting* (WRF) no ICEA. O WRF é a última geração de modelo numérico de previsão do tempo que servirá tanto para a operacionalidade dos centros meteorológicos como para as pesquisas atmosféricas. Com a sua implementação o ICEA consegue estar inserido no crescimento natural das pesquisas na área de previsão operacional do tempo.

1.1 Objetivo

O objetivo deste trabalho é descrever as etapas de instalação, compilação e execução da versão 3.4.1 do WRF, sob uma plataforma Linux de 64 bits, aperfeiçoando o projeto iniciado no ICEA em parceria com o IAE e o CNMA (IRIART, CARVALHO e PEREIRA NETO, 2011). Associado com a aquisição de novos recursos computacionais e de conectividade, a instalação dessa nova versão possibilitou melhorias no sistema de modelagem numérica instalado no ICEA, destacando-se: o aumento da resolução espacial (18 e 6 km), o redimensionamento dos domínios e aumento no número de rodadas do modelo, ou seja, a sua inicialização 4 vezes ao dia (00Z, 12Z, 18Z e 00Z).

2 DESCRIÇÃO DO MODELO WRF

O WRF é a última geração de modelo numérico de previsão do tempo que servirá tanto para a operacionalidade dos centros meteorológicos como para as pesquisas atmosféricas. Ele foi desenvolvido através de uma parceria entre diversos órgãos, entre eles o *National Center for Atmospheric Research* (NCAR), *National Oceanic and Atmospheric Administration* (NOAA), *National Center for Environmental Prediction* (NCEP), *Forecast Systems Laboratory* (FSL), *Air Force Weather Agency* (FAWA), e outros órgãos de pesquisa e desenvolvimento. Maiores informações sobre o WRF podem ser obtidas no sítio <http://www.wrf-model.org>.

As características do modelo que mais se destacam são: os múltiplos núcleos dinâmicos, sistema variável de assimilação de dados tri-dimensional e uma estrutura de software que permite o paralelismo computacional, bem como a extensibilidade do sistema. Pode ser instalado em diversas plataformas computacionais (p. ex. Linux), é muito portátil, flexível, de domínio público e disponibilizado gratuitamente pela internet. O modelo pode ser executado tanto para situações atmosféricas idealizadas como situações reais, em um espectro amplo de aplicações em escalas horizontais que variam de milhares de quilômetros a poucos metros.

O sistema de modelagem do WRF compreende diversos componentes (Fig. 1; para maiores detalhes, veja *ARW V3 Modeling System User's Guide*, 2011). Os principais são: no pré processamento o WRF Preprocessing System (WPS); inicialização do WRF (real); execução do WRF; e pós-processamento (ARWpost). O WPS é um conjunto de três subsistemas que prepara os dados de entrada para a simulação: geogrid, ungrrib e metgrib. O geogrid define o domínio do modelo e interpola os dados terrestres para as grades. O ungrrib extrai os campos meteorológicos do formato GRIB (V1 e V2) necessários para iniciar o modelo. O metgrib interpola horizontalmente os campos meteorológicos extraídos pelo ungrrib para as grades do modelo definida pelo geogrid. O trabalho de interpolar verticalmente os campos do WRF é feito dentro do programa real. O WRF é responsável pela execução do modelo propriamente dito. Por fim, o ARWpost é usado para converter as saídas do modelos para um formato que possam ser visualizados por aplicativos gráficos.

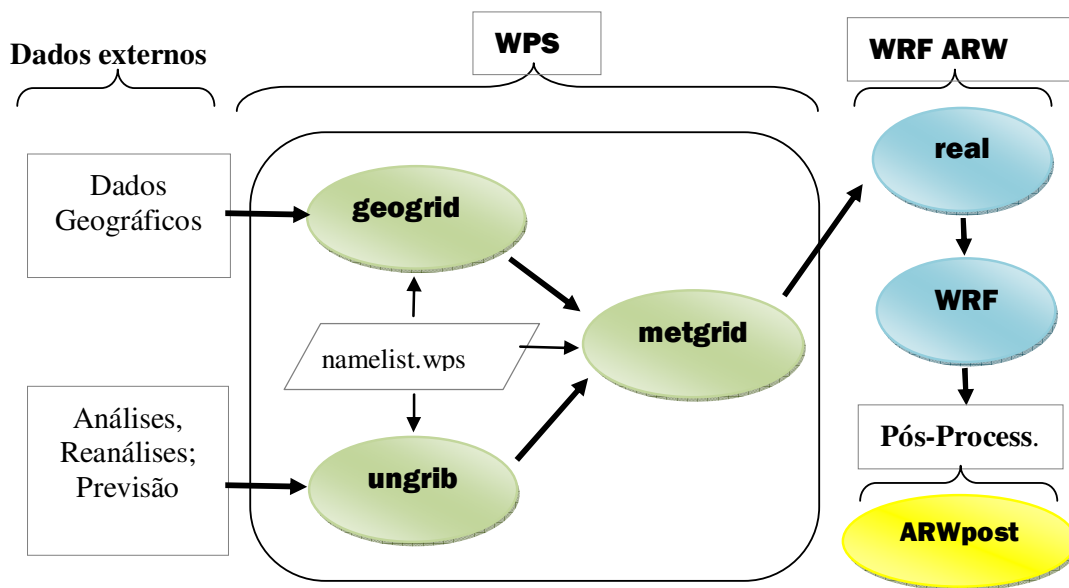


Figura 1 – Representação esquemática do sistema de modelagem. Adaptado de ARW V3 Modeling System User's Guide, 2011.

3. INSTALAÇÃO, COMPILAÇÃO E EXECUÇÃO DOS SUBSISTEMAS DO WRF

O modelo WRF versão 3.4.1 foi instalado sob o sistema operacional livre Debian¹ versão 7.0 “Wheezy” de 64 bits, em máquinas multicore, que possibilita diversos fluxos de execução simultâneos. O WRF foi desenvolvido de maneira a explorar o uso do paralelismo computacional.

Os principais componentes do *hardware* são: servidores Dell PowerEdge R720 com 01 (um) processador 6-Core, 12-threads com clock de 2.0 GHz com 64 (sessenta e quatro) GB de memória RAM; uma unidade de disco rígido interno de 300 GB; e memória cache de 256 MB. Cada servidor simulará um domínio. Maiores detalhes da configuração dos domínios serão fornecidos mais adiante.

Para que o WRF funcione corretamente, é necessária a instalação de diversos componentes de software (bibliotecas, compiladores, linguagens de programação, etc.), bem como o NetCDF e os arquivos de instalação dos subsistemas do modelo. Alguns aplicativos estão disponíveis no repositório de *software* do Debian, outros em sítios da *internet*. Para procurar, instalar ou manipular pacotes de *software* a partir dos repositórios do Debian utiliza-se a ferramenta *Advanced Packaging Tool* (APT). Os arquivos da *internet* podem ser baixados diretamente para um diretório a sua escolha (neste trabalho, `/usr/local/src/`). Os sítios necessários para instalação do modelo serão informados neste trabalho. A seguir, será apresentada uma descrição detalhada de cada etapa da instalação do modelo.

3.1 Preparação Inicial

Antes da instalação dos subsistemas do WRF, é necessário a instalação ou atualização de diversos componentes (bibliotecas, compiladores C/C++ e Fortran, linguagens de programação, etc.). Com a ferramenta APT, são instalados os seguintes componentes: `gfortran`, `gcc`, `g++`, `cpp`, `flex`, `csh`, `libncurses5-dev`, `bison`, `curl`, `m4`, `perl`, `make`, `zip`. Para a execução do modelo em paralelo, ainda são instalados: `openmpi-bin`, `libopenmpi-dev`. Todos devem ser instalados antes da instalação dos subsistemas do WRF. Os comandos no terminal do Linux são executados como “root” do sistema.

¹ www.debian.org/releases/wheezy/index.en.html

```
# apt-get update
# apt-get install gfortran gcc g++ cpp flex csh bison curl m4 perl make zip libncurses5-
dev openmpi-bin libopenmpi-dev
```

3.2 Bibliotecas opcionais para GRIB2

As condições iniciais e de contorno utilizadas no modelo estão no formato da segunda versão do código *General Regularly-distributed Information in Binary* (GRIB2). O código GRIB é um formato de arquivo padrão da *World Meteorological Organization* (WMO) para o armazenamento de campos regularmente distribuídos (p.ex., pontos de grade). Para a extração dos campos meteorológicos desses arquivos pelo WRF, é necessária a instalação de três bibliotecas: JasPer 1.900.1², Zlib 1.2.5³ e PNG 1.4.7⁴. As bibliotecas são, então, copiadas e descompactadas no diretório /usr/local.

```
# cp -f /home/webpca/WRF/install/jasPer-1.900.1.zip libpng-1.4.7.tar.gz zlib-
1.2.5.tar.gz /usr/local/
# cd /usr/local
# unzip jasPer-1.900.1.zip
# tar -vzxf libpng-1.4.7.tar.gz
# tar -vzxf zlib-1.2.5.tar.gz
```

Concluída a descompactação dos arquivos, deve-se entrar nos três diretórios criados (Fig.2, zlib-1.2.5, jasPer-1.900.1 e libpng-1.4.7) e compilar as respectivas bibliotecas:

```
# cd zlib-1.2.5
# ./configure --prefix=/usr/local
# make
# make intall
# cd ../jasper-1.900.1
# ./configure --prefix=/usr/local
# make
# make intall
# cd ../libpng-1.4.7
```

² www.ece.uvic.ca/~mdadams/jasper/

³ www.zlib.net

⁴ www.libpng.org/pub/png/libpng.html

```
# ./configure --prefix=/usr/local
# make check
# make intall
# cd ..
```

O *script* de instalação do WPS possui versões diferentes dessas bibliotecas. Mais adiante, será mostrado como alterar a versão das bibliotecas que estão localizadas dentro do arquivo `configure.wps`.

3.3 Instalação do NetCDF

A versão do NetCDF instalada é a 4.1.3 (não compilada), disponibilizada por *Unidata Program Center*⁵ (“NetCDF C/C++/Fortran Stable Releases”). Após o *download*, o arquivo é copiado e descompactado para o diretório `/usr/local`:

```
# cp -f /home/webpca/WRF/install/netcdf-4.1.3.tar.gz /usr/local/
# cd /usr/local/
# tar -vzxf netcdf-4.1.3.tar.gz
```

Entre no diretório criado (`netcdf-4.1.3`), configure e compile o NetCDF. Uma mensagem, no final da instalação, informará o êxito da instalação.

```
# cd netcdf-4.1.3
# ./configure --prefix=/usr/local/netcdf
# make
# make check
# make install
      “Congratulations! You have successfully installed netCDF”
# cd ..
```

Se ocorrerem erros, informações adicionais podem ser obtidas no Guia de Instalação do NetCDF⁶.

⁵ www.unidata.ucar.edu/downloads/netcdf/index.jsp

⁶ www.unidata.ucar.edu/software/netcdf/docs/netcdf-install

3.4 Instalação do WRF

Os arquivos necessários para instalação dos subsistemas do WRF estão disponibilizados no sítio *University Corporation for Atmospheric Research (UCAR)*⁷. O local escolhido para instalação do modelo é o diretório `/home/webpca/WRF`. Inicialmente, são baixados e transferidos para o novo diretório os arquivos de instalação do WRF (versão 3.4.1), WPS (versão 3.4.1) e ARWPost (versão 3.1), bem como os dados geográficos e de terreno para a montagem dos domínios.

```
# mkdir /home/webpca/WRF
# cp -f /home/webpca/WRF/install/geog_v3.1.tar.gz WPSV3.4.1.TAR WRFV3.4.1.TAR
ARWpost.tar.gz /home/webpca/WRF
# cd /home/webpca/WRF
```

Para todos os subsistemas do modelo, os procedimentos para instalação são semelhantes. Após descompactar o arquivo, o NetCDF é definido como variável ambiental e para que ele possa suportar arquivos grandes (>2GB). O próximo passo é realizar a configuração e compilação dos subsistemas. A sequência de comandos utilizados para a instalação de cada subsistema do WRF é mostrada abaixo:

3.4.1 Instalação do subsistema WRFV3

```
# tar -vxf WRFV3.4.1.TAR
# cd WRFV3
# export NETCDF=/usr/local/netcdf          ! (setar a variável de ambiente)
# export WRFIO_NCD_LARGE_FILE_SUPPORT=1    ! (arquivos grandes)
# ./configure                             ! (escolher a opção dentre as oferecidas, Fig. 2)
# ./compile em_real >& compile.log
# cd ..
```

O *script* `configure` cria o arquivo de configuração do WRF. Sua função é checar os recursos computacionais (*hardware* e *software*) e oferecer aos usuários opções para a configuração do WRF. O arquivo criado chama-se `configure.wrf`. Este novo arquivo também pode ser editado com outras opções, caso seja necessário. Para a compilação do

⁷ http://www.mmm.ucar.edu/wrf/users/download/get_source.html

WRF, utiliza-se o *script* compile. Se for executado com sucesso, os seguintes executáveis são criados no diretório /main: ndown.exe, tc.exe, nup.exe, real.exe e wrf.exe. Caso um ou mais destes arquivos não tenham sido criados, procurar erros dentro do arquivo compile.log, corrigi-los e repetir a seqüência de comando deste tópico. Antes de recompilar, execute o *script* clean (./clean -a). A Figura 2 mostra as opções escolhidas para a instalação do modelo nos servidores do ICEA e a Figura 3, os arquivos criados na pasta WRFV3.

```

-----
Please select from among the following supported platforms.
1. Linux i486 i586 i686, gfortran compiler with gcc (serial)
2. Linux i486 i586 i686, gfortran compiler with gcc (smpar)
3. Linux i486 i586 i686, gfortran compiler with gcc (dmpar)
4. Linux i486 i586 i686, gfortran compiler with gcc (dm+sm)
5. Linux i486 i586 i686, g95 compiler with gcc (serial)
6. Linux i486 i586 i686, g95 compiler with gcc (dmpar)
7. Linux i486 i586 i686, PGI compiler with gcc (serial)
8. Linux i486 i586 i686, PGI compiler with gcc (smpar)
9. Linux i486 i586 i686, PGI compiler with gcc (dmpar)
10. Linux i486 i586 i686, PGI compiler with gcc (dm+sm)
11. Linux x86_64 i486 i586 i686, ifort compiler with icc (serial)
12. Linux x86_64 i486 i586 i686, ifort compiler with icc (smpar)
13. Linux x86_64 i486 i586 i686, ifort compiler with icc (dmpar)
14. Linux x86_64 i486 i586 i686, ifort compiler with icc (dm+sm)
15. Linux i486 i586 i686 x86_64, PathScale compiler with pathcc (serial)
16. Linux i486 i586 i686 x86_64, PathScale compiler with pathcc (dmpar)

Enter selection [1-16]: 2
-----
Compile for nesting? (0=no nesting, 1=basic, 2=preset moves, 3=vortex following)
[default 0]: 1

```

Figura 2 – Figura ilustrativa do terminal mostrando as opções escolhidas para a compilação do WRF. Os retângulos vermelhos mostram as opções selecionadas.

```

webpca@pesquisa:/home/WRF/WRFV3/main$ ls
convert_em.F          module_wrf_top.o    nup.exe             tc_em.o
ideal.F              ndown_em.F         real_em.F          tc.exe
libwrf.lib.a         ndown_em.f90      real_em.f90       wrf.exe
Makefile             ndown_em.o        real_em.o         wrf.F
module_initialize_real.mod ndown_em.exe     real_em.F         wrf.f90
module_wrf_top.F     nup_em.F          tc_em.F           wrf.o
module_wrf_top.f90   nup_em.f90       tc_em.f90        wrf_SST_ESMF.F
module_wrf_top.mod   nup_em.o
webpca@pesquisa:/home/WRF/WRFV3/main$

```

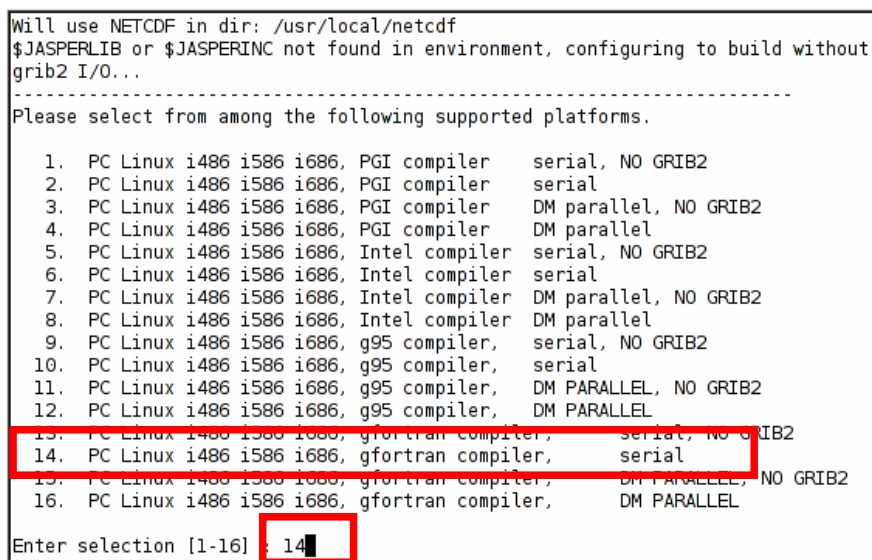
Figura 3 – Figura ilustrativa do terminal mostrando a pasta /home/webpca/WRF/WRFV3/main. Os retângulos vermelhos mostram os executáveis criados após a compilação do WRF.

Obs.: A compilação do WRF leva algum tempo, pois todo o código é compilado. Caso não ocorram alterações, a compilação é realizada somente uma vez.

3.4.2 Instalação do subsistema WPS:

```
# tar -vxf WPSV3.4.1.TAR
# cd WPS
# export NETCDF=/usr/local/netcdf
# export WRFIO_NCD_LARGE_FILE_SUPPORT=1
# ./configure ! (escolher a opção dentre as oferecidas, Fig. 4)
# ./compile >& compile.log
# cd ..
```

Como informado anteriormente, a seqüência de comandos utilizada para configuração e compilação do WPS é semelhante à mostrada para o WRF. Agora, o arquivo de configuração criado é o `configure.wps` e, após a compilação do WPS, os seguintes executáveis são criados nos diretórios: `geogrid.exe` (define o tamanho e localização do domínio) em `geogrid/src`; `ungrib.exe` (extrai os campos meteorológicos dos arquivos GRIB) em `ungrib/src`; e `metgrid.exe` (interpola horizontalmente os campos meteorológicos para a grade simulada definida) em `metgrid/src`. Os atalhos desses executáveis são criados no diretório `/WPS`. Caso um ou mais desses arquivos não tenham sido criados ou o nome deles estiver piscando, procurar erros dentro do arquivo `compile.log`, corrigi-los e repetir a seqüência de comandos deste tópico. Antes de recompilar, execute o *script* `clean` (`./clean -a`). A Figura 4 mostra as opções escolhidas para a instalação do modelo nos servidores do ICEA e a Figura 5, os atalhos criados no diretório `/WPS`.



```
Will use NETCDF in dir: /usr/local/netcdf
$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 i486 i586 i686, PGI compiler      serial, NO GRIB2
  2. PC Linux i486 i586 i686, PGI compiler      serial
  3. PC Linux i486 i586 i686, PGI compiler      DM parallel, NO GRIB2
  4. PC Linux i486 i586 i686, PGI compiler      DM parallel
  5. PC Linux i486 i586 i686, Intel compiler    serial, NO GRIB2
  6. PC Linux i486 i586 i686, Intel compiler    serial
  7. PC Linux i486 i586 i686, Intel compiler    DM parallel, NO GRIB2
  8. PC Linux i486 i586 i686, Intel compiler    DM parallel
  9. PC Linux i486 i586 i686, g95 compiler,     serial, NO GRIB2
 10. PC Linux i486 i586 i686, g95 compiler,     serial
 11. PC Linux i486 i586 i686, g95 compiler,     DM PARALLEL, NO GRIB2
 12. PC Linux i486 i586 i686, g95 compiler,     DM PARALLEL
 13. PC Linux i486 i586 i686, gfortran compiler, serial, NO GRIB2
 14. PC Linux i486 i586 i686, gfortran compiler, serial
 15. PC Linux i486 i586 i686, gfortran compiler, DM PARALLEL, NO GRIB2
 16. PC Linux i486 i586 i686, gfortran compiler, DM PARALLEL

Enter selection [1-16] : 14
```

Figura 4 – Figura ilustrativa do terminal mostrando as opções escolhidas para a compilação do WPS. Os retângulos vermelhos mostram as opções selecionadas.

Obs.: A execução do WPS ocorre em série, pois o ganho da execução deste subsistema em paralelo é muito pequeno.

```
FILE: 2010-03-22_18 met_em.d02.2010-03-21_12:00:00.nc
FILE: 2010-03-23_00 met_em.d02.2010-03-21_18:00:00.nc
FILE: 2010-03-23_06 met_em.d02.2010-03-22_00:00:00.nc
FILE: 2010-03-23_12 met_em.d02.2010-03-22_06:00:00.nc
geo_em.d01.nc met_em.d02.2010-03-22_12:00:00.nc
geo_em.d02.nc met_em.d02.2010-03-22_18:00:00.nc
geogrid met_em.d02.2010-03-23_00:00:00.nc
geogrid met_em.d02.2010-03-23_06:00:00.nc
geogrid met_em.d02.2010-03-23_12:00:00.nc
geogrid.log
GRIBFILE.AAA metgrid
GRIBFILE.AAB metgrid.exe
GRIBFILE.AAC metgrid.log
GRIBFILE.AAD namelist.wps
GRIBFILE.AAE namelist.wps.all_options
GRIBFILE.AAF namelist.wps.fire
GRIBFILE.AAG namelist.wps.global
GRIBFILE.AAH namelist.wps.nmm
GRIBFILE.AAI README
GRIBFILE.AAJ ungrid
GRIBFILE.AAK ungrid.exe
GRIBFILE.AAL ungrid.log
GRIBFILE.AAM util
link_grib.csh Vtable
webpca@pesquisa: /home/WRF/WPS$
```

Figura 5 – Figura ilustrativa do terminal mostrando a pasta /home/webpca/WRF/WPS. Os retângulos vermelhos mostram os executáveis criados após a compilação do WPS.

Erros podem ocorrer associados à ausência das variáveis ambientais. Para checar se elas estão criadas no diretório WPS, digite o seguinte comando no terminal:

```
# env | grep -i NETCDF
```

Durante a instalação do WPS, pode ser necessário mudar a versão da biblioteca libpng, indicada na variável COMPRESSION_LIBS que está localizada dentro do arquivo configure.wps:

```
# nano configure.wps
(...)
COMPRESSION_LIBS=-L/usr/local/lib -ljasper -lpng -lz
```

Os erros mais comuns que ocorrem nesta operação estão associados à versão incorreta das bibliotecas do GRIB2. Normalmente, o ungrid.exe não é criado após a compilação. Uma dica, para verificar se os erros de compilação estão associados às bibliotecas do GRIB2, é realizar a compilação com as opções NO GRIB2 (p.ex.: opção 13, Fig. 5). Outra dica, é a utilização do comando ldd para determinar quais são as bibliotecas necessárias de um executável. (p.ex.: ldd ungrid.exe). O comando ldd – List Dynamic Dependencies – fornece uma lista das dependências dinâmicas de que um determinado

programa precisa para o seu correto funcionamento. Normalmente, ele retorna o nome de uma ou mais bibliotecas compartilhadas e suas localizações esperadas. Caso elas estejam ausentes, elas podem ser transferidas para o local esperado (p.ex.: /usr/lib).

3.4.3 Instalação do subsistema ARWpost

```
# tar -vzxf ARWpost.tar.gz
# cd ARWpost
# export NETCDF=/usr/local/netcdf
# export WRFIO_NCD_LARGE_FILE_SUPPORT=1
# ./configure ! (escolher a opção dentre as oferecidas, Fig. 6)
# ./compile >& compile.log
# cd ..
```

No último subsistema, após a sua configuração e compilação, o arquivo ARWpost.exe deve ter sido criado. Caso isto não ocorra, procurar erros dentro do arquivo compile.log, corrigi-los e repetir a seqüência de comandos deste tópico. Antes de recompilar, execute o *script* clean (./clean-a). A Figura 6 mostra as opções escolhidas para a instalação do modelo nos servidores do ICEA e a Figura 7, o arquivo criado no diretório WRFV3.

```
webpca@pesquisa:/home/WRF/ARWpost$ ./configure
Will use NETCDF in dir: /usr/local/netcdf
-----
Please select from among the following supported platforms.

  1. PC Linux i486 i586 i686 x86_64, PGI compiler (no vis5d)
  2. PC Linux i486 i586 i686 x86_64, PGI compiler (vis5d)
  3. PC Linux i486 i586 i686 x86_64, Intel compiler (no vis5d)
  4. PC Linux i486 i586 i686 x86_64, Intel compiler (vis5d)
  5. PC Linux i486 i586 i686 x86_64, gfortran compiler (no vis5d)
  6. PC Linux i486 i586 i686 x86_64, gfortran compiler (vis5d)

Enter selection [1-6] : 5
```

Figura 6 – Figura ilustrativa do terminal mostrando as opções escolhidas para a compilação do ARWpost. Os retângulos vermelhos mostram as opções selecionadas.

```
webpca@pesquisa:/home/WRF/ARWpost$ ls
arch          configure.arwp  namelist.ARWpost  test.dat
ARWpost.exe  fields.plt     README            util
clean         gibinfo.txt    scripts           wrfout_d01_2010-03-20_12:00:00
compile      gibmap.txt     src              wrfout_d02_2010-03-20_12:00:00
configure    myLIST        test.ctl
webpca@pesquisa:/home/WRF/ARWpost$
```

Figura 7 – Figura ilustrativa do terminal mostrando a pasta /home/webpca/WRF/ARWpost. O retângulo vermelho mostra o executável criado após a compilação do ARWpost.

3.5 Dados geográficos e de terreno:

Os dados geográficos e de terreno são descompactados no diretório data, criado dentro de /home/webpca/WRF. Esses dados são obtidos dentro de um único arquivo (geog.tar.gz) que, após descompactado, cria um novo diretório (geog) com aproximadamente 11GB.

```
# mkdir /home/webpca/WRF/data
# cd /home/webpca/WRF/data
# tar -zxvf geog.tar.gz
```

Obs.: Mais adiante, na execução do WPS, será editado o arquivo namelist.wps (variável geog_data_path) para indicar o caminho dos dados geográficos.

O conjunto de dados da superfície terrestre pertencente ao arquivo geog.tar.gz (dados estáticos) foram obtidos através de mapeamento por satélite. Parte desse arquivo contém categorias compatíveis com as da *United States Geological Survey* (USGS) que estão descritos nos arquivos VEGPARM.TBL e SOILPARM.TBL, localizados no diretório run do WRF. As resoluções dos dados são variadas (1°, 10', 5', 2' e 30'') e contêm as seguintes informações: albedo da superfície mensal (albedo_ncep); fração da vegetação mensal (greenfrac), índice de inclinação (slope index); categoria de uso da terra (landuse); albedo máximo da neve (maxsnowalb); dados de superfície (modis_landuse_20class_30s; apenas para usuários de modelagem de superfície da Noah LSM); dados para os esquemas de ondas de gravidade (orogwd); media anual de temperatura do solo de acordo com a profundidade (soiltemp); tipo de solo da camada inferior (soiltype bot); tipo de solo da camada superior (soiltype top); altura da topografia (topo).

3.6 Instalação do pacote gráfico GrADS

O *Grid Analysis and Display System* (GrADS) é uma das ferramentas gráficas mais difundidas e utilizadas no mundo para facilitar o acesso, manipulação e visualização de dados das principais variáveis atmosféricas e terrestres. O GrADS é distribuído gratuitamente na *internet* e suporta muitos formatos de arquivo de dados, incluindo o GRIB (versão 2), utilizado como condições iniciais e de contorno no modelo instalado no ICEA.

Primeiramente, o código fonte da versão 2.0 do GrADS é copiado e descompactado em /usr/local. Os códigos fontes pré-compilados estão disponibilizados no sítio do GrADS⁸. Os arquivos executáveis (gradsc, gradenc, gribmap, etc) são copiados para o diretório /usr/local/bin. O próximo passo é criar o diretório grads em /usr/local/lib e copiar os arquivos das bibliotecas do GrADS, referentes aos arquivos de mapas e fontes.

```
# cp -f /home/webpca/WRF/install/grads-2.02-linuxRHE3.tar /usr/local
# tar -vzxf grads-2.02-linuxRHE3.tar.gz
# cd grads-2.0.2/bin
# cp -f * /usr/local/bin
# mkdir /usr/local/lib/grads
# cp -fr grads-2.0.2/data/* /usr/local/lib/grads
```

A variável ambiental GASCRP é criada para indicar ao sistema operacional e demais aplicativos a localização da lista de diretórios contendo *scripts* do GrADS e do usuário. Para tal, a variável é acrescentada no final do arquivo .bashrc, localizado no diretório /home/<usuário>. Esse arquivo pode ser visualizado ao digitar no terminal do Linux o comando ls -a.

```
# cd /home/<usuário>
# nano .bashrc
(...)
export GASCRP=/usr/local/lib/grads
```

Após a edição, salve e atualize o arquivo:

```
# source .bashrc
```

Obs: 1) Em algumas distribuições Linux pode ser necessário criar um *link* simbólico com o arquivo libtermcap.so, localizada dentro do diretório /usr/lib (Prestrelo, 2011):

```
# ln -s /usr/lib/libtermcap.so /usr/lib/libtermcap.so.2
```

2) Outros arquivos do GrADS, disponibilizados no sítio do GrADS e/ou criados por seus usuários, também podem ser copiados para o diretório /usr/local/lib/grads.

⁸ <http://grads.iges.org/grads/downloads.html>

4 CONFIGURAÇÕES DO WRF

O ICEA utiliza o modelo WRF, não-hidrostático, com aninhamento *two-way*. Atualmente, gera operacionalmente previsões das condições do tempo por um período de 48 horas, sendo inicializado 4 vezes ao dia (00Z, 06Z, 12Z e 18Z). As suas saídas fornecem informações em intervalos de 3 horas. Os prognósticos são gerados para quatro domínios diferentes, cada qual com uma grade aninhada. Assim, para a mesma região de interesse existem duas grades, entretanto com resoluções distintas. As resoluções espaciais são de 18 km, para o domínio principal (D1), de 6 km, para o domínio aninhado (D2). Os domínios denominados de NORTE, NORDESTE, SUDESTE e ALCÂNTARA estão centrados na latitude e longitude de 5°S e 60°W (Fig. 8a e b), 8°S e 39°W (Fig. 8c e d), 25,01° S e 49,17° W (Fig. 8e e f), 02, 40°S e 44, 40°W (Fig. 8g e h), respectivamente. A escolha dos domínios está associada ao projeto do Comando da Aeronáutica cujo objetivo é de pesquisa e desenvolvimento da modelagem numérica para fins aeronáuticos e aeroespaciais, tais como, disponibilizar prognósticos das condições do tempo em alta resolução para as Regiões de Informação de Vôo (FIRs) e para os dois Centros de Lançamentos do Brasil: Centro de Lançamento de Alcântara (CLA) e Centro de Lançamento da Barreira do Inferno (CLBI).

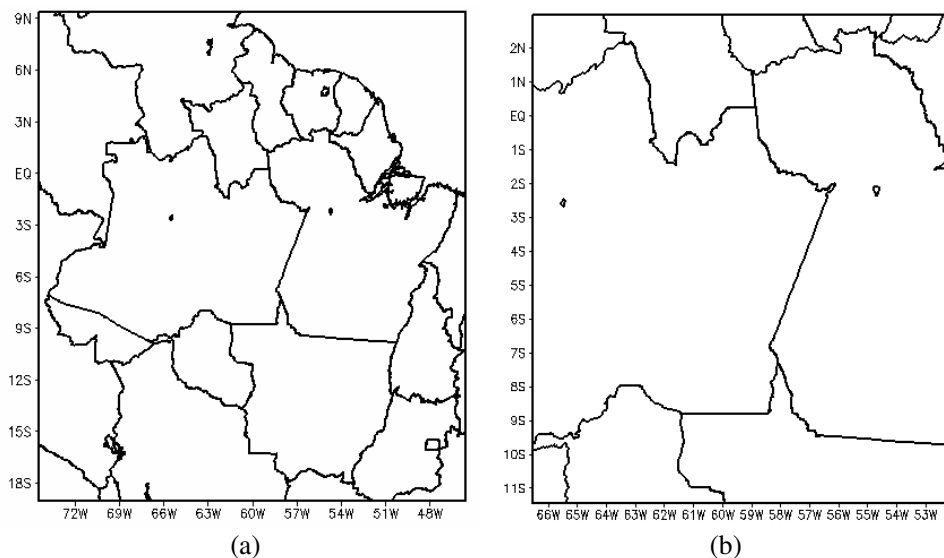


Figura 8 – Domínio das grades NORTE com resolução de 18 km (a) e 6 km (b); NORDESTE com resolução de 18 km (c) e 6 km (d); SUDESTE com resolução de 18 km (e) e 6 km (f); ALCANTARA com resolução de 18 km (g) e 6 km (h). (Continua)

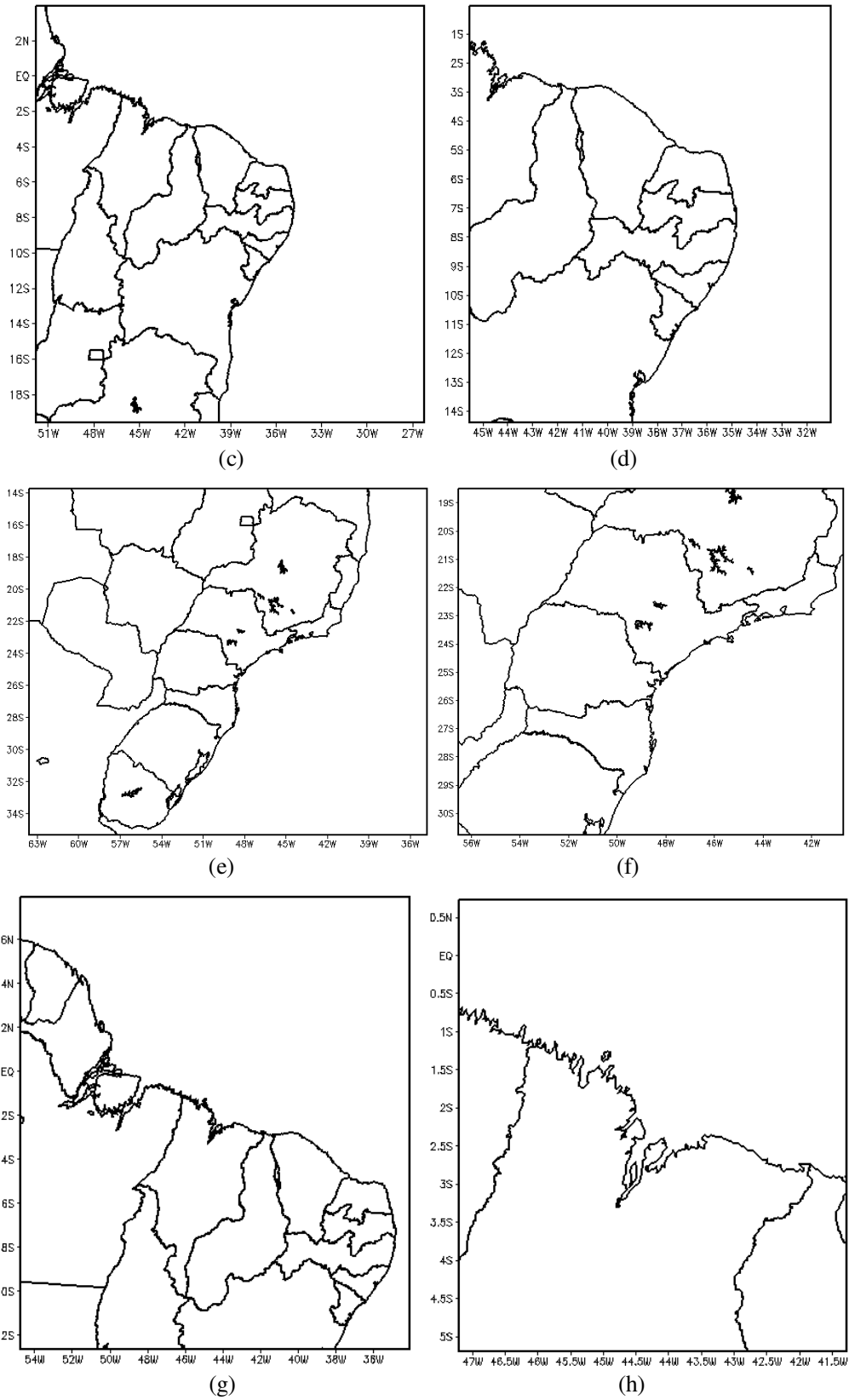


Figura 8 – Domínio das grades NORTE com resolução de 18 km (a) e 6 km (b); NORDESTE com resolução de 18 km (c) e 6 km (d); SUDESTE com resolução de 18 km (e) e 6 km (f); ALCANTARA com resolução de 18 km (g) e 6 km (h). (conclusão)

A seguir, é apresentada uma síntese das configurações para cada domínio (Tab. 1).

Tabela 1 – Configurações do modelo WRF.

DOMÍNIO	Resolução horizontal (km)	Nº de pontos zonal x meridional	Extensão do domínio (em graus)
NORTE	18	103 x 103	74,5-45,5°W/19°S-9,5°N
	6	181 x 181	66,5-53°W/11,5°S-3°N
NORDESTE	18	160 x 150	52-26°W/19,5°S-4°N
	6	271 x 271	39-30°W/12°S-3°N
SUDESTE	18	180 x 150	63,5-35°W/35,5-13,5°S
	6	295 x 253	49,5-44°W/26,5-20°S
ALCÂNTARA	18	130 x 130	55-34°W / 12,5°S-8°N
	6	112 x 112	47,5-41°W / 5,5°S-1°N

A resolução vertical utilizada é de 28 níveis, com pressão no topo da atmosfera de 50 hPa. As parametrizações físicas empregadas no modelo são: WSM3 (HONG, DUDHIA; CHEN, 2004) para microfísica (convecção explícita); RRTM (MLAWER ET AL., 1997) para radiação de onda longa; Dudhia (DUDHIA, 1989) para radiação de onda curta; MM5 similaridade (PAULSON; DYER; HICKS; WEBB, 1970) para camada superfície; Noah-LSM (CHEN; DUBHIA, 2001) para processos de superfície; Yonsei University Scheme (HONG; NOH; DUDHIA, 2006); e Kain-Fritsch (KAIN; FRITSCH, 1990; 1993) convecção rasa e profunda. A escolha das parametrizações físicas segue os valores *default* da instalação do modelo (Tab. 2).

Tabela 2 – Parametrizações adotadas para as simulações.

parâmetro ou processo físico	valor ou parametrização
microfísica (convecção explícita)	WSM3 (Hong; Dudhia; Chen, 2004)
radiação onda longa	RRTM (Malawer et al., 1997)
radiação de onda curta	Dudhia (Dudhia, 1989)
camada superfície	MM5 similaridade (Paulson et al., 1970)
processos de superfície	Noah-LSM (Chen e Dudhia, 2001)
camada limite planetária	Yonsei University (Hong; Noh; Dudhia, 2006)
convecção rasa e profunda	Kain-Fritsch (Kain; Fritsch, 1990; 1993)

O passo temporal utilizado para D1 (D2) é de 108s (36s). Para o WRF recomenda-se realizar as integrações com passo de tempo equivalente até seis vezes o valor do espaçamento de grade (Δx , em km, ARW V3 *Modeling System User's Guide*, 2011). As condições iniciais e de contorno provêm das análises do modelo global *Global Forecast System* (GFS) do NCEP⁹ no formato GRIB2, com resolução horizontal de 0,5°; resolução temporal de 6h; e 64 níveis na vertical. O conjunto de dados de tipo de solo e uso do terreno utilizado foi obtido da USGS, dividido em 24 categorias (*USGS 24-category data*). A Tabela 3 mostra características adicionais do WRF:

Tabela 3 – Característica do modelo WRF.

Equações Básicas	Totalmente compressíveis
Coordenadas verticais	Terrain following para a altura e pressão.
Estrutura horizontal da grade	Arakawa-C
Método de integração	3ª ordem Runge-Kutta
Equações	Conservação da massa, momentum e escalares usando equações prognósticas na forma de fluxo
Advecção	Diferenças centradas de 6ª ordem ou 5ª ordem “upwind”.

Fonte: SKAMAROCK et al, 2008

⁹ <http://nomads.ncep.noaa.gov/pub/data/nccf/com/gfs/prod>

5 EXECUTANDO O WRF

Para realizar as alterações descritas no tópico anterior, é necessário configurar alguns parâmetros dentro de três arquivos de texto (namelist): `namelist.wps` (pré-processamento), `namelist.input` (processamento) e `namelist.ARWpost` (pós-processamento). Um arquivo `namelist`, codificado em linguagem de programação FORTRAN, contém um conjunto de variáveis para ser lidas durante a execução do programa. O uso do `namelist` permite alterar a configuração do programa sem a necessidade de recompilar o código fonte. Como exemplo, serão mostrados os parâmetros dos `namelist` alterados da grade SUDESTE (Apêndice A). As alterações das outras grades, NORDESTE, NORTE e ALCÂNTARA, podem ser visualizados nos Apêndices B, C, D, respectivamente. Os parâmetros dos arquivos originais, que não forem modificados, serão representados somente por três pontos entre parênteses (...). As alterações dos parâmetros associados ao ano, mês, dia, hora, minuto e segundo do período inicial e final das simulações ocorrerão por meio de um script em SHELL e serão representados pelas palavras YYYY, MM, DD, hh, mm e ss, respectivamente. Todos horários estão em *Coordinated Universal Time* (UTC). O script em SHELL será executado automaticamente por meio do programa CRON. Este programa tem a função de programar a execução de comandos e processos automaticamente nas datas e horários pré-determinados. Neste tópico, são também mostrados os comandos utilizados para executar os subsistemas do WRF (WPS, WRF e WRFpost).

5.1 Execução do WPS

Dentro do diretório WPS, serão editados as variáveis do arquivo `namelist.wps`. Este arquivo é composto por quatro partes: `share`, `geogrid`, `ungrid` e `metgrid`. No `share`, são descritas as variáveis utilizadas por mais que um subsistema do WPS. No `geogrid`, `ungrid` e `metgrid` são especificadas as variáveis para o programa `geogrid`, `ungrid`, `metgrid`. Essas variáveis definem o tamanho e localização de todos os domínios do modelo e dos dados geográficos.

Para armazenar os dados de condições iniciais e de contorno, é criado o diretório GFS2 dentro de `/home/webpca/WRF/data`. Os arquivos GFS têm aproximadamente 55 MB e o seguinte formato: `gfs.tHHz.pgrb2fHH`, onde a primeira informação horária (HH) indica

o início da simulação e o segundo, o período da previsão. Por exemplo, gfs.18Zpgrb2f12, é um arquivo GFS das 12 horas de previsão com hora inicial da simulação às 18Z.

5.1.1 Execução do GEOGRID

Para o GEOGRID, são editadas duas partes do arquivo namelist.wps: share e geogrid. Nesta etapa, é definido o domínio do modelo e interpolado os dados terrestres para as grades do modelo.

```
# cd /home/webpca/WRF/WPS
# nano namelist.wps
```

Somente as variáveis mostradas abaixo foram editadas:

```
&share
(...)
max_dom: 2 # número máximo de domínios
(...)
&geogrid
(...)
i_parent_start = 1, 45, # localização inicial da grade aninhada (eixo x)
j_parent_start = 1, 35, # localização inicial da grade aninhada (eixo y)
(...)
e_we = 180, 295, # número de pontos de grade zonais (grade mãe, aninhada)
(...)
e_sn = 150, 253, # número de pontos meridionais (grade mãe, aninhada)
geog_data_res = '5m', '2m', # resolução dos dados de terreno (grade mãe, aninhada)
dx = 18000, # resolução horizontal no eixo x (grade mãe, em metros)
dy = 6000, # resolução horizontal no eixo y (grade mãe, em metros)
map_proj = 'mercator',
ref_lat = -25.01, # latitude do ponto central (grade-mãe)
ref_lon = -49.17, # longitude do ponto central (grade-mãe)
(...)
stand_lon = -49.17, # longitude padrão paralela ao eixo x da projeção mercator
```



```
geog_data_path = '/home/webpca/WRF/data/geog',      # localização dos dados
geográficos
(...)
```

Obs: Não é necessário informar a resolução da grade aninhada ($dx=dy=12000$ m), porque ela segue a razão da grade de 1/3 (*default*), declarada na variável `parent_grid_ratio`.

Antes de executar este subsistema, certifique-se que o arquivo `GEOGRID.TBL` esteja ligado corretamente a sua respectiva versão. Há vários arquivos desse tipo que suportam diferentes núcleos dinâmicos do WRF. `GEOGRID.TBL.ARW` deve ser utilizado para o núcleo ARW.

```
# ls -l geogrid/GEOGRID.TBL
GEOGRID.TBL -> GEOGRID.TBL.ARW
```

Agora, executa-se o `geogrid.exe`. Se a operação for realizada sem falhas, é mostrada na última linha uma mensagem que informará o êxito da operação na última linha do arquivo `log`:

```
# ./geogrid.exe>& geogrid.log
**** Successful completion of program geogrid.exe ****
```

Execute o comando `ls -ltr geo*.nc`. O resultado esperado é: `geo_em_d01.nc` e `geo_em.d02.nc`. Estes são os arquivos da grade criada (um arquivo para cada domínio), após a execução do `geogrid.exe`. Se não forem feitas alterações na grade, o uso do `geogrid.exe` não será mais necessário.

5.1.2 Execução do UNGRIB

Após a criação das grades e interpolado os dados de terreno com o `geogrid.exe`, executa-se o UNGRIB. O UNGRIB extrai os campos meteorológicos do formato GRIB2 necessários para inicializar o modelo. Para examinar se os arquivos GRIB podem ser lidos, utilize o `g2print.exe` (caso os arquivos estejam no formato GRIB 1, utiliza-se o `g1print.exe`):

```
# ./util/g2print.exe /home/webpca/WRF/data/GFS2/YYYYMMDDHH/<
arquivo GFS >
```

Para o UNGRIB, somente o share e ungrib necessitam estar editados no namelist.wps:

```
&share
(...)
start_date = DATA/HORA, DATA/HORA, # data/hora inicial simulação das grades
        mãe e aninhada (ex.: '2013-01-10_00:00:00', '2013-01-10_00:00:00')
end_date = DATA/HORA, DATA/HORA, # data/hora final simulação das grades mãe e
        inicial da aninhada (ex.: '2013-01-13_00:00:00', '2013-01-10_00:00:00')
interval_seconds = 10800, # frequência dos arquivos de saídas (em s). Modelo
        configurado para gerar arquivos de saídas a cada 3 h.
(...)
&ungrib
(...)
prefix = GFS2, # prefixo para os arquivos intermediários (ex.: GFS2:ANO-MES-
        DIA_HORA). Pode incluir um caminho.
```

O próximo passo é criar os *links* simbólicos para informar a localização dos arquivos GFS e dos arquivos Vtable. Os arquivos Vtable informam os campos meteorológicos a serem extraídos dos arquivos GRIB2. Alguns Vtables estão localizados no diretório WPS/ungrib/Variable_Tables (p.ex.: Vtable.GFS, Vtable.SST, Vtable.ECMWF).

```
# ln -s ungrib/Variable_Tables/Vtable.GFS Vtable
# ls -l Vtable
Vtable -> ungrib/Variable_Tables/Vtable.GFS
```

Após os *links* estarem prontos, os seguintes arquivos são visualizados: GRIBFILE.AAA, GRIBFILE.BBB, GRIBFILE.CCC, etc (um para cada arquivo GFS). O *script* link_grib.csh pode ser usado para ligar esses arquivos aos arquivos GRIB.

```
# ./link_grib.csh /home/webpca/WRF/data/GFS2/YYYYMMDDhh/gfs*
# ls -l GRIBFILE.*
GRIBFILE.AAA->/home/webpca/WRF/data/GFS2/YYYYMMDDhh/<nome.gfs>
```

Executado o ungrib, se a operação for realizada sem falhas, é mostrada uma mensagem que informará o êxito da operação na última linha do arquivo log.

```
# ./ungrid.exe >& ungrib.log
"*** Successful completion of program ungrib.exe ***".
```

Serão criados arquivos com o seguinte formato GFS2:YYYY-MM-DD_hh (p.ex.: GFS2:2013-01-11_00). Para examinar se os arquivos intermediários podem ser lidos, utilize o rd_intermediate.exe:

```
#!/util/rd_intermediate.exe GFS2: YYYY-MM-DD_hh
```

5.1.3 Execução do METGRID

O último subsistema do pós-processamento a ser executado é o METGRID. A sua função é interpolar horizontalmente os campos meteorológicos extraídos pelo ungrib.exe para as grades do modelo. Os dados de entrada do METGRID são os arquivos geo_em.dxx.nc (arquivos de saída do GEOGRID); e os arquivos de saída intermediários (provenientes do UNGRIB). Para o METGRID, somente o share e metgrid necessitam estar editados no namelist.wps:

```
&share
(...)
start_date = DATA/HORA, DATA/HORA, # data/hora inicial simulação das grades
        mãe e aninhada (ex.: '2013-01-10_00:00:00', '2013-01-10_00:00:00')
end_date = DATA/HORA, DATA/HORA, # data/hora final simulação das grades mãe e
        aninhada (ex.: '2013-01-13_00:00:00', '2013-01-13_00:00:00')
(...)
&metgrid
fg_name = 'GFS2',    # prefixo para os arquivos intermediários
(...)
opt_output_from_metgrid_path = 'home/WRF/WRFV3/test/em_real' # caminho para
saídas dos arquivos intermediários
```

Obs.: Sugere-se que a grade aninhada tenha tempos iniciais e finais idênticos ao tempo inicial desejado para a grade aninhada durante a execução do WPS. Isto ocorre porque a grade aninhada obtém as suas condições de contorno laterais da sua grade mãe, e, assim, somente o tempo inicial para uma grade aninhada necessita ser processado pelo WPS, exceto quando for usada análise nudging no WRF.

Antes de executar o METGRID, certifique-se que METGRID.TBL esteja com o *link* correto.

```
# ls -l metgrid/METGRID.TBL
METGRID.TBL -> METGRID.TBL.ARW
```

Para executá-lo, digite o comando:

```
# ./metgrid.exe >& metgrid.log
“*** Successful completion of program metgrid.exe ***”
```

Os dados de saída do metgrid são:

- met_em.d01.YYYY-MM_DD_hh_mm.nc – um arquivo por tempo; e
- met_em.dxx. YYYY-MM_DD_hh_mm.nc – um arquivo por aninhamento, somente para a hora inicial. Estes arquivos podem ser checados com a ferramenta ncdump (esta ferramenta encontra-se nos arquivos do netcdf):

```
# /usr/local/netcdf-4.1.3/bin/ncdump -h met_em.d01.YYYY-MM_DD_hh_mm.nc
```

5.2 Execução do real e WRF

Este tópico mostrará a execução de dois subsistemas do WRF: o real e o WRF. O trabalho de interpolar verticalmente os campos do WRF é feito dentro do programa real. O WRF é responsável pela execução do modelo propriamente dito. Nesta etapa, o primeiro passo, é editar o arquivo de texto namelist.input que se encontra dentro do diretório /home/webpca/WRF/WRFV3/test/em_real.

```
# cd /home/webpca/WRF/WRFV3/test/em_real
# nano namelist.input
```

Neste arquivo, são definidos parâmetros como o período da simulação (time_control), a configuração das grades (domains) e as parametrizações físicas (physics).

```
&time_control
run_days = 0, # dias simulados (Somente grade mãe)
run_hours = 48, # horas simuladas (Somente grade mãe)
(...)
start_year = YYYY, YYYY, # ano inicial grade mãe, aninhada (ex. 2013)
start_month = MM, MM, # mês inicial grade mãe, aninhada (ex. 01)
start_day = DD, DD, # dia inicial grade mãe, aninhada (ex. 10)
start_hour = hh,hh, # hora inicial grade mãe, aninhada (ex. 00)
```

```

start_minute = mm, mm,      # minutos inicial grade mãe, aninhada (ex. 00)
start_second  = ss, ss,      # segundos inicial grade mãe, aninhada (ex. 00)
end_year      = YYYY, YYYY, # ano final grade mãe, aninhada (ex. 2013)
end_month     = MM, MM,      # mês final grade mãe, aninhada (ex. 01)
end_day       = DD, DD,      # dia inicial grade mãe, aninhada (ex. 13)
end_hour      = hh, hh,      # hora inicial grade mãe, aninhada (ex. 00)
end_minute    = mm, mm,      # minutos inicial grade mãe, aninhada (ex. 00)
end_second    = ss, ss,      # segundos inicial grade mãe, aninhada (ex. 00)
interval_seconds = 10800,    # frequência dos arquivos de saídas (em s). Modelo
configurado para gerar arquivos de saídas a cada 3 h.
(...)
input_from_file = .true.,true., # Variável lógica, se uma grade aninhada exigir um
arquivo de entrada de dados (p. ex. wrfinput_d02).
(...)
&domains
(...)
max_dom       = 2,           # número de domínios (grade mãe e 1 grade aninhada)
(...)
e_we          = 180, 295,    # número de pontos na direção x (oeste-leste)
(...)
e_sn          = 150, 253,    # número de pontos na direção y (sul-norte)
(...)
e_vert        = 28, 28,      # número de pontos na direção z (vertical); a dimensão
vertical deve ser a mesma para todas as grades
(...)
dx            = 18000, 6000, # comprimento da grade no eixo x (em m)
dy            = 18000, 6000, #comprimento da grade no eixo y (em m). Nota: dx=dy
(...)
i_parent_start = 1, 45,     # início da grade ponto I (I grade mãe, I grade aninhada)
j_parent_start = 1, 35,     # início da grade ponto J (J grade mãe, J grade aninhada)
(...)

```

Obs.: Foram mantidas no namelist.input original parâmetros importantes, como por exemplo, o espaço temporal (180s) e as parametrizações físicas (Tab. 1). Futuramente,

serão realizados testes de sensibilidade para encontrar o conjunto de parametrizações disponíveis no modelo que melhor represente os sistemas atmosféricos atuantes nos domínios simulados.

Com o namelist.input editado, executa-se o arquivo real.exe:

```
# ./real.exe >& real.log  
“***Successful completion real.exe***”
```

Após a execução do real.exe, dois arquivos são criados: wrfinput_d01 e wrfinput_d02. A criação desses arquivos confirma que o modelo está pronto para ser executado. Testes podem ser realizados para verificar o seu conteúdo:

```
# /usr/local/netcdf-4.1.3/bin/ncdump -h wrfinput_d01  
# ncdump -v Times wrfinput
```

Por fim, a execução do modelo é dada com o seguinte comando:

```
# ./wrf.exe >& wrf.log  
“***Successful completion real.exe***”
```

Finalizada a execução, é criado dois novos arquivos (um para cada domínio): wrfout_d01_YYY-MM-DD_hh:mm:ss, wrfout_d02_YYY-MM-DD_hh:mm:ss. Estes arquivos são transferidos para o diretório /home/webpca/WRF/ARWpost.

```
# mv wrfout* /home/webpca/WRF/ ARWpost /
```

5.3 Execução do ARWpost

Executado o modelo, entraremos na etapa do pós-processamento. O ARWpost é usado para converter as saídas do modelos para um formato que possam ser visualizados por ferramentas gráficas (GrADS e Vis5d) . Para executá-lo, primeiro deve-se editar o arquivo namelist.ARWpost:

```
# cd ../ARWpost  
# nano namelist.ARWpost
```

No namelist.ARWpost são definidos a data inicial e final da simulação e o intervalo de tempo das saídas dos modelo (datetime); o nome do arquivo WRF que se deseja

visualizar, o nome do arquivo de saída para visualizar no GRADS; os campos meteorológicos (io); e os níveis de pressão (interp). As alterações para a grade mãe (D1) são:

```
&datetime
start_date = 'YYYY-MM-DD_hh:mm:ss',#data/hora inicial (ex.:2013-01-10_00:00:00)
end_date = 'YYYY-MM-DD_hh:mm:ss', # data/hora final (ex.:2013-01-13_00:00:00)
interval_seconds = 10800, # frequência dos arquivos de saídas (em s). Modelo
configurado para gerar arquivos de saídas a cada 3 h.
(...)
&io
(...)
input_root_name = './wrfout_d01_YYYY-MM-DD_hh:mm:ss' # diretório de entrada
dos dados (saídas do WRF)
output_root_name = './wrfd01_YYYY-MM-DD_hh:mm:ss' # diretório de saída dos
dados do ARWpost
plot = 'all_list' # Escolha dos campos a ser processados. “all_list”: campos do arquivo
WRF e listados na variável ‘fields’.
fields = 'cape, cin, mcape, mcin, clfr, dbz, max_dbz, geopt, height, lcl, lfc, pressure, rh,
rh2, theta, tc, tk, td, td2, slp, umet, vmet, u10m, v10m, wdir, wspd, wd10, ws10' # lista
os campos meteorológicos escolhidos pelo usuário. Utilizado somente com a opção
“list” na variável ‘plot’.
(...)
&interp
interp_method = 1, # 0 - níveis sigma; -1 – código define “adequados” níveis de
altura; 1- usuário define níveis de pressão (hPa) ou altura (km)
interp_levels = 1000., 975., 950., 925., 900., 850., 800., 750., 700., 650., 600., 550.,
500., 450., 400., 350., 300., 250., 200., 150., 100., # só usado se interp_method=1.
Indicar níveis para interpolar em hPa (pressão) ou km (altura acima do nível do mar).
Níveis informados de baixo para cima.
```

Obs.: A variável fields informa, respectivamente, os seguintes campos diagnósticos: *Convective Available Potential Energy* (CAPE), *Convective Inhibition* (CIN), CAPE máximo, CIN máximo, fração de nuvens baixas/medias e altas; refletividade, refletividade máxima, geopotencial, altura (em Km), nível de condensação por

levantamento (NCL), nível de convecção livre (NCL), pressão (em hPa), umidade relativa, umidade relativa 2m, temperatura potencial, temperatura em °C, temperatura em K, temperatura do ponto de orvalho em °C, temperatura do ponto de orvalho 2m, pressão ao nível do mar, ventos rotacionados para coordenadas terrestres, ventos rotacionados para coordenadas terrestres 10m, direção do vento, velocidade do vento, direção do vento 10m, velocidade do vento 10m. Com a opção “all_list” na variável plot é possível visualizar as 114 variáveis do arquivo WRF.

Após as modificações no namelist.ARWpost, execute o ARWpost.exe:

```
# ./ARWpost.exe >& ARWpostD1.log
```

Altere somente as variáveis input_root_name e output_root_name para gerar as saídas do modelo para o domínio aninhado (D2):

```
(...)  
input_root_name = './wrfout_d02_ YYYY-MM-DD_hh:mm:ss' # diretório de entrada  
dos dados  
output_root_name = './wrfd02_ YYYY-MM-DD_hh:mm:ss' # diretório de saída dos  
dados  
(...)
```

E execute novamente o ARWpost.exe:

```
# ./ARWpost.exe >& ARWpostD2.log
```

Concluído a execução do pós-processamento, são gerados para cada domínio um arquivo descritor (.ctl) e um de dados (.dat) que poderão ser visualizadas pelo GrADS: (p.ex.: wrfd1_YYYYMMDDhh.ctl ou .dat; e wrfd2_YYYYMMDDhh.ctl ou .dat).

6 CONSIDERAÇÕES FINAIS

O ICEA instala uma nova versão do modelo regional WRF (versão 3.4.1), com cobertura para as Regiões de Informação de Vôo (FIRs) do território brasileiro e para os dois Centros de Lançamentos do Brasil: Centro de Lançamento de Alcântara (CLA) e Centro de Lançamento da Barreira do Inferno (CLBI). A versão 3.4.1 do WRF disponibiliza prognósticos meteorológicos operacionalmente na homepage do ICEA <www.icea.gov.br/climatologia/modelagemWRF.html>. Esta nova versão do modelo oferece os mais recentes avanços da física, modelagem numérica e assimilação de dados e possui uma arquitetura de software pronta para o processamento paralelo. Concomitantemente, foram adquiridos novos recursos computacionais e de conectividade pelo Laboratório de Climatologia do ICEA.

Com essa modernização do sistema de modelagem numérica instalado no ICEA, foi possível realizar mudanças significativas nas configurações do modelo, destacando-se: i) aumento da resolução espacial (18 e 6 km); ii) redimensionamento dos domínios; e iii) um aumento do número de rodadas do modelo, ou seja, a sua inicialização 4 vezes ao dia (00Z, 12Z, 18Z e 00Z). Essas modificações visam aumentar a qualidade das previsões meteorológicas realizadas pelos Centros de Previsão Meteorológica nacionais (CNMA, CMV, CMA e CMM) e Centro de Gerenciamento de Navegação Aérea (CGNA).

Desta forma, o Programa de Modelagem Numérica do Tempo (PMNT), desenvolvido no ICEA, sob a coordenação de seu Grupo de Trabalho (GT), estará em posição de destaque, em relação às principais instituições acadêmicas do mundo, nas pesquisas na área de previsão operacional do tempo. Espera-se que o presente trabalho tenha contribuído para uma melhor compreensão das características e dos procedimentos necessários para a instalação, compilação e configuração do WRF. Informação detalhada e precisa da utilização desse modelo é necessária para o aperfeiçoamento do sistema de previsão de tempo, tornando-se útil para serem aplicadas nas atividades de interesse do Comando da Aeronáutica, tais como, gerenciamento do fluxo de tráfego aéreo e lançamento de foguetes.

7 AGRADECIMENTOS

Ao Grupo de Trabalho (GT) do Programa de Modelagem Numérica do Tempo (PMNT) pela iniciativa do desenvolvimento deste projeto. Ao ICEA, pela disponibilização de sua infraestrutura e dos recursos computacionais. Ao IAE, em especial aos pesquisadores Drº. Gilberto Fernando Fisch e Drº. Marcos Daisuke Oyama pelas valiosas sugestões. Com o intuito de aperfeiçoar este manual, quaisquer sugestões poderão ser encaminhadas para o e-mail: pmnt@icea.gov.br

Citação: CARVALHO, M. A. V. ; ABRUNHOSA, M. L.; LOPES, J. R. O. **Manual de instalação, compilação e execução do modelo de mesoescala WRF no ICEA (versão 3.4.1)**. Subdivisão de Climatologia e Arquivo Meteorológico (PBCA), Instituto de Controle do Espaço Aéreo (ICEA), São José dos Campos, 2013.

8 REFERÊNCIAS BIBLIOGRÁFICAS

ARW Version 3.4.1 Modeling System User's Guide. Weather Research & Forecasting. Mesoscale & Microscale Meteorology Division. National Center for Atmospheric Research. Aug. 2012.

CHEN, F.; DUDHIA, J. Coupling an advanced land-surface/ hydrology model with the Penn State/ NCAR MM5 modeling system. Part I: Model description and implementation. **Monthly Weather Review**, v. 129, p. 569-585, 2001.

DYER, A. J.; HICKS, B. B. Flux-gradient relationships in the constant flux layer, **Quart. J. Roy. Meteor. Soc.**, v. 96, p. 715-721, 1970.

DUDHIA, J. Numerical study of convection observed during the winter monsoon experiment using a mesoscale two-dimensional model. **J. Atmos. Sci.**, v. 46, p. 3077-3107, 1989.

HONG, S-Y; DUDHIA, J.; CHEN, S.-H. A revised approach to ice-microphysical processes for the bulk parameterization of clouds and precipitation. **Mon. Wea. Rev.**, 132, 1, 103-120, 2004.

HONG, S.-Y; NOH, Y.; DUDHIA, J. A new vertical diffusion package with an explicit treatment of entrainment processes. **Mon. Wea. Rev.**, v. 134, p. 2318–2341, 2006.

IRIART, P. G.; CARVALHO, M. V. C; PEREIRA NETO, A.V. **Manual de instalação, compilação e execução do sistema de modelagem numérica WRF no ICEA.** Subdivisão de Climatologia e Arquivo Meteorológico (PBCA), Instituto de Controle do Espaço Aéreo (ICEA), São José dos Campos, 2011.

KAIN, J. S.; FRITSCH, J. M. A one-dimensional entraining/detraining plume model and its application in convective parameterization. **J. Atmos. Sci.**, v. 47, n. 23, p. 2748-2802, 1990.

KAIN, J. S.; FRITSCH, J. M. Chapter 16 – Convective parameterization for mesoscale models: The Kain-Fritsch Scheme, in meteorological monographs. **Amer. Meteor. Soc.**, v. 24, n. 46, 1993.

MLAWER, E. J.; TAUBMAN, S. J.; BROWN, P. D.; IACONO, M. J.; CLOUGH, S. A. Radiative transfer for inhomogeneous atmosphere: RRTM, a validated correlated-k model for the longwave. **Journal of Geophysical Research**, v.102, n. D14, p. 16663-16682, 1997.

OYAMA, M.D. **Instalação do modelo de mesoescala MM5 na ACA/IAE/CTA.** São José dos Campos: Divisão de Ciências Atmosféricas, Instituto de Aeronáutica e Espaço, Centro Técnico Aeroespacial, 2003. Publicação interna, PI-002.

PAULSON, C. A. The mathematical representation of wind speed and temperature profiles in the unstable atmospheric surface layer. **Journal of Applied Meteorology**, v. 9, p. 857-861, 1970.

PRESTELO, C. **Instalação da biblioteca NETCDF e GRADS**. Disponível em: <http://prestrellocristiano.webnode.com.br/tutoriais/instalacaodeprogramas/>.

SKAMAROCK, W.C.; KLEMP, J.B.; DUDHIA, J.; GILL, D.O.; BARKER, D.M.; DUDA, M.G.; HUANG, X-Y; WANG, W.; POWERS, J.G. **A Description of the Advanced Research WRF Version 3**. NCAR Tech. Note NCAR/TN-475+STR, Jun, 2008. 113 pp.

WEBB, E. K. Profile relationships: The log-linear range, and extension to strong stability. **Quart. J. Roy. Meteor. Soc.**, v. 96, p. 67-90, 1970.

APÊNDICE A – NAMELISTS GRADE SUDESTE

➤ **namelist.wps**

```
&share
wrf_core = 'ARW',
max_dom = 2,
start_date = '2013-01-10_00:00:00', '2013-01-10_00:00:00',
end_date = '2013-01-13_00:00:00', '2013-01-13_00:00:00',
interval_seconds = 10800,
io_form_geogrid = 2,
/

&geogrid
parent_id = 1,1,
parent_grid_ratio = 1,3,
i_parent_start = 1,45,
j_parent_start = 1,35,
e_we = 180,295,
e_sn = 150,253,
geog_data_res = '5m','2m',
dx = 18000,
dy = 18000,
map_proj = 'mercator',
ref_lat = -25.01,
ref_lon = -49.17,
truelat1 = 0.0,
truelat2 = 0.0,
stand_lon = -49.17,
geog_data_path = '/home/webpca/WRF/data/geog'
/

&ungrib
out_format = 'WPS',
prefix = 'GFS2',
/

&metgrid
fg_name = 'GFS2',
io_form_metgrid = 2,
opt_output_from_metgrid_path = '/home/webpca/WRF/WRFV3/test/
em_real'
```

➤ **namelist.input**

```
&time_control
run_days = 0,
run_hours = 48,
run_minutes = 0,
run_seconds = 0,
start_year = 2013,2013,
start_month = 01,01,
start_day = 10,10,
```

```

start_hour           = 00,00,
start_minute        = 00,00,
start_second        = 00,00,
end_year            = 2013,2013,
end_month           = 01,01,
end_day             = 13,13,
end_hour            = 00,00,
end_minute          = 00,00,
end_second          = 00,00,
interval_seconds    = 10800
input_from_file     = .true.,.true.,
history_interval    = 180, 60,
frames_per_outfile  = 1000, 1000,
restart             = .false.,
restart_interval    = 5000,
io_form_history     = 2
io_form_restart     = 2
io_form_input       = 2
io_form_boundary    = 2
debug_level         = 0
/

```

```

&domains
time_step           = 108,
time_step_fract_num = 0,
time_step_fract_den = 1,
max_dom             = 2,
s_we                = 1, 1,
e_we                = 180, 295,
s_sn                = 1, 1,
e_sn                = 150, 253,
s_vert              = 1, 1,
e_vert              = 28, 28,
p_top_requested     = 5000,
num_metgrid_levels  = 27,
num_metgrid_soil_levels = 4,
dx                  = 18000, 6000,
dy                  = 18000, 6000,
grid_id             = 1, 2,
parent_id           = 0, 1,
i_parent_start      = 1, 45,
j_parent_start      = 1, 35,
parent_grid_ratio    = 1, 3,
parent_time_step_ratio = 1, 3,
feedback            = 1,
smooth_option       = 0
/

```

```

&physics
mp_physics          = 3, 3,
ra_lw_physics       = 1, 1,
ra_sw_physics       = 1, 1,
radt                = 30,30,
sf_sfclay_physics  = 1, 1,
sf_surface_physics = 2, 2,
bl_pbl_physics      = 1, 1,

```

```

bldt = 0, 0,
cu_physics = 1, 1,
cudt = 5, 5,
isfflx = 1,
ifsnow = 0,
icloud = 1,
surface_input_source = 1,
num_soil_layers = 4,
sf_urban_physics = 0, 0,
maxiens = 1,
maxens = 3,
maxens2 = 3,
maxens3 = 16,
ensdim = 144,
/

&fdda
/

&dynamics
w_damping = 0,
diff_opt = 1,
km_opt = 4,
diff_6th_opt = 0, 0,
diff_6th_factor = 0.12, 0.12,
base_temp = 290.
damp_opt = 0,
zdamp = 5000., 5000.,
dampcoef = 0.2, 0.2,
khdif = 0, 0,
kvdif = 0, 0,
non_hydrostatic = .true., .true.,
moist_adv_opt = 1, 1,
scalar_adv_opt = 1, 1,
/

&bdy_control
spec_bdy_width = 5,
spec_zone = 1,
relax_zone = 4,
specified = .true., .false.,
nested = .false., .true.,
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/

```

➤ **namelist. ARWpost**

```
&datetime
start_date = '2013-01-10_00:00:00',
end_date = '2013-01-13_00:00:00',
interval_seconds = 10800,
tacc = 0,
debug_level = 0,
/

&io
io_form_input = 2,
input_root_name = './wrfout_d02_2013-01-10_00:00:00'
output_root_name = './wrfd2_2013011000'
plot = 'all_list'
fields = 'height, geopt, theta, tc, td2, rh, rh2, umet, vmet,
slp, pressure, u10m, v10m, wdir, wspd, wd10, ws10, cape, mcin,
lcl, lfc, cape, cin, clfr'
output_type = 'grads'
mercator_defs = .true.
/

&interp
interp_method = 1,
interp_levels = 1000., 975., 950., 925., 900., 850., 800., 750.,
700., 650., 600., 550., 500., 450., 400., 350., 300., 250.,
200., 150., 100.,
/
```


APÊNDICE B – NAMELISTS GRADE NORDESTE

➤ **namelist.wps**

```
&share
wrf_core = 'ARW',
max_dom = 2,
start_date = '2013-05-21_12:00:00','2013-05-21_12:00:00',
end_date = '2013-05-24_12:00:00','2013-05-24_12:00:00',
interval_seconds = 10800,
io_form_geogrid = 2,
/
```

```
&geogrid
parent_id = 1,1,
parent_grid_ratio = 1,3,
i_parent_start = 1,44,
j_parent_start = 1,35,
e_we = 160,271,
e_sn = 150,271,
geog_data_res = '5m' '2m',
dx = 18000,
dy = 6000,
map_proj = 'mercator',
ref_lat = -8.00,
ref_lon = -39.00
truelat1 = 0.0,
truelat2 = 0.0,
stand_lon = -39.00
geog_data_path = '/home/webpca/WRF/data/geog'
/
```

```
&ungrib
out_format = 'WPS',
prefix = 'GFS2',
/
```

```
&metgrid
fg_name = 'GFS2',
io_form_metgrid = 2,
opt_output_from_metgrid_path =
'/home/webpca/WRF/WRFV3/test/em_real'
/
```

➤ **namelist.input**

```
&time_control
run_days = 0,
run_hours = 48,
run_minutes = 0,
run_seconds = 0,
start_year = 2013,2013,
start_month = 05,05,
start_day = 21,21,
```

```

start_hour           = 12,12,
start_minute        = 00,00,
start_second        = 00,00,
end_year            = 2013,2013,
end_month           = 05,05,
end_day             = 24,24,
end_hour            = 12,12,
end_minute          = 00,00,
end_second          = 00,00,
interval_seconds    = 10800
input_from_file     = .true.,.true.,
history_interval    = 180, 60,
frames_per_outfile  = 1000,1000,
restart             = .false.,
restart_interval    = 5000,
io_form_history     = 2
io_form_restart     = 2
io_form_input       = 2
io_form_boundary    = 2
debug_level         = 0
/

```

```

&domains
time_step           = 108,
time_step_fract_num = 0,
time_step_fract_den = 1,
max_dom             = 2,
s_we                = 1, 1,
e_we                = 160, 271,
s_sn                = 1, 1,
e_sn                = 150, 271,
s_vert              = 1, 1,
e_vert              = 28, 28,
p_top_requested     = 5000,
num_metgrid_levels = 27,
num_metgrid_soil_levels = 4,
dx                  = 18000, 6000,
dy                  = 18000, 6000,
grid_id             = 1, 2,
parent_id           = 0, 1,
i_parent_start      = 1, 40,
j_parent_start      = 1, 35,
parent_grid_ratio    = 1, 3,
parent_time_step_ratio = 1, 3,
feedback            = 1,
smooth_option       = 0
/

```

```

&physics
mp_physics          = 3, 3,
ra_lw_physics       = 1, 1,
ra_sw_physics       = 1, 1,
radt                = 30,30,
sf_sfclay_physics  = 1, 1,
sf_surface_physics = 2, 2,
bl_pbl_physics      = 1, 1,

```

```

bldt = 0, 0,
cu_physics = 1, 1,
cudt = 5, 5,
isfflx = 1,
ifsnow = 0,
icloud = 1,
surface_input_source = 1,
num_soil_layers = 4,
sf_urban_physics = 0, 0,
maxiens = 1,
maxens = 3,
maxens2 = 3,
maxens3 = 16,
ensdim = 144,
/

&fdda
/

&dynamics
w_damping = 0,
diff_opt = 1,
km_opt = 4,
diff_6th_opt = 0, 0,
diff_6th_factor = 0.12, 0.12,
base_temp = 290.
damp_opt = 0,
zdamp = 5000., 5000.,
dampcoef = 0.2, 0.2,
khdif = 0, 0,
kvdif = 0, 0,
non_hydrostatic = .true., .true.,
moist_adv_opt = 1, 1,
scalar_adv_opt = 1, 1,
/

&bdy_control
spec_bdy_width = 5,
spec_zone = 1,
relax_zone = 4,
specified = .true., .false.,
nested = .false., .true.,
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/

```

➤ **namelist. ARWpost**

```
&datetime
start_date = '2013-05-21_12:00:00',
end_date = '2013-05-24_12:00:00',
interval_seconds = 10800,
tacc = 0,
debug_level = 0,
/

&io
io_form_input = 2,
input_root_name = './wrfout_d02_2013-05-21_12:00:00'
output_root_name = './wrfd2_2013052112'
plot = 'all_list'
fields = 'height, geopt, theta, tc, td2, rh, rh2, umet, vmet,
slp, pressure, u10m, v10m, wdir, wspd, wd10, ws10, cape, mcin,
lcl, lfc, cape, cin, clfr'
output_type = 'grads'
mercator_defs = .true.
/

&interp
interp_method = 1,
interp_levels = 1000., 975., 950., 925., 900., 850., 800., 750.,
700., 650., 600., 550., 500., 450., 400., 350., 300., 250.,
200., 150., 100.,
/
```

APÊNDICE C – NAMELIST GRADE NORTE

➤ **namelist.wps**

```
&share
wrf_core = 'ARW',
max_dom = 2,
start_date = '2013-05-21_12:00:00','2013-05-21_12:00:00',
end_date = '2013-05-24_12:00:00','2013-05-24_12:00:00',
interval_seconds = 10800,
io_form_geogrid = 2,
/
```

```
&geogrid
parent_id = 1,1,
parent_grid_ratio = 1,3,
i_parent_start = 1,50,
j_parent_start = 1,50,
e_we = 180,271,
e_sn = 180,271,
geog_data_res = '5m','2m',
dx = 18000,
dy = 18000,
map_proj = 'mercator',
ref_lat = -5.00,
ref_lon = -60.00,
truelat1 = 0.0,
truelat2 = 0.0,
stand_lon = -60.00,
geog_data_path = '/home/webpca/WRF/data/geog'
/
```

```
&ungrib
out_format = 'WPS',
prefix = 'GFS2',
/
```

```
&metgrid
fg_name = 'GFS2',
io_form_metgrid = 2,
opt_output_from_metgrid_path =
'/home/webpca/WRF/WRFV3/test/em_real'
/
```

➤ **namelist.input**

```
&time_control
run_days = 0,
run_hours = 48,
run_minutes = 0,
run_seconds = 0,
start_year = 2013,2013,
start_month = 05,05,
start_day = 21,21,
```

```

start_hour           = 12,12,
start_minute        = 00,00,
start_second        = 00,00,
end_year            = 2013,2013,
end_month           = 05,05,
end_day             = 24,24,
end_hour            = 12,12,
end_minute          = 00,00,
end_second          = 00,00,
interval_seconds    = 10800
input_from_file     = .true.,.true.,
history_interval    = 180, 60,
frames_per_outfile  = 1000,1000,
restart             = .false.,
restart_interval    = 5000,
io_form_history     = 2
io_form_restart     = 2
io_form_input       = 2
io_form_boundary    = 2
debug_level         = 0
/

```

```

&domains
time_step           = 108,
time_step_fract_num = 0,
time_step_fract_den = 1,
max_dom             = 2,
s_we                = 1, 1,
e_we                = 180,271,
s_sn                = 1, 1,
e_sn                = 180,271,
s_vert              = 1, 1,
e_vert              = 28, 28,
p_top_requested     = 5000,
num_metgrid_levels  = 27,
num_metgrid_soil_levels = 4,
dx                  = 18000, 6000,
dy                  = 18000, 6000,
grid_id             = 1, 2,
parent_id           = 0, 1,
i_parent_start      = 1, 50,
j_parent_start      = 1, 50,
parent_grid_ratio    = 1, 3,
parent_time_step_ratio = 1, 3,
feedback            = 1,
smooth_option       = 0
/

```

```

&physics
mp_physics          = 3, 3,
ra_lw_physics       = 1, 1,
ra_sw_physics       = 1, 1,
radt                = 30,30,
sf_sfclay_physics  = 1, 1,
sf_surface_physics  = 2, 2,
bl_pbl_physics      = 1, 1,

```

```

bldt = 0, 0,
cu_physics = 1, 1,
cudt = 5, 5,
isfflx = 1,
ifsnow = 0,
icloud = 1,
surface_input_source = 1,
num_soil_layers = 4,
sf_urban_physics = 0, 0,
maxiens = 1,
maxens = 3,
maxens2 = 3,
maxens3 = 16,
ensdim = 144,
/

&fdda
/

&dynamics
w_damping = 0,
diff_opt = 1,
km_opt = 4,
diff_6th_opt = 0, 0,
diff_6th_factor = 0.12, 0.12,
base_temp = 290.
damp_opt = 0,
zdamp = 5000., 5000.,
dampcoef = 0.2, 0.2,
khdif = 0, 0,
kvdif = 0, 0,
non_hydrostatic = .true., .true.,
moist_adv_opt = 1, 1,
scalar_adv_opt = 1, 1,
/

&bdy_control
spec_bdy_width = 5,
spec_zone = 1,
relax_zone = 4,
specified = .true., .false.,
nested = .false., .true.,
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/

```

➤ **namelist. ARWpost**

```
&datetime
start_date = '2013-05-21_12:00:00',
end_date = '2013-05-24_12:00:00',
interval_seconds = 10800,
tacc = 0,
debug_level = 0,
/

&io
io_form_input = 2,
input_root_name = './wrfout_d02_2013-05-21_12:00:00'
output_root_name = './wrfd2_2013052112'
plot = 'all_list'
fields = 'height, geopt, theta, tc, td2, rh, rh2, umet, vmet,
slp, pressure, u10m, v10m, wdir, wspd, wd10, ws10, cape, mcin,
lcl, lfc, cape, cin, clfr'
output_type = 'grads'
mercator_defs = .true.
/

&interp
interp_method = 1,
interp_levels = 1000., 975., 950., 925., 900., 850., 800., 750.,
700., 650., 600., 550., 500., 450., 400., 350., 300., 250.,
200., 150., 100.,
/
```


APÊNDICE D – NAMELISTS GRADE ALCÂNTARA

➤ **namelist.wps**

```
&share
wrf_core = 'ARW',
max_dom = 2,
start_date = '2013-05-21_00:00:00','2013-05-21_00:00:00',
end_date = '2013-05-24_00:00:00','2013-05-24_00:00:00',
interval_seconds = 10800,
io_form_geogrid = 2,
/
```

```
&geogrid
parent_id = 1,1,
parent_grid_ratio = 1,3,
i_parent_start = 1,48,
j_parent_start = 1,48,
e_we = 130,112,
e_sn = 130,112,
geog_data_res = '5m','2m',
dx = 18000,
dy = 18000,
map_proj = 'mercator',
ref_lat = -2.40,
ref_lon = -44.40,
truelat1 = 0.0,
truelat2 = 0.0,
stand_lon = -44.40,
geog_data_path = '/home/webpca/WRF/data/geog'
/
```

```
&ungrib
out_format = 'WPS',
prefix = 'GFS2',
/
```

```
&metgrid
fg_name = 'GFS2',
io_form_metgrid = 2,
opt_output_from_metgrid_path =
'/home/webpca/WRF/WRFV3/test/em_real'
/
```

➤ **namelist.input**

```
&time_control
run_days = 0,
run_hours = 48,
run_minutes = 0,
run_seconds = 0,
start_year = 2013,2013,
start_month = 05,05,
start_day = 21,21,
```

```

start_hour           = 00,00,
start_minute        = 00,00,
start_second        = 00,00,
end_year            = 2013,2013,
end_month           = 05,05,
end_day             = 24,24,
end_hour            = 00,00,
end_minute          = 00,00,
end_second          = 00,00,
interval_seconds    = 10800
input_from_file     = .true.,.true.,
history_interval    = 180, 60,
frames_per_outfile  = 1000,1000,
restart             = .false.,
restart_interval    = 5000,
io_form_history     = 2
io_form_restart     = 2
io_form_input       = 2
io_form_boundary    = 2
debug_level         = 0
/

```

```

&domains
time_step           = 108,
time_step_fract_num = 0,
time_step_fract_den = 1,
max_dom             = 2,
s_we                = 1, 1,
e_we                = 130, 112,
s_sn                = 1, 1,
e_sn                = 130, 112,
s_vert              = 1, 1,
e_vert              = 28, 28,
p_top_requested     = 5000,
num_metgrid_levels  = 27,
num_metgrid_soil_levels = 4,
dx                  = 18000, 6000,
dy                  = 18000, 6000,
grid_id             = 1, 2,
parent_id           = 0, 1,
i_parent_start      = 1, 48,
j_parent_start      = 1, 48,
parent_grid_ratio    = 1, 3,
parent_time_step_ratio = 1, 3,
feedback            = 1,
smooth_option       = 0
/

```

```

&physics
mp_physics          = 3, 3,
ra_lw_physics       = 1, 1,
ra_sw_physics       = 1, 1,
radt                = 30,30,
sf_sfclay_physics   = 1, 1,
sf_surface_physics  = 2, 2,
bl_pbl_physics      = 1, 1,

```

```

bldt = 0, 0,
cu_physics = 1, 1,
cudt = 5, 5,
isfflx = 1,
ifsnow = 0,
icloud = 1,
surface_input_source = 1,
num_soil_layers = 4,
sf_urban_physics = 0, 0,
maxiens = 1,
maxens = 3,
maxens2 = 3,
maxens3 = 16,
ensdim = 144,
/

&fdda
/

&dynamics
w_damping = 0,
diff_opt = 1,
km_opt = 4,
diff_6th_opt = 0, 0,
diff_6th_factor = 0.12, 0.12,
base_temp = 290.
damp_opt = 0,
zdamp = 5000., 5000.,
dampcoef = 0.2, 0.2,
khdif = 0, 0,
kvdif = 0, 0,
non_hydrostatic = .true., .true.,
moist_adv_opt = 1, 1,
scalar_adv_opt = 1, 1,
/

&bdy_control
spec_bdy_width = 5,
spec_zone = 1,
relax_zone = 4,
specified = .true., .false.,
nested = .false., .true.,
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/

```

➤ **namelist. ARWpost**

```
datetime
start_date = '2013-05-21_00:00:00',
end_date = '2013-05-24_00:00:00',
interval_seconds = 10800,
tacc = 0,
debug_level = 0,
/

&io
io_form_input = 2,
input_root_name = './wrfout_d02_2013-05-21_00:00:00'
output_root_name = './wrfd2_2013052100'
plot = 'all_list'
fields
='height,geopt,theta,tc,td2,rh,rh2,umet,vmet,slp,pressure,u10m,v
10m,wdir,wspd,wd10,ws10,cape,mcin,lcl,lf,cafe,cin,clfr'
output_type = 'grads'
mercator_defs = .true.
/

&interp
interp_method = 1,
interp_levels =
1000.,975.,950.,925.,900.,850.,800.,750.,700.,650.,600.,550.,500
.,450.,400.,350.,300.,250.,200.,150.,100.,
/
```