5 PRODUTO DESENVOLVIDO

O produto desenvolvido por este trabalho foi o roteiro passo a passo da metodologia utilizada, a seguir detalhado:

Especificação da Configuração de Execução WRF

A execução do WRF necessita de vários módulos que fazem tratamento e manipulação de dados geográficos e meteorológicos com o fim de produzir previsões numéricas de tempo (NWP). O sítio de referência do modelo apresenta o fluxo de dados e respectivos programas/sistemas em https://www2.mmm.ucar.edu/wrf/users/model_overview.html.

Para automatizar o processamento tanto para finalidades de operação quanto de pesquisa foram criados *scripts* usando linguagem *Bash Shell* (https://www.gnu.org/software/bash/) e comandos/programas disponíveis em sistemas *Linux*.

O conjunto de *scripts* definem o sistema ***model-wrf*** pode ser encontrado em **https://github.com/glcamillo/model-wrf** e foi disponibilizado sob licença MIT *(open source)*. Na página no *GitHub* há explicações e informações sobre a execução do modelo e como parametrizá-lo. Os *scripts* do modelo só contemplam a execução, de forma que uma instalação prévia do WRF já deve estar operacional. O sítio do WRF disponibilizado pelo laboratório *Mesoscale and Microscale Meteorology Laboratory* (MMM) da universidade UCAR MMM já contempla um tutorial de instalação: https://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php

O download dos códigos para a máquina local pode ser feito por meio dos seguintes comandos:

a)acesso ao endereço: https://github.com/glcamillo/model-wrf/archive/refs/heads/main.zip ou

b) usando o comando *git clone*:

$ *git clone* https://github.com/glcamillo/model-wrf.git

Na página do sistema no *GitHub* há informações sobre configuração do sistema, principalmente diretórios e variáveis de ambiente, para acomodar a estrutura de instalação do WRF.

Configuração básica para execução

O modelo WRF é extremamente configurável através de parâmetros. As configurações são ajustadas em arquivos próprios de cada módulo. Nos *scripts* do modelo, os arquivos de configuração ficam em subdiretórios dentro de *~/model-wrf.* Dependendo da escolha do usuário ao executar o *script* de rodada *runwrf.sh*, os arquivos de configuração são alterados e copiados para os diretórios de execução dos módulos.

Segue exemplo de execução básica, considerando:

- Inicialização do modelo:2015-07-24 00UTC;
- Previsão de 24 horas;
- Configuração de domínio B, conforme especificado nos requisitos do projeto;
- Número de processos MPI para execução do módulo WRF (módulo de integração numérica; demais módulos têm sua execução serial).

```
$ ulimit -s unlimited
$ cd ~/model-wrf
$ chmod u+x runwrf.sh
$ ./runwrf.sh -conf B -ts 2015-07-24-00 -ti 24 -np 2
```

Observações:

- O comando "*ulimit - s unlimited*" é usado para não restringir o tamanho de pilha para os processos em execução;
- **[-conf B]** Escolha da configuração de domínio B. Neste endereço são especificados as configurações básicas: https://github.com/glcamillo/model-wrf/blob/main/README.md#configurations-some-domain-configurations-available-to-be-used;
- **[-ts 2015-07-24-00]** A opção *-ts* especifica data - hora da inicialização do modelo;
- **[-ti 24]** Especifica o tempo de integração (previsão) que pode ser: 24, 48 ou 72 (em horas);
- **[-np 2]** Parâmetro Opcional (padrão: 1). Especifica o número de processos.

Algumas outras opções que podem ser úteis:

$ ./runwrf.sh -conf B -ts 2015-07-24-00 -ti 24 -np 2 -gti 3 --wrf-time-step 40 –use-static-geogrid

- **[-gti 3]** Parâmetro opcional (padrão: 3). A opção *gti* indica a resolução temporal dos dados globais. Caso você utilize o GFS padrão e deseje iniciar a inicialização com dados a cada três horas, deverá especificar o seguinte formato: *“-gti 3”*. O padrão do *script* já é este; portanto, somente se houver necessidade de valores de intervalo de tempos diferentes, essa opção deve ser utilizada;
- **[--wrf-time-step 40]** Parâmetro opcional (padrão: conforme configuração de execução). Este parâmetro ajusta o passo de tempo para o módulo de integração numérica *wrf.exe*. Cada configuração possui um valor padronizado, mas, caso seja necessário alterá-lo, pode-se especificar através da opção mencionada. Para a configuração B, o valor padrão é 60 s;
- **[--use-static-gegrid]** Parâmetro opcional (padrão: geração dos dados a cada execução). Esta opção permite o uso de arquivos estáticos gerados anteriormente para cada configuração. Assim, não é necessário executar o módulo *geogrid.exe,* que, embora não impacte significativamente no tempo de processamento, demanda muito espaço em disco para os arquivos geográficos globais. Os arquivos devem estar localizados no seguinte diretório do modelo: ~/model-wrf/config-domains/r_sul-RS-SC-2d;
- **Nomeação dos arquivos:** Os arquivos devem ser nomeados da seguinte forma: *geo_em.d01.nc* (para domínio 1), *geo_em.d02.nc* (para domínio interno 2) e assim por diante.

Alteração/configuração avançada de parâmetros de execução

A alteração de parâmetros pode ser feita em dois locais (arquivos):

a) No próprio arquivo de execução do modelo *(runwrf.sh)*. Por exemplo: a configuração de domínio B (região sul, centrada em Santa Catarina) possui as seguintes configurações:

```
        [bB]) # ****************************
              CONFIG_NAME="r_sul-RS-SC-2d"
                    export _MAX_DOMAIN=2
              export _WRF_TIME_STEP=60
              if [ -z ${_T_INTERVAL_OUTPUT_1} ]; then  # Not set by the user
               export _T_INTERVAL_OUTPUT_1=180  ## interval time: 3 h for DOMAIN
1
              fi
             export _T_INTERVAL_OUTPUT_2=60  ## interval time: 1 h for DOMAIN 2
              export _T_INTERVAL_OUTPUT_3=60  ## interval time: 1 h for DOMAIN 3
              export _PARENT_ID_2=1;  export _PARENT_ID_3=""
              export _I_PARENT_START_2=51 ; export _I_PARENT_START_3=""
              export _J_PARENT_START_2=71 ; export _J_PARENT_START_3=""
             export _GEODATA_RES_1=2m; export _GEODATA_RES_2=30s; export
_GEODATA_RES_3=""
              export _MAP_PROJECTION=lambert
              export _E_WE_1=150;  export _E_WE_2=391;  export _E_WE_3=""
              export _E_SN_1=160;  export _E_SN_2=371;  export _E_SN_3=""
              export _DX_1=10000
              export _DY_1=10000
              # Spatial resolutions of nests are calculated from parent_grid_ratio.
              # No more necessary.
              export _DX_2=2000; export _DX_3=""
              export _DY_2=2000; export _DY_3=""
              export _REF_LAT=-30.477;   export _REF_LON=-53.302
                    export  _TRUELAT1=-30.477;  export  _TRUELAT2=-30.477;  export
_STAND_LON=-53.302
              GLOBAL_DATE_TIME_INTERVAL=3   # 3 hours
              GLOBAL_DATA=gfs0p25   # GFS_0p50  GFS_0p25   NCEP_WRF_1km
              GLOBAL_DATA_SOURCE="gfs"
              export _PARENT_RATIO_2=5   # Best values: 3 or 5
              export _PARENT_RATIO_3=""   # Best values: 3 or 5
             export  _FEEDBACK=1 # 0 one-way no feedback    1 two-way w/ feedbak
(default)
              export  _SMOOTH=2   # Default=2
```

```
export _MP_PHYSICS_1=5; export _MP_PHYSICS_2=5;
export _MP_PHYSICS_3=5
export _BL_PBL_PHYSICS_1=1; export _BL_PBL_PHYSICS_2=1; export
_BL_PBL_PHYSICS_3=1
export _CU_PHYSICS_1=16; _CU_PHYSICS_2=16; _CU_PHYSICS_3=0;
export _E_VERT=41
# Number of vertical levels. The levels are automatically calculated
# (auto_levels_opt=2) stretching in lower and in top of the
# atmosphere. Minimum number of levels when dzstretch_s and
# dzstretch_u in namelist.input
# dzstretch(dzstretch_s-dzstretch_u)=1.2-1.06  => 41 for ptop=50
```

Também podem ser realizadas configurações específicas para os módulos de inicialização WPS. Os arquivos são *scripts* que interpretam/substituem variáveis de ambiente e geram os respectivos arquivos de configuração *(namelist)*.

- Arquivos presentes em *~/model-wrf/wps: executes_ungrib.sh* e *namelist.wps.sh.* Eles contém os parâmetros para definição de domínio.
- Arquivos *~/model-wrf/wrf: namelist.input.wrf.afwa-diags.sh* e *namelist.input.wrf.sh*. Os dois arquivos contém basicamente as mesmas configurações para o programa *real.exe* e *wrf.exe*, exceto que o primeiro inclui parâmetros para configurações avançadas de diagnósticos fornecidas pelo módulo AFWA (http://www2.mmm.ucar.edu/wrf/users/docs/AFWA_Diagnostics_in_WRF.pdf). Esses arquivos podem ser editados para alterar diversas configurações e parâmetros do modelo (física, camada limite, etc).

Processamento para extração de dados usando R (e CDO)

O modelo gera saídas GrADS que podem ser processadas por *scripts* próprios (presentes em *~/model-wrf/post_processing*). Para o propósito do projeto, foram usados *scripts* R e comandos CDO para extração de dados pontuais de precipitação. Os *scripts* desenvolvidos estão disponíveis em repositório *Git* público, no seguinte endereço: https://github.com/glcamillo/model-extract-and-plot. Para obter (fazer *download*) desse *script*, deve-se executar um dos seguintes comandos:

- https://github.com/glcamillo/model-extract-and-plot/archive/refs/heads/main.zip
- *git clone* https://github.com/glcamillo/model-extract-and-plot.git
- Acesso direto ao arquivo: https://github.com/glcamillo/model-extract-and-plot/blob/main/wrfout-extract-PRP-with-R-casos-Eliseo-for-24-48-72h.R
- Informações de como executar serão exploradas a seguir.
- O processamento via *scripts* R acessa arquivos NetCDF através das coordenadas de grade i e j do modelo WRF.

Portanto, algumas considerações importantes foram observadas:

a) **Obtenção prévia das coordenadas i, j** referentes à localização especificada (coordenadas de latitude/longitude). Para esse propósito, você pode usar o programa **ncview** , que permite obter correspondências entre os dois tipos de coordenadas.

b) O projeto exige uma mudança média sobre uma área quadrada de aproximadamente 10 quilômetros de lado. Assim, o processamento da área por área exigiu a seguinte configuração:

- A partir do ponto central i, j do WRF, foram obtidos dados de outros 24 pontos distantes dois quilômetros entre si, de modo a compor uma abrangência aproximada de 10 km.
- Um resumo estatístico foi fornecido pela função `summary()` do R, contendo: mínimo, média, mediana, máximo e quartis.

c) A execução de *scripts* em R requer a instalação dos pacotes básicos conforme as informações disponíveis em https ://cran .r -project .org/ . Além da instalação básica, é necessário complementar o pacote ***r-cran-rnetcdf*** (pacote GNU R que fornece uma interface R para conjuntos de dados NetCDF) ao utilizar distribuições *Linux* derivadas do *Debian/Ubuntu*.

d) Aqui foram fornecidos *scripts* e comandos do CDO para a remoção de dados pontuais de arquivos NetCDF; no entanto, outras linguagens e recursos também podem ser utilizados.

Extração de dados por área

O seguinte *script* extrai dados de arquivos NetCDF produzidos como saída WRF e apresenta dados de precipitação por ponto e um sumário. O *script* foi programado em R e é parametrizado com:

- **[./wrfout-extract-PRP-with-R-casos-Eliseo-for-24-48-72h.R]**: caminho e nome do arquivo do script em R.

- **[CAMINHO_E_NOME_ARQUIVO_FINAL_WRF.nc]**: arquivo de saída do WRF em formato NetCDF, por exemplo, `wrfout_d02_2016-01-12_00.nc` (este nome foi renomeado do arquivo de saída do WRF).

- **[NOME_ARQUIVO_SAIDA]**: qualquer nome que indique especificamente os dados contidos. Por exemplo: `caso-01-20160114-riodocampo-d2-72h-`AREA.

- **[COORD_X]**: coordenada central da área que corresponde à coordenada **i** da grade do modelo WRF.

- **[COORD_Y]**: coordenada central da área que corresponde à coordenada **j** da grade do modelo WRF.

- **[TEMPO_PREVISAO]**: tempo de previsão presente no arquivo.

- **[TIPO_PROCESSAMENTO]**: são duas opções: ***POINT*** ou **AREA**, sendo que os processamentos foram realizados exclusivamente com a opção **AREA**.

Exemplos de comandos e resultados (em tela e que foram escritos em arquivos):

72h:./wrfout-extract-PRP-with-R-casos-Eliseo-for-24-48-72h.R wrfout_d02_2016-01-12_00.nc caso-01-20160114-riodocampo-d2-72h-AREA 278 244 72 AREA

```
 [1] 16.874371  8.975282  7.065782  3.649144  2.660214 17.417940 12.423160
 [8]  9.204975  3.703314  2.507992 12.673026  7.464438  4.392147  2.755603
[15]  3.059908 10.562754  6.255880  4.059951  3.111260  2.334455 10.226622
[22]  6.766695  5.202722  4.574454  3.505717
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.334   3.506   5.203   6.857   9.205  17.418
```

48h:./wrfout-extract-PRP-with-R-casos-Eliseo-for-24-48-72h.R wrfout_d02_2016-01-13_00.nc caso-01-20160114-riodocampo-d2-48h-AREA 278 244 48 AREA

[1] 4.4074465 3.1394828 2.5610304 1.9443040 0.7763767 3.4903296 3.0733880

[8] 2.0852000 1.3154784 0.8594064 3.6803546 1.9112629 1.4047916 1.4846264

[15] 0.8351218 2.8494633 1.9202628 1.7269880 1.1654787 0.9326722 2.2327309

[22] 1.5088931 1.5036165 1.4482994 0.6938900

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.6939 1.3155 1.7270 1.9580 2.5610 4.4074

24h:./wrfout-extract-PRP-with-R-casos-Eliseo-for-24-48-72h.R wrfout_d02_2016-01-14_00.nc caso-01-20160114-riodocampo-d2-24h-AREA 278 244 24 AREA

[1] 13.663426 10.793596 7.028851 3.543419 1.921384 13.992137 10.459312

[8] 6.698219 3.813109 2.323862 10.709391 9.689279 6.264982 4.876913

[15] 4.559804 9.402451 7.977650 7.245204 4.632228 3.708069 11.038969

[22] 7.266530 5.801353 4.633793 3.514930

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.921 4.560 6.698 7.022 9.689 13.992

Extração de dados por ponto

A extração de dados pontuais, embora não tenha sido realizada como parte deste trabalho, pode ser feita por meio de um *script* próprio em R, conforme descrito a seguir, ou usando comandos do CDO. A diferença é que o *script* em R requer as coordenadas i e j da grade do modelo, enquanto o comando do CDO recebe o ponto por meio de coordenadas geográficas.

Exemplo de execução R para o seguinte caso:

1- RIO DO CAMPO - 2016-01-14 - Verão - 130,6 mm

-- Coordenada WRF central (ponto): i=x=278 j=y=244

-- Previsão: 24 horas

./wrfout-extract-PRP-with-R-timeinterval.R wrfout_d02_2016-01-14_00.nc caso-1-24h 278 244 24 POINT 1

```
[1] "wrfout_d02_2016-01-14_00.nc" "caso-1-24h"
[3] "278"                         "244"
[5] "24"                          "POINT"
[7] "1"
[1] 390 370  25 390 370  25
[1] "Coordenada i=x:  278"
[1] "Coordenada j=y:  244"
[1]   0  60 120 180 240 300
[1] "minutes since 2016-01-14 00:00:00"
[1] "mm"
[1] 390 370  25
[1] "mm"
[1] 390 370  25
[1] "24h:  6.26498198509216  Result:  6.26498198509216"
[1] "Hour:  1 PRP in mm:  0.372971897308481"
[1] "Hour:  2 PRP in mm:  0.764474666382419"
[1] "Hour:  3 PRP in mm:  1.91679807379842"
[1] "Hour:  4 PRP in mm:  0.301705598831177"
[1] "Hour:  5 PRP in mm:  0.186107993125916"
[1] "Hour:  6 PRP in mm:  0.208443880081177"
[1] "Hour:  7 PRP in mm:  0.0297293663024902"
[1] "Hour:  8 PRP in mm:  0.000838994979858398"
[1] "Hour:  9 PRP in mm:  0.28931736946106"
[1] "Hour:  10 PRP in mm:  0.323705673217773"
[1] "Hour:  11 PRP in mm:  0.0163636207580566"
[1] "Hour:  12 PRP in mm:  0.246755123138428"
[1] "Hour:  13 PRP in mm:  0.260291576385498"
[1] "Hour:  14 PRP in mm:  0.0908946990966797"
[1] "Hour:  15 PRP in mm:  0.172902345657349"
[1] "Hour:  16 PRP in mm:  0.311793804168701"
[1] "Hour:  17 PRP in mm:  0.21945858001709"
[1] "Hour:  18 PRP in mm:  0.0646305084228516"
[1] "Hour:  19 PRP in mm:  0.033172607421875"
[1] "Hour:  20 PRP in mm:  0.0995430946350098"
[1] "Hour:  21 PRP in mm:  0.0401544570922852"
```

[1] "Hour: 22 PRP in mm: 0.00932884216308594"
[1] "Hour: 23 PRP in mm: 0.0795516967773438"
[1] "Hour: 24 PRP in mm: 0.226047515869141"
[1] 24 2
[1] 24
[1] 2

Para a extração de dados usando o CDO, podem ser utilizados comandos disponíveis diretamente do pacote CDO, instalado previamente. Neste caso, os comandos extraem dados de precipitação pontuais, tendo como referência a coordenada geográfica (latitude/longitude) específica do local

Exemplos de comandos:
- Para coordenada Lat/Lon: -28.5325/-49.315278

- Para previsão de 24 horas:
cdo -v select,name=RAINNC wrfout_d02_2016-07-15_00.nc wrf-output-d2-RAINNC.nc
cdo -outputtab,lon,lat,date,time,value -remapnn,lon=-49.315278_lat=-28.5325 wrf-output-d2-RAINNC.nc > caso-26-20160715-urussanga-D2-24h-PRP-from-cdo-49.31-28.53.txt

Cálculo das precipitações observadas correspondentes ao percentil 99,9%

O seguinte *script* em *Python* foi utilizado para o cálculo das precipitações observadas correspondentes ao percentil de 99,9% para cada mesorregião e estação do ano:

```
#-----------------------------------------------------------------------------------------------------
# MESTRADO PROFISSIONAL EM CLIMA E AMBIENTE IFSC - FLORIANOPOLIS
#   Author: Eliseo Breda  ---------- DATA:08/2023
## Este Script calcula o percentil 99,9% para cada Mesoregião / Estação de SC
#-----------------------------------------------------------------------------------------------------
#Importando Bibliotecas
import pandas as pd
```

```
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
plt.rcParams.update({'font.size': 12, 'font.family':'serif'})
#-------------------------------------------------------------------------------------
#Carregar arquivo xlsx, para ler o dataset e remover os valores zeros
xls = pd.ExcelFile('../dados/tabela_Eliseo.xlsx')
# arquivo com muitas abas, citar a aba de interesse no Dataframe (df)
df = pd.read_excel(xls, sheet_name='Dados detalhados')
df_remove = df.loc[(df['POBS'] <= 0)]
df
#Especificar as colunas que deseja remover ## Usar este passo caso queira remover colunas no DataFrame
colunas_para_remover = ['Data.1', 'POBS.1', 'Percentil?', 'Unnamed: 9', 'Unnamed: 10']
# Remova as colunas especificadas,
colunas_existentes = [coluna for coluna in colunas_para_remover if coluna in df.columns]
df = df.drop(columns=colunas_existentes)
df # este comando mostra o Dataframe
# Outra maneira de Exibir o DataFrame, após a remoção das colunas
print("\nDataFrame após a remoção das colunas:")
print(df.head())
#A funçaõ dropna remove linhas que contenham valores ausentes na coluna 'Estação do Ano'.
df = df.dropna(subset=['Estação do Ano'])
###-----------------------------------------------------------------------------------------------------
#Aqui o código itera sobre as mesorregiões e as estações do ano presentes nos dados.
#Combina a mesorregião e estação do ano, e calcula o quantil 0.999 (99.9º percentil) dos valores da coluna 'POBS'
#Esses percentis são armazenados em um dicionário chamado 'percentis', onde a chave é uma string concatenada
#da mesorregião e estação do ano, e o valor é o quantil calculado.
###-----------------------------------------------------------------------------------------------------
percentis = {}
```

```
mesoregioes = list(set(df['Mesoregião']))
estacoes = list(set(df['Estação do Ano']))
for meso in mesoregioes:
    df_meso = df.loc[(df['Mesoregião'] == meso)]
    for estacao in estacoes:
        df_meso_estacao = df_meso.loc[(df['Estação do Ano'] == estacao)]
        quantile = df_meso_estacao['POBS'].quantile(0.999)
        nome = f"{meso}_{estacao}"
        percentis[nome] = quantile
      #O dicionário 'percentis' é retornado.
percentis
###-----------------------------------------------------------------------------------------------------------
# Criar DataFrame a partir do dicionário para melhor visualizar os percentis
df_quantis = pd.DataFrame(list(quantis.items()), columns=['Mesorregião_Estação',
'Quantil'])
df_quantis
#Salva o Dataframe em planilha excel.xlsx
df_quantis.to_excel("../dados/perc_eliseo.xlsx")
###-----------------------------------------------------------------------------------------------------------
```