Readme for GCEM Data and Code

1 Data

1.1 Observed Precipitation Data

1.1.1 /DataSoftware/01Observed_precipitation_data/2019071612

Hourly precipitation from 00:00 to 12:00 UTC on July 16, 2019 for Case 1

surfr01h.nc
surfr02h.nc
surfr03h.nc
surfr04h.nc
surfr05h.nc
surfr06h.nc
surfr07h.nc
surfr08h.nc
surfr09h.nc
surfr10h.nc
surfr11h.nc
surfr12h.nc

1.1.2 /DataSoftware/01Observed_precipitation_data/2020061312

Hourly precipitation from 00:00 to 12:00 UTC on June 13,2020 for Case 2

surfr01h.nc
surfr02h.nc
surfr03h.nc
surfr04h.nc
surfr05h.nc
surfr06h.nc
surfr07h.nc
surfr08h.nc
surfr09h.nc
surfr10h.nc
surfr11h.nc
surfr12h.nc

1.2 Forecasted Precipitation Data

1.2.1 /DataSoftware/02Forecasted_precipitation_data/2019071612

Data for Case 1 during 00:00-12:00 UTC on July 16, 2019

WRF3.2019071600000.nc (initial field at 12:00 UTC on July 16, 2019)

WRF3.2019071600012.nc (12-hour accumulated precipitation during 00:00–12:00 UTC on July 16, 2019)

1.2.2 /DataSoftware/02Forecasted_precipitation_data/2020061312

Data for Case 2 during 00:00-12:00 UTC on June 13,2020

WRF3.2020061300000.nc (initial field at 12:00 UTC on June 13,2020)

WRF3.2020061300012.nc (12-hour accumulated precipitation during 00:00–12:00 UTC on June 13,2020)

2 Code and Configuration Files

2.1 GCEM of grid precipitation forecast data (/DataSoftware/03Software_Configuration_Results_of_GCEM)

pastonc6hd2.f90 Main program, reads observed and forecasted precipitation data, performs GCEM verification, and outputs result files.

module_skinput.f90 Subprogram, module for reading one or more observed precipitation grid file

module_ybinput.f90 Subprogram, module for reading the start (or end) forecasted precipitation grid file

mod_uxpasid2.f90 Subprogram, module for used to perform GCEM verification on forecasted data

module_outnc.f90 Subprogram, module for outputting the verification results in netCDF file format

compilePAS10mmd2.sh Used to compile source files to generate executable file under Linux

r12hfile.txt Configuration file, used to specify the latitude and longitude range for data source and verification

pastonc6hd2.exe Executable file

2.2 TS of grid precipitation forecast data (/DataSoftware/04Software_Configuration_Results_of_TS-Score)

tsmain01.f90 Main program, reads observed and forecasted precipitation data, performs TS verification, and outputs result files.

module_skinput.f90 Subprogram, module for reading one or more observed precipitation grid file

module_ybinput.f90 Subprogram, module for reading the start (or end) forecasted precipitation grid file

module_uxtsiTure.f90 Subprogram, module for used to perform TS verification on forecasted data

compileTS.sh Used to compile source files to generate executable file under Linux

r12hfile.txt Configuration file, used to specify the latitude and longitude range for data source and verification

tsmain01.exe Executable file

2.3 PAS mini-program (/DataSoftware/05Software_of_PAS)

pas10ux.f90 Main program, used to perform PAS verification on single point precipitation forecast

mod_uxpasid2.f90 Subprogram, Module for PAS of single point forecast

compilePAS10ux.sh Used to compile source files to generate executable file under Linux

pas10ux.exe Executable file

3 Output files

3.1 GCEM verification results (/DataSoftware/03Software_Configuration_Results_of_GCEM/Results_GCEM)

rainverd2019071612012.nc Result file in netCDF format

rainverd2019071612012.nc.txt Result explanation file in netCDF format

rainverd2020061312012.nc Result file in netCDF format

rainverd2020061312012.nc.txt Result explanation file in netCDF format

3.2 TS verification results (/DataSoftware/04Software_Configuration_Results_of_TS-Score/Results_TS)

ts12h2019071612.txt TS result file in text format

ts12h2020061312.txt TS result file in text format

4 Compiling Environment

The verification program runs in a UNIX environment and requires the intel compiler (v2017) and the netCDF (v4.6.1) support library

UNIX Environment Settings

```
# .bashrc
module load intel/intel-compiler-2017.5.239
module load intelmpi/2019.6.154
```

export F90=ifort
export NETCDF=/public/software/mathlib/netcdf/4.6.1_intel-2017_mpi-2017_hdf5-1.8.20-intel2017
export NETCDF_LIB=$NETCDF/lib
export NETCDF_INC=$NETCDF/include
export PATH=$NETCDF/bin:$PATH
export LD_LIBRARY_PATH=$NETCDF/lib:$LD_LIBRARY_PATH

5 Compiling and Running Steps

5.1 The steps for case 1 during 00:00–12:00 UTC on July 16, 2019

1. Creating an installation and running sub-directory
   mkdir p2019
2. Copying data sources, code files and configuration files to this directory
3. Running in this directory
   ./compilePAS10mmd2.sh Compile to generate executable file (pastonc6hd2.exe)
   ./compileTS.sh Compile to generate executable file (tsmain01.exe)
4. Modifying the configuration file (r12hfile.txt)
   Mainly modifying the data source path for lines 4, 19, and 23
5. Run the executable files
   ./pastonc6hd2.exe > outnc12hd22019071612.txt
   Creating the GCEM result file (rainverd2019071612012.nc), procedure file (outnc12hd22019071612.txt)
   ./tsmain01.exe >ts12h2019071612.txt
   Creating the TS result and procedure file (ts12h2019071612.txt)

5.2 The steps for case 2 during 00:00–12:00 UTC on June 13,2020

1. Creating an installation and running sub-directory
   mkdir p2020
2. Copying data sources, code files and configuration files to this directory
3. Running in this directory
   ./compilePAS10mmd2.sh Compile to generate executable file (pastonc6hd2.exe)
   ./compileTS.sh Compile to generate executable file (tsmain01.exe)
4. Modifying the configuration file (r12hfile.txt)
   Mainly modifying the data source path for lines 4, 19, and 23
5. Run the executable files
   ./pastonc6hd2.exe > outnc12hd22020061312.txt
   Creating the GCEM result file (rainverd2020061312012.nc), procedure file (outnc12hd22020061312.txt)
   ./tsmain01.exe >ts12h2020061312.txt
   Creating the TS result and procedure file (ts12h2019071612.txt)

5.3 The steps for PAS mini-program

1. Creating an installation and running sub-directory
   mkdir pas
2. Copying code files and configuration files to this directory
3. Running in this directory
   ./compilePAS10ux.sh Compile to generate executable file (pastonc6hd2.exe)

4. linking the executable file as pas

ln -sf pas10ux.exe pas

5. Running the PAS mini-program

for example：./pas 15 20

Parameter 1: 15 represents observed precipitation

Parameter 2: 20 represents forecasted precipitation

Output: 0.895 1

The following instructions for specific usage:

pas *rainsk rainyb [level]*

Input parameters

Parameter 1 (*rainsk*): observed precipitation (mm)

Parameter 2 (*rainyb*): forecasted precipitation (mm)

Parameter 3 (*level*): Specifing magnitude (Optional, default to ≥0.1mm)

Onput parameters

Parameter 1 (*ipas*): Pas score value (0-1) or correct value of no precipitation forecast (1);

-999.000 represents default.

Parameter 2 (*iTure*):

0 indicates that the rating is correct for a no precipitation forecast;

1 indicates a PAS score of ≥ the specified magnitude;

9 indicates that it is not in the no precipitation test, nor is it the verification the specified magnitude;

-999 indicates default.

6 Module code main interface description

6.1 skinput()

```
subroutine skinput(skfile,skfilenum,rain,gridskx,gridsky,longitude,latitude)
  integer,intent(in) :: skfilenum
  character(len=200),dimension(skfilenum),intent(in) :: skfile
  real,dimension(:,:),allocatable,intent(out) :: rain
  integer,intent(out) :: gridskx,gridsky
```

usage: Read a set of observed precipitation data files and output grid accumulated precipitation

skfile, A set of filenames that are arrays of strings (input)

skfilenum, Number of files (input)

rain, Accumulated precipitation, rain(nx,ny) (output)

gridskx, grid points, nx (output)

gridsky, grid points, ny (output)

gridlon, Longitude array, gridlon(nx) (output)

gridlat, Latitude array, gridlat(ny) (output)

6.2 ybinput()

```
subroutine ybinput(ybfile,apcp,gridybx,gridyby,gridyblon,gridyblat)
  character(len=200),intent(in) :: ybfile
  real,dimension(:,:),allocatable,intent(out) :: apcp,gridyblat,gridyblon
  integer,intent(out) :: gridybx,gridyby
```

usage: Read a set of forecasted precipitation data files and output forecast grid precipitation

ybfile, Forecast file (input)
apcp, forecasted precipitation array, apcp(nx, ny) (output)
gridybx, Number of grid points for forecast data, nx (output)
gridyby, Number of grid points for forecast data, ny (output)
gridyblon, Longitude of forecast data, gridyblon(nx, ny) (output)
gridyblat, Latitude of forecast data, gridyblat(nx, ny) (output)

6.3 uxpasid2()

```
subroutine uxpasid2(ui,xi,level,pas,iTure,iclass,ieps)
  real,intent(in)   ::   ui,xi,level
  real,intent(out) :: pas,ieps
  integer,intent(out) :: iTure,iclass
```

usage: Read in the observed and forecasted precipitation, and output the PAS score result

rainsk, Observed precipitation (input)
rainyb, Forecasted precipitation (input)
level, Specifing magnitude (input)
ipas, Pas score value (0-1) or correct value of no precipitation forecast (1)
iTure, 0 indicates that the rating is correct for a no precipitation forecast;
1 indicates a PAS score of ⩾ the specified magnitude;
9 indicates that it is not in the no precipitation test, nor is it the verification the specified magnitude;
-999 indicates default.
iclass, 0 indicates the category (no precipitation forecast is correct)
1 indicates the category of insufficient precipitation forecast (observation u<10mm)
2 indicates the category of excessive(or equal) precipitation forecast(observation u<10mm)
3 indicates the category of insufficient precipitation forecast (observation u⩾10mm)
4 indicates the category of excessive(or equal) precipitation forecast(observation u⩾10mm)
-999 indicates default.
ieps, 0 indicates the forecasted and observed precipitation are equal
<0 indicates insufficient precipitation forecast
>0 indicates excessive precipitation forecast
-999 indicates default.

6.4 outpasnc()

```
subroutine outpasnc(title,vtime,vhour,gridncx,gridncy,gridnclon,gridnclat,rainncsk,rainncyb,&
                    pasc,pas01,pas10,pas25,pas50, pas2p5,pas5,pas15, pascnc,pasnc01,pasnc10,&
```

```
                        pasnc25,pasnc50,pasnc2p5,pasnc5,pasnc15,ipsnc,epsnc,iepsnc,ips,eps,ieps)
  character(len=10),intent(in) :: vtime,title
  integer,intent(in) :: vhour,gridncx,gridncy
  character(len=10) :: chour
  real,dimension(gridncx),intent(in) :: gridnclon
  real,dimension(gridncy),intent(in) :: gridnclat
  real,dimension(gridncx,gridncy),intent(in) :: rainncsk,rainncyb,pascnc,pasnc01
  real,dimension(gridncx,gridncy),intent(in) :: pasnc10,pasnc25,pasnc50,ipsnc,epsnc,iepsnc
  real,dimension(gridncx,gridncy),intent(in) ::pasnc2p5,pasnc5,pasnc15
  real,intent(in) :: pasc,pas01,pas10,pas25,pas50,ips,eps,ieps,pas2p5,pas5,pas15

usage: Output data to a netCDF format file
      title,      File name tag (d)
      vtime,      time string (yyyymmdddhh, eg. 2019071612)
      vhour,      Accumulated precipitation duration (12)
      gridncx,        x grid points (240)
      gridncy,        y grid points (200)
      gridnclon,      x grid points longitude array
      gridnclat,      y grid points latitude array
      rainncsk,       Observed precipitation
      rainncyb,       Forecasted precipitation
      pasc,           PASC
      pas01,          ⩾0.1mm PAS
      pas10,          ⩾10mm PAS
      pas25,          ⩾25mm PAS
      pas50,          ⩾50mm PAS
      pas2p5,         ⩾2.5mm PAS
      pas5,           ⩾5mm PAS
      pas15,          ⩾15mm PAS
      pascnc,         PASC array
      pasnc01,        ⩾0.1mm PAS array
      pasnc10,        ⩾10mm PAS array
      pasnc25,        ⩾25mm PAS array
      pasnc50,        ⩾50mm PAS array
      pasnc2p5,       ⩾2.5mm PAS array
      pasnc5,         ⩾5mm PAS array
      pasnc15,        ⩾15mm PAS array
      ipsnc,          IPS array
      epsnc,          EPS array
      iepsnc,         IEPS array
      ips,            IPS
      eps,            EPS
      ieps            IEPS
```