

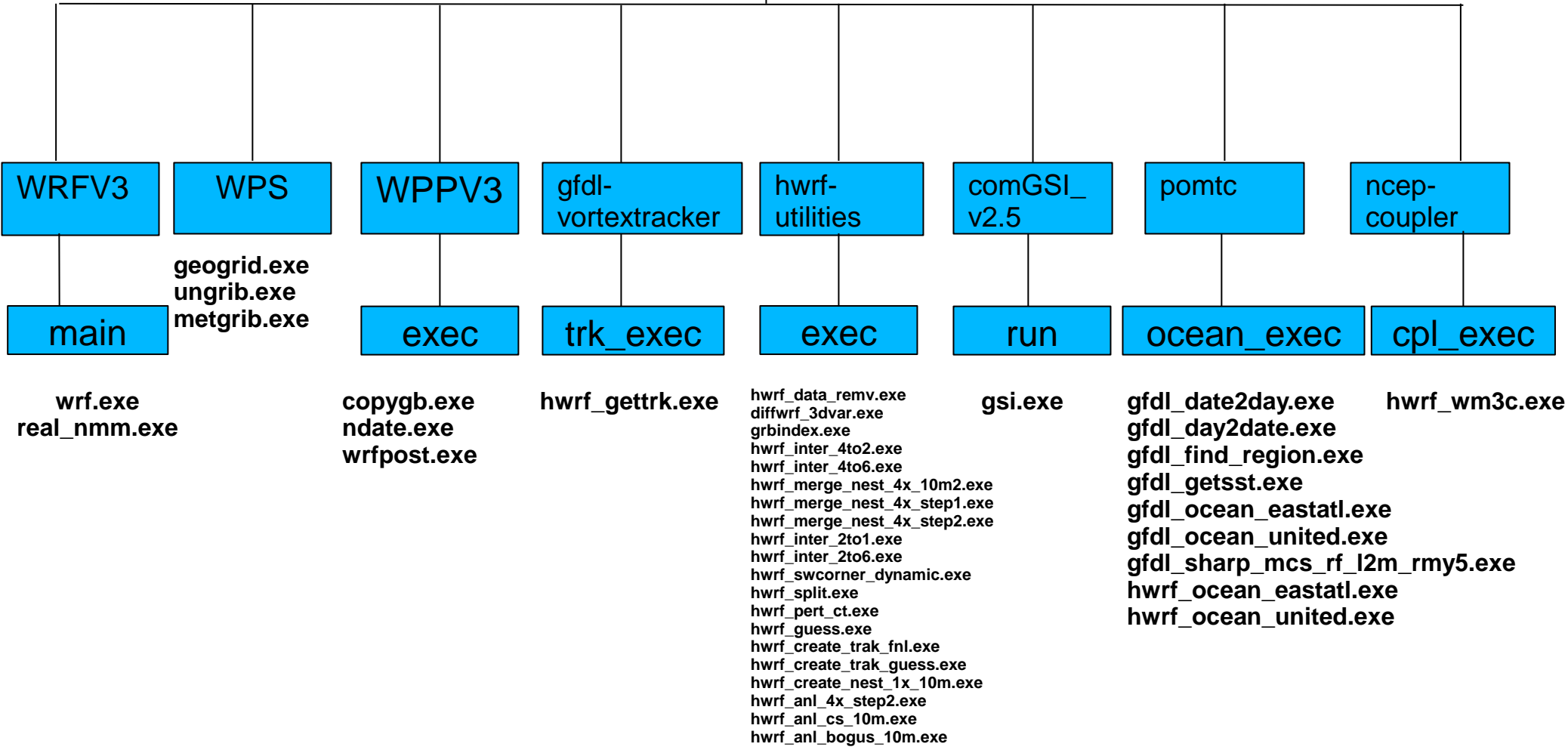
HWRF: Setup and Run

Shaowu Bao
CIRES/CU and NOAA/ESRL/GSD

WRF For Hurricanes Tutorial
Boulder, CO
April 28 2011

Source structure and executables

`${HOME}/HWRF/src`



Scripts and namelists

`${HOME}/HWRF/src`

WRFV3

WPS

WPPV3

gfdl-
vortextracker

hwrf-
utilities

comGSI_
v2.5

pomtc

ncep-
coupler

scripts

hwrfdomain.ksh
geogrid.ksh
ungrib.ksh
metgrid.ksh
real.ksh
wrf.ksh
relocate_stage3.ksh
relocate_stage2.ksh
relocate_stage1.ksh
run_gsi.ksh
merge.ksh
run_wrfpost
tracker.ksh
run_grads

parm

hwrf_namelist.wps
namelist_ghost.input
namelist_analysis.input
namelist_main.input
gsi_namelist.input

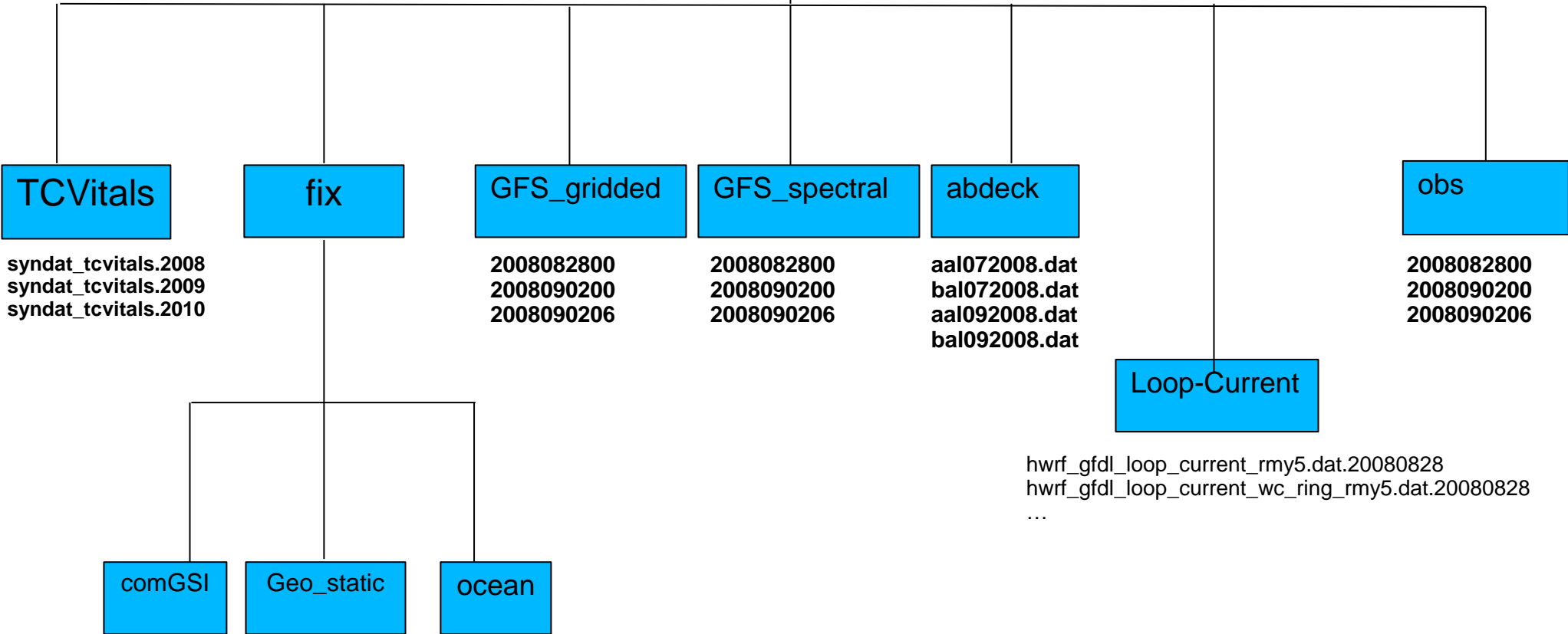
ocean_scripts

gfdl_pre_ocean_sortvit.sh
kickit00_region.sh
kickit01_sharpn.sh
kickit02_getsst.sh
kickit03_phase3.sh
kickit04_phase4.sh



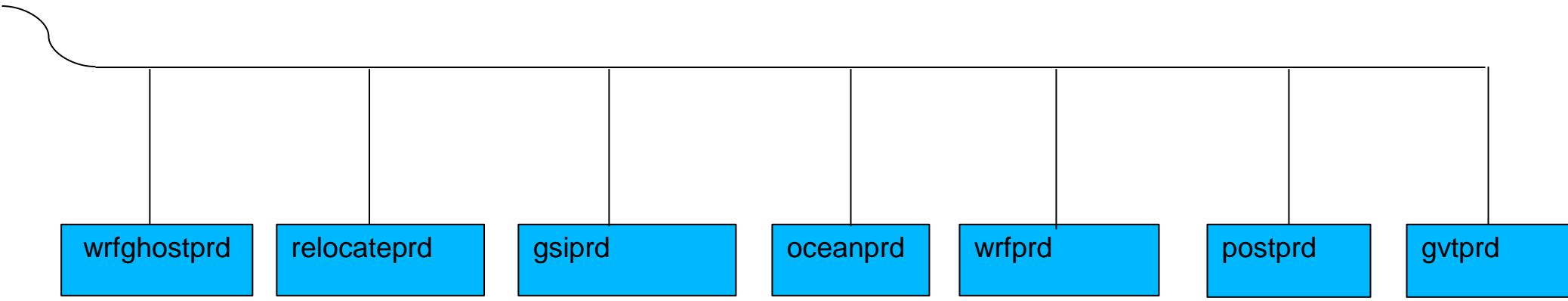
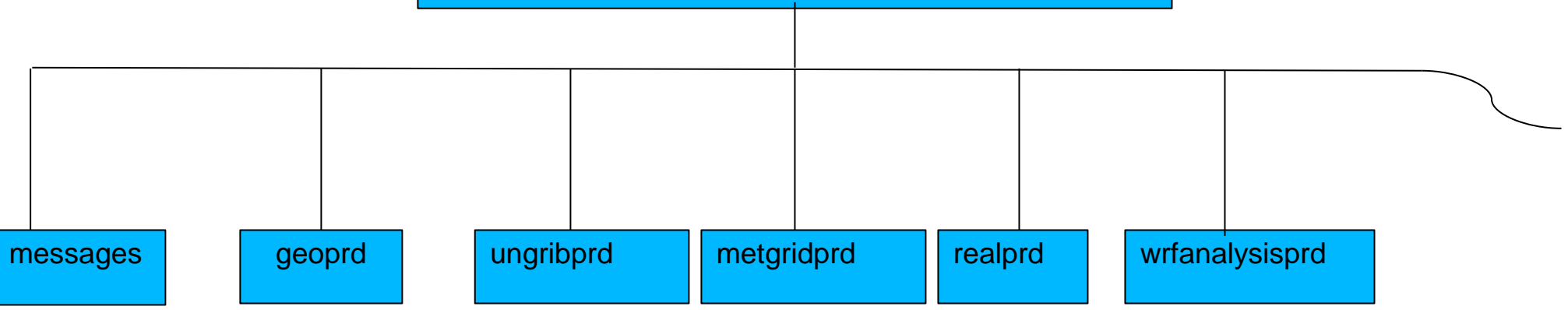
Datasets

/ptmp/HurrTutorial/datasets



Output directory structure

`/ptmp/$USER/HWRF/07L/2008082800`



Outline

→ Setup HWRF domain

→ Prepare GFS data

→ WRFV3 initialization

→ Vortex initialization

→ GSI

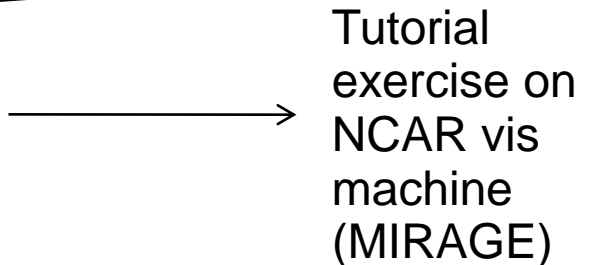
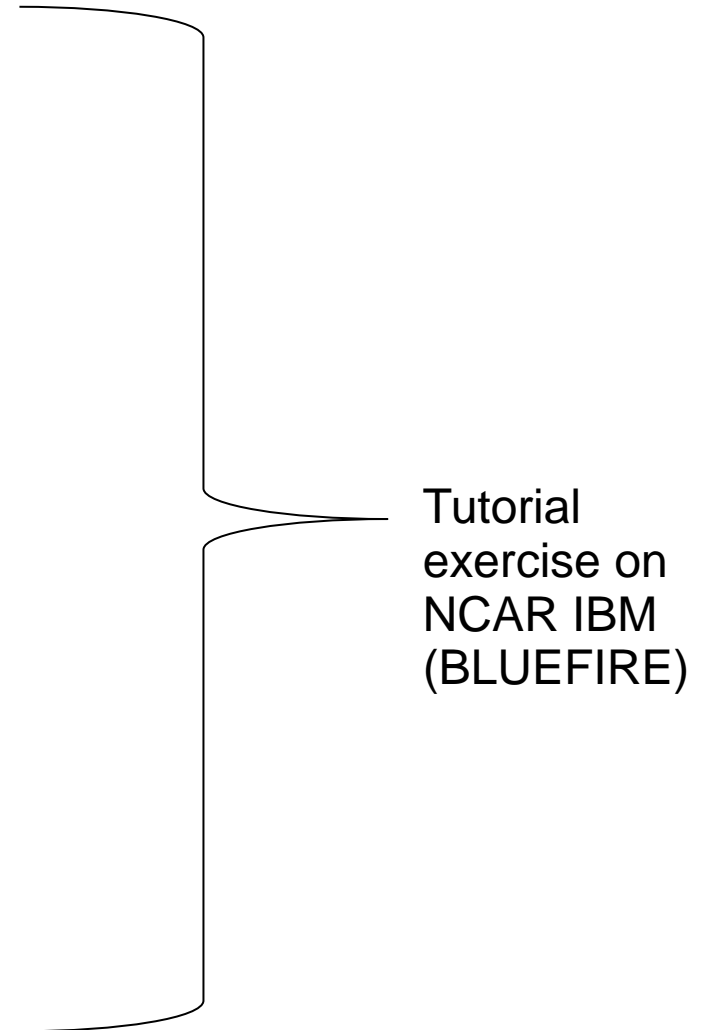
→ Ocean initialization

→ Coupled forecast

→ WRF post-processor

→ GFDL vortex tracker

→ WPP output, Track and intensity plotting



Set up HWRF domain

edit WPS namelist

\$HOME/HWRF/src/hwrf-utilities/parm/hwrf_namelist.wps

Start_date and
end_date

```
&share
wrf_core = 'MMM',
max_dom = 2,
start_date = '2008-08-28_00:00:00', '2008-08-28_00:00:00'
end_date = '2008-08-28_06:00:00', '2008-08-28_06:00:00'
interval_seconds = 21600,
io_form_geogrid = 2,
/

&geogrid
parent_id = 1,1,
parent_grid_ratio = 1,3,
i_parent_start = 1,99,
j_parent_start = 1,201,
e_we = 216,60,
e_sn = 432,100,
geog_data_res = '10m', '10m',
dx = 0.18,
dy = 0.18,
map_proj = 'rotated_ll',
ref_lat = DOMAINLAT,
ref_lon = DOMAINLON,
geog_data_path = '/ptmp/HurrTutorial/datasets/fix/geo_static',
opt_geogrid_tbl_path = 'OPTGGRIDPATH',
```

geo_data_path



Set up HWRF domain

edit script

`$HOME/HWRF/src/hwrf-utilities/scripts/hwrfdomain.ksh`

Storm ID

TC Vitals

WPS namelist

Path to output

```
#!/bin/ksh
# Set vars
SID=07L
TCVITALS=/ptmp/HurrTutorial/datasets/Tcvitals
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/hwrf_namelist.wps
# Get the initialization time from namelist.wps by the following steps
# 1. Grab the start_date line
# 2. Extract the part after the = sign
# 3. Delete all "-_:', ' and space characters
# 4. Extract the yyyymmddhh portion (throw away mm and ss)
YYYYMMDDHH=`cat ${NAMELIST} | grep start_date \
                | cut -d"=" -f 2 \
                | tr -d "[-_:', ']" \
                | cut -c1-10`
# Set base path of output directory
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYYMMDDHH}
# Print out information about this run
echo
echo "HWRF Domain parameter settings"
```



Set up HWRF domain

run *hwrfdomain.ksh* to define domain center location

In hwrf script directory \$HOME/HWRF/src/hwrf-utilities/scripts

hwrfdomain.ksh

verify the run is successful:

In directory /ptmp/\$USER/HWRF/07L/2008082800/messages:
find the following files:

domain.center

storm.center

tcvital

tcvital.as

domain.center is the file that contains the domain center location lat/lon, for example:

18.9

-79

Note: domain center is calculated based on the observed storm center in tcvital



Set up HWRF domain

edit script

`$HOME/HWRF/src/hwrf-utilities/scripts/geogrid.ksh`

Storm ID

Path to WPS

Path to WPS
namelist

```
#####  
# Set input variables:  
#####  
SID=07L  
WPS_ROOT=${HOME}/HWRF/src/WPS  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/hwrf_namelist.wps
```

Path to output

```
#####  
# Set base path of output directory  
#####  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYYMMDDHH}
```



Outline

→ Setup HWRF domain



→ Prepare GFS data

→ WRFV3 initialization

→ Vortex initialization

→ GSI

→ Ocean initialization

→ Coupled Forecast

→ HWRF Postprocessor

→ GFDL Vortex tracker



Prepare GFS data

edit script

Edit \$HOME/HWRF/src/hwrf-utilities/scripts/ungrib.ksh

Storm ID

Path to WPS

Type of data

WPS namelist

```
#####  
# Set input variables:  
#####  
SID=07L  
WPS_ROOT=${HOME}/HWRF/src/WPS  
SOURCE=GFS  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/hwrf_namelist.wps
```

Path to output

Path to data

```
#####  
# Set base path of output directory  
#####  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYYMMDDHH}  
#####  
# Set base path of the GFS gridded input data  
#####  
SOURCE_PATH=/ptmp/HurrTutorial/datasets/GFS_gridded/${YYYYMMDDHH}
```

Prepare GFS data

run ungrib to decode GFS data

In \$HOME/HWRF/src/hwrf-utilities/scripts:

ungrib.ksh

Verify the run was successful:

1) Screen output has this:

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

2)

In /ptmp/\$USER/HWRF/07L/2008082800/ungribprd:
Find the following intermediate files

GFS:2008-08-28_00

GFS:2008-08-28_06

3) if run was unsuccessful, check

/ptmp/\$USER/HWRF/07L/2008082800/ungribprd/ungrib.log



Prepare GFS data

edit script

\$HOME/HWRF/src/hwrf-utilities/scripts/metgrid.ksh

Storm ID

Path to WPS

Type of data

WPS namelist

```
#####  
# Set input variables  
#####  
SID=07L  
WPS_ROOT=${HOME}/HWRF/src/WPS  
SOURCE=GFS  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/hwrf_namelist.wps
```

Path to output dir

```
#####  
# Set base path of output directory  
#####  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYYMMDDHH}
```

Prepare GFS data

run metgrid.ksh to interpolate GFS data to HWRF grid

In \$HOME/HWRF/src/hwrf-utilities/scripts:

metgrid.ksh

Verify the run was successful:

1) Screen output has this:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
! Successful completion of metgrid. !  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

2)

In /ptmp/\$USER/HWRF/07L/2008082800/metgridprd:
Find the following output files

met_nmm.d01.2008-08-28_00:00:00.nc

met_nmm.d01.2008-08-28_06:00:00.nc

3) If run was unsuccessful, check

/ptmp/\$USER/HWRF/07L/2008082800/metgridprd/metgrid.log



Outline

→ Setup HWRF domain



→ Prepare GFS data



→ WRFV3 initialization

→ Vortex initialization

→ GSI

→ Ocean initialization

→ Coupled Forecast

→ HWRF Postprocessor

→ GFDL Vortex tracker



WRFV3 Initialization: real_nmm.exe

edit namelist_main.input

\$HOME/HWRF/src/hwrf-utilities/parm/namelist_main.input

Start time

```
&time_control
start_year           = 2008,    2008,
start_month          = 08,     08,
start_day            = 28,     28,
start_hour           = 00,     00,
start_minute         = 00,     00,
start_second         = 00,     00,
end_year             = 2008,    2008,
end_month            = 08,     08,
end_day              = 28,     28,
end_hour             = 06,     06,
end_minute           = 00,     00,
```

End time



WRFV3 Initialization: real_nmm.exe

edit script

```
$HOME/HWRF/src/hwrf-utilities/scripts/real.ksh
```

```
#####  
# LSF queuing options  
#####  
#BSUB -P 48503008      # Project 99999999 or use default if commented out  
#BSUB -n 1            # number of total (MPI) tasks  
#BSUB -J nmm_real     # job name  
#BSUB -o nmm_real.%J.out # output filename  
#BSUB -e nmm_real.%J.err # output filename  
#BSUB -W 00:30        # wallclock time  
#BSUB -q debug        # queue  
#BSUB -K              # Don't return prompt until the job is finished
```

```
#####  
# Set input variables  
#####  
WRF_ROOT=${HOME}/HWRF/src/WRFV3  
SID=07L  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_main.input
```

```
#####  
# Set base path of output directory  
#####  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}
```

Account number

Number of processor

wall clock time

Queue

Path to WRFV3

Storm ID

Namelist

Path to output



WRFV3 Initialization: real_nmm.exe

run real.ksh to generate IC/BC

```
cd $HOME/HWRF/src/hwrf-utilities/scripts
```

```
bsub < real.ksh
```

Batch system command

Verify the run was successful:

1) In /ptmp/\$USER/HWRF/07L/2008082800/realprd:
in the file **rsl.error.0000** find
“real_nmm: SUCCESS COMPLETE REAL_NMM INIT”

2) In /ptmp/\$USER/HWRF/07L/2008082800/realprd:
Find the following 2 files:
wrfinput_d01 and wrfbdy_d01

3) if run was unsuccessful, check
/ptmp/\$USER/HWRF/07L/2008082800/realprd/rsl.error.0000

Batch system

Parallel jobs are submitted using a batch system.

HWRF has three scripts submitted in a batch system (*real.ksh*, *wrf.ksh*, *run_gsi.ksh*)

Different machines may use different batch systems.

NCAR IBM (bluefire) :

Use the Load Sharing Facility (LSF). The command to submit the job:

bsub < your_job_script

in ***your_job_script*** the command to run executable:

SPMD (single program, multiple data)
mpirun.lsf real_nmm.exe

MPMD (multiple program, multiple data)
mpirun.lsf -cmdfile cmdfile ←

```
hwrp_wm3c.exe
hwrp_ocean_united.exe
wrf.exe
wrf.exe
.....
wrf.exe
wrf.exe
```

You need to understand the batch system used on the machines where you are going to run HWRF.



Outline

→ Setup HWRF domain



→ Prepare GFS data



→ WRFV3 initialization



→ Vortex initialization

→ GSI

→ Ocean initialization

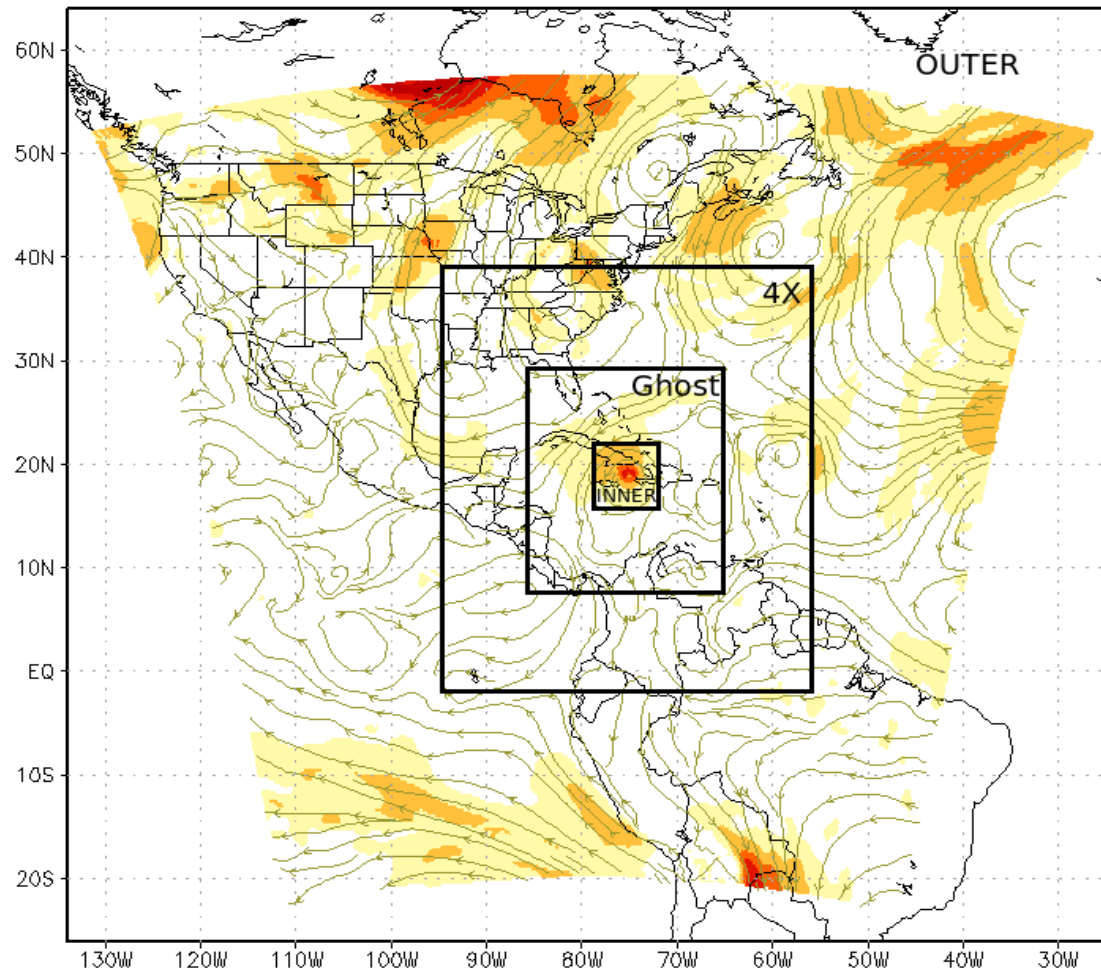
→ Coupled Forecast

→ HWRF Postprocessor

→ GFDL Vortex tracker



Vortex Initialization: the 4 domains



Vortex initialization outline

wrf.exe analysis-run: get the initial fields from GFS data on the inner nest domain

wrf.exe ghost-run: get the initial fields from GFS data on the ghost domain

stage 1: split the previous 6 hour forecast into an environment fields and a hurricane vortex (only for a cycled run)

stage 2: split the GFS initial condition into an environment fields and a hurricane vortex (always run)

stage 3: adjust the hurricane vortex and add the new hurricane vortex back to GFS environment fields -> new initial conditions.

wrf.ksh analysis-run

edit script

`$HOME/HWRF/src/hwrf-utilities/scripts/wrf.ksh`

```
#####  
# LSF queuing options  
#####  
#BSUB -P 48503008 # Project 99999999 or use default if commented out  
#BSUB -a poe # select poe  
#BSUB -n 32 # number of total (MPI) tasks  
#BSUB -R "span[ptile=32]" # run a max of 16 tasks per node  
#BSUB -J hwrf # job name  
#BSUB -o hwrf.%J.out # output filename  
#BSUB -e hwrf.%J.out # error filename  
#BSUB -W 02:30 # wallclock time  
#BSUB -q debug # queue  
#BSUB -K # Don't return prompt until the job is finished
```

```
#####  
# Set input variables  
#####  
SID=07L  
WRF_ROOT=${HOME}/HWRF/src/WRFV3  
WRF_MODE=analysis # Must be one of: analysis, ghost, or main  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_${WRF_MODE}.input  
DATA_DIR=/ptmp/HurrTutorial/datasets/
```

```
## Set base path of output directory  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}
```

Project number

Poe

N of processors

Wallclock time

Queue

Storm ID

Path to WRFV3

WRF MODE

namelist

Path to input data

Path to output dir



wrf.ksh analysis-run

edit namelist

\$HOME/HWRF/src/hwrf-utilities/parm/namelist_analysis.input

Start time



End time



only a **one-minute** run.

```
&time_control
start_year           = 2008,    2008,
start_month          = 08,      08,
start_day            = 28,      28,
start_hour           = 00,      00,
start_minute         = 00,      00,
start_second         = 00,      00,
end_year             = 2008,    2008,
end_month            = 08,      08,
end_day              = 28,      28,
end_hour             = 00,      00,
end_minute           = 01,      01,
interval_seconds     = 21600,
history_interval     = 360,360,
frames_per_outfile   = 1,1
analysis             = F,      F,
```



wrf.ksh analysis-run

run script

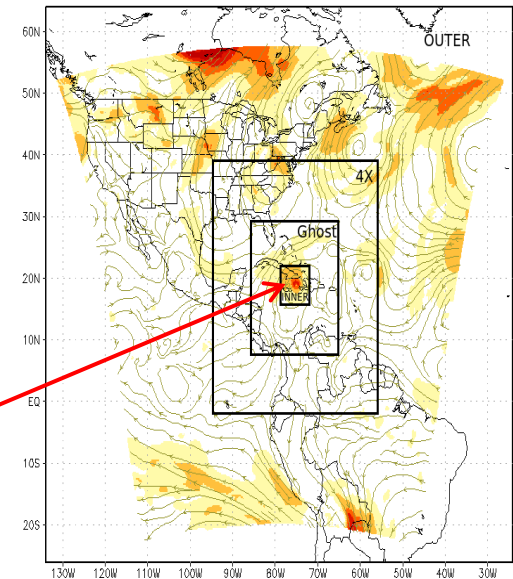
In \$HOME/HWRF/src/hwrf-utilities/scripts

bsub < wrf.ksh

Verify the run is successful:

In /ptmp/\$USER/HWRF/07L/2008082800/wrfanalysisprd

- 1) find in **rsl.error.0000** "SUCCESS COMPLETE WRF"
- 2) Find the output file **wrfanl_d02_2008-08-28_00:00:00**
- 3) If the run is unsuccessful, check **rsl.error.0000**



wrf.ksh ghost-run

edit script

`$HOME/HWRF/src/hwrf-utilities/scripts/wrf.ksh`

```
#####  
# LSF queuing options  
#####  
#BSUB -P 48503008           # Project 99999999 or use default if commented out  
#BSUB -a poe                # select poe  
#BSUB -n 32                 # number of total (MPI) tasks  
#BSUB -R "span[ptile=32]"   # run a max of 16 tasks per node  
#BSUB -J hwrf               # job name  
#BSUB -o hwrf.%J.out        # output filename  
#BSUB -e hwrf.%J.out        # error filename  
#BSUB -W 02:30              # wallclock time  
#BSUB -q debug               # queue  
#BSUB -K                     # Don't return prompt until the job is finished
```

```
#####  
# Set input variables  
#####  
SID=07L  
WRF_ROOT=${HOME}/HWRF/src/WRFV3  
WRF_MODE=ghost # Must be one of: analysis, ghost, or main  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_${WRF_MODE}.input  
DATA_DIR=/ptmp/HurrTutorial/datasets/
```

```
## Set base path of output directory  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}
```

Account number

poe

N of processors

Wallclock time

Queue

Storm ID

Path to WRFV3

WRF MODE

namelist

Path to input data

Path to output dir



wrf.ksh ghost-run edit namelist

\$HOME/HWRF/src/hwrf-utilities/parm/namelist_ghost.input

Start time



End time



only a **one-minute** run.

```
&time_control
start_year      = 2008,    2008,
start_month     = 08,     08,
start_day       = 28,     28,
start_hour      = 00,     00,
start_minute    = 00,     00,
start_second    = 00,     00,
end_year        = 2008,    2008,
end_month       = 08,     08,
end_day         = 28,     28,
end_hour        = 00,     00,
end_minute      = 01,     01,
interval_seconds = 21600,
history_interval = 360,360,
frames_per_outfile = 1,1
analysis        = F,                                F,
```



wrf.ksh ghost-run run script

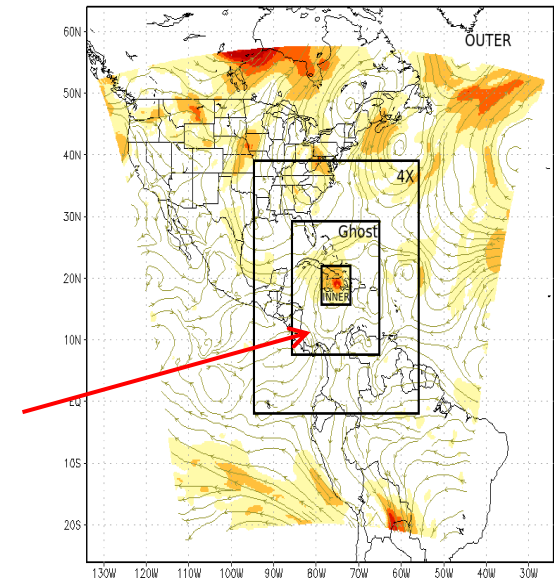
In \$HOME/HWRF/src/hwrf-utilities/scripts

bsub < wrf.ksh

Verify the run is successful:

In /ptmp/\$USER/HWRF/07L/2008082800/wrfghostprd

- 1) find in **rsl.error.0000** "SUCCESS COMPLETE WRF"
- 2) Find the output file **ghost_d02_0000-00-00_00:00:00**
- 3) If the run is unsuccessful, check **rsl.error.0000**



vortex initialization Stage 1

edit script

```
$HOME/HWRF/src/hwrf-utilities/scripts/relocate_stage1.ksh
```

Storm ID

Path to WRFV3

Namelist

Path to output dir

Path to previous 6hr
forecast

```
#####  
# Set input variables  
#####  
SID=07L  
WRF_ROOT=${HOME}/HWRF/src/WRFV3  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_analysis.input  
  
# Set base path of output directory  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}  
  
# Set base path of output directory containing previous 6hr forecast for cycling  
CYCLE_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYYMMDDHH_6hrs_ago}
```

vortex initialization Stage 1

run script

In \$HOME/HWRF/src/hwrf-utilities/scripts/relocate_stage1.ksh

relocate_stage1.ksh

Verify the run is successful:

---For a **cold** start run (in /ptmp/\$USER/HWRF/07L/2008082718/wrfprd no wrfinput_d01* and wrfinput_d2* are found)

The screen output will have “!!! THIS IS A COLD START !!!”

---For a **cycled** run (in /ptmp/\$USER/HWRF/07L/2008082718/wrfprd wrfinput_d01* and wrfinput_d2* are found)

- 1) “failed” is not found in the standard output (stdout)
- 2) In /ptmp/\$USER/HWRF/07L/2008082800/relocateprd find the files:
 - storm_size_p**
 - storm_pert_new**
 - storm_sym**
 - Storm_radius**
 - wrf_env**

vortex initialization Stage 2

edit script

`$HOME/HWRF/src/hwrf-utilities/scripts/relocate_stage2.ksh`

Path to WRFV3

```
# Set vars  
WRF_ROOT=${HOME}/HWRF/src/WRFV3
```

Storm ID

```
SID=07L
```

Namelist

```
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_analysis.input
```

Path to output dir

```
# Set base path of output directory  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}
```

vortex initialization Stage 2

run script

In \$HOME/HWRF/src/hwrf-utilities/scripts/relocate_stage2.ksh

relocate_stage2.ksh

Verify the run is successful:

- 1) “failed” is not found in your standard output (stdout).
- 2) In /ptmp/\$USER/HWRF/07L/2008082800/relocateprd find the following files:

gfs_env
Roughness
Storm_pert_gfs
Storm_radius_gfs

vortex initialization Stage 3

edit script

\$HOME/HWRF/src/hwrf-utilities/scripts/relocate_stage3.ksh

Path to WRFV3

```
# Set vars  
WRF_ROOT=${HOME}/HWRF/src/WRFV3
```

Storm ID

```
SID=07L
```

Namelist

```
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_analysis.input
```

Path to output dir

```
# Set base path of output directory  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}
```

Path to previous 6hr
forecast

```
# Set base path of output directory containing previous 6hr forecast for cycling  
CYCLE_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYYMMDDHH_6hrs_ago}
```

vortex initialization Stage 3

run script

In \$HOME/HWRF/src/hwrf-utilities/scripts

relocate_stage3.ksh

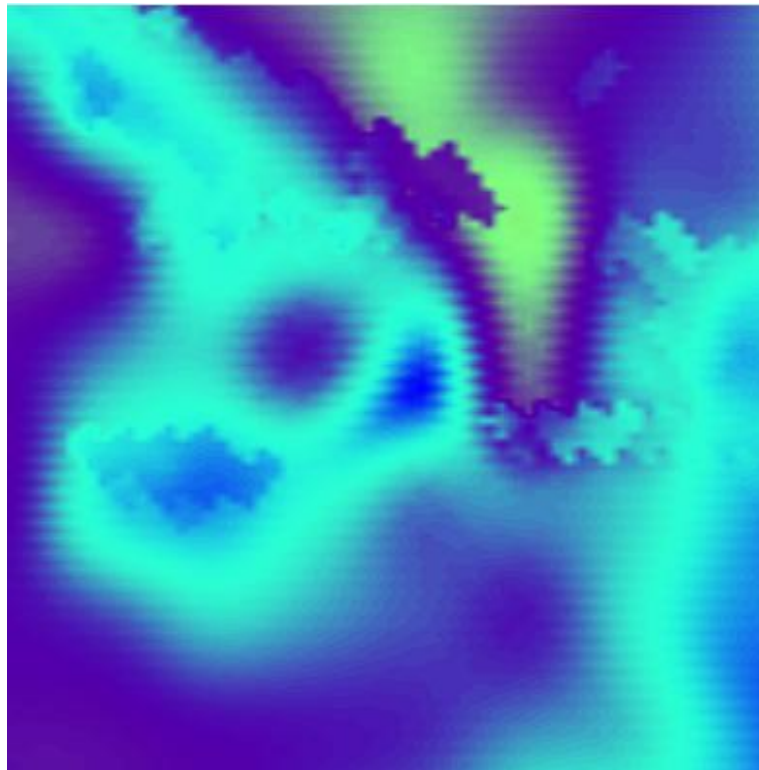
Verify the run is successful:

- 1) “failed” is not found in the standard output (stdout)
- 2) In /ptmp/\$USER/HWRF/07L/2008082800/relocateprd find the following files:

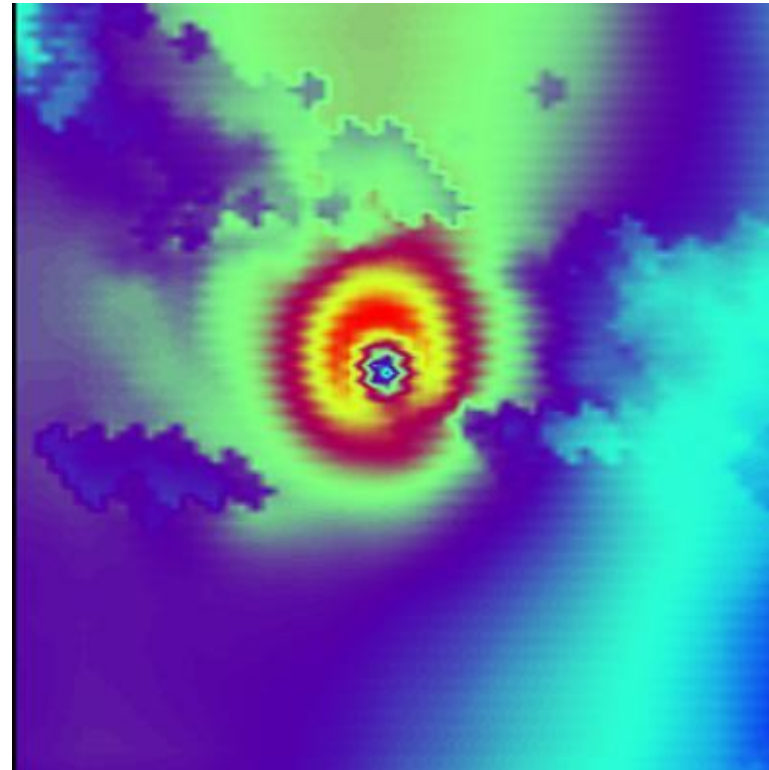
wrfinput_d01
wrfanl_d02*

Vortex Initialization examples

Gustav 2008 08 28 00 Nest domain 10-m winds
(observed. 10-m wind: 40 knots)



Without Vortex Initialization
Max: 22 knots



With Vortex Initialization
Max: 42 knots

Outline

→ Setup HWRF domain



→ Prepare GFS data



→ WRFV3 initialization



→ Vortex initialization



→ GSI

→ Ocean initialization

→ Coupled Forecast

→ HWRF Postprocessor

→ GFDL Vortex tracker



GSI for HWRF

edit script

/HOME/\${USER}/HWRF/src/hwrf-utilities/scripts/run_gsi.ksh

Account number → ## Below (IBM Bluefire LSF queueing system) commands
poe → #BSUB -P 48500053
N of processors → #BSUB -a poe
#BSUB -n 4 # number of total tasks
#BSUB -R "span[ptile=2]" # how many tasks per node (up to 16 tasks per node)
Wallclock time → #BSUB -J gsi # job name
#BSUB -o gsi.%J.out # output filename
#BSUB -e gsi.%J.err # output filename
Queue → #BSUB -W 00:30 # wallclock time
#BSUB -q debug # queue

N of processors → GSIPROC=4
Platform → ARCH=IBM_LSF

Domain → DOMAIN=2
Storm ID → SID=07L

Namelist → NAMELIST=\${HOME}/HWRF/src/hwrf-utilities/parm/namelist_analysis.input



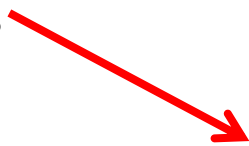
GSI for HWRF

edit script (continued)

/HOME/\${USER}/HWRF/src/hwrf-utilities/scripts/run_gsi.ksh

```
#####  
# Paths for namelist, executables, fix files, and output directories  
#####  
GSI_EXE=${HOME}/HWRF/src/comGSI_v2.5/run/gsi.exe  
HWRF_UTILITIES_ROOT=${HOME}/HWRF/src/hwrf-utilities  
DATA_REMV=${HWRF_UTILITIES_ROOT}/exec/hwrf_data_remv.exe  
SSRC=${HOME}/HWRF/src/comGSI_v2.5/util/ssrc/ssrc.exe  
  
FIX_ROOT=/ptmp/HurrTutorial/datasets/fix/comGSI  
CYCLE_DATA=/ptmp/${USER}/HWRF/${SID}/${ANAL_TIME}  
WORK_ROOT=/ptmp/${USER}/HWRF/${SID}/${ANAL_TIME}  
OBS_ROOT=/ptmp/HurrTutorial/datasets/obs/${ANAL_TIME}  
PREPBUFR=${OBS_ROOT}/gfs.${ANAL_TIME}.prepbuf
```

Executables



Fix data path



Work directory



Path to Obs



Obs file names



GSI for HWRF

run script

In \$HOME/HWRF/src/hwrf-utilities/scripts

bsub < run_gsi.ksh

Run 2 times, one for DOMAIN=1 (parent domain) and the other for DOMAIN=2 (ghost domain)

Verify the run is successful:

1)

```
ENDING DATE-TIME    APR 07,2011  16:27:31.831   97  THU   2455659
PROGRAM GSI_ANL HAS ENDED.  IBM RS/6000 SP
```

2)

```
In /ptmp/${USER}/HWRF/07L/2008082800/gsiprd/wrfinput_d01: wrf_inout
In /ptmp/${USER}/HWRF/07L/2008082800/gsiprd/wrfghost_d02: wrf_inout
```

Merge GSI+Vortex initialization

edit script

```
/HOME/${USER}/HWRF/src/hwrf-utilities/scripts/merge.ksh
```

Storm ID → **SID=07L**
Path to WRF → **WRF_ROOT=\${HOME}/HWRF/src/WRFV3**
Namelist → **NAMELIST=\${HOME}/HWRF/src/hwrf-utilities/parm/namelist_analysis.input**
Path to HWRF-Utilities → **HWRF_UTILITIES_ROOT=\${HOME}/HWRF/src/hwrf-utilities/exec**

Work directory → **DOMAIN_HOME=/ptmp/\${USER}/HWRF/\${SID}/\${YYYY}\${MM}\${DD}\${HH}**

Merge GSI+Vortex initialization

run script

In \$HOME/HWRF/src/hwrf-utilities/scripts

merge.ksh

Verify the run is successful:

in /ptmp/\${USER}/HWRF/07L/2008082800/mergeprd

wrfinput_d01

wrfanl_d02_2008-08-28_00:00:00



Outline

- Setup HWRF domain 
- Prepare GFS data 
- WRFV3 initialization 
- Vortex initialization 
- GSI 
- Ocean initialization
- Coupled Forecast
- HWRF Postprocessor
- GFDL Vortex tracker



Ocean Initialization

edit scripts

```
$HOME/HWRF/src/pomtc/ocean_scripts/kick*.sh
```

Storm ID

```
stormid=07L # e.g. SID = 07L
```

Start date

```
#####  
# Enter starting date for POM-TC as YYYYMMDDHH #  
#   YYYY = 4-digit year #  
#   MM = 2-digit month #  
#   DD = 2-digit day #  
#   HH = 2-digit hour #  
#####
```

```
start_date=2008082800 # e.g. YYYYMMDDHH = 2008082800
```

Path to data

Path to source

Path to output

```
#####  
# Define directories #  
#####
```

```
data_d=/ptmp/HurrTutorial/datasets
```

```
sorc_d=${HOME}/HWRF/src/pomtc
```

```
work_d=/ptmp/${USER}/HWRF/${stormid}/${start_date}/oceanprd
```



Ocean Initialization

run scripts

In \$HOME/HWRF/src/pomtc/ocean_scripts

kickit00_region.sh

Function:

From the storm id, select the appropriate track from the NHC hurricane message files and determine which POM region to use.

Verify the run is successful:

in /ptmp/\$USER/HWRF/07L/2008082800/oceanprd/
find ocean_region_info.txt, pom_region.txt

Ocean Initialization

run scripts

In \$HOME/HWRF/src/pomtc/ocean_scripts

kickit01_sharpn.sh

Function:

GDEM T/S climatology is interpolated to POMTC domain grid and modified using the feature-based model

Verify the run is successful:

in /ptmp/\$USER/HWRF/07L/2008082800/oceanprd/**sharpn**
find **gfdl_initdata.united.08**

If unsuccessfully, check the log file
/ptmp/\$USER/HWRF/07L/2008082800/oceanprd/sharpn/**sharpn.out**



Ocean Initialization

run scripts

In \$HOME/HWRF/src/pomtc/ocean_scripts

kickit02_getsst.sh

Function:

extract the SST, mask, and longitude/latitude information from the GFS spectral files

Verify the run is successful:

in /ptmp/\$USER/HWRF/07L/2008082800/oceanprd/**getsst**
find **lonlat.gfs mask.gfs.dat sst.gfs.dat**

If unsuccessfully, check the log file

/ptmp/\$USER/HWRF/07L/2008082800/oceanprd/sharprn/**getsst.out**



Ocean Initialization

run scripts

In \$HOME/HWRF/src/pomtc/ocean_scripts

kickit03_phase3.sh

Function:

POM-TC is then integrated for 48-hours, keeping SST constant, to generate geostrophically-adjusted currents.

Verify the run is successful:

in /ptmp/\$USER/HWRF/07L/2008082800/oceanprd/**phase3**
find **RST.phase3.united**

If unsuccessfully, check the log file

/ptmp/\$USER/HWRF/07L/2008082800/oceanprd/sharprn/**phase3.out**



Ocean Initialization

run scripts

In \$HOME/HWRF/src/pomtc/ocean_scripts

kickit04_phase4.sh

Function:

POM-TC is then integrated for 72-hours, with wind stress applied at the sea surface based on data extracted from the NHC hurricane message file.

Verify the run is successful:

in /ptmp/\$USER/HWRF/07L/2008082800/oceanprd/**phase4**
find **RST.final**

If unsuccessfully, check the log file

/ptmp/\$USER/HWRF/07L/2008082800/oceanprd/sharprn/**phase4.out**



Outline

→ Setup HWRF domain 

→ Prepare GFS data 

→ WRFV3 initialization 

→ Vortex initialization 

→ Ocean initialization 

→ Coupled Forecast

→ HWRF Postprocessor

→ GFDL Vortex tracker



Coupled forecast

edit namelist

\$HOME/HWRF/src/hwrf-utilities/parm/namelist_main.input

Start time

```
&time_control
start_year           = 2008,    2008,
start_month          = 08,      08,
start_day            = 28,      28,
start_hour           = 00,      00,
start_minute         = 00,      00,
start_second         = 00,      00,
end_year             = 2008,    2008,
end_month            = 08,      08,
end_day              = 28,      28,
end_hour             = 06,      06,
end_minute           = 00,      00,
```

End time



Coupled forecast

edit script

`$HOME/HWRF/src/hwrf-utilities/scripts/wrf.ksh`

```
#####  
# LSF queuing options  
#####  
#BSUB -P 48503008           # Project 99999999 or use default if commented out  
#BSUB -a poe                # select poe  
#BSUB -n 32                 # number of total (MPI) tasks  
#BSUB -R "span[ptile=32]"   # run a max of 16 tasks per node  
#BSUB -J hwrf               # job name  
#BSUB -o hwrf.%J.out        # output filename  
#BSUB -e hwrf.%J.out        # error filename  
#BSUB -W 02:30              # wallclock time  
#BSUB -q debug               # queue  
#BSUB -K                     # Don't return prompt until the job is finished
```

```
# Set vars  
WRF_ROOT=${HOME}/HWRF/src/WRFV3  
SID=07L  
WRF_MODE=main # Must be one of: analysis, ghost, or main  
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_${WRF_MODE}.input
```

```
## Set base path of output directory  
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}
```

Account number
Poe
Number of processor
wallclock time
Queue
Path to WRFV3
Storm ID
Mode
Namelist
Path to output dir



Coupled forecast

run script

In \$HOME/HWRF/src/hwrf-utilities/scripts

bsub < wrf.ksh

Function:


Coupled forecast

verify output the run is successful:

In /ptmp/\$USER/HWRF/07L/2008082800/wrfprd

- 1) find "SUCCESS COMPLETE WRF" in rsl.error.0000
- 2) find wrfout_d01* and wrfout_d02*

Outline

- Setup HWRF domain 
- Prepare GFS data 
- WRFV3 initialization 
- Vortex initialization 
- Ocean initialization 
- Coupled Forecast 
- HWRF Postprocessor (Jamie) 
- GFDL Vortex tracker



GFDL Vortex tracker

edit script

`$HOME/HWRF/src/hwrf-utilities/scripts/tracker.ksh`

Path to WPPV3

Storm ID

Namelist

ATCFNAME

Path to wgrib

Path to output dir

```
#!/bin/ksh
# Set vars
WPP_ROOT=${HOME}/HWRF/src/WPPV3
SID=07L
NAMELIST=${HOME}/HWRF/src/hwrf-utilities/parm/namelist_main.input
ATCFNAME=HTUT
WGRIB=/contrib/bin/wgrib
```

```
# Set base path of output directory
DOMAIN_HOME=/ptmp/${USER}/HWRF/${SID}/${YYYY}${MM}${DD}${HH}
```

GFDL Vortex Tracker

run script

In \$HOME/HWRF/src/hwrf-utilities/scripts:

tracker.ksh

Function:

Run GFDL Vortex Tracker to extract track information from HWRF forecast

Verify the run is successful:

In /ptmp/\$USER/HWRF/07L/2008082800/gvtprd
find **fort.68**

a sample tracker output:

```
AL, 07, 2008082800, 03, HTUT, 000, 189N, 751W, 43, 999, XX, 34, NEQ, 0042, 0028, 0036, 0042
AL, 07, 2008082800, 03, HTUT, 006, 188N, 758W, 59, 980, XX, 34, NEQ, 0068, 0074, 0043, 0070
AL, 07, 2008082800, 03, HTUT, 006, 188N, 758W, 59, 980, XX, 50, NEQ, 0047, 0043, 0027, 0041
AL, 07, 2008082800, 03, HTUT, 012, 187N, 767W, 76, 974, XX, 34, NEQ, 0089, 0095, 0054, 0071
AL, 07, 2008082800, 03, HTUT, 012, 187N, 767W, 76, 974, XX, 50, NEQ, 0044, 0043, 0019, 0039
AL, 07, 2008082800, 03, HTUT, 012, 187N, 767W, 76, 974, XX, 64, NEQ, 0026, 0026, 0000, 0025
AL, 07, 2008082800, 03, HTUT, 018, 186N, 774W, 77, 969, XX, 34, NEQ, 0103, 0103, 0068, 0078
AL, 07, 2008082800, 03, HTUT, 018, 186N, 774W, 77, 969, XX, 50, NEQ, 0044, 0035, 0000, 0034
AL, 07, 2008082800, 03, HTUT, 018, 186N, 774W, 77, 969, XX, 64, NEQ, 0029, 0029, 0000, 0024
AL, 07, 2008082800, 03, HTUT, 024, 187N, 780W, 68, 965, XX, 34, NEQ, 0115, 0104, 0069, 0101
AL, 07, 2008082800, 03, HTUT, 024, 187N, 780W, 68, 965, XX, 50, NEQ, 0053, 0040, 0035, 0042
AL, 07, 2008082800, 03, HTUT, 024, 187N, 780W, 68, 965, XX, 64, NEQ, 0032, 0018, 0000, 0031
```



Plotting the model output




- **WRF native output**
 - ncview

- **WPP output**
 - GrADS
sample:
`\${HOME}/HWRF/src/hwrf-utilities/scripts/run_grads`

- **POM-TC output**
 - GrADS
sample:
`\${HOME}/HWRF/src/pomtc/ocean_plot/pom-tc-united-grads.sh`
`\${HOME}/HWRF/src/pomtc/ocean_plot/pom-tc-eastatl-grads.sh`

- **GFDL Vortex Tracker output**
 - GrADS
gfdl-vortextracker/trk_plot/plottrak/atcfplot.sh`

Outline

- Setup HWRF domain 
- Prepare GFS data 
- WRFV3 initialization 
- Vortex initialization 
- Ocean initialization 
- Coupled Forecast 
- HWRF Postprocessor (Jamie) 
- GFDL Vortex tracker 