WRF-bPlume tutorial

Sudheer R Bhimireddy

UTSA (2015-2020)

email: fhl598@my.utsa.edu

Installing WRF-bPlume

- Get the latest link from the WRF-bPlume website (Link to be added here) or GitHub (https://github.com/sudheer-wrf/wrf-bplume)
- Follow usual WRF-ARW compiling steps
  - (https://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php)
- Compile WRF in Large-eddy simulation mode
- A successful compilation for idealized cases will result in ideal.exe and wrf.exe executables in the main/ directory
- Start building cases using the test case available within WRF build
- For a quick CBL case, use /test/em_les/ present in the main WRF directory
- Key files modified for WRF-bPlume are present in Registry/ and dyn_em/ directories

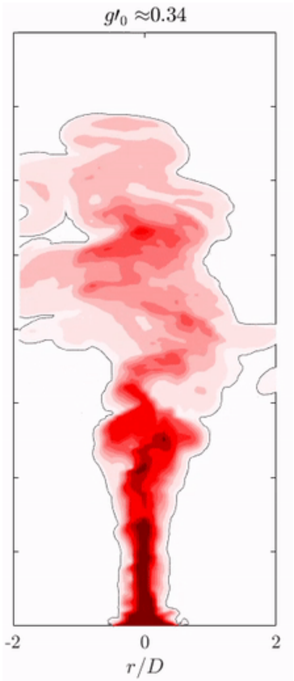
Useful WRF-ARW links

Technical guide	https://opensky.ucar.edu/islandora/object/opensky:2898
WRF Dynamics guide	https://www2.mmm.ucar.edu/wrf/users/docs/wrf-dyn.html
User guide	https://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/v4.0/contents.html
Tutorials page	https://www2.mmm.ucar.edu/wrf/users/supports/tutorial.html
Best Practices	https://www2.mmm.ucar.edu/wrf/users/namelist_best_prac_wrf.html

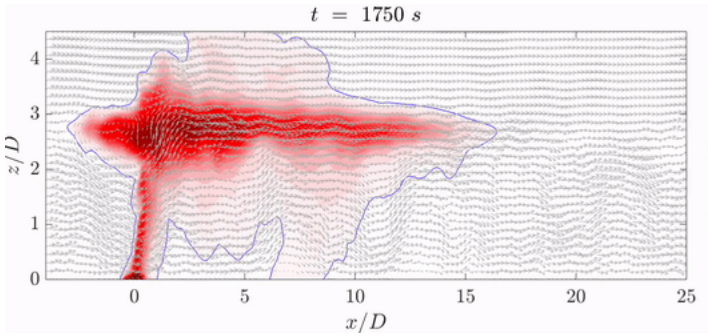
Type of flows to simulate using WRF-bPlume

Axisymmetric plumes

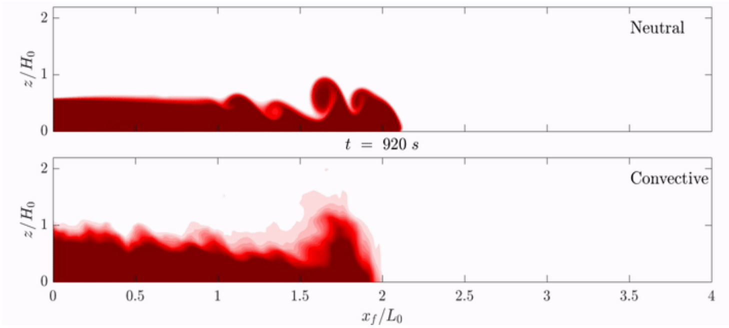
Thermal plumes



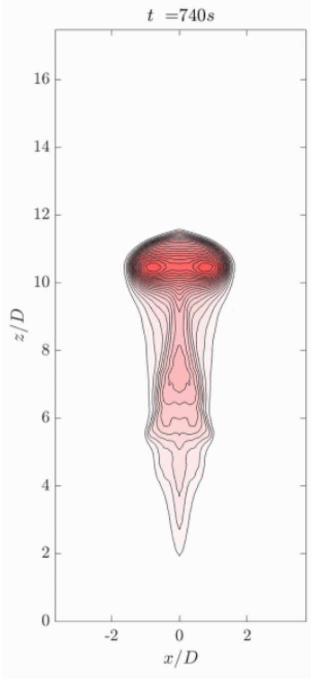
Buoyant gas plumes



Density currents



Buoyant bubbles



New in WRF-bPlume

New variables added in WRF-bPlume are:

- tke_heat_flux_hot – additional heat flux at the source location
- heat_flux_wait_mins – Minutes from simulation start to release the plume
- heat_flux_run_mins – Minutes from plume start to turn off the plume source
- density_current – Flag to simulate lock-exchange density current
- lock_length – density current lock length
- lock_height – density current lock height
- lock_mixture – density current lock fluid mixing ratio
- axisymmetric_plume – flag to simulate axisymmetric plume at the center of the bottom boundary
- plume_gas_constant – Gas constant of plume material
- plume_surface_flux – Surface flux of plume mixing ratio
- source_dia – Plume source diameter

Files modified for WRF-bPlume

For WRF-bPlume following files are modified based on the default WRF-ARW framework (v 4.0.3):

- /Registry/Registry.EM_COMMON
- /dyn_em/module_big_step_utilities.F
- /dyn_em/module_diffusion_em.F
- /dyn_em/module_em.F
- /dyn_em/module_initialize_ideal.F
- /dyn_em/solve_em.F
- /share/wrf_timeseries.F

Setting Simulation times

```
&time_control
 run_days                                = 0,
 run_hours                               = 2,
 run_minutes                             = 0,
 run_seconds                             = 0,
 start_year                              = 0001,
 start_month                             = 01,
 start_day                               = 01,
 start_hour                              = 00,
 start_minute                            = 00,
 start_second                            = 00,
 end_year                                = 0001,
 end_month                               = 01,
 end_day                                 = 01,
 end_hour                                = 04,
 end_minute                              = 00,
 end_second                              = 00,
 history_interval_m                      = 00,
 history_interval_s                      = 10,
 frames_per_outfile                      = 90,
 restart                                 = .false.,
 restart_interval_m                      = 15,
 io_form_history                         = 2
 io_form_restart                         = 2
 io_form_input                           = 2
 io_form_boundary                        = 2
 iofields_filename                       = "myoutfields.txt"
```

WRF stops running when either run_* or end_* condition is fulfilled

WRF writes the output for each domain.
Minimum time step to write is 1s

Restart runs are half-completed runs
The start time must match the time information in restart file (wrfrst_d01_*.nc)

Contents of myoutfields.txt

- Useful in reducing the WRFOUT file size
- Many of the default variables written by WRF are unimportant for ideal cases

```
-:h:0:LU_INDEX,DZS,VAR_SSO,NEST_POS,FNM,FNP,RDNW,RDN,DNW,DN,CFN,CFN1,RDX,RDY,RESM
-:h:0:ZETATOP,CF1,CF2,CF3,ITIMESTEP,XTIME,SHDMAX,SHDMIN,SNOALB,TSLB
-:h:0:SMOIS,SH2O,SEAICE,XICEM,SFROFF
-:h:0:UDROFF,IVGTYP,ISLTYP,VEGFRA,ACSNOM,SNOW,SNOWH,CANWAT,COSZEN,LAI,VAR
-:h:0:MAPFAC_M,MAPFAC_U,MAPFAC_V,MAPFAC_MX,MAPFAC_MY,MAPFAC_UX,MAPFAC_UY
-:h:0:MAPFAC_VX,MF_VX_INV,MAPFAC_VY
-:h:0:F,E,SINALPHA,COSALPHA,TISO,MAX_MSTFX,MAX_MSTFY,RAINC,RAINSH,RAINNC
-:h:0:SNOWNC,GRAUPELNC,HAILNC,SWDOWN
-:h:0:OLR,CLAT,ALBBCK,EMISS,NOAHRES,TMN,SNOWC,SR,C1H,C2H,C1F,C2F,C3H,C4H,C3F,C4F,PCB,PC
-:h:0:LANDMASK,LAKEMASK
-:h:0:HFX_FORCE_TEND,LH_FORCE_TEND,TSK_FORCE_TEND,HFX_FORCE,LH_FORCE,TSK_FORCE
-:h:0:ZS,SSTSK,GLW,SWNORM
-:h:0:ISEEDARR_SPPT,ISEEDARR_SKEBS,ISEEDARR_RAND_PERTURB,ISEEDARRAY_SPP_CONV
-:h:0:ISEEDARRAY_SPP_PBL,ISEEDARRAY_SPP_LSM,THIS_IS_AN_IDEAL_RUN
-:h:0:SAVE_TOPO_FROM_REAL,TLP,SST_INPUT
-:h:0:CLDFRA,ALBEDO,ZNU,ZNW
-:h:0:Q2,T2,TH2,U10,V10,GRDFLX,ACGRDFLX,HGT,P_STRAT,XLAT_U,XLONG_U,XLAT_V,XLONG_V
+:h:0:PBLH,ACHFX,ACLHF,SST
```

Anything that is required in addition to usual WRF output variables can be added to stream0 using a +:h:0: command

Setting Simulation domain

```
&domains
 time_step                            = 0,
 time_step_fract_num                  = 1,
 time_step_fract_den                  = 6,
 max_dom                              = 1,
 s_we                                 = 1,
 e_we                                 = 201,
 s_sn                                 = 1,
 e_sn                                 = 201,
 s_vert                               = 1,
 e_vert                               = 301,
 dx                                   = 20,
 dy                                   = 20,
 ztop                                 = 3000,
 grid_id                              = 1,
 parent_id                            = 0,
 i_parent_start                       = 0,
 j_parent_start                       = 0,
 parent_grid_ratio                    = 1,
 parent_time_step_ratio               = 1,
 feedback                             = 0,
 smooth_option                        = 0
```

Fractional timesteps are written as numerator and denominator (here, dt =1/6 sec)

Indices within WRF code starts at 1 rather than 0

Setting Simulation physics

- For Large-eddy simulations, no need to use PBL or Cummulus schemes.
- mp_physics is the key that invokes which microphysics variables are to be considered
- Buoyant gas in bPlume model is part of such schemes

```
&physics
 mp_physics                          = 0,
 ra_lw_physics                       = 0,
 ra_sw_physics                       = 0,
 radt                                = 0,
 sf_sfclay_physics                   = 1,
 sf_surface_physics                  = 0,
 bl_pbl_physics                      = 0,
 bldt                                = 0,
 cu_physics                          = 0,
 cudt                                = 0,
 isfflx                              = 2,
 ideal_xland                         = 1,
 num_soil_layers                     = 5,
```

mp_physics = 0 tells WRF to resolve only water vapor ratios and plume mixing ratios

Setting Simulation dynamics

- Dynamics module is the place where all the plume related inputs are given

```
&dynamics
 hybrid_opt                              = 0,
 rk_ord                                  = 3,
 diff_opt                                = 2,
 km_opt                                  = 2,
 damp_opt                                = 0,
 zdamp                                   = 200.,
 dampcoef                                = 0.2,
 khdif                                   = 1.,
 kvdif                                   = 1.,
 c_s                                     = 0.18
 c_k                                     = 0.10
 mix_isotropic                           = 1
 smdiv                                   = 0.1,
 emdiv                                   = 0.01,
 epssm                                   = 0.1,
 .
 .
 .
 /
```

Tells which sub-grid scale model is to be used

Setting Simulation dynamics

```
&dynamics
 tke_heat_flux                          = 0.24,
 tke_heat_flux_hot                      = 0.24,
 heat_flux_len                          = 1.,
 heat_flux_wait_mins                    = 35.,
 heat_flux_run_mins                     = 180.,
 density_current                        = 0.,
 axisymmetric_plume                     = 1.,
 lock_length                            = 100.,
 lock_height                            = 50.,
 lock_mixture                           = 0.0,
 plume_gas_constant                     = 488.92,
 plume_surface_flux                     = -10.,(g/kg m/s)
 source_dia                             = 400.,
 time_step_sound                        = 6,
 pert_coriolis                          = .false.,
 use_theta_m                            = 1,
 sfs_opt                                = 1,
 m_opt                                  = 1,
 tke_drag_coefficient                   = 0.005,
 top_lid                                = .false.,
 /
```

Used to control the heat-flux at plume source
Total source heat-flux = sum of both heat_flux*
As long as it is 0, heat_flux_len will have no effect

Tells when to start releasing the plume and how long to keep the source "ON"
Minutes are calculated from the simulation start time (not the restart time)

Plume source details
-ve for buoyant gas and +ve for dense gas

Trigger to turn Coriolis balance on/off

Setting boundary conditions

- For LES runs, periodic in horizontal boundary condition is used

```
&bdy_control
 periodic_x                              = .true.,
 symmetric_xs                            = .false.,
 symmetric_xe                            = .false.,
 open_xs                                 = .false.,
 open_xe                                 = .false.,
 periodic_y                              = .true.,
 symmetric_ys                            = .false.,
 symmetric_ye                            = .false.,
 open_ys                                 = .false.,
 open_ye                                 = .false.,
 /
```

Setting initial conditions

- For LES runs, initial sounding profiles are required as initial conditions
- These are given through input_sounding file in the same directory

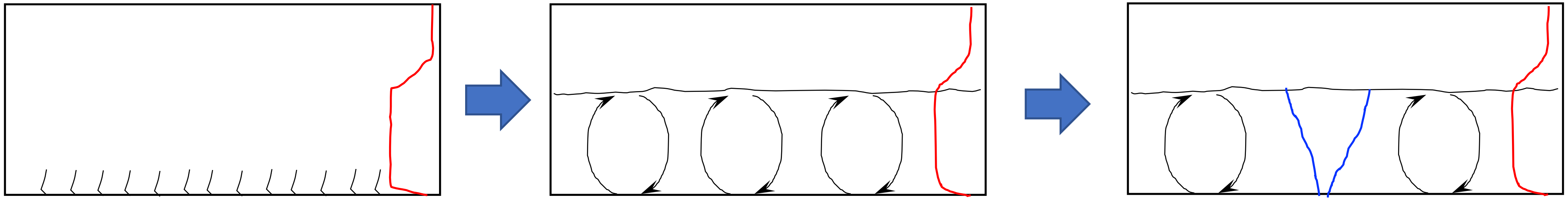
First line gives the temperature and moisture mixing ratio at the surface level (1000 corresponds to pressure)

```
1000.00     305.00      14.00
  25.00     300.00      10.00       10.00        0.00
  75.00     300.00      10.00       10.00        0.00
 125.00     300.00      10.00       10.00        0.00
 175.00     300.00      10.00       10.00        0.00
 225.00     300.00      10.00       10.00        0.00
 275.00     300.00      10.00       10.00        0.00
 325.00     300.00      10.00       10.00        0.00
 375.00     300.00      10.00       10.00        0.00
1725.00     300.00      10.00       10.00        0.00
1975.00     302.43      10.00       10.00        0.00
2025.00     305.63       4.00       10.00        0.00
2075.00     308.05       4.00       10.00        0.00
2125.00     308.20       4.00       10.00        0.00
2175.00     308.35       4.00       10.00        0.00
2225.00     308.50       4.00       10.00        0.00
```

Required inputs are temperature, moisture, U and V wind components
This example has no mean wind

Setting Convective Boundary Cases

- For practice with CBL runs, a good place to start - /Build_WRF/WRF-4.0.3/test/em_les/



- Initialize domain with surface heat-flux and soundings

- Run until boundary layer develops
- Write restart files

- Restart the simulation and start releasing the plume using `heat_flux_wait_mins`

Setting Simulation times – Plume + CBL

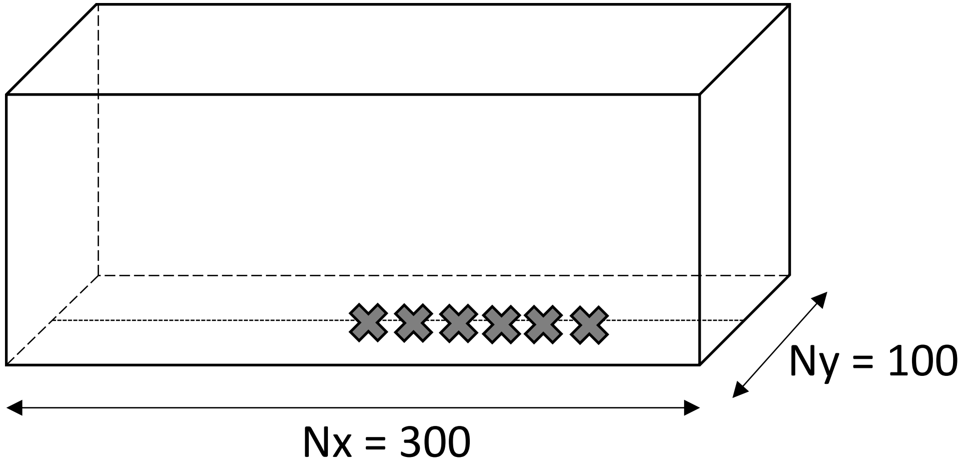
```
&time_control
 run_days                            = 0,
 run_hours                           = 2,
 run_minutes                         = 0,
 run_seconds                         = 0,
 start_year                          = 0001,
 start_month                         = 01,
 start_day                           = 01,
 start_hour                          = 02,
 start_minute                        = 00,
 start_second                        = 00,
 end_year                            = 0001,
 end_month                           = 01,
 end_day                             = 01,
 end_hour                            = 04,
 end_minute                          = 00,
 end_second                          = 00,
 history_interval_m                  = 00,
 history_interval_s                  = 10,
 frames_per_outfile                  = 90,
 restart                             = .true.,
 restart_interval_m                  = 15,
 io_form_history                     = 2
 io_form_restart                     = 2
 io_form_input                       = 2
 io_form_boundary                    = 2
 iofields_filename                   = "myoutfields.txt"
```

This is a restart run.
Lets say, CBL simulation took 2 hr to build and we have WRF restart files written every 15min.
So now we want to start the plume, so start_hour is set to 2 and make sure respective wrfrst_d01_* file is available in the same folder

Writing high-frequency data

- High-frequency (HF) data = data written at every timestep
- By default, WRF does not write any HF data
- Only way to get HF data is through tslist input
- For LES cases, grid indices are required

```
#-----------------------------------------------#
# 24 characters for name | pfx |   I   |    J   |
#-----------------------------------------------#
center001                 c0001   150       50
left0001                  i0001   151       50
left0002                  i0002   152       50
left0003                  i0003   153       50
left0004                  i0004   154       50
left0005                  i0005   155       50
```



Reading WRFOUT data

- WRF writes its output in NetCDF format
- Easiest and quickest way to read WRFOUT files is to use ‘ncl’ scripts
- Alternative way is to use netcdf reader present in MATLAB

* I use ncl script to write Eulerian data as a 1d array in csv files, then use MATLAB to read csv and build mat files

Reading High-frequency data

- WRF writes high-frequency data as ASCII files or log files
- Any ASCII reader should work to read the high-frequency data

Reading WRFOUT data

- MATLAB script to read Eulerian data written by WRF

```
% Specify domain size and output frequency used when writing the WRFOUT files
nt = 90;
nz = 700;
nx = 45;
ny = 45;

% Read T
data1 = csvread('LES_3D_T_NBL_7km_HFX_150_case3_1.csv',1,0);

data1 = reshape(data1,[nx ny nz nt]);

data = data1;

% If reading more than 1 csv written from WRFOUT files, concatenate time dimension
%data = cat(4,data1,data2,data3,data4,data5,data6);

clear data1 data2 data3 data4 data5 data6

save(strcat('NBL_3D_T_F',HFX,'.mat'),'data','-v7.3');
```

Reading High-frequency data

- MATLAB script to read high-frequency files written by WRF

```
% Specify directory where all the high-frequency files are stored
Files_directory = './Instantaneous/';

fileslist = dir(Files_directory);
nFiles    = size(fileslist,1);

for iFile = 3:nFiles % Check this, usually on Stampede system, actual files start from 3rd index
    filename  = fileslist(iFile).name;
    fileData  = strcat(Files_directory,filename);
    temp0     = importdata(fileData);
    totalRows = size(temp0,1); k = 1;
    for kk = 2:totalRows
        temp1 = temp0{kk,1};
        temp2 = strsplit(temp1,' ');
        j     = 1;
        for i = 2:size(temp2,2)
            data(k,j) = str2double(temp2{1,i});
            j         = j + 1;
        end
        k = k + 1;
    end
    saveFile = strcat('./Instantaneous_MAT_Files/',filename,'.mat');
    save(saveFile,'data');
    clear data
end
```