Guide for running Hurricane HIGRAD simulations

(1) Create a grid. Usually use a stretched grid so one can have fine resolution in the interior, but still cover a large domain. Edit “steve_mesh.f90” and “stretch.f” to generate a grid. For now, these are located on coyote at: /scratch3/stepheng/test/src . Select number of grid points in “steve_mesh.f90” and toy with the tuning parameters in “stretch.f” to get the desired grid. By running these two programs a file called “metric.dat” will be made that stores the 3-D grid (we use stretched in all three dimensions). This file must be present above the “src” directory when running the model.

(2) Process data for boundary and initial conditions. For boundary conditions, use ECMWF data and select a representative profile of potential temperature, dry air density, water vapor and components of the wind in the environment surrounding the TC. The data should be gridded to the model vertical spacing. For now, this file is called “thermo_ecmwf.dat” and it should be located above the “src” directory. A binary SST file gridded to (1) can be placed above the “src” directory as well as topography data. Radar data is kept on its native, constant resolution, but is gridded in height according to (1) and output in a specified format (use “higrad_grid.f”, now located on a COAPS machine at /mercury/guimond/AMS08/guillermo/0802/lanl).

(3) Edit the file “gridlist” located above the /src directory. NOTE: IN ALL PARAMETERS, 1 IS USED TO TURN ON THE FUNCTION AND 0 IS USED TO TURN OFF THE FUNCTION. For TC simulations, ihurr and isteve should both be 1. To enable the lightning module, icharge should be 1. The parameter irst indicates whether the current model run is a continuation from a previous run (irst=1) or a new, fresh run (irst=0). Parameter nrst indicates the number of restart files and is left as 1 for my purposes. Parameters n, m and l are the number of grid points in x, y and z; edit them accordingly. Parameter nt is the number of time steps to run the model for and dt is the time step in seconds (multiply these two to get total simulation time in seconds). Usually dt has a value of 0.5 seconds although sometimes 1.0 seconds is used. Parameters nprocx and nprocy are the number of processors in each horizontal dimension that will be used to run the MPI routines. Both nprocx and nprocy must be divisable by the number of grid points in that dimension. Multiplying nprocx and nprocy gives the total number of processors used in the simulation. Parameter ibcx and ibcy indicate that cyclic boundary conditions should be used. Parameter irlx and irly indicate that relaxation boundary condtions (flow near the boundaries relaxes to the background fields) should be used. Parameter frqplot determines the frequency of data output from the model in number of time steps. Parameter restartinname should be edited to the name of the higrad output file that will be used to restart the simulation if a restart is desired. Parameter topofile should be edited to the name of the topography file used in the simulation. These are all the parameters that need to be changed to run a successful TC simulation.

(4) Edit the file “name.inc” located in the /src directory. All that needs to be changed is the file location of “gridlist”.

(5) Subroutines that may need to be edited for a particular run:
   a. cloudbulk.f
      i. microphysics parameterization
   b. compute.f
      i. one part of the output routine
   c. definearray.f
      i. define new arrays and variables
   d. ecmwf.f
      i. read in background/boundary conditions
   e. frqwriteio.f
      i. one part of the output routine
   f. functionhurr.f
      i. solves Navier-Stokes as a residual. Most work done here.
   g. irstreadio.f
      i. a read routine for restart simulations
   h. main.f
      i. start program of the model, declare arrays/variables.
   i. read.f or readhurr.f
      i. read output of model data
   j. readio.f or readiog.f
      i. one part of the read routine for radar data
   k. rinithurr.f
      i. initialize and read routines for ECMWF and radar data
   l. tinithurr.f
      i. boundary conditions, part of the initialize routine
   m. writeio.f
      i. one part of the output routine
   n. rnkshurr.f
      i. computation of each variable at next time step, time numeric issues
   o. stresstkehurr.f90
      i. friction routine

(6) Compile the code. First, must load FORTRAN and MPI libraries. To do this type “source ~stepheng/module” in /src. I use coyote so type “make –f makefilecoyote” in the /src directory to compile the code. This will create an executable in the /higrad directory where the model should be run from.

(7) Request processors from coyote by typing “llogin –n #” where # is the number of processors. Once you are granted processors, type “mpirun –np # ./testexe” to run the model. NOTE: MUST COMPILE THE CODE ON BACK END MACHINES (i.e. cy-c1) AND RUN CODE ON FRONT END MACHINES (i.e. cy-1). NOTE: type “bhosts” to list information on processors. Type “bjobs” to list information on your current job, if any. Also, simulations can be run in batch

mode where a script handles requesting processors and running the code for a set amount of time. Look in /higrad for “steve_script” and edit according to the parameters in the file. To run the script type “bsub < steve_script”. Type “bpeek” to peek at a job run in batch mode. Default run time for a bsub submission is 4 hours, this can be changed to something larger, but the chances of getting what you want go down.

(8) The model output files are listed as “higrad.out.sec+1”. Where ‘sec+1’ is the amount of time (plus one) into the model run. NOTE: THE TIMES ON THE OUTPUT FILES ARE AHEAD BY 1 SEC. THUS, TO GET EXACT TIME OF DATA OUTPUT, MUST READ “time” IN THE FILE OR ELSE SUBTRACT 1 FROM THE FILE NAME. Example: the model outputs “higrad.out.101”. This is the file for 100 seconds into the simulation, not 101 seconds.

(9) To read the model output I have created some code called “reading.f” that reads the file and outputs to what you want.