h-NUMO Documentation Guide

Yao Gahounzo
Computing Ph.D.
Boise State University
Boise, ID

April 7, 2024

Contents

List of Figures

Foreword

1 Acknowledgements

Chapters 1 through 6 of this document were based on the NUMA documentation written by Frank Giraldo.

2 h-NUMO Directory Structure

A typical copy of NUMA consists of the following:

- Makefile
- config.user
- config.p4est
- INSTALL
- h-NUMO_Documentation_Guide.pdf
- graphics
- make_depend.pl
- output
- run_scripts
- src
- p4est

where the blue font denotes directories, while the black font represents files.

2.1 Makefiles and Dependencies

To make an executable version of h-NUMO, you need to run the Makefile. To do so, type "make platform_name" where platform_name must have been added to the file config.user. Inside config.user you will find options for different computers. So far we have for Borah and Falcon. Use these as a guideline to generate the proper calls for your platform. For example, you will see calls to the METIS graph-partitioning software and the LAPACK libraries.

The file Makefile uses the target platform given in config.user and also uses the perl script contained in make_depend.pl. Once you successfully make h-NUMO, additional directories will result. These are:

- bin
- depend.mk
- include
- libs

We now describe the contents of each of the directories.

2.2 INSTALL

This file contains instructions on how to install h-NUMO.

2.3 h-NUMO_Documentation_Guide

This file contains the h-NUMO documentation and a user guide.

2.4 BIN

Upon successfully compiling h-NUMO, the bin directory will be created along with the executable (numo3d).

2.5 GRAPHICS

This directory contains a collection of Matlab m-files that can be used to plot two-dimensional figures (e.g., contours, slices, etc.). Many of these m-files are denoted with a specific case number that allows you to use the m-files without worrying too much about the test case being run.

2.6 INCLUDE

This directory contains the *.mod files that are created upon compilation.

2.7 INPUT_FILES

This directory contains a collection of the input files that may be used to run h-NUMO with specific case numbers. All of the files in this directory should work exactly as they are. Simply copy one of these files to the directory you wish to run from and rename the file to numo3d.in.

2.8 Metis-Serial Graph Partitioning and Fill-reducing Matrix Ordering

This directory contains the METIS libraries that are used in h-NUMO. Note that Metis version 5.1.0 has been compiled. You may have to generate your own METIS libraries.

2.9 LIBS

This directory contains LAPACK libraries compiled from VERSION 3.11.0: November 2022.

2.10 OUTPUT

This directory contains a list of outputs for various test cases. It is sometimes helpful to see what the outputs of each test case should be.

2.11 RUN_SCRIPTS

This directory contains some simple run scripts for use with h-NUMO. It should give the user an idea of how to generate BASH scripts to run with h-NUMO.

2.12 SRC

This is the heart of the code and contains all of the source files actively used in h-NUMO. To add a new file to h-NUMO, you have to include it in the file Makefile.

2.13 p4est

The p4est library was used for the data structures and algorithms for parallel mesh generation, partitioning, and load balancing in h-NUMO.

3 Running h-NUMO

3.1 Setting up the Workspace

Once you have successfully compiled h-NUMO, we recommend the following steps to running h-NUMO.

1. In the home/root directory create a directory called tests. Inside of tests create another directory for a specific simulation (for example, let's call it bump.
2. Inside of bump copy numa3d.in.bump from input_files and rename it numa3d.in.
3. Inside of bump copy one of the run_numo3d from run_scripts. Make sure that the other batch submission commands make sense.

3.2 h-NUMO Output

1. Inside of bump run your runscript.

2. At the end of the run, you will have a collection of output files. In the input file, you can dump the simulation data as a "vtk" file for Paraview visualizations or simply as a "txt" file for Matlab visualizations. For the Matlab visualization, turn matlab_viz to true; otherwise, it will output "vtk" files.

4 h-NUMO Input Files

The h-NUMO input is divided into several namelists: gridnl, input, and other namelists.

The input file has the structure represented below, where the green strings mark the beginning (&) and end (/) of each namelist. The keywords in black identify the name of the variable whose value is assigned after the "=" sign; the values (blue strings) are the only elements that can be changed by the user.The strings and descriptions in magenta indicate the possible input options that can be used for each entry. The strings that begin with an exclamation mark (in red) are self-explicative comments to describe the entries that follow. Detailed explanation of each entry is reported in subections **??** and **??**.

```
&gridnl
 !!Number of elements in x,y, and z and order of the DG approx.
   nelx =  10,
   nely =  10,
   nelz =  1,
   nopx  =  4,
   nopy  =  4,
   nopz  =  0,

 !!Domain information
   xdims   =  0,2e3
   ydims   =  0,2e3
   ztop    =  0.0

 !!P4est Input
   lp4est =  .true.,
   lp6est =  .false.,
   lio_grid_ascii =  .false.,
  lread_external_grid =  .false., !chose .true. if using external mesh

 !!Boundary types:
   x_boundary =    4, 4  !Front and Back (x=-1 and x=+1)
   y_boundary =    4, 4  !Left and Right (y=-1 and y=+1)
   z_boundary =    0, 0  !Bottom and Top (z=-1 and z=+1)
    4 indicates NO-FLUX b.c. (free-slip)
    2 indicates no-slip b.c.
    0 indicates that nothing is enforced at the boundary.

 !!Domain geometry: cube (LAM) or sphere (Global)
  geometry_type = 'cube',
   !cube=3D Box;

 !!Domain decomposition:
  decomp_type = 'metis2d',
   !metis3d=3D metis partitioning;
   !metis2d=2D metis partitioning;
```

```
   !geom2=geometric along XY for Cube/Cartesian grid
/

&input
  dt= 100,
  dt_btp= 1.4,
  time_initial = 0,
  time_final   = 10800,
  time_restart = 1,
  time_scale = 1, !1=seconds, 3600=hours, 86400=days
  lrestart_file = .false.,

 !!mlswe paramter
  ad_mlswe= 0.0,
  botfr= 1,
  cd_mlswe = 1e-7,
  method_visc   = 2,
  visc_mlswe = 50.0,
  matlab_viz = .true., !choose .false. to output vtk files
  adjust_H_vertical_sum = 2, !choose 1 or 2
  mlswe_bc_strong = .true.,
  dg_integ_exact = .true.,
  dump_data = .true.,

  icase = 2023,
   !case2023=bump in the middle layer; ,
   !case2024 = lake-at-rest; ,
   !case1100=double-gyre; ,

 !!Time integration and Filters
  ti_method = 'btp_bcl',
  ti_method_btp = 'rk35', !default = 'explicit' (two-step method)
  si_dimension = '3d',
   !1d IMEX in the vertical/radial direction;
   !3d IMEX
  filter_mux = 0.05,
  filter_muy = 0.05,
  filter_muz = 0.05,
  ifilter = 1,
  filter_weight_type = 'erf',
  filter_basis_type = 'legendre',

 !!Output:
  fname_root = 'new_case201_visc200',
  lrestart_file=.false.,
  out_type = 'bvtk',
   !gks2d=ascii 2d slice;
   !gks3d=ascii 3d output;
   !nc=netcdf 3d output;
   !bvtk=Binary Visual ToolKit (open with Paraview or VisIt)
   !none=No Output

  lprint_diagnostics = T,
   !T=true=print diagnostics
```

```
 !F=false=do NOT print diagnostics
/
```

4.1 GRIDNL

This namelist contains the following variables:

- nelx = number of elements in the x-direction.
- nely = number of elements in the y-direction.
- nelz = number of elements in the z-direction.
- nop = is the polynomial order inside each element.
- xdims = for some tests, xdims gives the minimum and maximum values (in meters) in that direction.
- ydims = for some tests, ydims gives the minimum and maximum values (in meters) in that direction.
- ztop = for some tests, ztop gives the maximum vertical height. The minimum is assumed to be zero which means the ground.
- x_boundary = This gives the boundary condition at the front and back of the x-direction (more on boundary conditions, below).
- y_boundary = This gives the boundary condition at the front and back of the y-direction (more on boundary conditions, below).
- z_boundary = This gives the boundary condition at the front and back of the z-direction (more on boundary conditions, below).
- geometry_type = is the type of geometry being used. 'cube' means flow in a Cartesian box, 'sphere_hex' means the domain is spherical where hexahedral (cubed-sphere) grids are used. For now only 'cube' is used.
- decomp_type = calls a specific type of domain decomposition.

4.2 INPUT

This namelist contains the following variables:

- dt = the time-step in seconds.
- dt_btp = the time-step in seconds.
- time_initial = initial time (can be nonzero if it is a restart).
- time_final = final time (can be in seconds, minutes, hours, or days, depending on the test case. More on this in 5).
- time_restart = time at which restart files are produced. The units of time depend on the test case as mentioned previously. The restart only works for 'txt' outputs at the moment.
- cd_mlswe = is the bottom friction value.
- method_visc = is for the use of viscosity term. If it is 0, that means no use; otherwise, the viscosity solver is used.
- visc_mlswe = is the value of the viscosity in the Laplacian term.

- max_shear_dz = is used to discretize the vertical diffusion or friction between layer terms.
- matlab_viz = a logical flag that turns off and on to output "txt" file for Matlab visualizations.
- adjust_H_vertical_sum = is for applying consistency between barotropic and baroclinic pressure terms.
- dg_integ_exact = a logical flag that turns off and on whether you want to use $2N-1$ quadrature points or $2N+1$ for DG integrations. mlswe_bc_strong = a logical flag that turns off and on the strong boundary conditions for the free-slip boundary condition. We recommend you use "True".
- dump_data = a logical flag that turns off and on the dumping of the simulation data to a file
- case = is the specific test case. These are described in 5.
- ti_method = is the time-integration method (only the predictor-corrector is available now).
- ti_method_btp = is the time-integration method for the barotropic equations
- kstages = number of Runge-Kutta stages if ti_method='rk'. If ti_method='rk35' then the value of kstages does not matter.
- xmu = Boyd-Vandeven filter strength.
- ifilter = how often the filter will be called. ifilter=0 means it is never called and ifilter = 1 means it is called every time-step.
- fname_root = the root name of the output data (only for the vtk output). All data files for this run will have the prefix defined by fname_root.
- out_type = defines the type of output file.
- lprint_diagnostics = a logical flag that turns off and on the printing. For scalability studies we turn off the printing to the screen.

5 Standard Test Cases

In the input file numo3d.in the test cases are controlled by the input parameter 'icase'. h-NUMO contains the following ready-to-use test cases:

1. icase = 2023 is the perturbation in the bottom layer for the two-layer.
2. icase = 1100 is the double-gyre test case.
3. icase = 2024 lake-at-rest test case.

5.1 icase = 2023: Perturbation of the lake-at-rest

This test case consists of a two-layer system. To test whether our model captures the wave propagation speeds correctly, we consider a small perturbation in the interface between layers of the lake-at-rest case with a flat bottom topography. The layer initial interface positions are $z_0 = 0$ m, $z_1 = -20 + 0.5\left(1 + \cos\left(\frac{\pi r}{250}\right)\right)$ m and $z_2 = -40$ m. This test can be run with no-slip or free-slip boundary conditions, and more on these boundary conditions are in Sec. 6.

5.2 icase = 2024: Lake-at-rest

This test case is used to validate the well-balanced property of the DG scheme for the multilayer equations. We have initialized the layers fluid domain with densities $\rho_k = 1027.01037 + 0.2110 \times (k-1)\frac{kg}{m^3},\ k = 1, \ldots, N_l$ and the layer interface positions $z_k = -40/N_l$ m, where N_l is the number of layers. The horizontal extend of the domain was $(x, y) \in [0,\ 2000] \times [0,\ 2000]$ m with wall boundary conditions on both ends, and the bottom topography is given by

$$Z_b(x, y) = 3\left(1 + \cos\left(\frac{\pi r}{250}\right)\right), \tag{1}$$

where $r = \sqrt{(x - 1000)^2 + (y - 1000)^2}$.

5.3 icase = 1100: Double-gyre test

This test case consists of an idealized double-gyre benchmark test (**?**) to validate h-NUMO ability to simulate the mesoscale and submesoscale processes and compare the results with HYCOM. The domain is a closed rectangular ocean basin with a flat bottom. The forcing is spatially varying wind stress with intense western boundary currents, which, together with Coriolis force, results in a counter-clockwise circulation in the northern part and a clockwise circulation in the southern part of the domain.

The horizontal extend is $D = 2000\ km$ in both the zonal and the meridional direction. The depth of the basin is 10 km consisting of two layers, with the upper and lower layers initially having 1.5 km and 8.5 km depths respectively. The densities in the layers are $\rho_1 = 1027.01037\ \ kg/m^3$, $\rho_2 = 1027.22136\ \ kg/m^3$. The Coriolis force is prescribed using a beta-plane approximation centered at 45° N, with a parameter $f = f_0 + \beta(y - D/2)$, where $f_0 = 9.3 \times 10^{-4}\ s^{-1}$ and $\beta = 2^{-11}\ m/s$. We consider two different values of the horizontal viscosity $\nu = 50\ m^2/s$ and $\nu = 500\ m^2/s$, the dimensionless bottom drag coefficient in the linear bottom stress is $c_d = 10^{-7}\ s^{-1}$, and we assume no shear stress between layers. The system is forced by a purely zonal wind stress $\tau = (\tau_x,\ \tau_y)$, where $\tau_x = -\tau_0 \cos(2\pi y/D)$, $\tau_y = 0\ N/m^2$, and $\tau_0 = 0.1\ N/m^2$. We considered two different boundary conditions: free-slip and no-slip. Each model year consists of 360 days, divided into 12 months, with 30 days per month.

6 Boundary Conditions

NUMA reads in boundary conditions via the file numa3d.in through the variables x_boundary, y_boundary, and z_boundary. NUMA allows for the following types of boundary conditions:

- 0 = do nothing boundary conditions

- 2 = no-slip boundary condition
- 4 = free-slip boundary condition

6.1 Do Nothing Boundary Condition

Setting y_boundary=(0,0) tells h-NUMO to ignore these boundaries.

6.2 No-slip boundary conditio

This boundary condition cancels all direction velocity at a boundary (u = 0).

6.3 No-slip or No-flux Boundary Condition

This boundary condition cancels the normal velocity at a boundary (n·$\mathbf{u}$ = 0).

7 Time-Integrators

h-NUMO is equipped with a suite of time-integrators which include:

- Explicit 5-stage 3rd Order Runge-Kutta Methods (used only for the barotropic equations),
- Two-level or predictor-corrector method for both barotropic and baroclinic equations.

8 Graphics

In the directory graphics there are a collection of m-files that can be used to view the output of h-NUMO when written in txt format. The main files contained in this directory are:

-
-
-

References