HORSES3D

A **H**igh-**Or**der (DG) **S**pectral **E**lement **S**olver

Basic User Manual

Numath group

April 21, 2023

Contents

Chapter 1

Input and Output Files

This section lists the files that are read by HORSES3D and the files that are created after the simulation has run successfully.

1.1 Input Files

- Control file (*.control)
  This file controls the parameters of the simulation, is explained in grater detail in Section 1.3.
- Mesh file (*.mesh / *.h5)
  This file has the information about the computational mesh use in the simulation, which at the includes by definition the physical domain and geometry parameters (including the name of the boundaries).
- Polynomial order file (*.omesh) *Optional*
  This file is used when more control of the polynomial use of the DG method, should not be used in simple cases and will not be covered in this manual.
- Problem File (ProblemFile.f90) *Optional*
  Advanced users can have additional control over a simulation without having to modify the source code and recompile the code. To do that, the user can provide a set of routines that are called in different stages of the simulation via this file. It includes routines for user defined initial conditions and boundary conditions.

1.2 Output Files

- Solution file (*.hsol)
  It contains the information about the solution of the simulation in a specific HORSES3D format. It can be converted to other formats for visualization (See Section 4.1.)
- Horses mesh file (*.hmesh)
  This file contains information of the mesh converted by HORSES3D.
- Boundary information (*.bmesh)
  This file contains information of the mesh converted by HORSES3D.
- Partition file (*.pmesh) *Optional*
- Polynomial order file (*.omesh) *Optional*
  This file is used when more control of the polynomial use of the DG method, should not be used in simple cases and will not be covered in this manual.
- Monitor files (*.volume / *.surface / *.residuals)
  This files contain information for default (i.e. residuals) or user defined monitors of variables in specific points (i.e. probes) or averaged by some spec (i.e. surface monitor). The user defined monitors are specified in the control file and are explained in Chapter 2.

1.3 Control File (*.control) - Overview

The control file is the main file for running a simulation. It is a plane text file which uses a simple key and value format (using the equal '=' sign); blocks (using the hash '#' sign); and comments (using Fortran style, admiration '!' sign).

The definition section of a simple control file is shown along with the explanation of the principal parameters:

Flow equations = "NS" ⟶ The equation(s) to be solved, NS is short for Navier–Stokes.
mesh file name = "MESH/myMesh.mesh" ⟶ The mesh file to be used.
solution file name = "RESULTS/mySol.hsol" ⟶ The solution file to be created.
simulation type = time-accurate ⟶ Specifies if there must be performed a 'steady-state' or a 'time-accurate' simulation.
time integration = explicit ⟶ Specifies the type of the numerical scheme for the time integration, could be either 'explicit' or 'implicit'.
Polynomial order = 2 ⟶ Polynomial order to be assigned uniformly to all the elements of the mesh.
restart = .false. ⟶ Logical value. If .TRUE., initial conditions of simulation will be read from restart file specified using the keyword restart file name..
cfl = 0.3 ⟶ A constant related with the convective Courant- Friedrichs-Lewy (CFL) condition that the program will use to compute the time step size.
dcfl = 0.3 ⟶ A constant related with the diffusive Courant-Friedrichs- Lewy (DCFL) condition that the program will use to compute the time step size.
final time = 5.0 ⟶ Specifies the final time of the simulation. This keyword is mandatory for time-accurate solvers.
Number of time steps = 10000 ⟶ Maximum number of time steps that the program will compute.
Output Interval = 50 ⟶ In steady-state, this keyword indicates the interval of time steps to display the residuals on screen. In time-accurate simulations, this keyword indicates how often a 3D output file must be stored.
Convergence tolerance = 1.d-10 ⟶ Residual convergence tolerance for steady-state cases.
mach number = 0.3 ⟶ Physical variable that control the simulation.
Reynolds number = 200.0 ⟶ Physical variable that control the simulation.
Prandtl number = 0.72 ⟶ Physical variable that control the simulation.
AOA theta = 0.0 ⟶ Physical variable that control the simulation.
AOA phi = 90.0 ⟶ Physical variable that control the simulation.
LES model = Smagorinsky ⟶ Specifies the LES model to be used.
save gradients with solution = .true. ⟶ Logical values that specifies some output variables to be saved.
riemann solver = roe ⟶ Specifies the riemann solver to be used.

1.3.1 Blocks

The other kind of specification for the problem file consists of blocks of definitions, which have a name placed on the opening header. The general form is shown below.

```
#define myBlock
        keyword 1 = value 1
        keyword 2 = value 2
        .
        .
        keyword N = value N
# end
```

1.3.2 Boundary Conditions

The boundary conditions are specified as blocks in the control file. Each boundary condition can be individually defined or if multiple boundaries are set with the same definition, it could be done on the same block (with the name separated by a double under score '_' sign). The name of each boundary must match with the one specified at the mesh file.

The block in general can be seen below. Table 1.1 show the values for the type keyword, and the possible value for the parameters depends on the boundary condition.

```
#define boundary myBoundary1__myBoundary2__myBoundary3
        type        = typeValue
        parameter 1 = value 1
        parameter 2 = value 2
# end
```

Table 1.1: Keywords for Boundary Conditions.

Keyword	**Description**	**Default value**
type	*CHARACTER*: Type of boundary condition to be applied. Options are: Inflow, Outflow, NoSlipWall, FreeSlipWall, Periodic, User-defined.	N/A

Chapter 2

Monitors

The monitors are specified individually as blocks in the control file. The only general keyword that can be specified is explained in Table 2.1.

Table 2.1: Keywords for monitors.

Keyword	Description	Default value
monitors flush interval	*INTEGER*: Iteration interval to flush the monitor information to the monitor files.	100

2.1 Residual Monitors

For default, the residuals monitors are created without any specification on the control file.

2.2 Statistics Monitor

```
#define statistics
   initial time      = 1.d0
   initial iteration = 10
   sampling interval = 10
   dump interval     = 20
   @start
#end
```

By default, the statistic monitor will average following variables:

- u
- uu
- uv
- v
- vv
- uw
- w
- ww
- vw

A keyword preceded by @ is used in real-time to signalize the solver what it must do with the statistics computation:

- @start
- @stop
- @dump
- @pause
- @reset

After reading the keyword, the solver performs the desired action and marks it with a star, e.g. @start*.

ATTENTION: Real-time keywords may not work in parallel MPI computations. I depends on how the system is configured.

2.3 Probes

```
#define probe 1
   name     = SomeName
   variable = SomeVariable
   position = [0.d0, 0.d0, 0.d0]
#end
```

Table 2.2: Keywords for probes.

Keyword	Description	Default value
name	*CHARACTER*: Name of the monitor.	**Mandatory Keyword**
variable	*CHARACTER*: Variable to be monitored. Implemented options are: • pressure • v • k • velocity • w • u • mach	**Mandatory Keyword**
position	*REAL(3)*: Coordinates of the point to be monitored.	**Mandatory Keyword**

2.4 Surface Monitors

```
#define surface monitor 1
   name              = SomeName
   marker            = NameOfBoundary
   variable          = SomeVariable
   reference surface = 1.d0
   direction         = [1.d0, 0.d0, 0.d0]
#end
```

Table 2.3: Keywords for probes.

Keyword	Description	Default value
name	*CHARACTER*: Name of the monitor.	**Mandatory Keyword**
marker	*CHARACTER*: Name of the boundary where a variable will be monitored.	**Mandatory Keyword**
variable	*CHARACTER*: Variable to be monitored. Implemented options are: • mass-flow • force • flow • lift • pressure-force • drag • viscous-force • pressure-average	**Mandatory Keyword**
reference surface	*REAL*: Reference surface [area] for the monitor. Needed for "lift" and "drag" computations.	–

Table 2.3: Keywords for the p-adaptation algorithms - continued.

Keyword	Description	Default value
direction	*REAL(3)*: Direction in which the force is going to be measured. Needed for "pressure-force", "viscous-force" and "force". Can be specified for "lift" (default [0.d0,1.d0,0.d0]) and "drag" (default [1.d0,0.d0,0.d0])	–

2.5 Volume Monitors

Volume monitors compute the average of a quantity in the whole domain. They can be scalars(s) or vectors(v).

```
#define volume monitor 1
   name     = SomeName
   variable = SomeVariable
#end
```

Table 2.4: Keywords for volume monitors.

Keyword	Description	Default value
name	*CHARACTER*: Name of the monitor.	**Mandatory Keyword**
variable	*CHARACTER*: Variable to be monitored. The variable can be scalar (s) or vectorial (v). Implemented options are: **(s)** kinetic energy **(s)** kinetic energy rate **(s)** enstrophy **(s)** entropy **(s)** entropy rate **(s)** mean velocity **(v)** velocity **(v)** momentum **(v)** source	**Mandatory Keyword**

Chapter 3

Running the Simulation

3.1 General configuration

The files described in Chapter 1 must be placed in specific directories in order to be read (or written). The recommended basic configuration is shown below, but a different one can be used as long as the paths in the control file are properly changed.

```
myDirectory
├── RESULTS
├── SETUP
│   ├── ProblemFile.f90
│   └── **libProblemFile (The files that result as the compilation of the problem file)
├── MESH
│   └── myMesh.mesh
└── myControlFile.control
```

The directory RESULTS must be created (even if is empty) before running HORSES3D. The solution files and monitors will be written in it. In the SETUP directory, the binary files that result as the compilation of the problem file should be placed before running HORSES3D (the use of the problem file is optional, but it is recommended). In the MESH directory, the input mesh files must be placed and the resulted mesh files will be written.

After the main directory structure is in placed, the simulation can be run with the following command (assuming that the bin directory of HORSES3D is on the PATH).

```
$ horses3d.ns myControlFile.control
```

In case of running in a cluster with a queue system, this command can be used in the running script.

Chapter 4

Postprocessing

For postprocessing the Simulation Results

4.1 Visualization with Tecplot Format: *horses2plt*

HORSES3D provides a script for converting the native binary solution files (*.hsol) into tecplot ASCII format (*.tec), which can be visualized in Pareview or Tecplot. Usage:

```
$ horses2plt SolutionFile.hsol MeshFile.hmesh <<Options>>
```

The options comprise following flags:

Table 4.1: Flags for *horses2plt*.

Flag	Description	Default value
--output-order=	*INTEGER*: Output order nodes. The solution is interpolated into the desired number of points.	Not Present
--output-basis=	*CHARACTER*: Either *Homogeneous* (for equispaced nodes, or *Gauss*.	*Gauss**
--output-mode=	*CHARACTER*: Either *multizone* or *FE*. The option *multizone* generates a Tecplot zone for each element. The option *FE* generates only one Tecplot zone for the fluid and one for each boundary (if *--boundary-file* is defined). Each subcell is mapped as a linear finite element. This format is faster to read by Paraview and Tecplot.	*multizone*
--output-variables=	*CHARACTER*: Output variables separated by commas.A complete description can be found in Section 4.1.1.	Q
--dimensionless	Specifies that the output quantities must be dimensionless	Not Present
--partition-file=	*CHARACTER*: Specifies the path to the partition file (*.pmesh) to export the MPI ranks of the simulation.	Not Present
--boundary-file=	*CHARACTER*: Specifies the path to the boundary mesh file (*.bmesh) to export the surfaces as additional zones of the Tecplot file.	Not Present

* *Homogeneous* when *--output-order* is specified

Additionally, depending on the type of solution file, the user can specify additional options.

4.1.1 Solution Files (*.hsol)

For standard solution files, the user can specify which variables they want to be exported to the Tecplot file with the flag *--output-variables=*. The options are:

- Q (default)
- v
- T
- $Vabs$
- $rhou$
- rho
- w
- $Mach$
- V
- $rhov$
- u
- p
- s
- Ht
- $rhow$

- *rhoe*
- *c*
- *Nxi*
- *Neta*
- *Nzeta*
- *Nav*
- *N*
- *Ax_Xi*
- *Ax_Eta*
- *Ax_Zeta*
- *ThreeAxes*
- *Axes*
- *mpi_rank*
- *eID*
- *gradV*
- *u_x*
- *v_x*
- *w_x*
- *u_y*
- *v_y*
- *w_y*
- *u_z*
- *v_z*
- *w_z*
- *c_x*
- *c_y*
- *c_z*
- *omega*
- *omega_x*
- *omega_y*
- *omega_z*
- *omega_abs*
- *Qcrit*

4.1.2 Statistics Files (*.stats.hsol)

Statistics files generate following variables by default (being Sij the components of the Reynolds Stress tensor):

- Umean
- Vmean
- Wmean
- Sxx
- Syy
- Szz
- Sxy
- Sxz
- Syz

4.2 Merge statistics tool

Tool to merge several statistics files. The usage is the following:

```
$ horses.mergeStats *.hsol --initial-iteration=INTEGER --file-name=
CHARACTER
```

Some remarks:

- Only usable with statistics files that are obtained with the "reset interval" keyword and/or with individual consecutive simulations.
- Only constant time-stepping is supported.
- Dynamic p-adaptation is currently not supported.

4.3 Mapping result to different mesh

HORSES3D addons, *horsesConverter*, has a capability to map result into different mesh file, with both have a consistent geometry. A control input file is required and must has name *horsesConverter.convert*. The template of control input file will be generated by default when executing *./horsesConverter* in a directory without the control file. Error message is given when at least one node point of the new mesh is not within any element of the old mesh. After completion, a new result file is generated and named *Result_interpolation.hsol*. The required keywords in the control file are described in table 4.2. Command to execute:

```
$ ./horsesConverter
```

Table 4.2: Keywords for *meshInterpolation*.

Keyword	Description	Default value
Task	*meshInterpolation*	
Mesh Filename 1	Location of the origin mesh (*.hmesh)	
Boundary Filename 1	Location of the origin boundary mesh (*.bmesh)	
Result 1	Location of the solution file with origin mesh (*.hsol)	
Mesh Filename 2	Location of the target mesh (*.hmesh)	
Boundary Filename 2	Location of the target boundary mesh (*.bmesh)	
Polynomial Order	Polynomial order of the target mesh	(1, 1, 1)

4.4 Generate OpenFOAM mesh

Another functionality of *horsesConverter* addons is to convert the mesh files, (*.hmesh) and (*.bmesh), into OpenFOAM format, the *polyMesh* folder. After completion, a folder named *foamFiles* is generated. OpenFOAM mesh files, i.e. *points*, *faces*, *owner*, *neighbour*, and *boundary*, are located within *foamFiles/constant/polyMesh*. The required keywords in the control file are described in table 4.3. Command to execute:

```
$ ./horsesConverter
```

Table 4.3: Keywords for *horsesMesh2OF*.

Keyword	Description	Default value
Task	*horsesMesh2OF*	
Mesh Filename 1	Location of the origin mesh (*.hmesh)	
Boundary Filename 1	Location of the origin boundary mesh (*.bmesh)	
Polynomial Order	Number of division of each element (n_x, n_y, and n_z)	(1, 1, 1)

NOTE: Before running the mesh in the OpenFOAM environment, the type of boundaries inside the boundary file needs to be adjusted according to the actual type (*patch*, *wall*, and *symmetry*).

4.5 Generate HORSES3D solution file from OpenFOAM result

HORSES3D provides a capability to convert OpenFOAM result into HORSES3D solution file (*.hsol). The mesh of the OpenFOAM result must be generated by converting HORSES3D mesh files, see section 4.4. Beforehand, the OpenFOAM result must be converted into VTK format(*.vtk). This not only allows the result to be in the single file but also converts cell data into point data. In the OpenFOAM environment, the command for this conversion:

```
$ foamToVTK -fields "(U p T rho)" -ascii -latestTime
```

The necessary file (.vtk) required for the control file input is inside VTK folder, see section 4.3 for the control file template. The HORSES3D solution file is named *Result_OF.hsol*. The required keywords in the control file are described in table 4.4. Command to execute:

```
$ ./horsesConverter
```

Table 4.4: Keywords for *OF2Horses*.

Keyword	Description	Default value
Task	*OF2Horses*	
Mesh Filename 1	Location of the origin mesh (*.hmesh)	
Boundary Filename 1	Location of the origin boundary mesh (*.bmesh)	
Polynomial Order	Polynomial order of the solution file (.hsol)	(1, 1, 1)
VTK file	Location of VTK file (.vtk)	
Reynolds Number	Reynolds Number/m of the solution – L_{ref}=1.0m	
Mach Number	Mach Number of the solution	
Reference pressure (Pa)	Reference Pressure	101325
Reference temperature (K)	Reference Temperature	288.889

Chapter 5

Compiling the code

- Clone the git repository or copy the source code into a desired folder.
- Go to the folder Solver.
- Run configure script.

```
$ ./configure
```

- Install using the Makefile:

```
$ make all <<Options>>
```

  with the desired options (bold are default):

  - MODE=DEBUG/**RELEASE**
  - COMPILER=ifort/**gfortran**
  - COMM=PARALLEL/**SEQUENTIAL**
  - ENABLE_THREADS=NO/**YES**
  - WITH_MKL=YES/**NO**

  For example:

```
$ make all COMPILER=ifort WITH_MKL=YES
```

  The HORSES3D tools are created in the Solver/bin directory.

- If you use *environment modules*, it is advised to use the HORSES3D module file:

```
$ export MODULEPATH=$HORSES_DIR/utils/modulefile:$MODULEPATH
```

  where $HORSES_DIR is the installation directory.

- It is advised to run the *make clean* command if some options of the compilation rutine needs to be changed and it has been compiled before.
- The compilation of the ProblemFile presented at Chapter 1 must be done with the same options as the HORSES3D code.