HORSES3D

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

User Manual

Andrés Rueda
Many others (future?)

July 17, 2019

Contents

Chapter 1

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**
  - PLATFORM=MACOSX/**LINUX**
  - ENABLE_THREADS=NO/**YES**

  For example:

```
$ make all COMPILER=ifort COMM=PARALLEL
```

- 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.

Chapter 2

Input and Output Files

2.1 Input Files

- Control file (*.control)
- Mesh file (*.mesh / *.h5)
- Polynomial order file (*.omesh)
- Problem File (ProblemFile.f90)

2.2 Output Files

- Solution file (*.hsol)
- Horses mesh file (*.hmesh)
- Boundary information (*.bmesh)
- Partition file (*.pmesh)
- Polynomial order file (*.omesh)
- Monitor files (*.volume / *.surface / *.residuals)

Chapter 3

Running a Simulation

3.1 Control File (*.control) - Overview

The control file is the main file for running a simulation. A list of all the mandatory keywords for running a simulation and some basic optional keywords is presented in Table 3.1. The specific keywords are listed in the other chapters.

Table 3.1: General keywords for running a case.

Keyword	Description	Default value
solution file name	*CHARACTER*: Path and name of the output file. The name of this file is used for naming other output files.	**Mandatory keyword**
simulation type	*CHARACTER*: Specifies if NSLITE3D must perform a 'steady-state' or a 'time-accurate' simulation.	'steady-state'
time integration	*CHARACTER*: Can be 'implicit', 'explicit', or 'FAS'. The latter uses the Full Algebraic Storage (FAS) multigrid scheme, which can have implicit or explicit smoothers.	'explicit'
polynomial order	*INTEGER*: Polynomial order to be assigned uniformly to all the elements of the mesh. If the keyword *polynomial order file* is specified, the value of this keyword is overridden.	–*
polynomial order i polynomial order j polynomial order k	*INTEGER*: Polynomial order in the i, j, or k component for all the elements in the domain. If used, the three directions must be declared explicitly, unless you are using a polynomial order file. If the keyword *polynomial order file* is specified, the value of this keyword is overridden.	–*
polynomial order file	*CHARACTER*: Path to a file containing the polynomial order of each element in the domain.	–*
restart	*LOGICAL*: If .TRUE., initial conditions of simulation will be read from restart file specified using the keyword *restart file name*.	**Mandatory keyword**
cfl	*REAL*: A constant related with the **convective** Courant-Friedrichs-Lewy (CFL) condition that the program will use to compute the time step size.	–**
dcfl	*REAL*: A constant related with the **diffusive** Courant-Friedrichs-Lewy (DCFL) condition that the program will use to compute the time step size.	–**
dt	*REAL*: Constant time step size.	–**
final time	*REAL*: This keyword is mandatory for time-accurate solvers	–
mesh file name	*CHARACTER*: Name of the mesh file. The currently supported formats are *.mesh* (SpecMesh file format) and *.h5* (HOPR hdf5 file format).	**Mandatory keyword**
mesh inner curves	*LOGICAL*: Specifies if the mesh reader must suppose that the inner surfaces (faces connecting the elements of the mesh) are curved. This input variable only affects the hdf5 mesh reader.	.TRUE.
number of time steps	*INTEGER*: *Maximum* number of time steps that the program will compute.	**Mandatory keyword**

Table 3.1: General keywords for running a case - continued.

Keyword	Description	Default value
output interval	*INTEGER*: 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.	**Mandatory keyword**
convergence tolerance	*REAL*: Residual convergence tolerance for steady-state cases	**Mandatory keyword**
manufactured solution	*CHARACTER*: Must have the value '2D' or '3D'. When this keyword is used, the program will add source terms for the conservative variables taken into account an exact analytic solution for each primitive variable j (ρ, u, v, w, p) of the form: $j = j_C(1) + j_C(2)\sin(\pi j_C(5)x) + j_C(3)\sin(\pi j_C(6)y) + j_C(4)\sin(\pi j_C(7)z)$ Where $j_C(i)$ are constants defined in the file *ManufacturedSolutions.f90*. Proper initial and boundary conditions must be imposed (see the test case). The mesh must be a unit cube.	–
Number of boundaries	*INTEGER*: Specifies the number of boundaries of the geometry. This keyword must be followed by an equal number of lines in the control file that define the boundary conditions as follows: *boundaryName1 boundaryValue1 boundaryType1* *boundaryName2 boundaryValue2 boundaryType2* ...	**Mandatory keyword**

* One of these keywords must be specified

** For Euler simulations, the user must specify either the CFL number or the time-step size. For Navier-Stokes simulations, the user must specify the CFL and DCFL numbers **or** the time-step size.

3.2 Boundary conditions

Under construction.

Chapter 4

Restarting a Case

Table 4.1: Keywords for restarting a case.

Keyword	Description	Default value
restart	*LOGICAL*: If .TRUE., initial conditions of simulation will be read from restart file specified using the keyword *restart file name*.	**Mandatory keyword**
restart file name	*CHARACTER*: Name of the restart file to be written and, if keyword *restart* = .TRUE., also name of the restart file to be read for starting the simulation.	**Mandatory keyword**
restart polorder	*INTEGER*: Uniform polynomial order of the solution to restart from. This keyword is only needed when the restart solution is of a different order than the current case.	same as case's
restart polorder file	*CHARACTER*: File containing the polynomial orders of the solution to restart from. This keyword is only needed when the restart solution is of a different order than the current case.	same as case's
get discretization error of	*CHARACTER*: Path to solution file. This can be used to estimate the discretization error of a solution when restarting from a higher-order solution.	–

Chapter 5

Physics related keyword

5.1 Compressible flow

Table 5.1: Keywords for compressible flow (Euler / Navier-Stokes).

Keyword	Description	Default value
Mach number	*REAL*:	**Mandatory keyword**
Reynolds number	*REAL*:	**Mandatory keyword**
Prandtl number	*REAL*:	0.72
Turbulent Prandtl number	*REAL*:	Equal to Prandtl
LES model	*CHARACTER(*)*: Options are: • Smagorinsky • None	None
Wall model	*CHARACTER(*)*:	linear

5.2 Incompressible Navier-Stokes

5.3 Cahn-Hilliard

Chapter 6

Implicit Solvers

6.1 General Keywords

The keywords for the implicit solvers are listed in table 6.1

Table 6.1: Keywords for implicit solvers.

Keyword	Description	Default value
time integration	*CHARACTER*: This is the main keyword for activating the implicit solvers. The value of it should be set to 'implicit' for the BDF solvers and to 'rosenbrock' for Rosenbrock schemes.	'explicit'
jacobian by convergence	*LOGICAL*: When .TRUE., the Jacobian is only computed when the convergence falls beneath some threshold (see keyfords: blah and blah blah). This improves performance but can introduce big numerical errors for time-accurate simulations.	.FALSE.
linear solver	*CHARACTER*: Specifies the linear solver that has to be used. Options are: • 'petsc': PETSc library Krylov-Subspace methods. Available in serial, but use with care (PETSc is not thread-safe, so OpenMP is not recommended). Only available in parallel (MPI) for preallocated Jacobians (see next section). • 'pardiso': Intel MKL PARDISO. Only available in serial or with OpenMP. • 'matrix-free gmres': A matrix-free version of the GMRES algorithm. Can be used without preconditioner or with a recursive GMRES preconditioner using 'preconditioner=GMRES'. Available in serial and parallel (OpenMP+MPI) • 'smooth': Traditional iterative methods. One can select either 'smoother=WeightedJacobi' or 'smoother=BlockJacobi'. • 'matrix-free smooth': A matrix-free version of the previous solver. Only available with 'smoother=BlockJacobi'.	'petsc'
print newton info	*LOGICAL*: If .TRUE., the information of the Newton iterations will be displayed.	'.FALSE.'
implicit adaptive dt	*LOGICAL*: Specifies if the time-step should be computed according to the convergence behavior of the Newton iterative method and the linear solver.	.FALSE.

Table 6.1: Keywords for implicit solvers - continued.

Keyword	Description	Default value
newton tolerance	*REAL*: Specifies the tolerance for the Newton method.	10^{-6} or for time-accurate simulations and $MaxResidual \times 10^{-3}$ for steady-state simulations
max newton iter	*INTEGER*: Maximum number of Newton iterationsfor BDF solver.	30
compute jacobian every	*INTEGER*: Forces the Jacobian to be computed in an interval of iterations that is specified.	Inf
bdf order	*INTEGER*: If present, the solver uses a BDF solver of the specified order. BDF1 - BDF5 are available, and BDF2 - BDF5 require constant time steps.	1

6.2 Jacobian Specifications

The Jacobian must be defined using a block of the form:

```
#define Jacobian
    type = 2
    print info = .TRUE.
    preallocate = .TRUE.
#end
```

Table 6.2: Keywords for Jacobian definition block.

Keyword	Description	Default value
type	*INTEGER*: Specifies the type of Jacobian matrix to be computed. Options are: 1. Numerical Jacobian: Uses coloring algorithm for computing Jacobian (only available with shared memory parallelization). 2. Analytical Jacobian: Available with shared (OpenMP) or distributed (MPI) memory parallelization for advective and/or diffusive nonlinear conservation laws.	**Mandatory Keyword**
print info	*LOGICAL*: Specifies the verbosity of the Jacobian subroutines	.TRUE.
preallocate	*LOGICAL*: Specifies if the Jacobian must be allocated in preprocessing (.TRUE. - only available for advective/diffusive nonlinear conservation laws) or every time it is computed (.FALSE.)	.FALSE.

.TRUE.=Preallocate / .FALSE.=Perform allocation every time the Jacobian is constructed

6.3 Multigrid

Table 6.3: Keywords for the multigrid solver.

Keyword	Description	Default value
multigrid levels	*INTEGER*: Number of multigrid levels for the computations.	**Mandatory keyword**

Table 6.3: Keywords for the multigrid solver - continued.

Keyword	Description	Default value
delta n	*INTEGER*: Interval of reduction of polynomial order for creating coarser multigrid levels.	1
multigrid output	*LOGICAL*: If .TRUE., the residuals at the different multigrid levels will be displayed.	.FALSE.

Chapter 7

p-Adaptation Methods

The p-adaptation methods are used when the p-adaptation region is specified in the control file:

```
#define p-adaptation
    Truncation error type = isolated
    truncation error      = 1.d-2
    truncation error      = 1.d-4
    Nmax                  = [10,10,10]
    Nmin                  = [2 ,2 ,2 ]
    Conforming boundaries = [InnerCylinder,sphere]
    order across faces    = N*2/3
    increasing            = .FALSE.
    write error files     = .FALSE.
    adjust nz             = .FALSE.
    mode                  = time
    interval              = 1.d0
    restart files         = .TRUE.
    max N decrease        = 1
    padapted mg sweeps pre      = 10
    padapted mg sweeps post     = 12
    padapted mg sweeps coarsest = 20
#end
```

Table 7.1: Keywords for the p-adaptation algorithms.

Keyword	Description	Default value
truncation error type	*CHARACTER*: Can be either "isolated" or "non-isolated".	isolated
truncation error	*REAL*: Target truncation error for the p-adaptation algorithm.	**Mandatory keyword**
coarse truncation error	*REAL*: Truncation error used for coarsening.	same as truncation error
Nmax	*INTEGER*(3): Maximum polynomial order in each direction for the p-adaptation algorithm.	**Mandatory keyword**
Nmin	*INTEGER*(3): Minimum polynomial order in each direction for the p-adaptation algorithm.	[1,1,1]
conforming boundaries	*CHARACTER*(*): Specifies the boundaries of the geometry that must be forced to be conforming after the p-adaptation process.	–
order across faces	*CHARACTER*: Mathematical expression to specify the maximum polynomial order jump across faces. Currently, only $N*2/3$ and $N-1$ are supported.	$N-1$
increasing	*LOGICAL*: If .TRUE. the multi-stage FMG adaptation algorithm is used.	.FALSE.

Table 7.1: Keywords for the p-adaptation algorithms - continued.

Keyword	Description	Default value
write error files	*LOGICAL*: If .TRUE., the program writes a file per element containing the directional tau-estimations. The files are stored in the folder *./TauEstimation/*. When the simulation has several adaptation stages, the new information is just appended.	.FALSE.
adjust nz	*LOGICAL*: If .TRUE., the order accross faces is adjusted i n the directions xi, eta, and zeta of the face (being zeta the normal direction). If .FALSE., the order is only adjusted in the xi and eta directions. The adjustment currently consists (hard-cod ed) in allowing jumps in the polynomial order of at most 1.	.FALSE.
mode	*CHARACTER*: p-Adaptation mode. Can be *static*, *time* or *iteration*. Static p-adaptation is performed once at the beginning of a simulation for steady or unsteady simulations. Unsteady adaptation can be by *time* or by *iteration*.	*static*
interval	*INTEGER/REAL*: In dynamic p-adaptation cases, this keyword specifies the iteration (integer) or time (real) interval for p-adaptation.	*huge number*
restart files	*LOGICAL*: If .TRUE., the program writes restart files before and after the p-adaptation.	.FALSE.
max N decrease	*INTEGER*: Maximum decrease in the polynomial order in every p-adaptation procedure.	$N - N_{min}$
post smoothing residual	*REAL*: Specifies the maximum allowable deviation of $\partial_t q$ after the p-adaptation procedure.	–
post smoothing method	*CHARACTER*: Either RK3 or FAS.	RK3, if the last keyword is activated
estimation files	*CHARACTER*: Name of the folder that contains the error estimations obtained with the multi tau-estimation (section 7.1).	–
estimation files number	*INTEGER(2)*: First and last estimation stages to be used for p-adaptation.	Mandatory if last keyword is used.
padapted $\ll$ *keyword* $\gg$	*MULTIPLE*: Specifies control file keywords that should be replaced after the adaptation procedure. Currently, only 'mg sweeps ', 'mg sweeps pre', 'mg sweeps post', and 'mg sweeps coarsest' are supported.	–

7.1 Multiple truncation error estimations

A static p-adaptation procedure can be driven by a set of error estimations, which have to be performed beforehand in a simulation with the following block:

```
#define multi tau-estimation
   truncation error type = isolated
   interval              = 10
   folder                = MultiTau
#end
```

Chapter 8

Monitors

8.1 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

8.2 Probes

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

Table 8.1: Keywords for probes.

Keyword	Description	Default value
name	*CHARACTER*: Name of the monitor.	**Mandatory Keyword**

Table 8.1: Keywords for the p-adaptation algorithms - continued.

Keyword	Description	Default value
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**

8.3 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 8.2: 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.	–
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])	–

8.4 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 8.3: 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 9

Advanced User Setup

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 the Problem file (*ProblemFile.f90*). A description of the routines of the Problem File can be found in section 9.1.

9.1 Routines of the Problem File: *ProblemFile.f90*

- UserDefinedStartup: Called before any other routines
- UserDefinedFinalSetup: Called after the mesh is read in to allow mesh related initializations or memory allocations.
- UserDefinedInitialCondition: called to set the initial condition for the flow. By default it sets an uniform initial condition, but the user can change it.
- UserDefinedState1, UserDefinedNeumann: Used to define an user-defined boundary condition.
- UserDefinedPeriodicOperation: Called before every time-step to allow periodic operations to be performed.
- UserDefinedSourceTermNS: Called to apply source terms to the equation.
- UserDefinedFinalize: Called after the solution computed to allow, for example error tests to be performed.
- UserDefinedTermination: Called at the the end of the main driver after everything else is done.

9.2 Compiling the Problem File

The Problem Fie file must be compiled using a specific Makefile that links it with the libraries of the code. If you are using the *horses/dev* environment module, you can get templates of the *Problemfile.f90* and *Makefile* with the following commands:

```
$ horses-get-makefile
$ horses-get-problemfile
```

Otherwise, search the test cases for examples.

To run a simulation using user-defined operations, create a folder called SETUP on the path were the simulation is going to be run. Then, store the modified *ProblemFile.f90* and the *Makefile* in SETUP, and compile using:

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

where again the options are (bold are default):

- MODE=DEBUG/**RELEASE**
- COMPILER=ifort/**gfortran**
- COMM=PARALLEL/**SEQUENTIAL**

- PLATFORM=MACOSX/**LINUX**
- ENABLE_THREADS=NO/**YES**

Chapter 10

Postprocessing

For postprocessing the Simulation Results

10.1 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 10.1: Flags for *horses2plt*.

Flag	Description	Default value
–output-order=	*INTEGER*: Output order for equispaced nodes. The solution is interpolated in the desired number of points.	Not Present
–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-mesh=	*CHARACTER*: Specifies the path to the boundary mesh file (*.bmesh) to export the surfaces as additional zones of the Tecplot file.	Not Present

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

10.1.1 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)
- rho
- u
- v
- w
- p
- T
- $Mach$
- s
- $Vabs$
- V
- Ht
- $rhou$
- $rhov$
- $rhow$
- $rhoe$
- c
- Nxi
- $Neta$
- $Nzeta$
- Nav
- N
- Ax_Xi
- Ax_Eta
- Ax_Zeta
- $ThreeAxes$
- $Axes$
- mpi_rank
- $gradV$
- 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_x*
- *omega_z*
- *Qcrit*
- *omega*
- *omega_y*
- *omega_abs*

10.1.2 Statistics Files (*.stats.hsol)

Statistics files generate following variables by default:

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

10.2 Extract geometry

Under construction.

10.3 Merge statistics tool