CFD Consulting & Analysis

Dr.-Ing. Jiri Blazek

Snas3D

User's Guide

Version 8.0 from October 8, 2022

Snas3D – User’s Guide
Version 8.0

http://www.cfd-ca.de/

Sales information: info@cfd-ca.de
Technical support: support@cfd-ca.de

Contents

End-User License Agreement

CFD Consulting & Analysis (“CFD-CA”) licenses the software package ***Snas3D*** (“SOFTWARE”) to you as an individual or a legal entity (referenced in the following as “YOU” or “YOUR”), provided YOU accept all terms in this end-user license agreement (“AGREEMENT”). YOU assume responsibility for the selection of the SOFTWARE to achieve YOUR intended results, and for the installation, use, and results obtained from the SOFTWARE. Installing or using any part of the SOFTWARE constitutes YOUR acceptance of the AGREEMENT. If YOU do not accept these terms and conditions, then do not install the SOFTWARE and destroy all copies of it and of its documentation.

License

YOU may install the SOFTWARE and documentation on any number of YOUR machines. YOU or YOUR users may employ the SOFTWARE and use the documentation on any number of processors or processor cores at the same time. YOU may copy the SOFTWARE and documentation into any machine-readable or printed form for backup or support of YOUR use of the SOFTWARE and documentation.

YOU may not use, copy, transfer, sublicense, loan, lend, lease, distribute, rent, modify, translate, disassemble, reverse engineer, or create derivative works based upon the SOFTWARE or documentation, or any copy thereof, in whole or in part, except as provided for in this AGREEMENT. If YOU use, copy, transfer, sublicense, loan, lend, lease, distribute, rent, modify, translate, disassemble, reverse engineer, or create derivative works based upon the SOFTWARE or documentation, or any copy thereof, in whole or in part, except as expressly provided for in this AGREEMENT, YOUR license is automatically terminated.

The license becomes effective on the date YOU accept this AGREEMENT, and remains in effect for the licensing period agreed upon with CFD-CA, or until terminated as indicated above, or until YOU terminate it. If the license expires or is terminated for any reason, YOU agree to destroy the SOFTWARE and documentation, together with all copies thereof, in whole or in part, in any form, and to cease all use of the SOFTWARE and documentation.

Limited Warranty and Limitation of Remedies

The SOFTWARE, documentation and any support from CFD-CA, are provided “as is” and without warranty, express and implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose. In no event shall CFD-CA be liable for any direct, indirect, incidental, general, special, exemplary, or consequential damages arising out of the use or inability to use the SOFTWARE (including but not limited to loss of data, profit or savings; or data being rendered inaccurate; or losses sustained by you or third parties; or a failure of the SOFTWARE to operate with any other programs), even if CFD-CA is advised of the possibility of such damages, or for any claim by YOU or any third party.

General Terms

This AGREEMENT can only be modified by a written AGREEMENT signed by YOU and CFD-CA and changes from the terms and conditions of this AGREEMENT made in any other manner will be of no effect. If any portion of this AGREEMENT shall be held invalid, illegal, or unenforceable, the validity, legality, and enforceability of the remainder of the AGREEMENT shall not in any way be affected or impaired thereby. This AGREEMENT shall be governed by the laws of Bundesrepublik Deutschland (Germany), without giving effect to conflict of law’s provisions thereof.

Acknowledgment

YOU acknowledge that YOU have read this AGREEMENT, understand it, and agree to be bound by its terms and conditions. YOU further agree that it is the complete and exclusive statement of the AGREEMENT between YOU and CFD-CA which supersedes all proposals or prior agreements, oral or written, and all other communications between YOU and CFD-CA relating to the subject matter of this AGREEMENT.

1. General Description

Snas3D is a general purpose, 3-D flow solver for the solution of the Euler or the Navier-Stokes equations on structured, multiblock grids with conforming boundaries. Formulated for moving and/or deforming grids, the governing equations read in integral form

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{W}\,d\Omega + \oint_{\partial\Omega}\left(\vec{F}_c^M - \vec{F}_v\right)dS = \int_{\Omega}\vec{Q}\,d\Omega\,, \tag{1}$$

where $\vec{W}$ denotes the vector of the conservative variables

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix}, \tag{2}$$

$\vec{F}_c^M$ the convective fluxes on moving grid, $\vec{F}_v$ the viscous fluxes, and Ω is the control volume with the surface $\partial\Omega$. Furthermore, dS represents a surface element of $\partial\Omega$ and $\vec{Q}$ stands for the source term. The vector of convective fluxes on moving grid $\vec{F}_c^M$ is given by

$$\vec{F}_c^M = \vec{F}_c - V_t\vec{W} \tag{3}$$

with $\vec{F}_c$ denoting the standard convective fluxes and V_t being the contravariant velocity of the face of the control volume. Hence,

$$V_t = n_x\frac{\partial x}{\partial t} + n_y\frac{\partial y}{\partial t} + n_z\frac{\partial z}{\partial t}. \tag{4}$$

In the above Eq. (4), n_x, n_y and n_z represent components of the outward facing unit normal vector of the surface $\partial\Omega$.

In order to avoid errors induced by a deformation of the control volumes, the Geometric Conservation Law (GCL) must be satisfied. The integral form of the GCL reads [1]

$$\frac{\partial}{\partial t}\int_{\Omega}d\Omega - \oint_{\partial\Omega}V_t dS = 0. \tag{5}$$

Simultaneous solution of Eq. (1) and (5) is also termed the Arbitrary Lagrangian-Eulerian (ALE) method for dynamic grids.

Snas3D integrates the governing equations (1) on a hexahedral structured grid, which may be composed of any number of grid blocks. The spatial discretization employs cell-centered finite-volume methodology. It offers the choice between central scheme with scalar artificial dissipation [2] and Roe's upwind (flux-difference splitting) scheme [3].

Two different explicit methods are available for the integration of the governing equations (1) in time. The first one is a 5-stage hybrid scheme which is applicable in the case of stationary flow. It is accelerated by local time stepping, implicit residual smoothing and multigrid. The second explicit scheme is the classical 4-stage Runge-Kutta method, which is 4th-order accurate in time. A further approach available for unsteady flows is a 2nd-order dual-time stepping scheme, which allows for much larger time steps due to its implicit nature. It uses the above hybrid scheme and multigrid for an efficient solution of the equations [4].

Snas3D is prepared for the implementation of various fluid models and for additional species equations. Presently, the fluid model assumes a thermally and calorically perfect gas. Dynamic viscosity of the gas can be either calculated from the Sutherland's law or it can be set to a constant value. Turbulent flows are modeled using the Spalart-Allmaras one-equation model [5], [6]. Thanks to low Mach-number preconditioning [7], the flow solver is capable of simulating incompressible, compressible and mixed-type flows.

Currently, the following boundary conditions can be specified by the user:

- Slip and no-slip walls (adiabatic or with given temperature)
- Injection
- Inflow and outflow
- Far-field
- Symmetry
- Singular line
- Rotational or translational periodicity

The flow solver has furthermore interfaces for receiving and also for sending boundary data from/to an external driver program. This enables the coupling of ***Snas3D*** to shape optimizers or to multi-physics simulation programs. Furthermore, the solver can receive information about the movement of specific boundaries, and then utilize its own algorithms to move the interior grid.

The numerical algorithms of ***Snas3D*** are accelerated using parallel programing based on the Message Passing Interface (MPI) and on block decomposition. A more detailed description of the numerical schemes and the boundary conditions can be found in [8].

2. Basic Program Usage

Besides the flow solver itself, the program package contains utility programs, which are required in order to prepare all the necessary data for the solver, and to post-process the simulation results. Thus, the program package consists of:

- `snprep` – pre-processor to generate the initial flow solution
- `snsolv` or `snsolvpara` – ***Snas3D*** flow solver (serial or parallel version)
- `snpost` – post-processor to convert grid and solution file(s) into Tecplot's[1] or into Vis3D's[2] ASCII format for visualization purposes
- `snsplit` – utility program to split single-block grids into multiple blocks and to convert grid files between ASCII and binary formats
- `plt3dsnas3d` – utility program to convert grids from Plot3D format (ASCII, multiblock)

The executables of the above programs reside in the directory `bin` of the distribution. The commercial version of ***Snas3D*** requires further the presence of a valid license file (`license`).

The execution of ***Snas3D*** depends on the following five files with:

- Grid
- Topology (description of the boundary patches)
- User input
- Boundary conditions
- Solution (must always be present – the flow solver does not generate an initial solution)

It should be noted that there is just one grid and one solution file for all grid blocks. Also the other files are common to all blocks. There can be additional input files, which store distributions of flow quantities for boundary patches. Furthermore, ***Snas3D*** can generate so-called probe files, where values of the density, the velocity components, the static pressure and temperature are stored along with the physical time or the iteration number for a given location within the flow field. Finally, ***Snas3D*** also saves the convergence history and optionally the thrust into separate files.

Grid and solution files can be provided either in native ASCII or in binary format. All other files are in ASCII format. Convergence history, probe and thrust files are formatted to be readable either by Tecplot[1] or by Vis2D[2]. All files must have the name of the flow case (specified on the command line when invoking any of the programs) and an additional extension:

- `.rgra` for grid file in ASCII format, `.rgrb` for binary format
- `.rtp` for topology
- `.rin` for user input
- `.rbc` for boundary values
- `.rsoa` for solution file in ASCII format, `.rsob` for binary format
- `_rprobe_0000.tec` or `_rprobe_0000.v2d` for probe file(s)
- `_rthrust.tec` or `_rthrust.v2d` for thrust
- `_rconv.tec` for convergence history

Note that all physical values have to be specified in Standard International (SI) units. Thus, grid coordinates have to be provided in meters, velocities in meters per second, pressure in Pascal, temperature in degree Kelvin, and so on. All output from ***Snas3D*** is in SI units as well. Keep this in mind when working with grids or flow data specified in Imperial units.

The following Chapters present the details of the various files and explain the usage of the flow solver and of the helper programs.

---

[1] Commercial visualization package: http://www.tecplot.com
[2] Visualization software of CFD Consulting & Analysis

3. Grid and Topology Files

Snas3D uses its own grid format, which can be either ASCII or binary. Single grid file contains the data of all blocks. Grid files have the extension `.rgra` for the ASCII format and `.rgrb` for the binary format. In cases where the grid is moving and/or deforming, a time stamp of the form `_0.00000E±00` (format 1PE11.5) is added to the extension. Hence, the file

`casename.rgrb_1.23000E-01`

contains grid data at the time t = 0.123 seconds.

Grid file in the ASCII format starts with a header line, containing single floating-point number which represents the physical time of the grid (0.0 for stationary grid). For every grid block, there is then a line containing the number of the block (counting starts at 1) and the numbers of grid cells in i-, j-, k-direction. Next line contains the grid coordinates, x-values being the first, y- and z-values following. There are no line breaks between the coordinates. Also, all values can be in free format. Thus, an ASCII grid file for ***Snas3D*** can be written out like in this Fortran code snippet:

```
open(10, file='example_grid.rgra', form='formatted', status='unknown')
write(10, *) 0.0 ! initial time
do m=1,nblocks
  nmax = ni(m)*nj(m)*nk(m) ! total number of nodes
   write(10, *) m, ni(m)-1, nj(m)-1, nk(m)-1
   write(10, *) (x(m,i), i=1,nmax), &
                (y(m,i), i=1,nmax), &
                (z(m,i), i=1,nmax)
enddo
close(10)
```

The binary grid format (Fortran) is similar, there are just no separate lines.

The topology of a grid (regardless whether single- or multiblock) is stored in a separate file with the extension `.rtp`. The topology file holds the number of blocks and specifies for each block the enclosing boundary patches together with the type of their boundary condition. The header of the topology file consists of two lines with comments, followed by a line specifying the total number of blocks and grid levels. For each grid block, the topology file contains one line with the block number (counting starts at 1). A second line stores the number of patches defining the block boundaries, as well as the numbers of physical cells in i-, j- and k-direction. For each boundary patch there is a single line with 12 columns:

1. Type of the boundary condition (values itself are provided in the `.rbc` file):
   - 10 – inflow
   - 20 – outflow
   - 30 – block-coupling boundary (boundary shared by 2 blocks; continuous grid)
   - 60 – slip (Euler) wall
   - 70 – no-slip (viscous) wall
   - 80 – far-field
   - 90 – injection
   - 100 – symmetry
   - 110 – translational periodicity
   - 120 – rotational periodicity (rotational axis is assumed to coincide with the x-axis)
   - 130 – singular line
2. Number of the block face (see Figure 1)
3. Start index of the first patch coordinate (cf. Figure 2)
4. End index of the first patch coordinate
5. Start index of the second patch coordinate (l_1 in Figure 2)
6. End index of the second patch coordinate
7. Index of the source block (or 0 if the patch is neither shared by 2 blocks, nor is periodic)
8. Face number of the source block (Figure 1)
9. Start index of the first patch coordinate of the source patch (Figure 2)

10. End index of the first patch coordinate of the source patch
11. Start index of the second patch coordinate of the source patch
12. End index of the second patch coordinate of the source patch

Note that the boundary patches must be **non-overlapping**. However, their number (even on a single block side) is arbitrary. This allows the specification of different boundary conditions on the same block boundary. Obviously, the patches must cover completely all boundaries of a grid block. The origin, the height and the width of each patch have to be stored. For this purpose, coordinates L1BEG, L1END, L2BEG and L2END are used in Figure 2. The local coordinate system of the patch l_1, l_2 is oriented according to the cyclic directions. This means, that if we consider the i-coordinate, j and k will be the first and the second cyclic direction. In the case of the j-coordinate, the cyclic directions will become k and i, respectively. Therefore, since the patch in Figure 2 is on the j = JBEG boundary, the l_1-coordinate is oriented in the k-direction and l_2 in the i-direction. The application of the cyclic directions allows for a unique definition of the patch orientation.

In the case of block-coupling or periodic boundaries, those local patch coordinates l_1, l_2 of the current and the source block which correspond to each other are denoted by the minus sign (l_1 could correspond to l_2 of the source patch). The end indices of l_1, l_2 of the source patch can be smaller than their start indices (this occurs, e.g., in the case of C-type coordinate cuts).

To give an example, the topology file for a grid consisting of two blocks, with inflow, outflow and symmetry boundaries, might look like this:

```
# Topology file for 3-D channel with bump (2 blocks)
#
  2  4  ! total no. of blocks, no. of grid levels
  1     ! block number
  6   48   32   16  ! no. of patches, icells, jcells, kcells
 10 1   1   32   1   16  0  0   0    0   0    0
 30 2  -1  -32   1   16  2  1  -1  -32   1   16
100 5   1   48   1   32  0  0   0    0   0    0
100 6   1   48   1   32  0  0   0    0   0    0
 60 3   1   16   1   48  0  0   0    0   0    0
100 4   1   16   1   48  0  0   0    0   0    0
  2
  6   48   32   16
 30 1  -1  -32   1   16  1  2  -1  -32   1   16
 20 2   1   32   1   16  0  0   0    0   0    0
100 5   1   48   1   32  0  0   0    0   0    0
100 6   1   48   1   32  0  0   0    0   0    0
 60 3   1   16   1   48  0  0   0    0   0    0
100 4   1   16   1   48  0  0   0    0   0    0
```

Note that the patches can be stored in an arbitrary order: Patch at block side 1 can be directly followed by a patch at block side 5 and so on.

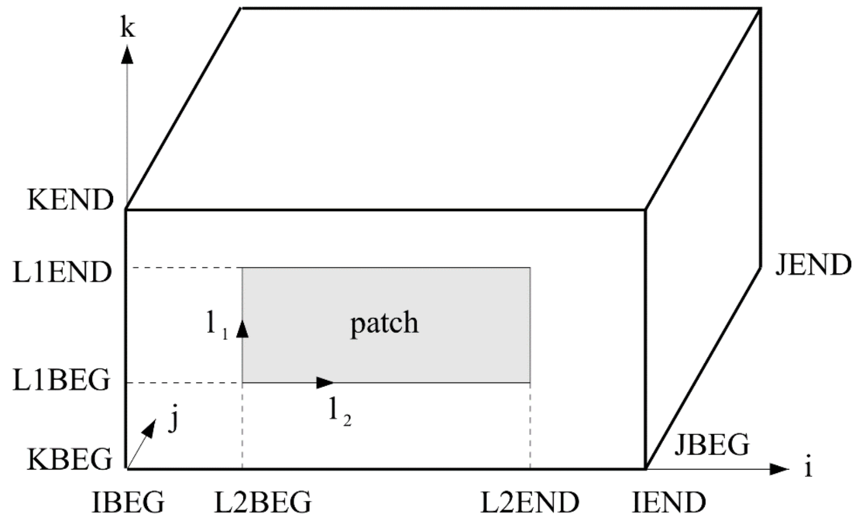

Figure 1: Numbering of the sides of the computational space and of the block boundaries.

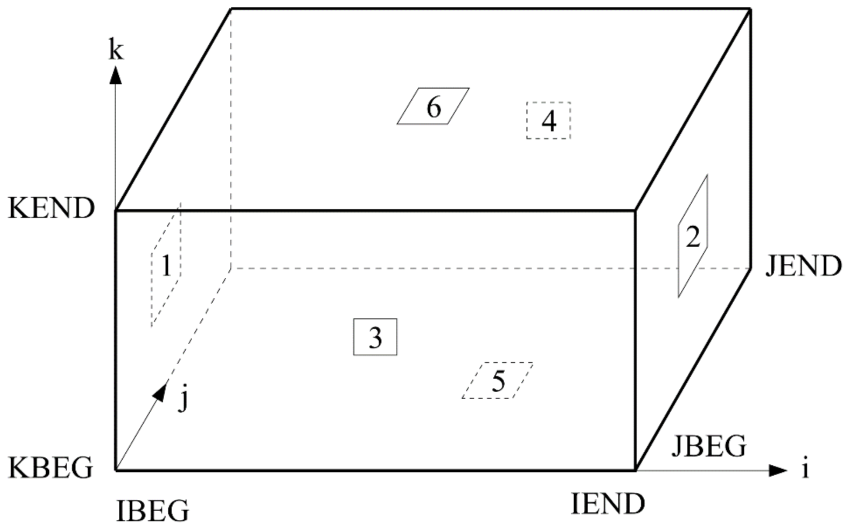

Figure 2: Coordinates of a boundary patch in the computational space. The patch has its own local coordinate system l_1, l_2. It is based on cyclic directions.

4. User Input File

The user input file has the extension `.rin`. It consists of a number of sections (like reference values, time stepping, numerical data, etc.). Certain number of options is associated with each section. Keywords are employed to distinguish between the sections and to designate particular user options. Begin of each section is marked by:

```
# SECTION_KEYWORD
```

where the section keyword has to be in capital letters. The end of a section is denoted by the hash sign (#). The ordering of sections within the file is free and sections can be repeated. All text outside the sections and between the options is discarded. The options consist of a keyword (must be upper case) and of one or multiple integer, real or character values. There can be a comment behind the value(s). Comments are denoted by an exclamation mark (!). No particular ordering is enforced for options within a section. However, it is important that the section designation and options are positioned at the beginning of a line in order to be recognized by the parser. Note also that all physical values have to be specified in Standard International (SI) units.

The user input file contains the following sections:

- FORMATS – formats of the grid, solution, convergence, probe and thrust files (4.1)
- INITFLOW – generation of the initial solution (4.2)
- FLOWMODEL – inviscid or viscous flow, moving grid (4.3)
- REFERENCE – reference values (4.4)
- VISCMODEL – model for computing the laminar viscosity (4.5)
- TURBULENCE – turbulence modeling and solution of the turbulence equations (4.6)
- ACCELERATION – type and values of acceleration terms (4.7)
- GRIDMOTION – grid motion settings (4.8)
- TIMESTEP – time stepping (4.9)
- MULTIGRID – multigrid or successive grid refinement (4.10)
- NUMERICS – parameters of the numerical method (4.11)
- FORCES – computation of pressure and viscous forces (4.12)
- PROBE – number and position(s) of probe(s) (4.13)
- THRUST – location of the thrust plane and the type of thrust (4.14)

An example of user input file with all currently available options and their default settings is provided in **`input_example.rin`**. In case some of the options or sections are not found in the input file, ***Snas3D*** supplies default values for the parameters. It is therefore advisable to check the parameters printed out by the flow solver, whether all have their intended values.

4.1 FORMATS Section

It contains options for specifying file formats:

- GRID – grid format: 0 = ASCII, 1 = binary
- SOLUTION – solution format: 0 = ASCII, 1 = binary
- CONVER – format of convergence history: 0 = Tecplot, 1 = Vis2D
- PROBE – format of probe data: 0 = Tecplot, 1 = Vis2D
- THRUST – format of thrust history: 0 = Tecplot, 1 = Vis2D

4.2 INITFLOW Section

It contains options for setting initial flow values separately for each block:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- VELX – velocity component in x-direction [m/s]

- VELY – velocity component in y-direction [m/s]
- VELZ – velocity component in z-direction [m/s]
- PRESS – static pressure [Pa]
- DENS – density [kg/m^3]

If some of the above values are changed, it is important to run `snprep` again in order to generate an updated initial solution file.

4.3 FLOWMODEL Section

It contains options for specifying the flow model and grid motion separately for each block:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- MODEL – solution of either Euler (0) or Navier-Stokes equations (1)
- MOVEGRID – grid is moving (0 = no, 1 = yes)

4.4 REFERENCE Section

It contains options for setting reference values:

- ABSVEL – velocity magnitude [m/s]
- PRESS – static pressure [Pa]
- DENS – density [kg/m^3]
- CP – specific heat coefficient at constant pressure c_p [J/(kgK)]
- GAMMA – ratio of specific heats c_p/c_v
- LENGTH – length [m]
- VISCLAM – laminar viscosity [kg/(ms)]
- PRLAM – laminar Prandtl number
- PRTURB – turbulent Prandtl number
- SCNLAM – laminar Schmidt number
- SCNTURB – turbulent Schmidt number

Laminar viscosity is used together with the density, the velocity magnitude and the length to calculate the Reynolds number. Optionally, the reference viscosity is modified according to the local temperature using the Sutherland's law. This depends on the settings in the VISCMODEL section described below.

Reference values are also used when computing limiter functions (Roe's upwind scheme) and for post-processing (calculation of pressure coefficients).

4.5 VISCMODEL Section

It contains options for specifying viscosity model and its options for each block:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- MODEL – viscosity model: 0 = Sutherland's law, 1 = fixed, 2 = Antibes formula
- SUTHCOEF – coefficient in Sutherland's law (110.0)
- REFTEMP – reference temperature for Sutherland's law (288.16)

4.6 TURBULENCE Section

It contains options for specifying turbulence modelling for each block:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- TURBMODEL – kind of turbulence model: 0 = laminar, 1 = Spalart-Allmaras
- TURBITER – offset between iterations to re-evaluate turbulence equations
- CFL – CFL number (turbulence equations only)
- SMOOCF – coefficient of implicit residual smoothing (<0: no smoothing)
- ENTRLIN – entropy correction for linear waves (turbulence equations only) [m/s]

4.7 ACCELERATION Section

It contains options for setting acceleration values:

- TYPE – acceleration terms: 0 = off, 1 = on
- ACCELX – acceleration in x-direction [m/s^2]
- ACCELY – acceleration in y-direction [m/s^2]
- ACCELZ – acceleration in z-direction [m/s^2]

4.8 GRIDMOTION Section

It contains options for specifying type and parameters of grid motion:

- TYPE – type of grid motion: 0 = region-wise, 1 = global
- NITER – number of Jacobi iterations (if TYPE = 1)
- WEIGHT – weighting factor for inverse node distance (if TYPE = 1)

Grid motion is enabled for a block or a range of blocks by setting the option MOVEGRID of the FLOWMODEL section (4.3) to 1.

4.9 TIMESTEP Section

It contains options for setting time-stepping parameters:

- FLOWTYPE – type of flow: 0 = steady, 1 = unsteady
- PRECONDON – low Mach-number preconditioning: 0 = off, 1 = on
- PRECONDCOEF – preconditioning parameter (typical values: 0.15-3.0)
- SOLVERTYPE – type of temporal discretization: 0 = explicit, 1 = implicit (means dual-time stepping if FLOWTYPE = 1)
- TIMESCHEME – type of scheme: 0 = explicit Runge-Kutta (the only option for now)

In the case of a steady flow, the following options apply:

- STARTITER – current iteration number (restart performed if greater than 0)
- MAXITER – last iteration (simulation is stopped when this number is reached)
- RESTOL – maximum density residual to stop the iteration process as having reached the steady state
- WRIITER – offset between iterations to store intermediate solutions
- PRNITER – offset between iterations to print convergence data to the screen and to store them in the convergence file

The following options apply in the case of an unsteady flow:

- STARTTIME – current physical time in seconds (restart performed if greater than 0.0)
- MAXTIME – maximum physical time for which to run the simulation
- TIMESTEP – maximum time step. If the time step based on the CFL-condition is smaller, multiple steps are performed
- WRITIME – time offset to store intermediate solutions
- PRNTIME – time offset to print convergence data to the screen and to store them in the convergence file

If the flow is unsteady (FLOWTYPE = 1) and dual-time stepping is selected (SOLVERTYPE = 1), the following additional options apply:

- ORDER – order of the time-stepping scheme: 2 = 2nd-oder (the only option for now)
- MAXSUBITER – maximum number of sub-iterations
- TOLSUBITER – convergence tolerance of sub-iterations

It is recommended to accelerate the sub-iterations of the dual-time stepping scheme by multigrid (with the option START of the MULTIGRID section set to 1).

4.10 MULTIGRID Section

It contains options for setting up multigrid acceleration:

- START – grid level to start the simulation; 1 denotes the finest grid
- CYCLE – type of multigrid cycle: 0 = no multigrid, 1 = V-cycle, 2 = W-cycle
- SMOOCF – coefficient for smoothing the coarse-grid corrections (<0 - no smoothing)
- REFINE – number of iterations before switching to the next finer grid level (Full Multigrid or successive grid refinement)

4.11 NUMERICS Section

It contains options for specifying numerical parameters separately for each block:

- REGION -- range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- CFL – CFL number
- SMOOCF – coefficient of implicit residual smoothing (steady flow, or dual-time stepping only). No smoothing if the value is zero or negative
- DISCR – type of space discretization: 0 = central scheme, 1 = Roe's upwind scheme
- K2 – artificial dissipation coefficient $k^{(2)}$; central scheme only
- 1/K4 – artificial dissipation coefficient $1/k^{(4)}$; central scheme only
- PSWTYPE – type of the pressure switch: 0 = standard, 1 = TVD type; central scheme only
- PSWOMEGA – blending coefficient for PSWTYPE = 1; central scheme only
- ORDER – spatial accuracy (only 2nd-order with the central scheme)
- LIMFAC – limiter coefficient for the upwind scheme; requires correct settings of the reference values
- ENTROPY – entropy correction coefficient, Roe's upwind scheme

In the case of a viscous flow and a highly stretched grid, it is recommended to employ Roe's upwind scheme (DISCR = 1).

4.12 FORCES Section

It contains the option:

- TYPE – type of forces to be computed: 0 = none, 1 = pressure, 2 = pressure and viscous

4.13 PROBE Section

It contains options for setting up probe(s) inside the flow field:

- NUMBER – number of probes; zero means there are none
- block ic jc kc – block number and cell indices in the i-, j-, and k-direction for each probe; the first physical cell has the number 1
- WRITIME – time offset (seconds) to store probe data

- WRIITER – offset between iterations to store probe data
- OPENCLOSE – open and close probe file every time data is written to disk (0 = no, 1 = yes)

4.14 THRUST Section

It contains options for setting up a thrust plane and for computing the value(s):

- TYPE – type of thrust: 0 = none, 1 = momentum thrust only, 2 = momentum and pressure thrust
- PLANE – thrust plane: x = const. (1), y = const. (2), z = const. (3)
- COORD – coordinate of the plane
- PAMB – ambient pressure given in Pascal (only if TYPE = 2)
- WRITIME – time offset (seconds) to store thrust history
- WRIITER – offset between iterations to store thrust history
- OPENCLOSE – open and close thrust file every time data is written to disk (0 = no, 1 = yes)

5. Boundary Conditions File

The file with boundary condition values has the extension `.rbc`. Likewise the user input, the file consists of various sections, each specifying a different boundary condition. The following section keywords are recognized:

- BC_SLIPW – slip (Euler) wall (5.1)
- BC_NOSLIP – noslip (viscous) wall (5.2)
- BC_INFLOW – inflow (5.3)
- BC_OUTFLOW – outflow (5.4)
- BC_FARF – far-field (5.5)
- BC_INJECT – injection (5.6)

The overall formatting of the file follows what was said above for the user input file. An example with all available options can be found in the file `bcond_example.rbc`. The file has to include all the relevant boundary conditions for the flow case in question. Otherwise, the execution of ***Snas3D*** will be stopped with an error message.

5.1 BC_SLIPW Section

It contains options for setting up slip walls:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- PATCH – range of boundary patches within the above range of blocks to which the following options apply; if both values are zero, the options apply to all patches within the given range of blocks
- EXTERNAL – coupling to an external program: 0 = no, 1 = yes
- EXTRAPOL – order of extrapolation to the dummy cells (0 or 1)
- MAXCHANGE – maximum relative change of ρ or ρE before the extrapolation to the dummy cells is switched from the standard 1st- to 0-th order (values between 0.1 and 1.0)
- COMPFORCE – computation of forces: 0 = no, 1 = yes

5.2 BC_NOSLIP Section

It contains options for setting up noslip walls:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- PATCH – range of boundary patches within the above range of blocks to which the following options apply; if both values are zero, the options apply to all patches within the given range of blocks
- EXTERNAL – coupling to an external program: 0 = no, 1 = yes
- ADIABAT – wall boundary condition: 0 = wall temperature is given, 1 = wall is adiabatic
- DISTRIB – single value (0) for all faces of the patch or distribution (1) to be read in from a file
- FILE – path and name of the file to read the distribution from (if DISTRIB = 1)
- TWALL – wall temperature (if ADIABAT = 0)
- COMPFORCE – computation of forces: 0 = no, 1 = yes

5.3 BC_INFLOW Section

It contains options for setting up inflow boundaries:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- PATCH – range of boundary patches within the above range of blocks to which the following options apply; if both values are zero, the options apply to all patches within the given range of blocks

- EXTERNAL – coupling to an external program: 0 = no, 1 = yes
- TYPE – type of inflow: 0 = supersonic, 1 = subsonic, 2 = mixed
- DISTRIB – single value (0) for all faces of the patch or distribution (1) to be read in from a file
- FILE – path and name of the file to read the distribution from (if DISTRIB = 1)
- BETAH – flow angle about the horizontal axis [deg]
- BETAV – flow angle about the vertical axis [deg]
- TURBSA1 – ratio of turbulent to laminar viscosity for the S-A turbulence model

If the inflow is subsonic or mixed (TYPE > 0)

- PTOT – total pressure [Pa]
- TTOT – total temperature [K]

If the inflow is supersonic or mixed (TYPE = 0 or TYPE = 2)

- DENS – static density [kg/m^3]
- ABSVEL – velocity magnitude [m/s]
- PRESS – static pressure [Pa]

5.4 BC_OUTFLOW Section

It contains options for setting up outflow boundaries:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- PATCH – range of boundary patches within the above range of blocks to which the following options apply; if both values are zero, the options apply to all patches within the given range of blocks
- EXTERNAL – coupling to an external program: 0 = no, 1 = yes
- TYPE – type of outflow: 0 = supersonic, 1 = subsonic, 2 = mixed
- DISTRIB – single value (0) for all faces of the patch or distribution (1) to be read in from a file
- FILE – path and name of the file to read the distribution from (if DISTRIB = 1)
- PRESS – static pressure [Pa] (only if TYPE = 1 or 2)

5.5 BC_FARF Section

It contains options for setting up far-field boundaries:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- PATCH – range of boundary patches within the above range of blocks to which the following options apply; if both values are zero, the options apply to all patches within the given range of blocks
- EXTERNAL – coupling to an external program: 0 = no, 1 = yes
- DISTRIB – single value (0) for all faces of the patch or distribution (1) to be read in from a file
- FILE – path and name of the file to read the distribution from (if DISTRIB = 1)
- MACH – Mach number
- ATTACK – angle of attack [deg]
- SLIP – angle of side slip [deg]
- PRESS – static pressure [Pa]
- TEMP – static temperature [K]

5.6 BC_INJECT Section

It contains options for setting up injection boundaries:

- REGION – range of blocks to which the following options apply. If both values are zero, the options apply to all blocks
- PATCH – range of boundary patches within the above range of blocks to which the following options apply; if both values are zero, the options apply to all patches within the given range of blocks
- EXTERNAL – coupling to an external program: 0 = no, 1 = yes
- EXTRAPOL – order of extrapolation to the dummy cells (0 or 1)
- MAXCHANGE – maximum relative change of temperature before the extrapolation to the dummy cells is switched from the standard 1st- to 0-th order (values between 0.1 and 1.0)
- DISTRIB – single value (0) for all faces of the patch or distribution (1) to be read in from a file
- FILE – path and name of the file to read the distribution from (if DISTRIB = 1)
- MFRATE – mass flow rate [kg/(m^2s)]
- TEMP – injection temperature [K]

6. Initial Solution

In order to start the solution process, ***Snas3D*** expects a file with either the initial flow data, or a solution from some previous iteration or time step. The utility program `snprep` reads parts of the user input file (sections INITFLOW, FORMATS, FLOWMODEL and REFERENCE – see Chapter 4 for details) and the grid topology. It then generates the initial flow solution and stores the data in a solution file. The pre-processor is executed by typing the name (and path if necessary) of the executable, followed by the name of the flow case and the grid level (1 is the finest grid). So, for example:

```
> path_to_snas3d/bin/snprep casename 1
```

It is assumed that all data is present in the directory, from which the pre-processor was started. It is further assumed that the base name of all files is `casename`.

Snas3D uses its own format to store flow data, which can be either ASCII or binary. Single solution file contains the data of all blocks. Solution files have the extension `.rsoa` for the ASCII format and `.rsob` for the binary format plus an additional stamp. The stamp can take on two different forms. In the case of a steady flow, it is a six digit integer (with all leading zeros -- format I6.6) which represents the iteration number. Hence,

```
casename.rsoa_000123
```

represents the solution at the 123rd iteration. In the case of an unsteady flow, stamp denotes the physical time (in seconds) of the solution. It has the format 1PE11.5. Thus,

```
casename.rsob_1.23000E-01
```

represents the solution at the time t = 0.123 seconds.

Solution files are composed of a header which consists of at least two floating point values – the solution time and the density residual. If turbulence is switched on (see Section 4.6), residual of the turbulence equation is added. For each block, the data starts with five (six if turbulence is on) integer values: the block number, numbers of cells in i-, j-, k-direction, and the number of dummy cells. Optionally, the number of turbulence equations is added. Next follow values of the conservative variables (Equation 1), first the density ρ for all cells (including dummy), then $\rho u, \rho v, \rho w$ and finally ρE. If turbulence is enabled, values of the turbulence variable(s) are stored next. In the case of a moving grid, x-, y-, and z-components of the grid velocities at the faces of the control volumes are stored at the end of the solution file. Note that binary files are in Fortran-specific formatting.

Using the post-processing tool `snpost`, grid and solution data can be transferred into Tecplot's or Vis3D's ASCII format for visualization. See Chapter 8 for details.

7. Running Snas3D

The flow solver is executed by typing the name (and path if necessary) of the executable (i.e., `snsolv` for the serial version, `snsolvpara` for the MPI version), followed by the name of the flow case and the verbosity level. So, for example:

```
> path_to_snas3d/bin/snsolv casename 2
```

It is assumed that all data required to run the solver (user input, grid, solution) is present in the directory, from which the solver is started. It is further assumed that the base name of all files is `casename`. The user can choose between the following verbosity levels:

- 0 – no output to console (*stdout*)
- 1 – moderate amount of output
- 2 – very verbose (required for checking the correctness of input values)

Errors are always written to *stderr*. In order to halt the flow solver during a run, create file named `STOP.txt` in the directory from which the solver was started. The file can be empty. If present, `STOP.txt` gets removed automatically when the flow solver is started again.

During its execution, the flow solver generates a file with the convergence history and optionally files with probe and thrust data (cf. Sections 4.13 and 4.14). These files can be directly visualized using either Tecplot or Vis2D. Grid and flow solutions can be post-processed by the `snpost` tool, which generates plot files in Tecplot's or Vis3D's ASCII format. This is described in the next Chapter.

8. Post-Processing

The post-processing utility `snpost` reads in parts of the user input file, the grid and solution files, plus optionally a control file. It then outputs plot data either in Tecplot's or in Vis3D's ASCII format for visualization. The utility is invoked in the following way:

```
> path_to_snas3d/bin/snpost <casename> <type> <format> <level> <time/iter> [file]
```

The particular arguments (enclosed in <>) have the following meaning:

- casename – name of the flow case (common to all files)
- type – type of the plot: 0 = grid only, 1 = grid and solution together
- format – format of the plot file: 1=Vis3D, 2=Tecplot
- level – grid level (> 0, 1 represents the finest grid)
- time/iter – physical time in seconds or iteration number of the solution file
- file – control file with further plot options (optional)

It is assumed that all data (user input, grid, solution) is present in the current directory. Example of a control file is provided in `post_example.rpo`.

9. Grid Conversion and Splitting

The utility program `snsplit` allows the user to split single-block grids into multiple blocks and to convert grid files between ASCII and binary formats. The utility is invoked in the following way:

```
> path_to_snas3d/bin/snsplit <casename> <parts> <format-in> <format-out>
```

The particular arguments (enclosed in <>) have the following meaning:

- casename – name of the flow case (common to all files)
- parts – new number of regions (0 = just convert)
- format-in – input grid in ASCII format (1) or in binary format (2)
- format-out – output grid in ASCII format (1) or in binary format (2)

It is assumed that the grid data is present in the current directory.

Bibliography

[1] P. Thomas and C. Lombard, "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal,* vol. 17, pp. 1030-1037, 1979.

[2] A. Jameson, W. Schmidt and E. Turkel, "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," in *AIAA Paper 81-1259*, 1981.

[3] P. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics,* vol. 43, pp. 357-372, 1981.

[4] A. Jameson, "Time Dependent Calculations Using Multigrid with Applications to Unsteady Flows Past Airfoils and Wings," in *AIAA Paper 91-1596*, 1991.

[5] P. Spalart and S. Allmaras, "A One-Equation Turbulence Model for Aerodynamic Flows," in *AIAA Paper 92-0439*, 1992.

[6] S. Allmaras, F. Johnson and P. Spalart, "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," in *ICCFD7-1902, 7th Int. Conf. on Computational Fluid Dynamics*, 2012.

[7] J. Weiss and W. Smith, "Preconditioning Applied to Variable and Constant Density Flows," *AIAA Journal,* vol. 33, pp. 2050-2057, 1995.

[8] J. Blazek, Computational Fluid Dynamics: Principles and Applications, Elsevier Science, 3rd extended edition, 2015.