

Open-Source CFD simulation package

«OpenHyperFLOW2D»

(User guide)

Version 1.03

(<http://github.com/sergeas67/openhyperflow2d>)

General description of CFD simulation package

OpenHyperFLOW2D it is open-source research-educational code for CFD simulation 2D (flat/axisymmetric) transient, viscous, compressible, multi-components sub/trans/supersonic reacting gas flows.

Solver and pre-processor written in C++ and have the following features:

- Using finite difference method (FDM);
- Use a regular orthogonal (rectangular) Cartesian grid;
- For solution used time-depended method with explicit blended marching finite-difference scheme, second-order central differences in the spatial and first order in time Explicit Euler Method (Forward Euler)
- To capture discontinuities use locally adapted non-linear blending factor function (BFF). The solution is found as linear combination of solutions obtained in the scheme with central differences (high-order scheme with low numerical dissipation) and the Lax-Friedrichs (low-order scheme with a high numerical dissipation)

$$\mathbf{F}_{\text{sol}} = \mathbf{BFF} * \mathbf{F}_{\text{cds}} + (1 - \mathbf{BFF}) * \mathbf{F}_{\text{LxF}}$$

where,

\mathbf{F}_{sol} — general solution obtained as a linear combination of solutions to the scheme with low numerical dissipation and solutions to the scheme with high numerical dissipation.

\mathbf{F}_{cds} — the solution obtained on central differences scheme (CDS).

\mathbf{F}_{LxF} — solution obtained by Lax-Friedrichs scheme (LxF).

\mathbf{BFF} — non-linear blending factor function, depending on local parameters.

- Non-orthogonality of boundaries taken into account through the slope-matrix at the boundary nodes of the grid (Immersed Boundary Method analogue);
- The ability to set the boundary conditions I, II and III-type for any of the dependent variables, assignments mixed boundary conditions;
- Temperature dependence of the physical properties of the gas mixture components;
- Can use various RANS/URANS turbulence models
 - Zero-equation models:
 - *Prandtl*;
 - *van Driest*;
 - *Escudier*;
 - *Klebanoff*;
 - *Smagorinsky-Lilly*;
 - One-equation models:
 - *Spalart-Allmaras* model;
 - Two-equations models:
 - Standard (*Spalding*) k - ϵ model;
 - *Chien* k - ϵ model;
 - *Jones-Launder* k - ϵ model;
 - *Launder and Sharma*, with *Yap*-correction k - ϵ model;
 - RNG k - ϵ model;
- Ability to calculate multi-components reacting flows;
- Parallel versions of solvers with automatic spatial domain decomposition with support OpenMP for shared memory systems and with support of MPI (Message Passing Interface) for HPC-clusters (Intel MPI, MVAPICH, OpenMPI, HP MPI). Also available parallel version of solver with support GPU (NVidia *CUDA*);
- Saving the results of calculation in ASCII format of [Tecplot](#) post-processor;
- Saving solver state on checkpoints with the possibility the restart of calculating;

Representation of the input data

To store input data for pre-processors and solvers use an **object** representation. Object is identified by its name and type. In the current version of the program (1.03) uses 3 types of storage objects:

1. Storage
2. A single data object
3. Table

Storage - **container** that can contain objects of type data and table

One input file contains a single storage. The syntax of this object in the input data file:

<start/[storage name]>

...

<end/[storage name]>

A single data object — contains a single text (string), integer (int), or real (float) value.

The syntax of this object in the source data file:

<data/[name]=[value]>

Example:

<data/A=1> - object data named «A» have **integer** value 1

<data/B=2.56> - object data named «B» have **real** value 2.56

<data/C=Test> - object data named «C» have **text** value «Test»

These all objects of type data is stored in plain text and if possible, can be converted by reading if such a transformation is correct.

For example <data/A=1> can be read as a text "1" as the integer 1, and as a real 1.0, <data/B=2.56>, as the text "2.56" and as real 2.56, while trying to read the value of this object as an integer will get a message about an incorrect data type.

Table - contains several coupled pairs of real numbers. Used to set the tabulated functions with one argument.

The syntax of this object in the input data file:

```
<table=[name]/[number coupled pair values (int)]>
```

```
[function argument(float)] [function value(float)]
```

...

```
<endtable>
```

Example:

```
<table=D/5>
```

```
0.0    2.0
```

```
1.0    4.0
```

```
2.0    5.0
```

```
3.0    5.5
```

```
4.0    6.0
```

```
<endtable>
```

The values in the pair may be separated by spaces or tabs, the number of pairs must match the number in the table header. The first column of the table corresponds to the function's arguments, the second, value of the function: $D(0.) = 2$. The argument can be any value, while if the argument falls within the range between the two adjacent values, the result is a linear approximation of the value of the function, such as $D(0.5) = 3$. If the argument is outside the upper and lower bounds of arguments defined in the table, value of the function takes an extreme value in the table, such as $D(-1.) = 2.0$, $D(100) = 6.0$

Any entries in the initial data do not correspond to the syntax will be ignored, lines that contain the symbols of comments “;” and “#” are also ignored from the comment symbol position to the end of the line.

Description of input data for «OpenHyperFLOW2D» solver

OpenHyperFLOW2D solver uses a single input file with one repository (except airfoil cases). In the [Table. 1](#) lists the required that must contain raw data file (for version 1.03 solver, other versions of the solver can have a different set of required objects.) If the source data file is not at least one of the required parameters, or its value is not a valid type/syntax, solver initialization will be interrupted with a corresponding error message. Parameters, highlighted by **gray** color is experimental and for this need additional tests.

OpenHyperFLOW2D solver has 4 versions:

1. Serial solver: used on systems with one processor. Currently, this version is used only for debugging.
2. Parallel solver for systems with shared memory: used on multi-core / multi-processor systems with support OpenMP.
3. Parallel solver for systems with distributed memory: used on computer clusters and multi-core/multi-processor workstation with support MPI libraries and is now the main version.
4. Parallel solver for systems with GPUs: used on computers with NVidia GPU (like NVidia K80,K40,K20,Quadro,GTX Titan, etc) with *CUDA* support (multi-GPU also). This version of OpenHyperFLOW2D solver is experimental.

Input data file compatible with all versions of the solvers. Specific parameters for different version of solver just ignored in others. All versions of the solver is running on GNU/Linux x86_64.

For viewing and post-processing the results of the calculation in the form of 2D distributions of the parameters it is recommended to use a professional post-processor [Tecplot](#), which is considered the “de facto” standard in the CFD.

Table 1

No	Object type	Object name	Value type	Description
1	storage	<arbitrary>	string	It identifies the <u>name of the data storage</u>
2	data	ProjectName	string	It identifies the <u>name of the project</u> . All files created when using the solver will include this name
3	data	OutputFile	string	Contains output file extension (usually '.plt'). This name is combined with the name of the project to specify the file name of results
4	data	ErrorFile	string	suffix + results file extension for error diagnostics (usually '-err.plt')
5	data	GasSwapFile	string	Extension of swap file, with contents computational area (mesh) for gas (usually .hf2d)
6	data	is_p_asterisk_out	int	Add in to the output file total pressure value instead mu_t/mu value 0 – No 1 – Yes
7	data	isSingleGPU	int	Use only single GPU in multi-GPU systems (only for CUDA version) 0 – No 1 - Yes
8	data	ActiveSingleGPU	int	Index of active GPU in single GPU mode (only for CUDA version)
9	data	isAdiabaticWall	int	The model used to calculate the heat transfer on wall: 0 - isothermal 1 - adiabatic (thermally insulated)
10	data	FlowType	int	Type of problem (2D formulation) 0 - flat 1 - axisymmetric
11	data	ProblemType	int	Type of problem (model) 0 – Euler (invisc.) 1 – Navier-Stokes (visc.)
12	data	MaxX	int	The dimension of the grid along the X axis
13	data	MaxY	int	The dimension of the grid along the Y axis
14	data	dx	float	The size of the grid cell along the axis X (m)
15	data	dy	float	The size of the grid cell along the axis Y (m)

Table 1

No	Object type	Object name	Value type	Description
16	data	MonitorIndex	int	Index of monitor, which used for check task convergence: 0 - max residual 1 - ρ residual 2 - ρU residual 3 - ρV residual 4 - ρE residual 5 - Time
17	data	ExitMonitorValue	float	Calculation is stop If exceed value of monitor
18	data	isAlternateRMS	int	Use an alternate algorithm for computing RMS residuals (L2-norm)
19	data	CFL	float	The maximum of Courant-Friedrichs-Levy number (CFL) usually not more than 0.1-0.15
20	table	CFL_Scenario		The scenario changes of CFL number, depending on the of iteration
21	data	beta	float	The base value of the blending factor (proportion by weight of the solution obtained in the CD scheme in the general solution in the steady state) is generally 0.985..0.99. When instability take place, this value can be reduced.
22	table	beta_Scenario		The scenario changes base blending factor (BBF), depending on the of iteration
23	data	beta_NonReflectedBC	float	The base value of the blending factor in nodes with non-reflected BC (is generally <0.5)
24	data	Nmax	int	The number of iterations after which the intermediate results is saved in the file containing the binary image of the computational domain
25	data	NOutStep	int	The number of iterations after which occurs current iteration number, the RMS residuals, current calculation speed (iterations / sec) and the current time step
26	data	NSaveStep	int	NoutStep*Nmax - The number of iterations after which the intermediate results is saved in the results file in Tecplot format (ASCII version)

Table 1

No	Object type	Object name	Value type	Description
27	data	isVerboseOutput	int	1 – output data described in pp. 15 Tab.1 0 - no data output
28	data	BFF	int	Blending Factor Function (BFF). Type of function, which is calculated by a local blending factor, in this version of the program available indexes from 0 to 5 (recommended values: 4 for transient problems, 5 for steady state problems).
29	data	TurbulenceModel	int	The type of turbulence model ¹
30	data	isTurbulenceReset	int	1 - reinitialize turbulence model (if there was a new set) 0 - use a turbulence model, stored in the binary image of the computational domain
31	data	SigW	float	Factor for adjustment total (molecular+turbulent) viscosity in the parietal cells $\mu_{\text{new}} = \text{SigW} * \mu_{\text{old}}$
32	data	SigF	float	Factor for adjustment total (molecular+turbulent) viscosity in the core stream $\mu_{\text{new}} = \text{SigF} * \mu_{\text{old}}$
33	data	delta_bl	float	The estimated thickness of the boundary layer (m), is used in some models of turbulence
34	data	TurbStartIter	int	Iteration number from which to solve the equation(s) selected turbulence model
35	data	TurbExtModel	int	Extended index of turbulence model ¹
36	data	NumMonitorPoints	int	Number of monitoring points
37	data	Point-[n].X	float	X coordinate of monitoring poin № [n] (m)
38	data	Point-[n].Y	float	Y coordinate of monitoring poin № [n] (m)

1 0 — turbulence model is not used (laminar)

1 — The integrated model (used to calculate the heat transfer with using criterial equations)

2 — [algebraic](#) RANS models (Zero equation models)

3 — RANS models with one equation ([Spalart-Allmaras](#) model, [Sekundov Nut-92](#) model)

4 — RANS models with two equations ([k-ε](#), [k-ω](#))

Table 1

No	Object type	Object name	Value type	Description
39	data	InitTime	float	Initial time value (sec)
40	data	Ts0	float	The wall temperature, K
41	data	K0	float	Stoichiometric ratio (used in the combustion problems)
42	data	gamma	float	Factor "completion" of a chemical reaction: 1.0 - no combustion 0.0 - complete combustion (used in the combustion problems)
43	data	Tf	float	Ignition temperature, K (used in the combustion problems)
44	data	NumSrc	int	The number of gas sources ("Src" objects)
45	data	Src ^[n] .GasSrcSX	int	^[n] — index of the gas source. The X coordinate (index of X coordinate in nodes) <u>starting point</u> of the segment along which introduces additional source terms
46	data	Src ^[n] .GasSrcSY	int	^[n] — index of the gas source. The Y coordinate (index of Y coordinate in nodes) <u>starting point</u> of the segment along which introduces additional source terms
47	data	Src ^[n] .GasSrcEX	int	^[n] — index of the gas source. The X coordinate (index of X coordinate in nodes) <u>end point</u> of the segment along which introduces additional source terms
48	data	Src ^[n] .GasSrcEY	int	^[n] — index of the gas source. The Y coordinate (index of Y coordinate in nodes) <u>end point</u> of the segment along which introduces additional source terms

2 ^[n] — index of the object "Src", 1..n
3 ^[n] — index of the object "Src", 1..n
4 ^[n] — index of the object "Src", 1..n
5 ^[n] — index of the object "Src", 1..n

Table 1

No	Object type	Object name	Value type	Description
49	data	Src[n].GasSrcIndex	int	[n] — index of the gas source. GasSrcIndex — index component received through this source 0 - "fuel" 1- "oxidizer" 2 – "combustion products" 3 - "inert ingredient" 4 - mixture of 4 above mentioned components in predetermined proportions
50	data	Src[n].Msrc	float	Mass flow component source in Src[n], kg/sec
51	data	Src[n].Tsrc	float	The temperature of the feed component source in Src[n], K
52	data	Src[n].Tf_src	float	Ignition temperature in source Src[n], K
53	data	Src[n].Y_cp	float	The relative concentration of combustion products in source Src[n] (feeding a mixture of components).
54	data	Src[n].Y_air	float	The relative concentration of the inert component (e.g. air) in source Src[n] (feeding a mixture of components).
55	data	Src[n].Y_fuel	float	The relative concentration of fuel in source Src[n] (feeding a mixture of components).
56	data	Src[n].Y_ox	float	The relative concentration of the oxidizer in source Src[n] (feeding a mixture of components).
57	data	NumFlow	int	The number of objects "Flow" (deprecated)
58	data	Flow[n]⁶.CompIndex	int	CompIndex — index of the integral component: 0 - "fuel" 1- "oxidizer" 2 – "combustion products" 3 - "inert ingredient"
59	data	Flow[n]⁷.p	float	p — static pressure (Pa) associated with the object "Flow"

6 [n] — index of the object "Flow", 1..n

7 [n] — index of the object "Flow", 1..n

Table 1

No	Object type	Object name	Value type	Description
60	data	Flow[n]⁸.Type	int	Type - How to set the speed of the object "Flow": 0 - absolute value of speed 1 - relative critical velocity - λ
61	data	Flow[n]⁹.W	float	W – absolute value of velocity, m/sec ⁱⁱ
62	data	Flow[n]¹⁰.Lam	float	Lam – value of relative critical velocity
63	data	Flow[n]¹¹.T	float	T – static temperature value, K
64	data	NumFlow2D	int	The number of objects "Flow2D"
65	data	Flow2D-[n]¹².CompIndex	int	CompIndex — index of the integral component: 0 - "fuel" 1- "oxidizer" 2 – "combustion products" 3 - "inert ingredient"
66	data	Flow2D-[n]¹³.Mode	int	Mode of definition object "Flow2D" 0 - Set static values p,T and velocity components magnitude (U,V) 1 – Set total values p*,T* and velocity components magnitude (U,V) 2 -Set Mach number, angle between the flow direction and the X-axis and static values p,T 3 – Set Mach number, angle between the flow direction and the X-axis and total values p*,T*
67	data	Flow2D-[n]¹⁴.p	float	p — static/total pressure (Pa) associated with the object "Flow2D"
68	data	Flow2D-[n]¹⁵.T	float	T – static/total temperature value, (K) for Flow2D
69	data	Flow2D-[n]¹⁶.U	float	The component of the velocity along the X-axis (for flat flow) or The axial component of the velocity (axi-symmetric flow), m/s

- 8 [n] — index of the object "Flow", 1..n
9 [n] — index of the object "Flow", 1..n
10 [n] — index of the object "Flow", 1..n
11 [n] — index of the object "Flow", 1..n
12 [n] — index of the object "Flow2D", 1..n
13 [n] — index of the object "Flow2D", 1..n
14 [n] — index of the object "Flow2D", 1..n
15 [n] — index of the object "Flow2D", 1..n

Table 1

No	Object type	Object name	Value type	Description
70	data	Flow2D-[n]¹⁷.V	float	The component of the velocity along the Y-axis (for flat flow) or The radial velocity component (for axi-symmetric flow), m/s
71	data	Flow2D-[n]¹⁸.Mach	float	Mach number
72	data	Flow2D-[n]¹⁹.Angle	float	The angle between the flow direction and the X-axis
73	data	NumRects	int	The number of macro-objects of solid type "Rect" (rectangle)
74	data	Rect[n]²⁰.Xstart	float	Coordinate X (m) of the lower left corner of the object "Rect"
75	data	Rect[n]²¹.Ystart	float	Coordinate Y (m) of the lower left corner of the object "Rect"
76	data	Rect[n]²².DX	float	Vertical (along coordinate Y) size of the object "Rect" (m)
77	data	Rect[n]²³.DY	float	Horizontal (along the coordinate X) size of the object "Rect" (m)
78	data	Rect[n]²⁴.Flow2D	int	The index of an object "Flow2D" for parameters initialized boundary of object "Rect"
79	data	Rect[n]²⁵.TurbulenceModel	int	The type of turbulence models ²⁶ on the border of the object "Rect"
80	data	NumCircles	int	The number of macro-objects of type "Circle"
81	data	Circle[n]²⁷.X0	float	Coordinate X (m) the center of solid macro-object type "Circle"
82	data	Circle[n]²⁸.Y0	float	Coordinate Y (m) the center of solid macro-object type "Circle"

16 [n] — index of the object "Flow2D", 1..n

17 [n] — index of the object "Flow2D", 1..n

18 [n] — index of the object "Flow2D", 1..n

19 [n] — index of the object "Flow2D", 1..n

20 [n] — index of the object "Rect", 1..n

21 [n] — index of the object "Rect", 1..n

22 [n] — index of the object "Rect", 1..n

23 [n] — index of the object "Rect", 1..n

24 [n] — index of the object "Rect", 1..n

25 [n] — index of the object "Rect", 1..n

26 0 — turbulence model is not used (laminar)

1 — The integrated model (used to calculate the heat transfer with using criterial equations)

2 — [algebraic](#) RANS models (Zero equation models)

3 — RANS models with one equation ([Spalart-Allmaras](#) model, [Sekundov Nut-92](#) model)

4 — RANS models with two equations ([k-ε](#), [k-ω](#))

27 [n] — index of the object "Circle", 1..n

Table 1

No	Object type	Object name	Value type	Description
83	data	Circle[n]²⁹.Xstart	float	Coordinate X (m) starting point solid macro-object type "Circle" ³⁰
84	data	Circle[n]³¹.Ystart	float	Coordinate Y (m) starting point solid macro-object type "Circle"
85	data	Circle[n]³².TurbulenceModel	int	The type of turbulence model ³³ on the border of the object "Circle"
86	data	Circle[n]³⁴.MaterialID	int	Index of material, which fill internal area of object "Circle" 0 – Gas 1 – Solid
87	data	Circle[n]³⁵.Flow2D	int	The index of an object "Flow2D" for parameters initialized boundary of object "Circle"
88	data	NumAirfoils	int	The number of macro-objects of solid type "Airfoil"
89	data	Airfoil[n]³⁶.Type		Airfoil geometry setting method 0 – Embedded NACA XXYY airfoil 1 – Setting from external file
90	data	Airfoil[n]³⁷.Xstart	float	Coordinate X (m) starting point solid macro-object type "Airfoil"
91	data	Airfoil[n]³⁸.Ystart	float	Coordinate Y (m) starting point solid macro-object type "Airfoil"
92	data	Airfoil[n]³⁹.pp	float	«X» (only for Type=0)
93	data	Airfoil[n]⁴⁰.mm	float	«Y» (only for Type=0)

28 [n] — index of the object "Circle", 1..n

29 [n] — index of the object "Circle", 1..n

30 An object of type "Circle" is a circle with the boundary of the "no-slip wall" and filled the interior of the cells of the "solid". Circle center has the coordinates X0, Y0. The radius of the circle $R = \sqrt{(X0 - Xstart)^2 + (Y0 - Ystart)^2}$. Starting point from which a circle is drawn has the coordinates Xstart, Ystart

31 [n] — index of the object Circle, 1..n

32 [n] — index of the object Circle, 1..n

33 0 — turbulence model is not used (laminar)

1 — The integrated model (used to calculate the heat transfer with using criterial equations)

2 — [algebraic](#) RANS models (Zero equation models)

3 — RANS models with one equation ([Spalart-Allmaras](#) model, [Sekundov Nut-92](#) model)

4 — RANS models with two equations ([k-ε](#), [k-ω](#))

34 [n] — index of the object Circle, 1..n

35 [n] — index of the object Circle, 1..n

36 [n] — index of the object Airfoil, 1..n

37 [n] — index of the object Airfoil, 1..n

38 [n] — index of the object Airfoil, 1..n

39 [n] — index of the object Airfoil, 1..n

Table 1

№	Object type	Object name	Value type	Description
94	data	Airfoil [n] ⁴¹ .thick	float	Airfoil thickness (%) «ZZ» (only for Type=0)
95	data	Airfoil [n] ⁴² .InputData	string	External file name with airfoil data*
96	data	Airfoil [n] ⁴³ .scale	float	Airfoil scale (corresponding to the length of the chord, m)
97	data	Airfoil [n] ⁴⁴ .attack_angle	float	The angle of attack of macro-object “Airfoil”
98	data	Airfoil [n] ⁴⁵ .Flow2D	int	The index of an object “Flow2D” for parameters initialized boundary of object “Airfoil”
99	data	Airfoil [n] ⁴⁶ .TurbulenceModel	int	The type of turbulence model ⁴⁷ on the border of the object “Airfoil”
100	data	isOutHeatFluxX	int	1 - output file HeatFlux-X-<project name>.plt maximum heat flux (W/m ²) on the walls along the coordinates X (m) 0 – no output
101	data	isOutHeatFluxY	int	1 - output file HeatFlux-Y-<project name>.plt maximum heat flux (W/m ²) on the walls along the coordinates Y (m) 0 – no output
102	data	y_max	int	Upper (max) limit of zone for output heat flux along X direction (in nodes). Affected only if isOutputHeatFluxX=1
103	data	y_min	int	Lower (min) limit of zone for output heat flux along X direction (in nodes). Affected only if isOutputHeatFluxX=1

40 [n] — index of the object Airfoil,1..n

41 [n] — index of the object Airfoil,1..n

42 [n] — index of the object Airfoil,1..n

43 [n] — index of the object Airfoil,1..n

44 [n] — index of the object Airfoil,1..n

45 [n] — index of the object Airfoil,1..n

46 [n] — index of the object “Airfoil”

47 0 — turbulence model is not used (laminar)

1 — The integrated model (used to calculate the heat transfer with using criterial equations)

2 — [algebraic](#) RANS models (Zero equation models)

3 — RANS models with one equation ([Spalart-Allmaras](#) model, [Sekundov Nut-92](#) model)

4 — RANS models with two equations ([k-ε](#), [k-ω](#))

Table 1

No	Object type	Object name	Value type	Description
104	data	NumSingleBounds	int	The number of objects "single boundary"
105	table	SingleBound[n]ⁱⁱⁱ.Points	-	Points — coordinates of the starting and end coordinates (m) of the segment in the form: $\begin{matrix} X_{\text{start}} & Y_{\text{start}} \\ X_{\text{end}} & Y_{\text{end}} \end{matrix}$
106	data	SingleBound[n]^{iv}.Cond	string	BC ^v defined along the object SingleBound
107	data	SingleBound[n]^{vi}.Flow2D	int	The index of an object "Flow2D" for parameters initialized boundary of object SingleBound
108	data	SingleBound[n]^{vii}.TurbulenceModel	int	The type of turbulence models ⁴⁸ on the border of the object "SingleBound"
109	data	SingleBound[n]^{viii}.MaterialID	int	Bound material ID 0 – Gas 1 – Solid
110	data	SingleBound[n]^{ix}.isReset	int	1 - reinitialize the object "SingleBound[n]" in restarting solver 0 — do not reinitialize the object "SingleBound[n]" in restarting solver
111	data	NumContour	int	The number of objects "Contour" (many boundaries united in a closed loop)
112	table	Contour[n]⁴⁹	-	The coordinates of the boundaries in closed loop
113	data	Contour[n]⁵⁰.Bound[m]⁵¹.Cond	string	BC ^x defined along a segment of the object "Contour[n]" index [m]
114	data	Contour[n]⁵².Bound[m].Flow2D	int	The index of an object "Flow2D", associated with the segment "Bound[m]" of object "Contour[n]"

- 48 0 — turbulence model is not used (laminar)
 1 — The integrated model (used to calculate the heat transfer with using criterial equations)
 2 — [algebraic](#) RANS models (Zero equation models)
 3 — RANS models with one equation ([Spalart-Allmaras](#) model, [Sekundov Nut-92](#) model)
 4 — RANS models with two equations ([k-ε](#), [k-ω](#))
- 49 [n] — index of the object "Contour"
- 50 [n] — index of the object "Contour"
- 51 [m] — index of the object "Bound" in object "Contour"
- 52 [n] — index of the object "Contour"

Table 1

No	Object type	Object name	Value type	Description
115	data	Contour[n]⁵³.Bound[m].TurbulenceModel	int	The type of turbulence models ⁵⁴ along of the segment “Bound[m]” of object “Contour[n]”
116	data	Contour[n]⁵⁵.Bound[m].isReset	int	1 - reinitialize the segment “Bound[m]” of object “Contour[n]” in restarting solver 0 — do not reinitialize the segment “Bound[m]” of object “Contour[n]” in restarting solver
117	data	Contour[n]⁵⁶.Bound[m].MaterialID	int	Bound material ID 0 – Gas 1 – Solid
118	data	NumArea	int	The number of objects “Area”
119	table	Area[n]	-	The coordinates of the beginning of the initialization object “Area” with index [n] (x,y) in nodes !
120	data	Area[n].MaterialID	int	Material ID of solid body nodes (only for Type = 0)
121	data	Area[n].Type	int	Type of initialized nodes 0 — «solid» 1 — «gas»
122	data	Area[n].Flow2D	int	The index of an object “Flow2D” which parameters is initialized object “Area[n]” (only for “gas” nodes)
123	data	Area[n].Turbulence	int	The type of turbulence model ⁵⁷ in object “Area[n]”(only for “gas” nodes)
124	data	H_cp	float	The heat of formation of combustion products (J/kg)

53 [n] — index of the object “Contour”

54 0 — turbulence model is not used (laminar)

1 — The integrated model (used to calculate the heat transfer with using criterial equations)

2 — [algebraic](#) RANS models (Zero equation models)

3 — RANS models with one equation ([Spalart-Allmaras](#) model, [Sekundov Nut-92](#) model)

4 — RANS models with two equations ([k-ε](#), [k-ω](#))

55 [n] — index of the object “Contour”

56 [n] — index of the object “Contour”

57 0 — turbulence model is not used (laminar)

1 — The integrated model (used to calculate the heat transfer with using criterial equations)

2 — [algebraic](#) RANS models (Zero equation models)

3 — RANS models with one equation ([Spalart-Allmaras](#) model, [Sekundov Nut-92](#) model)

4 — RANS models with two equations ([k-ε](#), [k-ω](#))

Table 1

No	Object type	Object name	Value type	Description
125	data	R_cp	float	The gas constant of the combustion products (J/(kg*K))
126	table	lam_cp	-	The thermal conductivity of the combustion products as a function of temperature (J/(kg*K))
127	table	mu_cp	-	The dynamic viscosity of the products of combustion as a function of temperature (Pa*sec)
128	table	Cp_cp	-	The heat capacity of combustion products at constant pressure as a function of temperature (J/(kg*K))
129	data	H_Fuel	float	The heat of formation of fuel (J/kg)
130	data	R_Fuel	float	The gas constant of fuel (J/(kg*K))
131	table	lam_Fuel	-	The thermal conductivity of fuel as a function of temperature (J/(kg*K))
132	table	mu_Fuel	-	The dynamic viscosity of fuel as a function of temperature (Pa*sec)
133	table	Cp_Fuel	-	The heat capacity of fuel at constant pressure as a function of temperature (J/(kg*K))
134	data	H_OX	float	The heat of formation of oxidizer (J/kg)
135	data	R_OX	float	The gas constant of oxidizer (J/(kg*K))
136	table	lam_OX	-	The thermal conductivity of oxidizer as a function of temperature (J/(kg*K))
137	table	mu_OX	-	The dynamic viscosity of oxidizer as a function of temperature (Pa*sec)
138	table	Cp_OX	-	The heat capacity of oxidizer at constant pressure as a function of temperature (J/(kg*K))
139	data	H_air	float	The heat of formation of inert component (e.g. air) (J/kg)
140	data	R_air	float	The gas constant of inert component (e.g. air) (J/(kg*K))
141	table	lam_air	-	The thermal conductivity of inert component (e.g. air) as a function of temperature (J/(kg*K))

Table 1

No	Object type	Object name	Value type	Description
142	table	mu_air	-	The dynamic viscosity of inert component (e.g. air) as a function of temperature (Pa*sec)
143	table	Cp_air	-	The heat capacity of inert component (e.g. air) at constant pressure as a function of temperature (J/(kg*K))
144	data	Cp_Flow_index	int	The index of an object “Flow2D” for calculating pressure coefficient along walls, $C_p =$ (effected only if isOutputHeatFluxX=1)
145	data	is_Cx_calc	int	Calculate the drag coefficient 0 – No 1 – Yes
146	data	x_body	float	Initial X coordinate (m) of the region containing the test body to calculate the drag coefficient (effected only if is_Cx_calc=1)
147	data	y_body	float	Initial Y coordinate (m) of the region containing the test body to calculate the drag coefficient (effected only if is_Cx_calc=1)
148	data	dx_body	float	Size of the region containing the test body in the X coordinate (m) to calculate the drag coefficient (effected only if is_Cx_calc=1)
149	data	dy_body	float	Size of the region containing the test body in the Y coordinate (m) to calculate the drag coefficient (effected only if is_Cx_calc=1)
150	data	Cx_Flow_index	int	The index of an object “Flow2D” for calculating drag coefficient
151	data	is_Cd_calc	int	Calculate the discharge coefficient C_d (for nozzle) 0 – No 1 – Yes
152	data	x_nozzle	float	Initial X coordinate (m) of tested nozzle
153	data	y_nozzle	float	Initial Y coordinate (m) of tested nozzle
154	data	dy_nozzle	float	Cross-section size of tested nozzle (m)
155	data	Cd_Flow_index	int	The index of an object “Flow2D” for calculating discharge coefficient

Table 1

No	Object type	Object name	Value type	Description
156	data	p_ambient	float	Reference ambient pressure for calculating velocity coefficient C_v (for nozzle)
157	data	NumXCut	int	Number of probed cross-sections along the X axis
158	data	CutX-[n].x0	float	Coordinate X (m) of the start cross-section with the index [n]
159	data	CutX-[n].y0	float	Coordinate Y (m) of the start cross-section with the index [n]
160	data	CutX-[n].dy	float	Size of cross-section(m) along the Y axis with the index [n]
161	data	isIgnoreUnsetNodes	int	0 – do not ignore the uninitialized nodes (In the presence of uninitialized nodes task is interrupted) 1 - ignore the uninitialized nodes
162	data	ThreadBlockSize	int	Size of <i>CUDA</i> threads block (only for <i>CUDA</i> version) 0 – calibrate size of threads block for maximum performance (recommended) >0 – number of threads in threads block

0 — algebraic model of Prandtl (the mixing length model)

1 — Van Driest algebraic model

2 — Eskudier algebraic model

3 — Klebanoff algebraic model

4 — Standard $k-\epsilon$ Spalding model

5 — Chien $k-\epsilon$ model

6 — Jones-Launder $k-\epsilon$ model

7 — Launder-Sharma $k-\epsilon$ model with Yapp correction

8 — RNG $k-\epsilon$ model

10 — Spalart-Allmaras model

ii For object “Flow” velocity component along the axis X (or axial) equals the value of W, and a velocity component along the axis Y (or radial) equals 0

iii [n] — The index of an object “SingleBound”

iv [n] — The index of an object “SingleBound”

v BC may have the following values:

Base BC:

CT_NO_COND_2D - BC not defined

CT_Ro_CONST_2D - $\rho = \text{const}$

CT_U_CONST_2D - $\rho U = \text{const}$

CT_V_CONST_2D - $\rho V = \text{const}$

CT_T_CONST_2D - $\rho E = \text{const}$

CT_Y_CONST_2D - $\rho Y = \text{const}$

CT_dRdx_NULL_2D - $d\rho/dx = 0$

CT_dUdx_NULL_2D - $d\rho U/dx = 0$

CT_dVdx_NULL_2D - $d\rho V/dx = 0$

CT_dTdx_NULL_2D - $d\rho E/dx = 0$

CT_dYdx_NULL_2D - $d\rho Y/dx = 0$

CT_dRdy_NULL_2D - $d\rho/dy = 0$

CT_dUdy_NULL_2D - $d\rho U/dy = 0$

CT_dVdy_NULL_2D - $d\rho V/dy = 0$

CT_dTdy_NULL_2D - $d\rho E/dy = 0$

CT_dYdy_NULL_2D - $d\rho Y/dy = 0$

CT_d2Rdx2_NULL_2D - $d^2\rho/dx^2 = 0$

CT_d2Udx2_NULL_2D - $d^2\rho U/dx^2 = 0$

CT_d2Vdx2_NULL_2D - $d^2\rho V/dx^2 = 0$

CT_d2Tdx2_NULL_2D - $d^2\rho E/dx^2 = 0$

CT_d2Ydx2_NULL_2D - $d^2\rho Y/dx^2 = 0$

CT_d2Rdy2_NULL_2D - $d^2\rho/dy^2 = 0$

CT_d2Udy2_NULL_2D - $d^2\rho U/dy^2 = 0$

CT_d2Vdy2_NULL_2D - $d^2\rho V/dy^2 = 0$

CT_d2Tdy2_NULL_2D - $d^2\rho E/dy^2 = 0$

CT_d2Ydy2_NULL_2D - $d^2\rho Y/dy^2 = 0$

CT_WALL_NO_SLIP_2D — BC «no-slip wall»

CT_WALL_LAW_2D — BC «wall law»

Special BC:

CT_NONREFLECTED_2D - non-reflected BC

BC definition as combination of base BC

NT_FC_2D: (Dirichlet BC)

- CT_Ro_CONST_2D
- CT_U_CONST_2D
- CT_V_CONST_2D
- CT_Y_CONST_2D
- CT_T_CONST_2D

NT_D0X_2D: (Neumann BC, gradientless flow in X direction)

- CT_dRox_NULL_2D
- CT_dUdx_NULL_2D
- CT_dVdx_NULL_2D
- CT_dTdx_NULL_2D
- CT_dYdx_NULL_2D

NT_D2X_2D: (Cauchy BC in X direction)

- CT_d2Rox2_NULL_2D
- CT_d2Udx2_NULL_2D
- CT_d2Vdx2_NULL_2D
- CT_d2Tdx2_NULL_2D
- CT_d2Ydx2_NULL_2D

NT_D0Y_2D: (Neumann BC, gradientless flow in Y/R direction)

- CT_dRody_NULL_2D
- CT_dUdy_NULL_2D
- CT_dVdy_NULL_2D
- CT_dTdy_NULL_2D
- CT_dYdy_NULL_2D

NT_D2Y_2D:(Cauchy BC in Y/R direction)

- CT_d2Rody2_NULL_2D
- CT_d2Udy2_NULL_2D
- CT_d2Vdy2_NULL_2D
- CT_d2Tdy2_NULL_2D
- CT_d2Ydy2_NULL_2D

NT_AY_2D: (Symmetry BC along the Y axis)

- CT_NODE_IS_SET_2D
- NT_D0X_2D
- CT_U_CONST_2D

NT_AX_2D: (Symmetry BC along the Y axis)

- CT_NODE_IS_SET_2D
- NT_D0Y_2D
- CT_V_CONST_2D

NT_WALL_LAW_2D: (wall law)

- CT_WALL_LAW_2D

NT_FARFIELD_2D: (non-reflected farfield BC)

- CT_Ro_CONST_2D
- CT_U_CONST_2D
- CT_V_CONST_2D

- CT_Y_CONST_2D
- CT_T_CONST_2D
- CT_NONREFLECTED_2D

NT_WNS_2D: (no-slip wall)

- CT_WALL_NO_SLIP_2D
- CT_U_CONST_2D
- CT_V_CONST_2D

NT_S_2D: (solid body)

- CT_SOLID_2D

- vi [n] — The index of an object “SingleBound”
- vii [n] — The index of an object “SingleBound”
- viii [n] — The index of an object “SingleBound”
- ix [n] — The index of an object “SingleBound”
- x BC may have the following values:

Base BC:

CT_NO_COND_2D - BC not defined
 CT_Ro_CONST_2D - $\rho = \text{const}$
 CT_U_CONST_2D - $\rho U = \text{const}$
 CT_V_CONST_2D - $\rho V = \text{const}$
 CT_T_CONST_2D - $\rho E = \text{const}$
 CT_Y_CONST_2D - $\rho Y = \text{const}$
 CT_dRdx_NULL_2D - $d\rho/dx = 0$
 CT_dUdx_NULL_2D - $d\rho U/dx = 0$
 CT_dVdx_NULL_2D - $d\rho V/dx = 0$
 CT_dTdx_NULL_2D - $d\rho E/dx = 0$
 CT_dYdx_NULL_2D - $d\rho Y/dx = 0$
 CT_dRdy_NULL_2D - $d\rho/dy = 0$
 CT_dUdy_NULL_2D - $d\rho U/dy = 0$
 CT_dVdy_NULL_2D - $d\rho V/dy = 0$
 CT_dTdy_NULL_2D - $d\rho E/dy = 0$
 CT_dYdy_NULL_2D - $d\rho Y/dy = 0$
 CT_d2Rdx2_NULL_2D - $d^2\rho/dx^2 = 0$
 CT_d2Udx2_NULL_2D - $d^2\rho U/dx^2 = 0$
 CT_d2Vdx2_NULL_2D - $d^2\rho V/dx^2 = 0$
 CT_d2Tdx2_NULL_2D - $d^2\rho E/dx^2 = 0$
 CT_d2Ydx2_NULL_2D - $d^2\rho Y/dx^2 = 0$
 CT_d2Rdy2_NULL_2D - $d^2\rho/dy^2 = 0$
 CT_d2Udy2_NULL_2D - $d^2\rho U/dy^2 = 0$
 CT_d2Vdy2_NULL_2D - $d^2\rho V/dy^2 = 0$
 CT_d2Tdy2_NULL_2D - $d^2\rho E/dy^2 = 0$
 CT_d2Ydy2_NULL_2D - $d^2\rho Y/dy^2 = 0$
 CT_WALL_NO_SLIP_2D — BC «no-slip wall»
 CT_WALL_LAW_2D — BC «wall law»

Spesial BC:

CT_NONREFLECTED_2D - non-reflected BC

Macro definition of BC as combination of base BC

NT_FC_2D: (Dirichlet BC)

- CT_Ro_CONST_2D
- CT_U_CONST_2D
- CT_V_CONST_2D
- CT_Y_CONST_2D
- CT_T_CONST_2D

NT_D0X_2D: (Neumann BC, gradientless flow in X direction)

- CT_dRox_NULL_2D
- CT_dUdx_NULL_2D
- CT_dVdx_NULL_2D
- CT_dTdx_NULL_2D
- CT_dYdx_NULL_2D

NT_D2X_2D: (Cauchy BC in X direction)

- CT_d2Rox2_NULL_2D
- CT_d2Udx2_NULL_2D
- CT_d2Vdx2_NULL_2D
- CT_d2Tdx2_NULL_2D
- CT_d2Ydx2_NULL_2D

NT_D0Y_2D: (Neumann BC, gradientless flow in Y/R direction)

- CT_dRody_NULL_2D
- CT_dUdy_NULL_2D
- CT_dVdy_NULL_2D
- CT_dTdy_NULL_2D
- CT_dYdy_NULL_2D

NT_D2Y_2D:(Cauchy BC in Y/R direction)

- CT_d2Rody2_NULL_2D
- CT_d2Udy2_NULL_2D
- CT_d2Vdy2_NULL_2D
- CT_d2Tdy2_NULL_2D
- CT_d2Ydy2_NULL_2D

NT_AY_2D: (Symmetry BC along the Y axis)

- CT_NODE_IS_SET_2D
- NT_D0X_2D
- CT_U_CONST_2D

NT_AX_2D: (Symmetry BC along the Y axis)

- CT_NODE_IS_SET_2D
- NT_D0Y_2D
- CT_V_CONST_2D

NT_WALL_LAW_2D: (wall law)

- CT_WALL_LAW_2D

NT_FARFIELD_2D: (non-reflected farfield BC)

- CT_Ro_CONST_2D
- CT_U_CONST_2D
- CT_V_CONST_2D

- CT_Y_CONST_2D
- CT_T_CONST_2D
- CT_NONREFLECTED_2D

NT_WNS_2D: (no-slip wall)

- CT_WALL_NO_SLIP_2D
- CT_U_CONST_2D
- CT_V_CONST_2D

NT_S_2D: (solid body)

- CT_SOLID_2D