Open-Source CFD simulation package

«OpenHyperFLOW2D»

(User guide)

Version 1.02

(http://openhyperflow2d.googlecode.com)

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)

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

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).

BFF — non-linear blending factor function, depending on local parameters.

- Nonorthogonality 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 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 (1.02) 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]/[coupled pair of values (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 since the comment character to the end of the line.

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 1.02 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 GPU: used on computers with Nvidia GPU (like Nvidia K80,K40,K20,GTX Titan, etc) with CUDA support. 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

№	Object type	Object name	Value type	Description
1	storage	<arbitrary>	string	It identifies the name of the data storage
2	data	**ProjectName**	string	It identifies the name of the project. 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 (first) GPU in multiGPU systems (only CUDA version) 0 – No 1 - Yes
8	data	**isAdiabaticWall**	int	The model used to calculate the heat transfer on wall: 0 - isothermal 1 - adiabatic (thermally insulated)
9	data	**FlowType**	int	Type of problem (2D formulation) 0 - flat 1 - axisymmetric
10	data	**ProblemType**	int	Type of problem (model) 0 – Euler (invisc.) 1 – Navier-Stokes (visc.)
11	data	**MaxX**	int	The dimension of the grid along the X axis
12	data	**MaxY**	int	The dimension of the grid along the Y axis
13	data	**dx**	float	The size of the grid cell along the axis X (m)
14	data	**dy**	float	The size of the grid cell along the axis Y (m)

Table 1

№	Object type	Object name	Value type	Description
15	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
16	data	**ExitMonitorValue**	float	Calculation is stop If exceed value of monitor
17	data	**isAlternateRMS**	int	Use an alternate algorithm for computing RMS residual
18	data	**CFL**	float	The maximum number of Courant-Friedrichs-Levy (CFL) usually no more than 0.1
19	table	**CFL_Scenario**		The scenario changes of CFL number, depending on the number of iteration
20	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.
21	table	**beta_Scenario**		The scenario changes base blending factor (BBF), depending on the number of iteration
22	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
23	data	**NOutStep**	int	The number of iterations after which occurs current iteration number, the RMS residual, current calculation speed (iterations / sec) and the current time step
24	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)
25	data	**isVerboseOutput**	int	1 – output data described in pp. 15 Tab.1 0 - no data output

Table 1

№	Object type	Object name	Value type	Description
26	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 numbers from 0 to 5 (recommended values: 4 for transient problems, 5 for steady state problems).
27	data	**TurbulenceModel**	int	The type of turbulence model[1]
28	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
29	data	**SigW**	float	Factor for adjustment total (molecular+turbulent) viscosity in the parietal cells mu_new = SigW*mu_old
30	data	**SigF**	float	Factor for adjustment total (molecular+turbulent) viscosity in the core stream mu_new = SigF*mu_old
31	data	**delta_bl**	float	The estimated thickness of the boundary layer (m), is used in some models of turbulence
32	data	**TurbStartIter**	int	Iteration number from which to solve the equation(s) selected turbulence model
33	data	**TurbExtModel**	int	Extended number of turbulence model[i]
34	data	**NumMonitorPoints**	int	Number of monitoring points
35	data	**Point-[n].X**	float	X coordinate of monitoring poin № [n] (m)
36	data	**Point-[n].Y**	float	Y coordinate of monitoring poin № [n] (m)
37	data	**InitTime**	float	Ititial time value (sec)

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 1st equation (Spalart-Allmaras model, Sekundov Nut-92 model)
4 — RANS models with 2 equations (k-ε, k-ω)

Table 1

№	Object type	Object name	Value type	Description
38	data	**Ts0**	float	The wall temperature, K
39	data	**K0**	float	Stoichiometric ratio (used in the combustion problems)
40	data	**gamma**	float	Factor "completion" of a chemical reaction: 1.0 - no combustion 0.0 - complete combustion (used in the combustion problems)
41	data	**Tf**	float	Ignition temperature, K (used in the combustion problems)
42	data	**NumSrc**	int	The number of gas sources (“Src” objects)
43	data	**Src[n][2].GasSrcSX**	int	[n] — number (index) of the gas source. The X coordinate (index of X coordinate in nodes) starting point of the segment along which introduces additional source terms
44	data	**Src[n][3].GasSrcSY**	int	[n] — number (index) of the gas source. The Y coordinate (index of Y coordinate in nodes) starting point of the segment along which introduces additional source terms
45	data	**Src[n][4].GasSrcEX**	int	[n] — number (index) of the gas source. The X coordinate (index of X coordinate in nodes) end point of the segment along which introduces additional source terms
46	data	**Src[n][5].GasSrcEY**	int	[n] — number (index) of the gas source. The Y coordinate (index of Y coordinate in nodes) end point of the segment along which introduces additional source terms

2 [n] — number (index) of the object “Src”,1..n
3 [n] — number (index) of the object “Src”,1..n
4 [n] — number (index) of the object “Src”,1..n
5 [n] — number (index) of the object “Src”,1..n

Table 1

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

6 [n] — number (index) of the object “Flow”,1..n
7 [n] — number (index) of the object “Flow”,1..n

Table 1

№	Object type	Object name	Value type	Description
58	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
59	data	**Flow[n][9].W**	float	W – absolute value of velocity, m/sec[ii]
60	data	**Flow[n][10].Lam**	float	Lam – value of relative critical velocity
61	data	**Flow[n][11].T**	float	T – static temperature value, K
62	data	**NumFlow2D**	int	The number of objects "Flow2D"
63	data	**Flow2D-[n][12].CompIndex**	int	CompIndex — index of the integral component: 0 - "fuel" 1- "oxidizer" 2 – "combustion products" 3 - "inert ingredient"
64	data	**Flow2D-[n][13].Mode**	int	Mode of defenition 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 -Reserved 3 – Set Mach number, total values p*,T* and flow direction by velocity components (U,V)
65	data	**Flow2D-[n][14].p**	float	p — static/total pressure (Pa) associated with the object "Flow2D"
66	data	**Flow2D-[n][15].T**	float	T – static/total temperature value, (K) для Flow2D
67	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] — number (index) of the object "Flow",1..n
9 [n] — number (index) of the object "Flow",1..n
10 [n] — number (index) of the object "Flow",1..n
11 [n] — number (index) of the object "Flow",1..n
12 [n] — number (index) of the object "Flow2D",1..n
13 [n] — number (index) of the object "Flow2D",1..n
14 [n] — number (index) of the object "Flow2D",1..n
15 [n] — number (index) of the object "Flow2D",1..n
16 [n] — number (index) of the object "Flow2D",1..n

Table 1

№	Object type	Object name	Value type	Description
68	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
69	data	**NumRects**	int	The number of macro-objects of solid type “Rect” (rectangle)
70	data	**Rect[n][18].Xstart**	float	Coordinate X (m) of the lower left corner of the object “Rect”
71	data	**Rect[n][19].Ystart**	float	Coordinate Y (m) of the lower left corner of the object “Rect”
72	data	**Rect[n][20].DX**	float	Vertical (along coordinate Y) size of the object “Rect” (m)
73	data	**Rect[n][21].DY**	float	Horizontal (along the coordinate X) size of the object “Rect” (m)
74	data	**Rect[n][22].Flow2D**	int	The index (number) of an object “Flow2D” for parameters initialized boundary of object “Rect”
75	data	**Rect[n][23].TurbulenceModel**	int	The type of turbulence models[24] on the border of the object “Rect”
76	data	**NumCircles**	int	The number of macro-objects of type “Circle”
77	data	**Circle[n][25].X0**	float	Coordinate X (m) the center of solid macro-object type "Circle"
78	data	**Circle[n][26].Y0**	float	Coordinate Y (m) the center of solid macro-object type "Circle"
79	data	**Circle[n][27].Xstart**	float	Coordinate X (m) starting point solid macro-object type "Circle"[28]

17 [n] — number (index) of the object “Flow2D”,1..n
18 [n] — number (index) of the object “Rect”,1..n
19 [n] — number (index) of the object “Rect”,1..n
20 [n] — number (index) of the object “Rect”,1..n
21 [n] — number (index) of the object “Rect”,1..n
22 [n] — number (index) of the object “Rect”,1..n
23 [n] — number (index) of the object “Rect”,1..n
24 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 1st equation (Spalart-Allmaras model, Sekundov Nut-92 model)
4 — RANS models with 2 equations (k-ε, k-ω)
25 [n] — number (index) of the object “Circle”,1..n
26 [n] — number (index) of the object “Circle”,1..n
27 [n] — number (index) of the object “Circle”,1..n
28 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

Table 1

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

coordinates Xstart, Ystart

29 [n] — number (index) of the object Circle,1..n

30 [n] — number (index) of the object Circle,1..n

31 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 1st equation (Spalart-Allmaras model, Sekundov Nut-92 model)

4 — RANS models with 2 equations (k-ε, k-ω)

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

33 [n] — number (index) of the object Circle,1..n

34 [n] — number (index) of the object Airfoil,1..n

35 [n] — number (index) of the object Airfoil,1..n

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

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

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

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

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

Table 1

№	Object type	Object name	Value type	Description
92	data	**Airfoil[n][41].scale**	float	Airfoil scale (corresponding to the length of the chord, m)
93	data	**Airfoil[n][42].attack_angle**	float	The angle of attack of macro-object “Airfoil”
94	data	**Airfoil[n][43].Flow2D**	int	The index (number) of an object “Flow2D” for parameters initialized boundary of object “Airfoil”
95	data	**Airfoil[n][44].TurbulenceModel**	int	The type of turbulence model[45] on the border of the object “Airfoil”
96	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
97	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
98	data	**y_max**	int	Upper (max) limit of zone for output heat flux along X direction (in nodes). Affected only if **isOutputHeatFluxX=1**
99	data	**y_min**	int	Lower (min) limit of zone for output heat flux along X direction (in nodes). Affected only if **isOutputHeatFluxX=1**
100	data	**NumSingleBounds**	int	The number of objects "single boundary"
101	table	**SingleBound[n][iii].Points**	-	Points — coordinates of the starting and end coordinates (m) of the segment in the form: X_{start} Y_{start} X_{end} Y_{end}

41 [n] — number (index) of the object Airfoil,1..n
42 [n] — number (index) of the object Airfoil,1..n
43 [n] — number (index) of the object Airfoil,1..n
44 [n] — number (index) of the object “Airfoil”
45 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 1st equation (Spalart-Allmaras model, Sekundov Nut-92 model)
4 — RANS models with 2 equations (k-ε, k-ω)

Table 1

№	Object type	Object name	Value type	Description
102	data	**SingleBound[n][iv].Cond**	string	BC[v] defined along the object SingleBound
103	data	**SingleBound[n][vi].Flow2D**	int	The index (number) of an object “Flow2D” for parameters initialized boundary of object SingleBound
104	data	**SingleBound[n][vii].TurbulenceModel**	int	The type of turbulence models[46] on the border of the object “SingleBound”
105	data	**SingleBound[n][viii].MaterialID**	int	Bound material ID 0 – Gas 1 – Solid
106	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
107	data	**NumContour**	int	The number of objects “Contour” (many boundaries united in a closed loop)
108	table	**Contour[n][47]**	-	The coordinates of the boundaries in closed loop
109	data	**Contour[n][48].Bound[m][49].Cond**	string	BC[x] defined along a segment of the object “Contour[n]” number [m]
110	data	**Contour[n][50].Bound[m].Flow2D**	int	The index (number) of an object “Flow2D”, associated with the segment “Bound[m]” of object “Contour[n]”
111	data	**Contour[n][51].** **Bound[m].TurbulenceModel**	int	The type of turbulence models[52] along of the segment “Bound[m]” of object “Contour[n]”

46 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 1st equation (Spalart-Allmaras model, Sekundov Nut-92 model)
4 — RANS models with 2 equations (k-ε, k-ω)

47 [n] — number (index) of the object “Contour”

48 [n] — number (index) of the object “Contour”

49 [m] — number (index) of the object “Bound” in object “Contour”

50 [n] — number (index) of the object “Contour”

51 [n] — number (index) of the object “Contour”

52 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 1st equation (Spalart-Allmaras model, Sekundov Nut-92 model)

Table 1

№	Object type	Object name	Value type	Description
112	data	**Contour[n][53].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
113	data	**Contour[n][54].Bound[m].MaterialID**	int	Bound material ID 0 – Gas 1 – Solid
114	data	**NumArea**	int	The number of objects "Area"
115	table	**Area[n]**	-	The coordinates of the beginning of the initialization object "Area" with index [n] (x,y) **in nodes** !
116	data	**Area[n].MaterialID**	int	Material ID of solid body nodes (only for Type = 0)
117	data	**Area[n].Type**	int	Type of initialized nodes 0 — «solid» 1 — «gas»
118	data	**Area[n].Flow2D**	int	The index (number) of an object "Flow2D" which parameters is initialized object "Area[n]" (only for "gas" nodes)
119	data	**Area[n].Turbulence**	int	The type of turbulence model[55] in object "Area[n]"(only for "gas" nodes)
120	data	**H_cp**	float	The heat of formation of combustion products (J/kg)
121	data	**R_cp**	float	The gas constant of the combustion products (J/(kg*K))
122	table	**lam_cp**	-	The thermal conductivity of the combustion products as a function of temperature (J/(kg*K))
123	table	**mu_cp**	-	The dynamic viscosity of the products of combustion as a function of temperature (Pa*sec)

4 — RANS models with 2 equations (k-ε, k-ω)

53 [n] — number (index) of the object "Contour"

54 [n] — number (index) of the object "Contour"

55 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 1st equation (Spalart-Allmaras model, Sekundov Nut-92 model)
4 — RANS models with 2 equations (k-ε, k-ω)

Table 1

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

Table 1

№	Object type	Object name	Value type	Description
140	data	**Cp_Flow_index**	int	The index (number) of an object “Flow2D” for calculating pressure coefficient along walls, Cp = (effected only if **isOutputHeatFluxX=1**)
141	data	**is_Cx_calc**	int	Calculate the drag coefficient 0 – No 1 – Yes
142	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**)
143	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**)
144	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**)
145	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**)
146	data	**Cx_Flow_index**	int	The index (number) of an object “Flow2D” for calculating drag coefficient
147	data	**is_Cd_calc**	int	Calculate the discharge coefficient Cd (for nozzle) 0 – No 1 – Yes
148	data	**x_nozzle**	float	Initial X coordinate (m) of tested nozzle
149	data	**y_nozzle**	float	Initial Y coordinate (m) of tested nozzle
150	data	**dy_nozzle**	float	Cross-section size of tested nozzle (m)
151	data	**Cd_Flow_index**	int	The index (number) of an object “Flow2D” for calculating discharge coefficient
152	data	**p_ambient**	float	Reference ambient pressure for calculating velocity coefficient Cv (for nozzle)
153	data	**NumXCut**	int	Number of probed cross-sections along the X axis

Table 1

№	Object type	Object name	Value type	Description
154	data	**CutX-[n].x0**	float	Coordinate X (m) of the start cross-section with the index [n]
155	data	**CutX-[n].y0**	float	Coordinate Y (m) of the start cross-section with the index [n]
156	data	**CutX-[n].dy**	float	Size of cross-section(m) along the Y axis with the index [n]

i

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-ε Spalding model
5 — Chien k-ε model
6 — Jones-Launder k-ε model
7 —Launder-Sharma k-ε model with Yapp correction
8 — RNG k-ε model
9 — 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 (number) of an object "SingleBound"

iv [n] — The index (number) of an object "SingleBound"

v BC may have the following values:

Base BC:

CT_NO_COND_2D - BC not defined
CT_Ro_CONST_2D - ρ=const
CT_U_CONST_2D - ρU=const
CT_V_CONST_2D - ρV=const
CT_T_CONST_2D - ρE=const
CT_Y_CONST_2D - ρY=const
CT_dRodx_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_dRody_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_d2Rodx2_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_d2Rody2_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»

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_dRodx_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_d2Rodx2_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_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 (number) of an object "SingleBound"
vii [n] — The index (number) of an object "SingleBound"
viii [n] — The index (number) of an object "SingleBound"
ix [n] — The index (number) of an object "SingleBound"
x BC may have the following values:

Base BC:

CT_NO_COND_2D - BC not defined
CT_Ro_CONST_2D - ρ=const
CT_U_CONST_2D - ρU=const
CT_V_CONST_2D - ρV=const
CT_T_CONST_2D - ρE=const
CT_Y_CONST_2D - ρY=const
CT_dRodx_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_dRody_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_d2Rodx2_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_d2Rody2_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»

Macro defenition 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_dRodx_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_d2Rodx2_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_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