CD2TET Manual

Version 1.1

Edition 0.1/15 July 2016

Farhat Research Group (FRG)
Stanford University

This manual was prepared with Texinfo (http://www.gnu.org/software/texinfo).

Short Contents

- [CD2TET](#)
- [1 INTRODUCTION](#)
- [2 WHAT CAN CD2TET BE USED FOR?](#)
- [3 INPUT FILES](#)
- [4 OUTPUT FILES](#)
- [5 LINE COMMAND INPUT](#)
- [6 APPENDIX1: FORMAT OF THE BOUNDARY CONDITIONS AND FLUID VOLUME AND SURFACE IDENTIFICATION FILE GENERATED BY ICEM-CFD](#)
- [7 APPENDIX2: XPOST FORMAT OF A FLUID MESH](#)
- [8 APPENDIX3: FIELDVIEW FORMAT OF A FLUID MESH](#)
- [9 APPENDIX4: COMMON XPOST AND SINUS FORMAT OF A DISTANCE-TO-THE-WALL FILE](#)
- [10 APPENDIX5: SINUS FORMAT OF A FLUID MESH](#)

Table of Contents

Up: [(dir)](#)

CD2TET

Next: [What Can CD2TET Be Used For?](#)

1 INTRODUCTION

CD2TET is a pre/post-processing software whose development started at the Center for Aerospace Structures at the University of Colorado at Boulder to perform, among others, the generation of the top-level input files of the three-dimensional **AERO-F3D** fluid code — that is, **SOWER**'s and **MATCHER**'s input files — from the output of the **ICEM CFD** mesh generation software. This User's Manual documents the basic operations of **CD2TET**.

Next: [InputFiles](#), Previous: [Introduction](#)

2 WHAT CAN CD2TET BE USED FOR?

CD2TET can be used to perform the following tasks:

- Read a binary **ICEM CFD** mesh file and generate the corresponding
  - ASCII **XPost** (or **FieldView**) fluid mesh geometry file for input to the **MATCHER** and/or **SOWER** code(s).
  - ASCII **FieldView** mesh file for post-processing using the **FieldView** software.
- Create the distance-to-the-wall input file required by **AERO-F** for turbulent flow simulations using the Spalart-Allmaras turbulence model.
- Convert meshes from the **XPost** format to the **Sinus** format and vice versa.
- Convert meshes from the **XPost** or **Sinus** format to the **ICEM CFD** format. This option is useful when a mesh is generated using a different software than **ICEM CFD**, but the user wishes to input it to **ICEM CFD** for assessing its quality or editing it as follows:
  - Smoothing volume elements.
  - Identifying and correcting meshing errors.
  - Refining parts or all of the mesh.
  - Extruding prism layers.

  In addition, if only a surface mesh is available to the user, this option enables the generation of a volume mesh and/or the extraction of a faceted geometry by loading this surface mesh into **ICEM CFD** and performing the desired operation.
- Symmetrize a mesh with respect to a user-specified symmetry plane.

The instructions for performing the above tasks are expressed via line commands as described in [LineCommandInput](#).

Next: [OutputFiles](#), Previous: [What Can CD2TET Be Used For?](#)

3 INPUT FILES

The **CD2TET** input files do not follow any naming convention. They are organized in two categories as follows.

- [Input Files Generated by ICEM-CFD](#)
- [Other Input Files](#)

3.1 INPUT FILES GENERATED BY ICEM-CFD

The most important purpose of **CD2TET** is perhaps to read a binary **ICEM CFD** mesh file and generate the corresponding ASCII **XPost** and/or **FieldView** mesh file(s) required by **AERO-F**'s pre- and post-processing codes **MATCHER** and **SOWER**. In this case, the two input files required by **CD2TET** are:

- `<file name>.uns`. This binary mesh (or domain) input file is generated by **ICEM CFD**. It describes the unstructured discretization of the fluid domain. The mesh usually contains multiple surface and element sets. Each set contains a certain number of elements and is usually referred to as a subdomain. However, it is important not to confuse **ICEM CFD**'s subdomains with those generated by **Metis** when applied to decompose the fluid mesh into subdomains.
- `<file name>.fbc`. This ASCII file specifies the boundary conditions and fluid material properties for the different subdomains defined in the mesh file. The format of this file is given in [Appendix1](#).

The boundary conditions and fluid material properties file is created by an **AERO-F**-**ICEM CFD** interface that the user should embed in the **ICEM CFD** software as follows:

- Get the **CD2TET** source code.
- Locate the files `aerof3d`, `aerof3d.bcinfo` and `aerof3d.tcl` in the directory `Cd2tet.d/icemcfd/`. These files constitute the source code of the **AERO-F**-**ICEM CFD** interface.
- Copy the aforementioned files to the directory `ICEM_ACN/icemcfd/output-interfaces`, where `ICEM_ACN` denotes the path to the **ICEM CFD** installation directory. For example, it can be `/ansys_inc/v110/icemcfd/linux64_amd` for the 64-bit version 11.0 of **ICEM CFD** or `/ansys_inc/v110/icemcfd/linux` for its 32-bit version 11.0.
- In the file `solver_table.tcl` located in the directory `ICEM_ACN/icemcfd/output-`

`interfaces`, add under

```
    set solver_interfaces {
```

the following line

```
        {{Aero-F3D}                    aerof3d    aerof3d      0 }
```

Previous: [Input Files Generated by ICEM-CFD](#), Up: [InputFiles](#)

3.2 OTHER INPUT FILES

CD2TET can also be used to convert meshes from the **XPost** format to the **Sinus** format (and vice versa), create the distance-to-the-wall input file required by **AERO-F** for turbulent flow simulations using the Spalart-Allmaras turbulence model, and convert mesh files from the **XPost** or **Sinus** format to the **ICEM CFD** format. To perform any of these tasks, **CD2TET** requires one of the following input files:

- `<file name>.top`. This file contains in the **XPost** format the fluid mesh.
- `<file name>.sinus`. This file contains in the **Sinus** format the fluid mesh.

Next: [LineCommandInput](#), Previous: [InputFiles](#)

4 OUTPUT FILES

Unless otherwise specified by the user, the output files generated by **CD2TET** follow the naming convention given below. The user should also note that existing files with the same names as the files to be outputted are overwritten.

If the (mesh) input file to **CD2TET** is in the **ICEM CFD** or **Sinus** format, the following output files are generated by **CD2TET**:

- `<file name>.top`. This output file can be generated in either the **XPost** or **FieldView** format. It contains the fluid mesh and boundary conditions. The fluid domain is specified as an element with the name "FluidMesh". The surface boundary conditions are organized in element sets with the following naming convention:
    - "OutletMoving" for the moving outflow boundary;
    - "InletMoving" for the moving inflow boundary;
    - "StickMoving" for the moving viscous wall boundary;
    - "SlipMoving" for the moving Euler wall boundary;
    - "Symmetry" for the symmetry boundary;
    - "SlipFixed" for the fixed Euler wall boundary;
    - "StickFixed" for the fixed viscous wall boundary;

- ○ "InletFixed" for the fixed inflow boundary;
- ○ "OutletFixed" for the fixed outflow boundary.

  The **XPost** and **FieldView** formats of a mesh file are given in [Appendix2](#) (see also **XPost**'s User's Manual and [Appendix3](#), respectively).

- `<file name>.top.dwall`. This output file is generated in the **XPost** format. It contains for every mesh point its distance to the closest wall. The format of this file is described in [Appendix4](#).

However, if the (mesh) input file to **CD2TET** is in the **XPost** format, **CD2TET** generates in this case the following output files:

- `<file name>.sinus`. This output file is created in the **Sinus** format. It contains the fluid mesh. The **Sinus** format of a mesh file is described in [Appendix5](#).
- `<file name>.sinus.dwall`. This output file is generated in the **Sinus** format. It contains for every mesh point its distance to the closest wall.

Finally, if **CD2TET** is used to convert a mesh from the **XPost** or **Sinus** format to the **ICEM CFD** format, the following bindary output file is generated:

- `<file name>.uns`. This output file is created in the **CD2TET** binary format. It contains the fluid mesh and boundary conditions.

Next: [Appendix1](#), Previous: [OutputFiles](#)

5 LINE COMMAND INPUT

The different line-commands for performing the various tasks described in [Introduction](#) are as follows:

- To generate the mesh and the distance-to-the-wall files in the **XPost** format starting from an **ICEM CFD** input mesh file

cd2tet	`-domain <domain file>`	**ICEM CFD** `mesh file.`
	`-boco <boundary conditions file>`	Boundary conditions and material properties file `<name.fbc>`.
	`[-output <output_file>]`	Output file name. Default is icem.top (for the mesh file) and icem.top.dwall (for the distance-to-the-wall file).
	`[-fieldview]`	Outputs the mesh in the **FieldView** format.
	`[-no_check]`	Skips building interior and boundary faces and checking for ``internal'' boundary faces and negative volumes.
	`[-nowall]`	Skips the generation of the distance-to-the-wall file.

	`[-xsym <location>]`	Symmetrizes the mesh with respect to a plane parallel to the yz-plane and located at x=location.
	`[-ysym <location>]`	Symmetrizes the mesh with respect to a plane parallel to the zx-plane and located at y=location.
	`[-zsym <location>]`	Symmetrizes the mesh with respect to a plane parallel to the xy-plane and located at z=location.

The domain and boundary conditions files are always required for the above command to work.

- To generate the mesh and distance-to-the-wall files in the **XPost** or **Sinus** format starting from a **Sinus** or **XPost** input mesh file

cd2tet	`-mesh <mesh file>`	**XPost** or **Sinus** input mesh file.
	`[-output <output_file>]`	Default is xpost.sinus and xpost.sinus.dwall for the **Sinus** output, and sinus.top and sinus.top.dwall for the **XPost** output.
	`[-no_check]`	Skips building interior and boundary faces and checking for ``internal'' boundary faces and for negative volumes.
	`[-nowall]`	Does not create the distance-to-the-wall file.
	`[-write_icem]`	Outputs the mesh in the **ICEM CFD** format. The default is xpost.sinus.uns.
	`[-xsym <location>]`	Symmetrizes the mesh with respect to a plane parallel to the yz-plane and located at x=location.
	`[-ysym <location>]`	Symmetrizes the mesh with respect to a plane parallel to the zx-plane and located at y=location.
	`[-zsym <location>]`	Symmetrizes the mesh with respect to a plane parallel to the xy-plane and located at z=location.

Notes:

1. by default, **CD2TET** computes for every mesh point its distance to the closest wall, i.e., to the "StickFixed" or "StickMoving" boundary faces; if the mesh does not contain this type of boundary faces, one must use the option `-nowall` because otherwise **CD2TET** will exit for not finding any solid wall;

2. when symmetrizing a mesh with respect to a symmetry plane, if the boundary faces that coincide with this symmetry plane are labeled "Symmetry" they are not duplicated; if the user wishes to duplicate these faces (and their corresponding nodes), he/she should not label these faces as "Symmetry";

3. if **CD2TET** finds that the input mesh contains boundary faces that are surrounded by fluid elements, it prints a warning message indicating that a certain number of boundary faces are internal, and organizes these in a separate element set in the output file; this allows the user to easily visualize these faces using **XPost** and consequently fix the problem.

Next: [Appendix2](), Previous: [LineCommandInput]()

6 APPENDIX1: FORMAT OF THE BOUNDARY CONDITIONS AND FLUID VOLUME AND SURFACE IDENTIFICATION FILE GENERATED BY ICEM-CFD

The format of the boundary conditions and fluid volume identification file generated by **ICEM CFD** is described by the following example:

```
# family_boco file written by ANSYS ICEM CFD by boumosle on Wed Nov 16 09:53:31 PST 2005
## solver Aero-F3D
#ARGS noname noname noname noname noname noname noname noname noname noname noname
"WALL1"
"WALL1" "Stick" "Moving" 1 0 0 0 0 0 0 0 0 \
0 0
#ARGS noname noname noname noname noname noname noname noname noname noname noname
"WALL2"
"WALL2" "Stick" "Moving" 2 0 0 0 0 0 0 0 0 \
0 0
#ARGS noname noname noname noname noname noname noname noname noname noname noname
"OUTLET"
"OUTLET" "Outlet" "Fixed" 3 0 0 0 0 0 0 0 0 \
0 0
#ARGS noname noname noname noname noname noname noname noname noname noname noname
"SYMMETRY"
"SYMMETRY" "Symmetry" "" 4 0 0 0 0 0 0 0 0 \
0 0
```

```
#ARGS noname noname noname noname noname noname noname noname noname noname noname
"FLUID1"
"FLUID1" "FLUID" "" 1 0 0 0 0 0 0 0 0 \
0 0
#ARGS noname noname noname noname noname noname noname noname noname noname noname
"FLUID2"
"FLUID2" "FLUID" "" 2 0 0 0 0 0 0 0 0 \
0 0
```

Notes:

1. every line starting with a "#" is a comment line that is ignored by **CD2TET**;
2. every element set (referring to a fluid volume or boundary surface) name is enclosed between quotations and followed by the name of the volume or the type of boundary conditions of the surface, and 11 numbers; the first number is an integer identifying the volume or the surface; the next six real numbers are the coordinates of two points

$$(x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2)$$

that define a rotation axis for the identified surface (for example, when it is a control surface); the last four numbers are ignored by **CD2TET**;

3. some versions of the **ICEM CFD** software output extra lines that contain the keyword "color"; such lines are ignored by **CD2TET**.

---

7 APPENDIX2: XPOST FORMAT OF A FLUID MESH

The format of an **XPost** fluid mesh (ASCII) file is described by the following example. Note that the comments used for explaining this format (right part of the page beginning with //) and the blank lines they induce are not part of the format. More details can be found in **XPost**'s User's Manual.

```
Nodes FluidNodes                          // Declaration and naming of a node
                                          set.
```

```
1 1.261110e-01 1.181100e-02 1.000000e-01        // Node number and x y z
                                                   coordinates.

2 1.270000e-01 1.270000e-02 1.000000e-01

...

Elements FluidMesh_1 using FluidNodes            // Declaration and naming of an
                                                    element set.

1 5 23236 165454 28776 55020                     // Global element number, type and
                                                    connectivity nodes.

2 5 71797 77629 962290 23279

...

Elements FluidMesh_2 using FluidNodes            // Declaration and naming of
                                                    another element set.

536117 5 963129 502907 50115 60473               // Global element number, type and
                                                    connectivity nodes.

536118 5 502904 502894 502896 502348

...

Elements StickMovingSurface_1 using FluidNodes   // Declaration, naming and
                                                    identification of a boundary faces
                                                    element set.

1 4 1 12437 67                                   // Global element number, type and
                                                    connectivity nodes.

2 4 1 559 12437

...

Elements StickMovingSurface_2 using FluidNodes   // Declaration, naming and
                                                    identification of another boundary
                                                    faces element set.

1 4 30 2 12465                                   // Global element number, type and
                                                    connectivity nodes.

2 4 2 30 12469

...

Elements OutletFixedSurface_0 using FluidNodes   // Declaration, naming and
                                                    identification of another boundary
                                                    faces element set.

1 4 3911 61498 12                                // Global element number, type and
                                                    connectivity nodes.

2 4 61498 4047 12

...
```

Note:

1. an integer that is appended to an element set name (and which becomes integral part of the element set name) identifies the volume or surface represented by that element set; the default (blank) integer value is 0.

8 APPENDIX3: FIELDVIEW FORMAT OF A FLUID MESH

The format of a **FieldView** mesh (ASCII) file is described by the following example. Note that the comments used for explaining this format (right part of the page beginning with //) and the blank lines they induce are not part of the format. More details can be found in **FieldView**'s User's Manual.

Code	Comment
`FIELDVIEW_Grids 2 4`	// Fieldview header and version identification (two integers).
`Grids`	// Grid declaration.
`1`	// Grid identification number.
`Boundary Table`	// Table of types of boundary conditions.
`4`	// Total number of types of boundary conditions in this table.
`0 Symmetry`	// The first entry is an integer that is either 0 if the boundary surface is not a wall, or 1 if it is a wall. The second entry is the name of the boundary condition.
`0 InletFixed`	
`0 OutletFixed`	
`1 StickMoving`	
`Nodes`	// Declaration of a node set.
`15610`	// Total number of nodes in this set.
`-5.000000e-01 1.000000e-01 2.667563e-04`	// Nodal x y z coordinates.
`-4.877294e-01 1.000000e-01 2.667563e-04`	
`...`	
`Boundary Faces`	// Declaration of a boundary faces elements set.
`31216`	// Total number of boundary faces in this set.
`1 3 2 1 4`	// Pointer to the line of the Boundary Table, number of vertices in the face and connectivity nodes.
`1 3 2 15610 1`	
`...`	

```
Elements                                              // Declaration of a volume element
                                                      set.

1 1 7805 7769 7807 7808                               // Element type, subtype (here,
                                                      always 1) and connectivity nodes.

1 1 7806 7769 7808 7807

...
```

---

9 APPENDIX4: COMMON XPOST AND SINUS FORMAT OF A DISTANCE-TO-THE-WALL FILE

The **XPost** (or **Sinus**) format of the distance-to-the-wall (ASCII) file is described by the following example. Note that the comments used for explaining this format (right part of the page beginning with //) and the blank lines they induce are not part of the format. More details on the format of a scalar result file can be found in **XPost**'s User's Manual.

```
Scalar walldist under load for FluidNodes          // Declaration of a scalar result
                                                   set named walldist for the node set
                                                   named FluidNodes.

15610                                              // Total number of nodes in
                                                   FluidNodes.

0.0                                               // Dummy time-step.

4.999433e-01                                      // Distance-to-the-wall value (for
                                                  each node in FluidNodes).

4.876741e-01

...
```

---

10 APPENDIX5: SINUS FORMAT OF A FLUID MESH

The format of a **Sinus** mesh (ASCII) file is described by the following example. Note that the comments used for explaining this format (right part of the page beginning with //) and the blank lines they induce are not part of the format.

```
15610 45288 31216                          // Total number of nodes, volume
                                           elements and boundary faces.

-5.000000e-01 1.000000e-01 2.667563e-04    // Nodal xyz coordinates.
-4.877294e-01 1.000000e-01 2.667563e-04

...
7769 7805 7807 7808                        // Connectivity nodes of each
                                           volume element.

7769 7806 7808 7807

...
6                                          // Boundary condition type of each
                                           boundary face.

6

6

4

...
2 1 4                                      // Connectivity nodes of each
                                           boundary face.

2 15610 1

4 1 3

15610 3 1

...
```