

Re-Formatting FUN3D ASCII Tecplot Surface Files for Paraview

By Brandon Gleeson Date: 2023-07-27

Introduction

This guide describes how a method to convert FUN3D (v14.0.1) ASCII (.dat) surface sampling and boundary output files into a format that Paraview (5.11.1) can read.

Breaking down the FUN3D Output File Header

The header of a surface sampling output file looks like this:

```
title="solution sampling"
variables = x y z rho u v w p
zone t="midPlaneSample", solutiontime= 0.1000000E+03, strandid=0, i=12585, j=22240, f=fepoint, et=triangle
0.000000E+00 0.000000E+01 0.000000E+00 0.000000E+00 0.112500E+00 0.500000E+00 0.151500E+01 0.2700150E+00
```

- *zone name* can stay
- *solutiontime* is not supported by Paraview, it does not prevent loading if left in place but will give a warning, so we wish to remove it
- *strandid* is not supported by Paraview, but again lets remove to avoid warning messages
- *i* is the number of nodes, should be replaced with “NODES=”
- *j* is the number of elements, should be replaced with “ELEMENTS=”
- *f=fepoint* should be replaced by “DATAPACKING=POINT” in this case – other DATAPACKING argument variations may be required for a specific case, see the Tecplot formatting guide as needed.
- *et=triangle* should be replaced by “ZONETYPE=FETRIANGLE” in this case – other argument variations may be needed.

Basic 'sed' Linux terminal commands

We can use the Linux terminal 'sed' (stream editor) command to convert the 3rd line to the format Paraview can read:

Let's start with the "solutiontime" argument, which should be removed:

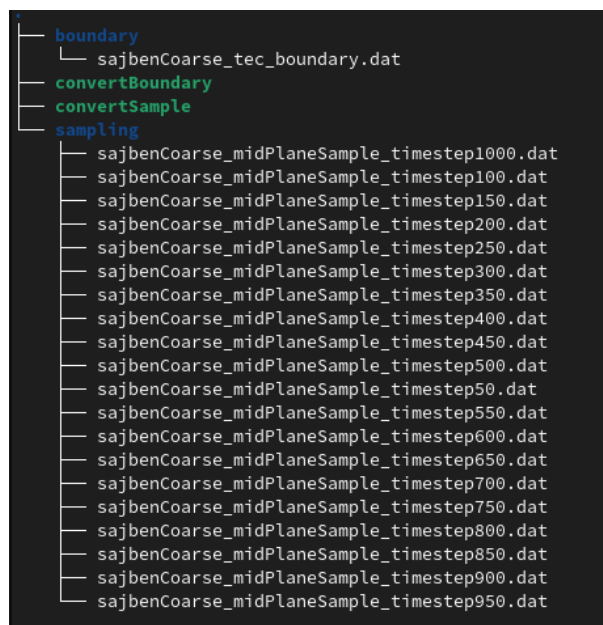
- ```
sed -i '3s/solutiontime= [^,]*, //g' *
```
- 3s = look only at line 3 – omit the 3 to look at all lines in the file, would want this if multiple zones are in the same file, otherwise leave it at 3 to avoid searching entire files
  - /solutiontime= [^,]\*,/ = the string we want to remove, will delete everything that is not a comma up to the comma, in this way all unique timestamp numerical values are removed
  - // = this would be the string we would replace, in this case we just want to remove, so leaving it without a replace argument achieves this
  - g' = is telling the sed command we wish to replace the pattern space with the hold space (i.e. we're not appending)

Similarly, for the other arguments:

```
sed -i '3s/strandid=[^,]*, //g' *
sed -i '3s/i=/NODES=/g' *
sed -i '3s/j=/ELEMENTS=/g' *
sed -i '3s/f=fepoint/DATAPACKING=POINT/g' *
sed -i '3s/et=triangle/ZONETYPE=FETRIANGLE/g' *
```

## Organizing the Data for Conversion

The previous commands can be issued from the terminal, but in practice we may wish to script this process. In the following steps, we will move (or copy, to be safe) the \*.dat output files to a /results/boundary and /results/sampling subdirectory and run a script file from the /results directory level. Here is an example of the results directory structure within the FUN3D run directory:



```
├── boundary
│ └── sajbenCoarse_tec_boundary.dat
├── convertBoundary
├── convertSample
└── sampling
 ├── sajbenCoarse_midPlaneSample_timestep1000.dat
 ├── sajbenCoarse_midPlaneSample_timestep100.dat
 ├── sajbenCoarse_midPlaneSample_timestep150.dat
 ├── sajbenCoarse_midPlaneSample_timestep200.dat
 ├── sajbenCoarse_midPlaneSample_timestep250.dat
 ├── sajbenCoarse_midPlaneSample_timestep300.dat
 ├── sajbenCoarse_midPlaneSample_timestep350.dat
 ├── sajbenCoarse_midPlaneSample_timestep400.dat
 ├── sajbenCoarse_midPlaneSample_timestep450.dat
 ├── sajbenCoarse_midPlaneSample_timestep500.dat
 ├── sajbenCoarse_midPlaneSample_timestep50.dat
 ├── sajbenCoarse_midPlaneSample_timestep550.dat
 ├── sajbenCoarse_midPlaneSample_timestep600.dat
 ├── sajbenCoarse_midPlaneSample_timestep650.dat
 ├── sajbenCoarse_midPlaneSample_timestep700.dat
 ├── sajbenCoarse_midPlaneSample_timestep750.dat
 ├── sajbenCoarse_midPlaneSample_timestep800.dat
 ├── sajbenCoarse_midPlaneSample_timestep850.dat
 ├── sajbenCoarse_midPlaneSample_timestep900.dat
 └── sajbenCoarse_midPlaneSample_timestep950.dat
```

From this example, sampling data files were written every 50 time steps, and one boundary file written at the end. *convertBoundary* and *convertSample* are the script files described in the following sections of this document.

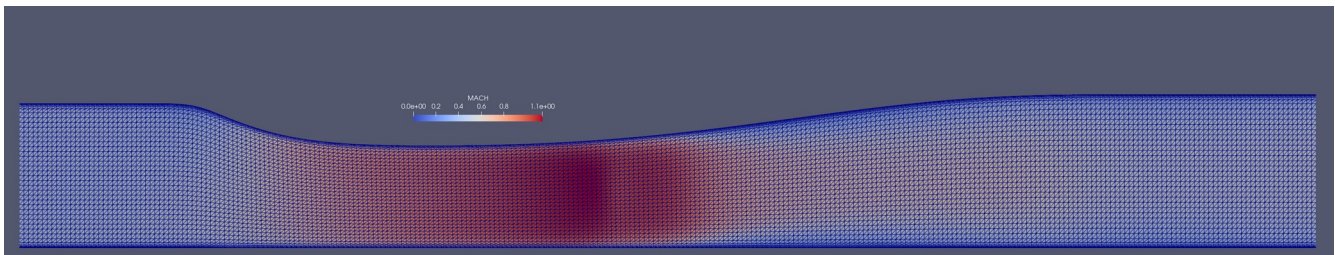
## Converting Sampling Parameter Files (mid-plane example shown)

The following commands could be issued from the terminal from within the /sampling directory

```
sed -i '3s/solutiontime= [^,]*, //g' *
sed -i '3s/strandid=[^,]*, //g' *
sed -i '3s/i=/NODES=/g' *
sed -i '3s/j=/ELEMENTS=/g' *
sed -i '3s/f=fepoint/DATAPACKING=POINT/g' *
sed -i '3s/et=triangle/ZONETYPE=FETRIANGLE/g' *
```

Alternatively the following block of arguments can be saved as a Linux script file and run as an executable from the \results directory. Each argument adds a command to search within the /sampling directory and execute the sed command as required:

```
find ./sampling -type f -exec sed -i '3s/solutiontime= [^,]*, //g' {} \;
find ./sampling -type f -exec sed -i '3s/strandid=[^,]*, //g' {} \;
find ./sampling -type f -exec sed -i '3s/i=/NODES=/g' {} \;
find ./sampling -type f -exec sed -i '3s/j=/ELEMENTS=/g' {} \;
find ./sampling -type f -exec sed -i '3s/f=fepoint/DATAPACKING=POINT/g' {} \;
find ./sampling -type f -exec sed -i '3s/et=triangle/ZONETYPE=FETRIANGLE/g' {} \;
```



Note that this domain is a hexahedral mesh, but when FUN3D interpolates a plane extraction from the domain, it triangulates the data.

## Converting Boundary Output Files (two wall boundaries shown)

For boundary files, a few changes are needed, and here let's assume more than 1 surface zone will be written (drop the 3) from ('3s/). The following individual arguments could be issued from the terminal in /boundary directory:

```
sed -i 's/solutiontime= [^,]*, //g' *
sed -i 's/strandid=[^,]*, //g' *
sed -i 's/i=/NODES=/g' *
sed -i 's/j=/ELEMENTS=/g' *
sed -i 's/f=feblock/DATAPACKING=BLOCK, ZONETYPE=FEQUADRILATERAL/g' *
```

Alternatively, this script from /results could be run:

```
find ./boundary -type f -exec sed -i 's/solutiontime= [^,]*, //g' {} \;
find ./boundary -type f -exec sed -i 's/strandid=[^,]*, //g' {} \;
find ./boundary -type f -exec sed -i 's/i=/NODES=/g' {} \;
find ./boundary -type f -exec sed -i 's/j=/ELEMENTS=/g' {} \;
find ./boundary -type f -exec sed -i 's/f=feblock/DATAPACKING=BLOCK, ZONETYPE=FEQUADRILATERAL/g' {} \;
```

