Converting Gambit Grids for Semtex

Erik Torres, Jörg Stiller

TU Dresden, Institute for Aerospace Engineering (ILR)
01062 Dresden, Germany

December 15, 2008

1 Introduction

This guide gives a brief overview on the conversion of gambit grid files to semtex session files using gambit2semtex. The following features are supported:

- Import of unstructured quadrilateral grids.
- Support of periodic boundaries.
- Support of semtex curves of the types `ARC` and `SPLINE`.
- Reuse or automated generation of spline data.

A drawback of the current version is the large overhead in the sources, which results from the fact that the code was adopted from another application.

2 Installation

To build the conversion program you need a Fortran 95 compliant compiler. Possible choices are the freely available *g95* or *gfortran*, however any other up-to-date compiler (e.g. Intel's *ifort*) should work as well. Change to the source directory `src` and edit `GNUmakefile` such that the variables `FC` (referencing the compiler) and `FOPTS` (compiler options) are initialised properly. To build the program just type

```
> make
```

For generating the session file proceed as follows:

1. Produce a suitable meshfile using gambit.
2. Create the input file for gambit2semtex and execute the program.
3. Add the problem specific sections to the resulting session file.

Some import details are discussed in the following sections. The corresponding example files are located in the directory `example`.

3 Preparing the grid file

It is beyond the scope of this document to give an introduction to gambit. Instead we summarise the aspects that are crucial for a successful application of the converter:

1. After starting gambit activate "FLUENT 5/6" in the "Solver" menu.
2. Create the two-dimensional geometry in the x, y plane ($z = 0$).
3. For curved boundaries use the built in primitives or import vertex data. Note that gambit requires three-dimensional data. The vertex file has to provide an ordered line-by-line listing of the cartesian coordinates with z set to zero. For an example see `back_geometry.dat` in `example/gambit`.
4. Periodic boundaries must be linked before meshing (`Operation / Mesh / Edge / Link Edge Meshes`).
5. Each boundary must be given a name and a type:
   - Activate the option `Zones` in the `Operation` toolpad.
   - Select the suboption `Specify boundary types`.
   - Now add the boundaries (termed `Edge` in gambit) one by one. Always select the type `WALL`, no matter what the actual boundary condition is. Specify a different name for each part of the boundary. Equally named boundaries will be merged. The names are reused in the gambit2semtex input and the semtex session files.
6. All parts (`Faces`) of the domain must be of type "FLUID" (`Zones / Specify continuum types`).

To export the final grid, select `File / Export / Mesh` from the menu bar, define the file name, and activate the option `Export / 2-D(X-Y) Mesh`. In the included example the grid file is named `example.msh`. The suffix ".msh" is added automatically. A gambit session file that corresponds to the example is provided in the directory `example/gambit`.

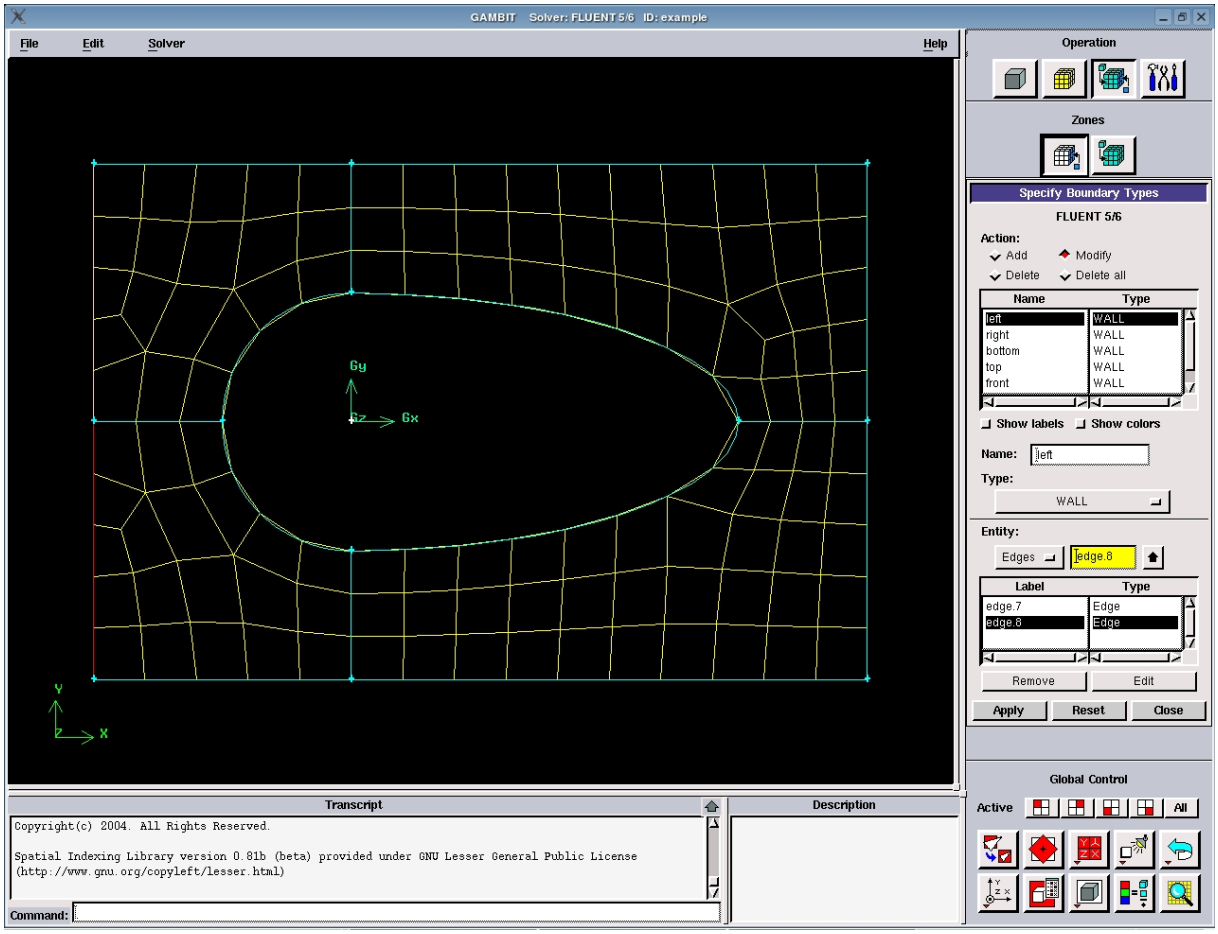

Figure 1: Defining the boundaries

4 The gambit2semtex input file

The next step is to create an input file for the converter. It has to provide the name of grid file and all information that is necessary to establish the boundary curves. A file matching our example is `example/example.inp`, which is listed in Figure 2. Looking at this, the experienced reader will recognise that the input data is specified in terms of Fortran namelists. Lines beginning with `#` are comments.

```
# case name for export
&ExportCase
casename = "example" /

# grid information
&grid
file   = "example.msh"
xScale = 1.000
yScale = 1.000 /

# boundary curves
&boundary_curve
name = "left", geom = "straight" /
&boundary_curve
name = "right", geom = "straight" /
&boundary_curve
name = "bottom", geom = "straight" /
&boundary_curve
name = "top", geom = "straight" /
&boundary_curve
name = "front", geom = "arc" , x0 = 0.0, y0 = 0.0 /
&boundary_curve
name = "back", geom = "spline" , file = "back_geometry.dat" /

# boundary conditions
&Boundary_Condition
name = "left", key = "v" /
&Boundary_Condition
name = "right", key = "o" /
&Boundary_Condition
name = "bottom", periodic_link = "top" /
&Boundary_Condition
name = "top", periodic_link = "bottom" /
&Boundary_Condition
name = "front", key = "w" /
&Boundary_Condition
name = "back", key = "w" /
```

Figure 2: Listing of `example.inp`.

The section `ExportCase` assigns the name of the semtex session file (`example`) to the variable `casename`. The next section defines the name of the mesh `file` (`example.msh`) and the scaling factors `xScale` and `yScale`. These factors can be used to scale the corresponding point coordinates. If skipped, they are set to 1 by default.

The next section defines the boundary curves. Here, `name` refers to the boundary name as specified before in the gambit session and `geom` defines the curve type. Three choices are possible:

- `geom = "straight"` defines a straight edge,
- `geom = "arc"` a circular arc, and
- `geom = "spline"` a spline curve.

In case of an arc it is mandatory to provide the mid point coordinates (`x0` and `y0`).

In case of a spline curve (`geom = "spline"`) the parameter `file` refers to a file containing the x, y, z coordinates of the spline points listed line by line. This allows to reuse the vertex file that was employed in gambit for creating the edge. During conversion a new geometry file is created. It contains the same coordinate information as the original file, but the third column (the z-coordinate) is omitted. This is necessary since Semtex expects 2D-data only.

If the specification of `file` is skipped, the converter generates a geometry file using the grid point coordinates on this boundary. In both cases the name of the geometry file is composed of the case name and the boundary name, e.g. `example_bndry_back.dat` for the boundary named "back" in the given example and a corresponding `CURVE` section is inserted in the semtex session file.

The final section contains the `Boundary_Condition` entries defining the semtex boundary groups. Each entry refers to one boundary that is identified by the `name` assigned in the gambit session. The `key` parameter (always a single characer) associates the boundary with one group defined in the `GROUP` section in the session file. It is used to create the corresponding entries in the `SURFACES` section. Concerning the conventions for choosing the group identifiers we refer to the semtex userguide. Typically `"w"` defines the "wall group", while periodic boundaries belong to no group. Thus, no key is specified for them. In the latter case the parameter `periodic_link` provides the name of the matching boundary. For instance, in the example given in Fig. 2 the boundaries "bottom" and "top" refer to each other.

Once the input file is completed, run the converter:

```
> gambit2semtex
```

The program prompts for the name of the input file and creates the session file in the same directory.

5 Editing the session file

To complete the session file, you have to fill in some missing data in following sections:

- `USER`
- `FIELDS`
- `TOKENS`
- `GROUPS`
- `BCS`

In the `GROUPS` section you will find entries for every `key` that you specified previously in the input file, now you can replace the generic `group_name` with the one you want to use.

Also, in the `BCS` section you will have boundary condition entries for every group with their respective `key`. Now you can modify the boundary conditions to suit your problem.

For further information we refer to the semtex userguide [1].

References

[1] H. M. Blackburn. Using semtex. CSIRO Doc. 97/159(M), CSIRO Manufacturing and Infrastructure Technology, October 20 2004.