

1 Einleitung

Das Programm mit dem Namen „Converter“ kann verwendet werden, um Geometriedateien (mesh files), die mit Gambit erstellt wurden, in Semtex-Sessiondateien umzuwandeln. Hier soll dieser Vorgang und die Besonderheiten, auf die es dabei zu achten gilt, beschrieben werden.

2 Installation

Bevor das Programm angewendet werden kann, muß der Quelltext übersetzt werden. Dafür wird ein Fortran 95-Compiler, z.B. g95 und GNU's **make** nötig sein.

Nach dem Entpacken der tgz-Datei sollte man zuerst die Ordnersrtuktur überprüfen. Im Hauptordner sollten das Makefile und die vier Ordner **converter**, **src**, **lib** und **example** enthalten sein. Sollte der g95-Compiler nicht auf der aktuellen Maschine vorhanden sein, muß man im Makefile einen anderen Compiler angeben. Die ersten drei Ordner enthalten den Quelltext für das Programm, während im Ordner **example** Beispieldateien, auf die in dieser Anleitung verwiesen werden, für die Anwendung des Converters enthalten sind. Den Quelltext übersetzt man mit dem Befehl:

```
> make lib converter
```

Wenn dieser Vorgang erfolgreich war, existiert im Ordner **converter** eine neue ausführbare Datei, auch mit dem Namen **converter**. Um den Converter auch aus einem anderen Verzeichnis aufzurufen, muß man nur eine symbolische Verknüpfung im gewünschten Ordner, oder im **/bin**-Verzeichnis des aktuellen Benutzers anlegen.

Das Programm selbst wird mit dem Befehl **converter** aufgerufen. Es verlangt vom Benutzer die Angabe einer Input-Datei (hier: **example.inp**). Damit das Programm korrekt abläuft, müssen sich in demselben Ordner die mit Gambit generierte Gitterdatei (hier: **example.msh**) und die Input-Datei befinden. Der Inhalt und Aufbau wird später in diesem Dokument anhand des Beispiels erklärt.

Während der Konvertierung werden Meldungen zum Fortschritt des Vorganges angezeigt. Wenn dieser Schritt erfolgreich ist, liegen dann die Semtex-Eingabedatei und evtl. die zusätzlichen Dateien für die gekrümmten Ränder vor.

3 Vorberetung der Gitterdateien in Gambit

Erster Schritt ist es die Geometrie in Gambit zu erstellen und vollständig zu vernetzen. Außerdem muß als „Solver“ FLUENT5/6 eingestellt sein, denn das Konvertierprogramm kann die Gitterdatei nur in diesem Format lesen. Dies ist normalerweise die Standardeinstellung. Beim automatischen Vernetzen ist zu beachten, daß Semtex nur mit Viereck-Elementen arbeitet. Da Semtex nur zweidimensionale Geometrien verwendet, darf unter Gambit auch nur eine zweidimensionale Geometrie erstellt werden, d.h. alle Gitterknoten müssen in der x,y-Ebene liegen ($z = 0$).

Das Konvertierprogramm erkennt die Ränder in der Gitterdatei anhand der Namen, die ihnen unter Gambit im **Operation toolpad** unter der Option **Zones** → **Specify boundary types** zugewiesen werden. Unter Gambit hat jeder Rand einen eigenen Namen und eine zugeordnete Randbedingung. Das Konvertierprogramm erkennt aber nur jene Ränder als solche, die unter Gambit als Typ „WALL“ markiert sind. Dabei ist es vorerst egal, welche Randbedingungen an diesen Rändern tatsächlich herrschen sollen. Dies wird erst später in der erzeugten Semtex-Eingabedatei festgelegt. Die Namen, die man den jeweiligen Rändern unter Gambit gibt, sind allerdings wichtig, denn sie werden später in der Input-Datei (example.inp) verwendet, um die Ränder zu identifizieren.

Damit der Converter die in Gambit gesetzten Ränder richtig verarbeitet, sind einige Einschränkungen zu beachten:

- Jede einzelne Randkurve darf keine Unterbrechungen aufweisen, d.h. sie muß aus zusammenhängenden Geradenabschnitten bestehen.
- Eine bestimmte Randkurve darf nur an einer Kante eines Elementes verlaufen, d.h. man kann die Randkurve nicht „um die Ecke“ eines Elementes legen. (Es ist natürlich erlaubt, daß zwei verschiedene Randkurven an einem Element entlaglaufen und an einem der Elementknoten aufeinander treffen.)

Außerdem ist es noch wichtig unter **Zones** → **Specify continuum types** für das gesamte vernetzte Gebiet (also alle dazu gehörenden „faces“) ein Kontinuum vom Typ „FLUID“ zu erzeugen. Hier ist der gewählte Name unwichtig.

Die fertige Gitterdatei wird im Hauptmenü unter **File** → **Export** → **Mesh** ausgegeben. Hier muß die Option **Export 2-D (X-Y) mesh** markiert sein. Im vorliegenden Beispiel wird die Gitterdatei als „example.msh“ exportiert.

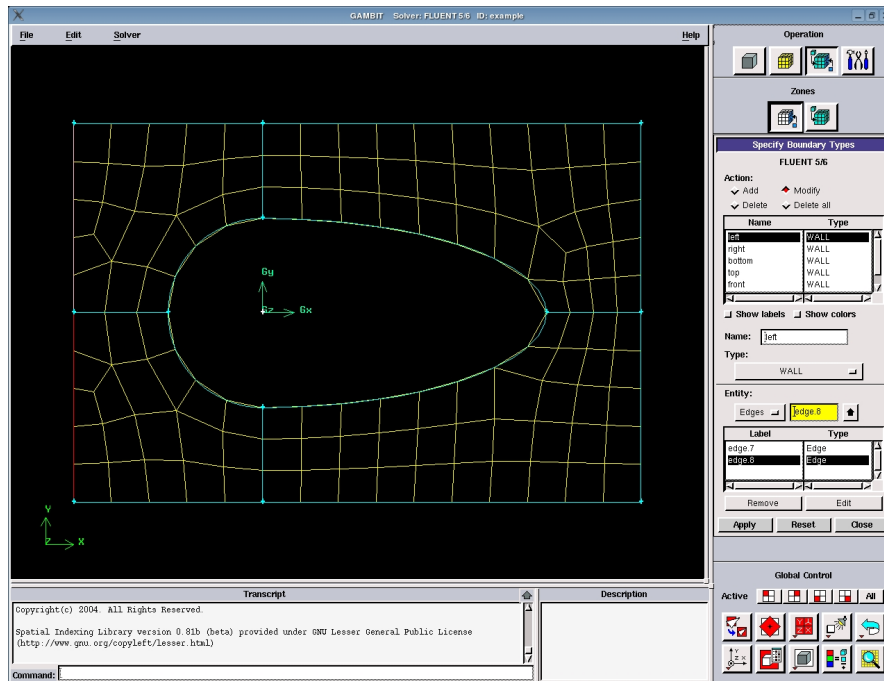


Abbildung 1: Den Rändern Namen geben und als WALL kennzeichnen

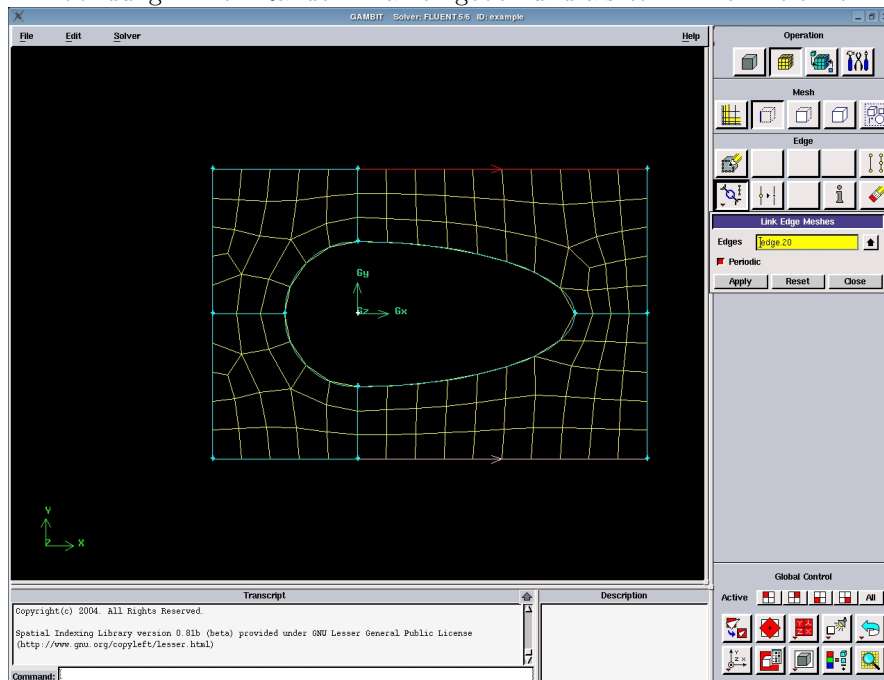


Abbildung 2: Periodische Ränder immer „gekoppelt“ vernetzen

4 Aufbau der Input-Datei

Um die fertige Gitterdatei (example.msh) in das Semtex-Format zu konvertieren ist außerdem noch die Input-Datei (hier example.inp) notwendig. Sie enthält Information über die Randgeometrie und die sog. Rand-Gruppen (boundary groups).

Ganz am Anfang der Datei, unter dem Eintrag `# case name for export` wird der Name der Semtex-Sessiondatei festgelegt (hier: `casename = example`). Als nächstes wird die mit Gambit erzeugte Gitterdatei unter `# grid information` angegeben (hier: `file = example.msh`). Die nächsten zwei Einträge (`xScale = 1.000`, `yScale = 1.000`) erlauben es die Skalierung in x- oder y- Richtung zu verändern. Der Letzte Eintrag (`type = 0`) gibt nur an, daß die Gitterdatei im FLUENT5/6 Format vorliegt. Der Converter kann vorläufig nur dieses Format interpretieren, deshalb braucht dieser Wert nie verändert zu werden. Danach kommen zwei größere Abschnitte, die direkt mit der Behandlung der Ränder zu tun haben.

Zuerst wird für jeden Rand im Abschnitt `boundary curves` der Typ der Randgeometrie festgelegt. Jeder Rand hat einen Namen, der ihm in Gambit zugewiesen wurde (z.B.: `name = left`). Es wird festgelegt, ob der Rand aus Geradenabschnitten zwischen den Knotenpunkten (`geom = 1`) besteht, oder ob die Randkurve mit kubischen Splines (`geom = 4`) oder Kreisbogensegmenten (`geom = 5`) beschrieben wird.

Hinweis: Die Werte `geom = 2`, bzw. `geom = 3` (jeweils für Bernstein Approximation und B-Splines) werden nicht verwendet. Hier handelt es sich um Überbleibsel aus einer früheren Version, die zwar im Quelltext des Converters noch vorhanden, in der aktuellen Version aber abgeschaltet sind.

Es gibt zwei verschiedene Varianten, wie kubische Splines verwendet werden können. Bei der ersten Variante werden die Elementknoten, die auf dem gekrümmten Rand liegen, als Kontrollpunkte für das Generieren einer Spline-Kurve verwendet. Die daraus erzeugte Kurve durchläuft dann genau alle diese Punkte. Sie wird in einer gesonderten Datei gespeichert, deren Namen sich aus dem „casename“ und dem Namen des jeweiligen Randes zusammensetzt. So heißt die für den Rand `back` erzeugte Datei `example_bndry-spline_back.dat`. Damit Semtex auf diese Datei zugreift, wird sie in der Eingabedatei unter dem Abschnitt `<CURVES>` aufgeführt.

Die zweite Variante benötigt eine vorher definierte Kurve als Ausgangsinformation. Diese Kurve muß als einfache Tabelle von x,y-Koordinaten in einer Textdatei gespeichert sein. Der Eintrag in der Input-Datei wird um die Angabe des Dateinamen erweitert, z.B.:

```
name = back, geom = 4, file='back_geometry.dat' /
```

Nun wird der Converter diese Datei als Referenz benutzen und die Elementknoten auf dem gegebenen Rand so verschieben, daß sie auf der Kurve liegen. In

diesem Fall wird in der Semtex-Eingabedatei auf die ursprüngliche Geometrie-Datei verwiesen und keine zusätzliche Spline-Kurve generiert.

Wenn für einen Rand Kreisbogensegmente verwendet werden sollen, ist es anstatt der Angabe einer Datei nötig, die Koordinaten des Kreismittelpunktes festzulegen (z.B.: `x0 = 0.0, y0 = 0.0`).

```
name = front, geom = 5, x0 = 0.0, y0 = 0.0 /
```

Der zweite große Abschnitt befasst sich mit den Randbedingungen, bzw. mit den unter Semtex verwendeten „boundary groups“. Wieder gibt es für jeden Rand einen Eintrag, zuerst kommt der Name, dann der zugeordnete „key“. Dieser key besteht aus nur einem Buchstaben. Er wird verwendet, um die Ränder in der Semtex-Sessiondatei den Randgruppen zuzuordnen, das heißt, alle Ränder die z.B. ein „w“ als Key zugeordnet haben, stellen unter Semtex eine Randgruppe dar. In Semtex werden dann alle betroffenen Elementränder im Abschnitt `<SURFACES>` mit diesem Key aufgeführt. Es ist möglich für verschiedene Ränder denselben Key anzugeben. Da in der Semtex-Sessiondatei jedes Randelement einzeln mit dem zugehörigen Key aufgeführt wird, können verschiedene Ränder zu einer Semtex-Randgruppe gehören. Prinzipiell sind alle Buchstaben als Key wählbar. Einzige Ausnahme ist das `p`, das zum kennzeichnen periodischer Ränder verwendet wird. Solche Ränder treten immer paarweise auf, und der zugehörige Rand muß angegeben werden (z.B.: `name = bottom, key = p, link = top`, bzw. `name = top, key = p, link = bottom`). Das Konvertierprogramm kann nicht erkennen, ob die Anzahl und Positionen der Knotenpunkte und die Abstände zwischen den Knoten auf den beiden Rändern übereinstimmen, deshalb muß dies schon in Gambit berücksichtigt werden. Dafür gibt es beim Vernetzen der Beiden Ränder in Gambit die Funktion `Link / Unlink edges`, diese ist über das Toolpad zu erreichen.

5 Ergänzung der Semtex-Eingabedatei

Damit die Eingabedatei mit Semtex verwendet werden kann, müssen noch einige Abschnitte hinzugefügt werden. Hauptsächlich die Abschnitte `<FIELDS>`, `<TOKENS>`, `<GROUPS>` und `<BCS>` sind für die meisten Rechnungen mit Semtex notwendig. Mehr Information dazu kann der Dokumentation für Semtex entnommen werden.

```

# case name for export
&ExportCase
casename = example /

# grid information
&grid
file    = example.msh
xScale  = 1.000
yScale  = 1.000
type    = 0 /

# boundary curves
&boundary_curve
name = left, geom = 1 /
&boundary_curve
name = right, geom = 1 /
&boundary_curve
name = bottom, geom = 1 /
&boundary_curve
name = top, geom = 1 /
&boundary_curve
name = front, geom = 5, x0 = 0.0, y0 = 0.0 /
&boundary_curve
name = back, geom = 4, file = 'back_geometry.dat' /

# boundary conditions
&Boundary_Condition
name = left, key = v /
&Boundary_Condition
name = right, key = o /
&Boundary_Condition
name = bottom, key = p, link = top /
&Boundary_Condition
name = top, key = p, link = bottom /
&Boundary_Condition
name = front, key = w /
&Boundary_Condition
name = back, key = w /

```

Abbildung 3: Inhalt der Datei *example.inp*