MeshGems

MeshGems-Cleaner: The automatic mesh fixing tool

User Manual

MeshGems-Cleaner Version: 1.1

Documentation Revision: 1.1

January 2013

Customer Service

Should you encounter any problem with MeshGems-Cleaner or the MeshGems® Suite, please contact the customer service by:

Address:	DISTENE S.A.S. Customer Support Service Campus Teratec 2, rue de la Piquetterie 91680 Bruyères-le-Chatel FRANCE
Telephone:	+33(0)970-650-219
Fax:	+33(0)169-269-033
Electronic mail:	support@distene.com

1st edition, January 2013

Information

This document is the user manual of the MeshGems-Cleaner software. It applies to all platforms running the software.

The MeshGems-Cleaner software is the automatic mesh fixing tool.

> **Important notice**: In order to use MeshGems-Cleaner, you must first install the tool and, if necessary, the license server. Please refer to the installation guide in the delivered CD-ROM or in the installed distribution.

Document History

Version	Date	Comment
1.1	January 2013	1st edition

Foreword

The MeshGems suite is a comprehensive set of meshing components, which aim at providing CAD/CAE application developers with reliable, fast and high quality meshing technologies. These components address all aspects of automatic meshing for real life industrial 3D numerical simulations, using very efficient algorithms:

- surface meshing from CAD,
- correction of data, whether geometrical or triangulated,
- volume mesh generation,
- remeshing,
- adaption driven by user-given local criteria.

The MeshGems suite is designed to make integration and maintenance easy by third party developers. The MeshGems suite today includes or will include:

- Tetra for tetrahedral volume mesh generation
- Hexa for hexahedral volume mesh generation
- CADSurf for triangular/quadrilateral mesh generation
- PreCAD and Cleaner for CAD and mesh correction respectively
- SurfOpt for surface remeshing and optimization
- Adapt for surface and/or volume mesh adaption

All these components can be used independently or as a compatible whole thanks to the common core module (MeshGems-Core) included in all the MeshGems components.

See section 2.5 for more information about coupling MeshGems-Cleaner with the other MeshGems® Suite components and section 4.3 for more information about MeshGems-Cleaner relationship with the physics and/or the CAD description.

Contents

Foreword 5
Contents 9

1 Introduction **11**

2 Getting started **13**
2.1 Installation 13
2.2 Requirements 13
2.2.1 Disk Space 13
2.2.2 CPU time 14
2.2.3 Memory usage 14
2.3 Launching the software 14
2.4 Errors and diagnostics 14
2.5 Coupling with other MeshGems® Suite components 14
2.6 Integration within third party products 15

3 MeshGems-Cleaner's principles **17**

4 MeshGems-Cleaner's utilisation **19**
4.1 Input Files 19
4.1.1 The `mesh` format 19
4.2 Output Files 20
4.3 Relationship with the physics and/or the CAD description 20
4.4 Invoking MeshGems-Cleaner 21
4.4.1 Running MeshGems-Cleaner 21
4.4.2 Mandatory options 21
4.4.2.1 Choosing the input file 21
4.4.3 Generic options 21
4.4.3.1 Setting the verbosity level 21
4.4.3.2 Setting the output file name 21
4.4.3.3 Getting online help 22
4.4.4 Module specific options and control parameters 23

4.4.4.1 Applicable operations . . . 24
4.4.4.2 Number of passes for the cleaning procedure . . . 24
4.4.4.3 Preserving topology . . . 24
4.4.4.4 Setting hole min size . . . 24
4.4.4.5 Setting tolerance parameter . . . 25
4.4.4.6 Setting resolution length . . . 25
4.4.4.7 Setting minimum angle . . . 25
4.4.4.8 Activating the remeshing of planar surface . . . 25
4.4.4.9 Setting the overlapping distance . . . 25
4.4.4.10 Setting the overlapping angle . . . 26
4.4.5 Using MeshGems-Cleaner with its default parameter values . . . 26
4.5 Detection of faults . . . 26

5 MeshGems-Cleaner's examples of use 27
5.1 Example 1: Cleaning to get a volume mesh . . . 27
5.1.1 Mesh checking . . . 27
5.1.2 Correcting the mesh with standard options . . . 30
5.2 Example 2: Cleaning to remesh and get a volume mesh . . . 31
5.2.1 Mesh checking . . . 31
5.2.2 Fixing the mesh . . . 33
5.2.3 Producing a surface mesh with better aspect ratio of triangles . . . 33
5.2.4 Improving mesh quality with MeshGems-Cleaner . . . 34

Appendices 37

A MeshGems-Cleaner's diagnostics 37
A.1 Syntax of the exe diagnostics . . . 37
A.2 List of MeshGems-Cleaner's diagnostics . . . 38

B File format description 39
B.1 the ".mesh" format . . . 39
B.1.1 Structure . . . 39
B.1.2 Sample . . . 39
B.2 the ".sol" format . . . 40
B.2.1 Structure . . . 40
B.2.2 Sample . . . 40
B.3 Keywords used by MeshGems-Cleaner . . . 41

C List of FAQ's 43

C.1 My mesh contains quads, is MeshGems-Cleaner capable of correcting these? 43
C.2 Is it a good idea to iterate MeshGems-Cleaner on itself? 43
C.3 How do I improve the quality of triangles? 44
C.4 Is MeshGems-Cleaner capable of guaranteeing a corrected mesh suitable for a Delaunay or front advancing tet mesher? 44
C.5 How is associativity taken care of ? 44
C.6 Can my geometry be altered ? 44
C.7 How can I preserve as much as possible my input topology ? 44
C.8 Does a specific treatment exist for planes? 44

D Release Note **45**
D.1 Version 1.1 45
D.1.1 New features 45
D.1.2 Behaviour changes 45
D.1.3 Improvements 45
D.1.4 Corrections 45

References **47**

Chapter 1

Introduction

This user manual describes the main features of MeshGems-Cleaner[1], the automatic mesh fixing tool.

MeshGems-Cleaner is an automatic meshing component designed to correct automatically surface meshes, in order to make them suitable for volume meshing. MeshGems-Cleaner aims at generating valid surface meshes out of faulty ones, and being automatic, does not require any interactive manipulations from the user.

MeshGems-Cleaner will either make an analysis of the surface mesh and output information highlighting possible faults, or make this analysis and proceed with corrective actions accordingly.

Typical problems detected and corrected include, but are not limited to:

- Overlaps
- Non conformities
- Holes or gaps preventing water-tightness
- Self-intersections
- Bad shaped elements
- Incorrect assemblies, up to a certain level of complexity.

MeshGems-Cleaner's behaviour and precision can be controlled with several user-tunable runtime parameters.

MeshGems-Cleaner is simply run as a command line executable with online options.

The input and output files, the interaction with the user, and the list of error messages are described in this document. Several typical application examples are provided to explain the various possibilities of the code. Several appendixes describe thoroughly the input/output formats and other useful information.

[1] Developed by M.Bory and Distene SAS

Chapter 2

Getting started

Contents

2.1 Installation . . . 13
2.2 Requirements . . . 13
2.2.1 Disk Space . . . 13
2.2.2 CPU time . . . 14
2.2.3 Memory usage . . . 14
2.3 Launching the software . . . 14
2.4 Errors and diagnostics . . . 14
2.5 Coupling with other MeshGems® Suite components . . . 14
2.6 Integration within third party products . . . 15

2.1 Installation

Please refer to the installation guide which describes the MeshGems® Suite component installation and its use. This guide is available both from the delivered media and from the distribution once installed under ".../DISTENE/Docs/"; in particular you may want to check out the section "Use of a Distene Tool" located in file "05-use-tool.html".

2.2 Requirements

The program is mainly written in C++ and the rest in C. MeshGems-Cleaner is currently available on 32-bit and 64-bit Windows and Linux architectures.

The executables have been designed to have minimal dependencies towards the Operating System.

The distribution provides different directories for Windows 7 and Windows XP platforms, but for MeshGems-Cleaner, the provided executables and libraries are actually identical.

2.2.1 Disk Space

The disk space requirement for this tool installation is about 43 MB.

2.2.2 CPU time

The CPU time required to obtain the final mesh (including the I/O) is related to the parameters and options specified. Nevertheless, one can reasonably expect a few minutes to process a mesh with several hundred thousand elements.

2.2.3 Memory usage

One can reasonably expect a memory usage about one million triangles per 250 Mb.

2.3 Launching the software

The name of the executable is mg-cleaner.exe whatever the platform.

Once the installation variables[1] and the execution path are set, MeshGems-Cleaner can be called by typing the command

```
mg-cleaner.exe
```

For more simplicity, we will use run_mg-cleaner as executable name in both cases.

If the command exits with a message such as the following:

```
Cannot check MG-CLEANER license
```

it means that one or more of the following is true:

1. the environment variables are incorrectly set;
2. the license is obsolete or invalid;
3. the license server is incorrectly installed.

The use of MeshGems-Cleaner and the other MeshGems® Suite components requires various license keys, depending on the activated features.

2.4 Errors and diagnostics

MeshGems-Cleaner has an error management which intercepts them and prints them on screen.

2.5 Coupling with other MeshGems® Suite components

MeshGems-Cleaner can be used efficiently with the following companion tools: MeshGems-Tetra, MeshGems-SurfOpt, MeshGems-Adapt (which will be available in a future release), MeshGems-CADSurf.

MeshGems-Cleaner has been initially designed as a **preprocessor to make surface meshes suitable for MeshGems-Tetra,** that is water-tight and not self-intersecting.

MeshGems-Cleaner may also be used as a **postprocessor of MeshGems-CADSurf or MeshGems-SurfOpt.** Indeed, depending on the quality and characteristics of their input, the MeshGems® Suite MeshGems-CADSurf or MeshGems-SurfOpt may produce occasionally meshes which may be unsuitable for MeshGems-Tetra.

MeshGems-Cleaner can also be used as a **preprocessor for MeshGems-SurfOpt or MeshGems-Adapt** (which will be available in a future release). Both try to keep all the geometrical structures they detect in their input. Because MeshGems-Cleaner improves the surface mesh quality and gets rid of unwanted small structures, numerical errors are avoided and confusing or unwanted small structures are reduced.

[1] The variable "DISTENE_LICENSE_FILE" is usually set through a script.

2.6 Integration within third party products

MeshGems-Cleaner is delivered as an executable and meshes come in and out of MeshGems-Cleaner as files, or as a library using the MeshGems® Suite common API.

Contact Distene SAS if you wish to embed MeshGems-Cleaner more tightly into your own software.

Chapter 3

MeshGems-Cleaner's principles

MeshGems-Cleaner scans the input mesh and detects several types of misfeatures such as overlaps, intersections, holes or gaps, ...

Once detected, it applies several corrective strategies in sequence and stops as soon as the problem has been corrected. If none of the strategies is successful, MeshGems-Cleaner proceeds with the other errors and will come back to this one again later.

Chapter 4

MeshGems-Cleaner's utilisation

Contents

4.1 Input Files 19
4.1.1 The mesh format 19
4.2 Output Files 20
4.3 Relationship with the physics and/or the CAD description 20
4.4 Invoking MeshGems-Cleaner 21
4.4.1 Running MeshGems-Cleaner 21
4.4.2 Mandatory options 21
4.4.3 Generic options 21
4.4.4 Module specific options and control parameters 23
4.4.5 Using MeshGems-Cleaner with its default parameter values 26
4.5 Detection of faults 26

4.1 Input Files

To describe the input surface mesh, the "mesh" file format is the main format used with the MeshGems® Suite. A "mesh" file can be written either in ASCII or binary format.

The several types of files used by MeshGems-Cleaner are briefly described below and in appendix B, 'File format description' on page 39.

The input surface mesh must be formed with triangles or quadrilaterals only. In the case of quadrilaterals, these will be momentarily converted to triangles to be able to be processed by MeshGems-Cleaner, but they will be converted back to the original quadrilaterals if they were left untouched by the cleaning process. Also, the attributes of the surface will be kept in the fixed mesh.

4.1.1 The mesh format

This format is composed of a single file, mybasename.mesh (ASCII) or mybasename.meshb (binary). This file contains the basic information needed to describe entirely the surface mesh. It is organized as a series of fields, identified by keywords. In addition to the vertex coordinates and the topology represented by a list of faces, it allows one to specify additional information such as faults detected in the mesh.

Refer to appendix B, 'File format description' on page 39 for a complete list of the keywords used by MeshGems-Cleaner.

4.2 Output Files

A single output file is produced at completion. This file, corresponding to the resulting surface mesh, is written using the `mesh` format.

The resulting mesh will be composed of triangles and untouched quadrilaterals only, and it may contain additional keywords depending on the runtime option selected.

Unless option `--mode check` has been used, this resulting mesh will have as many faults corrected as possible. It will conform to the given surface within the allowance given by the user or, if this tolerance has been left up to the program, computed automatically from the input data.

4.3 Relationship with the physics and/or the CAD description

Each entity (vertex, edge, face, tetrahedron, ...) carries an integer (often called «attribute» or «colour») which, depending on the runtime option selected, the MeshGems® Suite components preserve.

One defines an attribute integer to identify:

- a physical property such as material, boundary condition, ...
- a CAD group affiliation such as CAD patch, construction line, ...
- or a unique combination of properties.

These attributes are set in view of a particular treatment, and entities with the same attribute are usually processed the same way on the solver side. They can be sometimes also used for specific processing by some components of the MeshGems suite.

The user must simply set these attributes to suit his needs such that:

- two entities belonging to two different CAD patches must have two different attributes if one needs to keep track of which CAD patch entities belong to;
- two entities belonging to two different physical property sets must have two different attributes if one needs to keep track of which physical property set entities belong to;

This may need a pre/post treatment of the user data.

Advice: when possible, save CAD construction lines (surface normal discontinuities as ridges and surface curve changes as coloured edge lines) in the input mesh file as this helps rebuilding the geometrical information.

For example, attribute numbers on vertices can be used:

- to associate a certain number of variables to apply boundary conditions,
- to analyse the results corresponding to a particular attribute.

MeshGems-Cleaner preserves the attributes specified in the input file (vertex, edge and face attributes) unless the entity underwent a corrective operation.

Limitation: In a next version, a history of attribute modifications will be maintained that will help the user keep track even after corrective operations took place.

As can be easily understood, the numbering of the entities is affected by MeshGems-Cleaner's algorithms.

4.4 Invoking MeshGems-Cleaner and using Options

This section describes how to run MeshGems-Cleaner and the possible options and parameters needed to control the meshing process.

4.4.1 Running MeshGems-Cleaner

The usual way to start MeshGems-Cleaner is to type in the following line:

```
mg-cleaner.exe --in ⟨filein⟩ [⟨extra options⟩]
```

where:

- the parts between angle brackets are ⟨set by the user⟩;
- the parts between square brackets are [optional].

Moreover, the order in which the options are given may be altered at user's will, that is the following line will work as well:

```
mg-cleaner.exe [⟨extra options 1⟩] --in ⟨filein⟩ [⟨extra options 2⟩]
```

More precisely,

in the case of `.mesh` files, the syntax is the following:

```
mg-cleaner.exe --in ⟨my_case⟩.mesh [--out ⟨my_clean_case⟩.mesh] [⟨extra options⟩]
```

Although the program is mostly automatic, it requires a few options. The default options were chosen so as to comply with most of the realistic models.

4.4.2 Mandatory options

4.4.2.1 Choosing the input file: "`--in` ⟨file⟩" (MANDATORY)

This option tells the mesher the name of the input surface mesh, which should be in the `mesh` format.

4.4.3 Generic options

4.4.3.1 Setting the verbosity level: "`--verbose` ⟨level⟩"

This option (0 <= ⟨level⟩ <= 10) prints information on meshing steps. Default level is 3.

This allows the user to increase (higher ⟨level⟩ value) or decrease (lower ⟨level⟩ value) the amount of information that appears on the screen while MeshGems-Cleaner processes the mesh.

4.4.3.2 Setting the output file name: "`--out` ⟨file⟩"

This option tells to the mesher the name of the output mesh, which will be in the `mesh` format.

By default, the name of the output mesh will be the name of the input file with `_cleaner` inserted before the suffix.

4.4.3.3 Getting online help: "--help"

```
1
2     ==========================================
3     MG-CLEANER -- MeshGems 1.1-3 (January, 2013)
4     ==========================================
5
6         Distene SAS
7            Campus Teratec
8            2, rue de la Piquetterie
9            91680 Bruyeres le Chatel
10           FRANCE
11        Phone: +33(0)970-650-219   Fax: +33(0)169-269-033
12        EMail: <support@distene.com>
13
14        Running MG-CLEANER (Copyright 2004-2013 by Distene SAS)
15           date of run: 04-Mar-2013 AT 11:41:26
16           running on : Linux 3.4.28-2.20-desktop x86_64
17           using modules:
18                MeshGems-Core 1.1-3
19
20        MeshGems is a Registered Trademark of Distene SAS
21
22
23  ************* Cleaner --help   ******************
24
25
26 MG-CLEANER USAGE
27
28 Synopsis:
29     ../../bin/Linux_64/mg-cleaner.exe \
30     [--help] [--in <filein>] [--out <fileout>] [--mode <behaviour>] \
31     [--number_of_passes <number>] [--verbose <verblevel>] \
32     [--topology <behaviour>] [--min_hole_size <size>] \
33     [--tolerance_displacement <size>] [--resolution_length <size>] \
34     [--folding_angle <angle>] [--remesh_planes] \
35     [--overlap_distance <size>] [--overlap_angle <angle>]
36
37 Description:
38      Short option (if it exists)
39     /    Long option
40    |    /   Description
41    |   |   /
42    v   v  v
43
44       --help
45            gives this help.
46
47    -i --in <filein>
48            sets the input file.
49            (MANDATORY)
50
51    -o --out <fileout>
52            sets the output file.
53            If unset , _cleaner is appended to the input file basename.
54            Using an existing file is forbidden.
55            Using the same file as --in is forbidden.
56
57    -m --mode <behaviour>
58            sets the applicable operations.
59              'fix'  :analyses and fixes mesh with a cleaning procedure. Does not write
                    diagnostics into the output file.
60              'check':performs checks only (no fixing). Write diagnostics into the
                    output file
61            Default is 'fix'.
62
63    -N --number_of_passes <number>
64            sets the number of passes for the fix mode.
65              '1' : analyses and fixes mesh with a two pass cleaning procedure
66              '2' : analyses and fixes mesh with only the first stage of the cleaning
                    procedure
```

```
          Default is 2.

  -v --verbose <verblevel>
          sets the verbosity level.
          From 0 (no detail) to 10 (very detailed).
          Default is 3.

     --topology <behaviour>
          sets the applicable fixing operations.
            'ignore' : applies all fixing operations
            'respect': disables fixing operations which induce topology modifications
          Default is 'ignore'.

  -h --min_hole_size <size>
          sets the surface size threshold below which holes are filled.
          Default is not to fill holes.

  -t --tolerance_displacement <size>
          sets the displacement threshold below which modification is allowed.
          Unused in collision resolution.
          tolerance_displacement is set to resolution_length if it is lower.
          Default is computed from model.

  -s --resolution_length <size>
          sets the distance threshold above which 2 points are considered distinct.
          Sets the tolerance_displacement to 1/5 of this size.
          Default is computed from model.

  -a --folding_angle <angle>
          sets the threshold angle below which 2 connected triangles are considered
              overlapping.
          Reduce this value if model contains sharp angles below this threshold that
              must be kept.
          overlap_angle is set to this angle if it is higher.
          Default is 15 degrees.

  -p --remesh_planes
          inserts vertices on planes to improve mesh quality.
          May be useful for poor quality triangulations (eg. STL or DXF triangulations
              ).
          Default is not to mesh planes.

  -d --overlap_distance <size>
          sets the distance below which 2 unconnected triangles are considered
              overlapping.
          Reduce this value if too many overlaps are detected.
          Default is computed from model.

  -A --overlap_angle <angle>
          sets the angle below which 2 unconnected triangles are considered
              overlapping.
          folding_angle is set to this angle if it is lower.
          Default is 15 degrees.


================================================================================
                  MG-CLEANER -- MeshGems 1.1-3 (January, 2013)
       END OF SESSION - MG-CLEANER (Copyright 2004-2013 by Distene SAS)
         compiled Mar  1 2013 10:23:27 on Linux 2.6.18-6-amd64 x86_64
               MeshGems is Registered Trademark of Distene SAS
================================================================================
      ( Distene SAS
        Phone: +33(0)970-650-219     Fax: +33(0)169-269-033
        EMail: <support@distene.com> )
```

4.4.4 Module specific options and control parameters

4.4.4.1 Applicable operations: "--mode ⟨behaviour⟩" (default is to fix)

This option will activate or deactivate the mesh fixing. It handles the following values of ⟨behaviour⟩ parameter:

fix: analyses and fixes mesh with a cleaning procedure. It is the default value.
The output fixed mesh will be in the mesh format. There will be no added sections.

check: only performs detection of potential faults.
The output mesh will be written in the mesh ASCII format. It will contain the surface mesh only composed of triangles (that is, each quadrangles of the surface mesh are split into two triangles) and for each class of error, a new section containing the list of triangles potentially faulty or in an area of problem.

4.4.4.2 Number of passes for the cleaning procedure: "--number_of_passes ⟨int⟩" (default is to fix with two passes)

This option sets the number of correction passes performed when the fix mode is activated. The following values of ⟨int⟩ parameter are possible:

1: analyses and fixes mesh with only the first stage of the cleaning procedure.

2: analyses and fixes mesh with a two stage cleaning procedure. It is the default value.
The first stage is identical to the one pass fixing procedure above. However, as the second stage applies different strategies, it should improve the mesh more than the first stage procedure alone.

The output fixed mesh will be in the mesh format. There will be no added sections.

4.4.4.3 Preserving topology: "--topology ⟨behaviour⟩" (default is to modify topology)

This option may be used to prevent the mesh cleaning step from performing global corrections who are deemed to induce a topology change.

This option handles the following values of ⟨behaviour⟩ parameter:
ignore: allows changes to the topology when necessary for corrections and is the default value;
respect: disables corrections implying a topology change (only the resolution_length is kept in effect).

This option reduces considerably the spectrum of corrections that can be applied, so should be used with care.

Important note: even though this option aims at reducing significantly the topology changes, it does not guarantee that the topology will always be preserved. This can happen on some occasions when local corrections induce a local topology change.

4.4.4.4 Setting hole min size: "--min_hole_size ⟨size⟩" (default is 0 : not to fill holes)

This option activates automatic hole filling. If this option is not used, no holes will be filled. The size parameter gives the threshold above which the holes will not be filled.

If the option is activated, MeshGems-Cleaner will calculate the surface size S for each hole as $S = \sqrt{surface\ of\ the\ hole}$. If $S >$ ⟨size⟩ parameter, then the hole will not be filled.

Important note: only holes which are projectable on a plane can be filled with this option.

4.4.4.5 Setting tolerance parameter: "--tolerance_displacement ⟨size⟩" (default from the input surface)

The tolerance_displacement parameter specifies the maximum size of modification allowed. That is, no modification will be made if it implies a vertex displacement exceeding this value, unless this is to correct a collision (a vertex crossing a neighbouring surface).

It should be used with great care to avoid destroying the geometry.

The default value for tolerance_displacement parameter is computed from the input surface.

4.4.4.6 Setting resolution length: "--resolution_length ⟨size⟩" (default from the input surface)

This option sets the minimal distance between two vertices, or between one vertex and an edge. If two vertices are closer than this distance, they will be considered the same vertex and will be "glued" together. If one vertex is closer to an edge than this distance, the vertex will be considered part of the edge which will then in turn be split.

Important note: the tolerance induced by this option will take precedence over any parameter set by the user for the tolerance.
This implies that activating this parameter will set the tolerance value to a 1/5th of this new resolution size, if this new tolerance is smaller or equal to the default tolerance. This is necessary to prevent the tolerance from being too large compared to the resolution size.

The default value for resolution_length parameter is computed from the input surface.

4.4.4.7 Setting minimum angle: "--folding_angle ⟨size⟩" (default is 15 degrees)

This option sets the threshold angle below which two *connected* triangles will be considered as being overlapping each other.

See also options --overlap_distance and --overlap_angle.

The default value is set to 15 degrees.

4.4.4.8 Activating the remeshing of planar surface: "--remesh_planes" (default is to not remesh planes)

This option will activate remeshing of planar surfaces. This is only useful when dealing with poorly discretized surfaces where the number of elements was reduced very significantly (such as STL geometries). This option will generate more and better quality triangles on these automatically detected planes.

This option is not activated by default.

4.4.4.9 Setting the overlapping distance: "--overlap_distance ⟨size⟩" (default from the input surface)

This option gives another control on overlap detection. This sets the minimal distance between two *unconnected* triangles below which they will be considered as being overlapping.

You may reduce this value, depending on the input surface, if too many overlaps are detected.

The default value for overlap_distance parameter is computed from the input surface.

See also option --folding_angle.

4.4.4.10 Setting the overlapping angle: "--overlap_angle ⟨angle⟩" (default is 15 degrees)

This option sets the threshold angle below which two *unconnected* triangles will be considered as being overlapping each other.

See also options --overlap_distance and --folding_angle.

The default value is set to 15 degrees.

4.4.5 Using MeshGems-Cleaner with its default parameter values

By default, the sole option that needs to be specified is the name of input data file:

```
mg-cleaner.exe --in (filein)
```

4.5 Detection of faults

This section describes the potential faults detected and saved by MeshGems-Cleaner when used together with the --mode check option.

MeshGems-Cleaner optionally performs checks on the surface mesh to detect potential problems. The resulting areas are gathered as sets of triangles grouped by fault category, and are added to the surface mesh only composed of triangles (that is to say, all quadrangles of the surface mesh are split into two triangles)

The produced mesh file contains new keywords (refer to appendix B), which can be used by the embedding environment to visualize these potential faults.

Note that some problems may be highlighted by several different ways, so the same triangles may appear in several categories (eg, a hanging triangle on the surface may appear as "overlap", "near tri" and "free edge").

The categories of faults are:

SmallTri This highlights very small sized triangles. These may induce hard configurations for tet meshers, or very small simulation time steps.

BadShape This highlights narrow triangles. As for small triangles, these may induce hard configurations for tet meshers, and both very small simulation time steps and precision issues.

Overlap This highlights triangles which overlap. This particularly detects similar triangles on top of one another. General overlap areas are also detected by **NearTri** below.

FreeEdge This highlights free edges. This may not always be a real fault (in case of hanging surfaces for example), but there should not be any free edges on an outer surface.

Inter This highlights intersections.

NearTri This highlights triangles which are very close without being connected. This may also be an artifact of overlapping areas.

Chapter 5

MeshGems-Cleaner's examples of use

Contents

5.1 Example 1: Cleaning to get a volume mesh 27
5.1.1 Mesh checking 27
5.1.2 Correcting the mesh with standard options 30
5.2 Example 2: Cleaning to remesh and get a volume mesh 31
5.2.1 Mesh checking 31
5.2.2 Fixing the mesh 33
5.2.3 Producing a surface mesh with better aspect ratio of triangles 33
5.2.4 Improving mesh quality with MeshGems-Cleaner 34

5.1 Example 1: Cleaning to get a volume mesh

This section gives a typical example of what MeshGems-Cleaner can achieve. This is what is obtained on the case7 test case, provided in the installation.
The case7 test case uses the mesh format.

5.1.1 Mesh checking

Copy the file "case7.mesh" provided in the Examples directory of the distribution to your local work directory, then type:

```
mg-cleaner.exe --in case7.mesh --out case7-test.mesh --mode check
```

this will perform checks on the surface mesh and detect potential problems. The resulting areas are gathered as a set of triangles.
The produced .mesh file indeed contains new keywords, which can be used by the embedding environment to visualize these potential faults.

The two meshes, the initial surface mesh, and the mesh produced are shown in the pictures below (the embedding environment used here is Ensight, by CEI).

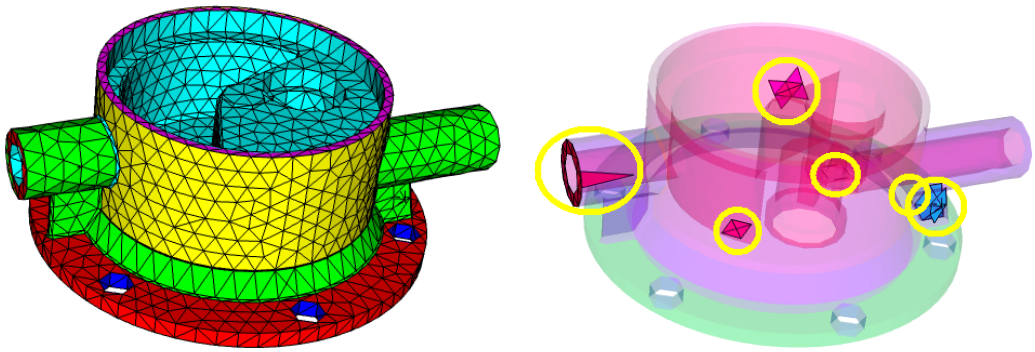
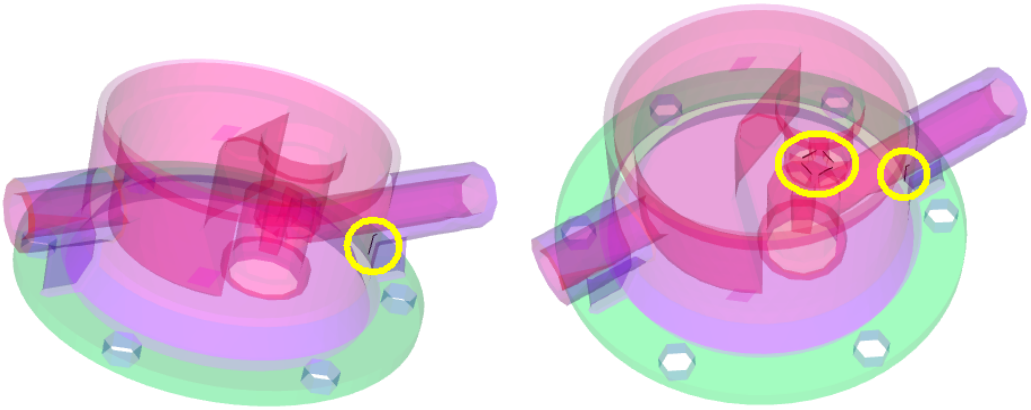
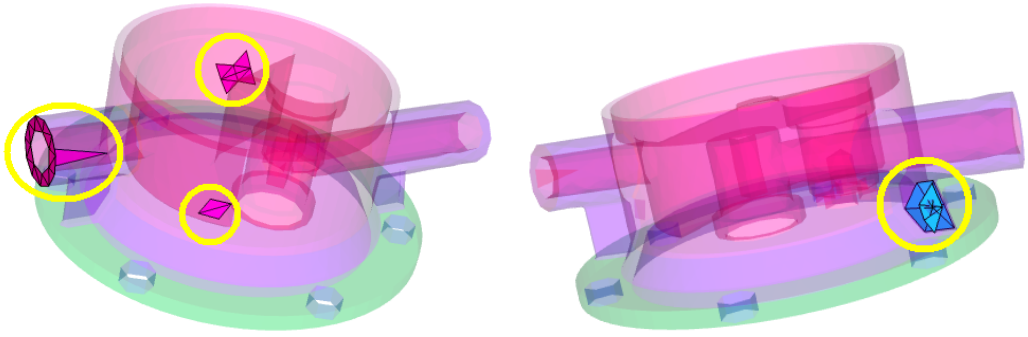


Figure 5.1: Original surface mesh (left) and "Checked" mesh (right)



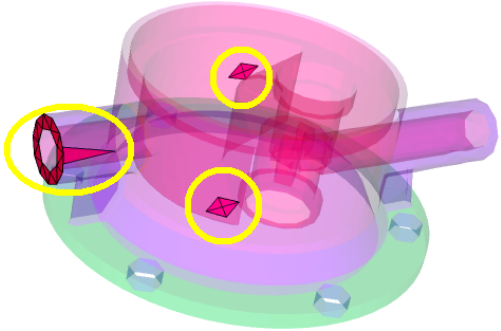
Small triangle

Bad shape



Free edges

Intersections



Near triangles (or overlaps)

Figure 5.2: Detected faults

This run will produce the following output in the console window:

```
   ============================================
    MG-CLEANER -- MeshGems 1.1-2 (January, 2013)
    ============================================

        Distene SAS
           Campus Teratec
           2, rue de la Piquetterie
           91680 Bruyeres le Chatel
           FRANCE
        Phone: +33(0)970-650-219   Fax: +33(0)169-269-033
        EMail: <support@distene.com>

        Running MG-CLEANER (Copyright 2004-2013 by Distene SAS)
           date of run: 08-Jan-2013 AT 15:13:08
           running on : Linux 2.6.37.6-24-desktop x86_64
           using modules:
                MeshGems-Core 1.1-2

        MeshGems is a Registered Trademark of Distene SAS

 ************* Cleaner --in case7.mesh --out case7-test.mesh --mode check
     *****************

 *** INPUT FILE    case7.mesh   ***

Warning, you only have 23 days left to use MeshGems-Cleaner version 1.1
  Cleaning parameters: sizeMin  0.500000  tolGeom 0.100000 sizeHole 0.000000
      SewingSize 1.000000
                       ovAngles 15.000000 15.000000 ovDist 0.500000

  nb_Tri: 4475  smallEdge average size 18.675639 smallest size 0.098646
  colliding triangles   74  near triangles 139
   Worst ratio  228.356140 badTri 2177

  nb FreeContours: 5  nb Overlaps 0 nb SmallSize 2  nb BadRatio 6


 **** CPU time  :         0.3

================================================================================
                 MG-CLEANER -- MeshGems 1.1-2 (January, 2013)
       END OF SESSION - MG-CLEANER (Copyright 2004-2013 by Distene SAS)
         compiled Jan  7 2013 15:45:02 on Linux 2.6.18-6-amd64 x86_64
               MeshGems is Registered Trademark of Distene SAS
================================================================================
      ( Distene SAS
        Phone: +33(0)970-650-219     Fax: +33(0)169-269-033
        EMail: <support@distene.com> )
```

and the "case7-test.mesh" output file.

Let's see all detected faults one by one (see figure 5.2):

1. Small tri : there are very small elements in the mesh.
2. Bad shape : several bad shaped elements are detected.
3. Free edges: a surface is overlapping another surface on an inlet, and is detected as free edge. At the same time, some overlapping triangles appear on the main body.
4. Intersections : the inlet is detected as intersection because some vertices of the overlapping triangles are very close to other triangles. At another location (on the right), a point went through the surface to emerge on the other side.
5. Near tri : overlaps are also detected because some triangles are very close to others, without having more than one point in common.

5.1.2 Correcting the mesh with standard options

Using the same case as above, type this time :

```
mg-cleaner.exe case7.mesh case7-fix.mesh --mode fix
```

This will produce a corrected mesh, without the additional analysis keywords. Note that the mesh attributes are preserved.

The two meshes, the initial surface mesh, and the corrected mesh produced are shown in the pictures below:

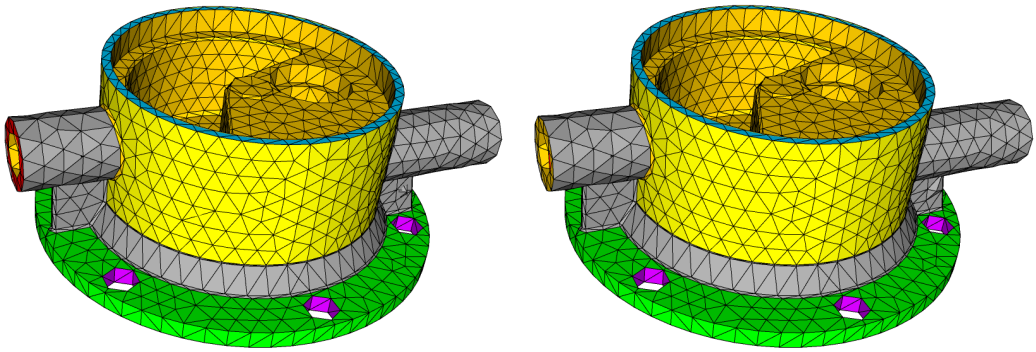
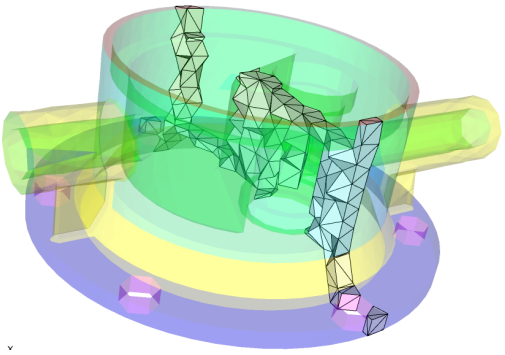


Figure 5.3: Original surface mesh (left) and corrected mesh (right)

Before correcting the mesh with MeshGems-Cleaner, MeshGems-Tetra cannot produce a volume mesh because the initial mesh contains a triangle with two identical vertices. With the corrected mesh, we are now able to launch the tet mesher MeshGems-Tetra to produce a tetrahedral mesh:

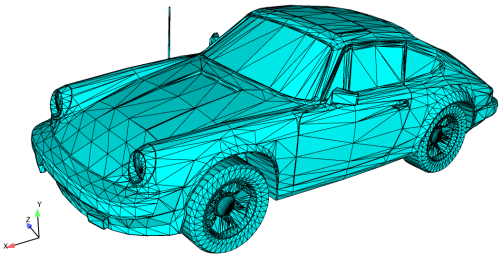


5.2 Example 2: Cleaning to remesh and get a volume mesh

This section gives another example of what MeshGems-Cleaner can do. We will use it as a correction tool prior to performing surface remeshing, then eventually volume meshing. We will also use it as a means to improve the quality of the surface mesh.

5.2.1 Mesh checking

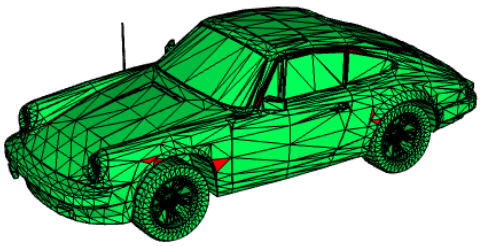
The mesh represents the geometry of a car, generated by a tessellation program, and contains many errors.



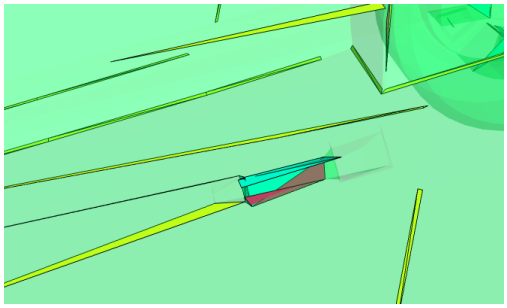
We will first start with the error checking:

```
mg-cleaner.exe --in Porsche.mesh --out Porsche-test.mesh --mode check
```

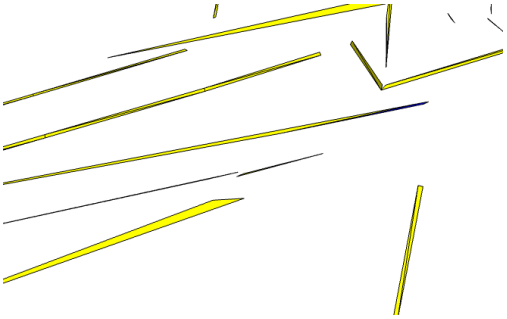
This is what we get:



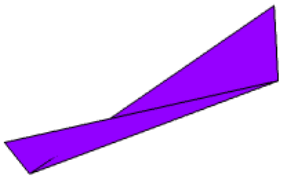
Let's analyze what is happening near the door handle on the front left of the car:



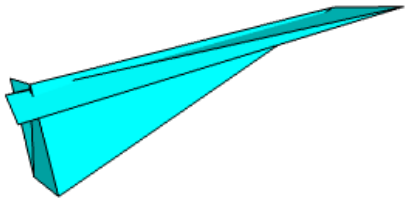
There are several problems. We will now visualize each in turn, at the same location (left door handle).



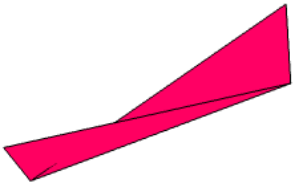
Fault_SmallTri (in dark blue) and Fault_BadShape (in yellow)



Fault_Overlap (in purple)



Fault_Inter (in light blue)



Fault_NearTri (in red)

1. badly shaped elements (very narrow triangles) or very small triangles. This is highlighted with the categories Fault_SmallTri (in dark blue) and Fault_BadShape (in yellow).

2. overlapping triangles. This happens when two or more triangles are overlapping, with shared vertex connections . This is highlighted with category Fault_Overlap (in purple) as shown below. We see one triangle on top of one another.
   The overlapping detection has here a default threshold value of 15 degrees.

3. Intersections : This happens when two or more triangles intersect. This is highlighted with category Fault_Inter (in light blue) as shown below. We see two triangles intersecting each other.

4. Triangles too close : This happens when one vertex of a triangle is almost contained in another triangle. This is highlighted with category Fault_NearTri (in red) as shown below. We see two triangles too close to one another (to the point of intersection).

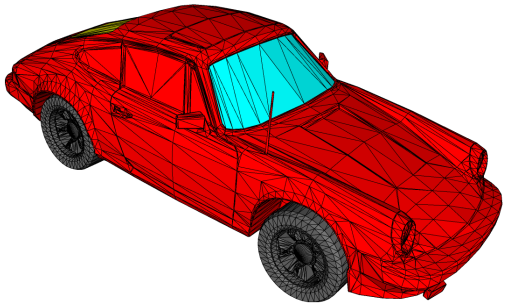
5.2.2 Fixing the mesh

Let's now try to correct the mesh above, typing this time :

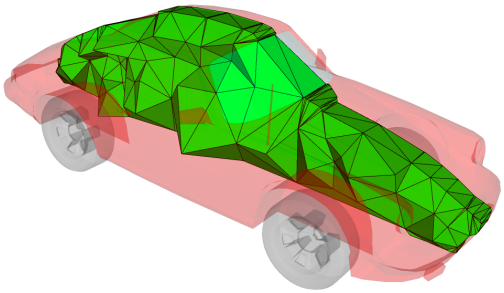
```
mg-cleaner.exe --in Porsche.mesh --out Porschefix.mesh --mode fix
```

This will produce a corrected mesh, without the additional analysis keywords. Noe that the mesh attributes are preserved.

Here is the corrected mesh:



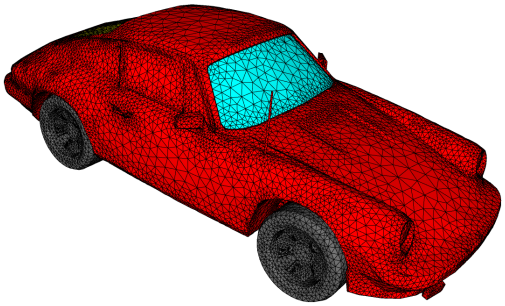
Before correcting the mesh with MeshGems-Cleaner, MeshGems-Tetra cannot produce a volume mesh because a surface edge intersects a surface face in the initial mesh. With the corrected mesh, we are now able to run the tet mesher to produce a tetrahedral mesh from the surface, using for example MeshGems-Tetra:



Note that MeshGems-Cleaner cannot guarantee that it will be able to produce a fixed mesh correcting all faults, especially in the case of complex assemblies.

5.2.3 Producing a surface mesh with better aspect ratio of triangles

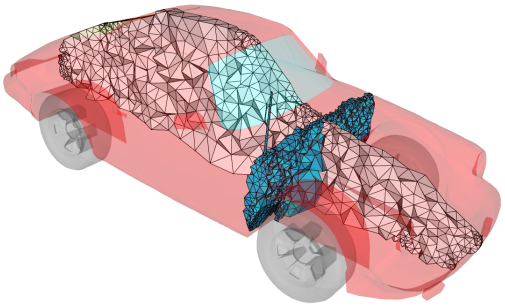
In our example, even if corrected and valid, the mesh is not suitable for classical finite elements computations because of the high aspect ratios of triangles. This can be improved with other tools, such as MeshGems-SurfOpt.
When running MeshGems-SurfOpt on the surface mesh above to produce better triangles, this is what one obtains:



5.2.4 Improving mesh quality with MeshGems-Cleaner

Even though this case can be meshed in volume, the quality of the worst element is quite poor. Such elements may occur when the input geometry is very "noisy", generating artificial geometrical constraints for MeshGems-SurfOpt. These cannot be corrected by MeshGems-SurfOpt because it always conforms to the input geometry.
These elements cannot be removed without altering the geometry, so we will use the mesh fixing component to achieve this goal.
For this, we will correct the mesh created by MeshGems-SurfOpt with MeshGems-Cleaner using for example the option `--resolution_length 0.03`.
This will collapse all vertices closer than the given threshold size. A "good" value for this threshold size may be hinted from the mesh fixing analysis. In this case, the minimal size computed by the mesh fixing analysis (as shown in the standard output) is 0.003. Enlarging this size by a factor of 10 may be a good guess (hence the 0.03 value above).
In this example, the worst quality is improved by a factor of 10 in one run of MeshGems-Cleaner.
Note that changing these values may very well destroy the geometry, so one must be careful with these values. In particular, small details might be lost!

Once the clean-up is performed, we can produce the tetrahedral mesh using MeshGems-Tetra:



Appendices

Appendix A

MeshGems-Cleaner's diagnostics

MeshGems-Cleaner intercepts and prints errors/warnings on screen.

You may get some help about valid options and values by typing:

```
mg-cleaner.exe --help
```

Only some errors and possible cause of failure are identified and listed below, the rest of the messages are informative messages and should be self explanatory.

A.1 Syntax of the exe diagnostics

During the execution of MeshGems-Cleaner, some messages may be printed. Each message is composed of a code number, a description and sometimes attached data (typically a list of CAD or mesh entities: vertices, edges...).

They are classified into three groups :

- the error messages with a negative code number,
- the warning/information messages with a positive code number,
- the unstructured text messages with a null code number,

A non null message code number can be written as follows : code $= \pm(1000000 \times \mathbf{q} + \mathbf{r})$,
where $0 \leq r < 1000000$

Where **q** identifies the module and **r** identifies the diagnostic independently of the module and defines the base number of the message. For a diagnostic coming from MeshGems-Cleaner, **q** is equal to 5.

In order to easily parse the messages with a non null code number, a string beginning by **MGMESSAGE** is printed. We give here the generic syntax of a printed error :

```
MGMESSAGE  code n_idata idata_1 ... idata_n_idata n_rdata rdata_1 ... rdata_n_rdata
```

where :

- MGMESSAGE is an identifier indicating a parsable string
- code is the code number of the printed message
- n_idata is the number of integer data attached to the message
- idata_1 is the first integer data attached

- idata_2 is the second integer data attached
- ...
- idata_n_idata is the last integer data attached
- n_rdata is the number of real data attached to the message
- rdata_1 is the first real data attached
- rdata_2 is the second real data attached
- ...
- rdata_n_rdata is the last real data attached

A.2 List of MeshGems-Cleaner's diagnostics

For the time being, an error message of MeshGems-Cleaner is composed of a description only. The error code number is indeed not available yet.

The list of available diagnostics appears below, whenever possible, a way of correcting the problem is also provided.

- ERROR: required argument not found for option '–(string)'
- ERROR: Missing input file.
- ERROR: Unable to create output file name.
  cure : Check if the destination directory is writable.
- ERROR: Missing output file.
  cure : As the output file name is computed by default, this means there has been a memory problem.
- ERROR: file not found or empty '(string)'
- ERROR: No input file...
- ERROR: Incorrect value (string) for option (string)
- ERROR: Unknown option (string)
- ERROR: Unable to write check file binary. The output file must be in ASCII mode.
  cure : Check that the output file is in ASCII mode.
- ERROR : in file and out file are identical.
  cure : Give a different name to your output file
- ERROR : out mesh file exists.
  cure : Delete your output mesh file or change its name thans to the option `--out`.

Appendix B

File format description

In this chapter, we will briefly describe two types of data files used with MeshGems-Cleaner:

- the surface mesh formats are:
  - "`.mesh`";
- the size map format is:
  - "`.sol`".

> **Important notice:** For a complete description of these formats as well as the complete list of the keywords used by the MeshGems® Suite suite refer to the documentation describing the mesh format. This document can be found in the Docs sub-directory of the distribution.

B.1 the "`.mesh`" format

A "`.mesh`" file can be written either in Ascii or binary format.

B.1.1 Structure

Unlike the other file description formats, the "`.mesh`" format is a flexible structure which is composed of optional fields. We will not describe the structure *in extenso* here, but only list the main fields used by the MeshGems-Cleaner.

B.1.2 Sample

```
1  MeshVersionFormatted 1
2
3  Dimension
4  3
5
6  # Set of mesh vertices
7  Vertices
8  682
9  94.444450378418 94.4444427490234 -3.99593202971005e-09 0
10 89.3760375976562 84.1671295166016 -2.91678103891968e-09 0
11 84.1671295166016 89.3760375976562 -3.46373307813508e-09 0
```

```
1
2  # Set of mesh triangles (v1,v2,v3,tag)
```

```
3 Triangles
4 1364
5 1 2 3 1
6 4 2 1 1
7 5 6 7 1
```

B.2 the ".sol" format

A ".sol" file can be written either in ASCII or binary format. This ASCII file contains scalar or tensor values associated with a mesh file. As MeshGems-Cleaner output, it contains the sizes at vertices.

B.2.1 Structure

The .sol format is very close to the .mesh format. The first field is identical to the .mesh format.

The keyword SolAtVertices is however slightly more complex.

The first integer which follows the keyword indicates the number of lines related to the keyword (eg, number of entity sizes provided) The second and third integers are not used here and must be both set to 1 in the case of MeshGems-Cleaner.

Then comes the list of sizes associated to the entities of the supporting mesh (in the case of MeshGems-Cleaner, these entities are vertices). This list follows the list of vertices given in a corresponding .mesh file.

B.2.2 Sample

```
1  MeshVersionFormatted 1
2
3  Dimension 3
4
5  SolAtVertices
6  166
7  1 1
8
9  7.1996
10 6.17712
11 7.21029
12 6.19397
13 7.22313
14 6.19397
15 7.21029
```

...

```
85  10.587
86  7.07913
87  10.5525
88  10.5525
89  10.5208
90  12.4216
91  14.4762
92  12.4216
93  10.587
94  7.07913
95  10.5525
96  10.5525
97  10.5208
98  12.4216
99  10.587
100 7.07913
```

...

```
6.95805
6.95805
6.9673
6.95805
6.95805

End
```

B.3 Keywords used by MeshGems-Cleaner

- MeshVersionFormatted
- Dimension
- Vertices
- Edges
- RequiredEdges
- Triangles
- Ridges
- Corners
- Fault_SmallTri
- Fault_BadShape
- Fault_Overlap
- Fault_FreeEdge
- Fault_Inter
- Fault_NearTri

Appendix C

List of FAQ's

Contents

C.1 My mesh contains quads, is MeshGems-Cleaner capable of correcting these? 43
C.2 Is it a good idea to iterate MeshGems-Cleaner on itself? 43
C.3 How do I improve the quality of triangles? 44
C.4 Is MeshGems-Cleaner capable of guaranteeing a corrected mesh suitable for a Delaunay or front advancing tet mesher? 44
C.5 How is associativity taken care of ? 44
C.6 Can my geometry be altered ? 44
C.7 How can I preserve as much as possible my input topology ? 44
C.8 Does a specific treatment exist for planes? 44

In this section, we will attempt to answer some frequently asked questions about the MeshGems-Cleaner product.

C.1 My mesh contains quads, is MeshGems-Cleaner capable of correcting these?

In theory, no. However, MeshGems-Cleaner is capable of processing quad meshes. Quads are indeed split into two triangles automatically once read by MeshGems-Cleaner, so that it can process the mesh and clean it up. A pointer is kept to be able to revert these split quads into their original form, once the cleaning process is over, unless some of their triangles were removed or swapped. Attributes are also reverted to the original in the process

C.2 Is it a good idea to iterate MeshGems-Cleaner on itself?

It may prove useful on some cases, but in general this does not help so much because failures are often caused by very complex geometries or assemblies. One can think of using the `--number_of_passes 2` (default) and `--number_of_passes 1` options alternatively, together with `--mode fix`. These options indeed generate a different result, and may prove more useful than to iterate MeshGems-Cleaner on itself.

Refer to 4.4.4.2 for more information about the default option `--number_of_passes` .

C.3 How do I improve the quality of triangles?

One can use the `--resolution_length` ⟨size⟩ parameter to force the removal of bad quality triangles, or use MeshGems-SurfOpt to improve the surface mesh quality. While MeshGems-SurfOpt will always conform to the geometry and thus may be limited in its scope of mesh improvement, MeshGems-Cleaner is less limited than MeshGems-SurfOpt, even though there are some geometry accuracy considerations taken into account.

C.4 Is MeshGems-Cleaner capable of guaranteeing a corrected mesh suitable for a Delaunay or front advancing tet mesher?

Unfortunately no, as the current approach tries to conform as much as possible to the original geometry. There are cases where the geometry is composed of a heap of intersecting triangles which cannot even define a geometry.

C.5 How is associativity taken care of ?

Since MeshGems-Cleaner only works on triangulated surfaces, there is no direct associativity with the CAD system. However, since MeshGems-Cleaner preserves attributes of vertices, elements and since release 1.1, edges, it is possible to keep track of modified items throughout the meshing process from CAD. One can then identify where the corrected mesh entities are originating from

C.6 Can my geometry be altered ?

MeshGems-Cleaner may indeed modify the geometry, as it aims at correcting several issues preventing the tet mesher from succeeding. However , it contains a failsafe built-in automatic parameter set-up which prevents the geometry to be altered too significantly, as long as the automatically computed default parameters are used.

C.7 How can I preserve as much as possible my input topology ?

MeshGems-Cleaner provides an option `--topology` respect[1] to prevent topology changes. This however reduces significantly the amount of issues which can be corrected successfully by MeshGems-Cleaner.

C.8 Does a specific treatment exist for planes?

Yes, MeshGems-Cleaner provides an option `--remesh_planes` which generates more and better quality triangles on these automatically detected planes.

[1] Was previously `-T`.

Appendix D

Release Note

D.1 Version 1.1

D.1.1 New features

- This version is the first version which contains the MeshGems API

D.1.2 Behaviour changes

- Change of the default behaviour of MeshGems-Cleaner concerning the number of correction passes performed. The default behaviour is now to analyse and fix the mesh with a two pass cleaning procedure.

D.1.3 Improvements

- The mesh binary format is now supported for input and output (except for the check mode).

D.1.4 Corrections

- Correction in the check mode when the input surface mesh contains quadrangles. The result output is now only composed of triangles (quadrangles are split into two triangles) because the element number in the keyword of the check mode refers to triangles and not to quadrangles.
- Minor corrections

References

[1] Boissonnat, Jean-Daniel, Yvinec, and Mariette. *Géométrie Algorithmique*. Collection informatique. Ediscience international, Paris, 1995.

[2] Distene SAS. *MeshGems-SurfOpt: The automatic surface remeshing tool of the MeshGems® Suite*, January 2013.

[3] Distene SAS. *MeshGems-Tetra: The automatic tetrahedral mesh generator of the MeshGems® Suite*, January 2013.

[4] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, August 2007.

List of Figures

5.1 Original surface mesh (left) and "Checked" mesh (right) 28
5.2 Detected faults 28
5.3 Original surface mesh (left) and corrected mesh (right) 30