



MeshGems

**MeshGems-SurfOpt: The automatic surface remeshing
tool of the MeshGems[®] Suite**

User Manual

MeshGems-SurfOpt Version: 1.1

Documentation Revision: 1.1

February 2013



Customer Service

Should you encounter any problem with MeshGems-SurfOpt 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

MeshGems[®]: a brand of DISTENE

Copyright © Distene SAS 2013

Reproduction of this document is permitted except for resale and on condition its origin is specified. It may not be changed in any way without the prior agreement of the authors.

1st edition, February 2013

Information

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

The MeshGems-SurfOpt software is the automatic surface remeshing tool of the MeshGems[®] suite.

Important notice: In order to use MeshGems-SurfOpt, 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	February 2013	1 st 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 optimisation
- 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-SurfOpt with the other MeshGems[®] Suite components and section 4.3 for more information about MeshGems-SurfOpt relationship with the physics and/or the CAD description.



MeshGems

User Manual

MG-SURFOPT

Release 1.1

Contents

Foreword	5
Contents	7
1 Introduction	13
2 Getting started	15
2.1 Installation	15
2.2 Requirements	15
2.2.1 Supported platforms	15
2.2.2 Disk Space	15
2.2.3 CPU time	16
2.2.4 Memory usage	16
2.3 Launching the software	16
2.4 Errors and diagnostics	16
2.5 Coupling with other MeshGems [®] Suite tools	17
2.6 Integration within third party products	17
3 MeshGems-SurfOpt's principles	19
3.1 Some definitions	19
3.1.1 Triangle quality ("aspect ratio")	19
3.1.2 Mesh quality	19
3.1.3 Normalised length	20
3.1.4 Efficiency index	20
3.1.5 Chordal deviation	21
3.1.6 Geometric approximation	21
3.2 The algorithm	21
3.2.1 Flowchart	21
4 MeshGems-SurfOpt utilisation	23
4.1 Input Files	23

4.1.1	Geometry description	23
4.1.2	Metric specification	24
4.2	Output Files	24
4.3	Relationship with the physics and/or the CAD description	25
4.4	Invoking MeshGems-SurfOpt	25
4.4.1	Running MeshGems-SurfOpt	25
4.4.2	Mandatory options	26
4.4.2.1	Name of the input surface mesh	26
4.4.3	Generic options	26
4.4.3.1	Getting online help	26
4.4.3.2	Verbose mode	30
4.4.3.3	Deprecated error mode	30
4.4.3.4	Backward compatibility	30
4.4.4	Memory usage	31
4.4.4.1	Memory usage	31
4.4.5	I/O options and formatting data options	32
4.4.5.1	Setting the input mesh file name	32
4.4.5.2	Setting the output mesh file name	32
4.4.5.3	Setting size map to be read	32
4.4.5.4	Setting size map to be written	32
4.4.6	Remeshing options and control parameters	33
4.4.6.1	Coarsening, enrichment & optimisation options	33
4.4.6.1.1	Disabling enrichment	33
4.4.6.1.2	Disabling optimisation	33
4.4.6.1.3	Using Hausdorff-like sizemap	33
4.4.6.2	Activating the sand papering algorithm	33
4.4.6.3	Uniform subdivision	33
4.4.6.3.1	Activating the uniform subdivision algorithm	33
4.4.6.3.2	Setting the uniform subdivision level	33
4.4.6.4	Setting size & surface parameters	34
4.4.6.4.1	Minimum element's size	34
4.4.6.4.2	Maximum element's size	34
4.4.6.4.3	Gradation	34
4.4.6.4.4	Metric type	34
4.4.6.4.5	Chordal error	34
4.4.6.4.6	Geometric approximation	35

4.4.6.4.7	Assume non-linear size on edges	35
4.4.6.5	Computing ridges	36
4.4.6.5.1	Setting the ridge angle	36
4.4.6.5.2	Restricting ridge detection to reference lines	36
4.4.6.6	Setting surface & volume proximity parameters	36
4.4.6.6.1	Activating surface proximity	36
4.4.6.6.2	Setting the surface proximity layers	37
4.4.6.6.3	Setting the surface proximity gradation	37
4.4.6.6.4	Activating volume proximity	37
4.4.6.6.5	Setting the volume proximity layers	37
4.4.6.6.6	Setting the volume proximity gradation	37
4.4.6.7	Setting the element type	37
4.4.6.8	Setting the element order	38
4.4.6.9	Create nodes on flat edge	38
4.4.6.10	Setting the tags behavior	38
4.4.6.11	Targeting a vertex count	39
4.4.6.12	Cleaning input surface mesh	39
4.4.6.12.1	Removing input bad element	39
4.4.6.12.2	Setting quality threshold	39
4.4.6.13	Splitting over-constrained element	39
4.4.7	Using MeshGems-SurfOpt with its default parameter values	39
5	MeshGems-SurfOpt's examples of use	41
5.1	First example	41
5.1.1	Optimisation and coarsening	41
5.1.1.1	Using MeshGems-SurfOpt with its default parameter values	41
5.1.1.2	Using ridge angle and maximal chordal deviation	41
5.1.2	Optimisation, coarsening and enrichment	44
5.1.2.1	Using MeshGems-SurfOpt with its default parameter values	44
5.1.2.2	Using ridge angle and maximal chordal deviation	44
5.2	Surface simplification	46
5.2.1	Curvature-controlled coarsening	46
5.2.2	Distance controlled coarsening	46
5.3	Surface mesh enrichment	48
5.4	Surface mesh optimisation	48

Appendices	51
A MeshGems-SurfOpt’s diagnostics	51
A.1 Syntax of the exe diagnostics	51
A.2 List of MeshGems-SurfOpt’s diagnostics	52
A.2.1 I/O errors	52
A.2.2 Data provided incorrect	53
A.2.3 Memory	53
A.2.4 Data provided not suitable for module	54
A.2.5 Analysis	54
A.2.6 Licensing	55
A.2.7 Module	55
A.2.8 Result	56
A.2.9 Statistics	56
B File format description	57
B.1 The “ mesh ” format	57
B.1.1 Structure	57
B.1.2 Sample	57
B.2 The “ sol ” format	58
B.2.1 Structure	58
B.2.2 Sample	58
B.3 Keywords used by MeshGems-SurfOpt	59
C List of FAQ’s	61
C.1 How do I get a mesh suitable for computations?	61
C.2 How do I reduce the size of a large mesh?	61
C.3 How can I avoid node relocation?	61
C.4 How can I preserve some specific mesh entities?	62
C.5 How do I improve noisy data?	62
C.6 How can I remove small features?	62
C.7 How do I proceed with scanned data?	62
C.8 How can I control rapid size variations?	62
C.9 How do I visualize surface meshes?	63
C.10 What should I do about MeshGems-SurfOpt curve warnings?	63
C.11 How can I control ridge detection?	63

C.12 How can I ensure a surface size on a plane, for example for an imposed flow rate boundary condition?	63
D List of renamed options	65
E Release Notes	67
E.1 Version 1.1	67
E.1.1 New features	67
E.1.2 Behavior changes	67
E.1.3 Improvements	68
E.1.4 Corrections	68
F References	69



MeshGems

User Manual

MG-SURFOPT

Release 1.1

CONTENTS

Chapter 1

Introduction

This user manual describes the main features of MeshGems-SurfOpt¹, an automatic surface remeshing component.

MeshGems-SurfOpt is a *fully automatic adaptive surface remeshing tool* which allows one to obtain a governed surface mesh from an arbitrary given surface triangulation. In addition, the adaptive MeshGems-SurfOpt software is governed by both the geometric surface properties (geometric size map) and optionally by a *metric map*, defined at the vertices of the input mesh. This map provides the desired sizes of the elements.

The initial surface triangulation can be any triangulation, for instance supplied by a CAD-CAM package or any surface reconstruction algorithm. The only requirement for the input data is to contain triangles and/or quadrilaterals (in particular non-manifold models can be processed by MeshGems-SurfOpt). Unlike previous versions, this version of Yams is also now capable of processing non orientable surfaces² by inserting ridges to remove ambiguities.

The surface remeshing algorithm gives another surface mesh as output, but for which the element density and the mesh quality are controlled by the user through the input data. If desired, the algorithm can also be controlled such that the distance between the input and output surfaces is bound (see tolerance in §4.4.6.4.5).

The geometric surface properties may include additional entities and information that will be used to control and drive the surface remeshing (such as, for example: constrained entities, corners, ridges³, normals, etc.).

MeshGems-SurfOpt also produces diagnostics about the correctness of the surface (regarding its topology) and histograms (regarding the quality of the approximation and the edge lengths).

The input and output files, the interaction with the user, and the list of error messages are defined in this document. Several typical application examples are provided to explain the various possibilities of the code. Finally, a short technical description of the algorithm is given. Several appendices describe the input/output formats and other useful information.

¹Originally developed by the Gamma project team of Inria, set to industrial standards, developed and distributed now by Distene SAS

²For instance Moebius rings are not orientable.

³Also called “feature lines”.



MeshGems

User Manual

MG-SURFOPT

Release 1.1

CHAPTER 1. INTRODUCTION

Chapter 2

Getting started

Contents

2.1	Installation	15
2.2	Requirements	15
2.2.1	Supported platforms	15
2.2.2	Disk Space	15
2.2.3	CPU time	16
2.2.4	Memory usage	16
2.3	Launching the software	16
2.4	Errors and diagnostics	16
2.5	Coupling with other MeshGems[®] Suite tools	17
2.6	Integration within third party products	17

2.1 Installation

Please refer to the installation guide which describes the Tool 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

2.2.1 Supported platforms

MeshGems-SurfOpt is currently available on Windows and Linux architectures, both 32 and 64 bits.

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

2.2.2 Disk Space

The disk space requirement for this tool installation is about 32 Mb.



2.2.3 CPU time

The CPU time required to obtain the final mesh (including the I/O) is obviously related to the parameters and options specified as well as to the size of the initial triangulation. Specifying a metric map as input of MeshGems-SurfOpt may increase the computing time quite considerably. Nevertheless, one can reasonably expect to complete MeshGems-SurfOpt:

- a few seconds to simplify a mesh of several thousands of elements and
- a few minutes to enrich a mesh of several hundreds of thousands of elements.

2.2.4 Memory usage

MeshGems-SurfOpt allocates automatically the memory it needs. The user can also specify a maximal memory allocation.

See §4.4.4, ‘Memory usage’ on page 31, for a complete description of the memory management option.

2.3 Launching the software

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

Once the installation variables¹ and the execution path are set, MeshGems-SurfOpt can be called by typing the command:

```
mg-surfopt.exe
```

followed by various command line options which are described in chapter 4.

The use of the other meshing components requires various license keys.

You can get additional information by reading the section “Distene Tool Use” in the installation manual available on the delivery media or located in the installed distribution under “`.../DISTENE/Docs/`”, in particular in file “`05-use-tool.html`”.

2.4 Errors and diagnostics

At completion of the procedure, two cases may happen:

- a “`xxx.mesh`” (or “`xxx.meshb`”) file was created : the resulting mesh is most probably correct. Check for warning messages in the output.
- no output file was produced: either the data are incorrect (or missing), or the algorithm detected a unrecoverable error and stopped, writing an error code (see below).

In case of failure, MeshGems-SurfOpt provides error messages which help detect and correct the cause of failure.

The list of these errors is provided in appendix A together with recommended correcting actions when available. Their description allows you to correct and to restart the execution.

¹The variable “`DISTENE_LICENSE_FILE`” is usually set through a script.

2.5 Coupling with other MeshGems[®] Suite tools

All MeshGems[®] Suite tools can be easily coupled with each other:

- the executable versions share the same file formats;
- the library versions can be coupled using the MeshGems[®] Suite common API.

MeshGems-SurfOpt can be used efficiently with the following companion tools:

- MeshGems-Tetra:
MeshGems-SurfOpt has been initially designed to **remesh surface meshes** (which are already valid to make them suitable for computations. That is why it can be used as a **preprocessor for MeshGems-Tetra**: by improving the quality of the surface mesh, MeshGems-SurfOpt helps generate volume mesh suitable for computations.
- MeshGems-Cleaner:
MeshGems-SurfOpt may also be **used after MeshGems-Cleaner**. Indeed, as MeshGems-SurfOpt keeps all the geometrical structures it detects in its input, very bad aspect ratio elements may generate numerical errors and faulty regions or unwanted small structures may lead to unnecessary fine meshes. MeshGems-Cleaner may correct faulty surface meshes and get rid of very bad aspect ratio elements or unwanted small structures, thus reducing numerical errors and unwanted refinements.
- MeshGems-CADSurf:
MeshGems-SurfOpt may also be used as a **postprocessor of MeshGems-CADSurf**. MeshGems-CADSurf may be used to produce a very fine mesh: this fine mesh will be the ancestor of a family of meshes, each of them with different characteristics depending on the settings of MeshGems-SurfOpt.

2.6 Integration within third party products

MeshGems-SurfOpt is delivered: as an executable and meshes come in an out of MeshGems-SurfOpt as files; or as a library using the MeshGems[®] Suite common API (documentation of MeshGems-SurfOpt API will be available soon).



MeshGems

User Manual

MG-SURFOPT

Release 1.1

CHAPTER 2. GETTING STARTED

Chapter 3

MeshGems-SurfOpt's principles

Contents

3.1	Some definitions	19
3.1.1	Triangle quality ("aspect ratio")	19
3.1.2	Mesh quality	19
3.1.3	Normalised length	20
3.1.4	Efficiency index	20
3.1.5	Chordal deviation	21
3.1.6	Geometric approximation	21
3.2	The algorithm	21
3.2.1	Flowchart	21

3.1 Some definitions

3.1.1 Triangle quality ("aspect ratio")

For a triangle, the quality function is defined as follows:

$$Q_k = \mu \frac{h_{\max}}{\rho_k}$$

where

- h_{\max} is the element diameter, *i.e.*, its longest edge
- ρ_k is the inradius of triangle k
- μ is a normalisation coefficient chosen so that the quality of the equilateral triangle is 1.

Notice that this function ranges between 1 and ∞ , the optimal value being, of course, 1. Hence, Q_k measures, so to speak, the degradation of the element k .

Notice also that this function defines an aspect ratio of the triangle.

3.1.2 Mesh quality

For a mesh \mathcal{T} , a global quality is often computed as:

$$Q_{\mathcal{T}} = \max_{k \in \mathcal{T}} Q_k$$

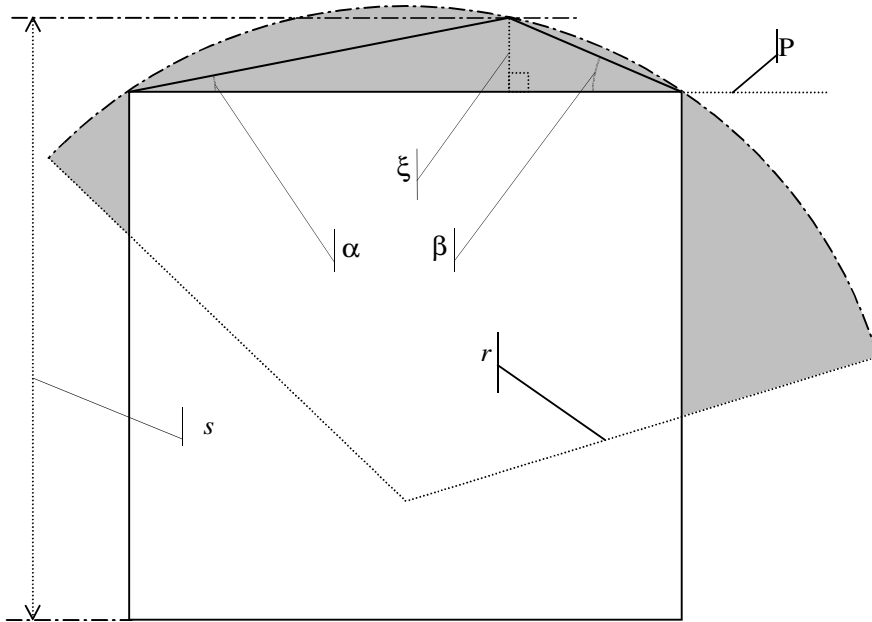


Figure 3.1: Definitions: chordal deviation ξ ; curvature $1/r$; bounding box s ; angles α and β

To get a better understanding, the average and histogram of the mesh qualities may also be computed.

3.1.3 Normalised length

The normalised length of a mesh edge AB is computed as:

$$l_{AB} = d(A, B) \int_0^1 \frac{1}{h(t)} dt$$

where

- $d(A, B)$ is the distance between A and B
- $h(t)$ is a monotonous size variation function along AB

If the normalised length equals 1, the edge obeys the metric map defined by $h(t)$.

3.1.4 Efficiency index

The efficiency index is then defined as:

$$\tau = 1 - \frac{1}{N_a} \sum_1^{N_a} (1 - e_i)^2$$

where

- N_a is the number of mesh edges
- $e_i = \begin{cases} l_i & \text{if } l_i \leq 1 \\ 1/l_i & \text{otherwise} \end{cases}$
- l_i is the normalised length of edge i

The efficiency index τ is thus defined as the average of the differences to 1 of the mesh edges. A value of τ greater than 0.93 (or 93%) indicates that the resulting mesh conforms well to the size map.

3.1.5 Chordal deviation

The chordal deviation ξ is the distance between the curve and the plane P of the corresponding mesh face approximating that curve (see Figure 3.1, ‘Definitions: chordal deviation ξ ; curvature $1/r$; bounding box s ; angles α and β ’ on page 20).

3.1.6 Geometric approximation

The geometric approximation ε is the chordal deviation relatively to the local curvature (see Figure 3.1, ‘Definitions: chordal deviation ξ ; curvature $1/r$; bounding box s ; angles α and β ’ on page 20).

This parameter relates to the angle deviation and can be written as:

$$\varepsilon = \frac{\xi}{r}$$

3.2 The algorithm

The aim of the algorithm is to achieve geometric isotropic or anisotropic meshes consisting of well-shaped triangular elements in the prescribed metric. The optimal is the equilateral triangle which has all its edges with a normalised length of 1. Thus, in terms of aspect ratio, the resulting mesh must contain as many well-shaped elements in the metric as possible.

At completion of the remeshing procedure, histograms are produced to help the user appreciate the mesh quality, both in terms of element shapes and sizes. Hence, a histogram of the edge lengths is written to see whether or not the mesh conforms to the size map. An efficiency index is also provided to summarise the result.

The mesh algorithm is based on local modifications only.

3.2.1 Flowchart

The general scheme of the method includes the following steps:

1. read the input file(s), metric file included if specified;
2. analyse the surface (normal computation, ridge and corners identification, etc.);
3. remesh to conform to the size map (internally built and, if provided, user modified) by iterative local modifications using geometrical operators (node relocation, vertex insertion/deletion) and topological operators (edge flipping);
4. optimise mesh to improve the element aspect ratio by vertex displacement and topological operators (conforming to the size map is not enforced explicitly here but this step should not move vertices far away from the location they had when it started);
5. write output file(s).

More technical details can be found in the references given in the bibliography section of this document.



Chapter 4

MeshGems-SurfOpt utilisation

The surface remeshing tool MeshGems-SurfOpt is easy to use. We illustrate this by describing the structure of its input and output files and how the program is run.

The execution can only be started in batch mode with all the necessary options on the command line, as a background process or not.

Contents

4.1	Input Files	23
4.1.1	Geometry description	23
4.1.2	Metric specification	24
4.2	Output Files	24
4.3	Relationship with the physics and/or the CAD description	25
4.4	Invoking MeshGems-SurfOpt	25
4.4.1	Running MeshGems-SurfOpt	25
4.4.2	Mandatory options	26
4.4.3	Generic options	26
4.4.4	Memory usage	31
4.4.5	I/O options and formatting data options	32
4.4.6	Remeshing options and control parameters	33
4.4.7	Using MeshGems-SurfOpt with its default parameter values	39

4.1 Input Files

Two files describing the geometry of the surface and the optional metric map are used by the program. Their formats are briefly described below. For more information see appendix B, ‘File format description’ on page 57.

4.1.1 Geometry description

The surface geometry is described via the “[mesh](#)” file format. For a complete description of this format, refer to the documentation describing the “[mesh](#)” format. This document can be found in the “[Docs](#)” sub-directory of the distribution. This is the common file format supported by all the MeshGems[®] Suite for meshes.



The surface triangulation is often conformal (although this is not strictly required if no finite element computation is expected).

It is also recommended that the significant edges and faces of the triangulation (sharp edges, material boundaries, etc..) be explicitly given as such in order to provide additional information to the mesher and preserve data useful for the user. Vertices may be tagged as corners or required to be preserved.

Note: the data files are used as geometry reference and as a discrete basis for geometrical information; therefore there is no direct link to a CAD system and no use is made of CAD data even if present in the same directory.

The mesh format.

This format is composed of a single file, “xxx.mesh” (ASCII) or “xxx.meshb” (binary). This file contains all the information needed to describe entirely the surface mesh and its underlying geometry. It is organized as a series of fields, identified by keywords. In addition to the vertex coordinates and the list of faces, it allows one to specify additional information such as corners, edges, ridges, required entities, etc. For a complete list of the keywords used by MeshGems-SurfOpt, refer to appendix B (end of this document).

4.1.2 Metric specification

The metric map (or size map) is described via the “sol” file format. For a complete description of this format, refer to the documentation describing the “mesh” format. This document can be found in the “Docs” sub-directory of the distribution. This is the common file format supported by all the MeshGems® Suite for size maps.

See also option “--sizemap_is_nodal” described in §4.4.6.4.7 which modifies the size map interpretation.

A metric map is a sizing information associated to a location of the given surface mesh.

The sizing information at a given location may be:

- a scalar value for an isotropic map or
- a matrix for an anisotropic map.

To be a valid metric, these matrices must be nondegenerate, symmetric and bilinear.

Size map source.

As pointed out in the introduction, MeshGems-SurfOpt can be used either as a surface remeshing & optimisation tool or can be integrated in the mesh adaptation loop of a computation:

- in most cases an internal size map (based on the surface properties) is computed and is used to govern the mesh modifications;
- if a “sol” file is provided to MeshGems-SurfOpt with option “--read_sizemap “xxx.sol””, it will be used to locally lower the metric computed above and help the user fine tune the target metric as desired.

See also §4.2 below.

4.2 Output Files

A single output mesh file is produced at completion. This file, corresponding to the resulting mesh, is written using the “mesh” format (“mesh” suffix for an ASCII file, or “meshb” for binary).

The size map associated with the output mesh can be written out at the end of the execution in a “sol” file thanks to “--write_sizemap “xxx.sol”” (see §4.4, ‘Invoking MeshGems-SurfOpt’ on page 25 and more particularly §4.4.5.4, ‘Setting size map to be written’ on page 32).

4.3 Relationship with the physics and/or the CAD description

We define an attribute as an integer (sometimes called «colour») which characterises a given edge or face.

An attribute will help identify a certain property:

- a physical property such as material, boundary condition, ...
- a CAD group affiliation such as CAD patch, construction line, ...

It is set in view of a particular treatment: entities with the same attribute usually have the same meaning or are processed the same way on the client side.

For example attribute numbers on faces can be used:

- to apply boundary conditions, or/and
- to identify a subset of a region.

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

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

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.

The enriched “`mesh`” file format allows one to specify this information.

By itself, MeshGems-SurfOpt algorithms does not need to make any use of the physical attributes included in the input files.

However, MeshGems-SurfOpt preserves this information in the result mesh for edge and faces attributes: they are treated as constraints to maintain a continuous line of the same colour or a manifold connected set of faces sharing the same colour (note that the edges and faces are not kept individually). The vertex attribute is preserved as long as the vertex is kept (required, non-manifold or corner).

As can be easily understood, the numbering of the entities is affected by the algorithm.

Hence based on the ridge detection, other geometrical entities and attributes, the various components of the surface mesh are automatically identified.

4.4 Invoking MeshGems-SurfOpt and using Options

This section describes how to run MeshGems-SurfOpt and the possible options and parameters needed to control the surface remeshing process.

4.4.1 Running MeshGems-SurfOpt

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

```
mg-surfopt.exe --in < filein > [--out < fileout >] [<extra options set>]
```

where:

- the parts between angle brackets <> are parameters given 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-surfopt.exe [<extra option set 1>] --in <filein> [<extra option set 2>] --out <fileout>
```

If options are repeated, the last occurrence on the command line is taken into account.

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 Name of the input surface mesh: “--in <filein>”

This option tells to the mesher the name of the input surface mesh, which must be in the “[mesh](#)” format.

4.4.3 Generic options

4.4.3.1 Getting online help: “--help”

```

1
2  =====
3  MG-SurfOpt -- 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-SurfOpt (Copyright 2012-2013 by Distene SAS)
15     date of run: 07-Mar-2013 AT 11:19:00
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
24 MG-SurfOpt USAGE
25 mg-surfopt.exe [--help] \
26   [--verbose <verb>] [--in <filein>] [--out <fileout>] \
27   [--read_sizemap <sizefile>] [--write_sizemap <sizefile>] \
28   [--use_deprecated_errors <yes|no>] \
29   [--backward_compatibility <yes|no>] [--max_memory <maxmem>] \
30   [--min_size <hmin>] [--max_size <hmax>] [--gradation <grad>] \
31   [--metric <isomode>] [--use_surface_proximity <yes|no>] \
32   [--surface_proximity_layers <prox2Dlayers>] \
33   [--surface_proximity_gradation <prox2Dgradation>] \
34   [--use_volume_proximity <yes|no>] \
35   [--volume_proximity_layers <prox3Dlayers>] \
36   [--volume_proximity_gradation <prox3Dgradation>] \
37   [--chordal_error <tol>] [--geometric_approximation_angle <geomapp>] \
38   [--ridge_on_reference_lines <yes|no>] \
39   [--uniform_flat_subdivision <yes|no>] \

```

```

40     [--uniform_subdivide_level <div>] [--ridge_angle <ridge>] \
41     [--tags <tagmode>] [--remove_bad_input_surface_elements <yes|no>] \
42     [--bad_input_surface_element_aspect_ratio <qllevel>] \
43     [--sizemap_is_nodal <yes|no>] [--optimisation <optim>] \
44     [--split_overconstrained_elements <FEM>] [--element_order <order>] \
45     [--element_generation <gen>] [--enrich <yes|no>] \
46     [--Hausdorff_like <yes|no>] [--sand_paper <yes|no>] \
47     [--create_nodes_on_flat_edge <yes|no>] [--compute_ridges <yes|no>] \
48     [--target_vertex_count <nbv>]
49
50     Short option (if it exists)
51     /      Long option
52     |      /      Description
53     |      |      /
54     v      v      v
55
56     -h --help
57           gives this help.
58
59     -v --verbose <verb>
60           sets the verbosity level.
61           0 : no detail
62           10 : very detailed
63           Default is 3
64
65     -i --in <filein>
66           sets the input file.
67           (MANDATORY OPTION)
68
69     -o --out <fileout>
70           sets the output file.
71           Default is derived from input filename
72
73     --read_sizemap <sizefile>
74           sets the sizemap input file.
75           Default is not to read a sizemap
76
77     --write_sizemap <sizefile>
78           sets the sizemap output file.
79           Default is not to write a sizemap
80
81     --use_deprecated_errors <yes|no>
82           sets error codes for messages:
83           no : use the current error codes
84           yes : use the deprecated error codes used prior to 1.1 release
85           Default is no
86
87     --backward_compatibility <yes|no>
88           switch back to the old parsing mechanism.
89           Default is new style
90
91     --max_memory <maxmem>
92           sets the maximum memory usage allowed in megabytes.
93           Default is for 50000 vertices
94
95     --min_size <hmin>
96           sets the minimal size allowed in the size map.
97           Actual sizes may be smaller if input sizes were smaller and some
98           constraint, usually geometrical, prevented coarsening.
99           Default is 0.005r
100
101     --max_size <hmax>
102           sets the maximal size allowed in the size map.
103           Actual sizes may be bigger if input sizes were bigger and some
104           constraint, usually required edges, prevented refinement.
105           Default is unset
106

```



```
107 --gradation <grad>
108     sets the surface gradation.
109     Default is 1.3
110
111 --metric <isomode>
112     sets the metric type.
113     Values may be: iso, isotropic, aniso or anisotropic.
114     Default is isotropic
115
116 --use_surface_proximity <yes|no>
117     activates the proximity between surface structures.
118     Default is not active
119
120 --surface_proximity_layers <prox2Dlayers>
121     sets the desired number of mesh layers between surface structures.
122     Default is not active
123
124 --surface_proximity_gradation <prox2Dgradation>
125     sets the desired gradation to use for layer growth between surface
126     structures.
127     Default is surface gradation
128
129 --use_volume_proximity <yes|no>
130     activates the proximity between volume borders.
131     Default is not active
132
133 --volume_proximity_layers <prox3Dlayers>
134     sets the desired number of mesh layers between volume borders.
135     Default is not active
136
137 --volume_proximity_gradation <prox3Dgradation>
138     sets the desired gradation to use for layer growth between volume
139     borders.
140     Default is volume gradation
141
142 --chordal_error <tol>
143     sets the admissible chordal error.
144     Default is 0.001r
145
146 --geometric_approximation_angle <geomapp>
147     sets the admissible angular error in degrees.
148     Default is 15
149
150 --ridge_on_reference_lines <yes|no>
151     tells module that geometry ridges are only a subset of reference
152     edges (useful for CAD output)
153     Default is no
154
155 -U --uniform_flat_subdivision <yes|no>
156     activates the uniform subdivision algorithm
157     Default is no
158
159 --uniform_subdivide_level <div>
160     sets the uniform subdivision parameter.
161     Default is 2
162
163 --ridge_angle <ridge>
164     defines a ridge edge when facets sharing edge form an angle above
165     this value in degrees.
166     Default is 45
167
168 --tags <tagmode>
169     sets the tag processing.
170     respect : follow tags;
171     clear   : remove all tags
172     ignore  : leaves them as they are but ignores them (NOT
173             IMPLEMENTED).
```

```

174         Default is respect
175
176     --remove_bad_input_surface_elements <yes|no>
177         activates the very poor quality elements cleaning procedure.
178         Default is no
179
180     --bad_input_surface_element_aspect_ratio <Qlevel>
181         sets the aspect ratio level above which elements go through the
182         cleaning procedure.
183         Aspect ratio values are from 1 (best) to infinity (worse).
184         Default is 200
185
186     --sizemap_is_nodal <yes|no>
187         tells module that size map is to be considered purely nodal i.e.
188         gradation is applied on existing edges
189         Default is to consider size map piecewise-linear
190
191     --optimisation <optim>
192         sets the optimisation processing.
193         yes : performs edge flipping and node displacement after
194             coarsening and refining mesh.
195         no  : disables edge flipping and node displacement.
196         only : does not coarsen or refine mesh.
197         Default is yes
198
199     --split_overconstrained_elements <FEM>
200         splits elements having all their vertices on the surface boundaries.
201         Values may be: yes, no or only
202         Default is no
203
204     --element_order <order>
205         sets the output element order desired.
206         Possible values are: linear or quadratic.
207         Default is linear
208
209     --element_generation <gen>
210         sets the type of mesh generation desired.
211         Possible values are: triangle or full_quad (other values are NOT
212             IMPLEMENTED yet).
213         Default is triangle
214
215     --enrich <yes|no>
216         enables/disables enrichment.
217         Default is yes
218
219     --Hausdorff_like <yes|no>
220         uses Hausdorff-like sizemap if yes.
221         Default is no
222
223     --sand_paper <yes|no>
224         activates the sand papering (no-shrinkage vertex smoothing).
225         Default is no
226
227     --create_nodes_on_flat_edge <yes|no>
228         creates new nodes on straight edges (i.e. not on SURFOPT computed
229         curves).
230         Default is no
231
232     --compute_ridges <yes|no>
233         enables/disables ridge detection.
234         Default is yes
235
236     -p --target_vertex_count <nbv>
237         specifies a number of desired vertices.
238         Default is no target
239
240

```

```

241      ELAPSED TIME  3.4e-05 SEC. ( 0 SEC. WITHOUT OVERHEAD)
242
243      =====
244              MG-SurfOpt -- MeshGems 1.1-3 (January, 2013)
245      END OF SESSION - MG-SurfOpt (Copyright 2012-2013 by Distene SAS)
246              compiled Mar  7 2013 08:59:08 on Linux_64
247              MeshGems is a Registered Trademark of Distene SAS
248      =====
249      ( Distene SAS
250      Phone: +33(0)970-650-219   Fax: +33(0)169-269-033
251      EMail: <support@distene.com> )

```

4.4.3.2 Verbose mode: “`--verbose <level >`” (default: 3)

This option allows the user to control the amount of information issued on meshing steps while MeshGems-SurfOpt processes the mesh.

This option values range from 0 (lower amount of information) to 10 (highest amount of information). Default `<level >` is 3.

4.4.3.3 Deprecated error mode: “`--use_deprecated_errors <yes/no >`” (default: no)

In the process of unifying MeshGems[®] Suite, MeshGems-SurfOpt error diagnostic codes have been changed from those used by Yams.

The messages are now easier to parse in a calling programme and they may contain additional data easier to extract. See §A.1.

If for compatibility reasons (with your scripts or command calls for example) you wish to *temporarily* use the old error system, setting “`--use_deprecated_errors yes`” will switch back to the old code numbers.

4.4.3.4 Backward compatibility: “`--backward_compatibility <yes/no >`” (default: no)

The new common framework of SuiteName imposes that all arguments be preceded by an option.

This means that:

1. instead of using the first non option argument as the input file name (and try several suffixes if it was missing) and the second non option argument as the output file name, MeshGems-SurfOpt requires now the use of options “`--in <filein >`” and “`--out <fileout >`”;
2. likewise, MeshGems-SurfOpt does not try anymore to read a size map which name would be constructed from the input file with its suffix replaced by “.bb” and option “`--read_sizemap <insizefile >`” has to be used.

Old options have been renamed and their corresponding new ones are described in §D, page 65.

If for compatibility reasons (with your scripts or command calls for example) you wish to *temporarily* use the old system, most of the old options will still work for now but changing back to the old parsing mechanism requires an additional option. Place the option “`--backward_compatibility yes`” as the first option on the command line like this:

```
mg-surfopt.exe --backward_compatibility yes <old options>
```

to be able to parse a Yams command line.

4.4.4 Memory usage

The program requires about 350 bytes per point on average (between 145 and 465 bytes per point depending on the choice of options). It is thus possible to estimate *a priori* the required memory size to complete a job. As an example of memory space needed, a mesh of 543,652 points (1,087,716 triangles) requires 182 MiB (about 351 bytes per point). Conversely, a workstation having 512 MiB of memory can handle a mesh of about 1,500,000 points (about 358 bytes per point).

The memory needed for mesh modification is allocated dynamically at the beginning of the procedure (data reading), based on the initial triangulation size. When the program reaches the amount of maximal memory allowed, it will stop inserting points and, if possible, the current mesh will be saved as it is, valid and conformal, provided the input mesh was valid and conformal.

The user can however run the program using option “`--max_memory < maxmem >`”.

4.4.4.1 Memory usage: “`--memory_max < maxmem >`”

This option is useful if:

- the automatically allocated memory reveals insufficient when the user asked to enrich the given mesh;
- the user wants to limit the amount of memory used by the program.

Typing:

```
mg-surfopt.exe --max_memory 60 ...
```

asks the program to allocate at most 60 millions bytes of memory for the meshes.

If the input mesh size requires more memory than requested or if the allocated memory (user defined or not) exceeds the machine capabilities, the tool will stop because of insufficient memory.



4.4.5 I/O options and formatting data options

4.4.5.1 Setting the input mesh file name: “`--in <filein >`”

This option sets the input “`mesh`” file. The file name `<filein >` may be specified on the command line such as:

```
mg-surfopt.exe --in <filein >
```

Important notice: The suffix “`mesh`” (or “`meshb`” for binary files) is mandatory.

This option is sufficient to perform a run of MeshGems-SurfOpt. If no other option is specified, MeshGems-SurfOpt will assume the parameters to be used by the software are the default values, refer to 4.4.7 for more details.

4.4.5.2 Setting the output mesh file name: “`--out <fileout >`”

This option sets the output “`mesh[b]`” file name. The file name `<fileout >` may be specified on the command line such as:

```
mg-surfopt.exe --in <filein > --out <fileout >
```

If unset, the output mesh file name is built by adding “`_surfopt.mesh`” to the input file base name.

Important notice:

- The suffix “`mesh`” (or “`meshb`” for binary files) is mandatory.
- The output file name and the input file name must be different.
- Using an existing file as output file is forbidden.

4.4.5.3 Setting size map to be read: “`--read_sizemap <insizefile >`”

This option sets the input “`sol`” file which provides an isotropic or anisotropic sizemap.

The provided size map must be coherent with the value of option “`--metric`” if it is specified (see §4.4.6.4.4).

4.4.5.4 Setting size map to be written: “`--write_sizemap <sizefileout >`”

This option asks MeshGems-SurfOpt to write in the “`sol`” file named `<sizefileout >` the sizemap associated with the mesh result.

4.4.6 Remeshing options and control parameters

4.4.6.1 Coarsening, enrichment & optimisation options

4.4.6.1.1 Disabling enrichment: “`--enrich < yes/no >`” (default: yes)

This option disables (or reenables) the mesh enrichment.

By default, MeshGems-SurfOpt enriches the mesh as asked for by the computed or user provided size map.

4.4.6.1.2 Disabling optimisation: “`--optimisation < yes/no >`” (default: yes)

This option disables (or reenables) the mesh optimisation.

By default, MeshGems-SurfOpt optimises the mesh quality by moving vertices on the surface and by flipping edges, after the mesh has been coarsened and / or enriched as asked for by the size map.

4.4.6.1.3 Using Hausdorff.like sizemap: “`--Hausdorff_like < yes/no >`” (default: no)

This option enables (or disables) the purely geometric mesh generation controlled only by tolerance.

By default, MeshGems-SurfOpt builds a mesh suitable for computations, that is, a mesh that obeys the tolerance and geometric approximation and that also avoids bad aspect ratio elements and rapid edge size variations.

4.4.6.2 Activating the sand papering algorithm: “`--sand_paper < yes/no >`” (default: no)

if set to yes, the standard algorithm is replaced by a **surface sandpapering** that avoids shrinkage of the given surface triangulation, i.e., the high curvature variations of the given surface will be smoothed out without shrinking the volume in doing so. If ridges and corners are defined, they will be kept as they are in the resulting mesh.

It depends on the “`--chordal_error < tol >`” parameter which limits the amount of displacement allowed for vertices.

4.4.6.3 Uniform subdivision

4.4.6.3.1 Activating the uniform subdivision algorithm: “`--uniform_flat_subdivision < yes/no >`” (default: no)

If set to yes, the standard algorithm is replaced by a **uniform subdivision** of the given surface triangulation, i.e., each triangle of the given surface triangulation is considered flat and is divided into identical triangles.

It depends on the parameter “`--uniform_subdivide_level < n_{div} >`”: each triangle will be divided into n_{div}^2 triangles.

4.4.6.3.2 Setting the uniform subdivision level: “`--uniform_subdivide_level < n_{div} >`” (default: 2)

This parameter sets n_{div} the number of triangle side subdivisions when the uniform subdivision is used (see 4.4.6.3, ‘Uniform subdivision’ on page 33).



4.4.6.4 Setting size & surface parameters

4.4.6.4.1 Minimum element's size: "`--min_size < minsize >`" (default: 0.005r)

This parameter sets the minimal size used to truncate the small values of the size map. It can be absolute or relative to the bounding box, in which case, it is followed by the letter 'r'.

By default, it is 0.005r, which means 5/1000 of the bounding box longest diagonal.

4.4.6.4.2 Maximum element's size: "`--max_size < maxsize >`" (default: unset)

This parameter sets the maximal size used to truncate the big values of the size map. It can be absolute or relative to the bounding box, in which case, it is followed by the letter 'r'.

By default, no such truncation is performed.

4.4.6.4.3 Gradation: "`--gradation < ψ >`" (default: 1.3)

This parameter enables the user to control the **element size variation** in the triangulation *i.e.* MeshGems-SurfOpt will avoid getting two adjacent edges which sizes differ by a factor bigger than ψ .

To avoid rapid size variations, a size correction procedure is applied to the size map which is essentially a slope limiter. More precisely, the derivative of the size map is limited to $\log \psi$: if two adjacent vertices have sizes respectively s_1 and s_2 , $s_1 < s_2$, and the distance between those two vertices is d , then s'_2 the new size at the second vertex will be:

$$s'_2 = \min(s_2, s_1 + d \log \psi)$$

This procedure is de-activated if the value of ψ is set to -1 .

The default value is 1.3, which is the usual value set for computational meshes.

4.4.6.4.4 Metric type: "`--metric < isomode >`" (default: iso)

These parameters control whether MeshGems-SurfOpt is allowed or not to read / create anisotropic elements. Anisotropic elements are stretched elements *i.e.* elements with a higher aspect ratio than 1.

The effect of this option is :

"`--metric iso`": by default and if this option is selected, only isotropic elements are targeted in the resulting mesh.

Isotropic elements have a better quality grade (*i.e.* they are closer to an equilateral element) and consequently are usually more suitable for computation than anisotropic elements.

"`--metric aniso`": if selected, anisotropic elements are allowed in the resulting mesh (anisotropic elements are also allowed by default with option "`--Hausdorff_like yes`", see 4.4.6.1.3).

These anisotropic elements will be aligned with the main directions of curvature of the surface mesh such that their long side is aligned with the least curved direction of the surface.

The anisotropic mesh contains usually less vertices and elements than its isotropic counterpart.

4.4.6.4.5 Chordal error: "`--chordal_error < tol >`" (default: 0.001r)

This parameter enables the user to control the accuracy of the piecewise linear approximation of the surface, together with the geometric approximation (see §4.4.6.4.6).

This parameter enables the user to bound the maximal chordal deviation allowed, that is, the **maximal distance** allowed between the detected curve and the plane P of the corresponding mesh face (see Figure 3.1, ‘Definitions: chordal deviation ξ ; curvature $1/r$; bounding box s ; angles α and β ’ on page 20). In other words, it avoids having faces too far away from the curve they represent (or should represent).

This parameter enables the user to specify the maximal chordal deviation ξ_{\max} . Following that criterion:

- if the chordal deviation ξ is much smaller than ξ_{\max} , it is acceptable to remove the considered point;
- if the chordal deviation ξ is greater than ξ_{\max} , the considered mesh face should be refined by adding a point on the curve.

If the parameter given value $\langle tol \rangle$ is

$\delta_{\max}\mathbf{r}$: it means the given value is relative; ξ_{\max} corresponds to $s \times \delta_{\max}$, where s is the size of the bounding box longest diagonal of the domain;

δ_{\max} : it means the given value is absolute; $\xi_{\max} = \delta_{\max}$ and it is expressed in model units, e.g. if $\delta_{\max} = 2$ and point coordinates are given in millimetres, it means that the maximal chordal deviation is 2 mm.

This field can be used for all remeshing options.

By default $\langle tol \rangle$ is 0.001r and ξ_{\max} is then set to $s \times 0.001$.

4.4.6.4.6 Geometric approximation: “`--geometric_approximation_angle < geomapp >`” (default: 15)

This parameter enables the user to control the accuracy of the piecewise linear approximation of the surface, together with the chordal error (see §4.4.6.4.5).

This parameter enables the user to control the **maximal angle** allowed between two adjacent faces. It can be used to bound the maximal deviation of the mesh faces from the tangent planes at mesh vertices. In other words, it avoids having sharp angles between faces representing a smooth curve. On the Figure 3.1, ‘Definitions: chordal deviation ξ ; curvature $1/r$; bounding box s ; angles α and β ’ on page 20, this parameter bounds the angles α and β .

This parameter enables the user to specify the maximal chordal deviation ε_{max} relatively to the curvature r . Following that criterion:

- if the chordal deviation ξ is much smaller than $\varepsilon_{max} \times r$, it is acceptable to remove the considered point;
- if the chordal deviation ξ is greater than $\varepsilon_{max} \times r$, the considered mesh face should be refined by adding a point on the curve.

One can see that the smaller the radius r , the harder it is to satisfy this criterion: ε_{max} is a real value corresponding to a percentage, the ratio between the chordal deviation to the local curvature.

This field is used only for size-based remeshing mode.

By default, $\langle geomapp \rangle$ is 15 degrees and $\varepsilon_{max} = 1 - \cos \langle geomapp \rangle \approx 0.034$ which should lead to angles of less than 15 degrees between two adjacent faces *i.e.* not separated by a ridge.

4.4.6.4.7 Assume non-linear size on edges: “`--sizemap_is_nodal < yes/no >`”

If the user provides a size map, size maps are then assumed linear between nodes by default. That means that, for example, the size in the middle of an edge is the average of the two edge vertex sizes.

If the user has not provided a “`sol`” size map, MeshGems-SurfOpt computes its own internal size map and the gradation is applied to the size map between vertices along the edges: by default, the internal MeshGems-SurfOpt size map is what we call *nodal*. That is, if the gradation is ψ , the sizes on the edge vertices is respectively s_1 and s_2 and the edge length is d , the size at $t \in [0, 1]$ of the edge is:

$$s_t = \min(s_1 + td \log \psi, s_2 + (1 - t)d \log \psi)$$

Linear size maps lead sometime to unwanted very dense meshes in particular when the input mesh is not fine enough to capture the size map details. This is typically the case inside a mesh adaptation loop.



The option “`--sizemap_is_nodal yes`” tells MeshGems-SurfOpt to assume the user size map is nodal. This usually leads to much smaller meshes and is well suited inside an adaptation loop because it speeds up the adaptation convergence while avoiding going through unnecessary dense meshes.

4.4.6.5 Computing ridges: “`--compute_ridges <yes/no>`” (default: yes)

When set to no, MeshGems-SurfOpt will not try to detect any new ridge edge by its own mechanism, *i.e.* it will consider as ridge only the ridges given in the input mesh file. All non-ridge edges that would have been detected as ridge by the “`--ridge_angle α` ” parameter (see §4.4.6.5.1) will be considered as part of the same continuous patch.

This option should be used when all the known ridges of the mesh are given and when all other possible ridges are not geometric ridges to take into account.

4.4.6.5.1 Setting the ridge angle: “`--ridge_angle < α >`” (default: 45)

This parameter specifies the angular value α for the **automatic detection of ridges and corners**. A ridge is automatically detected by MeshGems-SurfOpt if the angle between the normal vectors of two adjacent faces exceeds the value α . A corner is automatically detected if the angle between two ridge edges sharing the corner candidate exceeds α or if at least three normal vectors of the corner candidate adjacent faces exceeds the value α with each other.

The default value is 45 degrees.

For example, to detect more ridges than with the default value set the ridge angle to, say, 35 degrees by:

```
mg-surfopt.exe --in mymesh --ridge_angle 35
```

4.4.6.5.2 Restricting ridge detection to reference lines: “`--ridge_on_reference_lines <yes/no>`” (default: no)

Because the detection of ridges by angle above can sometimes lead to false ridges, this option is useful for the cases where different patches have different attributes and ridge lines cannot lie inside patches. For example, CAD exported cases usually have this property.

The option “`--ridge_on_reference_lines yes`” tells MeshGems-SurfOpt to look for ridge edges on reference lines only.

Reference lines are defined as:

- a set of connected edges that separates element regions with two different attributes;
This kind of reference lines are automatically detected by MeshGems-SurfOpt (unless “`--tags clear`” is activated) and saved in the output “`mesh`” file under keyword section `Edges` with attribute zero.
- a set of connected edges with the same non-zero attribute defined in the input mesh (in this case, the two regions separated by this line may or may not have different attributes).

MeshGems-SurfOpt will always preserve such reference lines unless “`--tags clear`” is activated.

4.4.6.6 Setting surface & volume proximity parameters

4.4.6.6.1 Activating surface proximity: “`--use_surface_proximity <yes/no>`” (default: no)

If set to yes, this option activates the surface proximity algorithm.

This parameter asks MeshGems-SurfOpt to:

1. compute the distance between geometric and reference structures on the surface;

2. modify the size map such that π_{2d} surface element layers should be created between geometric and reference structures.

where π_{2d} is a given number of layers (see §4.4.6.6.2).

Please note that, as it is not strictly possible to obey the given or computed size map, the actual number of levels may differ slightly.

If π_{2d} is equal to 0, the size map is modified such that elements may stretch between structures but not more than what is permitted by the active gradation (see §4.4.6.6.3).

4.4.6.6.2 Setting the surface proximity layers: “`--surface_proximity_layers < π_{2d} >`”

This option sets π_{2d} the number of layers for the surface proximity algorithm (see §4.4.6.6.1).

4.4.6.6.3 Setting the surface proximity gradation: “`--surface_proximity_gradation < ψ_{2d} >`”

This option sets ψ_{2d} the gradation parameter used in the surface proximity computation.

By default, it is the general gradation (see §4.4.6.4.3).

4.4.6.6.4 Activating volume proximity: “`--use_volume_proximity < yes/no >`” (default: no)

This parameter asks MeshGems-SurfOpt to:

1. compute the distance between opposite surfaces of the volume;
2. modify the size map such that π_{3d} volume element layers should be created between surfaces.

where π_{3d} is a given number of layers (see §4.4.6.6.5).

Please note that:

- as it is not strictly possible to obey the given or computed size map, the actual number of levels may differ slightly;
- the surface recognition mechanism does not use the geometric and reference structures yet;
- the volume recognition mechanism may fail. In this case, using a refined surface mesh may help.

If π_{3d} is equal to 0, the size map is modified such that elements may stretch between structures but not more than what is permitted by the active gradation (see §4.4.6.6.6).

Limitation: For now, this option works:

1. for almost all types of meshes (but not all of them yet);
2. only in isotropic mode (anisotropic mode will be implemented very soon and will be made available as a patch).

4.4.6.6.5 Setting the volume proximity layers: “`--volume_proximity_layers < π_{3d} >`”

This option sets π_{3d} the number of layers for the surface proximity algorithm (see §4.4.6.6.4).

4.4.6.6.6 Setting the volume proximity gradation: “`--volume_proximity_gradation < ψ_{3d} >`”

This option sets ψ_{3d} the gradation parameter used in the volume proximity computation.

By default, it is the general gradation (see §4.4.6.4.3).

4.4.6.6.7 Setting the element type: “`--element_generation < type >`” (default: triangle)

This option sets the element type that MeshGems-SurfOpt will consider for the remeshing process.



- By default, the element type is set to "triangle".
- When set to "full_quad", MeshGems-SurfOpt will switch to full quadrilateral generation and a mesh containing only quadrilaterals will be created.

Limitation for the quadrilateral generation: Because the pure quadrilateral generation algorithm relies on splitting existing edges, for now:

1. this option only works in standard enrichment mode i.e. "`--enrich yes`" "`--Hausdorff_like no`" "`--sand_paper no`" "`--uniform_flat_subdivision no`";
2. MeshGems-SurfOpt cannot enforce the following required directive keywords:
RequiredEdges, RequiredTriangles, RequiredQuadrilaterals.

4.4.6.8 Setting the element order: "`--element_order <order>`" (default: linear)

When *order* is different from *linear* MeshGems-SurfOpt creates new vertices placed on the curved surface and adds them to elements such that an edge ends up with a number of extra vertices depending on the desired order.

It may be used to obtain higher order elements.

MeshGems-SurfOpt creates the same linear mesh basis independently from the value of *order*, i.e. the resulting mesh will contain the same number of elements with "`--element_order quadratic`", and with "`--element_order linear`", the default value.

When *<order>* is higher than linear, new vertices are created and saved in the "`mesh`" file under keyword section `Vertices`, and used under new keyword sections `ExtraVerticesAtTriangles` and, if applicable, `ExtraVerticesAtEdges`. Thanks to this, the format is easily extensible to higher order elements and the user may:

1. read the linear mesh as usual without taking care of the two new keywords
`ExtraVerticesAtTriangles` and `ExtraVerticesAtEdges`;
2. ignore and store as he wishes the two new keyword data.

Note this is quite different from the "`uniform_flat_subdivision yes`" (see §4.4.6.3 and §4.4.6.3.2): "`uniform_flat_subdivision yes`" creates new vertices and new elements and places the new nodes on the straight edges or on the flat elements while "`--element_order quadratic`" creates only new vertices and places them on the curved surface.

Limitation: For now,

1. this option can be applied only to triangles;
2. *order* can be only:
 - linear* : which means no extra vertex at all (P1 or linear triangles),
 - or *quadratic* : which means one extra vertex on edge (P2 or quadratic triangles).

4.4.6.9 Create nodes on flat edge: "`--create_nodes_on_flat_edge <yes/no>`" (default: no)

When set to yes, MeshGems-SurfOpt will create points **on straight edges**, that is, it does no mapping on detected curves.

This option should be used when the user does not trust the detected curves, a situation which usually arises when the initial mesh is too poor (i.e. there is not enough relevant data around a point or the numerical error is too high to define the curves correctly).

4.4.6.10 Setting the tags behavior: "`--tags <behavior>`" (default: respect)

This option specifies how surface attributes must be considered. The effect of this option is:

"`--tags respect`": the mesher will preserve surface attributes. This is the default behaviour.

“`--tags clear`”: the mesher clears the surface attributes (references). This will remove constraints not coming from the geometry and usually facilitates the geometry recovery, but all surface attributes will be lost.

4.4.6.11 Targeting a vertex count: “`--target_vertex_count < N_v >`” (default: unset)

When set, MeshGems-SurfOpt will try to refine/coarsen the mesh such that the resulting mesh has N_v vertices.

Limitation: Today, this only works using Hausdorff like remeshing (see §4.4.6.1.3) because to achieve such result MeshGems-SurfOpt must assume the only constraint is the distance of the mesh to the surface.

4.4.6.12 Cleaning input surface mesh

4.4.6.12.1 Removing input bad element: “`--remove_bad_input_surface_elements < yes/no >`” (default: no)

This parameter allow MeshGems-SurfOpt to perform cleaning depending on a user defined quality threshold.

4.4.6.12.2 Setting quality threshold: “`--bad_input_surface_element_aspect_ratio < Qlevel >`” (default: 200)

This parameter lets the user decide what is the quality above which cleaning is to be tried (quality is between 1 and ∞ and rises as the element gets bad).

Please note that:

- lowering this parameter may lead to the loss of mesh geometric features;
- increasing this parameter may be needed to preserve mesh geometric features.

4.4.6.13 Splitting over-constrained element: “`--split_overconstrained_elements < yes/no >`” (default: no)

When set to yes, MeshGems-SurfOpt will correct the resulting mesh for finite-element computation. It is the last stage of MeshGems-SurfOpt where it tries to avoid over-constrained corners. It is disabled by default.

4.4.7 Using MeshGems-SurfOpt with its default parameter values

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

```
mg-surfopt.exe --in < filein >
```

The output filename will be computed automatically by inserting “`_surfopt`” before the suffix.

The default processing is to enrich or coarsen the mesh if necessary, and to optimise it.



Chapter 5

MeshGems-SurfOpt's examples of use

This is probably the section of main interest for the user as it will emphasise the various features of the software package.

5.1 First example

In order to give the user a better understanding of MeshGems-SurfOpt's behaviour, a relatively simple example is shown below with different settings of the most common options and parameters.

5.1.1 Optimisation and coarsening: “`--enrich no`”

See Figure 5.1, ‘Examples of optimisation and coarsening (default values)’ on page 42.

5.1.1.1 Using MeshGems-SurfOpt with its default parameter values

In the cube faces shown in the example, the coarsening with “`--enrich no`” was able to remove elements on the top corner and face below:

⇒ the curvature from the hole and from the small subdomain¹ was propagated by the mesh gradation parameter preventing mesh size to vary too quickly.

The difference between “`--Hausdorff_like no`” and “`--Hausdorff_like yes`” is that the first one uses the geometric approximation and uses the mesh gradation propagation parameter while the second option does not:

⇒ as “`--enrich no`” with “`--Hausdorff_like yes`” uses only the maximal chordal deviation parameter, it is able to coarsen most of the points but those defining the cylindrical hole curve.

With optimisation “`--optimisation only`”, most of the effect of MeshGems-SurfOpt takes place in the face below and the hole:

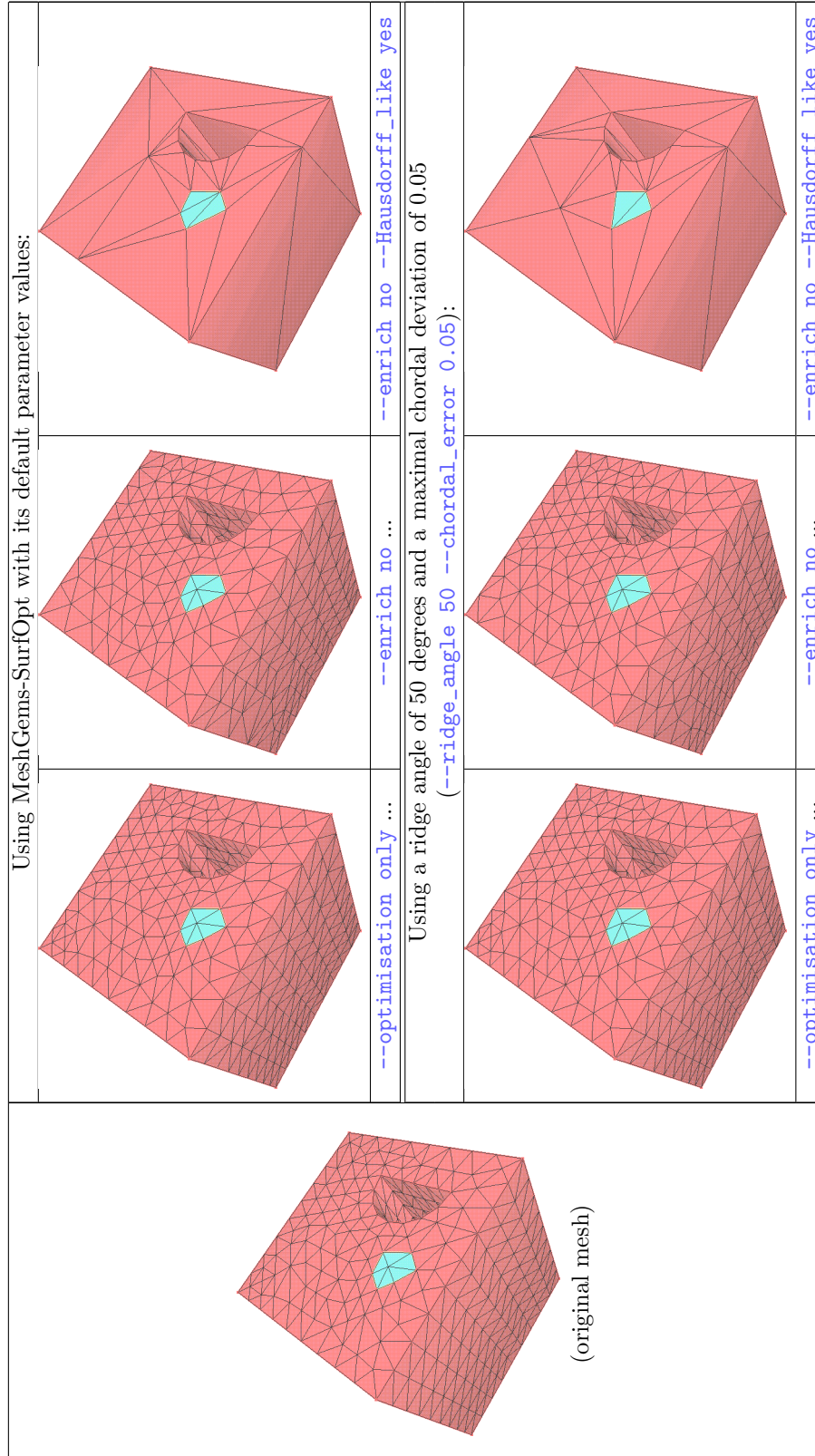
⇒ only smoothing and edge swapping has been used to improve the mesh quality.

5.1.1.2 Using ridge angle and maximal chordal deviation

When the ridge angle α is set to 50 degrees and the maximal chordal deviation is set to 0.05, the main differences with the previous case are that

¹On the figure, it is on the left of the hole and it is in blue color or brighter in monochrome.

Figure 5.1: Examples of optimisation and coarsening (default values)





- for the small subdomain, one of its points is no more considered a corner by MeshGems-SurfOpt (effect of a greater angle of ridge value);
- a stronger constraint has been set for the curvature rendering (effect of the new maximal chordal deviation value).

One can see that:

- for “`--optimisation only`”, there is no visible difference (although the number of corners detected has changed);
- for “`--enrich no`”, MeshGems-SurfOpt has coarsened less than previously because of the stronger constraint on the chordal deviation;
- for “`--enrich no`” with “`--Hausdorff_like yes`”, MeshGems-SurfOpt has coarsened one more point that is no more considered a corner.

Warning:

- using “`--Hausdorff_like yes`”, likely to be unsuited for FE like computations, do not keep the boundary conditions (see the small subdomain transformation);
- using “`--Hausdorff_like no`”, although keeping the shape of the boundary conditions, may modify its area.

The user may explicitly set as corners all the nodes of a boundary condition to ensure that the shape and area of this boundary condition are preserved.

5.1.2 Optimisation, coarsening and enrichment: “`--enrich yes`”

See Figure 5.2, ‘Examples of optimisation and coarsening/enrichment (default values)’ on page 43.

5.1.2.1 Using MeshGems-SurfOpt with its default parameter values

In the cube faces shown in the example, the coarsening/enrichment with “`--enrich yes`” has coarsened the top corner and face below as it does in the pure coarsening case (“`--enrich no`”) and it has enriched the hole curved part and the small subdomain:

⇒ the curvature from the hole and from the small subdomain has been propagated by the mesh gradation parameter preventing mesh size to vary too quickly.

The difference between “`--enrich yes`” with “`--Hausdorff_like no`” and “`--Hausdorff_like yes`” is that the first one uses the geometric approximation to compute the map size while the second option does not:

⇒ With “`--Hausdorff_like yes`”, MeshGems-SurfOpt uses only the maximal chordal deviation parameter and the mesh propagation parameter to compute its size map; in this case it leads to slightly smaller sizes than with “`--Hausdorff_like no`” and then to a more refined mesh.

With optimisation “`--optimisation only`”, most of the effect of MeshGems-SurfOpt takes place in the face below and the hole:

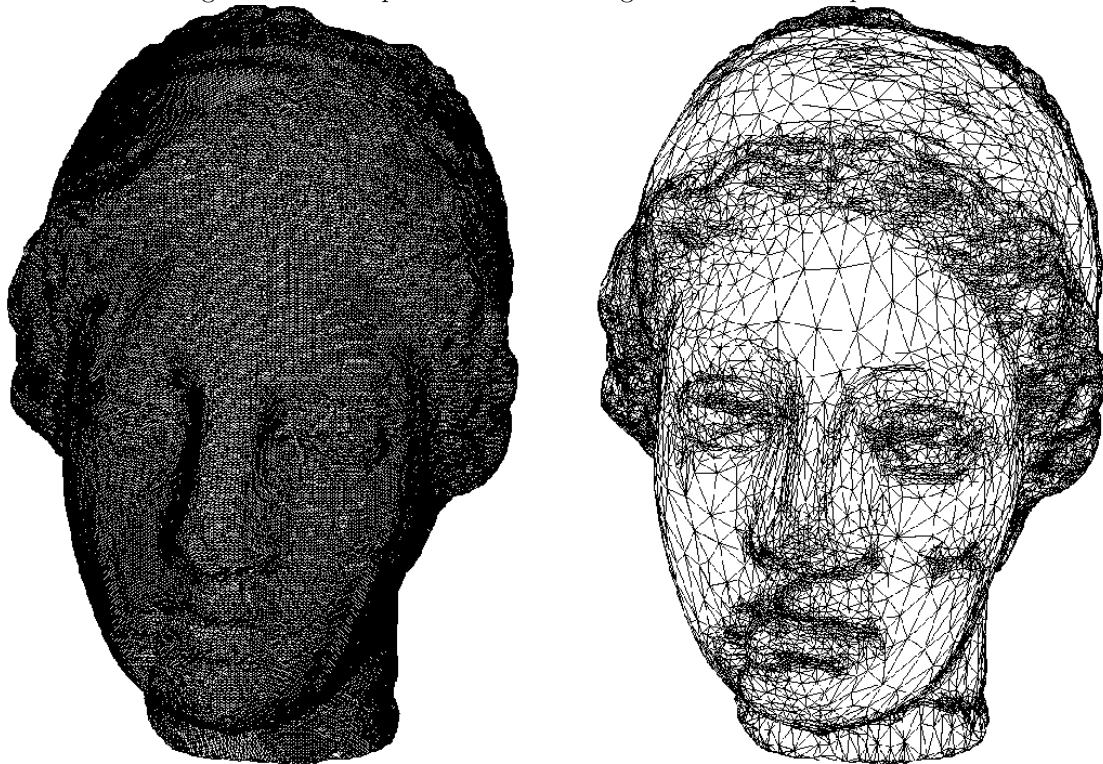
⇒ only smoothing and edge swapping has been used to improve the mesh quality.

5.1.2.2 Using ridge angle and maximal chordal deviation

When the ridge angle α is set to 50 degrees and the maximal chordal deviation is set to 0.05, the main differences with the previous case are that

- for the small subdomain, one of its points is no more considered by MeshGems-SurfOpt a corner (effect of a greater angle of ridge value);
- a stronger constraint has been set for the curvature rendering (effect of the new maximal chordal deviation value).

Figure 5.3: Example of finite element geometric mesh simplification.

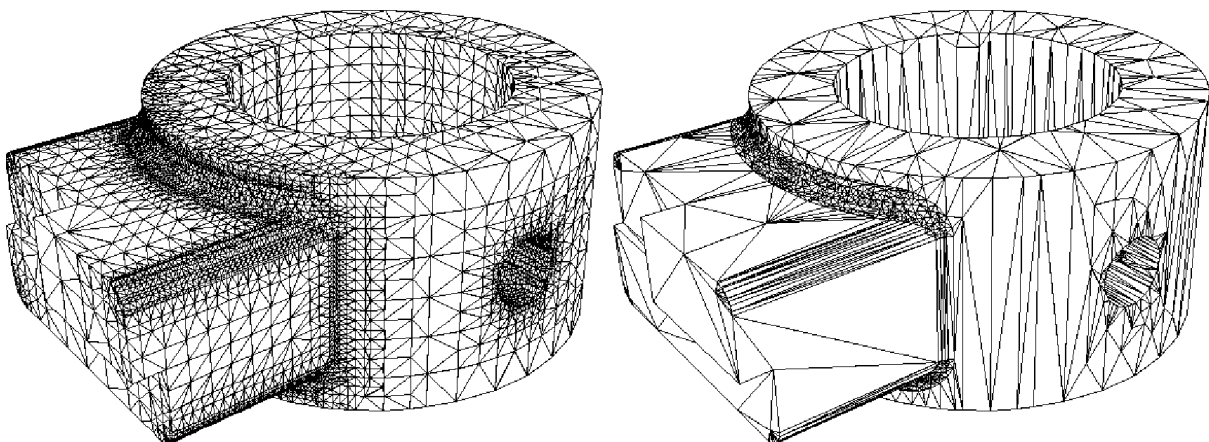


Left: original mesh

Right: simplified mesh

Parameters: $\varepsilon = 0.16$, $\alpha = 45$ degrees, Relative Values, $\delta = 10$
 MeshGems-SurfOpt parameters: `--geometric_approximation_angle 0.16`
 `--ridge_angle 45`
 `--chordal_error 0.01r`

Figure 5.4: Example of geometric mesh simplification for visualisation purposes.



Left: original mesh

Right: simplified mesh

Parameters: $\alpha = 45$ degrees, Relative Values, $\delta = 1$
 MeshGems-SurfOpt parameters: `--ridge_angle 45`
 `--chordal_error 0.001r`



One can see that:

- for “`--optimisation` only”, there is no visible difference (although the number of corners detected has changed);
- for “`--enrich` yes”, MeshGems-SurfOpt has refined the top part of the small subdomain as a curve (point no more considered a corner) and that it has refined more than with the default values because of the stronger constraint on the chordal deviation;
- for “`--enrich` yes” with “`--Hausdorff_like` yes”, MeshGems-SurfOpt has flattened the top part of the small subdomain.

5.2 Surface simplification

One of the most important features of MeshGems-SurfOpt is the possibility to reduce the number of mesh entities while controlling:

- the quality of the geometric approximation,
- the surface properties (curvatures, ridges, etc.) and
- the global mesh quality.

Numerous fields of application are concerned with this possibility, including computer graphics, cartography, virtual reality, numerical simulation, etc.

Given a surface triangulation, two types of mesh simplifications procedures are proposed in the software package, allowing the construction of:

- finite element surface meshes (meshes that are primarily intended for numerical simulations, having a elements with a good aspect ratio, see §3.1.1)
- geometric meshes (more suitable for visualization purposes).

5.2.1 Curvature-controlled coarsening: “`--enrich` no”

With this option (“`--enrich` no”), the original surface triangulation is simplified according to the intrinsic size map (automatically computed by the program). The goal is to get a smaller geometric mesh (good geometric approximation) in which the element shapes and sizes are controlled. Two parameters enables to control

1. the chordal deviation (“`--chordal_error` δ ”)
2. the surface smoothness (“`--geometric_approximation_angle` ε ”).

Typically, this option triggers the creation of finite element geometric meshes in which the density of the elements is related to the local curvatures of the surface. The size gradation is controlled by the parameter `--gradation`.

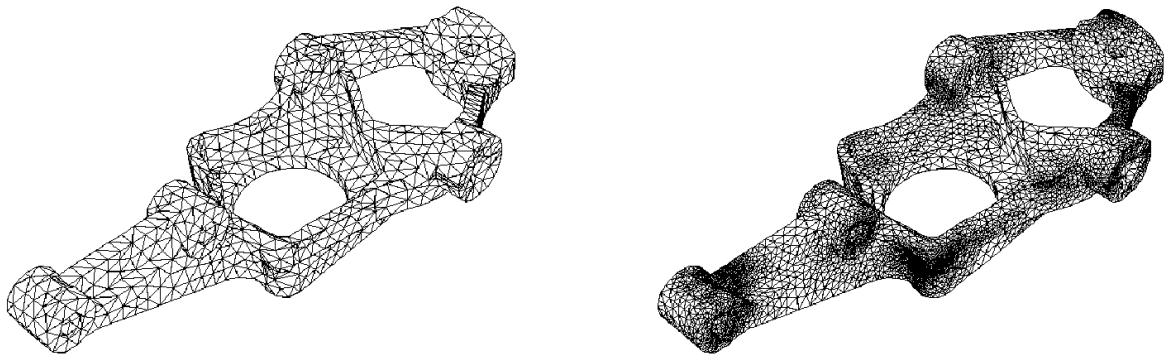
See Figure 5.3, ‘Example of finite element geometric mesh simplification’ on page 45.

5.2.2 Distance controlled coarsening: “`--enrich` no” and “`--Hausdorff_like` yes”

Using the option “`--Hausdorff_like` yes” allows the user to generate a surface approximation that is guaranteed to deviate from the initial surface by no more than a user-specified amount while preserving the topology. The tolerance can be seen as a *protective envelope* surrounding the original triangulation.

This approach usually results in geometric meshes, in which the geometric approximation is well preserved, but the mesh quality (in terms of aspect ratio) may be significantly degraded. It is typically used for creating small meshes for visualization purposes. The resulting meshes usually contain *anisotropic* elements (*i.e.*, stretched elements aligned with the main directions of curvature). As the gradation is disabled to somehow bound the aspect ratio of the elements (useful when defining a geometry), the user can use the `--max_size` h_{\max} parameter.

Figure 5.5: Example of geometric mesh enrichment.



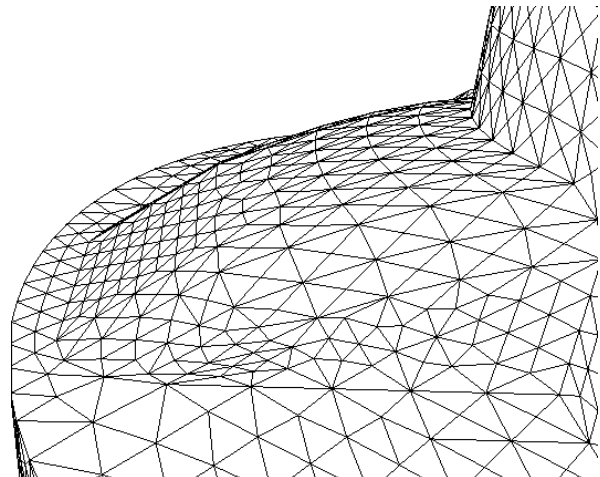
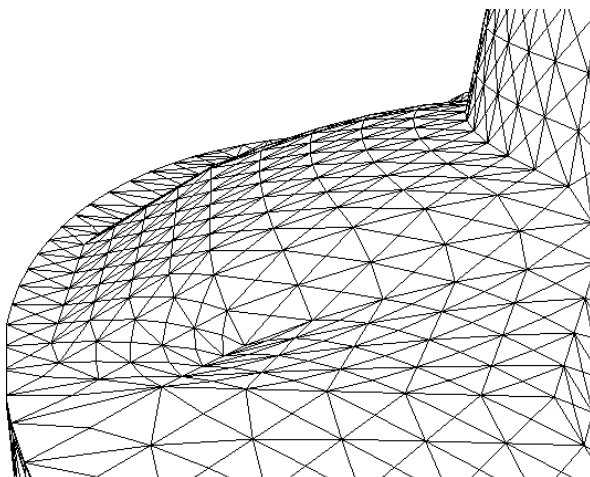
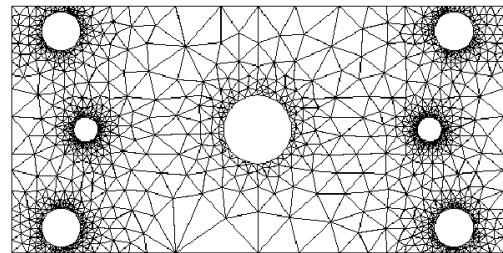
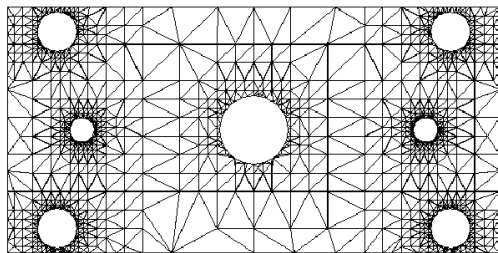
Left: original mesh

Right: simplified mesh

Parameters: $\varepsilon = 0.04$, $\alpha = 45$ degrees, Relative Values, $\delta = 1$

MeshGems-SurfOpt parameters:
`--geometric_approximation_angle 0.04`
`--ridge_angle 45`
`--chordal_error 0.001r`

Figure 5.6: Examples of mesh optimisation in two and three dimensions.



original mesh

optimised mesh



See Figure 5.4, ‘Example of geometric mesh simplification for visualisation purposes’ on page 45.

5.3 Surface mesh enrichment: “`--enrich yes`” and “`--Hausdorff_like yes`”

In some cases, it may be desirable to enrich a given surface mesh, for instance to improve the geometric approximation of the surface. This operation requires the definition of a geometric support (*i.e.* the analytical definition of the underlying geometry) in order to control the point insertion procedure. MeshGems-SurfOpt uses the given triangulation as a representation of the geometry and creates new points in regions where the discretization is too far from the geometry.

See Figure 5.5, ‘Example of geometric mesh enrichment’ on page 47.

5.4 Surface mesh optimisation: “`--optimisation only`”

MeshGems-SurfOpt allows one to improve the quality of any given surface triangulation, while preserving the geometric approximation. Indeed, most of the numerical simulation software packages (especially based on finite elements/volumes techniques) are very sensitive to the quality of the meshes (to some extent, depending on the solver used). The quality of the surface mesh may also affect the quality of the 3D mesh if a 3D tetrahedral mesh generation algorithm is to be used.

A straightforward way of optimising a mesh is to keep the number of mesh entities unchanged and simply make *local* modifications so as to improve the quality. In this case, geometric operations (node relocation) and topological changes (edge flipping) are involved. In any case, **the number of elements and points are most of the time kept by the algorithm.**

See Figure 5.6, ‘Examples of mesh optimisation in two and three dimensions’ on page 47.



Appendices

Appendix A

MeshGems-SurfOpt's diagnostics

Upon completion of an execution of MeshGems-SurfOpt, several cases may occur:

1. No error or warning occurred during execution of MeshGems-SurfOpt and all the output files have been created: the resulting mesh is most probably correct.
2. No output file was produced:
 - either the data is incorrect (or missing), or
 - the algorithm detected an unrecoverable error and stopped.

If an error occurs, it can be:

1. some input file does not exist, is not readable or writable;
2. some input file content is not correct;
3. there is a memory problem, usually set maximum is too little and needs to be increased;
4. conflicting arguments exists;
5. the programme has detected an inconsistency in the data;
6. a license seat has not been granted;
7. the algorithm has failed.

Warnings, errors and possible cause of failure are identified and listed in the next paragraphs.

A.1 Syntax of the exe diagnostics

During the execution of MeshGems-SurfOpt, 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 :

$$code = \pm(1\,000\,000 \times \mathbf{q} + \mathbf{r})$$

where:

- q**: identifies the module sending the message; **q** is equal to **6** for MeshGems-SurfOpt;
- r**: is the base number of the message between $0 \leq r < 10000$; **r** is independent of the module.



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

We give below an example of message and its meaning:

```
MGMESSAGE  -6002341  2 3 11 0
```

The number of this message is -6002341. This is an error message (the code number is negative). Here **q** is 6 which means that this is an error coming from MeshGems-SurfOpt (error 2341). This message has two integer data attached which are vertex indices "3" and "11", and no real data attached.

A.2 List of MeshGems-SurfOpt's diagnostics

The list of available diagnostics with non null code number appears below, and, whenever possible, a way of correcting the problem is also provided. For simplicity reason we only list the base of each diagnostic number. For example -6002341 would appear in the list as 2341.

- **XXXX** : Short explanation of the diagnostic.
type : Error, information or both
integer data count : give the number **ni** of integer data
integer data [1] : give the first integer data, or none
integer data [2] : give the second integer data
... :
integer data [ni] : give the last integer data
real data count : give the number **nr** of real data
real data [1] : give the first real data, or none
real data [2] : give the second real data
... :
real data [nr] : give the last real data
cure : the cure if applicable.

A.2.1 I/O errors

- **1** : The code is unable to treat a data file.
type : error
cure : Check message indications, read/write permissions for file and/or directory, ...

A.2.2 Data provided incorrect

- **2110** : No vertex found in the input mesh.
type : error
cure : Check the mesh input file.
- **2120** : No element found in the input mesh.
type : error
cure : Check the input file (refer to §4.1).
- **2200** : Invalid sizemap.
type : error
cure : Check requirements and sizemap.
- **2211** : A vertex index is out of range.
type : information
cure : Check the mesh input file.
- **2220** : The face in the mesh file is ill defined.
type : information
cure : Check the mesh input file (refer to §4.1.1).
- **2221** : Total number of faces discarded.
type : information
cure : Check the mesh input file (refer to §4.1.1).
- **2240** : Invalid edge discarded: index is out of bounds.
type : information
cure : Check the mesh input file.
- **2241** : count of dangling edges found.
type : information
- **2242** : dangling edge has been found.
type : information
- **2270** : The sizemap provided is not a metric tensor.
type : error
cure : Check sizemap.
- **2341** : Inconsistent tags on edge vertices.
type : information
cure : Check your input surface.
- **2370** : Sizemap unsuitable.
type : error
cure : Check if the sizemap file is readable and its content valid.

A.2.3 Memory

- **3000** : Not enough memory space to store the mesh structure.
type : error
cure : Increase maximum memory with option “`--max_memory (memory)`” (refer to §4.4.4).
- **3000** : The program was unable to allocate heap memory space.
type : error
- **3000** : The program was unable to extend memory space.
type : error



- **3001** : not enough memory to split edges. operation cancelled.
type : information
cure : Use flag “`--max_memory (memory)`” to increase size (refer to §4.4.4).
- **3040** : The set of edges is discarded (memory problem not related to “`--max_memory (memory)`”).
type : information
- **3200** : not enough memory to create new point/triangle.
type : error
cure : Use flag “`--max_memory (memory)`” to increase size (refer to §4.4.4).
- **3201** : insufficient memory to create new point/triangle.
type : information
cure : Use flag “`--max_memory (memory)`” to increase size (refer to §4.4.4).

A.2.4 Data provided not suitable for module

- **4000** : unknown option, missing or invalid value.
type : error
cure : Use indication and resolve conflict.
- **4200** : bad value for parameter.
type : error
cure : Use indication and resolve conflict.
- **4201** : The user is warned that some parameter values were overridden.
type : information
- **4300** : Incompatible options found.
type : error
cure : Use indication and resolve conflict.
- **4301** : invalid range
type : information
cure : Use indication and resolve conflict.
- **4370** : Incompatible option and sizemap.
type : error
cure : Resolve conflict.
- **4800** : Value for option not implemented yet.
type : error
cure : If it is not a misuse, contact hotline.

A.2.5 Analysis

- **5240** : error on edge (i,j) of face k
type : error
cure : Check your input surface topology.
If it is correct, contact hotline.
- **5241** : problem on edge (i,j): face k discarded.
type : information
cure : Check your input surface topology.
If it is correct, contact hotline.

- **5260** : Adjacency problem for triangles i and j.
type : error
cure : Check your input surface topology.
If it is correct, contact hotline.
- **5310** : the angles between facets around these vertices exceed more than once the angle of ridge value (like in a cone): these vertices will be tagged and treated as corners.
type : information
cure : Correct mesh in particular by refining the region around one such vertex that is not a corner, enrich mesh data (defining mesh ridges in particular) or tune parameters adequately.
- **5350** : the surface mesh is not orientable as it is.
type : information
cure : MeshGems-SurfOpt defines new ridge edges to correct the problem as it scans them. It is usually better for the user to define such ridges himself.
- **5351** : the tangents around these non-ridge edges differ too much: the tangents have been corrected.
type : information
cure : Correct mesh in particular by refining these regions, enrich mesh data (defining mesh ridges in particular) or tune parameters adequately.
- **5560** : there is no seed for connected component i.
type : information
cure : Check if the mesh contains at least that number of components.
- **5620** : the element i has a poor shape quality.
type : information
cure : Correct mesh or tune parameters adequately.
- **5750** : the curve is not well defined internally during point creation.
type : error
cure : Correct mesh, enrich mesh data (lower element aspect ratio in these regions in particular) or tune parameters adequately.

A.2.6 Licensing

- **6000** : License not found, not available, incorrect or expired.
type : error
cure : Check the license mechanism you use: features; expiration dates; free seats; license server up, running and accessible; ...

A.2.7 Module

- **7000** : an abnormal situation has been caught.
type : error
cure : Check the message and try to correct the situation.
If everything seems correct on your side, contact hotline.
- **7220** : the set of connected triangles around vertex i is incorrect.
type : information
cure : Check mesh.
- **7240** : an edge is not swappable, is ill defined or MeshGems-SurfOpt was not able to process it.
type : error
cure : Check and correct the topology.



- **7400** : an internal error has been encountered or a signal has been received. Current mesh will not be saved.
type : error
cure : Contact hotline.
- **7570** : there was a problem while computing the 3D proximity; continuing without it.
type : information
cure : It might be an internal problem but check mesh correctness nonetheless.

A.2.8 Result

- **8200** : the current mesh is most likely invalid (non conformal).
type : information
cure : check previous error messages.

A.2.9 Statistics

- **9000** : Program has received an interruption or a termination signal sent by the user or the system administrator. Current mesh will not be saved.
type : error
cure : Not applicable.
- **9460** : some topology configuration prevented mesh correction.
type : information
cure : No cure available as this usually means something has not been done to protect you mesh; however it is advisable to check mesh.
- **9620** : element i has a very small inradius x (badly-shaped).
type : information
cure : Correct mesh.
- **9621** : quadrilateral i has a very bad quality or is very twisted.
type : information
cure : Correct mesh.
- **9720** : element i has a very small surface x (badly-shaped).
type : information
cure : Correct mesh (scaling may sometimes be enough in particular when the bounding box size is around or below $1e - 5$).
- **9750** : the curve is not well defined internally during point creation.
type : information
cure : Correct mesh, enrich mesh data (lower element aspect ratio in these regions in particular) or tune parameters adequately.
- **9760** : normal variation in element i is x degrees.
type : information
cure : Correct mesh, enrich mesh data (defining mesh ridges in particular) or tune parameters adequately.
- **9761** : the angle between facet and curve is too big and may lead to errors.
type : information
cure : Correct mesh in particular by refining this region.

Appendix B

File format description

In this chapter, we will briefly describe two types of data files used with MeshGems-SurfOpt: the surface mesh format “`mesh`” and the size map format “`sol`”.

Important notice: For a complete description of these formats as well as the complete list of the keywords used by the MeshGems[®] 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-SurfOpt.

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
```

...



```
691
692 # Set of mesh triangles (v1,v2,v3,tag)
693 Triangles
694 1364
695 1 2 3 1
696 4 2 1 1
```

...

```
4730
4731 End
```

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-SurfOpt 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 (e.g., 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-SurfOpt.

Then comes the list of sizes associated to the entities of the supporting mesh (in the case of MeshGems-SurfOpt, 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
```

...

```
170 6.95805
171 6.95805
172 6.9673
173 6.95805
174 6.95805
175
176 End
```

B.3 Keywords used by MeshGems-SurfOpt

The first characters indicate if the keyword is read (R) and/or written (W) by MeshGems-SurfOpt.

- R/W – Corners
- R/W – Dimension
- R/W – Edges
- R/W – End
- R/W – MeshVersionFormatted
- R/W – NormalAtQuadrilateralVertices
- R/W – NormalAtTriangleVertices
- R/W – NormalAtVertices
- R/W – Normals
- R/W – Quadrilaterals
- R/W – RequiredEdges
- R/W – RequiredVertices
- R/W – Ridges
- R/W – Triangles
- R/W – Vertices
- R/- – RequiredQuadrilaterals
- R/- – RequiredTriangles
- -/W – ExtraVerticesAtEdges
- -/W – ExtraVerticesAtTriangles
- -/W – Tangents
- -/W – TangentsAtEdges



Appendix C

List of FAQ's

In this section, we will attempt to answer some frequently asked questions about surface remeshing with MeshGems-SurfOpt.

C.1 How do I get a mesh suitable for computations?

Use

```
mg-surfopt.exe --in inputfile
```

to force the evaluation of the intrinsic (geometric) size map and the optimisation of element shape and quality. However, if you wish to improve the mesh without changing the number of nodes, you may also only optimise the element shapes by running:

```
mg-surfopt.exe --enrich no --optimisation only --in inputfile
```

on the mesh.

C.2 How do I reduce the size of a large mesh?

Use the “`--min_size`” parameter (see §4.4.6.4.1) in conjunction with the coarsening optimisation style “`--enrich no`” (see §4.4.6.1) like so:

```
mg-surfopt.exe --enrich no --min_size hmin ...
```

and set the h_{\min} value high enough.

In order to respect the geometric approximation and the tolerance, MeshGems-SurfOpt will only coarsen regions of high curvature up to a certain degree.

C.3 How can I avoid node relocation?

Add the “`--optimisation no`” flag on the MeshGems-SurfOpt command line (see §4.4.6.1.2) to prevent the mesh nodes to be smoothed.

```
mg-surfopt.exe --optimisation no ...
```



C.4 How can I preserve some specific mesh entities?

In a mesh file use the fields `RequiredVertices`, `RequiredEdges` and `RequiredTriangles` to set and enforce mesh entities. Notice also that a corner is a point that cannot be modified.

C.5 How do I improve noisy data?

If the vertices do not all belong to the surface, set the “`--chordal_error`” parameter to an adequate value (higher than the noise level) and run MeshGems-SurfOpt with “`--Hausdorff_like`”yes. Once the mesh has been cleaned up, use “`--enrich` yes” with “`--Hausdorff_like`”no to get a finite element mesh.

C.6 How can I remove small features?

Set the parameter “`--min_size`” to a suitable value. Notice that this value is used to truncate the intrinsic metric. Hence, if the value is too high, the geometry of the surface can be slightly changed.

C.7 How do I proceed with scanned data?

Usually scanned data are quite noisy. Therefore use

```
mg-surfopt.exe --enrich no --Hausdorff_like yes ...
```

to extract a geometric mesh with a suitable value of the “`--chordal_error`” parameter. Do not hesitate to perform this operation twice (or even more), with small tolerance values (twice with a tolerance of 1. is usually better than one single run with a tolerance of 2!). Then, run

```
mg-surfopt.exe --enrich yes ...
```

to get a better mesh quality.

If the mesh contains ”staircases”-like structures, it may help to use

```
mg-surfopt.exe --sand_paper yes ...
```

to smooth out the mesh.

C.8 How can I control rapid size variations?

The mesh size variation between adjacent elements is controlled by the “`--gradation ψ` ” parameter. Its default value is $\psi = 1.3$ and it should be fine in most cases.

To get a mesh coarser than the default value, use $\psi = 2$:

```
mg-surfopt.exe --gradation 2 ...
```

Use $\psi = 1.2$ to get a finer mesh. $\psi = -1$ turns off mesh gradation control altogether.

C.9 How do I visualize surface meshes?

As other meshing components, MeshGems-SurfOpt does not have a graphical user interface (GUI). However, the mesh file is easy to read and can thus be sent out to standard graphic software packages. A natural companion of MeshGems-SurfOpt is Medit, a 3D interactive mesh display software (INRIA-Simulog).

C.10 What should I do about MeshGems-SurfOpt curve warnings?

You get "curve tangent" or "curve definition" warnings when there are probably not enough triangles and nodes in the surface mesh for MeshGems-SurfOpt to evaluate curves.

Try using the "`--create_nodes_on_flat_edge`" option set to `no` (see §4.4.6.9) so as to create new nodes on the existing edges of the mesh and hence avoiding creation of new nodes on an ill defined curve.

C.11 How can I control ridge detection?

You can use the "`--ridge_angle`" parameter. If, when you change this parameter, MeshGems-SurfOpt detects too many or too few ridges, you are probably using the "`msh2`" format or the "`mesh`" format without specifying the mesh ridges.

You can correct this by using the "`mesh`" format indicating relevant known geometrical structures of the mesh, including ridges, corners, normals, tangents, ... (see §4.1.1). Then, run

```
mg-surfopt.exe --compute_ridges no ...
```

to avoid a computation of other ridges by MeshGems-SurfOpt.

C.12 How can I ensure a surface size on a plane, for example for an imposed flow rate boundary condition?

In some situations, one may want to ensure that the surface of an area defined by a set of triangles (for example all of a given attribute) will be the same, although the surface was remeshed by MeshGems-SurfOpt. This may happen for example if numerical values were computed for the original mesh based on the surface, and you want to use the same values for the resulting mesh.

To achieve this, you need to specify as required all edges which are on the boundary of the attribute you wish to preserve. This will ensure that the total surface will be the same as before.



MeshGems

User Manual

MG-SURFOPT

Release 1.1

APPENDIX C. LIST OF FAQ'S

Appendix D

List of renamed options

In the harmonization process of options for all MeshGems[®] Suite components, several options have changed between Yams, MeshGems-SurfOpt's ancestor, and the current version of MeshGems-SurfOpt.

Table D.1: Correspondence table between old Yams and MeshGems-SurfOpt options.

Old Yams options	MeshGems-SurfOpt options	Remarks
	<code>--in < infile ></code>	First non option argument was used as the input file name.
	<code>--read_sizemap < insizefile ></code>	Input file suffix was replaced by “.bb”, Yams would read the file if present.
	<code>--out < outfile ></code>	Second non option argument was used as the output file name; if absent a “.d” was inserted before the suffix.
	<code>--write_sizemap < outsizefile ></code>	Output file suffix was replaced by “.bb”, Yams would write the file if one of options <code>-b</code> or <code>-B?</code> was present.
<code>-h</code>	<code>--help</code>	
<code>-v < level ></code>	<code>--verbose < level ></code>	
<code>-m < maxmem ></code>	<code>--max_memory < maxmem ></code>	
<code>-O 1</code>		default mode.
<code>-O 2</code>	<code>--Hausdorff_like yes</code>	
<code>-O 0</code>	<code>--optimisation only</code>	
<code>-O -1</code>	<code>--enrich no</code>	
<code>-O -2</code>	<code>--Hausdorff_like yes --enrich no</code>	
<code>-O U</code>	<code>--uniform_flat_subdivision yes</code>	
<code>-Dsubdiv=< div ></code>	<code>--uniform_subdivide_level < div ></code>	
<code>-O S</code>	<code>--sand_paper yes</code>	
<code>-I / -Diso</code>	<code>--metric iso</code>	
<code>-A / -Daniso</code>	<code>--metric aniso</code>	

continued on next page



Old Yams options	MeshGems-SurfOpt options	Remarks
<code>-nm</code>	<code>--create_nodes_on_flat_edge yes</code>	
<code>-nr</code>	<code>--compute_ridges no</code>	
<code>-ns</code>	<code>--optimisation no</code>	
<code>-p < targ ></code>	<code>--target_vertex_count < targ ></code>	
<code>-q</code>	<code>--element_generation full_quad</code>	
<code>-noREF</code>	<code>--tags clear</code>	
<code>-noFEM</code>	<code>--split_overconstrained_elements no</code>	
<code>-FEM</code>	<code>--split_overconstrained_elements yes</code>	
<code>-Dridge=< α ></code>	<code>--ridge_angle < α ></code>	
<code>-Dgeomapp=< ϵ_{\max} ></code>	<code>--geometric_approximation_angle < geomapp ></code>	Relationship is: $\langle \epsilon_{\max} \rangle = 1 - \cos \langle geomapp \rangle$
<code>-Dtolerance=< tol ></code>	<code>--chordal_error < tol ></code>	
<code>-Dmaxsize=< maxsize ></code>	<code>--max_size < maxsize ></code>	
<code>-Dminsize=< minsize ></code>	<code>--min_size < minsize ></code>	
<code>-Dgradation=< ψ ></code>	<code>--gradation < ψ ></code>	
<code>-Drelative</code>		'r' appended to the dimension values as the same effect, such as <code>--min_size 0.005r</code> .
<code>-Dabsolute</code>		Default.
<code>-Dridge_on_ref</code>	<code>--ridge_on_reference_lines yes</code>	
<code>-Dsize_is_nodal</code>	<code>--sizemap_is_nodal yes</code>	
<code>-Dsplitedge=1</code>	<code>--element_order quadratic</code>	
<code>-DQlevel=< Qlevel ></code>	<code>--remove_bad_input_surface_elements yes</code> <code>--bad_input_element_aspect_ratio < Qlevel ></code>	Deactivated by default now.
<code>-Dprox2D=< π_{2d} ></code>	<code>--use_surface_proximity yes</code> <code>--surface_proximity_layers < π_{2d} ></code>	
<code>-Dprox2Dgrad=< ψ_{2d} ></code>	<code>--use_surface_proximity yes</code> <code>--surface_proximity_gradation < ψ_{2d} ></code>	
<code>-Dprox3D=< π_{3d} ></code>	<code>--use_volume_proximity yes</code> <code>--volume_proximity_layers < π_{3d} ></code>	
<code>-Dprox3Dgrad=< ψ_{3d} ></code>	<code>--use_volume_proximity yes</code> <code>--volume_proximity_gradation < ψ_{3d} ></code>	
<code>-f</code>		Useless.
<code>-vrml</code>		Removed.

Appendix E

Release Notes

Released versions are described below.

E.1 Version 1.1

This version is the first version of MeshGems-SurfOpt, which is part of the MeshGems[®] Suite and which replaces Yams. The changes and new features introduced by this new module are detailed below.

E.1.1 New features

- **Common framework:** MeshGems-SurfOpt now uses MeshGems[®] Suite's common framework: same convention for options, same input and output file formats, same diagnostic codes, . . . The common API is also available now. It allows working on a mesh without going through files and having a finer control of the treatments applied.
- **Error diagnostics:** the error diagnostic system has been unified for MeshGems[®] Suite, so MeshGems-SurfOpt error numbers have changed. The messages are now easier to parse and may contain formatted additional data. If you wish to temporarily use the old error system, see §4.4.3.3, page 30.

E.1.2 Behavior changes

- **File overwriting forbidden** now. Using an existing file as output is forbidden to prevent accidental overwriting.
- **Suffix needed for filenames.** Yams use to try several suffixes but it was confusing if several files with the same base name existed. The “`mesh`” suffix is now mandatory for both input and output files.
- **Metric map not read by default** anymore, even if they have the same base name as the input file. It was confusing not to be sure the size map has been read or not, and now it allows size maps for the same mesh to be switched at the command line level without having to copy input mesh and size map into files with the same base name. See “`--read_sizemap < insizefile >`” in §4.4.5.3, page 32.
- **Option names and parsing mechanism** have changed and makes old scripts unusable. New names may be found in §D, page 65. If you wish to temporarily use the old system, most of the old options will still work for now but changing back to the old parsing mechanism requires an additional option: see §4.4.3.4, page 30.



- **No interactive mode** anymore. It was unused.
- **No VRML output** anymore. It was unused.
- **2D meshes not supported** anymore. It is quite easy to convert the mesh in three dimensions by adding a third coordinate $z = 0$, apply MeshGems-SurfOpt and convert back the mesh in 2D by restricting to only the first two coordinates.
If necessary, 2D meshes could be added in a future version.

E.1.3 Improvements

- **Full compatibility with the other tools of the MeshGems[®] Suite.**
- Various minor improvements.

E.1.4 Corrections

- Corrections of several anomalies.

Appendix F

References

- [1] SIMULOG license server. User Manual Release 1.0 (1998) .
- [2] SIMULOG's license server, WINDOWS/NT environnement. User's Manual Version 1.1 (1999).
- [3] TetMesh-GHS3D, User's Manual, Version 3.2, April 2002.
- [4] H. Borouchaki, F. Hecht, P.J. Frey (1998), Mesh gradation control, *Int. J. Numer. Methods Eng.*, 43(6), 1143-1157
- [5] H. Borouchaki, F. Hecht, E. Saltel, P.L. George (1995), Reasonably efficient Delaunay based mesh generator in 3 dimensions, in 4th International Meshing Roundtable, Albuquerque, New Mexico, pp. 3-14.
- [6] E. Brière De L'Isle, P.L. George (1995), Optimization of tetrahedral meshes, *IMA Volumes in Mathematics and its Applications*, ed. I. Babuska, W.D. Henshaw, J.E. Oliger, J.E. Flaherty, J.E. Hopcroft and T. Tezduyar, vol 75, pp. 97-128.
- [7] P.J Frey ET P.L. George (1999), *Maillages. Applications aux éléments finis*, Hermes, Paris. Also as
- [8] P.J Frey ET P.L. George (2000), *Mesh Generation*, Hermes Science Publications.
- [9] P.J. Frey , H. Borouchaki (1999), Surface mesh quality evaluation, *Int. J. Numer. Engng* 45.
- [10] P.J. Frey , H. Borouchaki (1998) Geometric surface mesh optimization, *Computing and Visualisation in Science*, 1, 113-121.
- [11] P.L. George, F. Hecht, E. Saltel (1990), Fully automatic mesh generator for 3d domains of any shape, *Impact of Comp. in Sci. and Eng.*, 2, pp 187-218.
- [12] P.L. George (1991), Génération automatique de maillage. *Applications aux méthodes d'éléments finis*, RMA 16, Masson Paris. Also as
- [13] P.L. George (1991), *Automatic mesh generation. Applications to finite element methods*, Wiley.
- [14] P.L. George, F. Hecht, E. Saltel (1991), Automatic mesh generator with specified boundary, *Comp. Meth. in Appl. Mech. and Eng.*, vol 92, pp. 269-288.
- [15] P.L. George, F. Hecht, M. G. Vallet (1991), Creation of internal points in Voronoi's type method, *Control and adaptation*, *Adv. in Eng. Soft.*, 13, n°5/6,, pp. 303-313.
- [16] P.L. George, F. Hermeline (1992), Delaunay's mesh of a convex polyhedron in dimension d. Application to arbitrary polyhedra, *Int. Jour. Num. Meth. Eng.*, vol 33, pp. 975-995.
- [17] P.L. George (1996), Improvement on Delaunay based 3D automatic mesh generator, *Finite Elements in Analysis and Design*, vol 25(3-4), pp. 297-317.



- [18] P.L. George ET H. Borouchaki (1997), Triangulation de Delaunay et maillage. Applications aux éléments finis, Hermes, Paris. Also as
- [19] P.L. George AND H. Borouchaki (1998), Delaunay triangulation and meshing. Applications to finite elements, Hermès.
- [20] P.L. George, H. Borouchaki (1998), Génération automatique de maillages tridimensionnels respectant une carte de taille, Revue européenne des éléments finis, 7(4), 339-363.
- [21] P.S. Heckbert, M. Garland (1997), Survey of polygonal surface simplification algorithms, in Multiresolution surface modelling course, Siggraph'97.
- [22] P. J. Frey. About surface remeshing. Proceedings of 9th International Meshing Roundtable, pages 123–136, 2000.

List of Figures

3.1	Definitions: chordal deviation ξ ; curvature $1/r$; bounding box s ; angles α and β	20
5.1	Examples of optimisation and coarsening (default values)	42
5.2	Examples of optimisation and coarsening/enrichment (default values)	43
5.3	Example of finite element geometric mesh simplification.	45
5.4	Example of geometric mesh simplification for visualisation purposes.	45
5.5	Example of geometric mesh enrichment.	47
5.6	Examples of mesh optimisation in two and three dimensions.	47



List of Tables

D.1 Correspondence table between old Yams and MeshGems-SurfOpt options. 65