

# GoUpScale Developer Documentation

Marc-Olivier Titeux - 21 Septembre 2009

Gocad Research Consortium – Centre de Recherches Géochimiques et Pétrographiques

Institut National Polytechnique de Lorraine

Generated with Doxygen 1.6.1





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Requirements	1
1.2	Presentation	1
1.3	How To Use GoUpScale plugin	1
1.4	Copyright	2
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List	3
<b>3</b>	<b>Data Structure Index</b>	<b>5</b>
3.1	Class Hierarchy	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List	9
<b>6</b>	<b>Namespace Documentation</b>	<b>11</b>
6.1	Gocad Namespace Reference	11
6.2	GoUpScale Namespace Reference	12
6.2.1	Variable Documentation	12
6.2.1.1	CSCell	12
6.2.1.2	Experiments	12
6.2.1.3	FSCell	12
6.2.1.4	ShearN	13
6.2.1.5	ShearU	13
6.2.1.6	ShearV	13
6.2.1.7	TractionN	13
6.2.1.8	TractionU	13
6.2.1.9	TractionV	13

<b>7</b>	<b>Data Structure Documentation</b>	<b>15</b>
7.1	GoUpScale::CSCell Class Reference	15
7.1.1	Detailed Description	15
7.1.2	Constructor & Destructor Documentation	16
7.1.2.1	CSCell	16
7.1.2.2	~CSCell	16
7.1.3	Member Function Documentation	16
7.1.3.1	build_fine_description	16
7.1.3.2	cell_index	22
7.1.3.3	get_children	22
7.1.3.4	set_cell_index	22
7.1.3.5	set_materials	22
7.1.3.6	solid_model	23
7.1.3.7	voxets_link	23
7.2	Experiment Class Reference	24
7.2.1	Detailed Description	24
7.3	GoUpScale::Experiments Class Reference	25
7.3.1	Detailed Description	26
7.3.2	Constructor & Destructor Documentation	26
7.3.2.1	~Experiments	26
7.3.2.2	Experiments	27
7.3.3	Member Function Documentation	27
7.3.3.1	boundary_conditions	27
7.3.3.2	comp_state	27
7.3.3.3	compute_means	27
7.3.3.4	computed	28
7.3.3.5	direction	29
7.3.3.6	displacement	29
7.3.3.7	flush	29
7.3.3.8	get_strain_means	29
7.3.3.9	get_stress_means	29
7.3.3.10	init	30
7.3.3.11	prob_name	30
7.3.3.12	run	30
7.3.3.13	sample	31
7.3.3.14	set_prob_name	31
7.3.3.15	type	31
7.3.4	Field Documentation	31

7.3.4.1	Experiments	31
7.4	GoUpScale::FSCell Class Reference	33
7.4.1	Detailed Description	33
7.4.2	Constructor & Destructor Documentation	33
7.4.2.1	FSCell	33
7.4.2.2	~FSCell	34
7.4.3	Member Function Documentation	34
7.4.3.1	build_material_from_voxet_properties	34
7.4.3.2	build_stiffness_tensor	35
7.4.3.3	cell_index	36
7.4.3.4	get_stiffness	36
7.4.3.5	material	36
7.4.3.6	set_cell_index	36
7.4.3.7	set_material	36
7.4.3.8	tetra_list	37
7.4.3.9	tetra_list	37
7.4.3.10	voxets_link	37
7.5	GoUpScale::MechUpscaleAPI Class Reference	38
7.5.1	Detailed Description	38
7.5.2	Member Function Documentation	38
7.5.2.1	mean_prop_on_voxet	38
7.5.2.2	upscale	39
7.6	GoUpScale::S3D Class Reference	46
7.6.1	Detailed Description	46
7.6.2	Constructor & Destructor Documentation	46
7.6.2.1	S3D	46
7.6.2.2	~S3D	46
7.6.3	Member Function Documentation	46
7.6.3.1	get	46
7.6.3.2	operator()	47
7.6.4	Field Documentation	47
7.6.4.1	size2_	47
7.6.4.2	size_	47
7.6.4.3	tab_	47
7.7	GoUpScale::ShearN Class Reference	48
7.7.1	Detailed Description	49
7.7.2	Constructor & Destructor Documentation	49
7.7.2.1	ShearN	49

7.7.2.2	~ShearN	49
7.7.3	Member Function Documentation	49
7.7.3.1	boundary_conditions	49
7.7.3.2	direction	50
7.7.3.3	type	50
7.7.4	Field Documentation	50
7.7.4.1	ShearN	50
7.8	GoUpScale::ShearU Class Reference	51
7.8.1	Detailed Description	52
7.8.2	Constructor & Destructor Documentation	52
7.8.2.1	ShearU	52
7.8.2.2	~ShearU	52
7.8.3	Member Function Documentation	52
7.8.3.1	boundary_conditions	52
7.8.3.2	direction	53
7.8.3.3	type	53
7.8.4	Field Documentation	53
7.8.4.1	ShearU	53
7.9	GoUpScale::ShearV Class Reference	54
7.9.1	Detailed Description	55
7.9.2	Constructor & Destructor Documentation	55
7.9.2.1	ShearV	55
7.9.2.2	~ShearV	55
7.9.3	Member Function Documentation	55
7.9.3.1	boundary_conditions	55
7.9.3.2	direction	56
7.9.3.3	type	56
7.9.4	Field Documentation	56
7.9.4.1	ShearV	56
7.10	GoUpScale::StrainMeanVector< T > Class Template Reference	57
7.10.1	Detailed Description	57
7.10.2	Constructor & Destructor Documentation	57
7.10.2.1	StrainMeanVector	57
7.10.2.2	~StrainMeanVector	57
7.10.3	Member Function Documentation	57
7.10.3.1	aggregate	57
7.10.3.2	start_aggregate	58
7.11	GoUpScale::StressMeanMatrix< T > Class Template Reference	59

7.11.1	Detailed Description	59
7.11.2	Constructor & Destructor Documentation	59
7.11.2.1	StressMeanMatrix	59
7.11.2.2	~StressMeanMatrix	59
7.11.3	Member Function Documentation	60
7.11.3.1	aggregate	60
7.11.3.2	start_aggregate	60
7.12	GoUpScale::TractionN Class Reference	62
7.12.1	Detailed Description	63
7.12.2	Constructor & Destructor Documentation	63
7.12.2.1	TractionN	63
7.12.2.2	~TractionN	63
7.12.3	Member Function Documentation	63
7.12.3.1	boundary_conditions	63
7.12.3.2	direction	64
7.12.3.3	type	64
7.12.4	Field Documentation	64
7.12.4.1	TractionN	64
7.13	GoUpScale::TractionU Class Reference	65
7.13.1	Detailed Description	66
7.13.2	Constructor & Destructor Documentation	66
7.13.2.1	TractionU	66
7.13.2.2	~TractionU	66
7.13.3	Member Function Documentation	66
7.13.3.1	boundary_conditions	66
7.13.3.2	direction	67
7.13.3.3	type	67
7.13.4	Field Documentation	67
7.13.4.1	TractionU	67
7.14	GoUpScale::TractionV Class Reference	68
7.14.1	Detailed Description	69
7.14.2	Constructor & Destructor Documentation	69
7.14.2.1	TractionV	69
7.14.2.2	~TractionV	69
7.14.3	Member Function Documentation	69
7.14.3.1	boundary_conditions	69
7.14.3.2	direction	70
7.14.3.3	type	70

7.14.4	Field Documentation	70
7.14.4.1	TractionV	70
<b>8</b>	<b>File Documentation</b>	<b>71</b>
8.1	include/GoUpScale/gapi/mech_upscale_api.h File Reference	71
8.2	include/GoUpScale/gridding/cs_cell.h File Reference	72
8.2.1	Detailed Description	72
8.3	include/GoUpScale/gridding/fs_cell.h File Reference	73
8.3.1	Detailed Description	73
8.4	include/GoUpScale/math/strain_mean_vector.h File Reference	74
8.5	include/GoUpScale/math/stress_mean_matrix.h File Reference	75
8.6	include/GoUpScale/mechanics/experiments.h File Reference	76
8.7	include/GoUpScale/utis/assert.h File Reference	77
8.7.1	Define Documentation	77
8.7.1.1	goupscale_assert	77
8.8	src/lib/gapi/mech_upscale_api.cpp File Reference	78
8.9	src/lib/gridding/cs_cell.cpp File Reference	79
8.10	src/lib/gridding/fs_cell.cpp File Reference	80
8.11	src/lib/mechanics/experiments.cpp File Reference	81



# Chapter 1

## Introduction

### 1.1 Requirements

The GoUpScale-1.0.0 plugin requires RestorationSuite (composed of RestorationLab, Models, SolidExplorer, and FemPack), all built on Gocad-2.5.2p1

### 1.2 Presentation

This plugin intends to upscale isotropic elastic materials defined on structured regular grids (Voxel) to transversely isotropic elastic materials. For more information on theoretical side, please refer to [[Titeux and Royer, 2008](#)].

### 1.3 How To Use GoUpScale plugin

The algorithm is only restricted to voxets that share the same geometry (the cages need to be the same), for the moment. It takes as inputs, two voxets, with different resolution. On the fine meshed voxel, Young modulus and Poisson's ratio have to be specified as two scalar properties. A density property has also to be defined for computations purposes (density is upscaled during the process).

Several mechanical experiments (with boundary conditions expressed in displacements) are performed on each of the coarse-scale cells. The number and the types of these experiments can be specified in the extended dialog box. A displacement value for all the experiments can also be changed. In this way, it is also possible to change the strain formulation (from hookean to neo-hookean).

The results is computed on the coarse meshed voxel as 5 properties:

- mod\_young\_1: Young modulus in the plane of isotropy,
- mod\_young\_3: Young modulus along the normal axis,
- pois\_rat\_1: Poisson's ratio in the plane of isotropy,
- pois\_rat\_3: Poisson's ratio along the normal axis,
- mod\_shear\_3: Engineering Modulus along the normal axis.

Three examples are given in the test folder at the root of the plugin.

## 1.4 Copyright

Copyright, Association Scientifique pour la Geologie et ses Applications (A.S.G.A.), 2008, All rights reserved.

This Research Plugin is provided as-is and without warranty of any kind, express, implied or otherwise, including without limitation, any warranty of merchantability or fitness for a particular purpose. In no event shall A.S.G.A. or Earth Decision Sciences be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever resulting from loss of use, data or profits, whether or not advised of the possibility of damage, and on any theory of liability, arising out of or in connection with the use or performance of this software.

*GoUpScale* is a plugin from the Gocad consortium. You must be a sponsor of the Gocad consortium to use this plugin (Please visit <http://www.gocad.org> for more information). Note that possible bugs or misuse of this plugin may corrupt gOcad or other independent gOcad plugins.

# Chapter 2

## Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Gocad</a> . . . . .	11
<a href="#">GoUpScale</a> . . . . .	12



# Chapter 3

## Data Structure Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

GoUpScale::CSCell . . . . .	15
Experiment . . . . .	24
GoUpScale::Experiments . . . . .	25
GoUpScale::ShearN . . . . .	48
GoUpScale::ShearU . . . . .	51
GoUpScale::ShearV . . . . .	54
GoUpScale::TractionN . . . . .	62
GoUpScale::TractionU . . . . .	65
GoUpScale::TractionV . . . . .	68
GoUpScale::FSCell . . . . .	33
GoUpScale::MechUpscaleAPI . . . . .	38
GoUpScale::S3D . . . . .	46
GoUpScale::StrainMeanVector< T > . . . . .	57
GoUpScale::StressMeanMatrix< T > . . . . .	59



# Chapter 4

## Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">GoUpScale::CSCell</a> (Class for coarse-scale cells, made from coarse-scale description /a Voxet ) . . . . .	15
<a href="#">Experiment</a> . . . . .	24
<a href="#">GoUpScale::Experiments</a> . . . . .	25
<a href="#">GoUpScale::FSCell</a> (Abstract class for cellules mapping within voxets ) . . . . .	33
<a href="#">GoUpScale::MechUpscaleAPI</a> . . . . .	38
<a href="#">GoUpScale::S3D</a> (Class for managing Atom* tabs[][][]) . . . . .	46
<a href="#">GoUpScale::ShearN</a> . . . . .	48
<a href="#">GoUpScale::ShearU</a> . . . . .	51
<a href="#">GoUpScale::ShearV</a> . . . . .	54
<a href="#">GoUpScale::StrainMeanVector&lt; T &gt;</a> . . . . .	57
<a href="#">GoUpScale::StressMeanMatrix&lt; T &gt;</a> (Aggregate a matrix from the stress components of the experiments on a <a href="#">CSCell</a> ) . . . . .	59
<a href="#">GoUpScale::TractionN</a> . . . . .	62
<a href="#">GoUpScale::TractionU</a> . . . . .	65
<a href="#">GoUpScale::TractionV</a> . . . . .	68





# Chapter 5

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">include/GoUpScale/gapi/mech_upscale_api.h</a>	71
<a href="#">include/GoUpScale/gridding/cs_cell.h</a>	72
<a href="#">include/GoUpScale/gridding/fs_cell.h</a>	73
<a href="#">include/GoUpScale/maths/strain_mean_vector.h</a>	74
<a href="#">include/GoUpScale/maths/stress_mean_matrix.h</a>	75
<a href="#">include/GoUpScale/mechanics/experiments.h</a>	76
<a href="#">include/GoUpScale/utils/assert.h</a>	77
<a href="#">src/lib/gapi/mech_upscale_api.cpp</a>	78
<a href="#">src/lib/gridding/cs_cell.cpp</a>	79
<a href="#">src/lib/gridding/fs_cell.cpp</a>	80
<a href="#">src/lib/mechanics/experiments.cpp</a>	81



## **Chapter 6**

# **Namespace Documentation**

### **6.1 Gocad Namespace Reference**

## 6.2 GoUpScale Namespace Reference

### Data Structures

- class [MechUpscaleAPI](#)
- class [CSCell](#)  
*Class for coarse-scale cells, made from coarse-scale description /a Voxet.*
- class [S3D](#)  
*Class for managing Atom\* tabs[][][].*
- class [FSCell](#)  
*Abstract class for cellules mapping within voxets.*
- class [StrainMeanVector](#)
- class [StressMeanMatrix](#)  
*Aggregate a matrix from the stress components of the experiments on a [CSCell](#).*
- class [Experiments](#)
- class [TractionU](#)
- class [TractionV](#)
- class [TractionN](#)
- class [ShearU](#)
- class [ShearV](#)
- class [ShearN](#)

### Variables

- [CSCell](#)
- [FSCell](#)
- [Experiments](#)
- [TractionU](#)
- [TractionV](#)
- [TractionN](#)
- [ShearU](#)
- [ShearV](#)
- [ShearN](#)

#### 6.2.1 Variable Documentation

##### 6.2.1.1 GoUpScale::CSCell

Definition at line 43 of file cs\_cell.cpp.

##### 6.2.1.2 GoUpScale::Experiments

Definition at line 38 of file experiments.cpp.

##### 6.2.1.3 GoUpScale::FSCell

Definition at line 33 of file fs\_cell.cpp.

#### **6.2.1.4 GoUpScale::ShearN**

Definition at line 460 of file experiments.cpp.

#### **6.2.1.5 GoUpScale::ShearU**

Definition at line 360 of file experiments.cpp.

#### **6.2.1.6 GoUpScale::ShearV**

Definition at line 411 of file experiments.cpp.

#### **6.2.1.7 GoUpScale::TractionN**

Definition at line 310 of file experiments.cpp.

#### **6.2.1.8 GoUpScale::TractionU**

Definition at line 209 of file experiments.cpp.

#### **6.2.1.9 GoUpScale::TractionV**

Definition at line 260 of file experiments.cpp.



# Chapter 7

## Data Structure Documentation

### 7.1 GoUpScale::CSCell Class Reference

Class for coarse-scale cells, made from coarse-scale description /a Voxet.

```
#include <cs_cell.h>
```

#### Public Member Functions

- [CSCell](#) (GObjLink \*voxets\_link, GridIndex uvw, GridPropData \*\*prop\_grids)

*Constructor of the class.*

- virtual [~CSCell](#) ()

*Destructor.*

- const GObjLink \* [voxets\\_link](#) () const
- const GridIndex [cell\\_index](#) () const
- void [set\\_cell\\_index](#) (GridIndex uvw)
- const PtrSequence< [FSCell](#) > & [get\\_children](#) () const
- bool [build\\_fine\\_description](#) ()

*Gathering fine scale cell within one coarse scale cell and creating a mobile mesh (SolidModel) using the 8 points of each fs cell. It is done only on the first Coarse scale to upscale.*

- bool [set\\_materials](#) (GridIndex=0)

*Creates a material for each fs\_cell and assign it to corresponding tetras.*

- SolidModel \* [solid\\_model](#) ()

#### 7.1.1 Detailed Description

Class for coarse-scale cells, made from coarse-scale description /a Voxet.

##### Author:

Marc-Olivier Titeux

##### Version:

3.0

**Date:**

26-10-2007

This class describes a coarse-scale cell

Definition at line 64 of file cs\_cell.h.

## 7.1.2 Constructor & Destructor Documentation

### 7.1.2.1 GoUpScale::CSCell::CSCell (GObjLink \* voxets\_link, GridIndex uvw, GridPropData \*\* prop\_grids)

Constructor of the class.

**Parameters:**

*voxets\_link* GObjLink between the two Voxets, from different scales

*uvw* position of the VCell in voxet\_father

*prop\_grids* is a pointer on the tab for material properties

### 7.1.2.2 GoUpScale::CSCell::~~CSCell () [virtual]

Destructor.

Definition at line 57 of file cs\_cell.cpp.

```
57         {
58     fs_cells_.remove_all();
59     GObjAPI::destroy( smod_ );
60 }
```

## 7.1.3 Member Function Documentation

### 7.1.3.1 bool GoUpScale::CSCell::build\_fine\_description ()

Gathering fine scale cell within one coarse scale cell and creating a mobile mesh (*SolidModel*) using the 8 points of each fs cell. It is done only on the first Coarse scale to upscale.

**Author:**

Marc-Olivier Titeux

**Version:**

1.0

**Date:**

03-12-2007

It creates a *SolidModel* on the first *CSCell* to upscale. Dimensions of the Voxet must be modulo.

USER MANUAL REMARK

Definition at line 63 of file cs\_cell.cpp.

References GoUpScale::FSCell, goup\_scale\_assert, solid\_model(), GoUpScale::FSCell::tetra\_list(), and voxets\_link().





```

131         } else {
132             return false;
133         }
134
135         // collect the 8 points corners of the FSCell to build the 5
tetras per FS_Cell
136         //
137         //          (u,v+1,w+1)      (u+1,v+1,w+1
)
138         //          -----
139         //          /|              /|
140         //          / | (u+1,v,w+1) / |
141         //          (u,v,w+1) /-----/ |
142         //          | |              | |
143         //          | |              | |
144         //          | |              | |
145         //          | |              | |
146         //          | | (u,v+1,w)   | | (u+1,v+1,
w)
147         //          | /-----|-/
148         // w   v              | /              | /
149
150         // ^   ^              (u,v,w)   |/-----/
(u+1,v,w)
151         // | /
152         // |/-->u
153
154         // A cube is made of 5 tetras, which constructions flip accor
ding
155         // to the parity of the cell index
156         if( cellindex%2 ) {
157             // construct a tetra
158             Tetra* nt1 = tvol->create_simplex(
159                 atom_list( u, v, w ),
160                 atom_list( u+1, v+1, w ),
161                 atom_list( u, v+1, w+1 ),
162                 atom_list( u+1, v, w+1 )
163             );
164
165             if( nt1 != nil ) {
166                 // add it to the Atomic
167                 tvol->add_simplex( nt1 );
168                 // set a corresponding constraint (in the way of Fe
mPack Constraint)
169                 TetraConstraints* cnstr = new TetraConstraints;
170                 smod->set_tetra_constraints(
171                     nt1, cnstr
172                 );
173
174                 fs_cell->tetra_list(0) = nt1;
175             } else {
176                 Logger::error( Format::tr("Error in building mobile
mesh\n") );
177                 return false;
178             }
179
180
181         // A cell (cube) /home/titeux/Datas/Validating_Upscal.gp
rjcontains 5 tetras
182         Tetra* nt2 = tvol->create_simplex(
183             atom_list( u+1, v+1, w ),
184             atom_list( u, v+1, w+1 ),
185             atom_list( u+1, v, w+1 ),
186             atom_list( u+1, v+1, w+1 )
187         );
188
189         if( nt2 != nil ) {
190             tvol->add_simplex( nt2 );
191             TetraConstraints* cnstr = new TetraConstraints;
192             smod->set_tetra_constraints(

```

```

193             nt2, cnstr
194         );
195
196         fs_cell->tetra_list(1) = nt2;
197     } else {
198         Logger::error( Format::tr("Error in building mobile
199 mesh\n") );
200         return false;
201     }
202
203     Tetra* nt3 = tvol->create_simplex(
204         atom_list( u, v, w ),
205         atom_list( u, v+1, w+1 ),
206         atom_list( u+1, v+1, w ),
207         atom_list( u, v+1, w )
208     );
209
210     if( nt3 != nil ) {
211         tvol->add_simplex( nt3 );
212         TetraConstraints* cnstr = new TetraConstraints;
213         smod->set_tetra_constraints(
214             nt3, cnstr
215         );
216         fs_cell->tetra_list(2) = nt3;
217     } else {
218         Logger::error( Format::tr("Error in building mobile
219 mesh\n") );
220         return false;
221     }
222
223     Tetra* nt4 = tvol->create_simplex(
224         atom_list( u, v, w ),
225         atom_list( u+1, v+1, w ),
226         atom_list( u+1, v, w+1 ),
227         atom_list( u+1, v, w )
228     );
229
230     if( nt4 != nil ) {
231         tvol->add_simplex( nt4 );
232         TetraConstraints* cnstr = new TetraConstraints;
233         smod->set_tetra_constraints(
234             nt4, cnstr
235         );
236         fs_cell->tetra_list(3) = nt4;
237     } else {
238         Logger::error( Format::tr("Error in building mobile
239 mesh\n") );
240         return false;
241     }
242
243     Tetra* nt5 = tvol->create_simplex(
244         atom_list( u, v, w ),
245         atom_list( u+1, v, w+1 ),
246         atom_list( u, v+1, w+1 ),
247         atom_list( u, v, w+1 )
248     );
249
250     if( nt5 != nil ) {
251         tvol->add_simplex( nt5 );
252         TetraConstraints* cnstr = new TetraConstraints;
253         smod->set_tetra_constraints(
254             nt5, cnstr
255         );
256         fs_cell->tetra_list(4) = nt5;
257     } else {
258         Logger::error( Format::tr("Error in building mobile
259 mesh\n") );
260         return false;

```

```

260     }
261     } else {
262         Tetra* nt1 = tvol->create_simplex(
263             atom_list( u, v+1, w ),
264             atom_list( u+1, v+1, w+1 ),
265             atom_list( u, v, w+1 ),
266             atom_list( u+1, v, w )
267         );
268         if( nt1 != nil ) {
269             tvol->add_simplex( nt1 );
270             TetraConstraints* cnstr = new TetraConstraints;
271             smod->set_tetra_constraints(
272                 nt1, cnstr
273             );
274
275             fs_cell->tetra_list(0) = nt1;
276         } else {
277             Logger::error( Format::tr("Error in building mobile
mesh\n") );
278
279             return false;
280         }
281
282         Tetra* nt2 = tvol->create_simplex(
283             atom_list( u+1, v+1, w+1 ),
284             atom_list( u, v, w+1 ),
285             atom_list( u+1, v, w ),
286             atom_list( u+1, v, w+1 )
287         );
288         if( nt2 != nil ) {
289             tvol->add_simplex( nt2 );
290             TetraConstraints* cnstr = new TetraConstraints;
291             smod->set_tetra_constraints(
292                 nt2, cnstr
293             );
294
295             fs_cell->tetra_list(1) = nt2;
296         } else {
297             Logger::error( Format::tr("Error in building mobile
mesh\n") );
298
299             return false;
300         }
301
302         Tetra* nt3 = tvol->create_simplex(
303             atom_list( u, v+1, w ),
304             atom_list( u, v, w+1 ),
305             atom_list( u+1, v+1, w+1 ),
306             atom_list( u, v+1, w+1 )
307         );
308         if( nt3 != nil ) {
309             tvol->add_simplex( nt3 );
310             TetraConstraints* cnstr = new TetraConstraints;
311             smod->set_tetra_constraints(
312                 nt3, cnstr
313             );
314
315             fs_cell->tetra_list(2) = nt3;
316         } else {
317             Logger::error( Format::tr("Error in building mobile
mesh\n") );
318
319             return false;
320         }
321
322         Tetra* nt4 = tvol->create_simplex(
323             atom_list( u, v+1, w ),
324             atom_list( u+1, v+1, w+1 ),
325             atom_list( u+1, v, w ),
326             atom_list( u+1, v+1, w )
327         );

```

```

328         if( nt4 != nil ) {
329             tvol->add_simplex( nt4 );
330             TetraConstraints* cnstr = new TetraConstraints;
331             smod->set_tetra_constraints(
332                 nt4, cnstr
333             );
334
335             fs_cell->tetra_list(3) = nt4;
336         } else {
337             Logger::error( Format::tr("Error in building mobile
338 mesh\n") );
339             return false;
340         }
341
342         Tetra* nt5 = tvol->create_simplex(
343             atom_list( u, v+1, w ),
344             atom_list( u+1, v, w ),
345             atom_list( u, v, w+1 ),
346             atom_list( u, v, w )
347         );
348         if( nt5 != nil ) {
349             tvol->add_simplex( nt5 );
350             TetraConstraints* cnstr = new TetraConstraints;
351             smod->set_tetra_constraints(
352                 nt5, cnstr
353             );
354
355             fs_cell->tetra_list(4) = nt5;
356         } else {
357             Logger::error( Format::tr("Error in building mobile
358 mesh\n") );
359             return false;
360         }
361     }
362 }
363 }
364
365 smod->clean_region_model();
366
367 Cage3d cage = CS_voxet->cage();
368 float tolerance[3];
369 tolerance[0] = 0.05;
370 tolerance[1] = 0.05;
371 tolerance[2] = 0.05;
372
373
374 // It is mandatory to have our Voxets aligned with XYZ axis. Requested by BC
375 loading using
376 // RestorationLab. To be reported in user manual (if any).
377
378 // Build region_model with already assigned 2D regions (useful for the exper
379 iments)
380 SolidModelAPI::assign_2d(
381     smod_,
382     cage.u()/cage.u().length(),
383     tolerance,
384     CopyCString( solid_model()->name()->string() ) + "_face1"
385 );
386 SolidModelAPI::assign_2d(
387     smod_,
388     -cage.u()/cage.u().length(),
389     tolerance,
390     CopyCString( solid_model()->name()->string() ) + "_face2"
391 );
392 SolidModelAPI::assign_2d(
393     smod_,
394     cage.v()/cage.v().length(),
395     tolerance,
396     CopyCString( solid_model()->name()->string() ) + "_face3"

```

```

396     );
397     SolidModelAPI::assign_2d(
398         smod_,
399         - cage.v()/cage.v().length(),
400         tolerance,
401         CopyCString( solid_model()->name()->string() ) + "_face4"
402     );
403     SolidModelAPI::assign_2d(
404         smod_,
405         cage.w()/cage.w().length(),
406         tolerance,
407         CopyCString( solid_model()->name()->string() ) + "_face5"
408     );
409     SolidModelAPI::assign_2d(
410         smod_,
411         - cage.w()/cage.w().length(),
412         tolerance,
413         CopyCString( solid_model()->name()->string() ) + "_face6"
414     );
415
416     SolidModelAPI::assign_2d( smod_, "CS_voxet_SM_mobile_other" );
417     SolidModelAPI::assign( smod_, "CS_voxet_SM_mobile_inside" );
418
419     smod_->update_topology();
420
421     return true;
422 }

```

### 7.1.3.2 `const GridIndex GoUpScale::CSCell::cell_index () const` **[inline]**

Definition at line 85 of file `cs_cell.h`.

```

85                                     {
86         return uvw_;
87     };

```

### 7.1.3.3 `const PtrSequence<FSCell>& GoUpScale::CSCell::get_children () const` **[inline]**

Definition at line 93 of file `cs_cell.h`.

```

93                                     {
94         return fs_cells_;
95     };

```

### 7.1.3.4 `void GoUpScale::CSCell::set_cell_index (GridIndex uvw)` **[inline]**

Definition at line 89 of file `cs_cell.h`.

```

89                                     {
90         uvw_ = uvw;
91     };

```

### 7.1.3.5 `bool GoUpScale::CSCell::set_materials (GridIndex index_transl = 0)`

Creates a material for each `fs_cell` and assign it to corresponding tetras.

**Author:**

Marc-Olivier Titeux

**Version:**

1.0

**Date:**

27-02-2008

Definition at line 425 of file cs\_cell.cpp.

References [GoUpScale::FSCell::build\\_material\\_from\\_voxet\\_properties\(\)](#), [get\\_children\(\)](#), [GoUpScale::FSCell::material\(\)](#), [solid\\_model\(\)](#), and [GoUpScale::FSCell::tetra\\_list\(\)](#).

```

425                                     {
426
427     SolidModelAPI::set_material_everywhere( solid_model(), "default" );
428
429     for( PtrSequence<FSCell>::iterator itr( get_children() ); itr.more(); itr.ne
xt() ) {
430         FSCell* fs_cur = itr.cur();
431         if( !( fs_cur->build_material_from_voxet_properties( prop_grids_, index
_transl ) ) ) {
432             return false;
433         }
434
435         for( int it = 0; it < 5; ++it ) {
436             solid_model()->set_material( fs_cur->tetra_list( it ), fs_cur->mat
erial() );
437         }
438     }
439     return true;
440 }
441 }
```

**7.1.3.6 SolidModel\* GoUpScale::CSCell::solid\_model () [inline]**

Definition at line 118 of file cs\_cell.h.

```

118                                     {
119     return smod_;
120 }
```

**7.1.3.7 const GObjLink\* GoUpScale::CSCell::voxets\_link () const [inline]**

Definition at line 81 of file cs\_cell.h.

```

81                                     {
82     return voxets_link_;
83 }
```

The documentation for this class was generated from the following files:

- [include/GoUpScale/gridding/cs\\_cell.h](#)
- [src/lib/gridding/cs\\_cell.cpp](#)

## 7.2 Experiment Class Reference

```
#include <GoUpScale/mechanics/experiments.h>
```

### 7.2.1 Detailed Description

Generic [Experiment](#)

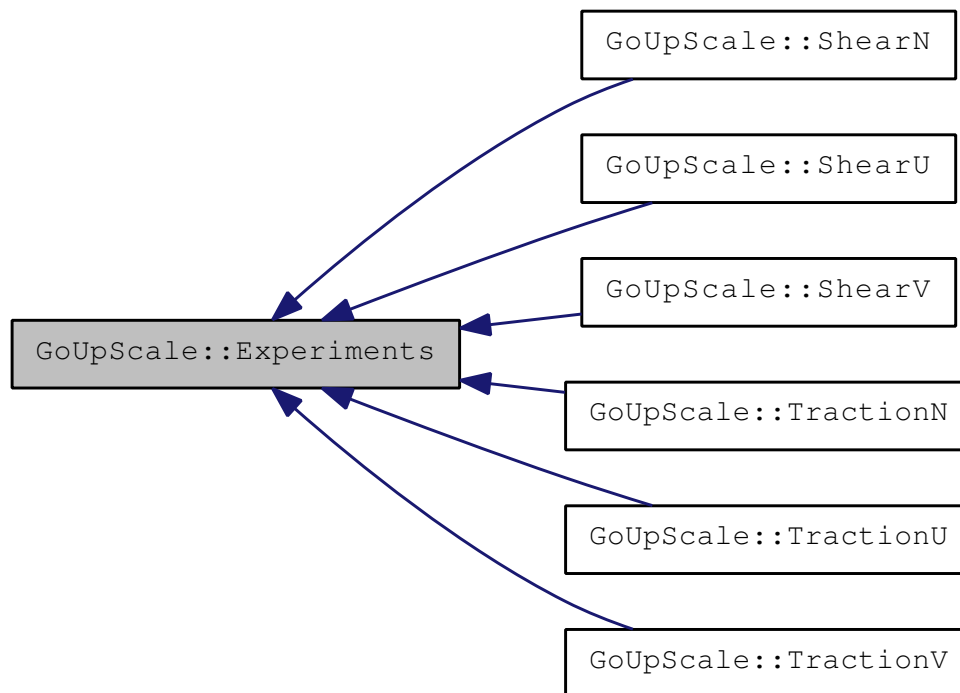
The documentation for this class was generated from the following file:

- [include/GoUpScale/mechanics/experiments.h](#)

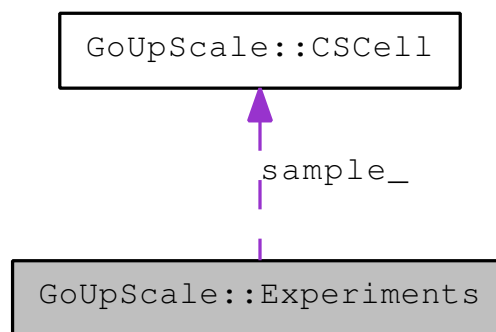


### 7.3 GoUpScale::Experiments Class Reference

#include <experiments.h>Inheritance diagram for GoUpScale::Experiments:



Collaboration diagram for GoUpScale::Experiments:



#### Public Member Functions

- virtual `~Experiments ()`  
*Destructor of the class.*
- virtual void `boundary_conditions ()=0`  
*Sets the boundary conditions for the experiment.*
- virtual const CopyCString `type ()=0`
- virtual const CopyCString `direction ()=0`
- void `init ()`

*Initialize experiment (name, problem).*

- void `flush ()`  
*Removing properties at the end of each run on the SoliModel.*
- void `run ()`  
*Run the FE problem.*
- void `compute_means ()`  
*Computes means of strain and stress for one experiment.*
- `CSCell * sample ()`
- const double `displacement ()` const
- void `set_prob_name (CString name)`
- const CopyCString & `prob_name ()` const
- const bool `computed ()` const
- void `comp_state (bool state)`
- const DenseVector< double > & `get_strain_means ()` const
- const DenseVector< double > & `get_stress_means ()` const

## Data Fields

- [Experiments](#)

## Protected Member Functions

- [Experiments \(CSCell \\*sample, const double displacement, const bool large\\_deformation\)](#)  
*Constructor of the class.*

### 7.3.1 Detailed Description

Definition at line 38 of file experiments.h.

### 7.3.2 Constructor & Destructor Documentation

#### 7.3.2.1 GoUpScale::Experiments::~~Experiments () [virtual]

Destructor of the class.

Definition at line 56 of file experiments.cpp.

```
56         {
57 //         flush();
58 }
```

### 7.3.2.2 `GoUpScale::Experiments::Experiments (CSCell * sample, const double displacement, const bool large_deformation)` [`protected`]

Constructor of the class.

#### Parameters:

*sample* `CSCell` on which the experiments are currently performed

*displacement* value of the displacement to apply during the experiments (the displacement is the same for all the experiments)

*large\_deformation* sets whether the experiments must be ran in large deformation or not

## 7.3.3 Member Function Documentation

### 7.3.3.1 `virtual void GoUpScale::Experiments::boundary_conditions ()` [`pure virtual`]

Sets the boundary conditions for the experiment.

Implemented in `GoUpScale::TractionU`, `GoUpScale::TractionV`, `GoUpScale::TractionN`, `GoUpScale::ShearU`, `GoUpScale::ShearV`, and `GoUpScale::ShearN`.

### 7.3.3.2 `void GoUpScale::Experiments::comp_state (bool state)` [`inline`]

Definition at line 99 of file `experiments.h`.

```
99                                     {
100         computed_ = state;
101     };
```

### 7.3.3.3 `void GoUpScale::Experiments::compute_means ()`

Computes means of strain and stress for one experiment.

Definition at line 134 of file `experiments.cpp`.

References `computed()`, `GoUpScale::FSCell::get_stiffness()`, `goupscale_assert`, `prob_name()`, `sample()`, `GoUpScale::CSCell::solid_model()`, and `GoUpScale::FSCell::tetra_list()`.

```
134                                     {
135     if( !computed() ) {
136         Logger::error( Format::tr( "FE problems must have been ran before compu
137         ting mean values\n" ) );
138         return;
139     }
140     // MeanStressComponents and MeanStrainComponents set to 0. at each cell
141     MeanStressComponents_.set( 0. );
142     MeanStrainComponents_.set( 0. );
143
144     // Matrices used for strain and stress averaging
145     RestorationLab::DenseVector<double> E_tet(6);
146     RestorationLab::DenseVector<double> S_tet(6);
147     RestorationLab::Matrix3x3 E_tmp;
148
149     int prop_index = sample()->solid_model()->datapack().find_property_index(
150     prob_name() );
151     for( PtrSequence<FSCell>::iterator fs_itr( sample()->get_children() ); fs_it
152     r.more(); fs_itr.next() ) {
```

```

152     FSCell* cur_fscell = fs_itr.cur();
153
154     goupscale_assert( cur_fscell != nil );
155
156     for( int j = 0; j < 5; ++j ) {
157         E_tmp.set( 0. );
158         E_tet.set( 0. );
159         S_tet.set( 0. );
160
161         Tetra* cur_tetra = cur_fscell->tetra_list( j );
162
163         // 4 tetras on 5 have the same volume. There may be an alternate
164         // way to do this (storing the volumes somewhere for all the tetra
165         s)
166         double vol = cur_tetra->tetra3d().volume();
167
168         //- Computation of strain tensor
169         strain_tensor( cur_tetra, prop_index, E_tmp );
170
171         E_tet[0] = E_tmp.ref( 0, 0 );
172         E_tet[1] = E_tmp.ref( 1, 1 );
173         E_tet[2] = E_tmp.ref( 2, 2 );
174         E_tet[3] = E_tmp.ref( 1, 2 );
175         E_tet[4] = E_tmp.ref( 0, 2 );
176         E_tet[5] = E_tmp.ref( 0, 1 );
177
178         MeanStrainComponents_[0] += E_tet[0]*vol;
179         MeanStrainComponents_[1] += E_tet[1]*vol;
180         MeanStrainComponents_[2] += E_tet[2]*vol;
181         MeanStrainComponents_[3] += E_tet[3]*vol;
182         MeanStrainComponents_[4] += E_tet[4]*vol;
183         MeanStrainComponents_[5] += E_tet[5]*vol;
184
185         //- Computation of stress tensor
186         S_tet = vol*( cur_fscell->get_stiffness()*E_tet );
187
188         MeanStressComponents_[0] += S_tet[0];
189         MeanStressComponents_[1] += S_tet[1];
190         MeanStressComponents_[2] += S_tet[2];
191         MeanStressComponents_[3] += S_tet[3];
192         MeanStressComponents_[4] += S_tet[4];
193         MeanStressComponents_[5] += S_tet[5];
194     }
195
196
197 //DEBUG
198 //     cout << endl;
199 //     cout << "MeanStrainComponents[3] = " << MeanStrainComponents_[3] << e
200 //         endl;
201 //     cout << "MeanStrainComponents[4] = " << MeanStrainComponents_[4] << e
202 //         endl;
203 //     cout << "MeanStressComponents[3] = " << MeanStressComponents_[3] << e
204 //         endl;
205 //     cout << "MeanStressComponents[4] = " << MeanStressComponents_[4] << e
206 //         endl;
207 //     cout << endl;
208
209     return;
210 }

```

### 7.3.3.4 const bool GoUpScale::Experiments::computed () const [inline]

Definition at line 95 of file experiments.h.

```

96         return computed_;
97     };

```

### 7.3.3.5 virtual const CopyCString GoUpScale::Experiments::direction () [pure virtual]

#### Returns:

the direction of the experiment

Implemented in [GoUpScale::TractionU](#), [GoUpScale::TractionV](#), [GoUpScale::TractionN](#), [GoUpScale::ShearU](#), [GoUpScale::ShearV](#), and [GoUpScale::ShearN](#).

### 7.3.3.6 const double GoUpScale::Experiments::displacement () const [inline]

Definition at line 83 of file experiments.h.

```

83         {
84             return displacement_;
85     };

```

### 7.3.3.7 void GoUpScale::Experiments::flush ()

Removing properties at the end of each run on the SoliModel.

Definition at line 62 of file experiments.cpp.

References [prob\\_name\(\)](#), and [sample\(\)](#).

```

62     {
63         SolidRestorationAPI::destroy( sample()->solid_model(), prob_name() );
64         // It may be useful to move this to SolidRestorationAPI::destroy()
65         // Why was this useful? (M-O T)
66         PropertyAPI::remove_property( sample()->solid_model(), prob_name() );
67     }

```

### 7.3.3.8 const DenseVector<double>& GoUpScale::Experiments::get\_strain\_means () const [inline]

Definition at line 103 of file experiments.h.

```

103         {
104             return MeanStrainComponents_;
105     };

```

### 7.3.3.9 const DenseVector<double>& GoUpScale::Experiments::get\_stress\_means () const [inline]

Definition at line 107 of file experiments.h.

```

107         {
108             return MeanStressComponents_;
109     };

```

### 7.3.3.10 void GoUpScale::Experiments::init ()

Initialize experiment (name, problem).

Definition at line 104 of file experiments.cpp.

References comp\_state(), direction(), prob\_name(), sample(), set\_prob\_name(), and type().

```
104         {
105     Logger::info( Format::tr( "building and running deformation problem for %1 a
long %2\n") % type().string() % direction().string() );
106
107
108 //DEBUG
109 //     cout << "building and running deformation problem for "
110 //         << type().string()
111 //         << " along "
112 //         << direction().string()
113 //         << endl;
114
115
116     // New FE Problem
117     set_prob_name(
118         "prob_CS_voxet_"
119         + CopyCString::number( sample()->cell_index() )
120         + "_"
121         + type().string()
122         + direction().string()
123         );
124
125     SolidRestorationAPI::create( sample()->solid_model(), "Static", prob_name()
);
126
127     comp_state( false );
128
129     return;
130 }
```

### 7.3.3.11 const CopyCString& GoUpScale::Experiments::prob\_name () const [inline]

Definition at line 91 of file experiments.h.

```
91         {
92     return prob_name_;
93     };
```

### 7.3.3.12 void GoUpScale::Experiments::run ()

Run the FE problem.

Definition at line 70 of file experiments.cpp.

References boundary\_conditions(), comp\_state(), compute\_means(), flush(), goup\_scale\_assert, init(), prob\_name(), and sample().

```
70         {
71     goup_scale_assert( sample()->solid_model() != nil );
72
73     // Creating the FE problem
74     init();
75
76     // Set if hookean or neo hookean (small or large deformations)
77     // By default, hookean deformations are assumed
```

```

78     if( large_deformation_ ) {
79         SolidRestorationAPI::set_constitutive_model( sample()->solid_model(),
prob_name(), "NeoHookean" );
80     }
81
82     // Set the BC
83     boundary_conditions();
84
85     // Needed options for numeric solver (please refer to FemPack API and docs f
or changing this)
86     const char* method("TaucsCG_ICC");
87     SolidRestorationAPI::set_task_solver( sample()->solid_model(), prob_name(),
"Linear", method );
88     SolidRestorationAPI::set_task_solver( sample()->solid_model(), prob_name(),
"NonLinear", method );
89
90     SolidRestorationAPI::run( sample()->solid_model(), prob_name() );
91
92     comp_state( true );
93
94     // Compute the means of strain and stress tensors on one CS cell
95     compute_means();
96
97     // Removing the current FE problem
98     flush();
99
100    return;
101 }

```

### 7.3.3.13 CCell\* GoUpScale::Experiments::sample () [inline]

Definition at line 79 of file experiments.h.

```

79         {
80             return sample_;
81     };

```

### 7.3.3.14 void GoUpScale::Experiments::set\_prob\_name (CString name) [inline]

Definition at line 87 of file experiments.h.

```

87         {
88             prob_name_ = name;
89     };

```

### 7.3.3.15 virtual const CopyCString GoUpScale::Experiments::type () [pure virtual]

#### Returns:

what kind of experiment this (traction, shear,...)

Implemented in [GoUpScale::TractionU](#), [GoUpScale::TractionV](#), [GoUpScale::TractionN](#), [GoUpScale::ShearU](#), [GoUpScale::ShearV](#), and [GoUpScale::ShearN](#).

## 7.3.4 Field Documentation

### 7.3.4.1 GoUpScale::Experiments::Experiments

Definition at line 42 of file experiments.h.

The documentation for this class was generated from the following files:

- [include/GoUpScale/mechanics/experiments.h](#)
- [src/lib/mechanics/experiments.cpp](#)



## 7.4 GoUpScale::FSCell Class Reference

Abstract class for cellules mapping within voxets.

```
#include <fs_cell.h>
```

### Public Member Functions

- [FSCell](#) (GObjLink \*voxets\_link, GridIndex uvw)  
*Constructor of the class.*
- virtual [~FSCell](#) ()  
*Destructor.*
- bool [build\\_material\\_from\\_voxet\\_properties](#) (GridPropData \*\*, GridIndex)  
*Create Material corresponding to inputed mechanical properties onto the FS\_voxet .*
- void [build\\_stiffness\\_tensor](#) ()  
*Build an ISOTROPIC compliance tensor for this FS cell (Stress = C \* Strain).*
- DenseMatrix< double > [get\\_stiffness](#) () const
- Tetra \*& [tetra\\_list](#) (int index)
- const Tetra \* [tetra\\_list](#) (int index) const
- const GObjLink \* [voxets\\_link](#) () const
- const RestorationLab::Material \* [material](#) () const
- void [set\\_material](#) (const RestorationLab::Material \*cmat)
- const GridIndex [cell\\_index](#) () const
- void [set\\_cell\\_index](#) (GridIndex uvw)

### 7.4.1 Detailed Description

Abstract class for cellules mapping within voxets.

**Author:**

Marc-Olivier Titeux

**Version:**

3.0

**Date:**

26-10-2007

This class describes a fine-scale cell

Definition at line 59 of file fs\_cell.h.

### 7.4.2 Constructor & Destructor Documentation

#### 7.4.2.1 GoUpScale::FSCell::FSCell (GObjLink \* voxets\_link, GridIndex uvw)

Constructor of the class.

## Parameters:

*voxets\_link* *GObjLink* between the two *Voxets*, from different scales  
*uvw* position of the *FSCell* in *voxet\_father*

### 7.4.2.2 GoUpScale::FSCell::~~FSCell () [virtual]

Destructor.

Definition at line 51 of file *fs\_cell.cpp*.

```
51         {
52     delete[] tetra_list_;
53     if( cmat_ != nil ) {
54         if( cmat_>name() != "default" ) {
55             RestorationLab::MaterialLib::instance()->remove( cmat_>name() );
56         }
57     }
58 }
```

## 7.4.3 Member Function Documentation

### 7.4.3.1 bool GoUpScale::FSCell::build\_material\_from\_voxet\_properties (GridPropData \*\*prop\_list, GridIndex index\_transl)

Create Material corresponding to inputed mechanical properties onto the *FS\_voxet* . The method is for isotropic materials, the created materials are scalar

## Parameters:

*prop\_list* list of properties used to build matrix

*index\_transl,:* used to translate the numbering of *FSCell* from the index of *CSCell* such as:

```
cs_ug.u()
+ cs_ug.v()*FS_voxet->cell_dimensions().size_u()
+ cs_ug.w()*FS_voxet->cell_dimensions().size_uv()*size;
```

Definition at line 61 of file *fs\_cell.cpp*.

References *build\_stiffness\_tensor()*, *cell\_index()*, *goupscale\_assert*, *set\_material()*, and *voxets\_link()*.

```
61         {
62     goupscale_assert( prop_list != nil );
63
64     GridIndex index = this->cell_index() + index_transl;
65
66     double E = 1.e9*prop_list[0]->get_data( index, 0 );
67     double nu = prop_list[1]->get_data( index, 0 );
68     double rho = prop_list[2]->get_data( index, 0 );
69
70     double lambda = 0.;
71     double mu = 0.;
72
73     // \a FemPack::Material are built from Lamé coefficients
74     if( nu != 0.5 ) {
75         lambda = ( E*nu )/( (1 + nu)*( 1 - 2*nu ) );
76         mu = E/( 2 + 2*nu );
77     } else {
78         cout << "wrong nu in fine cell (no 0.5 for Poisson's ratio) : " << inde
x << endl;
79         return false;
80     }
```

```

81
82 CopyCString matname = CopyCString( "mat_" +
83     *(dynamic_cast<Voxel*>( this->voxets_link()->matel() )->name() ) +
84     "_" +
85     CopyCString::number( index )
86     );
87
88 RestorationLab::Material* m = new RestorationLab::Material( matname );
89 Resource::ref(m);
90
91 m->set_lambda(
92     lambda
93 );
94
95 m->set_mu(
96     mu
97 );
98
99 m->set_rho(
100     rho
101 );
102
103 set_material( m );
104
105 Resource::unref( m );
106
107 build_stiffness_tensor();
108
109 return true;
110 }

```

#### 7.4.3.2 void GoUpScale::FSCell::build\_stiffness\_tensor ()

Build an ISOTROPIC compliance tensor for this FS cell (Stress = C \* Strain).

Definition at line 113 of file fs\_cell.cpp.

References material().

```

113                                     {
114     double mu = material()->mu();
115     double lambda = material()->lambda();
116
117     C_.set( 0. );
118
119     for( int i = 0; i < 3; ++i ) {
120         C_(i,i) = 2*mu;
121         for( int j = 0; j < 3; ++j ) {
122             C_(i,j) += lambda;
123         }
124     }
125
126     for( int k = 3; k < 6; ++k ) {
127         C_(k,k) = mu;
128     }
129
130 //         cout << endl;
131
132 //         for( int k = 0; k<6; ++k ) {
133 //             for( int p = 0; p<6; ++p ) {
134 //                 cout << C_(p,k) << " ";
135 //             }
136 //             cout << endl;
137 //         }
138 //
139 //         cout << endl;
140
141     return;

```

```
142 }
```

#### 7.4.3.3 `const GridIndex GoUpScale::FSCell::cell_index () const [inline]`

Definition at line 127 of file `fs_cell.h`.

```
127                                     {
128         return uvw_;
129     };
```

#### 7.4.3.4 `DenseMatrix<double> GoUpScale::FSCell::get_stiffness () const [inline]`

Definition at line 92 of file `fs_cell.h`.

```
92                                     {
93         return C_;
94     };
```

#### 7.4.3.5 `const RestorationLab::Material* GoUpScale::FSCell::material () const [inline]`

Definition at line 110 of file `fs_cell.h`.

```
110                                     {
111         return cmat_;
112     };
```

#### 7.4.3.6 `void GoUpScale::FSCell::set_cell_index (GridIndex uvw) [inline]`

Definition at line 131 of file `fs_cell.h`.

```
131                                     {
132         uvw_ = uvw;
133     };
```

#### 7.4.3.7 `void GoUpScale::FSCell::set_material (const RestorationLab::Material * cmat) [inline]`

Definition at line 114 of file `fs_cell.h`.

```
114                                     {
115         if( cmat_ == cmat ) {
116             return;
117         }
118         if( cmat_ != nil ) {
119             Resource::unref( cmat_ );
120         }
121         cmat_ = const_cast<RestorationLab::Material*>( cmat );
122         Resource::ref( cmat_ );
123     };
```

#### 7.4.3.8 `const Tetra* GoUpScale::FSCell::tetra_list (int index) const` `[inline]`

Definition at line 101 of file `fs_cell.h`.

References `goupscale_assert`.

```
101                                     {
102                                     goupscale_assert( index < 5 );
103                                     return tetra_list_[index];
104                                     };
```

#### 7.4.3.9 `Tetra* & GoUpScale::FSCell::tetra_list (int index)` `[inline]`

Definition at line 96 of file `fs_cell.h`.

References `goupscale_assert`.

```
96                                     {
97                                     goupscale_assert( index < 5 );
98                                     return tetra_list_[index];
99                                     };
```

#### 7.4.3.10 `const GObjLink* GoUpScale::FSCell::voxets_link () const` `[inline]`

Definition at line 106 of file `fs_cell.h`.

```
106                                     {
107                                     return voxets_link_;
108                                     };
```

The documentation for this class was generated from the following files:

- [include/GoUpScale/gridding/fs\\_cell.h](#)
- [src/lib/gridding/fs\\_cell.cpp](#)

## 7.5 GoUpScale::MechUpscaleAPI Class Reference

```
#include <mech_upscale_api.h>
```

### Static Public Member Functions

- static bool [upscale](#) (const Voxet &FS\_voxet, Voxet \*CS\_voxet, const CString &YoungMod, const CString &PoissonRat, const CString &Rho, const double displacement=0.5, const bool large\_deformation=false, const bool traction\_u=false, const bool traction\_v=false, const bool traction\_n=false, const bool shear\_u=false, const bool shear\_v=false, const bool shear\_n=false)

*API for upscaling a fine voxet to a coarse one.*

- static void [mean\\_prop\\_on\\_voxet](#) (Voxet \*, const CString &, const CString &, int=1)

*API for averaging a property by plane or set of planes.*

### 7.5.1 Detailed Description

Definition at line 38 of file mech\_upscale\_api.h.

### 7.5.2 Member Function Documentation

#### 7.5.2.1 void GoUpScale::MechUpscaleAPI::mean\_prop\_on\_voxet (Voxet \* voxet, const CString & prop\_name, const CString & new\_prop\_name, int num\_planes = 1) [static]

API for averaging a property by plane or set of planes.

#### Author:

Marc-Olivier Titeux

#### Version:

4.0

#### Date:

05-06-2008

This API average a property on the cells that have the same W

#### Parameters:

*voxet* onto average

*prop\_name* is the name of the scalar property to average

*new\_prop\_name* is the new property into which the arithmetic average will be store

*num\_planes* is the number of horizontal planes onto average

Definition at line 378 of file mech\_upscale\_api.cpp.

References [goupscale\\_assert](#).

```
384 {  
385     goupscale_assert( voxet != nil );  
386     goupscale_assert( voxet->find_grid( prop_name ) != nil );
```

```

387
388     Property* prop = voxet->find_grid( prop_name )->property();
389
390     if( prop->number_of_elements() != 1 ) {
391         Logger::info( Format::tr( "This command does not deal with non-scalar p
roperties\n" ) );
392         return;
393     }
394
395     Property* prop_new = MechUpscaleAPI::find_or_create_property( voxet, new_pro
p_name );
396
397     int nu = voxet->nu();
398     int nv = voxet->nv();
399     int nw = voxet->nw();
400
401     double m_prop_val = 0.;
402     double tmp = 0.;
403
404     int cellindex = 0;
405
406     for( int w = 0; w <nw; ++w ) {
407         m_prop_val = 0.;
408         tmp = 0.;
409         for( int v = 0; v <nv; ++v ) {
410             for( int u = 0; u <nu; ++u ) {
411                 cellindex = u+v*nu+w*nu*nv;
412                 prop->property_accessor()->get_real_value( GridObjT
opoHandle( cellindex,
413                                                     voxet->cell_dimensions() ),
414                                                     &tmp
415                                                     );
416                 m_prop_val += tmp;
417             }
418         }
419
420         m_prop_val /= nu*nv;
421 //DEBUG
422 //         cout << "m_prop_val = " << m_prop_val << " in plane " << w << en
dl;
423
424         for( int v = 0; v <nv; ++v ) {
425             for( int u = 0; u <nu; ++u ) {
426                 cellindex = u+v*nu+w*nu*nv;
427                 prop_new->property_accessor()->set_real_value( Grid
ObjTopoHandle( cellindex,
428                                                         voxet->cell_dimensions() ),
429                                                         &m_prop_val
430                                                         );
431             }
432         }
433     }
434
435     return;
436
437 }

```

**7.5.2.2** `bool GoUpScale::MechUpscaleAPI::upscale (const Voxet & FS_voxet, Voxet * CS_voxet, const CString & YoungMod, const CString & PoissonRat, const CString & Rho, const double displacement = 0.5, const bool large_deformation = false, const bool traction_u = false, const bool traction_v = false, const bool traction_n = false, const bool shear_u = false, const bool shear_v = false, const bool shear_n = false) [static]`

API for upscaling a fine voxet to a coarse one.

**Author:**

Marc-Olivier Titeux

**Version:**

4.0

**Date:**

05-11-2007

This API upscales 3 mechanical properties of a fine voxel (describing isotropic materials) to 5 mechanical properties (describing transversely isotropic materials). The *Voxel*

**Parameters:**

*FS\_voxel* is a fine scale description of the model

*CS\_voxel* is coarse scale description of the model The size of *CS\_voxel* must be chosen modulo the size of *FS\_voxel*

*YoungMod* name of the property of Young's Modulus on *FS\_voxel*

*PoissonRat* name of the property of Poisson ratio on *FS\_voxel*

*Rho* name of the property of density on *FS\_voxel*

*displacement* value of the displacement to apply on the samples during the experiment

*large\_deformation* sets whether the experiments are to be ran in large deformations

*traction\_u* sets if the traction in U direction must be taken into account

*traction\_v* sets if the traction in V direction must be taken into account

*traction\_n* sets if the traction in N direction must be taken into account

*shear\_u* sets if the shearing in U direction must be taken into account

*shear\_v* sets if the shearing in V direction must be taken into account

*shear\_n* sets if the shearing in N direction must be taken into account

Definition at line 67 of file mech\_upscale\_api.cpp.

References `GoUpScale::CSCell::build_fine_description()`, `GoUpScale::CSCell::cell_index()`, `goupscale_assert`, `GoUpScale::Experiments::run()`, `GoUpScale::CSCell::set_cell_index()`, `GoUpScale::CSCell::set_materials()`, `GoUpScale::ShearN`, `GoUpScale::ShearU`, `GoUpScale::ShearV`, `GoUpScale::StressMeanMatrix< T >::start_aggregate()`, `GoUpScale::StrainMeanVector< T >::start_aggregate()`, `GoUpScale::TractionN`, `GoUpScale::TractionU`, and `GoUpScale::TractionV`.

```

82 {
83     // benchmarking
84     Clock* time = new Clock;
85
86     //- Need to create a GobjLink to link Fine-scale and Coarse-scale cells
87     GObjLink* voxets_link = new GObjLink( const_cast<Voxel*>( &FS_voxel), CS_vox
    et );
88
89     goupscale_assert( voxets_link->matel() != nil );
90     goupscale_assert( voxets_link->mate2() != nil );
91     goupscale_assert( FS_voxel.property_db()->find_property( YoungMod ) != nil )
    ;
92     goupscale_assert( FS_voxel.property_db()->find_property( PoissonRat ) != nil
    );
93     goupscale_assert( FS_voxel.property_db()->find_property( Rho ) != nil );
94
95
96     // FS and CS voxel origin must be the same and must have the same dimensions
    (in

```



```

97 // terms of
98 if( FS_voxet.cage().o() != CS_voxet->cage().o()  &&
99     FS_voxet.cage().min() != CS_voxet->cage().min() &&
100     FS_voxet.cage().max() != CS_voxet->cage().max() ) {
101     Logger::fatal( Format::tr( "The grids do not completely overlap\n" ) );
102     return false;
103 }
104
105
106 // - Verifying and initialize input and output properties
107 GridPropData* prop_grids[3];
108 prop_grids[0] = new GridPropData(
109     FS_voxet.property_db()->find_property( YoungMod )
110 );
111 prop_grids[1] = new GridPropData(
112     FS_voxet.property_db()->find_property( PoissonRat )
113 );
114 prop_grids[2] = new GridPropData(
115     FS_voxet.property_db()->find_property( Rho )
116 );
117
118 // CHECK VALIDITY OF PROPERTIES ON FS_VOXET
119
120 // Preparing output properties
121 // Young's Modulus (plane)
122 Property* mod_young_1 = MechUpscaleAPI::find_or_create_property( CS_voxet, "
mod_young_1" );
123
124 // Young's Modulus (axis)
125 Property* mod_young_3 = MechUpscaleAPI::find_or_create_property( CS_voxet, "
mod_young_3" );
126
127 // Poisson ratio (plane)24
128 Property* pois_rat_1 = MechUpscaleAPI::find_or_create_property( CS_voxet, "p
ois_rat_1" );
129
130 // Poisson ratio (axis)
131 Property* pois_rat_3 = MechUpscaleAPI::find_or_create_property( CS_voxet, "p
ois_rat_3" );
132
133 // Shear modulus (axis)
134 Property* mod_shear_3 = MechUpscaleAPI::find_or_create_property( CS_voxet, "
mod_shear_3" );
135
136
137 // -----
138 // Building wrappers
139 // -----
140
141 // Mobile CS cell and its mesh made with tetras
142 CSCell cs_cell( voxets_link, 0, prop_grids );
143
144 GOCAD_ASSERT(
145     cs_cell.build_fine_description(),
146     {
147         Logger::fatal( Format::tr( "Error in building mobile mesh\n" )
148 );
149         return false; }
150 );
151
152 // volume of a CS cell
153 GridSize3d propsize = CS_voxet->property_dimensions(CS_voxet->property_align
ment());
154 double cs_volume = CS_voxet->cell13d( UGridIndex3d(cs_cell.cell_index(), prop
size ) ).volume();
155
156 GOCAD_ASSERT(
157     cs_volume != 0,
158     {
159         Logger::fatal( Format::tr( "Error in Defining Coarse-Scale ce

```

```

159         ll %1. Volume of cell is equal to zero.\n") % cs_cell.cell_index() );
160         return false; }
161     );
162
163     // A possible improvement of the code (especially in the gapi) would be to e
ncapsulate the algo
164     // in a functor
165
166
167 //GATHERING
168
169     // We select the different experiments to run and put them in a PtrList
170     PtrList<Experiments> exp_list;
171     int num_exp = 0;
172
173     if( traction_u ) {
174         ++num_exp;
175         TractionU* tract_u = new TractionU( &cs_cell, displacement, large_
deformation );
176         exp_list.append( tract_u );
177     }
178     if( traction_v ) {
179         ++num_exp;
180         TractionV* tract_v = new TractionV( &cs_cell, displacement, large_
deformation );
181         exp_list.append( tract_v );
182     }
183     if( traction_n ) {
184         ++num_exp;
185         TractionN* tract_n= new TractionN( &cs_cell, displacement, large_d
eformation );
186         exp_list.append( tract_n );
187     }
188     if( shear_u ) {
189         ++num_exp;
190         ShearU* shear_u = new ShearU( &cs_cell, displacement, large_deform
ation );
191         exp_list.append( shear_u );
192 }
193     if( shear_v ) {
194         ++num_exp;
195         ShearV* shear_v = new ShearV( &cs_cell, displacement, large_deform
ation );
196         exp_list.append( shear_v );
197     }
198     if( shear_n ) {
199         ++num_exp;
200         ShearN* shear_n = new ShearN( &cs_cell, displacement, large_deform
ation );
201         exp_list.append( shear_n );
202     }
203
204
205 //END GATHERING
206
207 //         num_exp = upscalor.build_experiment_set( ... )
208
209     GOCAD_ASSERT(
210         num_exp != 0,
211         {
212             Logger::fatal( Format::tr( "Need to select at least one exper
iment (3 is preferable)\n" ) );
213             return false; }
214     );
215
216
217 //EXECUTOR
218
219     // Matrices and vectors used for inversion
220     RestorationLab::Assembler::Vector b_vect( 5 );

```

```

221 RestorationLab::Assembler::Matrix sq_syst( 5 );
222 StrainMeanVector<double> s_vect( 4*num_exp );
223 StressMeanMatrix<double> rc_syst( 4*num_exp );
224 RestorationLab::DenseMatrix<double> rl_syst( 4*num_exp, 5 );
225 RestorationLab::Assembler::Vector X0( 5 );
226
227 // Size of CS voxel
228 GridIndex cs_nu = CS_voxel->cell_dimensions().size_u();
229 GridIndex cs_nv = CS_voxel->cell_dimensions().size_v();
230 GridIndex cs_nw = CS_voxel->cell_dimensions().size_w();
231
232 // Size of FS voxel
233 GridIndex fs_nu = FS_voxel.cell_dimensions().size_u();
234 GridIndex fs_nuv = FS_voxel.cell_dimensions().size_uv();
235
236
237 // Index of current cs cell to upscale. Nothing is moved only true index are
238
239 // computed to capture correct property values
240 GridIndex cscellindex = 0;
241
242 for( GridIndex w = 0; w < cs_nw; ++w ) {
243     for( GridIndex v = 0; v < cs_nv; ++v ) {
244         for( GridIndex u = 0; u < cs_nu; ++u ) {
245
246             cscellindex = u + v*cs_nu + w*cs_nu*cs_nv;
247             cs_cell.set_cell_index( cscellindex );
248
249             GridIndex index_transl = ( u + fs_nu*v + fs_nuv*w )
250 *5;
251
252             // Correspondance between index of current CSCell a
253 nd first index of FSCell within
254             // the current CSCell
255             GOCAD_ASSERT( cs_cell.set_materials( index_transl )
256 , return false; );
257
258             for( PtrList<Experiments>::iterator exp_itr( exp_list
259 ); exp_itr.more(); exp_itr.next() ) {
260                 Experiments* expcur = exp_itr.cur();
261                 expcur->run();
262             }
263
264             // Agregation of means of strain tensor in a second
265 member vector
266             s_vect.start_aggregate( exp_list );
267
268             // Agregation of means of stress tensor in a rectan
269 gular (column) system
270             rc_syst.start_aggregate( exp_list );
271
272             // Computation of the transpose
273             rl_syst.set( 0. );
274             rl_syst = rc_syst;
275
276             rl_syst.transpose();
277
278             // Computation of the square matrix system: tA*A*X
279 =tA*B
280             for( int i = 0; i<5; ++i ) {
281                 for( int j = 0; j<5; ++j ) {
282                     sq_syst(i,j) = 0.;
283                 }
284             }
285
286             for( int i = 0; i<5; ++i ) {
287                 for( int j = 0; j<5; ++j ) {
288                     for( int k = 0; k < rl_syst.nb_colum
289 ns(); ++k ) {
290                         sq_syst(i,j) += rl_syst(i,
291 k)*rc_syst(k,j);

```

```

282         }
283     }
284 }
285
286 b_vect.set( 0. );
287
288 for( int i = 0; i<5; ++i) {
289     for( int k = 0; k < rl_syst.nb_columns();
290         ++k ) {
291         b_vect[i] += rl_syst(i,k)*s_vect[k];
292     }
293 }
294 // Solution vector
295 X0.set( 0. );
296
297 Logger::info( Format::tr( "solving with superlu mat
hematical library\n" ) );
298
299 RestorationLab::solve_with_superlu(
300     sq_syst,
301     b_vect,
302     X0
303 );
304
305 if( X0 != RestorationLab::Assembler::Vector::zero(5
) &&
306     X0[0] != 0. &&
307     X0[1] != 0. &&
308     X0[4] != 0. ) {
309     X0[0] = 1/X0[0];
310     X0[1] = 1/X0[1];
311     X0[2] *= X0[0];
312     X0[3] *= X0[1];
313     X0[4] = 1./( X0[4] );
314 } else {
315     Logger::info( Format::tr( "Nil solution\n" ) );
316 }
317
318 //DEBUG
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326
327 mod_young_1->property_accessor()->set_real_value( G
ridObjTopoHandle( cscellindex,
328     CS_voxet->cell_dimensions() ),
329     &X0[0]
330 );
331 mod_young_3->property_accessor()->set_real_value( G
ridObjTopoHandle( cscellindex,
332     CS_voxet->cell_dimensions() ),
333     &X0[1]
334 );
335 pois_rat_1->property_accessor()->set_real_value( Gr
idObjTopoHandle( cscellindex,
336     CS_voxet->cell_dimensions() ),
337     &X0[2]
338 );
339 pois_rat_3->property_accessor()->set_real_value( Gr

```

```

    idObjTopoHandle( cscellindex,
340                                     CS_voxet->cell_dimensions() ),
341                                     &X0[3]
342                                     );
343                                     mod_shear_3->property_accessor()->set_real_value( G
    ridObjTopoHandle( cscellindex,
344                                     CS_voxet->cell_dimensions() ),
345                                     &X0[4]
346                                     );
347                                     }
348                                     }
349     }
350
351 //END EXECUTOR
352
353 //DEBUG
354 //      GObj* obj = GObjAPI::copy( *(cs_cell.solid_model()), "output", true,
    true, true );
355 //
356 //      GeobaseLib::instance()->add_gobj( obj );
357
358 //      GObjAPI::apply_script( cs_cell.solid_model(), CopyCString( "X+=prob_C
    S_voxet_0_ShearU[0];Y+=prob_CS_voxet_0_ShearU[1];Z+=prob_CS_voxet_0_ShearU[2];"
    );
359
360 // Cleaning ptrs
361 for( int i = 0; i<3; ++i) {
362     delete prop_grids[i];
363 }
364 for( PtrList<Experiments>::iterator exp_itr( exp_list ); exp_itr.more(); exp
    _itr.next() ) {
365     delete exp_itr.cur();
366 }
367 delete voxets_link;
368
369 cout << "Upscaling finished in ";
370 delete time;
371 cout << "seconds" << endl;
372
373 return true;
374 }

```

The documentation for this class was generated from the following files:

- [include/GoUpScale/gapi/mech\\_upscale\\_api.h](#)
- [src/lib/gapi/mech\\_upscale\\_api.cpp](#)

## 7.6 GoUpScale::S3D Class Reference

Class for managing Atom\* tabs[][][].

```
#include <cs_cell.h>
```

### Public Member Functions

- [S3D](#) (int size)
- virtual [~S3D](#) ()
- const Atom \* [get](#) (int u, int v, int w) const
- Atom \*& [operator](#)() (int u, int v, int w)

### Data Fields

- int [size\\_](#)
- int [size2\\_](#)
- Atom \*\* [tab\\_](#)

#### 7.6.1 Detailed Description

Class for managing Atom\* tabs[][][].

Definition at line 131 of file cs\_cell.h.

#### 7.6.2 Constructor & Destructor Documentation

##### 7.6.2.1 GoUpScale::S3D::S3D (int size) [inline]

Definition at line 133 of file cs\_cell.h.

```
133           : size_( size+1 ), size2_( (size+1)*(size+1) ), tab_(  
134     new Atom*[size_*size2_] ) {  
135     };
```

##### 7.6.2.2 virtual GoUpScale::S3D::~S3D () [inline, virtual]

Definition at line 135 of file cs\_cell.h.

References [tab\\_](#).

```
135     {  
136     delete[] tab_;  
137     };
```

#### 7.6.3 Member Function Documentation

##### 7.6.3.1 const Atom\* GoUpScale::S3D::get (int u, int v, int w) const [inline]

Definition at line 139 of file cs\_cell.h.

References [goupscale\\_assert](#), [size2\\_](#), [size\\_](#), and [tab\\_](#).

```
139                                     {
140         goupscale_assert( u>0 && u<size_ );
141         goupscale_assert( v>0 && v<size_ );
142         goupscale_assert( w>0 && w<size_ );
143
144         return tab_[u+v*size_+w*size2_];
145     };
```

### 7.6.3.2 Atom\* & GoUpScale::S3D::operator() (int u, int v, int w) [inline]

Definition at line 147 of file cs\_cell.h.

References `size2_`, `size_`, and `tab_`.

```
147                                     {
148         return tab_[u+v*size_+w*size2_];
149     };
```

## 7.6.4 Field Documentation

### 7.6.4.1 int GoUpScale::S3D::size2\_

Definition at line 153 of file cs\_cell.h.

### 7.6.4.2 int GoUpScale::S3D::size\_

Definition at line 149 of file cs\_cell.h.

### 7.6.4.3 Atom\*\* GoUpScale::S3D::tab\_

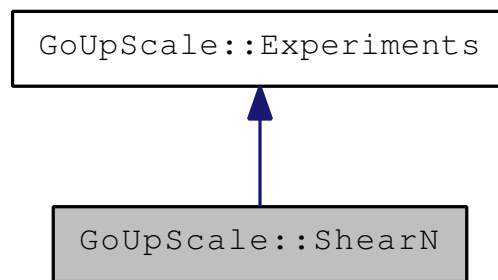
Definition at line 154 of file cs\_cell.h.

The documentation for this class was generated from the following file:

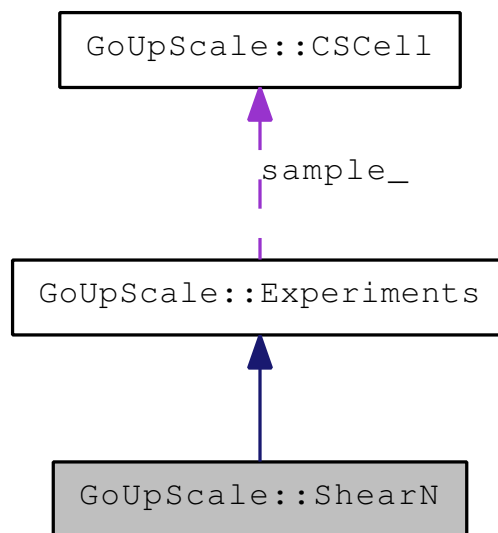
- [include/GoUpScale/gridding/cs\\_cell.h](#)

## 7.7 GoUpScale::ShearN Class Reference

#include <experiments.h>Inheritance diagram for GoUpScale::ShearN:



Collaboration diagram for GoUpScale::ShearN:



### Public Member Functions

- `ShearN (CSCell *sample, const double displacement, const bool large_deformation)`  
*Constructor of the class.*
- `virtual ~ShearN ()`  
*Destructor of the class.*
- `virtual void boundary_conditions ()`  
*Sets the boundary conditions for experiment `ShearN`.*
- `const CopyCString type ()`
- `const CopyCString direction ()`

### Data Fields

- `ShearN`



## 7.7.1 Detailed Description

Implements a traction in the U direction

Definition at line 313 of file experiments.h.

## 7.7.2 Constructor & Destructor Documentation

### 7.7.2.1 GoUpScale::ShearN::ShearN (CSCell \* *sample*, const double *displacement*, const bool *large\_deformation*)

Constructor of the class.

#### Parameters:

*sample* [CSCell](#) on which the experiments are currently performed

*displacement* value of the displacement to apply during the experiments (the displacement is the same for all the experiments)

*large\_deformation* sets whether the experiments must be ran in large deformation or not

### 7.7.2.2 GoUpScale::ShearN::~~ShearN () [**virtual**]

Destructor of the class.

Definition at line 467 of file experiments.cpp.

```
467         {
468     }
```

## 7.7.3 Member Function Documentation

### 7.7.3.1 void GoUpScale::ShearN::boundary\_conditions () [**virtual**]

Sets the boundary conditions for experiment [ShearN](#).

Implements [GoUpScale::Experiments](#).

Definition at line 479 of file experiments.cpp.

References [GoUpScale::Experiments::displacement\(\)](#), [GoUpScale::Experiments::prob\\_name\(\)](#), and [GoUpScale::Experiments::sample\(\)](#).

```
479         {
480     // ShearN
481     bool is_moving6[3] = {false, true , true};
482     float fixed61[3] = {0., 0., 0.};
483     SolidRestorationAPI::set_fixed_components(
484         sample()->solid_model(),
485         prob_name(),
486         2,
487         CopyCString( sample()->solid_model()->name()->string() ) + "_face5+",
488         is_moving6,
489         fixed61,
490         "Static",
491         0.9
492     );
493     float fixed62[3] = {0., displacement(), 0.};
494     SolidRestorationAPI::set_fixed_components(
495         sample()->solid_model(),
```

```

496         prob_name(),
497         2,
498         CopyCString( sample()->solid_model()->name()->string() ) + "_face6+",
499         is_moving6,
500         fixed62,
501         "Static",
502         0.9
503     );
504
505
506     return;
507 }

```

### 7.7.3.2 const CopyCString GoUpScale::ShearN::direction () [virtual]

#### Returns:

the direction of the experiment

Implements [GoUpScale::Experiments](#).

Definition at line 474 of file experiments.cpp.

```

474         {
475     return CopyCString( "N" );
476 }

```

### 7.7.3.3 const CopyCString GoUpScale::ShearN::type () [virtual]

#### Returns:

what kind of experiment this (traction, shear,...)

Implements [GoUpScale::Experiments](#).

Definition at line 470 of file experiments.cpp.

```

470         {
471     return CopyCString( "Shear" );
472 }

```

## 7.7.4 Field Documentation

### 7.7.4.1 GoUpScale::ShearN::ShearN

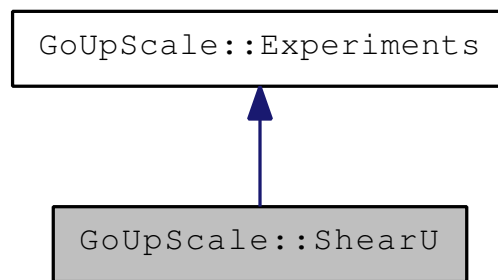
Definition at line 317 of file experiments.h.

The documentation for this class was generated from the following files:

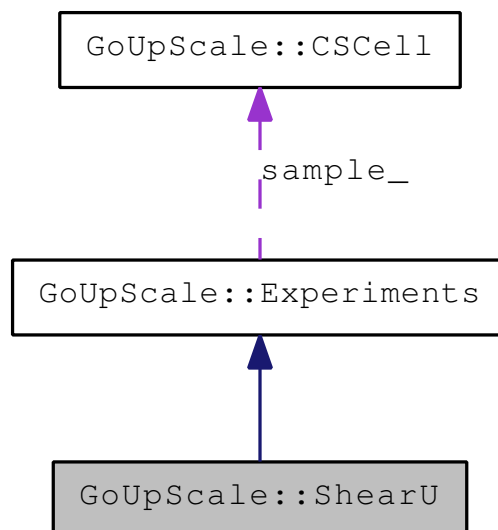
- [include/GoUpScale/mechanics/experiments.h](#)
- [src/lib/mechanics/experiments.cpp](#)

## 7.8 GoUpScale::ShearU Class Reference

#include <experiments.h> Inheritance diagram for GoUpScale::ShearU:



Collaboration diagram for GoUpScale::ShearU:



### Public Member Functions

- `ShearU (CSCell *sample, const double displacement, const bool large_deformation)`  
*Constructor of the class.*
- `virtual ~ShearU ()`  
*Destructor of the class.*
- `virtual void boundary_conditions ()`  
*Sets the boundary conditions for experiment `ShearU`.*
- `const CopyCString type ()`
- `const CopyCString direction ()`

### Data Fields

- `ShearU`

## 7.8.1 Detailed Description

Implements a traction in the U direction

Definition at line 243 of file experiments.h.

## 7.8.2 Constructor & Destructor Documentation

### 7.8.2.1 GoUpScale::ShearU::ShearU (CSCCell \* *sample*, const double *displacement*, const bool *large\_deformation*)

Constructor of the class.

#### Parameters:

*sample* [CSCCell](#) on which the experiments are currently performed

*displacement* value of the displacement to apply during the experiments (the displacement is the same for all the experiments)

*large\_deformation* sets whether the experiments must be ran in large deformation or not

### 7.8.2.2 GoUpScale::ShearU::~~ShearU () [virtual]

Destructor of the class.

Definition at line 367 of file experiments.cpp.

```
367         {
368     }
```

## 7.8.3 Member Function Documentation

### 7.8.3.1 void GoUpScale::ShearU::boundary\_conditions () [virtual]

Sets the boundary conditions for experiment [ShearU](#).

Implements [GoUpScale::Experiments](#).

Definition at line 379 of file experiments.cpp.

References [GoUpScale::Experiments::displacement\(\)](#), [GoUpScale::Experiments::prob\\_name\(\)](#), and [GoUpScale::Experiments::sample\(\)](#).

```
379         {
380     // ShearU
381
382     bool is_moving4[3] = {true, false, true};
383     float move41[3] = {0., 0., 0.};
384     SolidRestorationAPI::set_fixed_components(
385         sample()->solid_model(),
386         prob_name(),
387         2,
388         CopyCString( sample()->solid_model()->name()->string() ) + "_face1+",
389         is_moving4,
390         move41,
391         "Static",
392         0.9
393     );
394
395     float move42[3] = {0., 0., displacement()};
```

```

396     SolidRestorationAPI::set_fixed_components (
397         sample()->solid_model(),
398         prob_name(),
399         2,
400         CopyCString( sample()->solid_model()->name()->string() ) + "_face2+",
401         is_moving4,
402         move42,
403         "Static",
404         0.9
405     );
406
407     return;
408 }

```

### 7.8.3.2 const CopyCString GoUpScale::ShearU::direction () [virtual]

#### Returns:

the direction of the experiment

Implements [GoUpScale::Experiments](#).

Definition at line 374 of file experiments.cpp.

```

374     {
375     return CopyCString( "U" );
376 }

```

### 7.8.3.3 const CopyCString GoUpScale::ShearU::type () [virtual]

#### Returns:

what kind of experiment this (traction, shear,...)

Implements [GoUpScale::Experiments](#).

Definition at line 370 of file experiments.cpp.

```

370     {
371     return CopyCString( "Shear" );
372 }

```

## 7.8.4 Field Documentation

### 7.8.4.1 GoUpScale::ShearU::ShearU

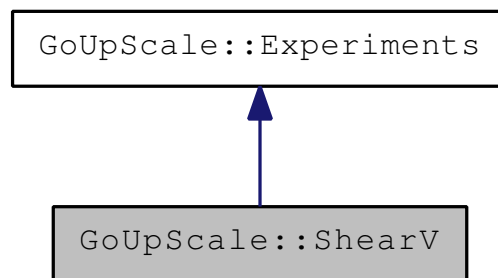
Definition at line 247 of file experiments.h.

The documentation for this class was generated from the following files:

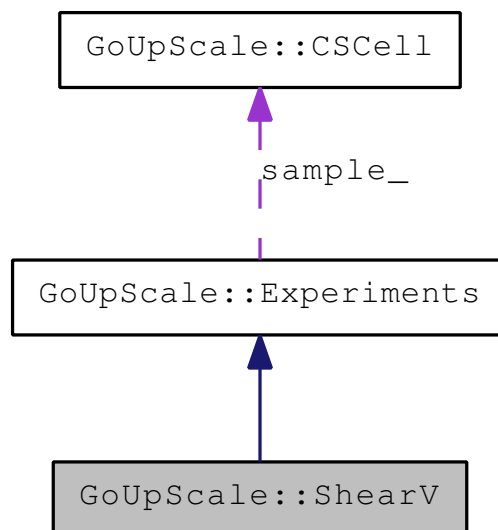
- include/GoUpScale/mechanics/[experiments.h](#)
- src/lib/mechanics/[experiments.cpp](#)

## 7.9 GoUpScale::ShearV Class Reference

#include <experiments.h>Inheritance diagram for GoUpScale::ShearV:



Collaboration diagram for GoUpScale::ShearV:



### Public Member Functions

- [ShearV](#) ([CSCell](#) \*sample, const double displacement, const bool large\_deformation)  
*Constructor of the class.*
- virtual [~ShearV](#) ()  
*Destructor of the class.*
- virtual void [boundary\\_conditions](#) ()  
*Sets the boundary conditions for experiment [ShearV](#).*
- const CopyCString [type](#) ()
- const CopyCString [direction](#) ()

### Data Fields

- [ShearV](#)

## 7.9.1 Detailed Description

Implements a traction in the U direction

Definition at line 278 of file experiments.h.

## 7.9.2 Constructor & Destructor Documentation

### 7.9.2.1 GoUpScale::ShearV::ShearV (CSCell \* *sample*, const double *displacement*, const bool *large\_deformation*)

Constructor of the class.

#### Parameters:

*sample* [CSCell](#) on which the experiments are currently performed

*displacement* value of the displacement to apply during the experiments (the displacement is the same for all the experiments)

*large\_deformation* sets whether the experiments must be ran in large deformation or not

### 7.9.2.2 GoUpScale::ShearV::~~ShearV () [**virtual**]

Destructor of the class.

Definition at line 418 of file experiments.cpp.

```
418         {
419     }
```

## 7.9.3 Member Function Documentation

### 7.9.3.1 void GoUpScale::ShearV::boundary\_conditions () [**virtual**]

Sets the boundary conditions for experiment [ShearV](#).

Implements [GoUpScale::Experiments](#).

Definition at line 430 of file experiments.cpp.

References [GoUpScale::Experiments::displacement\(\)](#), [GoUpScale::Experiments::prob\\_name\(\)](#), and [GoUpScale::Experiments::sample\(\)](#).

```
430         {
431     // ShearV
432     bool is_moving5[3] = {false, true, true};
433     float fixed51[3] = {0., 0., 0.};
434     SolidRestorationAPI::set_fixed_components(
435         sample()->solid_model(),
436         prob_name(),
437         2,
438         CopyCString( sample()->solid_model()->name()->string() ) + "_face3+",
439         is_moving5,
440         fixed51,
441         "Static",
442         0.9
443     );
444     float move52[3] = {0., 0., displacement()};
445     SolidRestorationAPI::set_fixed_components(
446         sample()->solid_model(),
```

```

447         prob_name(),
448         2,
449         CopyCString( sample()->solid_model()->name()->string() ) + "_face4+",
450         is_moving5,
451         move52,
452         "Static",
453         0.9
454     );
455
456     return;
457 }

```

### 7.9.3.2 const CopyCString GoUpScale::ShearV::direction () [virtual]

#### Returns:

the direction of the experiment

Implements [GoUpScale::Experiments](#).

Definition at line 425 of file experiments.cpp.

```

425         {
426     return CopyCString( "V" );
427 }

```

### 7.9.3.3 const CopyCString GoUpScale::ShearV::type () [virtual]

#### Returns:

what kind of experiment this (traction, shear,...)

Implements [GoUpScale::Experiments](#).

Definition at line 421 of file experiments.cpp.

```

421         {
422     return CopyCString( "Shear" );
423 }

```

## 7.9.4 Field Documentation

### 7.9.4.1 GoUpScale::ShearV::ShearV

Definition at line 282 of file experiments.h.

The documentation for this class was generated from the following files:

- [include/GoUpScale/mechanics/experiments.h](#)
- [src/lib/mechanics/experiments.cpp](#)



## 7.10 GoUpScale::StrainMeanVector< T > Class Template Reference

```
#include <strain_mean_vector.h>
```

### Public Member Functions

- [StrainMeanVector](#) (int *taille*)
- [~StrainMeanVector](#) ()
- void [start\\_aggregate](#) (const PtrList< [Experiments](#) > &exps)

### Protected Member Functions

- void [aggregate](#) (const [Experiments](#) \*exp, int ik)

### 7.10.1 Detailed Description

```
template<class T> class GoUpScale::StrainMeanVector< T >
```

Definition at line 51 of file strain\_mean\_vector.h.

### 7.10.2 Constructor & Destructor Documentation

**7.10.2.1** `template<class T> GoUpScale::StrainMeanVector< T >::StrainMeanVector (int taille) [inline]`

Definition at line 54 of file strain\_mean\_vector.h.

```
54                                     : DenseVector<T>( taille ), taille_( t
55   taille ) {
56   };
```

**7.10.2.2** `template<class T> GoUpScale::StrainMeanVector< T >::~~StrainMeanVector () [inline]`

Definition at line 57 of file strain\_mean\_vector.h.

```
57   {
58   };
```

### 7.10.3 Member Function Documentation

**7.10.3.1** `template<class T> void GoUpScale::StrainMeanVector< T >::aggregate (const Experiments *exp, int ik) [inline, protected]`

Definition at line 75 of file strain\_mean\_vector.h.

References [GoUpScale::Experiments::get\\_strain\\_means\(\)](#), and [goupscale\\_assert](#).

```
75   {
76   }
77   goupscale_assert( nil != exp );
78   goupscale_assert( 4*ik+4 <= taille_ );
79 }
```

```

80         DenseVector<double> MeanStrainComponents( exp->get_strain
    _means() );
81
82         // Strain from Experiment ik
83         (*this)[4*ik+0] = MeanStrainComponents[0] + MeanStrainCom
    ponents[5];
84
85         (*this)[4*ik+1] = MeanStrainComponents[1] + MeanStrainCom
    ponents[5];
86
87         (*this)[4*ik+2] = MeanStrainComponents[2];
88
89         (*this)[4*ik+3] = 1.*( MeanStrainComponents[3] + MeanStra
    inComponents[4] );
90
91
92         return;
93     };

```

### 7.10.3.2 `template<class T> void GoUpScale::StrainMeanVector< T >::start_aggregate (const PtrList< Experiments > & exps) [inline]`

Definition at line 60 of file `strain_mean_vector.h`.

References `GoUpScale::StrainMeanVector< T >::aggregate()`.

```

60
61         // Initialization
62         this->set( 0. );
63
64         int ik = 0;
65         for( PtrList<Experiments>::iterator itr( exps ); itr.more
    (); itr.next() ) {
66
67             aggregate( itr.cur(), ik );
68             ++ik;
69         }
70         return;
71     };

```

The documentation for this class was generated from the following file:

- `include/GoUpScale/maths/strain\_mean\_vector.h`

## 7.11 GoUpScale::StressMeanMatrix< T > Class Template Reference

Aggregate a matrix from the stress components of the experiments on a [CSCell](#).

```
#include <stress_mean_matrix.h>
```

### Public Member Functions

- [StressMeanMatrix](#) (int taille)
- [~StressMeanMatrix](#) ()
- void [start\\_aggregate](#) (const PtrList< [Experiments](#) > &exps)

### Protected Member Functions

- void [aggregate](#) (const [Experiments](#) \*exp, int ik)

#### 7.11.1 Detailed Description

**template<class T> class GoUpScale::StressMeanMatrix< T >**

Aggregate a matrix from the stress components of the experiments on a [CSCell](#).

#### Author:

Marc-Olivier Titeux

#### Version:

2.0

#### Date:

07-11-2007

Definition at line 53 of file stress\_mean\_matrix.h.

#### 7.11.2 Constructor & Destructor Documentation

**7.11.2.1 template<class T> GoUpScale::StressMeanMatrix< T >::StressMeanMatrix (int *taille*) [inline]**

Definition at line 56 of file stress\_mean\_matrix.h.

```
56                                     : DenseMatrix<T>( taille, 5 ), taille_  
57     ( taille ) {  
                                     };
```

**7.11.2.2 template<class T> GoUpScale::StressMeanMatrix< T >::~~StressMeanMatrix () [inline]**

Definition at line 59 of file stress\_mean\_matrix.h.

```
59                                     {  
60                                     };
```

## 7.11.3 Member Function Documentation

### 7.11.3.1 `template<class T> void GoUpScale::StressMeanMatrix< T >::aggregate (const Experiments * exp, int ik) [inline, protected]`

Definition at line 77 of file stress\_mean\_matrix.h.

References `GoUpScale::Experiments::get_stress_means()`, and `goupscale_assert`.

```
77                                     {
78                                     goupscale_assert( nil != exp );
79                                     goupscale_assert( 4*ik+4 <= taille_ );
80
81                                     DenseVector<double> MeanStressComponents( exp->get_stress
82                                     _means() );
83
84                                     // Stress from Experiment ik
85                                     (*this)( 4*ik + 0, 0 ) = MeanStressComponents[0] + Mean
86                                     StressComponents[5];
87                                     (*this)( 4*ik + 0, 2 ) = - MeanStressComponents[1] + Mean
88                                     StressComponents[5];
89                                     (*this)( 4*ik + 0, 3 ) = - MeanStressComponents[2];
90                                     (*this)( 4*ik + 1, 0 ) = MeanStressComponents[1] + Mean
91                                     StressComponents[5];
92                                     (*this)( 4*ik + 1, 2 ) = - MeanStressComponents[0] + Mean
93                                     StressComponents[5];
94                                     (*this)( 4*ik + 1, 3 ) = - MeanStressComponents[2];
95                                     (*this)( 4*ik + 2, 1 ) = MeanStressComponents[2];
96                                     (*this)( 4*ik + 2, 3 ) = - MeanStressComponents[0] - Mean
97                                     StressComponents[1];
98                                     (*this)( 4*ik + 3, 4 ) = 1.*( MeanStressComponents[3] +
99                                     MeanStressComponents[4] );
100
101                                     return;
102
103                                     };
```

### 7.11.3.2 `template<class T> void GoUpScale::StressMeanMatrix< T >::start_aggregate (const PtrList< Experiments > & exps) [inline]`

Definition at line 62 of file stress\_mean\_matrix.h.

References `GoUpScale::StressMeanMatrix< T >::aggregate()`.

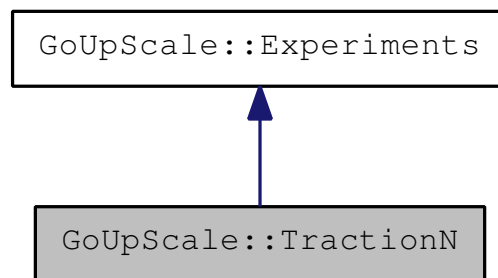
```
62                                     {
63                                     // Initialization
64                                     this->set( 0. );
65
66                                     int ik = 0;
67                                     for( PtrList<Experiments>::iterator itr( exps ); itr.more
68                                     (); itr.next() ) {
69                                     aggregate( itr.cur(), ik );
70                                     ++ik;
71                                     }
72                                     return;
73                                     };
```

The documentation for this class was generated from the following file:

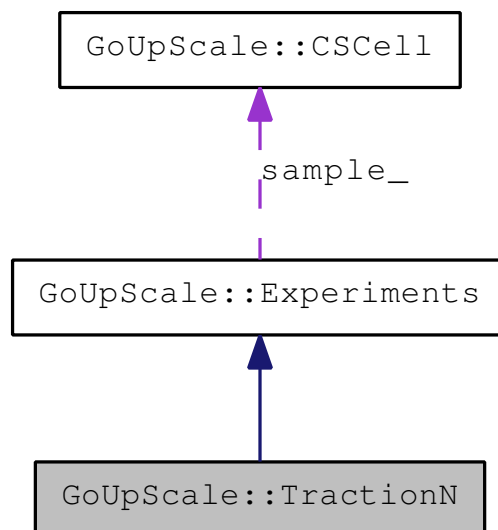
- [include/GoUpScale/maths/stress\\_mean\\_matrix.h](#)

## 7.12 GoUpScale::TractionN Class Reference

#include <experiments.h> Inheritance diagram for GoUpScale::TractionN:



Collaboration diagram for GoUpScale::TractionN:



### Public Member Functions

- [TractionN](#) ([CSCell](#) \*sample, const double displacement, const bool large\_deformation)  
*Constructor of the class.*
- virtual [~TractionN](#) ()  
*Destructor of the class.*
- virtual void [boundary\\_conditions](#) ()  
*Sets the boundary conditions for experiment [TractionN](#).*
- const CopyCString [type](#) ()
- const CopyCString [direction](#) ()

### Data Fields

- [TractionN](#)

## 7.12.1 Detailed Description

Implements a traction in the U direction

Definition at line 208 of file experiments.h.

## 7.12.2 Constructor & Destructor Documentation

### 7.12.2.1 GoUpScale::TractionN::TractionN (CSCell \* *sample*, const double *displacement*, const bool *large\_deformation*)

Constructor of the class.

#### Parameters:

*sample* [CSCell](#) on which the experiments are currently performed

*displacement* value of the displacement to apply during the experiments (the displacement is the same for all the experiments)

*large\_deformation* sets whether the experiments must be ran in large deformation or not

### 7.12.2.2 GoUpScale::TractionN::~~TractionN () [virtual]

Destructor of the class.

Definition at line 317 of file experiments.cpp.

```
317         {
318     }
```

## 7.12.3 Member Function Documentation

### 7.12.3.1 void GoUpScale::TractionN::boundary\_conditions () [virtual]

Sets the boundary conditions for experiment [TractionN](#).

Implements [GoUpScale::Experiments](#).

Definition at line 329 of file experiments.cpp.

References [GoUpScale::Experiments::displacement\(\)](#), [GoUpScale::Experiments::prob\\_name\(\)](#), and [GoUpScale::Experiments::sample\(\)](#).

```
329         {
330     // TractionN
331     bool is_fixed31[3] = {false, false, true};
332     float fixed31[3] = {0., 0., -displacement()/2.};
333     SolidRestorationAPI::set_fixed_components(
334         sample()->solid_model(),
335         prob_name(),
336         2,
337         CopyCString( sample()->solid_model()->name()->string() ) + "_face5+",
338         is_fixed31,
339         fixed31,
340         "Static",
341         0.9
342     );
343     bool is_moving32[3] = {false, false, true};
344     float move32[3] = {0., 0., displacement()/2.};
345     SolidRestorationAPI::set_fixed_components(
```

```

346         sample()->solid_model(),
347         prob_name(),
348         2,
349         CopyCString( sample()->solid_model()->name()->string() ) + "_face6+",
350         is_moving32,
351         move32,
352         "Static",
353         0.9
354     );
355
356     return;
357 }

```

### 7.12.3.2 const CopyCString GoUpScale::TractionN::direction () [virtual]

#### Returns:

the direction of the experiment

Implements [GoUpScale::Experiments](#).

Definition at line 324 of file experiments.cpp.

```

324         {
325     return CopyCString( "N" );
326 }

```

### 7.12.3.3 const CopyCString GoUpScale::TractionN::type () [virtual]

#### Returns:

what kind of experiment this (traction, shear,...)

Implements [GoUpScale::Experiments](#).

Definition at line 320 of file experiments.cpp.

```

320         {
321     return CopyCString( "Traction" );
322 }

```

## 7.12.4 Field Documentation

### 7.12.4.1 GoUpScale::TractionN::TractionN

Definition at line 212 of file experiments.h.

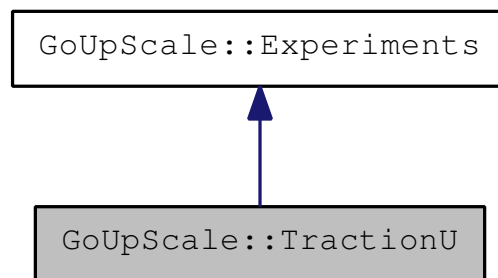
The documentation for this class was generated from the following files:

- [include/GoUpScale/mechanics/experiments.h](#)
- [src/lib/mechanics/experiments.cpp](#)

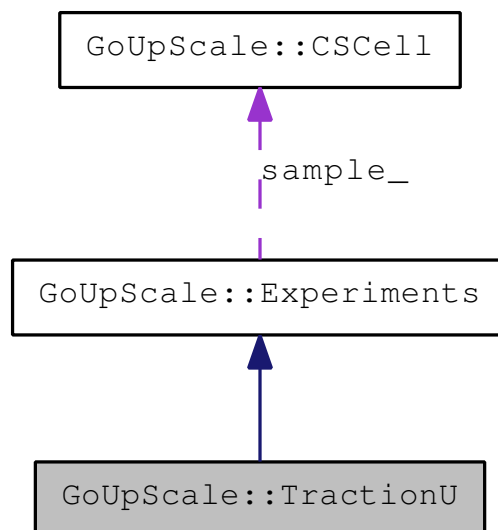


## 7.13 GoUpScale::TractionU Class Reference

#include <experiments.h> Inheritance diagram for GoUpScale::TractionU:



Collaboration diagram for GoUpScale::TractionU:



### Public Member Functions

- [TractionU](#) ([CSCell](#) \*sample, const double displacement, const bool large\_deformation)  
*Constructor of the class.*
- virtual [~TractionU](#) ()  
*Destructor of the class.*
- void [boundary\\_conditions](#) ()  
*Sets the boundary conditions for experiment [TractionU](#).*
- const CopyCString [type](#) ()
- const CopyCString [direction](#) ()

### Data Fields

- [TractionU](#)

## 7.13.1 Detailed Description

Implements a traction in the U direction

Definition at line 138 of file experiments.h.

## 7.13.2 Constructor & Destructor Documentation

### 7.13.2.1 GoUpScale::TractionU::TractionU (CSCell \* *sample*, const double *displacement*, const bool *large\_deformation*)

Constructor of the class.

#### Parameters:

*sample* [CSCell](#) on which the experiments are currently performed

*displacement* value of the displacement to apply during the experiments (the displacement is the same for all the experiments)

*large\_deformation* sets whether the experiments must be ran in large deformation or not

### 7.13.2.2 GoUpScale::TractionU::~~TractionU () [virtual]

Destructor of the class.

Definition at line 225 of file experiments.cpp.

```
225         {
226     }
```

## 7.13.3 Member Function Documentation

### 7.13.3.1 void GoUpScale::TractionU::boundary\_conditions () [virtual]

Sets the boundary conditions for experiment [TractionU](#).

Implements [GoUpScale::Experiments](#).

Definition at line 229 of file experiments.cpp.

References [GoUpScale::Experiments::displacement\(\)](#), [GoUpScale::Experiments::prob\\_name\(\)](#), and [GoUpScale::Experiments::sample\(\)](#).

```
229         {
230     // TractionU
231     bool is_fixed11[3] = {true, false, false};
232     float fixed11[3] = {-displacement()/2., 0., 0.};
233     SolidRestorationAPI::set_fixed_components(
234         sample()->solid_model(),
235         prob_name(),
236         2,
237         CopyCString( sample()->solid_model()->name()->string() ) + "_face1+",
238         is_fixed11,
239         fixed11,
240         "Static",
241         0.9
242     );
243     bool is_moving12[3] = {true, false, false};
244     float move12[3] = {displacement()/2., 0., 0.};
245     SolidRestorationAPI::set_fixed_components(
```

```

246         sample()->solid_model(),
247         prob_name(),
248         2,
249         CopyCString( sample()->solid_model()->name()->string() ) + "_face2+",
250         is_moving12,
251         move12,
252         "Static",
253         0.9
254     );
255
256     return;
257 }

```

### 7.13.3.2 const CopyCString GoUpScale::TractionU::direction () [virtual]

#### Returns:

the direction of the experiment

Implements [GoUpScale::Experiments](#).

Definition at line 219 of file experiments.cpp.

```

219         {
220     return CopyCString( "U" );
221 }

```

### 7.13.3.3 const CopyCString GoUpScale::TractionU::type () [virtual]

#### Returns:

what kind of experiment this (traction, shear,...)

Implements [GoUpScale::Experiments](#).

Definition at line 215 of file experiments.cpp.

```

215         {
216     return CopyCString( "Traction" );
217 }

```

## 7.13.4 Field Documentation

### 7.13.4.1 GoUpScale::TractionU::TractionU

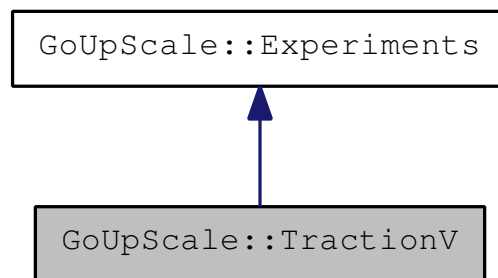
Definition at line 142 of file experiments.h.

The documentation for this class was generated from the following files:

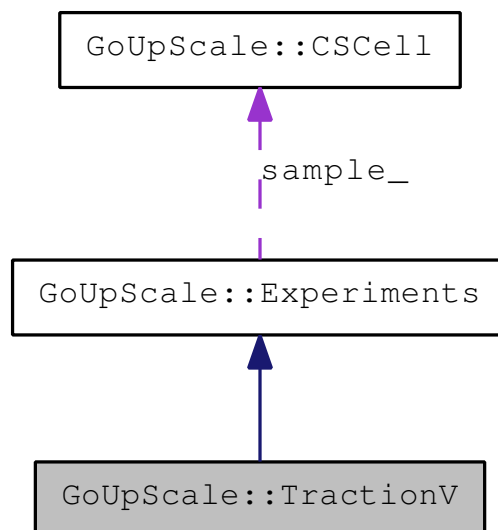
- [include/GoUpScale/mechanics/experiments.h](#)
- [src/lib/mechanics/experiments.cpp](#)

## 7.14 GoUpScale::TractionV Class Reference

#include <experiments.h> Inheritance diagram for GoUpScale::TractionV:



Collaboration diagram for GoUpScale::TractionV:



### Public Member Functions

- [TractionV](#) ([CSCell](#) \*sample, const double displacement, const bool large\_deformation)  
*Constructor of the class.*
- virtual [~TractionV](#) ()  
*Destructor of the class.*
- virtual void [boundary\\_conditions](#) ()  
*Sets the boundary conditions for experiment [TractionV](#).*
- const CopyCString [type](#) ()
- const CopyCString [direction](#) ()

### Data Fields

- [TractionV](#)

## 7.14.1 Detailed Description

Implements a traction in the V direction

Definition at line 173 of file experiments.h.

## 7.14.2 Constructor & Destructor Documentation

### 7.14.2.1 GoUpScale::TractionV::TractionV (CSCell \* *sample*, const double *displacement*, const bool *large\_deformation*)

Constructor of the class.

#### Parameters:

*sample* [CSCell](#) on which the experiments are currently performed

*displacement* value of the displacement to apply during the experiments (the displacement is the same for all the experiments)

*large\_deformation* sets whether the experiments must be ran in large deformation or not

### 7.14.2.2 GoUpScale::TractionV::~~TractionV () [virtual]

Destructor of the class.

Definition at line 267 of file experiments.cpp.

```
267         {
268     }
```

## 7.14.3 Member Function Documentation

### 7.14.3.1 void GoUpScale::TractionV::boundary\_conditions () [virtual]

Sets the boundary conditions for experiment [TractionV](#).

Implements [GoUpScale::Experiments](#).

Definition at line 279 of file experiments.cpp.

References [GoUpScale::Experiments::displacement\(\)](#), [GoUpScale::Experiments::prob\\_name\(\)](#), and [GoUpScale::Experiments::sample\(\)](#).

```
279         {
280     // TractionV
281     bool is_fixed21[3] = {false, true, false};
282     float fixed21[3] = {0., -displacement()/2., 0.};
283     SolidRestorationAPI::set_fixed_components(
284         sample()->solid_model(),
285         prob_name(),
286         2,
287         CopyCString( sample()->solid_model()->name()->string() ) + "_face3+",
288         is_fixed21,
289         fixed21,
290         "Static",
291         0.9
292     );
293     bool is_moving22[3] = {false, true, false};
294     float move22[3] = {0., displacement()/2., 0.};
295     SolidRestorationAPI::set_fixed_components(
```

```

296         sample()->solid_model(),
297         prob_name(),
298         2,
299         CopyCString( sample()->solid_model()->name()->string() ) + "_face4+",
300         is_moving22,
301         move22,
302         "Static",
303         0.9
304     );
305
306     return;
307 }

```

### 7.14.3.2 `const CopyCString GoUpScale::TractionV::direction () [virtual]`

#### Returns:

the direction of the experiment

Implements [GoUpScale::Experiments](#).

Definition at line 274 of file experiments.cpp.

```

274         {
275     return CopyCString( "V" );
276 }

```

### 7.14.3.3 `const CopyCString GoUpScale::TractionV::type () [virtual]`

#### Returns:

what kind of experiment this (traction, shear,...)

Implements [GoUpScale::Experiments](#).

Definition at line 270 of file experiments.cpp.

```

270         {
271     return CopyCString( "Traction" );
272 }

```

## 7.14.4 Field Documentation

### 7.14.4.1 `GoUpScale::TractionV::TractionV`

Definition at line 177 of file experiments.h.

The documentation for this class was generated from the following files:

- [include/GoUpScale/mechanics/experiments.h](#)
- [src/lib/mechanics/experiments.cpp](#)

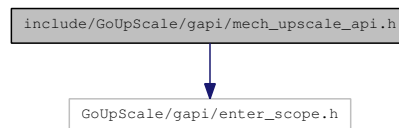
# Chapter 8

## File Documentation

### 8.1 include/GoUpScale/gapi/mech\_upscale\_api.h File Reference

```
#include <GoUpScale/gapi/enter_scope.h>
```

Include dependency graph for mech\_upscale\_api.h:



#### Data Structures

- class [GoUpScale::MechUpscaleAPI](#)

#### Namespaces

- namespace [Gocad](#)
- namespace [GoUpScale](#)

## 8.2 include/GoUpScale/gridding/cs\_cell.h File Reference

```
#include <assert.h>
#include <GoUpScale/gridding/enter_scope.h>
#include <GoUpScale/gridding/fs_cell.h>
#include <RestorationLab/solidmodel/volumes/solid_model.h>
#include <Gocad/utils/misc/rtti.h>
#include <Gocad/utils/containers/sequence.h>
#include <Gocad/utils/containers/list.h>
#include <Gocad/math/misc/defs.h>
```

Include dependency graph for cs\_cell.h:



## Data Structures

- class [GoUpScale::CSCell](#)  
*Class for coarse-scale cells, made from coarse-scale description /a Voxet.*
- class [GoUpScale::S3D](#)  
*Class for managing `Atom* tabs[][][]`.*

## Namespaces

- namespace [Gocad](#)
- namespace [GoUpScale](#)

### 8.2.1 Detailed Description

Definition in file [cs\\_cell.h](#).



## 8.3 include/GoUpScale/gridding/fs\_cell.h File Reference

```
#include <GoUpScale/gridding/enter_scope.h>
#include <GoUpScale/utils/assert.h>
#include <Gocad/utils/misc/rtti.h>
#include <Gocad/utils/containers/set.h>
#include <Gocad/math/misc/defs.h>
#include <Gocad/volumes/griddata/gridpropdata.h>
#include <Gocad/geobase/base/association.h>
#include <Gocad/volumes/tsolid/tetra.h>
#include <RestorationLab/problems/material/material.h>
```

Include dependency graph for fs\_cell.h:



### Data Structures

- class [GoUpScale::FSCell](#)  
*Abstract class for cellules mapping within voxets.*

### Namespaces

- namespace [Gocad](#)
- namespace [GoUpScale](#)

#### 8.3.1 Detailed Description

Definition in file [fs\\_cell.h](#).

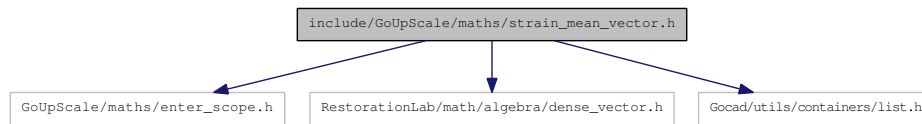
## 8.4 include/GoUpScale/maths/strain\_mean\_vector.h File Reference

```
#include <GoUpScale/maths/enter_scope.h>
```

```
#include <RestorationLab/math/algebra/dense_vector.h>
```

```
#include <Gocad/utills/containers/list.h>
```

Include dependency graph for strain\_mean\_vector.h:



### Data Structures

- class [GoUpScale::StrainMeanVector< T >](#)

### Namespaces

- namespace [GoUpScale](#)
- namespace [Gocad](#)

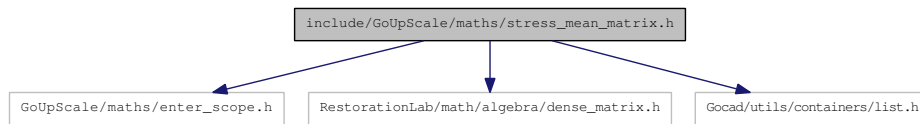
## 8.5 include/GoUpScale/maths/stress\_mean\_matrix.h File Reference

```
#include <GoUpScale/maths/enter_scope.h>
```

```
#include <RestorationLab/math/algebra/dense_matrix.h>
```

```
#include <Gocad/utills/containers/list.h>
```

Include dependency graph for stress\_mean\_matrix.h:



### Data Structures

- class [GoUpScale::StressMeanMatrix< T >](#)

*Aggregate a matrix from the stress components of the experiments on a [CSCell](#).*

### Namespaces

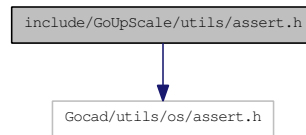
- namespace [GoUpScale](#)
- namespace [Gocad](#)



## 8.7 include/GoUpScale/utils/assert.h File Reference

```
#include <Gocad/utils/os/assert.h>
```

Include dependency graph for assert.h:



### Defines

- `#define goup_scale_assert(cond) GOCAD_ASSERT(cond, abort(); )`

#### 8.7.1 Define Documentation

##### 8.7.1.1 `#define goup_scale_assert(cond) GOCAD_ASSERT(cond, abort(); )`

Definition at line 23 of file `assert.h`.

## 8.8 src/lib/gapi/mech\_upscale\_api.cpp File Reference

```
#include <GoUpScale/gapi/mech_upscale_api.h>
#include <GoUpScale/gridding/cs_cell.h>
#include <GoUpScale/gridding/fs_cell.h>
#include <GoUpScale/mechanics/experiments.h>
#include <GoUpScale/math/stress_mean_matrix.h>
#include <GoUpScale/math/strain_mean_vector.h>
#include <RestorationLab/solidmodel/volumes/solid_model.h>
#include <RestorationLab/math/algebra/dense_matrix.h>
#include <RestorationLab/math/solvers/superlu.h>
#include <Gocad/geobase/base/association.h>
#include <Gocad/geobase/base/geobaselib.h>
#include <Gocad/geobase/base/gobj.h>
#include <Gocad/geobase/property/property.h>
#include <Gocad/volumes/voxet/voxet.h>
#include <Gocad/volumes/voxet/grid3d.h>
#include <Gocad/volumes/griddata/datacube.h>
#include <Gocad/volumes/griddata/gridpropdata.h>
#include <Gocad/gapi/property_api.h>
#include <Gocad/gapi/gobj_api.h>
#include <Gocad/utils/gocad/jobtracer.h>
#include <Gocad/utils/os/clock.h>
```

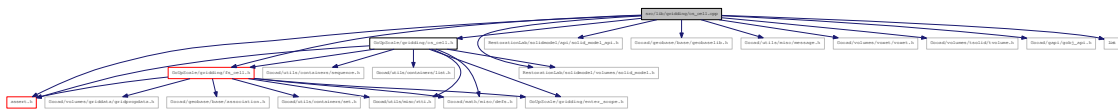
### Namespaces

- namespace [GoUpScale](#)

## 8.9 src/lib/gridding/cs\_cell.cpp File Reference

```
#include <GoUpScale/gridding/cs_cell.h>
#include <GoUpScale/gridding/fs_cell.h>
#include <GoUpScale/utils/assert.h>
#include <RestorationLab/solidmodel/volumes/solid_model.h>
#include <RestorationLab/solidmodel/api/solid_model_api.h>
#include <Gocad/geobase/base/geobaselib.h>
#include <Gocad/utils/misc/message.h>
#include <Gocad/volumes/voxet/voxet.h>
#include <Gocad/volumes/tsolid/tvolume.h>
#include <Gocad/gapi/gobj_api.h>
#include <list>
```

Include dependency graph for cs\_cell.cpp:



### Namespaces

- namespace [GoUpScale](#)

### Variables

- [GoUpScale::CSCell](#)

## 8.10 src/lib/gridding/fs\_cell.cpp File Reference

```
#include <GoUpScale/gridding/fs_cell.h>
#include <Gocad/utils/misc/message.h>
#include <Gocad/volumes/voxet/voxet.h>
#include <RestorationLab/problems/material/material_library.h>
```

Include dependency graph for fs\_cell.cpp:



### Namespaces

- namespace [GoUpScale](#)

### Variables

- [GoUpScale::FSCell](#)



## 8.11 src/lib/mechanics/experiments.cpp File Reference

```
#include <GoUpScale/mechanics/experiments.h>
#include <GoUpScale/utils/assert.h>
#include <RestorationLab/problems/base/problem.h>
#include <RestorationLab/math/algebra/dense_matrix.h>
#include <RestorationLab/math/algebra/dense_vector.h>
#include <RestorationLab/solidmodel/volumes/solid_model.h>
#include <RestorationLab/strains/strain_analyser.h>
#include <RestorationLab/gapi/restoration_api.h>
#include <Gocad/gapi/property_api.h>
#include <RestorationLab/math/algebra/matrix3x3.h>
```

Include dependency graph for experiments.cpp:



### Namespaces

- namespace [GoUpScale](#)

### Variables

- [GoUpScale::Experiments](#)
- [GoUpScale::TractionU](#)
- [GoUpScale::TractionV](#)
- [GoUpScale::TractionN](#)
- [GoUpScale::ShearU](#)
- [GoUpScale::ShearV](#)
- [GoUpScale::ShearN](#)

# Index

- ~CSCell
  - GoUpScale::CSCell, [16](#)
- ~Experiments
  - GoUpScale::Experiments, [26](#)
- ~FSCell
  - GoUpScale::FSCell, [34](#)
- ~S3D
  - GoUpScale::S3D, [46](#)
- ~ShearN
  - GoUpScale::ShearN, [49](#)
- ~ShearU
  - GoUpScale::ShearU, [52](#)
- ~ShearV
  - GoUpScale::ShearV, [55](#)
- ~StrainMeanVector
  - GoUpScale::StrainMeanVector, [57](#)
- ~StressMeanMatrix
  - GoUpScale::StressMeanMatrix, [59](#)
- ~TractionN
  - GoUpScale::TractionN, [63](#)
- ~TractionU
  - GoUpScale::TractionU, [66](#)
- ~TractionV
  - GoUpScale::TractionV, [69](#)
- aggregate
  - GoUpScale::StrainMeanVector, [57](#)
  - GoUpScale::StressMeanMatrix, [60](#)
- assert.h
  - goupscale\_assert, [77](#)
- boundary\_conditions
  - GoUpScale::Experiments, [27](#)
  - GoUpScale::ShearN, [49](#)
  - GoUpScale::ShearU, [52](#)
  - GoUpScale::ShearV, [55](#)
  - GoUpScale::TractionN, [63](#)
  - GoUpScale::TractionU, [66](#)
  - GoUpScale::TractionV, [69](#)
- build\_fine\_description
  - GoUpScale::CSCell, [16](#)
- build\_material\_from\_voxel\_properties
  - GoUpScale::FSCell, [34](#)
- build\_stiffness\_tensor
  - GoUpScale::FSCell, [35](#)
- cell\_index
  - GoUpScale::CSCell, [22](#)
  - GoUpScale::FSCell, [36](#)
- comp\_state
  - GoUpScale::Experiments, [27](#)
- compute\_means
  - GoUpScale::Experiments, [27](#)
- computed
  - GoUpScale::Experiments, [28](#)
- CSCell
  - GoUpScale, [12](#)
  - GoUpScale::CSCell, [16](#)
- direction
  - GoUpScale::Experiments, [29](#)
  - GoUpScale::ShearN, [50](#)
  - GoUpScale::ShearU, [53](#)
  - GoUpScale::ShearV, [56](#)
  - GoUpScale::TractionN, [64](#)
  - GoUpScale::TractionU, [67](#)
  - GoUpScale::TractionV, [70](#)
- displacement
  - GoUpScale::Experiments, [29](#)
- Experiment, [24](#)
- Experiments
  - GoUpScale, [12](#)
  - GoUpScale::Experiments, [26, 31](#)
- flush
  - GoUpScale::Experiments, [29](#)
- FSCell
  - GoUpScale, [12](#)
  - GoUpScale::FSCell, [33](#)
- get
  - GoUpScale::S3D, [46](#)
- get\_children
  - GoUpScale::CSCell, [22](#)
- get\_stiffness
  - GoUpScale::FSCell, [36](#)
- get\_strain\_means
  - GoUpScale::Experiments, [29](#)
- get\_stress\_means
  - GoUpScale::Experiments, [29](#)
- Gocad, [11](#)
- GoUpScale, [12](#)
  - CSCell, [12](#)
  - Experiments, [12](#)
  - FSCell, [12](#)

- ShearN, 12
- ShearU, 13
- ShearV, 13
- TractionN, 13
- TractionU, 13
- TractionV, 13
- GoUpScale::CSCell, 15
  - ~CSCell, 16
  - build\_fine\_description, 16
  - cell\_index, 22
  - CSCell, 16
  - get\_children, 22
  - set\_cell\_index, 22
  - set\_materials, 22
  - solid\_model, 23
  - voxets\_link, 23
- GoUpScale::Experiments, 25
  - ~Experiments, 26
  - boundary\_conditions, 27
  - comp\_state, 27
  - compute\_means, 27
  - computed, 28
  - direction, 29
  - displacement, 29
  - Experiments, 26, 31
  - flush, 29
  - get\_strain\_means, 29
  - get\_stress\_means, 29
  - init, 29
  - prob\_name, 30
  - run, 30
  - sample, 31
  - set\_prob\_name, 31
  - type, 31
- GoUpScale::FSCell, 33
  - ~FSCell, 34
  - build\_material\_from\_voxel\_properties, 34
  - build\_stiffness\_tensor, 35
  - cell\_index, 36
  - FSCell, 33
  - get\_stiffness, 36
  - material, 36
  - set\_cell\_index, 36
  - set\_material, 36
  - tetra\_list, 36, 37
  - voxets\_link, 37
- GoUpScale::MechUpscaleAPI, 38
  - mean\_prop\_on\_voxel, 38
  - upscale, 39
- GoUpScale::S3D, 46
  - ~S3D, 46
  - get, 46
  - operator(), 47
  - S3D, 46
  - size2\_, 47
  - size\_, 47
- tab\_, 47
- GoUpScale::ShearN, 48
  - ~ShearN, 49
  - boundary\_conditions, 49
  - direction, 50
  - ShearN, 49, 50
  - type, 50
- GoUpScale::ShearU, 51
  - ~ShearU, 52
  - boundary\_conditions, 52
  - direction, 53
  - ShearU, 52, 53
  - type, 53
- GoUpScale::ShearV, 54
  - ~ShearV, 55
  - boundary\_conditions, 55
  - direction, 56
  - ShearV, 55, 56
  - type, 56
- GoUpScale::StrainMeanVector, 57
  - ~StrainMeanVector, 57
  - aggregate, 57
  - start\_aggregate, 58
  - StrainMeanVector, 57
- GoUpScale::StressMeanMatrix, 59
  - ~StressMeanMatrix, 59
  - aggregate, 60
  - start\_aggregate, 60
  - StressMeanMatrix, 59
- GoUpScale::TractionN, 62
  - ~TractionN, 63
  - boundary\_conditions, 63
  - direction, 64
  - TractionN, 63, 64
  - type, 64
- GoUpScale::TractionU, 65
  - ~TractionU, 66
  - boundary\_conditions, 66
  - direction, 67
  - TractionU, 66, 67
  - type, 67
- GoUpScale::TractionV, 68
  - ~TractionV, 69
  - boundary\_conditions, 69
  - direction, 70
  - TractionV, 69, 70
  - type, 70
- goupscale\_assert
  - assert.h, 77
- include/GoUpScale/gapi/mech\_upscale\_api.h, 71
- include/GoUpScale/gridding/cs\_cell.h, 72
- include/GoUpScale/gridding/fs\_cell.h, 73
- include/GoUpScale/math/strain\_mean\_vector.h, 74
- include/GoUpScale/math/stress\_mean\_matrix.h, 75
- include/GoUpScale/mechanics/experiments.h, 76

- include/GoUpScale/utils/assert.h, 77
- init
  - GoUpScale::Experiments, 29
- material
  - GoUpScale::FSCell, 36
- mean\_prop\_on\_voxel
  - GoUpScale::MechUpscaleAPI, 38
- operator()
  - GoUpScale::S3D, 47
- prob\_name
  - GoUpScale::Experiments, 30
- run
  - GoUpScale::Experiments, 30
- S3D
  - GoUpScale::S3D, 46
- sample
  - GoUpScale::Experiments, 31
- set\_cell\_index
  - GoUpScale::CSCell, 22
  - GoUpScale::FSCell, 36
- set\_material
  - GoUpScale::FSCell, 36
- set\_materials
  - GoUpScale::CSCell, 22
- set\_prob\_name
  - GoUpScale::Experiments, 31
- ShearN
  - GoUpScale, 12
  - GoUpScale::ShearN, 49, 50
- ShearU
  - GoUpScale, 13
  - GoUpScale::ShearU, 52, 53
- ShearV
  - GoUpScale, 13
  - GoUpScale::ShearV, 55, 56
- size2\_
  - GoUpScale::S3D, 47
- size\_
  - GoUpScale::S3D, 47
- solid\_model
  - GoUpScale::CSCell, 23
- src/lib/gapi/mech\_upscale\_api.cpp, 78
- src/lib/gridding/cs\_cell.cpp, 79
- src/lib/gridding/fs\_cell.cpp, 80
- src/lib/mechanics/experiments.cpp, 81
- start\_aggregate
  - GoUpScale::StrainMeanVector, 58
  - GoUpScale::StressMeanMatrix, 60
- StrainMeanVector
  - GoUpScale::StrainMeanVector, 57
- StressMeanMatrix
  - GoUpScale::StressMeanMatrix, 59
- tab\_
  - GoUpScale::S3D, 47
- tetra\_list
  - GoUpScale::FSCell, 36, 37
- TractionN
  - GoUpScale, 13
  - GoUpScale::TractionN, 63, 64
- TractionU
  - GoUpScale, 13
  - GoUpScale::TractionU, 66, 67
- TractionV
  - GoUpScale, 13
  - GoUpScale::TractionV, 69, 70
- type
  - GoUpScale::Experiments, 31
  - GoUpScale::ShearN, 50
  - GoUpScale::ShearU, 53
  - GoUpScale::ShearV, 56
  - GoUpScale::TractionN, 64
  - GoUpScale::TractionU, 67
  - GoUpScale::TractionV, 70
- upscale
  - GoUpScale::MechUpscaleAPI, 39
- voxets\_link
  - GoUpScale::CSCell, 23
  - GoUpScale::FSCell, 37

# Bibliography

M.-O. Titeux and J. J. Royer. Upscaling Mechanical Properties in Layered Geological Formations. In *28<sup>th</sup> Gocad Meeting Proceedings*, 2008.