
MuPoSim Documentation

Release 0.1

Robert Cimrman, Eduard Rohan

April 09, 2010

CONTENTS

1 Abstract	3
1.1 User's Guide	3
1.2 Developer Guide	4
2 Indices and tables	7
Module Index	9
Index	11

MuPoSim (Multiscale Porous Simulator) is an application project for implementing hierarchical multiscale models of porous media built upon SfePy.

ABSTRACT

The SW allows for computing poroelastic coefficients of porous material consisting of the solid and fluid phases which both are connected, i.e. the solid skeleton forms a connected domain and the pores form a connected network as well. In response to a static loading the pressure associated with the fluid in the pores is constant. The specific geometry of the microstructure and the solid and fluid elasticity coefficients are required as the input data. The homogenization method provides effective elastic and poroelastic coefficients of the Biot model (relevant to a quasi-static loading). Then also the so-called “undrained test” poroelasticity coefficients can be computed. The SW provides the option of a hierarchical homogenization of a porous material perforated by fluid pores. This enables one to obtain homogenized poroelasticity coefficients for multi-porous structures which may have connected, or disconnected porosities. The double porosity medium is one special case covered within the modeling options.

Keywords: porous material; poroelasticity coefficients; homogenization; microstructure;

Contents:

1.1 User’s Guide

The purpose of this software is to compute poroelastic coefficients in multiporous materials in 2D and 3D.

1.1.1 Introduction

1.1.2 Running a simulation

Just run the *three_levels.py* script:

```
python three_levels.py
```

Even though it is tailored for a three levels of connected porosities at the moment, it can be easily adapted to another kind of hierarchy.

1.1.3 Postprocessing

Results of a simulation are:

- the homogenized coefficients for various levels of the multiporous hierarchy in a text file or a HDF5 file,
- solutions of the related corrector problems in the standard VTK format or a custom HDF5-based format.

1.2 Developer Guide

This section purports to document the *MuPoSim* internals.

1.2.1 three_levels module

```
get_coefs_1 (coefs2, regenerate=True)
```

```
get_coefs_2 (coefs3, regenerate=True)
```

```
get_coefs_3 (coefs4, regenerate=True)
```

```
main ()
```

Save regions using:

```
from sfepy.fem import ProblemDefinition
problem = ProblemDefinition.from_conf(conf, init_variables=False)
problem.save_regions_as_groups('regions')
```

1.2.2 micro package

micro.micro_1 module

```
class MicroConf1 (filename, approx, region_selects, mat_pars, incwd, coefs_save_name='coefs',
                  corr_save_names=None, output_dir=None, post_process_hook=None)
```

micro.micro_2 module

```
class CorrectorsPressure (name, problem, kwargs)
```

```
    get_variables (data)
        data: None
```

```
class MHatCoef (name, problem, kwargs)
```

```
    get_variables (problem, data)
```

```
class MicroConf2 (filename, approx, region_selects, mat_pars, incwd, coefs_save_name='coefs',
                  corr_save_names=None, output_dir=None, post_process_hook=None)
```

```
    define_coefs ()
```

```
    define_equations ()
```

```
    define_fields ()
```

```
    define_materials ()
```

```
    define_requirements ()
```

```
    define_variables ()
```

```
class NCoef2_1 (name, problem, kwargs)
```

```
    get_variables (problem, data)
```



```
class NCoef2_2 (name, problem, kwargs)
```

```
    get_variables (problem, data)
```

```
class NCoef2_3 (name, problem, kwargs)
```

```
    get_variables (problem, data)
```

```
class PoroElasticCoef2_1 (name, problem, kwargs)
    Homogenized Biot-like coefficient.
```

```
    get_variables (problem, ir, ic, data, mode)
```

```
class PoroElasticCoef2_2 (name, problem, kwargs)
    Homogenized Biot-like coefficient.
```

```
    get_variables (problem, ir, ic, data, mode)
```

micro.micro_3 module

```
class MicroConf3 (filename, approx, region_selects, mat_pars, incwd,
                  coefs_save_name='coefs',
                  corrs_save_names=None, output_dir=None, post_process_hook=None)
```

```
    define_coefs ()
```

```
    define_equations ()
```

```
    define_materials ()
```

```
    define_requirements ()
```

```
class PoroElasticCoef3_1 (name, problem, kwargs)
    Homogenized Biot-like coefficient.
```

```
    get_variables (problem, ir, ic, data, mode)
```

```
class PoroElasticCoef3_2 (name, problem, kwargs)
    Homogenized Biot-like coefficient.
```

```
    get_variables (problem, ir, ic, data, mode)
```

micro.micro_on_level module

```
class MicroConf (filename, approx, region_selects, mat_pars, incwd,
                 coefs_save_name='coefs',
                 corrs_save_names=None, output_dir=None, post_process_hook=None)
```

```
    static any_by_name (name, *args, **kwargs)
```

Create an instance of a MicroConf class according to the configuration.

```
    define_bcs ()
```

```
    define_coefs ()
```

```
    define_equations ()
```

```
    define_fields ()
```

```
    define_materials ()
```

```
    define_regions ()
```

```
define_requirements ()
```

```
define_variables ()
```

```
static post_process (out, pb, state, extend=False)
```

```
    Strain and stress for given displacements.
```

```
class SurfaceMomentCoef (name, problem, kwargs)
```

INDICES AND TABLES

- *Index*
- *Module Index*
- *Search Page*

MODULE INDEX

M

micro.micro_1, 4
micro.micro_2, 4
micro.micro_3, 5
micro.micro_on_level, 5

T

three_levels, 4

INDEX

A

any_by_name() (micro.micro_on_level.MicroConf static method), 5

C

CorrectorsPressure (class in micro.micro_2), 4

D

define_bcs() (micro.micro_on_level.MicroConf method), 5

define_coefs() (micro.micro_2.MicroConf2 method), 4

define_coefs() (micro.micro_3.MicroConf3 method), 5

define_coefs() (micro.micro_on_level.MicroConf method), 5

define_equations() (micro.micro_2.MicroConf2 method), 4

define_equations() (micro.micro_3.MicroConf3 method), 5

define_equations() (micro.micro_on_level.MicroConf method), 5

define_fields() (micro.micro_2.MicroConf2 method), 4

define_fields() (micro.micro_on_level.MicroConf method), 5

define_materials() (micro.micro_2.MicroConf2 method), 4

define_materials() (micro.micro_3.MicroConf3 method), 5

define_materials() (micro.micro_on_level.MicroConf method), 5

define_regions() (micro.micro_on_level.MicroConf method), 5

define_requirements() (micro.micro_2.MicroConf2 method), 4

define_requirements() (micro.micro_3.MicroConf3 method), 5

define_requirements() (micro.micro_on_level.MicroConf method), 5

define_variables() (micro.micro_2.MicroConf2 method), 4

define_variables() (micro.micro_on_level.MicroConf method), 6

G

get_coefs_1() (in module three_levels), 4

get_coefs_2() (in module three_levels), 4

get_coefs_3() (in module three_levels), 4

get_variables() (micro.micro_2.CorrectorsPressure method), 4

get_variables() (micro.micro_2.MHatCoef method), 4

get_variables() (micro.micro_2.NCoef2_1 method), 4

get_variables() (micro.micro_2.NCoef2_2 method), 5

get_variables() (micro.micro_2.NCoef2_3 method), 5

get_variables() (micro.micro_2.PoroElasticCoef2_1 method), 5

get_variables() (micro.micro_2.PoroElasticCoef2_2 method), 5

get_variables() (micro.micro_3.PoroElasticCoef3_1 method), 5

get_variables() (micro.micro_3.PoroElasticCoef3_2 method), 5

M

main() (in module three_levels), 4

MHatCoef (class in micro.micro_2), 4

micro.micro_1 (module), 4

micro.micro_2 (module), 4

micro.micro_3 (module), 5

micro.micro_on_level (module), 5

MicroConf (class in micro.micro_on_level), 5

MicroConf1 (class in micro.micro_1), 4

MicroConf2 (class in micro.micro_2), 4

MicroConf3 (class in micro.micro_3), 5

N

NCoef2_1 (class in micro.micro_2), 4

NCoef2_2 (class in micro.micro_2), 4

NCoef2_3 (class in micro.micro_2), 5

P

PoroElasticCoef2_1 (class in micro.micro_2), 5

PoroElasticCoef2_2 (class in micro.micro_2), 5

PoroElasticCoef3_1 (class in micro.micro_3), 5

PoroElasticCoef3_2 (class in micro.micro_3), 5

post_process() (micro.micro_on_level.MicroConf static method), 6

S

SurfaceMomentCoef (class in micro.micro_on_level), 6

T

three_levels (module), 4