BioDeg

Biodegradation and Corrosion Simulation using Finite Element

User Manual

Version 0.8

(generated August 30, 2021)

Mojtaba Barzegari
Liesbet Geris

BioDeg website

KU LEUVEN

Contents

1 Introduction

BioDeg is an open source software written in FreeFEM (a domain-specific language for finite element programming), C++, and Python for modeling the degradation of metallic biomaterials and simulating the biodegradation behavior of medical devices, implants and scaffolds in corrosion experiments. It can handle any geometry of desire and supports parallel computing to simulate models with a high level of complexity.

1.1 Authors

BioDeg is developed by the Biomechanics Research group at KU Leuven and University of Liege. The code is currently maintained by its principal developer, who manage the development of the mathematical models and the core functionalities.

Principal developer

- Mojtaba Barzegari (University of Leuven, Belgium).

Previous Contributors

- Yann Guyot (University of Liege, Belgium).
- Piotr Bajger (University of Oxford, UK).

Mentor

- Liesbet Geris (University of Leuven, Belgium).

1.2 Acknowledgments

The development of BioDeg open source code is financially supported by the Prosperos project, funded by the Interreg VA Flanders – The Netherlands program, CCI grant no. 2014TC16RFCB046 and by the Fund for Scientific Research Flanders (FWO), grant G085018N. The developers also acknowledge support from the European Research Council under the European Union's Horizon 2020 research and innovation programmen, ERC CoG 772418.

1.3 Referencing BioDeg

Please refer to BioDeg repository, section "Publications and referencing" to properly cite the use of BioDeg in your scientific work.

2 Useful background information

You may refer to the following articles for a background of the methods and algorithms implemented in BioDeg.

In addition, below are some useful references on finite element method and some online resources that provide a background of finite elements and their application to the solution of partial differential equations.

3 Installation

All the underlying installation instructions assume a Linux operating system. We assume standard tools and libraries like CMake, compilers- (C, C++ and Fortran), and MPI libraries are pre-installed. Most high-performance computers would have the latest version of these libraries in the default environment.

3.1 Compiling and installing external libraries

3.1.1 Instructions for PETSc and Qt

3.1.2 Instructions for FreeFEM

3.2 Obtaining and Compiling BioDeg

Assuming that you have already installed the above external dependencies, next follow the steps below to obtain and compile BioDeg.

4 Running BioDeg

4.1 Configuring the simulation

4.1.1 Example 1

4.1.2 Example 2

5 Future extensions to BioDeg

The future versions of BioDeg will focus on the implementation of the following methodologies/features.

-

6 Finding answers to more questions

If you have questions that go beyond this manual, there are a number of resources:

- For questions/suggestions about BioDeg installation, bugs, or similar stuff please use the BioDeg issue tracker.
- BioDeg is primarily based on the FreeFEM. If you have particular questions about FreeFEM, contact the community at `https://community.freefem.org/`.
- If you have specific questions about BioDeg that are not suitable for public and archived mailing lists, you can contact the primary developer and mentor:
  - Mojtaba Barzegari: `mojtaba.barzegari@kuleuven.be`.
  - Liesbet Geris: `liesbet.geris@kuleuven.be` (Mentor).

A Run-time input parameters

The underlying description of the input parameters also includes a "Standard/Advanced/Developer" label, which signifies whether an input parameter is a standard one, or an advanced level parameter, or a developer level one only meant for development purposes. The default values of the "Advanced" and "Developer" labelled parameters are good enough for almost all cases. However, in some cases user may need to use "Advanced" labelled parameters. For user convenience, all input parameters are also indexed at the end of this manual in Section A.2.

A.1 Mesh and geometry parameters

- *Parameter name:* `mesh_file`
  Default: Should be provided
  Description: [Standard]
  Possible values: Any string value
- *Parameter name:* `label_scaffold`
  Default: 1
  Description: [Standard]
  Possible values: Any positive integer value
- *Parameter name:* `label_medium`
  Default: 2
  Description: [Standard]
  Possible values: Any positive integer value
- *Parameter name:* `label_wall`
  Default: 3
  Description: [Advanced]
  Possible values: Any positive integer value
- *Parameter name:* `label_inlet`
  Default: 4
  Description: [Advanced]
  Possible values: Any positive integer value
- *Parameter name:* `label_outlet`
  Default: 5
  Description: [Advanced]
  Possible values: Any positive integer value
- *Parameter name:* `box_length`
  Default: 20.0
  Description: [Standard]
  Possible values: Any positive floating point number
- *Parameter name:* `cube_size_x`
  Default: 13.0
  Description: [Standard]
  Possible values: Any positive floating point number
- *Parameter name:* `cube_size_y`
  Default: 13.0
  Description: [Standard]
  Possible values: Any positive floating point number

- *Parameter name:* `cube_size_z`
  Default: 4.0
  Description: [Standard]
  Possible values: Any positive floating point number
- *Parameter name:* `mesh_size`
  Default: 32
  Description: [Standard]
  Possible values: Any positive integer number
- *Parameter name:* `refine_initial_mesh`
  Default: false (0)
  Description: [Advanced]
  Possible values: A boolean value (1 or 0)
- *Parameter name:* `mshmet_error`
  Default: 0.01
  Description: [Advanced]
  Possible values: Any floating point number
- *Parameter name:* `mesh_size_min`
  Default: 0.04
  Description: [Advanced]
  Possible values: Any floating point number
- *Parameter name:* `mesh_size_max`
  Default: 0.8
  Description: [Advanced]
  Possible values: Any floating point number

A.2 Materials and boundary conditions parameters

- *Parameter name:* `material_density`
  Default: 1.735e-3
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `film_density`
  Default: 2.3446e-3
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `material_satur`
  Default: 0.134e-3
  Description: [Advanced]
  Possible values: Any floating point number

- *Parameter name:* `material_eps`
  Default: 0.55
  Description: [Advanced]
  Possible values: Any floating point number between 0 and 1
- *Parameter name:* `material_tau`
  Default: 1.0
  Description: [Advanced]
  Possible values: Any floating point number
- *Parameter name:* `d_mg`
  Default: 0.05
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `d_cl`
  Default: 0.05
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `d_oh`
  Default: 25.2
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `k1`
  Default: 7.0
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `k2`
  Default: 1e15
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `fluid_nu`
  Default: 0.85
  Description: [Advanced]
  Possible values: Any floating point number
- *Parameter name:* `fluid_in_x`
  Default: 0.1
  Description: [Advanced]
  Possible values: Any floating point number

- *Parameter name:* `fluid_in_y`
  Default: 0
  Description: [Advanced]
  Possible values: Any floating point number
- *Parameter name:* `fluid_in_z`
  Default: 0
  Description: [Advanced]
  Possible values: Any floating point number
- *Parameter name:* `initial_cl`
  Default: 5.175e-6
  Description: [Standard]
  Possible values: Any floating point number
- *Parameter name:* `initial_oh`
  Default: 1e-7
  Description: [Standard]
  Possible values: Any floating point number

Index of run-time parameters with section names

The following is a listing of all run-time parameters, sorted by the section in which they appear.