Yade Technical Archive (2019)
Document Link
www.yade-dem.org

COMSOL-Yade Interface (ICY) instruction guide

Pouyan Pirnia[1,*], Francois Duhaime[1], Yannic Ethier[1], and Jean-Sébastien Dubé[1]

[1]École de technologie supérieure, Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), Montreal, Quebec
[*]seyed-pouyan.pirnia.1@ens.etsmtl.ca

Keywords: COMSOL; Yade; fluid-particle coupling; install guide; ICY

1 Introduction

This guide outlines the installation of ICY, which is a software interface published in the paper titled *Pirnia, Pouyan, et al. "ICY: An interface between COMSOL multiphysics and discrete element code YADE for the modelling of porous media." Computers & Geosciences 123 (2019): 38-46.* ICY is an interface between COMSOL Multiphysics finite element engine and YADE ("Yet Another Dynamical Engine") open source discrete element code. COMSOL is capable of solving systems of partial differential equations to model multiple physical phenomena simultaneously, such as flow, seepage, chemical reactions, stress-strain behaviour and heat transfer. YADE is capable of solving highly flexible and dynamic particulate simulations. User input for COMSOL is accomplished using a GUI while YADE input is controlled by Python scripts. Meanwhile, the ICY interface enables communication between YADE and COMSOL by a JAVA class. Two project folders named Verification and Application codes are provided by the authors (Pirnia et al., 2018) (file download). The Verification project code simulates a particle falling in water according to Stokes' law while the Application project code simulates an internal erosion test. The project files include JAVA classes (ICY.java, Clientcaller.java, Reader.java), property files (define.properties), YADE interface scripts (test.py), client-server scripts (client.py and server.py), mesh file (4.mesh), test0.yade (including initial specimen composed of two layers of glass beads: a finer layer on top and a coarser layer at the bottom) and COMSOL models (test.mph). The two projects involve the exchange of different data between COMSOL and YADE. The files that differ for the two projects are test.py (YADE model), test.mph (COMSOL model) and the ICY.java class (main interface code). These are also the file that should be modified to create new applications for the interface.

2 Prerequisites

YADE and ICY are only executable on Linux operating systems. YADE, COMSOL and a JAVA integrated development environment (IDE) need to be installed before using the interface. The codes have been tested under Linux Ubuntu version 14.04 using YADE version 1.14.1 and COMSOL version 5.2. The JAVA classes were compiled and run using the NetBeans IDE version 8.0.2. The JAVA classes and Python scripts may need to be adapted if ICY is run with different versions of YADE and COMSOL, or under a different JAVA IDE.

3 Preparing the COMSOL model

The COMSOL file (test.mph) contains information on geometry, materials, fluid properties, boundary conditions, and mesh. The easiest way to edit these parameters or to define new ones is through COMSOL's graphical user interface (GUI). The main JAVA class can also modify the parameters of the COMSOL model, for example the particle velocity in the verification example, and the permeability values (k1, k2, k3, k4 or k5) or hydraulic head at the top of the specimen (Hupstream) in the Application example.

Figures 1-4 show how to prepare the GUI for the Application project code. Figure 1 shows how to define parameters using the COMSOL GUI. For the Application example, the initial parameter values are arbitrary as they are controlled by the interface. Figure 2 shows how to define the points where the pressure values at the top and bottom of each cell will be obtained from the COMSOL model. Figure 3 shows how the relationship between hydraulic conductivity and the z coordinate (x in COMSOL) is defined using the parameters defined in Figure 1. The GUI can also be used to modify any other parameter on the FEM side of the model, such as the finite-element mesh (Figure 4).

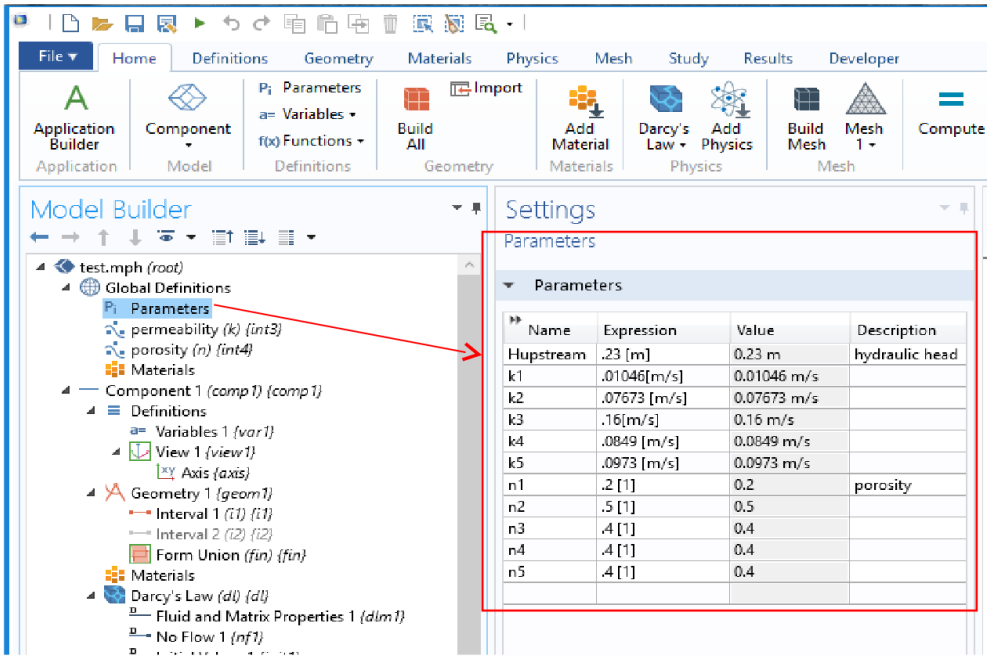

Figure 1. Parameter definition in COMSOL.

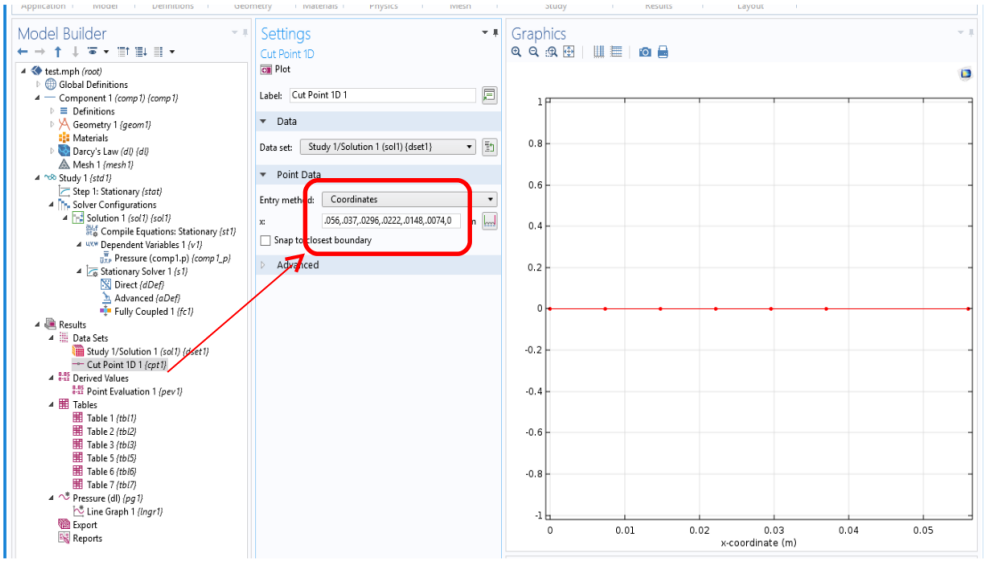

Figure 2. Definition of the points where p values will be obtained

4 Preparing the project in NetBeans

After creating an IDE project folder, the COMSOL plugins have to be added to the project through:
Properties → Libraries → Add JAR/Folder → add all .jar files in plugins folder in COMSOL installation folder (Figure 5).

For running the examples, the COMSOL file (test.mph), YADE script (test.py), client-server (client.py and server.py), mesh file (4.mesh), pressure file (pressure.txt including arbitrary initial pressures for the Application test), dragforce.txt (including drag force for the verification test), test0.yade (including initial specimen) and the JAVA source packages (src folder including ICY.java, Clientcaller.java, Reader.java, define.properties) need to be added to the IDE project folder (Figure 6).

The project's files ICY.java, Reader.java, Clientcaller.java and define.properties have first to be opened in NetBeans. The directories (MainPath and SavingFolder) in the property file (define.properties) have to be changed to correspond to the project directories on the computer, an example of the Application test was presented in Figure 7.

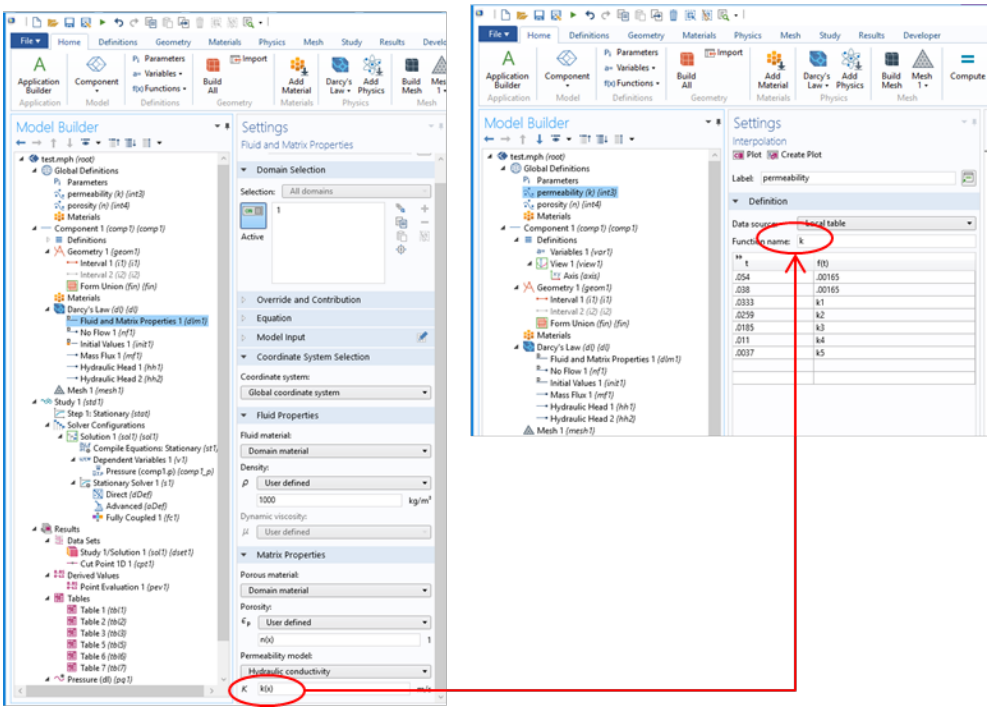

Figure 3. Definition of the hydraulic conductivity function and assignment to the domain

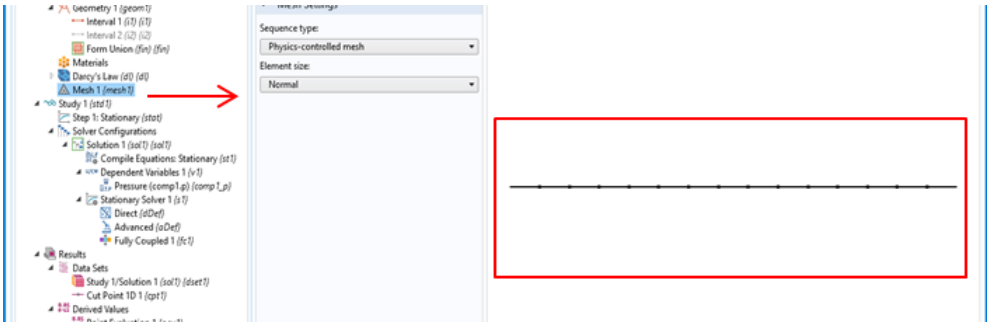

Figure 4. Mesh definition in COMSOL

Users do not need to change anything in Reader.java and clientcaller.java files for the application test. The command lines only need the YADE model and output files names which are taken from the property file (Figure 8).

Figure 9 shows the tasks are performed in the ICY.java class for the application test.

5 Preparing the YADE script

Parameters can be modified in the YADE script for the Application example as presented in Figure 10.

6 Running the interface

Before compiling ICY in NetBeans, the COMSOL server and the python client-server need to be launched. To start the COMSOL server manually, a terminal window is opened and the following command is typed in the COMSOL installation directory:

```
$ . / comsol mphserver
```

For connecting the client to the server, a second terminal window is opened. The following command is typed in the directory containing the server.py file (MainPath directory):

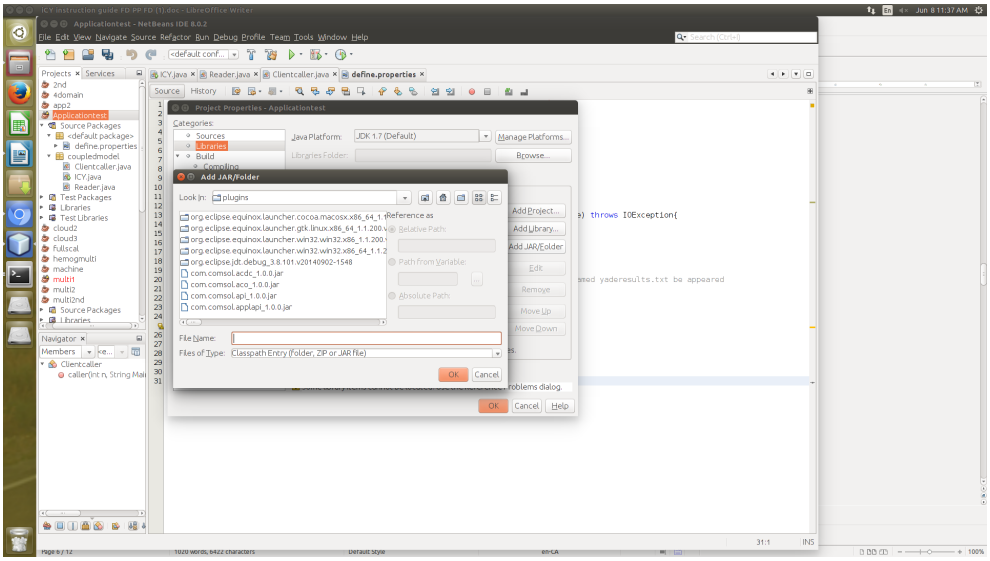

Figure 5. Add .jar files to plugins folder in COMSOL

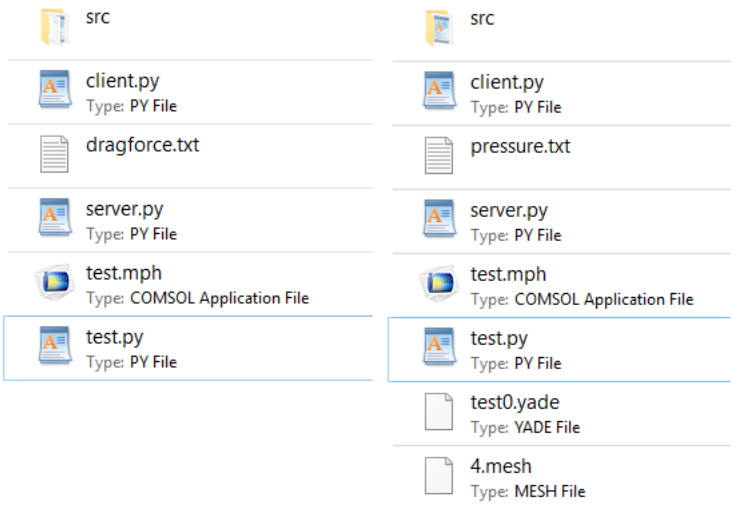

Figure 6. Required files and folder for running the a) Verification example and b) Application test.

```
$ python server.py
```

At this point, ICY can be compiled and run. The simulation progress is printed step by step on the NetBeans screen. It lets users follow the YADE and COMSOL outputs during the simulation.

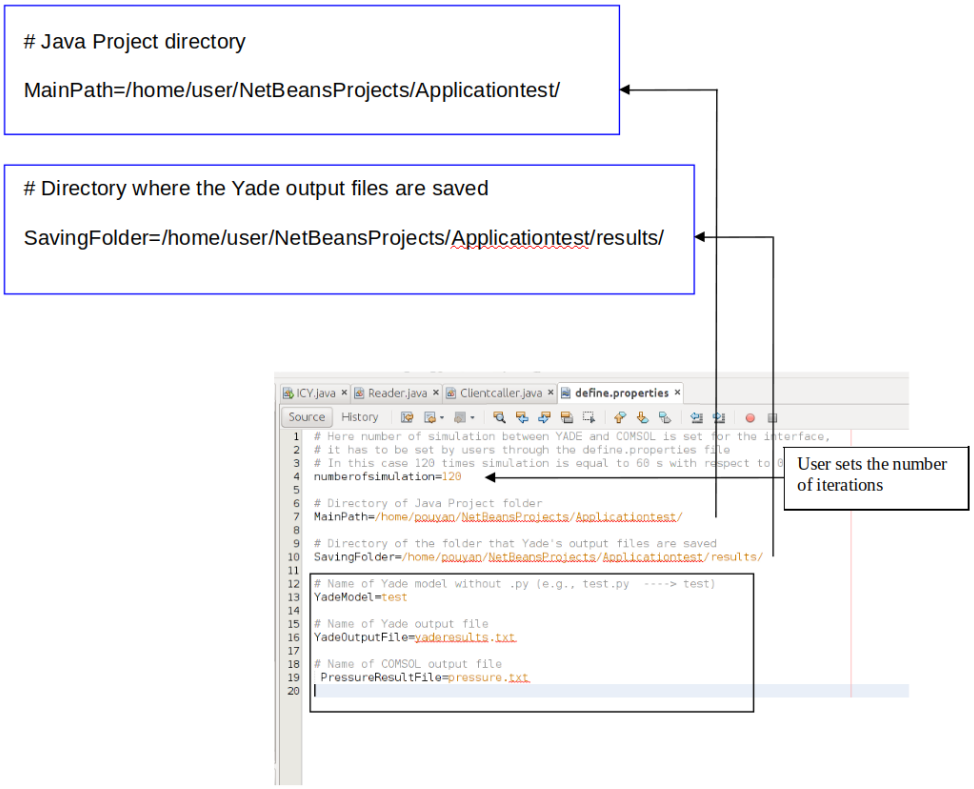

Figure 7. Setting variables and directories in the property file.

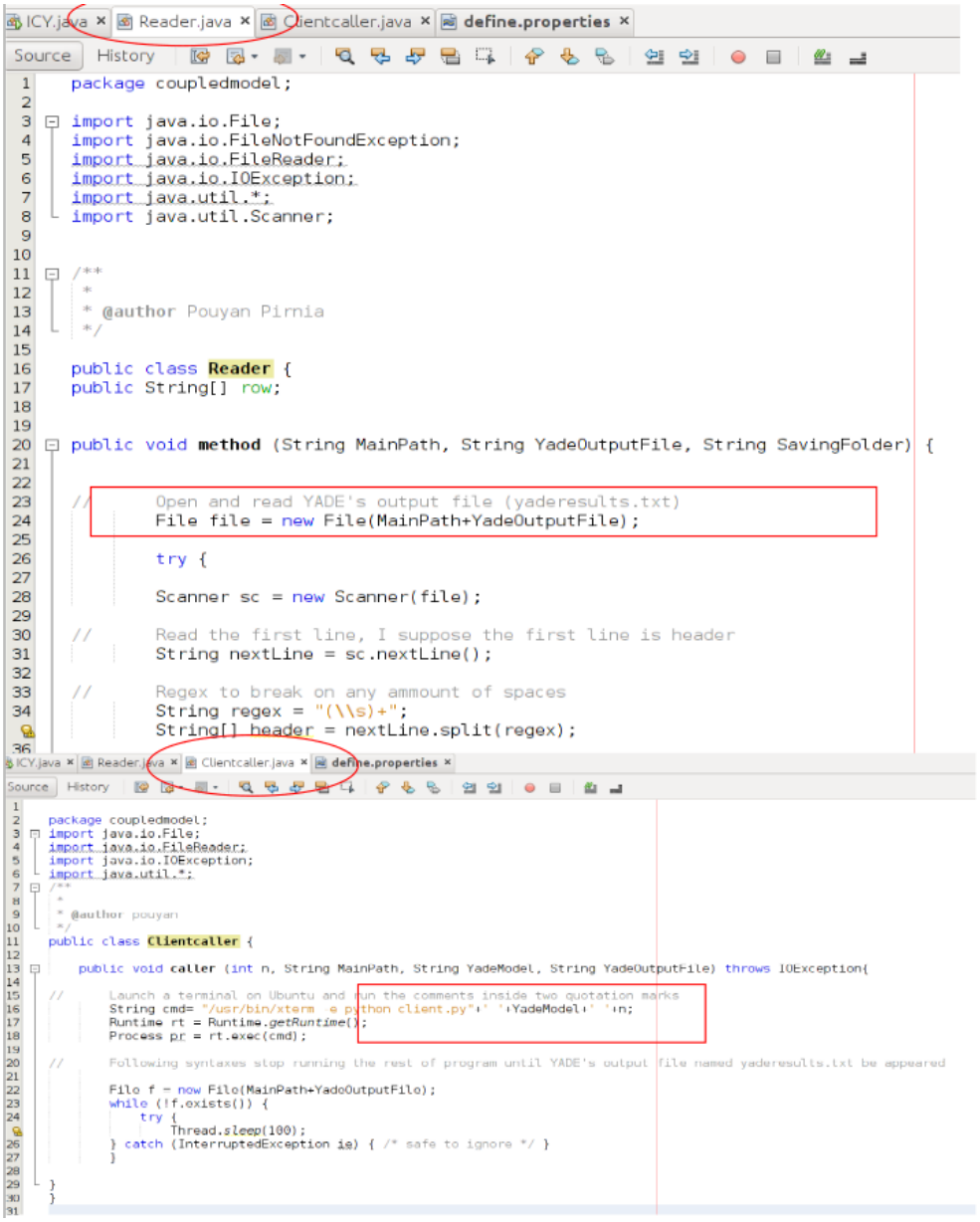

Figure 8. Names in Reader.java and clientcaller.java files taken from the property file.

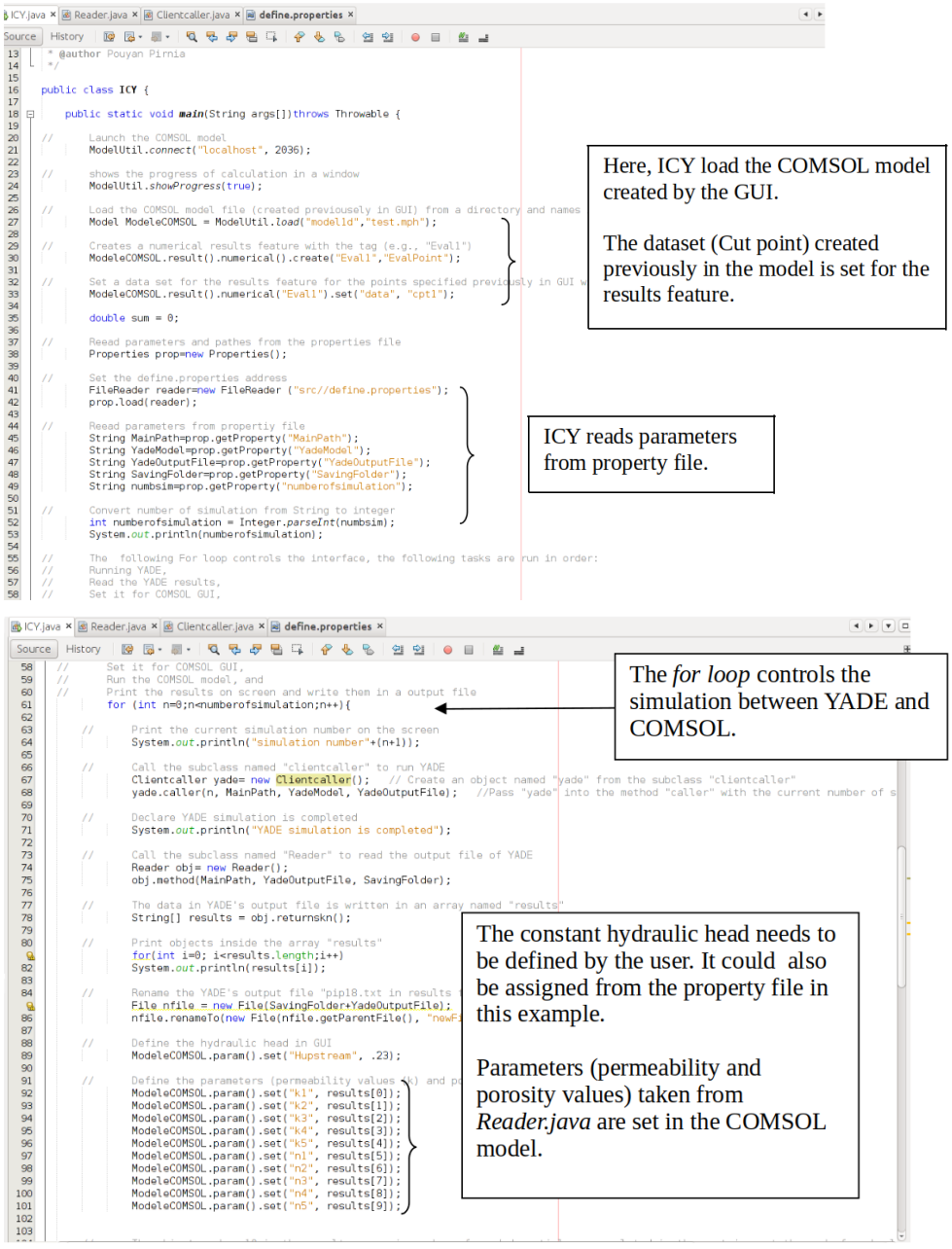

Figure 9. Main tasks in the ICY.java class script.

Figure 10. Components in the YADE script for the Application example.