A quick install guide to particleShear

A Python package for simulating spherical and crosslinked granular ensembles

Code by:
Thomas Braschler

With many thanks for help to:
Daniel Lyubenov
Fabien Bonini

Guide prepared by F. Bonini and J. Brefie-Guth

I - System requirements

Python: Python (version 3.7.0 or higher)

Minimum system requirements

•Processors: Any modern processor capable of running Python 3. For instance: Intel Atom® processor or Intel® Core™ i3 processor

• Disk space for Installation:
10 MB for the package per se
typically 100MB to 1GB for Python and an integrated development environment

•Disk space for simulation output:
Per run: Minimum 5kB (minimal text output). Maximum 50MB (images and full stress tensor ASCII file)

• Operating systems: Tested on Windows (7 and 10), macOS (10.11.1), and Linux (CentOS 6.1)

Software dependencies

Python package Pillow (version 5.3.0)
For graphical display of results (default): Python with Tkinter.
For image saving in jpeg format: GhostScript (version 9.19)

II - Installation guide

Typical install time: 10 minutes

1) Download Python v3.5 or higher,
Preferentially 32 bit on Windows (https://www.python.org/downloads/)

2) Install the software according to the instructions shown

3) For window users, run “cmd” **as administrator**; for MacOSX and Linux users, launch the Terminal application (Shell)

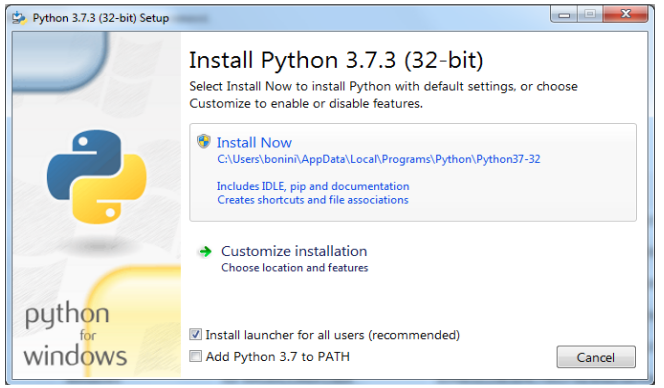
4) Install the *particleShear* module:

```
python –m pip install particleShear
```

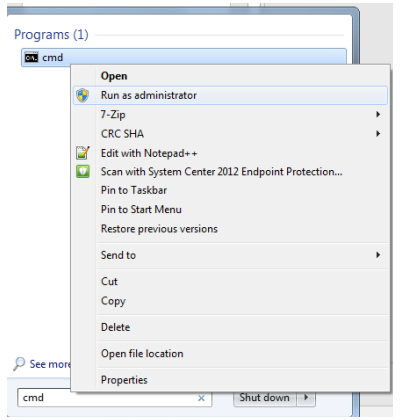
Note: In some cases, *Pillow* module might not install automatically. To install manually the *Pillow* module, execute the following command (for MacOSX and Linux users: “sudo” may be necessary):

```
python –m pip install Pillow
```

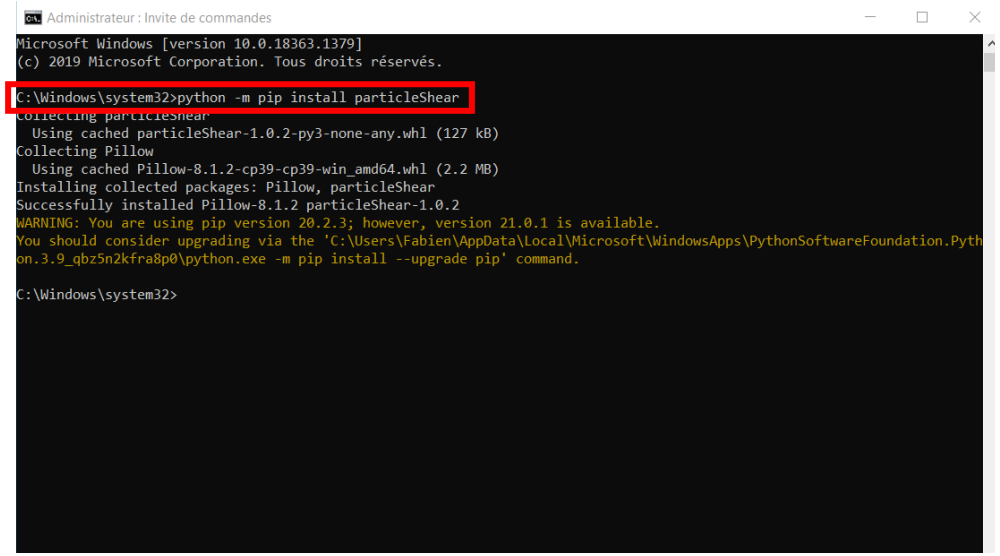
2



3



4



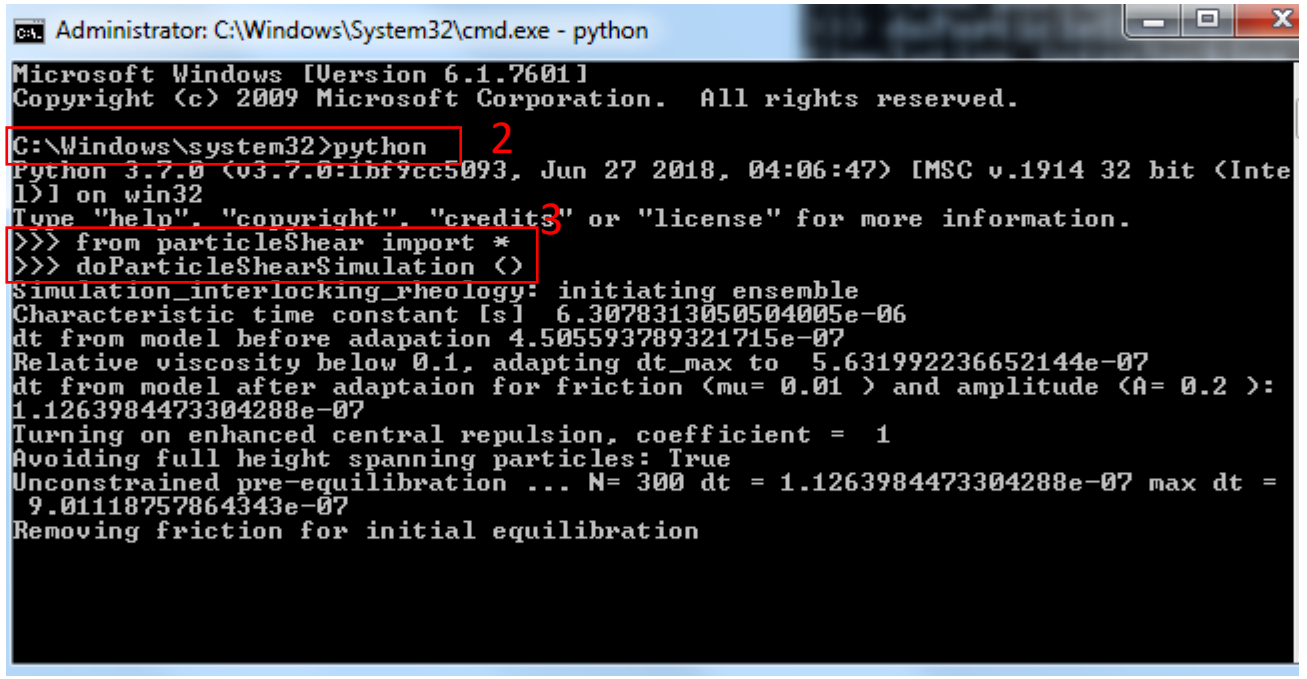
III - Demo

1) Run “cmd”

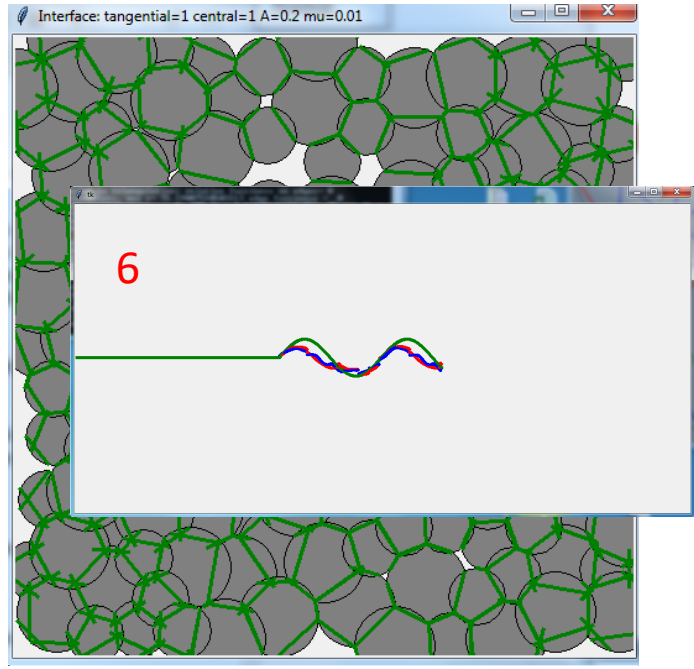
2) Execute Python:
`python`

3) Import the *particleShear* package
`from particleShear import *`

4) Execute the simulation using basic parameters:
`doParticleShearSimulation(N=20)`



Here, N=20 spheres are used to enhance speed of execution (20-30min). Full simulation options can be found in Supplementary 2, Table S2-1. For the default value of N=150, the simulation may take several hours.



5) The graphical interface is displayed: first the particle assembly (5), later the output of digital rheology (6). The simulation can be interrupted at any time by closing the graphical window.
Textual output during the simulation includes simulation details. After completion, (automatic closure of windows 5 and 6), nominal G’ and G” should be displayed, and for control, approximately matching G’ and G” from external force without inertial correction.

IV - Self-test

1) Run “cmd” (or Terminal/shell for MacOSX / Linux)

2) Execute the self-test using the following code:

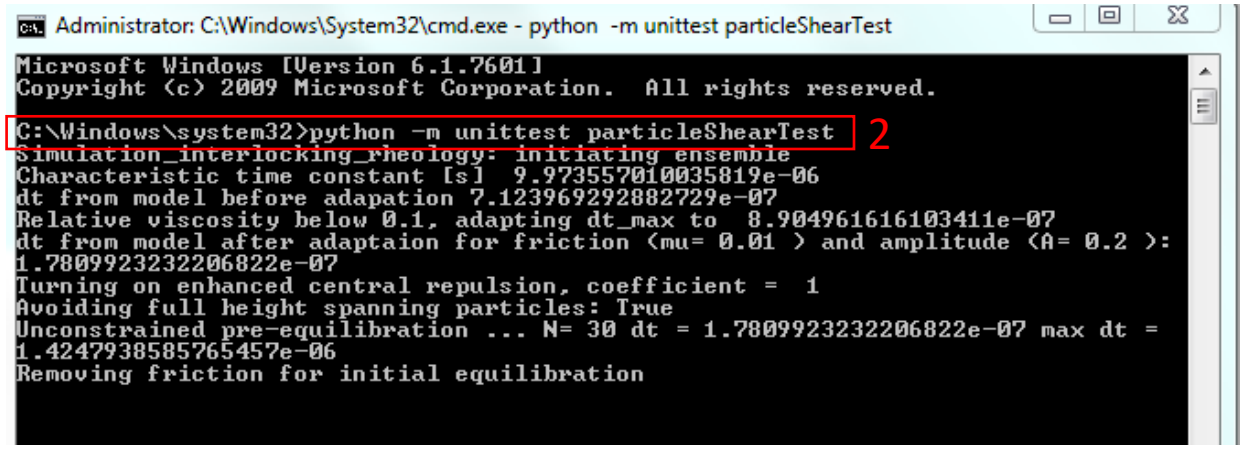
```
python -m unittest particleShearTest
```

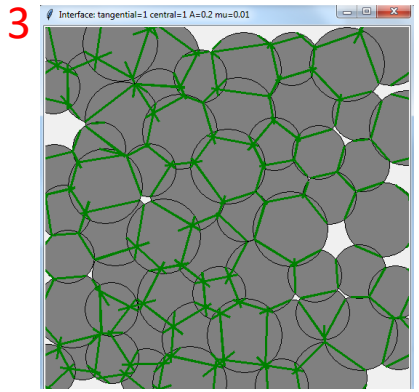
3) Different tests are executed by the software. This might take some minutes. Graphical interface appears in additional pop-up windows.

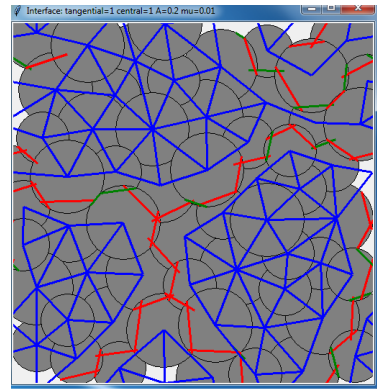
Expected output

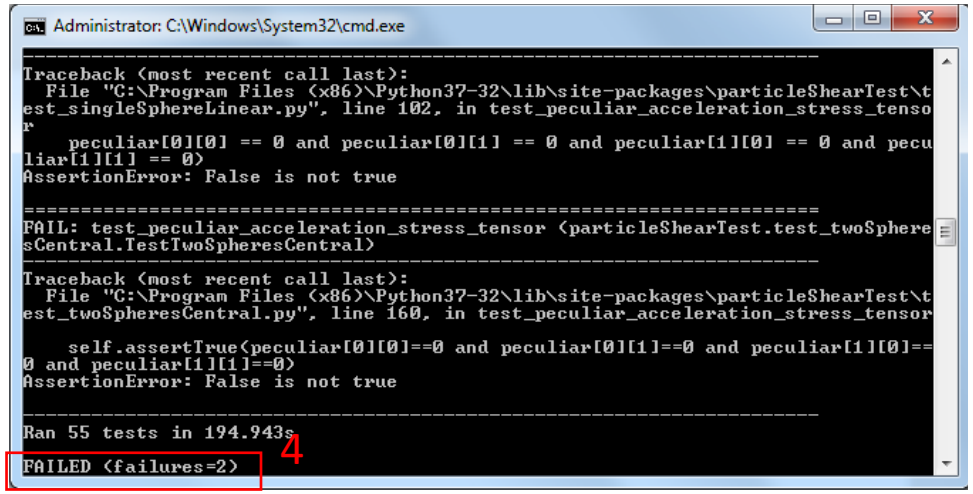
4) Current status after self-test display 2 failures out of 55 tests. To note, these 2 failures result from the implementation of tests with a published formula that does always fully conserve angular momentum. This formula is not used by default in the regular simulations.

<u>Run time</u>: 194.943 seconds









V – Special cases

No Graphical display

Some systems, particularly Linux servers, do not have graphical display capacity. Running the simulation with default options will cause errors on these systems. In that case, it is mandatory to turn off graphical display. This can be done by invoking the simulation without graphical display (plus of course other options if desired):

```
doParticleShearSimulation(doDrawing=False)
```

Since images of the simulation are generated as snapshots of the actual graphical canvas, it is not possible to save images of the simulation on such systems (in this case, the argument `saveOutputImages` will be ignored even if it is set to `True`).

No Ghostscript installed

The simulation can still be run, and also graphically displayed on systems without Ghostscript or with Ghostscript in a location where Python cannot find it (incomplete system Path configuration). In this case, however, trying to save images by passing `saveOutputImages=True` to `doParticleShearSimulation` will result in an error. Installing Ghostscript and setting up the system path in such a way that the Python installation can find it can be rather tricky, so we recommend `saveOutputImages=True` only for advance use.

VI – Trouble shooting

When trying to install Python and the particleShear simulation on different systems, we've had various smaller and bigger problems. As their solution may help other users, we report them here.

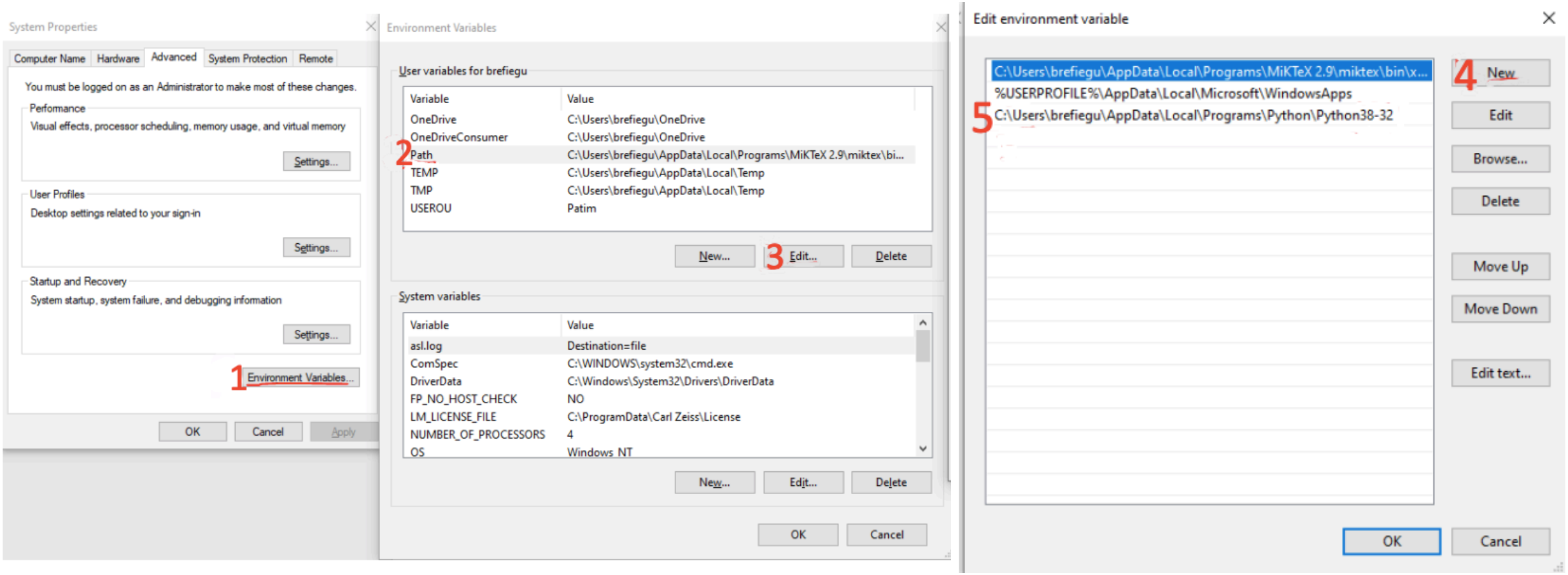
1) Headless environments: These are typically Server environments (in our case, the Baobab cluster of the University of Geneva, and the CodeOcean capsule). There are no physical graphical displays (computer screens) in these environments, and graphical output is difficult to configure. Section V in this guide indicates how to run the simulations without graphical display. The main readme file of the CodeOcean capsule provides further (more sophisticated) ways of configuring file saving or taking of virtual screenshots in such headless environments.

2) Windows 10: Somehow the installation of Python didn't work out of the box for us. The following slide provides trouble shooting instructions on Windows 10.

Windows 10: Problems with Python path configuration

(Particular issue: When you want to open Python on the command line prompt, but instead the Microsoft Store opens) :

- Search for the path of the python application in the folder Explorer and copy it. This can be in many places, it helps to look at the properties of a desktop or startup menu link.
- Run “env” from the Start menu and open “Edit the system environment variables”. Click the button “Environment Variables” (1). In the table “User variables”, find “Path” (2) and click “Edit” (3).
- Paste the path of Python copy on a new line (4, 5). Save everything by hitting OK buttons as you go.



For Windows 10 Problems with Python path configuration

(Particular issue: When you want to open Python on the command line prompt, but instead the Microsoft Store opens even after having carried out the instructions on the previous slide) :

- Run “Settings”, open “Apps”, “Apps & Features“ and click on “App execution aliases”.
Turn off “Programme installation application…” of Python (6). If there are several Pythons, just turn them all off.