This document aims to explain how the StressStrainControlBoundary in the MercuryDPM (Trunk) works. This is a more detailed documentation which explains different deformation modes in a representative elementary volume (REV).

The StressStrainControlBoundary allows user to define different deformation mode, based on the user inputs: Target Stress Tensor, Target Strainrate Tensor and GainFactor Tensor. Note that each tensor has 9 components, we only use 4 in the current implementation: xx, yy, zz and xy. Below we are going to give a brief introduction to the background and possible deformation modes.

1 How do we define stress, strain and strain-rate

It is worth to introduce our terminology regarding strain and stress, before we go into the each deformation mode we make from this boundary.

1.1 Strain

For any deformation, the isotropic part of the infinitesimal strain tensor ϵ_{v} (in contrast to the true strain ε_{v}) is defined as:

$$\epsilon_{\mathrm{v}} = \dot{\epsilon}_{\mathrm{v}}\mathrm{dt} = \frac{\epsilon_{xx}+\epsilon_{yy}+\epsilon_{zz}}{3} = \frac{1}{3}\mathrm{tr}(\mathbf{E}) = \frac{1}{3}\mathrm{tr}(\dot{\mathbf{E}})\mathrm{dt}, \tag{1}$$

where $\epsilon_{\alpha\alpha}= \dot{\epsilon}_{\alpha\alpha}\mathrm{dt}$ with $\alpha\alpha = xx$, yy and zz as the diagonal elements of the strain tensor $\mathbf{E}$ in the Cartesian x, y, z reference system. The integral of $3\epsilon_{\mathrm{v}}$ denoted by $\varepsilon_{\mathrm{v}} = 3\int_{V_0}^{V}\epsilon_{\mathrm{v}}$, is the true or logarithmic strain, i.e., the volume change of the system, relative to the initial reference volume, V_0 [1].

Several definitions are available in literature [6] to define the deviatoric magnitude of the strain. Here, we use the objective definition of the deviatoric strain in terms of its eigenvalues ${\epsilon_{\mathrm{d}}}^{(1)}$, ${\epsilon_{\mathrm{d}}}^{(2)}$ and ${\epsilon_{\mathrm{d}}}^{(3)}$ which is independent of the sign convention.

The deviatoric strain is defined as:

$$\epsilon_{\mathrm{dev}} = \sqrt{\frac{\left(\epsilon_d^{(1)}-\epsilon_d^{(2)}\right)^2+\left(\epsilon_d^{(2)}-\epsilon_d^{(3)}\right)^2+\left(\epsilon_d^{(3)}-\epsilon_d^{(1)}\right)^2}{2}}, \tag{2}$$

where $\epsilon_{\rm dev} \geq 0$ is the magnitude of the deviatoric strain.

1.2 Stress

From the simulations, one can determine the stress tensor (compressive stress is positive as convention) components:

$$\sigma_{\alpha\beta} = \frac{1}{V}\left(\sum_{p\in V} m^p v_\alpha^p v_\beta^p - \sum_{c\in V} f_\alpha^c l_\beta^c\right), \tag{3}$$

with particle p, mass m^p, velocity v^p, contact c, force f^c and branch vector l^c, while Greek letters represent components x, y, and z [4, 2]. The first sum is the kinetic energy density tensor while the second involves the contact-force dyadic product with the branch vector. Averaging, smoothing or coarse graining [8, 7] in the vicinity of the averaging volume, V, weighted according to the vicinity is not applied in this study, since averages are taken over the total volume. The isotropic stress is denoted as hydrostatic pressure:

$$p = \sigma_{\rm v} = \frac{1}{3}{\rm tr}(\boldsymbol{\sigma}) \tag{4}$$

As already mentioned, we will focus on the eigenvalues of the deviatoric stress tensor $\lambda_i^s = \sigma_i^{\rm D} = \sigma_i - p$, as defined in section **??**, with the principal directions being the same for $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}^D$.

The (scalar) deviatoric stress for our $3D$ simulations is:

$$\sigma_{\rm dev} = \sqrt{\frac{(\lambda_1^s-\lambda_2^s)^2 + (\lambda_1^s-\lambda_3^s)^2 + (\lambda_2^s-\lambda_3^s)^2}{2}}. \tag{5}$$

The deviatoric stress ratio, $s_{\rm dev} = \sigma_{\rm dev}/p$, quantifies the "stress anisotropy" - where $\sigma_{\rm dev} = \sqrt{3J_2^\sigma}$, with J_2^σ the second invariant of the deviatoric stress tensor. The third stress invariant $J_3^\sigma = \lambda_1^s\lambda_2^s\lambda_3^s = \lambda_1^s\lambda_2^s(-\lambda_1^s - \lambda_2^s) = {\lambda_1^s}^3(-{\Lambda_1}^\sigma - ({\Lambda_1}^\sigma)^2)$ can be replaced by the shape factor $\Lambda^\sigma := \lambda_2^s/\lambda_1^s$, which switches from +1 at maximum uniaxial loading to -1/2 after some unloading as will be shown below.

Note that the wall motion is strain controlled and the infinitesimal strain corresponds to the external applied strain. Hence the eigenvalues for the strain tensor are in the Cartesian coordinate system (thus no transformation is needed). For the purely isotropic strain, $\boldsymbol{\epsilon}^{\rm ISO} = \epsilon_{\rm v}\boldsymbol{I}$, with $\epsilon_{\rm dev} = 0$, which is direction independent by definition. The corresponding shape factor for the deviatoric strain $\Lambda^{(-\epsilon)}$, is represented as the ratio $\Lambda^{(-\epsilon)} := \epsilon_d^{(2)}/\epsilon_d^{(1)}$.

2 Introduction to StressStrainControlBoundary

The StressStrainControlBoundary is developed based on the recent user codes which are focusing on a stress controlled simple shear, see [5] for more details. In this boundary, we combined the stress control and strainrate control in one setup, according to the user inputs, the program will know how to adapt the boundary as well as activating the stress servo control or not. The strainrate tensor is defined as:

$$\dot{\mathbf{E}} = \begin{pmatrix} a & d & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} + \Delta\dot{\mathbf{E}},$$

The change in strain rate tensor $\Delta\dot{\mathbf{E}}$ is defined as,

$$\Delta\dot{\mathbf{E}} = \Delta\dot{\mathbf{E}} + \mathbf{g}\Delta\boldsymbol{\sigma}$$

where the stress difference $\Delta\boldsymbol{\sigma}$ is,

$$\Delta\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} - \sigma_{xx}^{goal} & \sigma_{xy} - \sigma_{xy}^{goal} & 0 \\ 0 & \sigma_{yy} - \sigma_{yy}^{goal} & 0 \\ 0 & 0 & \sigma_{zz} - \sigma_{zz}^{goal} \end{pmatrix}$$

and our gain factor $\mathbf{g}$ is,

$$\mathbf{g} = \begin{pmatrix} g_{xx} & g_{xy} & 0 \\ 0 & g_{yy} & 0 \\ 0 & 0 & g_{zz} \end{pmatrix}$$

the gain factor is a user defined small pre-factor to change the strain-rate tensor according to the stress difference, if the correction is too larger, or too small, the stress (servo) control might not work properly and the target stress will never be reached. Therefore, use has to determine this value first whenever start a new simulations with new material parameters and/or new strain-rate inputs. Normally we start with a very small gain factor, e.g. 0.0001, then depending on how the system is evolving and the servo-control is responding, we have to tune this parameter.

Below is a short summary of the 12 user defined parameters in the 3 above mentioned tensors:

- a or σ_{xx}^{goal}
- b or σ_{yy}^{goal}
- c or σ_{zz}^{goal}
- d or σ_{xy}^{goal}
- g_{xx}
- g_{yy}
- g_{zz}
- g_{xy}

Note that you could not set up both the strainrate and target stress at the same time, e.g. you can only set a non-zero value for "a" or for σ_{xx}^{goal}, otherwise the code will complain. The reason is that if you set a target stress, the corresponding component in the strain-rate tensor will be changed to adapt for the stress difference, therefore it is not possible to keep the strain-rate component also as a constant value at the same time.

In addition, there is also a boolean parameter that user has to set:

- isStrainRateControlled = true

By default it is set to true, which means the particles are following affined movement every timestep based on the strainrate tensor, in this way, the whole particle bed deforms more homogeneously. If user sets it to "false", the particles will only follow the boundary movement, so in the case of compression, the boundary is pushing the closest layer of particles like a piston.

In the following sections, we will explain some examples of different deformation modes that could be achieved using this boundary.

3 Isotropic compression of a cubiod REV

The first deformation we would like to introduce is the isotropic compression, it is a very simple way of deforming your sample and could let you achieve a more homogenous packing that random generation could not achieve.

We first generate the sample randomly by inserting particles in a REV, then isotropically compress it with a constant strainrate tensor. Here, we choose the homogeneous isotropic strain-rate controlled deformation driving mode because it will produce more homogeneous systems, leads to more clean results and eliminate the wall effect, details on different driving mode can also be found in [3]. In this driving method, we apply a homogeneous strain rate to all particles in the ensemble and to the walls in each time-step, such that each particle experiences an affine simultaneous displacement according to the diagonal strain rate tensor:

$$\dot{\mathbf{E}} = \dot{\epsilon}_{\mathrm{v}} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

where $\dot{\epsilon}_{\mathrm{v}}$ (> 0) is the rate amplitude applied until a certain time.

In the example code shown in the tutorial, user could set $a = b = c = -\dot{\epsilon}_{\mathrm{v}}$, while leave $d = 0$. For the target stress tensor and gainFactor tensor, user could leave it as default (all set to zero) or like in the example, specifically set them to zero. Because the stress control is deactivated in this case, therefore it is not needed to set these two tensors.

If user wants to achieve a certain pressure/stress in the sample, instead of setting the strainrate tensor, one could set the corresponding target stress tensor with the related gain factors. Then the system will use the servo-control to achieve the target stress. The target stress tensor is defined as:

$$\boldsymbol{\sigma}^{\boldsymbol{goal}} = \begin{pmatrix} \sigma_{xx}^{goal} & 0 & 0 \\ 0 & \sigma_{yy}^{goal} & 0 \\ 0 & 0 & \sigma_{zz}^{goal} \end{pmatrix}$$

A simple variation of the isotropic compression is the uni-axial compression, in which the control parameter is only set in a single direction.

Or a more complex variation: Triaxial test. The user has to define the control parameters in each direction independently. This will not be explained in more details as it is very straightforward if user understands the isotropic compression mode.

If user wants to have a sample that is relaxed, it could simply be achieved by setting all the control parameters to zero after certain time and run for longer time.

4 Volume conserving pure shear in a cubiod REV

Similarly, the user could also define a pure shear deformation mode, here we define a volume conserving pure shear in x-y direction, which has dilation in x direction and compression in y direction. So the strain rate tensor looks like this:

$$\dot{\mathbf{E}} = \dot{\epsilon}_{\mathrm{D2}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where $\dot{\epsilon}_{\mathrm{D2}}$ is the strain-rate (compression > 0) amplitude applied to the periodic wall with normal in y-direction. We use the nomenclature D2 since two periodic walls are moving while the third one is stationary [2]. The chosen deviatoric deformation path is on the one hand similar to the pure-shear situation, and on the other hand allows for simulation of the biaxial experiment (with two walls static, while four walls are moving), in contrast to the more difficult isotropic compression, where all the six periodic walls are moving. Different types of volume conserving deviatoric deformations can be applied to shear the system, but very similar behavior has been observed [2].

In the example code shown in the tutorial, user could set $a = -b = \dot{\epsilon}_{\mathrm{D2}}$, while leave $c = d = 0$. For the target stress tensor and gainFactor tensor, user could leave it as default (all set to zero) or like in the previous example, specifically set them to zero. Because the stress control is deactivated in this case, therefore it is not needed to set these two tensors.

The user could also define the pure shear deformation mode using the stress control, however, if the sample is highly inhomogeneous, the response to the stress in each direction is also different, this will result in a changing volume simulation, which violates our goal. Besides, it might also cause a shear induced inhomogeneity, then the actual deformation achieved might not be named as pure shear anymore.

5 Volume conserving simple shear in a cuboid REV

Another commonly used deformation mode is volume conserving simple shear, normally we take the samples that relaxed after isotropic compression (at certain volume fraction/bulk density or a given pressure) Then initiate the simple shear by defining the strainrate tensor. The samples are in cuboid volume with Lees-Edwards periodic boundary in x-y-direction and normal periodic boundaries in z-direction. The particles are sheared along x-direction with a constant velocity V_x and the stress is kept constant along y direction as σ_{yy}.

In analogy to the strain-rate controlled isotropic compression, the shear movement of particles is achieved by moving the particles in the x-direction according to a prescribed constant shear strain-rate tensor:

$$\dot{\mathbf{E}} = \dot{\gamma} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

Where, the shear rate $\dot{\gamma} = \partial V_x / \partial y$, and V_x is the shear velocity in x-direction.

In the example code shown in the tutorial, user could set $d = \dot{\gamma}$. while leave $a = b = c = 0$. For the target stress tensor and gainFactor tensor, user could leave it as default (all set to zero) or like in the previous example, specifically set them to zero. Because the stress control is deactivated to attain the constant volume in this case, therefore it is not needed to set these two tensors.

6 Stress controlled simple shear in a cuboid REV

Additionally, one could also achieve a simple shear combined with a stress controlled deformation mode. This actually presents more realistic cases in our real material testing, e.g. sand or/and powders sheared in different shear cells. In which the sample normally has a constant stress perpendicular to the direction of shear. In the code presented in the tutorial, we use the same shear in x direction as the previous case and initiate the stress control in y direction.

The strainrate tensor is the same as the volume conserving simple shear:

$$\dot{\mathbf{E}} = \begin{pmatrix} 0 & \dot{\gamma} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

Since we want to keep our normal stress constant along y direction during shearing, we have to also introduce the target stress in y direction σ^{goal}:

$$\boldsymbol{\sigma^{goal}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sigma_{yy}^{goal} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

with also the gain factor tensor $\mathbf{g}$:

$$\mathbf{g} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & g_{yy} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

In the example code shown in the tutorial, user needs to set $d = \dot{\gamma}$. while leave $a = b = c = 0$. Then σ_{yy}^{goal} = your target stress value, $\sigma_{xx}^{goal} = \sigma_{zz}^{goal}\sigma_{xy}^{goal} = 0$; $g_{yy} = 0.0001$(this value can be tuned), $g_{xx} = g_{zz} = g_{xy} = 0$.

It is also worth to mention that as we have a constant shear deformation rate defined by shear rate $\dot{\gamma}$ but varying deformation rate $\dot{\epsilon}_{yy}$ in y direction given by the stress control, the two deformations might create the resonance effect, which will lead to a failure of the stress control. The general rule of thumb is to avoid the two deformation rates getting close to each other. However, this problem really depends on the material parameters and the shear rate that are applied, which we will not be addressed further here.

References

[1] F. Göncü, O. Durán, and S. Luding. Constitutive relations for the isotropic deformation of frictionless packings of polydisperse spheres. *C. R. Mécanique*, 338(10-11):570–586, 2010.

[2] Olukayode I. Imole, Nishant Kumar, Vanessa Magnanimo, and Stefan Luding. Hydrostatic and Shear Behavior of Frictionless Granular Assemblies under Different Deformation Conditions. *Kona Powder Part. J.*, 30(30):84–108, 2013.

[3] Olukayode I. Imole, Mateusz Wojtkowski, Vanessa Magnanimo, and Stefan Luding. Micro-macro correlations and anisotropy in granular assemblies under uniaxial loading and unloading. *Phys. Rev. E*, 89(4):42210, 2014.

[4] Stefan Luding. Anisotropy in cohesive, frictional granular media. *J. Phys. Condens. Matter*, 17(24), 2005.

[5] Hao Shi, Sudeshna Roy, Thomas Weinhart, Vanessa Magnanimo, and Stefan Luding. Steady state rheology of homogeneous and inhomogeneous cohesive granular materials. *Granul. Matter.*, 22(1):14, 2020.

[6] C. Thornton and L. Zhang. A numerical examination of shear banding and simple shear non-coaxial flow rules. *Philosophical Magazine*, 86(21-22):3425–3452, 2006.

[7] T. Weinhart, R. Hartkamp, A. R. Thornton, and S. Luding. Coarse-grained local and objective continuum description of three-dimensional granular flows down an inclined surface. *Physics of Fluids*, 25(7):070605, 2013.

[8] T. Weinhart, A. R. Thornton, S. Luding, and O. Bokhove. From discrete particles to continuum fields near a boundary. *Granul. Matter.*, 14(2):289–294, 2012.