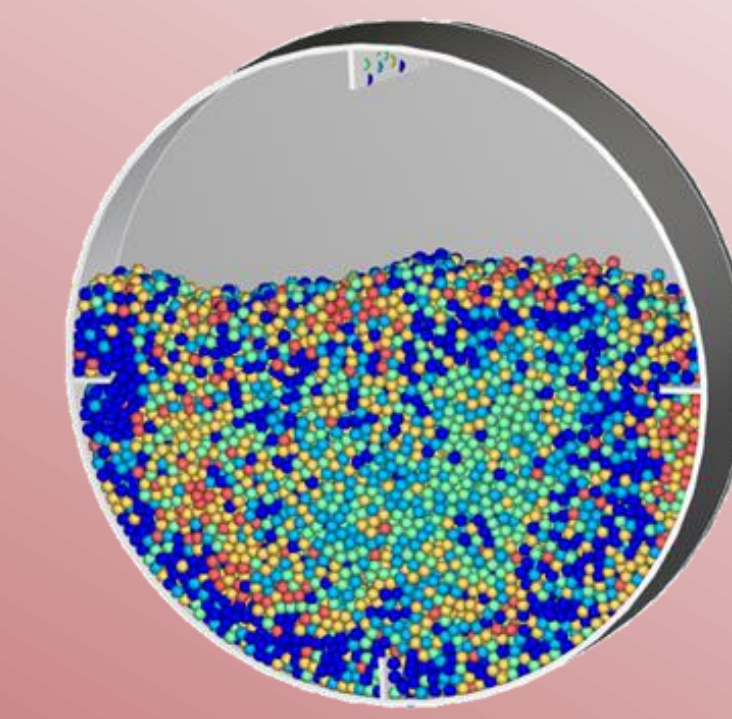
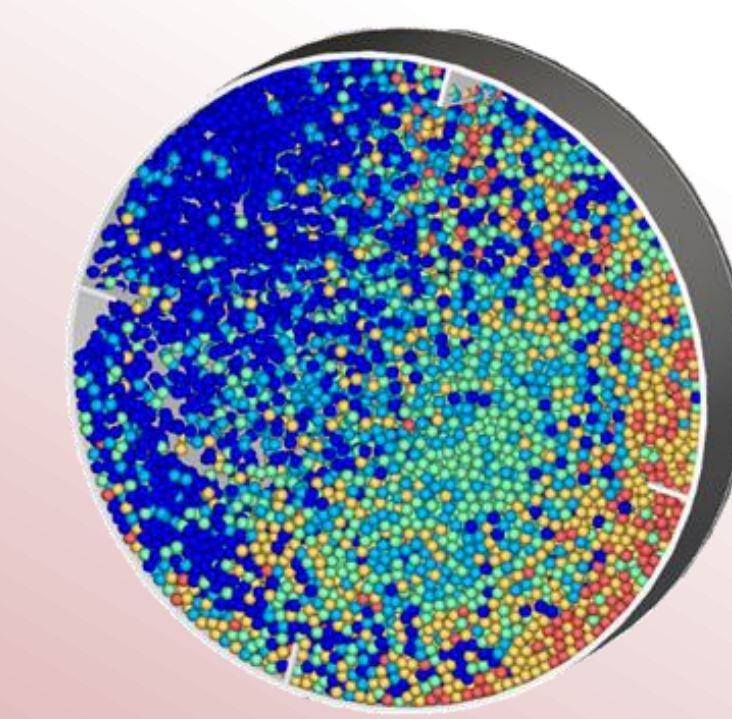
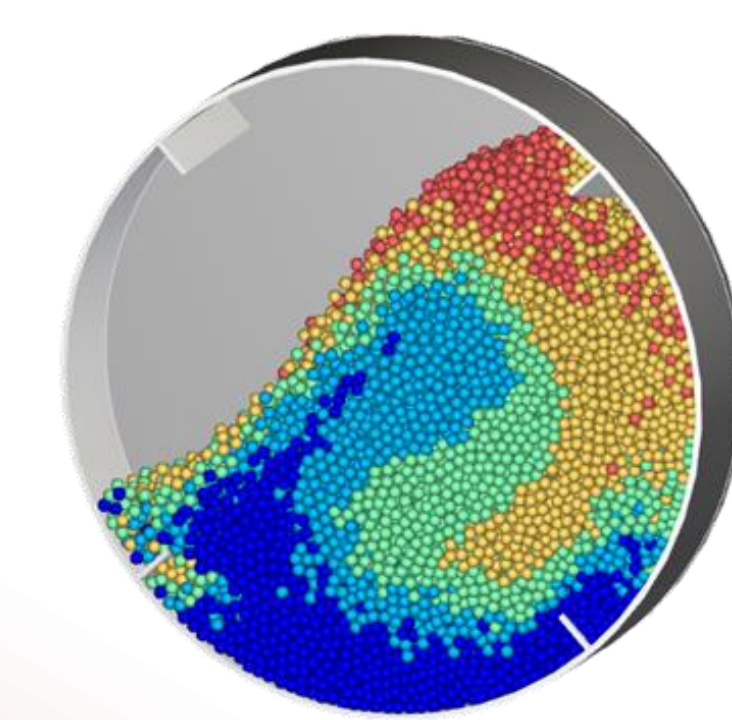
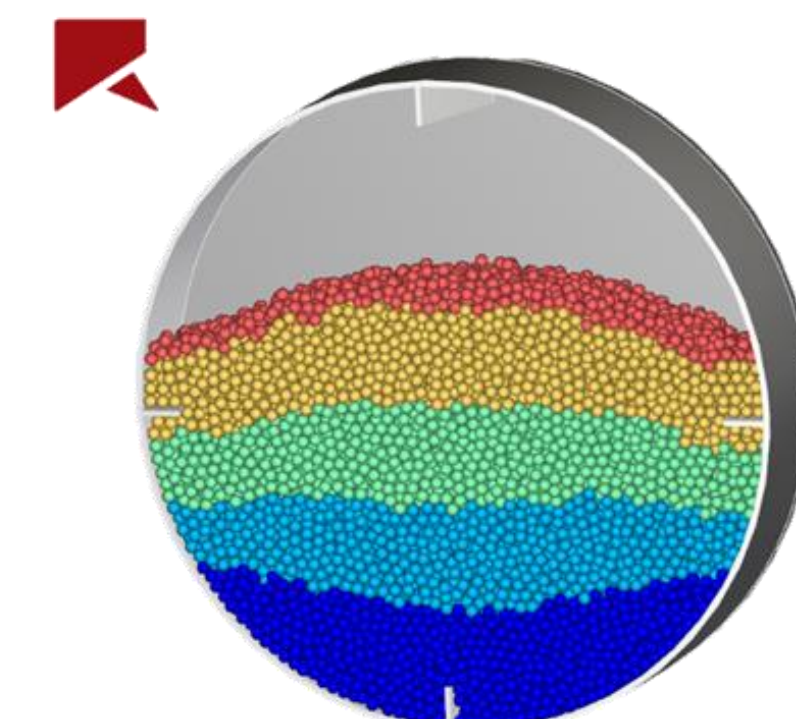
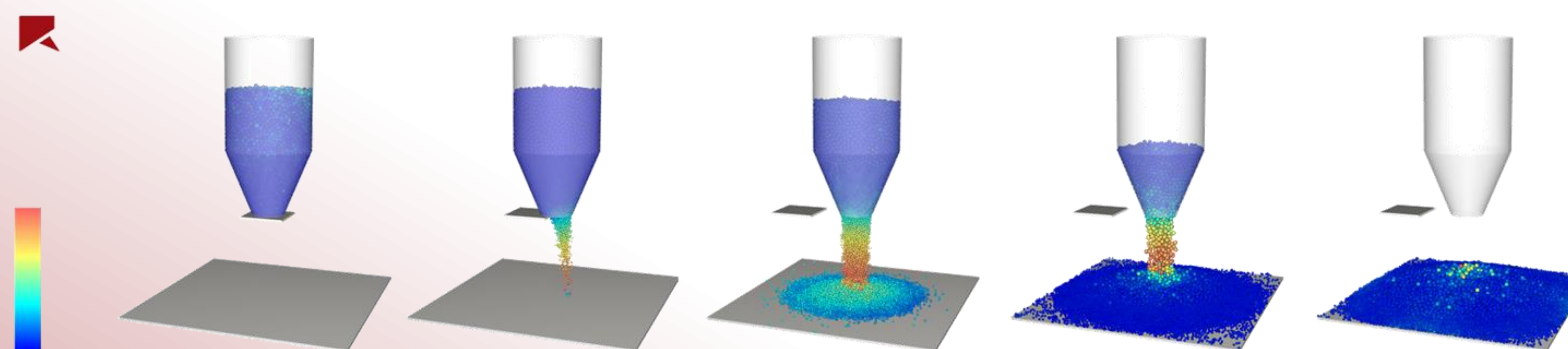


ROCKY



Start with this Particle Type	To create this shape	Start with this Particle Type	To create this shape	Start with this Particle Type	To create this shape
Rounded Polyhedron	Barley, Pea, Rice, Rock, Potato, Orange, Raisin, Coal	Rounded Cylinder	Rod, Capsule, Sharp Rock, Wafer, Brick, Firewood, Newspaper, Package	Rounded Polyhedron	Chocolate, Wood Chip, Poker Chip, Corn, Pill, Navy Bean, Sheet Metal
Faceted		Briquette			

Rocky 4.2 (Windows 64-bit) - Rocky 4.2 (Windows 64-bit)

File Edit Window View Options Tools Help

Time: 0:00 s

Particles Details -<01>

Particle <01>

Motion Preview -<01>

Expressions/Variables

Name	Value
PPStaticFriction	0.21
PPDynamicFriction	0.75
PPAffriction	0.8

Simulation Log

== 01/24/19 08:50:09 ==

Simulation done!

```

def self_init(self):
    #size to calculate the repose angle
    self.analysis_time_step = 1e-05
    self.analysis_time_step = 1e-05

    #angle to create the cylinder slice
    self.cylinder_angle = 30

    #number of the particle size to calculate the bins size
    self.nbins_size = 10

def go_to_time_step(self):
    #go to the time to calculate the repose angle
    self.analysis_time_step = 1e-05
    self.analysis_time_step = 1e-05

def define_cylinder_process(self):
    collection = project.defineProcessCollection()
    #get the particles equivalent diameters
    diameters = particles.diameterFunction("Particle Equivalent Diameter").dataarray()

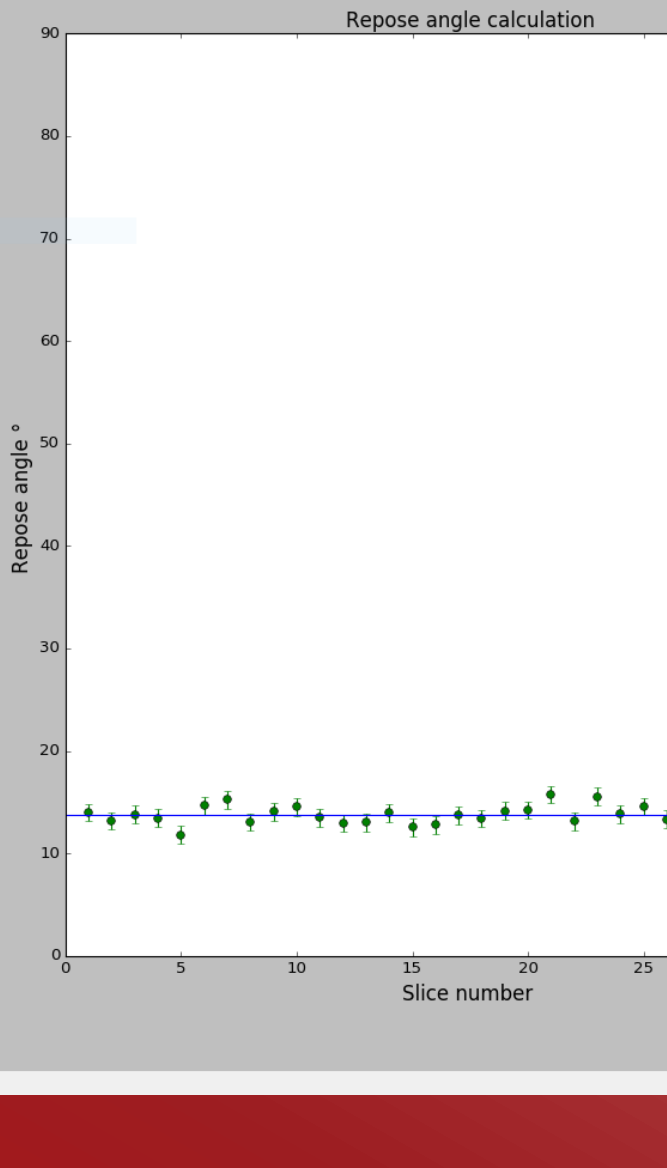
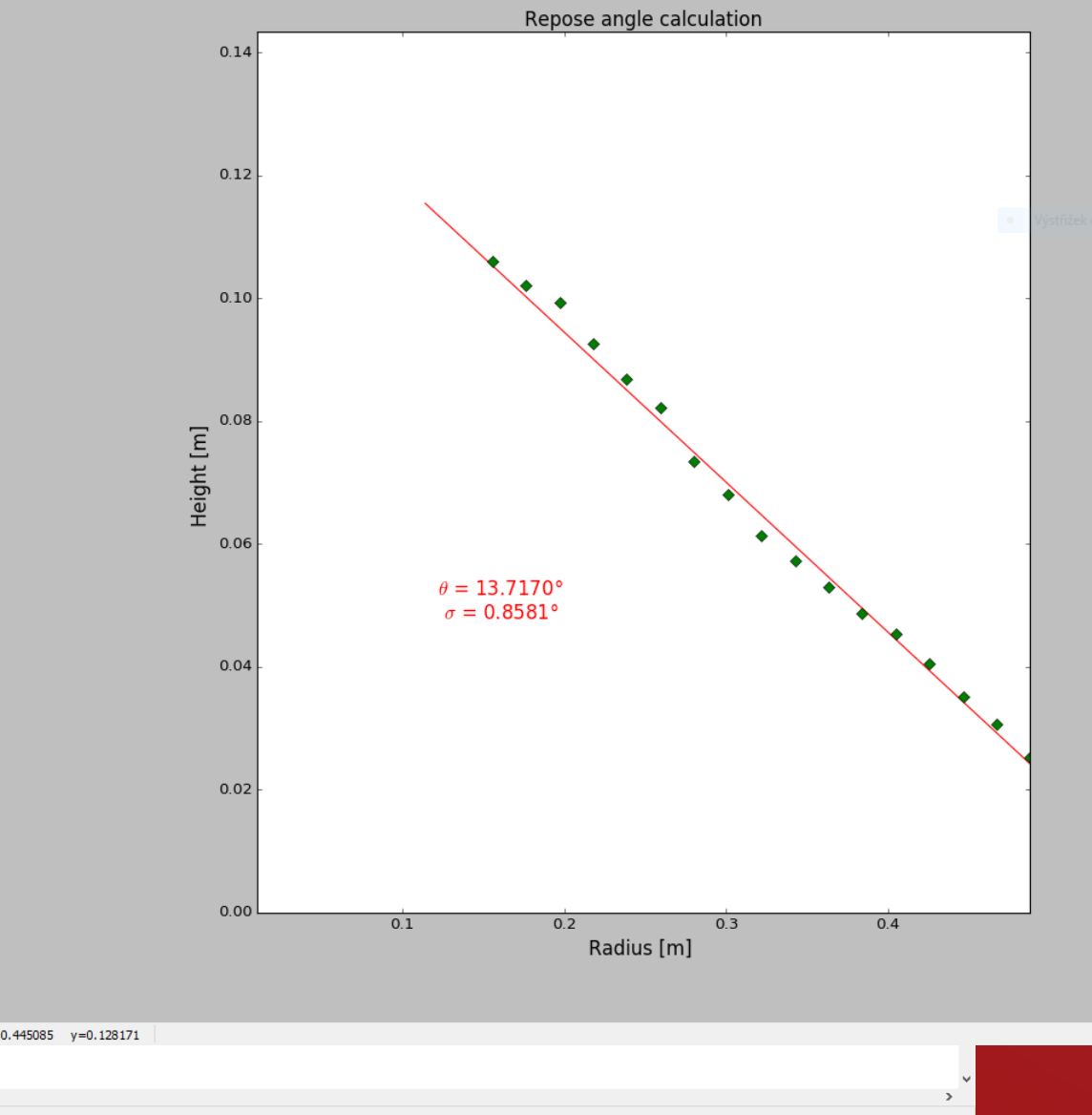
    #get the limits of particles positions
    particles_limits_min = particles.diameterFunction("Particle Equivalent Diameter").dataarray()
    particles_limits_max = particles.diameterFunction("Particle Equivalent Diameter").dataarray()

    #get the dimension of the particles positions at each direction
    particles_size = (particles_limits_max[0] - particles_limits_min[0])
    particles_size = (particles_limits_max[1] - particles_limits_min[1])
    particles_size = (particles_limits_max[2] - particles_limits_min[2])

    #calculate the cylinder radius based on the particles distribution
    cylinder_radius = float(max(particles_size/2.0, particles_size/2.0))
    cylinder_center = (0, particles_limits_min[1], particles_size/2.0, 0)
    cylinder_height = particles_size

    #creating cylinder
    cylinder = collection.createCylinderProcess(particles)
    #center cylinder
    cylinder.setPosition(0,0,0)
    #center cylinder
    cylinder.setPosition(cylinder_center[0], cylinder_center[1], cylinder_center[2])
    #set cylinder size
    cylinder.setSize(float(2*cylinder_radius), float(cylinder_height), float(2*cylinder_radius))
    #setting cylinder hole ratio
    cylinder.setInternalFactor(0)
    #set initial angle
    cylinder.setInitialAngle(0)
    #set final angle
    cylinder.setFinalAngle(self.cylinder_angle)
    end = collection.createUserStatistics(cylinder)

    #setting number of divisions
    
```



Rocky 4.2 (Windows 64-bit) - Rocky 4.2 (Windows 64-bit)

File Edit Window View Options Tools Help

Time: 0:00 s

Expressions/Variables

Name	Value
PPStaticFriction	0.21
PPDynamicFriction	0.75
PPAffriction	0.8

Simulation Log

== 01/24/19 08:50:09 ==

Simulation done!