On the Validation of a Differential Variational Inequality Approach for the Dynamics of Granular Material

D. Melanz, M. Tupy, B. Smith, K. Turner, and D. Negrut

December 21, 2009
Abstract

The validation of a DVI approach for the dynamics of granular material project focuses on comparing the experimental and simulation results of granular flow for two tests in the Chrono::Engine simulation environment. An aluminum rig was designed and fabricated to measure the mass flow rate of a given amount of granular material flowing due to gravity through a slit. The mass flow rate was initiated by using a Newport UMR8.25 translational stage and Newport LTA-HL precision linear actuator to open and close the slit steadily. Once the slit was open, the weight of the granular material was transmitted to the processor via a router connected to a Cooper LFS242 Tension/Compression Cell (Serial No. 286284) and graphed over time. A model of the flow meter was created in Chrono::Engine and the results were matched to experimental runs by changing the friction coefficient between particles. After the friction coefficient of the particles was determined to be 0.15, several experimental runs with differing slit sizes were run. These flow rates were compared to the mass versus time data that Chrono::Engine output for the corresponding slit size. Runs for gap sizes of 1.5 mm, 2 mm, 2.5 mm, and 3 mm were performed with 0.0624 N of granular material, which amounted to approximately 40,000 bodies. These gap sizes corresponded to an experimental flow rate of 1.41E-2 N/s, 2.59E-2 N/s, 4.00E-2 N/s, and 4.44E-2 N/s, and a simulated flow rate of 1.40E-2 N/s, 2.62E-2 N/s, 4.05E-2 N/s, and 4.48E-2 N/s, respectively. Based on this experiment, Chrono::Engine had less than a 2% error in calculating the flow rate of the granular material through a slit. In addition to comparing mass flow rates, the pile repose angle from the experimental runs was compared to the simulation results.
Contents
1. Introduction .................................................................................................................. 5
2. Equipment and Procedure ............................................................................................. 5
   2.1 Description of Design with Iterations ....................................................................... 5
      2.1.1 Initial Design ........................................................................................................ 5
      2.1.2 Second Version of Design .................................................................................. 6
      2.1.3 Final Design ........................................................................................................ 7
2.2 Description of Part Manufacturing ............................................................................... 8
      2.2.1 Angled Insert ....................................................................................................... 8
      2.2.2 Angled Insert Holder ............................................................................................ 9
      2.2.3 Static Slot ............................................................................................................ 10
      2.2.4 Base Plate ........................................................................................................... 11
      2.2.5 Assembly ............................................................................................................. 12
2.3 Description of Translational Stage .............................................................................. 13
2.4 Description of Actuator .............................................................................................. 13
2.5 Description of Load Cell ............................................................................................ 14
2.6 Schematic of Experimental Setup and Data Transmission .......................................... 14
3. Summary of Experimental Results ............................................................................. 14
   3.1 Results of 3 mm Gap .................................................................................................. 15
   3.2 Results of 2.5 mm Gap .............................................................................................. 15
   3.3 Results of 2 mm Gap .................................................................................................. 16
   3.4 Results of 1.5 mm Gap .............................................................................................. 16
4. Large Scale Multibody Dynamics on the GPU ............................................................... 17
   4.1 The Formulation of the Equations of Motion ............................................................ 17
   4.2 The Time Stepping Solver ......................................................................................... 19
   4.3 The GPU Formulation of the CCP Solver ................................................................. 21
5. A Setup of the Simulation in Chrono::Engine ............................................................... 24
6. Procedure Used to Select the Value of Mu .................................................................. 26
7. Comparison of the Experimental and Simulation Results that Validates Chrono::Engine ...................................................................................................................... 27
   7.1 Results of 3mm Gap .................................................................................................. 27
   7.2 Results of 2.5 mm Gap .............................................................................................. 27
   7.3 Results of 2 mm Gap .................................................................................................. 28
   7.4 Results of 1.5 mm Gap .............................................................................................. 29
7.5 Summary of Results ........................................................................................................ 30
8. Validation of Chrono::Engine in Regards to the Pile Angle ........................................... 31
9. Future Directions of Work .............................................................................................. 32
10. Project Conclusions ...................................................................................................... 33
11. Acknowledgements ....................................................................................................... 33
12. References ..................................................................................................................... 33
1. Introduction

This paper discusses the ongoing attempts by the Simulation-Based Engineering Laboratory (SBEL) of the University of Wisconsin-Madison to validate the Chrono::Engine simulation environment [18]. More specifically, this project focuses on confirming the results, in Chrono::Engine, of granular flow through a channel [16]. To accomplish this task, an aluminum rig was designed and fabricated to measure the mass flow rate of a given amount of granular material flowing through a slit due to gravity. Additionally, a model of the flow meter was created in Chrono::Engine and the results were matched to experimental runs by changing the friction coefficient between particles. After the friction coefficient of the particles was calibrated, several experimental runs with differing slit sizes were run. These flow rates were compared to the mass versus time data that Chrono::Engine output for the corresponding slit size. Along with comparing mass flow rates, the pile repose angle from the experimental runs was compared to the simulation results.

2. Equipment and Procedure

This section focuses on the different components of the equipment used for the validation experiment.

2.1 Description of Design with Iterations

Three criteria were considered essential for proper functionality of the granular flow meter:

- The granular material must be contained in the cavity and leave only through the slit.
- The walls of the cavity must have a similar friction coefficient to the granular material.
- The shape of the cavity must be easily altered.

In an effort to achieve these design goals, several versions of the flow meter were created.

2.1.1 Initial Design

Due to the limitations of the simulation hardware at the time this experiment was performed, only 50 to 60 thousand particles (about a tenth of a Newton) could be simulated. Because of this small amount of granular mass, it was necessary to be able to open the slit quickly so that the transient nature of the startup did not impact the measure of mass flow rate. Along with this, the opening mechanism had to have enough damping in it to reduce any dynamic response that might occur. To deal with these issues, a preexisting translational stage controlled by an actuator (described in Sections 2.3 and 2.4, respectively) was used. The cavity was created using two sub-parts: an angled slot and a static slot. The angled slot was attached to the translational stage whereas the static slot was fixed to the base. Both parts had removable inserts that could be replaced with different shapes, which was favored over a complex mechanical device to alter the shape of the cavity. In order to form the outside walls of the cavity, a smooth glass was to be attached to both sides by an all-purpose adhesive or clamps. Not only would the glass walls have properties closely resemble those of the granular material, they would also
provide a viewing window to the user. The initial design, without the glass walls attached, is shown in Figure 1. The translational stage is identified with a “TS”.

![Figure 1: Initial design of the experimental setup.](image)

To measure the mass of granular material flowing through the gap, a load cell would be placed under the gap. A large Petri dish would be attached to the load cell to contain the falling material.

### 2.1.2 Second Version of Design

The second version of the design, shown in Figure 2, had several departures from the first. The most notable is that the static slot components were replaced with a single part and the slot would be machined into the piece, rather than formed with the glass walls. This approach was favored because it was easier to machine and provided a consistent friction coefficient across the wall surfaces. The initial design required bolting the experimental rig to a table, whereas this version incorporated a base plate. This design aspect was chosen for portability and ease of installation.
2.1.3 Final Design

The final design closely resembles its predecessor; however, the dimensions have been altered to maximize the aluminum stock that was made available by the University of Wisconsin-Madison Student Shop. The height of the entire assembly was increased to ensure sufficient room for the load cell. The final design is shown in Figure 3.
2.2 Description of Part Manufacturing

All of the parts were manufactured with a tolerance of 2 thousandths of an inch at the University of Wisconsin-Madison CoE Student Shop using the Trak 2 mill shown in Figure 4.

2.2.1 Angled Insert

The angled insert was machined out of 1/2" thick aluminum plate. A chop saw and band saw were used to cut out the rough shape. The surfaces were then finished using a 1¼" end mill at a speed of 900 rpm. To guarantee an exact angle, 45º parallels were used to angle the piece in the vice. The part along with dimensions can be seen in Figure 5.
2.2.2 Angled Insert Holder

The angled insert holder was machined out of a 3"x3" aluminum post. The bulk of the material was removed using a 1¼" end mill at a speed of 900 rpm. The slot at the top of the piece was machined with ¼" end mill at a speed of 1100 rpm. The thru-holes on the base of the piece were drilled with a ¼" bit to allow clearance for ¼"-20 bolts to fasten it to the translational stage. Finally, the slot holes were drilled with a #7 bit and threaded with a ¼"-20 tap. This allowed the angled insert to be fastened into place by bolts. The part along with dimensions can be seen in Figure 6.

Figure 5: Angle insert and associated dimensions.
2.2.3 Static Slot

The static slot was machined out of a 3/4" thick aluminum plate. The bulk of the material was removed using a band saw, with the remaining surfaces being finished with a 1¼" end mill at a speed of 900 rpm. Fabrication was complicated by the 90° inside corner, which should have been replaced by a fillet. The slot at the front of the piece was machined with ¼" end mill at a speed of 1100 rpm. Two threaded holes (not shown) were drilled into the base of the piece using a #7 bit and ¼"-20 tap so the part could be fastened to the base. The part along with dimensions can be seen in Figure 7.

Figure 6: Angled insert holder and associated dimensions.
2.2.4 Base Plate

The angled insert was machined out of 1/2” thick aluminum plate. A chop saw and band saw were used to cut out the rough shape and the surfaces were finished with a 1¼” end mill at a speed of 900 rpm. Four threaded holes were drilled with a #7 bit and threaded with a ¼”-20 tap so the translational stage could be fastened to the base. Two thru-holes were drilled with a ¼” bit to allow the static slot to be fastened to the base. The part along with dimensions can be seen in Figure 8.
2.2.5 Assembly

The final assembly, including the translational stage, actuator, and load cell, is shown in Figure 9. To complete the assembly, \( \frac{1}{4} \)-20 bolts were used to fasten the components together.
2.3 Description of Translational Stage

The Newport UMR8.25 Linear Translational Stage [11] was used to constrain the motion of the angled insert. A schematic of the UMR8.25 is shown in Figure 10.

2.4 Description of Actuator

The actuator used in this experiment was the Newport LTA-HL Precision Linear Actuator [10]. It has a 25 mm range and a minimum incremental motion of 0.05 microns. The Newport LTA-HL features a manual knob that allows the user to match the actuator to the zero position of the rig. Figure 11 shows a picture of the actuator used in this experiment.
2.5 Description of Load Cell

The load cell used in this experiment is a Cooper LFS242 Tension/Compression Cell (Serial No. 286284) with a maximum load of .25 lbs and a repeatability of ±0.05% [4]. Although load cells typically work better in tension, the load cell was used in compression for this experiment. Several different mass displacements were tested to ensure that irregular piling did not cause an error in the mass reading. A Cooper DFI Infinity Digital Indicator was used to send the load cell signal through a router and to the processor.

2.6 Schematic of Experimental Setup and Data Transmission

Data sent from the load cell was interpreted using a script written in MATLAB. The script took an initial reading from the load cell and considered that value to be zero. A representation of the data flow can be seen in Figure 12.

An analog signal is generated by the load cell which is converted into a digital output by the indicator [5]. The digital signal is sent to the router which is then sent to the processor to be analyzed by MATLAB. The signal, that is, the granular material weight as a function of time, is read at approximately 4-5 Hz.

3. Summary of Experimental Results

Due to the ease with which the actuator could alter the gap size of the flow meter, several experimental runs were conducted for varying gap sizes. Since the actuator measures distance with a unit called steps, it was necessary to measure the desired gap size using a digital caliper. A fixed mass of 0.0624 N of granular material was used for each test with an uncertainty of 0.0004 N due to measurement error. Based on the results
of these experiments, an allowable "bandwidth" could be created with which the simulation results could be compared.

### 3.1 Results of 3 mm Gap

Figure 13 shows the experimental results with a gap size of 3 mm. Nine experimental runs were completed.

![Figure 13: Mass versus time experimental data for granular flow with a 3 mm gap.](image)

### 3.2 Results of 2.5 mm Gap

Shown below in Figure 14 are the experimental results with a gap size of 2.5 mm. Eight experimental runs were completed.

![Figure 14: Mass versus time experimental data for granular flow with a 2.5 mm gap.](image)
3.3 Results of 2 mm Gap

Shown in Figure 15 are the experimental results with a gap size of 2 mm. Nine experimental runs were completed.

![Figure 15: Mass versus time experimental data for granular flow with a 2 mm gap.](image)

3.4 Results of 1.5 mm Gap

Figure 16 shows the experimental results with a gap size of 1.5 mm. Nine experimental runs were completed.

![Figure 16: Mass versus time experimental data for granular flow with a 1.5 mm gap.](image)
4. Large Scale Multibody Dynamics on the GPU

This section briefly introduces the theoretical background for mechanical systems made up of multiple rigid bodies whose time evolution is controlled by external forces, frictional contacts, bilateral constraints and motors.

4.1 The Formulation of the Equations of Motion

The state of a mechanical system with \( n_b \) rigid bodies in three dimensional space can be represented by the generalized coordinates

\[
\mathbf{q} = \begin{bmatrix} \mathbf{r}_1^T, \mathbf{\dot{r}}_1^T, \ldots, \mathbf{r}_{n_b}^T, \mathbf{\dot{r}}_{n_b}^T \end{bmatrix}^T \in \mathbb{R}^{7n_b}
\]

and their time derivatives

\[
\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{\dot{r}}_1^T, \mathbf{\ddot{r}}_1^T, \ldots, \mathbf{\dot{r}}_{n_b}^T, \mathbf{\ddot{r}}_{n_b}^T \end{bmatrix}^T \in \mathbb{R}^{7n_b},
\]

where \( \mathbf{r}_j \) is the absolute position of the center of mass of the \( j \)-th body and the quaternion \( \mathbf{\dot{\omega}}_j \) expresses its rotation. One can also introduce the generalized velocities

\[
\mathbf{v} = \begin{bmatrix} \mathbf{\dot{r}}_1^T, \mathbf{\dot{\omega}}_1^T, \ldots, \mathbf{\dot{r}}_{n_b}^T, \mathbf{\dot{\omega}}_{n_b}^T \end{bmatrix}^T \in \mathbb{R}^{6n_b},
\]

directly related to \( \dot{\mathbf{q}} \) by means of the linear mapping

\[
\dot{\mathbf{q}} = L(\mathbf{q})\mathbf{v}
\]

where

\[
L(\mathbf{q}) = \begin{bmatrix}
+\epsilon_1 & +\epsilon_0 & -\epsilon_3 & +\epsilon_2 \\
+\epsilon_2 & +\epsilon_3 & +\epsilon_0 & -\epsilon_1 \\
+\epsilon_3 & -\epsilon_2 & +\epsilon_1 & +\epsilon_0
\end{bmatrix}
\]

Mechanical constraints, such as revolute or prismatic joints, can exist between the parts: they translate into algebraic equations that constrain the relative position of pairs of bodies. Assuming a set \( B \) of constraints is present in the system, they lead to the scalar equations

\[
\Psi_i(\mathbf{q}, t) = 0, \quad i \in B.
\]

To ensure that constraints are not violated in terms of velocities, one must also satisfy the first derivative of the constraint equations, that is

\[
\nabla \Psi_i^T \mathbf{v} + \frac{\partial \Psi_i}{\partial t} = 0, \quad i \in B
\]

with the Jacobian matrix \( \nabla \Psi_i = [\partial \Psi_i / \partial \mathbf{q}]^T \) and \( \nabla \Psi_i^T = \nabla_q \Psi_i^T L(\mathbf{q}) \). Note that the term \( \partial \Psi_i / \partial t \) is null for all scleronomic constraints, but it might be nonzero for constraints that impose some trajectory or motion law, such as in the case of motors and actuators.

If contacts between rigid bodies must be taken into consideration, colliding shapes must be defined for each body. A collision detection algorithm must be used to provide a set of pairs of contact points for bodies whose shapes are near enough, so that a set \( A \) of inequalities can be used to concisely express the non-penetration condition between the volumes of the shapes:

\[
\Phi_i(\mathbf{q}) \geq 0, \quad i \in A.
\]

Note that for curved convex shapes, such as spheres and ellipsoids, there is a unique pair of contact points, that is the pair of closest points on their surfaces, but in case of faceted or non-convex shapes there might be multiple pairs of contact points, whose definition is not always trivial and whose set may be discontinuous.
Given two bodies in contact \(A, B \in \{1, 2, \ldots, n_b\}\) let \(n_i\) be the normal at the contact point toward the exterior of body \(A_i\), and let \(u_i\) and \(w_i\) be two vectors in the contact plane such that \(n_i, u_i, w_i \in \mathbb{R}^3\) are mutually orthogonal vectors. When a contact \(i\) is active, that is, for \(\Phi_i(q) = 0\), the frictional contact force acts on the system by means of multipliers \(\lambda_{i,n} \geq 0, \lambda_{i,w}\), and \(\lambda_{i,w}\). Specifically, the normal component of the contact force acting on body \(B\) is \(F_{i,N} = \lambda_{i,n} n_i\) and the tangential component is \(F_{i,T} = \lambda_{i,u} u_i + \lambda_{i,w} w_i\) (for body \(A\) these forces have the opposite sign).

Also, according to the Coulomb friction model, in the case of nonzero relative tangential speed, \(v_{i,T}\), the direction of the tangential contact force is aligned to \(v_{i,T}\) and it is proportional to the normal force as \(\|F_{i,T}\| = \mu_{i,d}\|F_{i,N}\|\) by means of the dynamic friction coefficient \(\mu_{i,d} \in \mathbb{R}^+\). However, in the case of null tangential speed, the strength of the tangential force is limited by the inequality \(\|F_{i,T}\| \leq \mu_{i,s}\|F_{i,N}\|\) using a static friction coefficient \(\mu_{i,s} \in \mathbb{R}^+\), and its direction is one of the infinite tangents to the surface. In our model we assume that \(\mu_{i,d}\) and \(\mu_{i,s}\) have the same value that we will write \(\mu_i\) for simplicity, so the abovementioned Coulomb model can be stated succinctly as follows:

\[
\begin{align*}
\lambda_{i,n} &\geq 0, \quad \Phi_i(q) \geq 0, \quad \Phi_i(q)\lambda_{i,n} = 0, \\
\mu_i\lambda_{i,n} &\geq \sqrt{\lambda_{i,u}^2 + \lambda_{i,w}^2} \quad (F_{i,T}, v_{i,T}) = -\|F_{i,T}\| \|v_{i,T}\| \\
\|v_{i,T}\| \left(\mu_i\lambda_{i,n} - \sqrt{\lambda_{i,u}^2 + \lambda_{i,w}^2}\right) &= 0
\end{align*}
\]

Note that the condition \(\lambda_{i,n} \geq 0, \Phi_i(q) \geq 0, \Phi_i(q)\lambda_{i,n} = 0\) can also be written as a complementarity constraint: \(\lambda_{i,n} \geq 0 \perp \Phi_i(q) \geq 0\), see [14]. This model can also be interpreted as the Karush-Kuhn-Tucker first order conditions of the following equivalent maximum dissipation principle [8-9]:

\[
(\lambda_{i,u}, \lambda_{i,w}) = \arg\min_{\lambda_{i,u} \perp \lambda_{i,w}} \lambda_{i,u} v^T_{i,T} (\lambda_{i,u} u_i + \lambda_{i,w} w_i) . \quad (4.1.1)
\]

Finally, one should also consider the effect of external forces with the vector of generalized forces \(f(t, q, v) \in \mathbb{R}^{6n_b}\) that might contain gyroscopic terms, gravitational effects, forces exerted by springs or dampers, and torques applied by motors; i.e. all forces except joint reaction and frictional contact forces.

Considering the effects of both the set \(A\) of frictional contacts and the set \(B\) of bilateral constraints, the system cannot be reduced to either a set ordinary differential equations (ODEs) of the type \(\dot{v} = f(q, v, t)\), or to a set of differential-algebraic equation (DAEs). This is because the inequalities and the complementarity constraints turn the system into a differential inclusion of the type \(\dot{v} \in F(q, v, t)\), where \(F(\cdot)\) is a set-valued multifunction [12]. In fact, the time evolution of the dynamical system is governed by the following differential variational inequality (DVI):

\[
\begin{align*}
\dot{q} &= L(q) v \\
M\dot{v} &= f(t, q, v) + \sum_{i \in B} \lambda_{i,n} \nabla \psi_i + \\
&+ \sum_{i \in A} (\gamma_{i,n} D_{i,n} + \gamma_{i,u} D_{i,u} + \gamma_{i,w} D_{i,w}) \\
\dot{\lambda}_{i,n} &= 0, \quad \psi_i(q, t) = 0 \\
\lambda_{i,n} \gamma_{i,n} &\geq 0, \quad \gamma_{i,u} \gamma_{i,w} = 0, \quad \gamma_{i,u} + \gamma_{i,w} = 0 \\
(\gamma_{i,u}, \gamma_{i,w}) &= \arg\min_{\mu_i\gamma_{i,n} \geq \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}} v^T (\gamma_{i,u} D_{i,u} + \gamma_{i,w} D_{i,w}) \quad (4.1.2)
\end{align*}
\]
Here, to express the contact forces in generalized coordinates, we used the tangent space generators \( D_i = [D_{i,n}, D_{i,u}, D_{i,w}] \in \mathbb{R}^{6n_i \times 3} \) that are sparse and are defined given a pair of contacting bodies \( A \) and \( B \) as:

\[
D_i^T = \begin{bmatrix}
0 & \cdots & -A_{i,p}^T & A_{i,p}^T A_{i,A} \tilde{s}_{i,A} & 0 & \cdots \\
0 & \cdots & A_{i,p}^T & -A_{i,p}^T A_{i,B} \tilde{s}_{i,B} & 0 & \cdots 
\end{bmatrix}
\]  
(4.1.3)

Here \( A_{i,p} = [n_i, u_i, w_i] \) is the \( \mathbb{R}^{3 \times 3} \) matrix of the local coordinates of the \( i \)-th contact, and the vectors \( \tilde{s}_{i,A} \) and \( \tilde{s}_{i,B} \) to represent the positions of the contact points expressed in body coordinates. The skew matrices \( \tilde{s}_{i,A} \) and \( \tilde{s}_{i,B} \) are defined as

\[
\tilde{s}_{i,A} = \begin{bmatrix}
0 & -s_{i,A_y} & s_{i,A_y} \\
+s_{i,A_z} & 0 & -s_{i,A_x} \\
-s_{i,A_y} & +s_{i,A_x} & 0
\end{bmatrix} \quad \tilde{s}_{i,B} = \begin{bmatrix}
0 & -s_{i,B_y} & +s_{i,B_y} \\
+s_{i,B_z} & 0 & -s_{i,B_x} \\
-s_{i,B_y} & +s_{i,B_x} & 0
\end{bmatrix}
\]

The DVI in (4.1.2) can be solved by time-stepping methods. The discretization requires the solution of a complementarity problem at each time step, and it has been demonstrated that it converges to the solution to the original differential inclusion for \( h \to 0 \) [14-15]. Moreover, the differential inclusion can be solved in terms of vector measures: forces can be impulsive and velocities can have discontinuities, thus supporting also the case of impacts and giving a weak solution to otherwise unsolvable situations like in the Painlevé paradox [13].

### 4.2 The Time Stepping Solver

Within the aforementioned measure differential inclusion approach, the unknowns are not the reaction forces and the accelerations \( \dot{v} \) as in usual ODEs or DAEs. Instead, given a position \( q^{(l)} \) and velocity \( v^{(l)} \) at the time step \( t^{(l)} \), the unknowns are the impulses \( \gamma_s \), for \( s = n, u, w, b \) (that, for smooth constraints, can be interpreted as \( \dot{\gamma}_n = h \gamma_n, \dot{\gamma}_u = h \gamma_u, \dot{\gamma}_w = h \gamma_w, \dot{\gamma}_b = h \gamma_b \) ) and the speeds \( v^{(l+1)} \) at the new time step \( t^{(l+1)} = t^{(l)} + h \). These unknowns are obtained by solving the following optimization problem with equilibrium constraints [1]:

\[
\begin{align*}
M(v^{(l+1)} - v^{(l)}) = h f(t^{(l)}, q^{(l)}, v^{(l)}) + \sum_{i \in A} \gamma_i, b \nabla \psi_i + \\
+ \sum_{i \in B} (\gamma_i, n D_{i,n} + \gamma_i, u D_{i,u} + \gamma_i, w D_{i,w}),
\end{align*}
\]

\[i \in B: \quad \frac{1}{h} \psi_i(q^{(l)}(t)) + \nabla \psi_i^T v^{(l+1)} + \frac{d \psi_i}{dt} = 0,\]

\[i \in A: \quad 0 \leq \frac{1}{h} \Phi_i(q^{(l)}) + D^T_{i,n} v^{(l+1)} \perp \gamma_i, n \geq 0,\]

\[
\begin{align*}
(\gamma_i, n, \gamma_i, u) &= \argmin_{\mu \geq \gamma_i, n} v^T (\gamma_i, u D_{i,u} + \gamma_i, w D_{i,w}) \\
q^{(l+1)} &= q^{(l)} + h L(q^{(l)}) v^{(l+1)}. \quad \text{(4.1.4)}
\end{align*}
\]

The \( \frac{1}{h} \psi_i(q^{(l)}) \) term is introduced to ensure contact stabilization, and its effect is discussed in [2]. Similarly, the term \( \frac{1}{h} \psi_i(q^{(l)}) \) achieves stabilization for bilateral constraints.

Several numerical methods can be used to solve (4.1.4). For instance, one can approximate the Coulomb friction cones in 3D as faceted pyramids, thus leading to a LCP whose solution is possible by using off-the-shelf pivoting methods. However, these methods usually require a large computational overhead and can be used only for a limited number of variables.
Therefore, in a previous work [17] we demonstrated that the problem can be cast as a monotone optimization problem by introducing a relaxation over the complementarity constraints, replacing

\[
0 \leq \frac{1}{h} \Phi_i(q^{(l)}) + D_{i,n}^T v^{(l+1)} + \gamma_i^a \geq 0 \quad \text{with} \quad 0 \leq \frac{1}{h} \Phi_i(q^{(l)}) + D_{i,n}^T v^{(l+1)} - \mu_i \sqrt{(v^T D_{i,u})^2 + (v^T D_{i,w})^2} \leq \gamma_i^u \geq 0 .
\]

The solution of the modified time stepping scheme approaches the solution of the original differential inclusion for \( h \to 0 \) just as the original scheme [2]. Most importantly, the modified scheme becomes a Cone Complementarity Problem (CCP), which can be solved efficiently by an iterative numerical method that relies on projected contractive maps. Omitting for brevity some of the details discussed in [3], the algorithm makes use of the following vectors and matrices:

\[
\begin{align*}
\gamma_{i,a} & \equiv \{\gamma_{i,n}, \gamma_{i,a}, \gamma_{i,w}\}^T, \quad i \in A, \\
b_i & \equiv \left\{ \frac{1}{h} \Phi_i(q^{(l)}), 0, 0 \right\}^T, \quad i \in A, \\
b_i & \equiv \frac{1}{h} \Phi_i(q^{(l)}, t) + \frac{\partial \Phi_i}{\partial t}, \quad i \in B
\end{align*}
\]

(4.1.5)

The solution of the CCP is obtained by iterating the following expressions on \( r \) until convergence, or until \( r \) exceeds a maximum number of iterations, starting from \( v^0 = v^{(l)} \):

\[
\begin{align*}
\forall i \in A: & \quad \gamma_{i,a}^{r+1} = \Pi_{T_i} \left[ \gamma_{i,a}^r - \omega_i (D_i^T v^r + b_i) \right] \\
\forall i \in B: & \quad \gamma_{i,b}^{r+1} = \gamma_{i,b}^r - \omega_i (\nabla \Phi_i^T v^r + b_i) \\
v^{r+1} = v^r + M^{-1} \left( \sum_{z \in A} D_z \gamma_{z,a}^{r+1} + \sum_{z \in B} \nabla \Phi_z \gamma_{z,b}^{r+1} + h f(t^{(l)}, q^{(l)}, v^{(l)}) \right)
\end{align*}
\]

(4.1.6)

(4.1.7)

(4.1.8)

Note that the superscript \( (l + 1) \) was omitted for brevity.

The iterative process uses the projector \( \Pi_{T_i}(\cdot) \), which is a non-expansive metric map \( \Pi_{T_i} : \mathbb{R}^3 \to \mathbb{R}^3 \) acting on the triplet of multipliers associated with the \( i \)-th contact [17]. In detail, if the multipliers fall into the friction cone

\[
T_i = \{ \gamma_{i,a} \in \mathbb{R}^3 : (\gamma_{i,u}^2 + \gamma_{i,w}^2)^{1/2} \leq \mu_i \gamma_{i,n} \}
\]

they are not modified; if they are in the polar cone

\[
T^\circ_i = \{ x_i \in \mathbb{R}^3 : \langle x_i, \gamma_{i,a} \rangle \leq 0, \forall \gamma_{i,a} \in T_i \}
\]

they are set to zero; in the remaining cases they are projected orthogonally onto the surface of the friction cone. The over-relaxation factor \( \omega \) and \( \eta_i \) parameters are adjusted to control the convergence. Interested readers are referred to [3] for a proof of the convergence of this method.

For improved performance, the summation of Eq. (4.1.8) can be computed only once at the beginning of the CCP iteration, while the following updates can be performed using an incremental version that avoids adding the \( f(t^{(l)}, q^{(l)}, v^{(l)}) \) term all the time; in case there is no initial guess for the multipliers and \( \gamma_{i,a}^0 = 0, \gamma_{i,b}^0 = 0 \), Eq. (4.1.8) turns into:

\[
\begin{align*}
v^0 &= v^{(l)} + M^{-1} h f(t^{(l)}, q^{(l)}, v^{(l)}) \\
v^{r+1} &= v^r + \sum_{i} \Delta v_i
\end{align*}
\]

(4.1.9)

(4.1.10)

where
\[ i \in A: \quad \Delta \mathbf{v}_i = \sum_{i \in A} M^{-1} D_i \Delta \gamma^{r+1}_{i,a} \]
\[ i \in B: \quad \Delta \mathbf{v}_i = \sum_{i \in B} M^{-1} \nabla \Psi_i \Delta \gamma^{r+1}_{i,b} \]

In the case that only bilateral constraints are used, this method behaves like the typical fixed-point Jacobi iteration for the solution of linear problems. If one interleaves the update (4.1.8) after each time that a single \( i \)-th multiplier is computed in (4.1.6) or (4.1.7), the resulting scheme behaves like a Gauss-Seidel method. This variant can benefit from the use of Eq. (4.1.10) instead of Eq. (4.1.8) because it can increment only the \( \Delta \mathbf{v}_i \) term corresponding to the constraint that has been just computed. Also, this immediate update of the speed vector provides better properties of convergence (especially in case of redundant constraints) but it does not fit well in a parallel computing environment because of its inherently sequential nature.

### 4.3 The GPU Formulation of the CCP Solver

Since the CCP iteration is a computational bottleneck of the numerical solution proposed, a great benefit will follow from an implementation that can take advantage of the parallel computing resources available on GPU boards.

In the proposed approach, the data structures on the GPU are implemented as large arrays (buffers) to match the execution model associated with NVIDIA’s CUDA. Specifically, threads are grouped in rectangular thread blocks, and thread blocks are arranged in rectangular grids. Four main buffers are used: the contacts buffer, the constraints buffer, the reduction buffer, and the bodies buffer. Since repeated transfers of large data structures can adversely impact the performance of the entire algorithm, an attempt was made to organize the data structures in a way that minimized the number of fetch and store operations and maximized the arithmetic intensity of the kernel code. This ensures that the latency of the global memory can be hidden by the hardware multithread scheduler if the GPU code interleaves the memory access with enough arithmetic instructions.

Figure 17 shows the data structure for contacts, which contains two pointers \( B_A \) and \( B_B \) to the two touching bodies. There is no need to store the entire \( D_i \) matrix for the \( i \)-th contact because it has zero entries everywhere except for the two 12x3 blocks corresponding to the coordinates of the two bodies in contact. In detail, we store only the following 3x3 matrices:

\[
\begin{align*}
D^T_{i,v_A} &= -A^T_{i,p}, & D^T_{i,\omega_A} &= A^T_{i,p} A_A \hat{s}_{i,A} \\
D^T_{i,v_B} &= A^T_{i,p}, & D^T_{i,\omega_B} &= -A^T_{i,p} A_B \hat{s}_{i,B}
\end{align*}
\]

Once the velocities of the two bodies \( \dot{\mathbf{r}}_A, \mathbf{\dot{\omega}}_A, \dot{\mathbf{r}}_B \), and \( \mathbf{\dot{\omega}}_B \), have been fetched, the product \( D^T_{i,v} \mathbf{v}^r \) in Eq. (4.1.6) can be performed as

\[ D^T_{i,v} \mathbf{v}^r = D^T_{i,v_A} \dot{\mathbf{r}}_A + D^T_{i,\omega_A} \mathbf{\dot{\omega}}_A + D^T_{i,v_B} \dot{\mathbf{r}}_B + D^T_{i,\omega_B} \mathbf{\dot{\omega}}_B \]  \hspace{1cm} (4.2.1)

Since \( D^T_{i,v_A} = -D^T_{i,v_B} \) there is no need to store both matrices, so in each contact data structure only a matrix \( D^T_{i,v_A} \) is stored, which is then used with opposite signs for each of the two bodies.

Also, the velocity update vector \( \Delta \mathbf{v}_i \), needed for the sum in Eq. (4.1.10) is sparse: it can be decomposed in small 3x1 vectors. Specifically, given the masses and the inertia
tensors of the two bodies \( m_A, m_B, J_A, \) and \( J_B, \) the term \( \Delta \mathbf{v}_i \) will be computed and stored in four parts as follows:

\[
\Delta \mathbf{r}_{Ai} = m_A^{-1} D_{i,v_A} \Delta \gamma_{i,a}^{r+1}, \quad \Delta \omega_{Ai} = J_A^{-1} D_{i,\omega_A} \Delta \gamma_{i,a}^{\omega+1} \\
\Delta \mathbf{r}_{Bi} = m_B^{-1} D_{i,v_B} \Delta \gamma_{i,a}^{r+1}, \quad \Delta \omega_{Bi} = J_B^{-1} D_{i,\omega_B} \Delta \gamma_{i,a}^{\omega+1} \quad (4.2.2)
\]

Note that those four parts of the \( \Delta \mathbf{v}_i \) terms are not stored in the \( i \)-th contact or data structures of the two referenced bodies (because multiple contacts may refer to the same body, hence they would overwrite the same memory position). These velocity updates are instead stored in the reduction buffer, which will be used to efficiently perform the summation in Eq. (4.1.10). This will be discussed shortly.

![Figure 17: Data structures in GPU global memory.](image)

The constraints buffer, shown in Figure 18, is based on a similar concept. Jacobians \( \nabla \Psi_i \) of all scalar constraints are stored in a sparse format, each corresponding to four rows \( \nabla \Psi_{i,v_A}, \nabla \Psi_{i,\omega_A}, \nabla \Psi_{i,v_B}, \nabla \Psi_{i,\omega_B} \). Therefore, the product \( \nabla \Psi_i^T \mathbf{v}^r \) in Eq. (4.1.7) can be performed as the scalar value:

\[
\nabla \Psi_i^T \mathbf{v}^r = \nabla \Psi_{i,v_A}^T \dot{\mathbf{r}}_{Ai} + \nabla \Psi_{i,\omega_A}^T \dot{\Omega}_{Ai} + \nabla \Psi_{i,v_B}^T \dot{\mathbf{r}}_{Bi} + \nabla \Psi_{i,\omega_B}^T \dot{\Omega}_{Bi} \quad (4.2.3)
\]

Also, the four parts of the sparse vector \( \Delta \mathbf{v}_i \) can be computed and stored as

\[
\Delta \dot{\mathbf{r}}_{Ai} = m_A^{-1} \nabla \Psi_{i,v_A} \Delta \gamma_{i,b}^{r+1}, \quad \Delta \dot{\Omega}_{Ai} = J_A^{-1} \nabla \Psi_{i,\omega_A} \Delta \gamma_{i,b}^{\omega+1} \\
\Delta \dot{\mathbf{r}}_{Bi} = m_B^{-1} \nabla \Psi_{i,v_B} \Delta \gamma_{i,b}^{r+1}, \quad \Delta \dot{\Omega}_{Bi} = J_B^{-1} \nabla \Psi_{i,\omega_B} \Delta \gamma_{i,b}^{\omega+1} \quad (4.2.4)
\]

Figure 18 shows that each body is represented by a data structure containing the state (velocity and position), the mass moments of inertia and mass values, and the external applied force \( \mathbf{F}_j \) and torque \( C_j \). Those data are needed to compute the CCP iteration and solve for unknowns.

When it comes to the implementation of the CCP solver on the GPU, using kernels that operate on the abovementioned data buffers, the task is not trivial because the iteration cannot be performed with a single kernel. In fact, considering the iteration over Eqs. (4.1.6), (4.1.7), and (4.1.10), one can see that Eqs. (4.1.6) and (4.1.7) fit into parallel kernels that operate, respectively, one thread per contact and one thread per bilateral constraint. Moreover, the summation in Eq. (4.1.10) cannot be easily parallelized in the same way because it may happen that two or more contacts need to add their velocity
updates $\Delta v_i$ to the same rigid body: this would cause a race condition where multiple threads might need to update the same memory value, something that can cause errors or indefinite/nondeterministic behaviors on the GPU hardware. Therefore, in order to parallelize Eq. (4.1.10), a parallel segmented scan algorithm [6] was adopted that operates on an intermediate reduction buffer (see Figure 18); this method sums the values in the buffer using a binary-tree approach that keeps the computational load well balanced among the many processors. In the example of Figure 18, the first constraint refers to bodies 0 and 1, the second to bodies 0 and 2; multiple updates to body 0 are then accumulated with parallel a segmented reduction.

Note that several other auxiliary kernels that have minimal impact on the computation time are used to prepare pre-process data before the CCP starts, for example, to compute Eq. (4.1.9). Also, to speed up the computation, matrices $D^T_{i,v,A}$, $D^T_{i,v,A}$ and $D^T_{i,\omega,B}$ are not provided by the host; instead they are computed on the GPU using the data coming from the collision detection code, that is, $\bar{s}_{i,A}$, $\bar{s}_{i,B}$ and $\bar{n}_i$.

The pseudocode in Table 1 shows the sequence of the main computational stages at each time step, which, for the most part, are executed as parallel kernels on the GPU.

### Table 1: Pseudocode for the CCP solver.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Context</th>
<th>Operations / kernels</th>
</tr>
</thead>
</table>
| 1     | HOST    | Copy memory CPU $\rightarrow$ GPU  
|       | Serial  | Copy contact and body data structures from host memory to GPU buffers.  
|       |         | Copy constraint data (residuals $\bar{b}_i$ and Jacobians) into the constraint buffer. |
| 2     | GPU     | Force kernel  
|       | Parallel on | For each body, compute forces $f(t^{(i)}), q(t^{(i)}, v(t^{(i)})$, if any. Store these forces and torques into $F_j$ and $C_j$.  
| bodies|         | |
| 3     | GPU     | Contact preprocessing kernel  
|       | Parallel on | For each contact, given contact normal and position, compute in place the matrices $D^T_{i,v,A}$, $D^T_{i,\omega,A}$ and $D^T_{i,\omega,B}$, then compute $\eta_i$ and the contact residual $b_i = (\frac{1}{\rho}\Phi_i(q), 0, 0)^T$.  
| contacts|         | |
4 GPU Parallel on bodies

**CCP force kernel** 

For each body $j$, initialize body velocities: 

$$\dot{\mathbf{r}}_j^{(l+1)} = h \mathbf{m}_j^{-1} \mathbf{F}_j$$

and

$$\omega_j^{(l+1)} = h J_j^{-1} \mathbf{C}_j.$$

5 GPU Parallel on contacts

**CCP contact iteration kernel**

For each contact $i$, do 

$$\gamma_{i,a}^{r+1} = \Pi_{i,a} \left[ \gamma_{i,a}^r - \omega \eta_i \left( \mathbf{D}_i^T \mathbf{v}^r + \mathbf{b}_i \right) \right].$$

Note that $\mathbf{D}_i^T \mathbf{v}^r$ is evaluated with sparse data, using Eq. (4.2.1).

Store $\Delta \gamma_{i,a}^{r+1} = \gamma_{i,a}^{r+1} - \gamma_{i,a}^r$ in contact buffer. Use Eq. (4.2.2) to compute sparse updates $\Delta \dot{\mathbf{r}}$ and $\Delta \dot{\omega}$ to the velocities of the two connected bodies $A$ and $B$, and store them in the $R_{i,A}$ and $R_{i,B}$ slots of the reduction buffer.

6 GPU Parallel on constraints

**CCP constraint iteration kernel**

For each constraint $i$, do 

$$\gamma_{i,b}^{r+1} = \gamma_{i,b}^r - \omega \eta_i \left( \nabla \psi_i^T \mathbf{v}^r + h_i \right).$$

Note that $\nabla \psi_i^T \mathbf{v}^r$ is evaluated with sparse data, using Eq. (4.2.1).

Store $\Delta \gamma_{i,b}^{r+1} = \gamma_{i,b}^{r+1} - \gamma_{i,b}^r$ in contact buffer. Use Eq. (4.2.4) to compute sparse updates $\Delta \dot{\mathbf{r}}$ and $\Delta \dot{\omega}$ to the velocities of the two connected bodies $A$ and $B$, and store them in the $R_{i,A}$ and $R_{i,B}$ slots of the reduction buffer.

7 GPU Parallel on reduction slots

**Segmented reduction kernel**

Sum all the $\Delta \dot{\mathbf{r}}$ and $\Delta \dot{\omega}$ terms belonging to the same body, in the reduction buffer. This may require a sequence of short kernels.

8 GPU Parallel on bodies

**Body velocity updates kernel**

For each $j$ body, add the cumulative velocity updates which can be fetched from the reduction buffer, using the index $R_j$.

9 HOST Serial

**Check convergence and repeat from stage 5 if convergence tolerance is not reached**

10 HOST Serial

**Copy memory GPU - CPU**

Copy contact multipliers from GPU buffers to host memory, if interested in reaction forces.

Copy constraints multipliers from GPU buffers to host memory, if interested in reaction forces.

Copy rigid body velocities from GPU buffers to host memory.

---

Stages 1 and 10 can be avoided if one manages to keep all the data on the GPU, by letting the collision detection engine communicate with the CCP solver directly. Even if those memory transfers are executed only at the beginning and at the end of the CCP solution process, their impact on the overall simulation time might be significant.

### 5. A Setup of the Simulation in Chrono::Engine

In order to validate the experimental results described above with the DVI Approach it is necessary to build a simulation model which represents the actual setup with as much detail as possible. The experimental setup consists of four major parts, the trough, the spheres, the translational stage, and the load cell. Each of these parts is well described in
the Chrono::Engine model. Figure 19 shows these parts as they are implemented in the simulation model.

![Diagram of simulation model](image)

Figure 19: Simulation model

The glass spheres are described as perfect spheres. Each sphere has the same mass and friction coefficients as the others.

The trough, with its three walls and slope, is integrated as a mathematical description of planes. Each plane represents one wall or slope respectively. All planes are part of one body. This is necessary to give the planes certain properties such as friction and is also important for the collision detection. Not only do the spheres need to be checked for collision with each other, but each sphere also needs to be checked for collision with the planes.

In the experiment, the trough is opened by a translational stage. This stage moves the slope out of the holder. In the simulation model, this behavior is described as a motion of the plane representing the slope. The motion is captured from the data sheet of the translational stage. Different gap sizes can be used for each simulation and the opening time can be varied.

The load cell in the experimental setup measures the mass outflow through the gap. Because this measurement is based on the weight per time, in the simulation, it can be implemented as a count of spheres under a certain height for each time step and outputted into a data file for later post processing. The number of spheres multiplied by the mass and gravity gives the weight which can be compared with experimental results.

The spheres leaving the trough are falling on a plane described as the collector in the experimental setup.

In order to save computational time the simulation is split into two parts, one containing the filling process of the trough and the other containing the opening and measure process. If the same design of the trough is observed multiple times, for example, at different gap sizes, the filling process is always equal. Therefore it is unnecessary to compute this process for each observation; it only needs to be run once. The second process is then used for observation of the behavior of the system.

In the filling process the same amount of spheres will be created as used in the real experiment and randomly distributed in the trough. After the spheres have settled, the x-, y-, and z-position of each sphere is saved for the following simulation.
At the beginning of the outflow simulation, the position data set of the spheres is loaded into the model and the spheres will be created at the same positions they appeared in the filling process. The material properties of the spheres can be set as desired. The outflow and the position of each sphere can be saved at each time step for further post processing observation.

Approximately 39,000 spheres were used in both simulations for the representation of the behavior of the trough at a 45° slope and a step size of 5e-4 seconds.

6. Procedure Used to Select the Value of Mu

The friction coefficient of a certain material is not a constant value. It can depend on various environmental influences such as humidity, surface quality, temperature, etc. The friction coefficient of glass was an unknown in the validation process and needed to be determined before further observations could be done.

To achieve this, one experiment at a given gap size was performed and multiple simulations with the same setup and different friction coefficients were computed. The resulting mass outflows are distributed about the experimental results, as shown in Figure 20.

![Finding mu](image)

**Figure 20: Selection of mu**

This set of observation was done for a gap size of 2.5 mm. Four simulations were run with friction coefficients varying from 0.05 to 0.4.

Each simulation reaches a certain force value at a different time. Looking at Figure 20 at fixed values of force, it is reasonable that the time needed to reach a force value depends on the friction. The black lines indicate the different force values chosen for examination. For each line, a second order polynomial can be fit to the data points and be evaluated for the experimental time value. The mean of these eight resulting values is 0.1498. Therefore, the kinetic friction coefficient of the granular material was chosen to be 0.15.
7. Comparison of the Experimental and Simulation Results that Validates Chrono::Engine

Because the actuator made it so easy to alter the gap size of the flow meter, several experimental runs were conducted for varying gap sizes. Since the actuator measures distance with a unit called steps, it was necessary to measure the desired gap size using a digital caliper. A fixed mass of 0.0624 N of granular material was used for each test with an uncertainty of 0.0004 N due to measurement error. Based on the results of these experiments, an allowable "bandwidth" could be created with which the simulation results could be compared.

7.1 Results of 3mm Gap

Shown in Figure 21 are the experimental results with a gap size of 3mm. Nine experimental runs were completed. The average slope of the experimental runs was 1.41E-2 [N/s] and the slope of the simulation was 1.40E-2 [N/s].

![Figure 21: Mass versus time experimental and simulation data for granular flow with a 3 mm gap.](image)

7.2 Results of 2.5 mm Gap

The experimental results with a gap size of 2.5 mm are shown in Figure 22. Eight experimental runs were completed. The average slope of the experimental runs was 2.59E-2 [N/s] and the slope of the simulation was 2.62E-2 [N/s].
7.3 Results of 2 mm Gap

Shown in Figure 25 are the experimental results with a gap size of 2mm. Nine experimental runs were completed. The average slope of the experimental runs was 4.00E-2 [N/s] and the slope of the simulation was 4.05E-2 [N/s].
Figure 23: Mass versus time experimental and simulation data for granular flow with a 2 mm gap.

7.4 Results of 1.5 mm Gap

Figure 24 shows the experimental results with a gap size of 1.5 mm. Nine experimental runs were completed. The average slope of the experimental runs was 4.44E-2 [N/s] and the slope of the simulation was 4.48E-2 [N/s].
Figure 24: Mass versus time experimental and simulation data for granular flow with a 1.5 mm gap.

### 7.5 Summary of Results

Figure 25 summarizes the experimental flow rates compared to the flow rates of the simulation. The experimental flow rate is based on the average of several experimental runs and the uncertainty of the rates is determined by a Student’s T distribution with 95% confidence.
Figure 25: Comparison of theoretical (simulation) and experimental results. Uncertainty bars are based off a Student's T distribution with a 95% confidence interval.

In an effort to provide a more detailed representation of results, the values of the experimental and simulation results, along with percent error and bounds of uncertainty can be seen in Table 2.

<table>
<thead>
<tr>
<th>Gap Size [mm]</th>
<th>Experimental [N/s]</th>
<th>Simulation [N/s]</th>
<th>% Error</th>
<th>Upper Bound [N/s]</th>
<th>Lower Bound [N/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>1.41E-02</td>
<td>1.40E-02</td>
<td>0.66</td>
<td>1.43E-02</td>
<td>1.38E-02</td>
</tr>
<tr>
<td>2</td>
<td>2.59E-02</td>
<td>2.62E-02</td>
<td>1.03</td>
<td>2.73E-02</td>
<td>2.45E-02</td>
</tr>
<tr>
<td>2.5</td>
<td>4.00E-02</td>
<td>4.05E-02</td>
<td>1.48</td>
<td>4.43E-02</td>
<td>3.56E-02</td>
</tr>
<tr>
<td>3</td>
<td>4.44E-02</td>
<td>4.48E-02</td>
<td>0.75</td>
<td>5.02E-02</td>
<td>3.87E-02</td>
</tr>
</tbody>
</table>

8. Validation of Chrono::Engine in Regards to the Pile Angle

Granular material forms a pile when poured onto a flat surface. The shape of this pile can vary in its angle between the surface of the pile and the horizontal ground. This angle is called the angle of repose and depends on the shape of the particles, cohesion, and friction. Both the Chrono::Engine model and the experiment should show the same shape of the pile, meaning the angle of repose needs to be the same. In our experimental setup, as well as in the simulation model, the shape of the particles is spherical with a diameter of 500 µm and cohesion is not part of the observation in this work. The Mohr-Coulomb criterion shows that the angle of repose is approximately equal to the friction angle [7]. In order to validate that the simulation model behaves in the same way as the experiment, the friction angle needs to be determined from the pile formed in the experiment and used in the Chrono::Engine model. Because Chrono::Engine can only handle one friction
value, either dynamic or static friction, the value needs to be switched during the run of
the simulation.

Figure 26 shows the conical pile formed in the experiment. The friction is determined
through the measurement of the angle $\phi$ between the ground and the pile surface.

$$\mu_{\text{static}} = \tan^{-1}(\phi)$$

The static friction coefficient was determined to be 0.35, a value subsequently used in
simulation in Chrono::Engine.

Figure 27 shows the pile built up in the simulation model with exactly the same shape
as in the experiment. The angle measurement shows that the angle of repose is the same
for the simulation and the experiment.

9. Future Directions of Work

The work performed during this experiment leads to several directions of study. Of
these directions, the easiest to pursue with the current experimental setup would be
testing different cavity shapes and particle sizes. By changing the material properties of
the walls (such as using plastic instead of aluminum) the effects of boundary conditions
on the mass flow rate could be judged. Additionally, the streamlines of the granular flow
could be mapped by using different colored sand. Finally, to capture yet another property
of the system, a cone penetrometer could be used to measure the force required to
penetrate a container of sand with a cone. This experiment would require an entirely
different experimental and simulation setup, as well as a larger number of particles.
10. Project Conclusions
Based on an error of less than 2%, the simulated results match the experimental results well. Despite this, there are several factors that could be improved upon. Namely, the design of the rig did not exactly match the simulated trough. The rig was machined from aluminum, whereas the trough in the simulation had the same coefficient of friction as the granular material. Likewise, the simulation was conducted in essentially vacuum, ignoring any aerodynamic forces, while the experiment was performed at ambient conditions. Lastly, the effects of humidity on the experiment were not taken into account. With that said, the trends of the flow rates are encouraging because the simulation engine is able to predict the nonlinear behavior that occurs near a 3 mm gap.

11. Acknowledgements
The authors would like to acknowledge Alessandro Tasora and Mihei Anitescu for their work in GPU-based parallel computing for the simulation of complex multibody systems.

12. References


