VIII International Conference on Particle-Based Methods
PARTICLES 2023
U. Perego, M. Cremonesi, A. Franci (Eds)

SIMULATING INDUSTRIAL SCENARIOS WITH THE OPEN-SOURCE SOFTWARE MERCURYDPM

A.R. Thornton[1,2], Q.H. Nguyen[3], H. Polman[1,2], M.J. Post[1,2], J.W. Bisschop[2], R. Weinhart-Mejia[2], D. Fitzsimmons[2], M.R. Vesal[2], T. Plath[1], I. Ostanin[1], T. Weinhart[1,2]

[1]Faculty of Engineering Technology, Department of Thermal and Fluid Engineering, MESA+, University of Twente, The Netherlands

[2]MercuryLab B.V., The Netherlands

[3]Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands

Abstract:

Creating predictive computer simulations, i.e. virtual prototypes, of complex granular industrial processes has many challenges. In this paper we review recent advances in creating such virtual prototypes. We introduce the open-source code MercuryDPM [1], which is often applied to complex industrial applications via the spin-off company MercuryLab. We briefly discuss how to import complex industrial geometries and how to deal with large numbers of particles and wide size-distributions.

Then we focus on how to create a computer representation of an actual granular material, the so-called model calibration. For calibration, we start by reviewing what parameters need to be measured and what experimental characterisation machines are available. We present an industrially practical calibration method, where certain parameters are directly measured and others are indirectly calibrated, using a variety of machine-learning techniques, implemented in the open-source codes GrainLearning [2], TensorFlow [3] and scikit-learn [4]. With GrainLearning, one can find local optima in only two to three iterations, even for complex contact models with many microscopic parameters. On the other hand, TensorFlow and scikit-learn use two popular supervised learning algorithms, Neural Network (NN) and Random Forest (RF) regression, respectivly. After a training period consisting of hundreds of particle simulations, NN and RF are capable of providing a mapping between the micro-parameters and the bulk behaviour, which can be used to find the optimal micro-parameters that correspond to the experimentally observed behaviour.

Keywords. Granular Materials, Virtual Prototyping, Calibration, DEM

1 INTRODUCTION

Developing quantitatively accurate, predictive computer simulations, often referred to as virtual prototypes, for complex granular industrial processes is a formidable undertaking. Three key issues loom large: First, an accurate model of the complex industrial

geometry needs to be imported into the software. Second, the algorithm used to simulate the granular bulk needs to be capable of efficiently and accurately simulating the wide shape- and size- distributions found in industrial materials. Third, the material model needs to be calibrated, ensuring that the simulated materials behave in a manner that faithfully replicates their real-world counterparts.

In this paper, we will describe all three problems; however, here we focus on the third problem, *i.e.*, calibration.

1.1 Discrete particle simulations in MercuryDPM

All simulations presented in this paper are performed in MercuryDPM, an open-source code for discrete particle simulations [1]. That is, it simulates the motion of particles, or atoms, by applying forces and torques that stem either from external body forces, (e.g. gravity, magnetic fields, etc...) or from particle interaction laws (e.g. Lennard-Jones). For granular particles, these are typically contact forces (elastic, plastic, viscous, frictional), while for molecular simulations, forces typically stem from interaction potentials (e.g. Lennard-Jones). The code has been developed extensively for granular applications, but could be adapted to include long-range interactions as well.

1.2 Dealing with industrial geometries

Industrial geometries often exhibit complexity and curvature. MercuryDPM features an STL reader designed to efficiently and accurately process both, static and moving/rotating walls, while also providing robust support for intricate curved shapes without requiring triangulation. For many industrial applications, the ability to preserve the curved geometries and to import external geometry files is paramount. This capability significantly enhances simulation speed and accuracy. Further details on this critical aspect are available in the comprehensive MercuryDPM paper [1], which we recommend to interested readers for a more thorough exploration of this topic.

1.3 Dealing with industrial materials

Industrial granular materials are often poly-dispersed and complex in shape, both of which present challenges in simulations. Firstly, in traditional simulations, the computational cost of finding particle contacts escalates with increasing size ratio. MercuryDPM mitigates this issue by using the hierarchical grid method which means the code scales linearly with particle number even for wide size-distributions. It also provides support for importing (and reconstructing) particle-size distribution from measured data. These topics are elaborated in greater detail in [1, 5].

Secondly, the creation of arbitrarily shaped particles can be a formidable task. MercuryDPM supports several methods for creating complex shapes [1]. Among these methods, the newly introduced implementation of particle 'clumps' [5] stands out as a remarkably flexible solution. This is a significant new feature and its implementation is described in a separate paper [6].

2 CALIBRATING AN INDUSTRIAL MATERIAL

Unlike fluid dynamics, there are no standardised measuring procedures for calibrating granular materials [7]. Instead, a multitude of apparatuses exist that measure various material properties. These may or may not be appropriate for the material that needs to be simulated. Here, we will present the ideas currently used by MercuryLab in order to calibrate a contact model for a real industrial material. Even though the workflow was developed by MercuryLab, all codes are fully available in open-source projects.

There are many contact models for granular materials, we will not review them here; however, in general, to fully calibrate a contact model we need to know:

Size (and shape) distribution	Stiftness
(Material) density	Charge
Cohesion	Attrition and breakage
Moisture content	Wall friction/adhesion
	Bond properties
Friction properties (sliding, rolling, torsion)	Liquid properties

The list is ordered based on our experience of which properties mostly affect dynamic industrial processes; it should be noted that for static processes the order will be different. The properties in grey are only required for specialised materials, e.g. for materials that triboelectrically charge, break, etc, and will not be considered here. The vertical line represents the minimal parameters that need to be calibrated in order to get meaningful results. Of course, the more parameters calibrated, the more accurate the results.

We will distinguish two types of calibration, direct and indirect calibration. Direct calibration means we can directly measure the contact model parameters using particle or particle-pair measurements; whereas, indirect calibration means we use characterisation machines measuring bulk properties (using samples of typically 0.1-1 litres) and match the results of these measurements with simulations of the equipment.

It is possible to directly calibrate all parameters, see e.g. [8, 9]. This method works very well but has several limitations: it only can be done for simple materials (e.g. glass beads) [9], the experiments are very time-consuming and require expensive equipment [8, 9], and measurements have to be repeated for lots of particles to get a good calibration of the average (bulk) material [10].

We use a hybrid approach, calibrating the green parameters via direct calibration and indirect calibrate other parameters in a second step. For size distributions we use either laser diffraction or air jet sieving; however, it should be noted that even though both methods work well, care has to be taken interpreting the data as the two machines measure different properties and there are still some open issues with very small particles. For the material density we use gas pycnometry, whereas, the moisture is easily calibrated with a simple oven-drying test. The last important parameter, cohesion, we will calibrate indirectly. Note that there are many machines on the market which create cohesion indexes for materials; however, it is not clear how to extract cohesion strength from these dimensionless indexes. Also, stiffness (restitution) and frictional properties will be

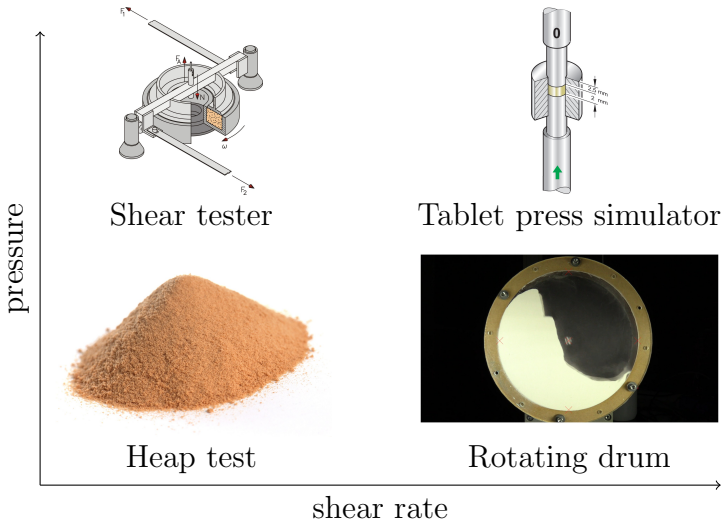

Figure 1: Sketch of the pressure shear rate phase space indicating which characterisation machines to use in which parameter range

Figure 2: From left to right: (1) Design of the THOR. (2) Snapshot of material (Microcrystalline cellulose). (3) White: average filled volume, magenta: location of the centre of mass. (4) White volume is filled 90% of time, yellow 50% of time and red 10% of time. (5) Mean free surface profile and linear fit to free surface profile

calibrated indirectly. This requires us to perform characterisation experiments and match the results in a virtual prototype of the same apparatus.

2.1 Choosing your characterisation experiment

Calibrating a material model is very hard; calibrating a material model that is valid under arbitrary conditions is even harder. Therefore we calibrate using characterisation machines that operate at shear rates and confining pressures similar to the process of interest. Our recommended characterisation machines are shown in Figure 1. Next we will discuss further exactly what we measure for the three most common cases.

2.1.1 Rotating drum

Rotating drums are designed to measure the dynamic angle of repose, i.e., the slope angle at which a granular material flows at a fixed shear rate. There are many companies that make rotating drums, e.g. the GranuDrum; however, we developed our own setup

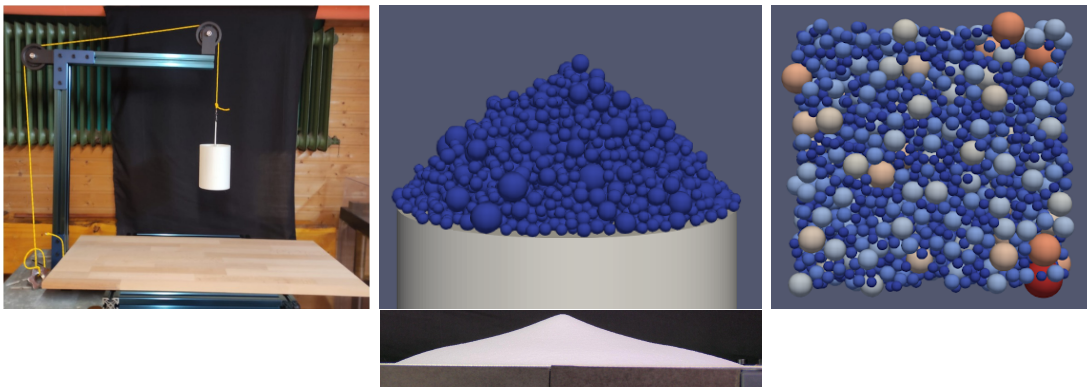

Figure 3: Left: Image of 'pull up' heap tester experiment; Middle: Top, snapshot of a virtual prototype of a 'free-flowing' tester. Bottom, view of heap formed from salt in the heap tester, after black-white threshold is applied to identify the pixels containing granular materials; Right: Snapshot of a shear tester simulation in MercuryDPM

called the THOR at the University of Twente, which was designed for studying segregation [11]. There were many reasons for developing a new drum. However, one of the key reasons is the fact that it can operate with many different drum sizes, up to 60cm in diameter, allowing us to calibrate a wide range of materials, from fine powders to gravel-like material such as sinter. A schematic of the THOR with its flexible drum design can be seen in Figure 2. The virtual prototype is a rotating drum whose diameter is scaled down to 30 times the average particle size for efficiency.

From the rotating drum we can extract the mean location and spread of the centre of mass, a cubic fit to the free surface, a linear fit to the free surface, the variation of the fill volume. All of these are illustrated in Figure 2 and can easily be extracted from a virtual prototype as well. Often, we use the centre of mass as this is a more robust definition than the fits to the free surface. However, care has to be taken in the experiments to correctly determine the location of the drum centre; otherwise, you introduce a systematic error.

2.1.2 Heap test

Heap testers are designed to measure the static angle of repose, i.e., the maximum slope angle at which a heap of granular material is at rest. There are several different heap tester designs which differ in the way the heap is constructed. Often people use the the so-called 'cylinder pull-up test' of Roessler and Katterfeld [12]; however, it should be noted that with this method you have to be careful with your preparation steps as the results are sensitive to initial conditions. Therefore it may be better to use a heap tester which utilises initially flowing, not static material. We used the Granuheap for the limestone calibration example later in the paper. In our virtual prototype the geometry is smaller than in the experiments to reduce the number of particles in the simulation (the cylinder diameter equals 30 times the average particle diameter); this is possible as the results are known to be scale-independent at low velocities [12].

After the heap is formed we measure the angle of repose by taking a sideway snapshot

of the heap, then find the area that contains granular material, fit a triangle which has the maximum overlap with this area, and measure the triangle's inclination. For consistency, we apply the same procedure to the simulated heap.

2.1.3 Shear tester

Shear testers are designed to measure the shear strength of a granular material. The shear strength can be defined either as the maximum shear stress the material can withstand before it undergoes shear failure, or the shear stress of the material under steady-state conditions, i.e. long after failure. Measurements are usually done at a constant compressive stress and shear rate. Again, there are several commercially available shear testers, which all use slightly different operating conditions. We typically use a Schulze ring shear tester.

Simulating the full ring shear tester would be too computationally costly because of the large amount of bulk material to be simulated. Thus, we instead use a periodic shear cell that deforms with controlled compressive stress and shear rate. The side length of the periodic box is only 15 times the average particle size, which vastly reduces the amount of material that needs to be simulated. We simulate all stages of the experiment:

1. Precompression:
   (a) Compress the sample until steady state is reached (controlled compressive stress, controlled shear rate)
   (b) Shear back until shear stress is zero (controlled compressive stress, controlled negative shear rate)
   (c) Relax the sample (controlled compressive stress, free shear rate)
2. Shear stage (for each compression stress value):
   (a) Read in sample from stage 1-(c) and decompress to the new, lower compression stress (controlled compressive stress, free shear rate)
   (b) Shear and measure the peak value of shear stress (controlled compressive stress, controlled shear rate)

This is not a simple simulation to undertake: To create the periodic box, we use Lees-Edwards boundary conditions, where the size of the periodic box is controlled by a PI software controller [13], with constant proportional and integral gain parameters. A simulation snapshot can be found in Figure 3.

2.2 Calibrating your material

So far we have discussed which type of characterisation machines to choose, how to create a virtual prototype of these characterisation machines, and which parameters to measure. Here we will discuss how to obtain the contact model parameters such that the virtual prototype matches the experimental results. In the following, the calibration will be treated as an optimisation problem: First, one selects a set of model parameters $\boldsymbol{\theta}$ that need to be calibrated and a set of experimental measurements $\mathbf{X}_{\text{exp}}$ that need to be matched. The virtual prototypes can then be thought of as a function $\mathbf{X}(\boldsymbol{\theta})$ that outputs

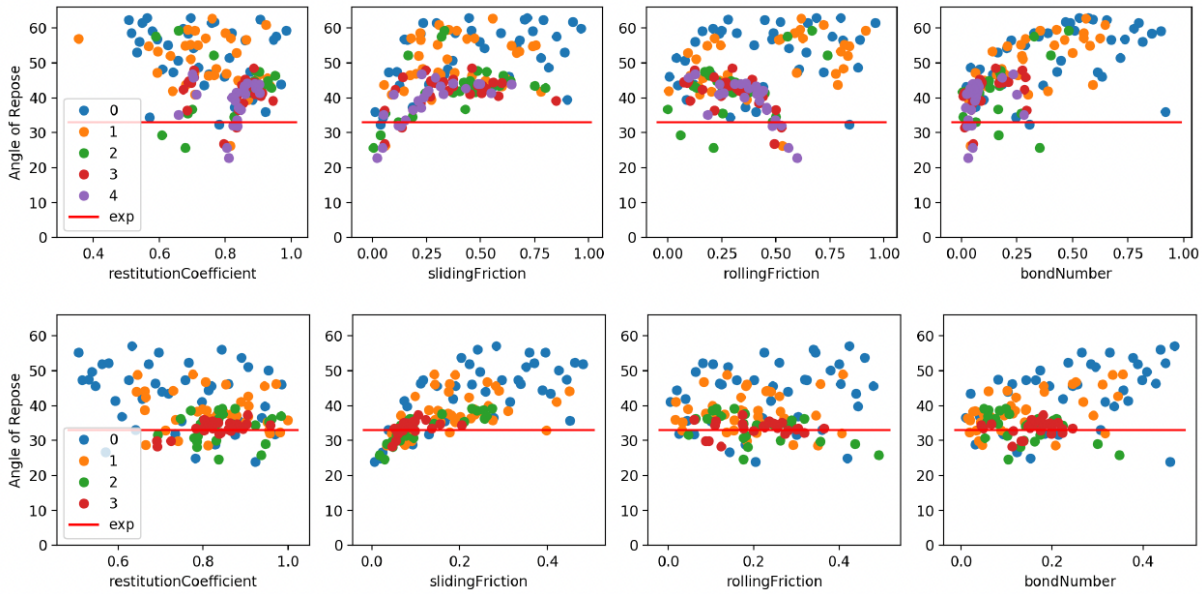

Figure 4: Value of angle of repose found by each simulation performed, colour by interaction count. Red line indicates the value obtained from the characterisation experiments. The different rows indicate two different runs with different initial ranges

the same measurements for given model parameters. Then we select an optimisation technique to find the input parameters $\boldsymbol{\theta}_{\text{opt}}$ such that $|\mathbf{X}(\boldsymbol{\theta}) - \mathbf{X}_{\text{exp}}|$ is minimised.

In the case discussed below, the model parameters are the restitution coefficient, sliding friction, rolling friction and bond number (cohesion), and the experimental measurement is the static angle of repose for limestone powder, measured to be 33° in a heap test. For optimisation, we will first use GrainLearning (version 0.2), the tool we currently utilise. Thereafter, we briefly discuss alternatives.

Given we have to fit four model parameters, but only one experimental measurement, we do not expect the optimisation to result in a single optimal value.

2.2.1 GrainLearning

GrainLearning (GL) is a calibration toolbox developed by Cheng *et al.* [2], which utilises a recursive Bayesian filtering algorithm to estimate the uncertainty of parameters in DPM against experimental observations. Initially, a wide parameter range is quasi-randomly sampled, assuming a uniform prior distribution. Then, conditioned on the experimental observations, a posterior distribution is computed by Sequential Monte-Carlo Filtering (SMC Filter) and fitted to a Gaussian Mixture Model, and used as a prior distribution for the next iteration. It should be noted that subsequent interactions can leave the initial parameter range. For the interested reader, more details can be found in either [2] or [14].

The results of the optimisation is shown in Figure 4. The markers show the static angle of repose obtained for each sample of model parameters; blue markers denote results

Method	Restitution c.	Sliding fr.	Rolling fr.	Bond number	AoR
Experiment					33
GL Attempt 1, Iter. 0	0.7812	0.037	0.84	0.3061	32.3163
Iter. 1	0.869	0.2725	0.3652	0.0325	39.4803
Iter. 2	0.831	0.1194	0.484	0.064	33.0406
Iter. 3	0.8332	0.135	0.5262	0.0137	31.5243
Iter. 4	0.8348	0.1517	0.497	0.0297	33.6745
GL Attempt 2, Iter. 0	0.6406	0.037	0.36	0.3061	33.0979
Iter. 1	0.9546	0.3983	0.032	0.0334	32.9993
Iter. 2	0.8263	0.0917	0.2226	0.1411	34.1421
Iter. 3	0.8025	0.0710	0.2802	0.1748	32.9812
NN	0.6875	0.2500	0.0417	0	32.4263
	0.5833	0.1667	0.1250	0	32.7926
	0.7241	0.2759	0.0345	0	32.3962
RF	0.7041	0.0612	0.2245	0.1633	32.4872
	0.7347	0.0816	0.4082	0.1225	33.0331

Table 1: Best calibration results of the calibration of Limestone powder

from the initial (zeroth) iteration, orange, green, red and purple denote results for the consequent iterations. The red line shows the experimentally measured static angle of repose (AoR) for comparison.

In the first attempt (top row of Figure 4), the restitution coefficient was initially assumed to be in the range 0.5–1.0 and all other parameters in the range 0–1.0. One remarkable observation is that GL did not converge immediately to an optimal solution. Instead, the first iteration returned an optimal solution which had an AoR of 39.48°, far from the experimental value of 33°. The third iteration's resulted in an optimal solution with an AoR of 33.04°, however, the optimal sample seemed to be an outlier. Consequently, an additional iteration was performed, and the results here verify the observation: The optimal parameter set produced an AoR of 31.52° and it is also an outlier. A fourth iteration was aided which starts to converge to the expected value.

After this first calibration attempt, it became clear that the combinations that would result in the desired static AoR have a restitution coefficient of 0.8 – 1.0 and the rest of the contact parameters lie in the range 0–0.5. Therefore a second attempt was performed with the above-mentioned initial ranges. This time, the algorithm converged quickly, resulting in an optimal solution in the second iteration, with an AoR of 32.9993°.

The best cases of each calibration run are shown in the top half of Table 1. It should be noted that overall, GrainLearning has demonstrated the capability to identify a (local) optimum, i.e. a cluster of near-optimal values, as long as the initial ranges are correctly chosen. However, it can also give inconsistent performance sometimes clustering in the wrong range and only slowly converge to good values, as seen by attempt 1. More details and a second test case can be found in the BSc thesis of Nguyen [14]. MercuryLab has used GrainLearning for many industrial materials (not just the example above) and often

it gives a nice cluster of possible results; however one has to carefully check that the best result is not an outlier. In these cases it can be solved by running again with a different 'improved' initial range or even the same range with a new initial quasi-random sample. This can be seen by the Limestone example; however, it should be noted this example was chosen as it shows this behaviour and this is not a common occurrence, one just has to be 'unlucky'. However, this problem can be minimised by using 'educated' initial parameter ranges.

Running an iterative optimiser like GrainLearning is very computationally costly in both terms of CPU time and wall time, as one can parallelise the simulation of any given iteration, but you cannot parallelise across the iterations. Therefore we end this section with a few practical suggestions on how to calibrate materials with GrainLearning to reduce the computational cost:

- Only use two iterations (3 runs in total), as the change after the second iteration is often very small if you have good clustering, i.e. 'convergence' of the posterior distribution.
- Check there are many similarly good results, i.e. a cluster of close values; if not, restart the process with a new initial range informed from the first attempt.
- Take the best results of all data, not just from the final iteration; as sometimes there are better results in an earlier iteration.
- You could use the Gaussian Mixture to interpolate the best result; however, we prefer to choose optimal values only from the sampled parameter values, for which the actual simulation was run.
- Prematurely terminate an iteration when most of the simulations have finished. It is our experience that some simulations take a lot longer than the average and these often produce very poor results as they are unphysical. Terminating these simulations and moving on to the next iteration can significantly reduce the computation wall time, with minimal effect on the results.

2.3 Supervised learning methods

In this section we will do a preliminary investigation using two supervised learning methods for calibration namely, Neural Network and Random Forests. For these methods we use the open-source library TensorFlow [3] and scikit-learn [4], respectively.

Artificial Neural Network (NN) is a set of algorithms that seeks to identify correlations in data utilising a technique inspired by how the human brain operates - mimicking how each neurone in the brain signals each other. Here we will use a Feedforward Multilayer Perceptron Neural Network architecture.

A Random Forest (RF) is a set (forest) of decision tress. The decision is an algorithm that generates a tree graph of decisions based on the input provided and their possible outcomes, and as a consequence, it partitions the input space into multiple regions, with each region accounting for a different outcome. Decision trees are relatively simple, explainable, easy to train and interpolate with little computational resources. However, one crucial drawback of a decision tree is its instability: minor data changes might affect the

tree structure, making the decision tree a high variance estimator. Random forests make up for the high variance of a single decision tree by averaging the results over a 'forest' of decision trees, with each tree representing an independent sampled vector [15, 14]. Note, both NN and RF need to be trained before they can be utilised.

We now consider the Limestone powder case. 500 particle simulations with randomised combinations of input parameters were performed to train the NN and RF model. To find the optimal value, 800,000 and 2,000,000 input cases were considered for the NN and RF models respectively. Results which were within 0.1° of the experimental AoR were kept. This resulted in 12 combinations for the NN and 22 for the RF; however, many are very close in parameters, hence we grouped them to 9 distinct parameter sets. Finally, a particle simulation was run for each of these sets, which resulted in 3 good combinations for the NN and 2 for the RF. The good values can be found in Table 1. For more details and a second example we refer the interested reader to the BSc thesis of Nguyen [14].

2.4 Comparison of techniques

The simulation time to compute the AoR in a heap test varies between 2 hours and 24 hours on a single processor. Due to the parallel nature of the calibration routines it was not deemed necessary to parallel the individual codes. GrainLearning requires around 120 to 150 simulations per optimisation; hence, if it has to be rerun, it can be up to 300 simulations to get good results. Evaluating the RF and NN require around 400-500 particle simulations in order to train the model, and about 20 validation simulations to confirm the optimal results. Evaluating the trained model is so quick that it is in essence free. The Achilles' heal of these approaches is the fact that training data is created for a given moisture content, material density and particle-size distribution. Hence, it is very hard to reuse the training data for a second calibration problem.

The spread between the various 'optimal' model calibration values shows that it is hard to uniquely determine model parameters from a single characterisation result; several distinct parameter values all give acceptable results in the same shear-rate versus confining pressure state as the characterisation machine. Hence we repeat our earlier point that it is important to choose the right characterisation machine for the right processes. We are still not able to create a universal calibration of a complex industrial process; however, it is possible to relatively quickly and automatically get values that can be practically used.

Finally, we did not discuss here how to choose the contact model and only showed results using the simple linear spring-dashpot model. In essence for industrial cases we have a hierarchy of increasingly complex contact models and an outer loop increases the complexity of the model if the current model is not able to match the experimental data within a given tolerance.

In the future developing an approximate NN or RF for generic particle size distributions and using this as a first step to determine the initial range of the GL algorithm could be a good approach. Therefore we feel the future may be a hybrid approach combining the pros and cons of the presented methods.

3 ACCESSING THE CODE

MercuryDPM is freely available open source. All the codes and calibration scripts can be found at https://bitbucket.org/mercurydpm/ including the image analysis code. Also MercuryLab is developing a simple to use cloud based version of both MercuryDPM and the calibration tool, which due to its graphical user interface is far easier to use. The so-called MercuryCloud is deployed on AWS, providing computing power, scalability and on-demand particle simulations. Further, the cloud simulations are cost-optimised with respect to AWS. For more information email cloud@mercurylab.org.

4 CONCLUSION

Particle simulations are just getting to the maturity where they can be applied and used to design industrial processes. Here, we review the three main challenges in moving from academic to industrial problems, i.e. dealing with complex geometries, dealing with industrial materials and accurately calibrating the models. We show how the open-source code MercuryDPM is able to solve these challenges. This is now often applied to industrial problems via the spin-off company MercuryLab.

In this paper we focus on the calibration problem and showed which parameters can be measured and which need to be inferred from characterisation measurements. We demonstrate three methods of machine learning to obtain these inferred parameters, namely GrainLearning, Neural Network, and Random Forest algorithm, for the case of one bulk parameter. Overall, it has been found that all tested algorithms can search for correct micro-parameters combinations to reproduce the experimentally measured static AoR in MercuryDPM – albeit in vastly different ways. While GL iteratively samples the new set of parameters that is getting closer to the optimal value based on previously-learned knowledge, NN and RF models require a three-step method: training with different particle simulations, then feeding it with multiple combinations and selecting the output that matches the experimental values, and finally validating it using a particle simulation. It has also been shown that GrainLearning does not always converge quickly, presumably due to the initial parameter range being too large. Meanwhile, the NN and especially the RF model demonstrate the capability to learn the mapping between the particle input parameters and its bulk behaviour. However, in the future a hybrid method may be the best, using pre-trained NN or RF mappings to narrow the initial parameter range of the GL method.

In conclusion, we are able to simulate industrial scenarios with the open-source code MercuryDPM. However, care has to be taken with respect to calibration in order to obtain meaningful results.

Acknowledgements

UT authors acknowledge support of the Dutch Research Council, NWO-TTW grants 15050 and 16604 and ML authors acknowledge support of the Provincie Overijssel for the MIT-haalbaarheidsproject ‘MercuryLab - New Calibration Method for Micro-Scale Discrete Particle Simulations’.

REFERENCES

[1] T. Weinhart et al. Fast, flexible particle simulations — an introduction to MercuryDPM. *Computer Physics Communications*, 249:107129, 2020.

[2] H. Cheng et al. An iterative Bayesian filtering framework for fast and automated calibration of DEM models. *Computer Methods in Applied Mechanics and Engineering*, 2019.

[3] TensorFlow Developers. Tensorflow, August 2023. https://doi.org/10.5281/zenodo.8256979.

[4] F. Pedregosa, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] A. R. Thornton, T. Plath, I. Ostanin, H. Götz, J. W. Bisschop, M. Hassan, R. Roeplal, X. Wang, S. Pourandi, and T, Weinhart. Recent advances in MercuryDPM. *Mathematics in Computer Science*, 17(2):13, 2023.

[6] I. Ostanin, V. Angelidakis, T. Plath, S. Pourandi, A. Thornton, and T. Weinhart. Rigid clumps in the MercuryDPM particle dynamics code. *Computer Physics Communications*, Under review, 2023.

[7] C. J. Coetzee. Review: Calibration of the discrete element method. *Powder Technology*, 310:104–142, 2017.

[8] C. J. Coetzee and D.N.J. Els. Calibration of granular material parameters for dem modelling and numerical verification by blade–granular material interaction. *Journal of Terramechanics*, 46(1):15–26, 2009.

[9] R. Fuchs, et al. Rolling, sliding and torsion of micron-sized silica particles: experimental, numerical and theoretical analysis. *Granular Matter*, 16(3):281–297, Jun 2014.

[10] T. A. H. Simons, et al. A ring shear tester as calibration experiment for dem simulations in agitated mixers – a sensitivity study. *Procedia Engineering*, 102:741–748, 2015. New Paradigm of Particle Science and Technology Proceedings of The 7th World Congress on Particle Technology.

[11] C. R. K. Windows-Yule, et al. Understanding and exploiting competing segregation mechanisms in horizontally rotated granular media. *New Journal of Physics*, 18(2):023013, 2016.

[12] T. Roessler and A. Katterfeld. Scaling of the angle of repose test and its influence on the calibration of DEM parameters using upscaled particles. *Powder technology*, 330:58–66, 2018.

[13] K. J. Åström and T. Hägglund. *PID controllers: theory, design, and tuning*. Instrument Society of America, Research Triangle Park, NC, USA, 1995.

[14] Q. H. Nguyen. Machine learning in the calibration process of discrete particle model, July 2022. https://purl.utwente.nl/essays/91991

[15] L., Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.