Ansys

CASE STUDY /

Calibration of Dem Simulations for Vertical Filling: How to Handle Randomness

At Robert Bosch Packaging Technology, optiSLang was used in conjunction with Rocky DEM to obtain accurate models for the simulation of vertical filling of granular foods.

/ Introduction

Vertical Filling

Vertical filling is a flexible process and commonly used in industrial packaging of granular foods, such as candy, snacks and bakery goods. The process is shown schematically in Figure 1. By increasing the frequency of drops of granulate portions, the output rate can be easily increased. However, the time distance between the portions must be kept large enough so that there is enough time to perform sealing. Otherwise, particles get caught between the sealing jaws, which often results in need for maintenance. Thus, compact falling of the portions is important for keeping the process reliable.

Discrete Element Method (DEM)

1. Overview

The Discrete Element Method (DEM) simplifies contacts by assuming particles to be stiff. Deformation is implemented by allowing a small overlap between particles. Contact forces are then calculated with simple relations to the current overlap. A variety of contact models are available in different DEM implementations. The model used here is the linear hysteresis model developed by Walton and Braun [1, 2] (Figure 2).

2. Model Calibration

Identifying model parameters for DEM simulations is challenging [4, 5]. An attractive and commonly used method is numerical model calibration, which consists of varying the model parameters while comparing the simulations to experimental results until reality is reproduced to a satisfactory extend. Calibration is usually performed in a relatively simple representative experiment [6]. A consecutive validation step can be then performed to verify if the model parameters hold up in the actual process of interest.

3. Solver Noise

Since granular systems are highly chaotic, small variations in initial conditions, such as the precise positions of individual particles in the collection bin before the drop [3], can dramatically affect the process outcome [7]. Physical randomness, just as process design, can be of great importance in achieving a desirable outcome and avoiding unfavorable ones. This is true for the physical process as well as for the simulations.

Goal

For this study, model parameters for a granular sample food had to be found. The chosen good was sugar-coated, bite-size chocolate candy with a porous cookie core. As calibration trial, a drop test that is very similar to the industrial process was used representing in-situ calibration [5]. Further, the necessity to incorporate the physical randomness in the DEM simulations and their effect on the calibration was evaluated. Finally, the methods were compared with regard to their feasibility, robustness and accuracy.

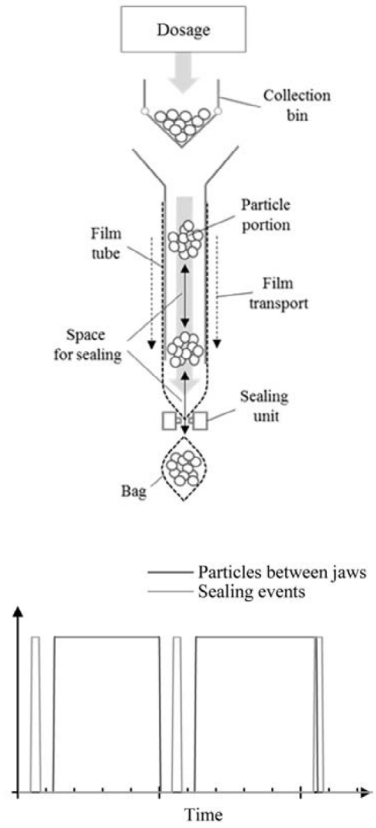

Figure 1. The vertical filling process. Schematic overview over process principle. Successful sealing (bottom left) and likely defect due to particles getting caught in the sealing unit (bottom right).

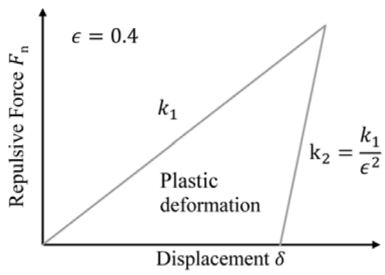

Figure 2. Relationship between particle e overlap and Force f_n for restitution coefficient ε=0.4 (from [3]).

/ Experiment

The drop setup has been described in [3] and is shown in Figure 3. Two rectangular falling tubes with different inner areas A□,1 and A□,2 where available. By varying the sample mass, a total of three scenarios were performed (Table 1).

The experiment was initiated by opening the flaps at the bottom of the sample container. The time stamps of the first and last particle leaving the tube at the bottom were recorded. The difference between these residence times Δt_{res} is equivalent to the portion range τ_{rg} discussed in [3].

$$\tau_{rg}=\Delta t_{res}=t_{res,lp}-t_{res,fp}$$

Then, the degree of filling a_p of the tube was tracked over time and normalized to the maximal possible value (entire tube filled). Figure 3 (see previous page) shows a frame cropped to the tube and the relative particle occupancy a_p plotted over time.

/ Simulation

Discrete Element Method

The experimental design (see chapter Experiment) was replicated with CAD tools and imported into the DEM environment. The pieces of candy were nearly spherical, so a spherical particle representation was chosen. The average sieve diameter of the particles was used as the sphere's diameter.

Young's modulus was chosen with regard to numerical criteria (computational cost and numerical stability) and left constant at 10^8 Pa [8]. The calibration parameters x (Table 2) were friction coefficients μ, respectively for the static (sticking) and the dynamic (sliding) case and the coefficients of restitution ε. Each parameter was assumed different for the interaction between the particles (P-P) and the interaction between particles and the boundary (P-B). Additionally, a factor for rolling resistance was calibrated to account for the increased rolling of spherical particles compared to the real particles [9]. The eventual model parameters x differ from the "true" physical parameters due to model shortcomings [10, 11].

Calibration

The goal of model calibration is to identify the parameter set x that produces the best match between simulations w and the experimental results u. For the drop test, we aim to accurately predict the portion range t_{rg} from the experiment. This goal can be formulated as an optimization problem, where the error between the simulation and the experiment has to be minimized. Several optimization strategies have been used for DEM model calibration, such as manual comparison [12], gradient-based methods [13], genetic algorithms [14] and Artificial Neural Networks [4]. A recently followed approach is to create a metamodel with a kriging algorithm from several anchor points in the parameter space and perform the optimization on the resulting surrogate model [15].

The benefit of the latter method is that the number of solver runs can be reduced and evaluation of the goal function on the surrogate model is quick.

The procedure was implemented in an automated calibration workflow (Figure 4) in optiSLang. The DEM solver was called at different parameter sets (samples) and the results were compared to the experimental data. The data was then processed into a metamodel of the solver behavior.

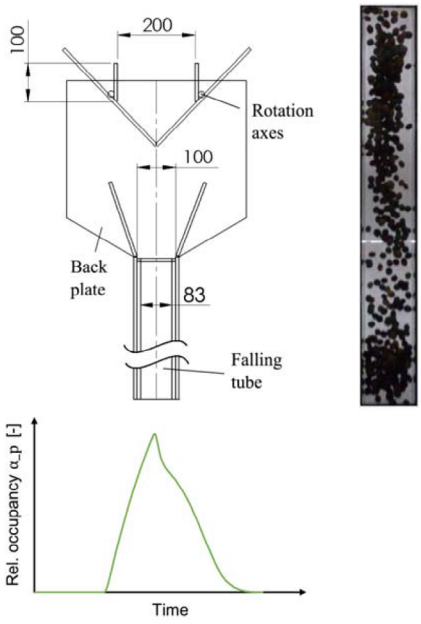

Figure 3. Drop setup described in [3] and snapshot of drop test. Measures in mm.

Sample mass	Inner tube area	Used for
500g	A□,1=76 cm²	Calibration
	A□,2=100 cm²	Validation
700g	A□,1=76 cm²	Validation

Table 1. Scenarios of the drop test.

Parameter	Material and Scenario		Symbol
Friction	Particles – Particles	Static	$x1 = \mu_{s,P-P}$
		Dynamic	$x2 = \mu_{d,P-P}$
	Boundary – Particles	Static	$x3 = \mu_{s,P-B}$
		Dynamic	$x4 = \mu_{d,P-B}$
Restitution	Particles – Particles		$x5 = \epsilon_{P-P}$
	Boundary – Particles		$x6 = \epsilon_{P-B}$
Rolling Resistance	Particles		$x7 = RR$

Table 2. Calibration parameters.

1. Metamodeling

Kleijnen [16] gives a comprehensive theoretical overview over metamodeling methods, so we will use part of his nomenclature here. The solver output w has to be approximated by the output ŵ of the metamodel f_{meta}.

$w = f_{sim}(x,r) = f_{meta}(x) + e$

f_{sim} is the noisy solver function which depends on the calibration parameters x and the seed of the random number generator r. The metamodel function is f_{meta} with its value depending only on the calibration parameters x. e is the residual vector, in which the local error of the metamodel at anchor point i is

$e_i = \hat{w} - w_i$

If we make the assumption that the kriging algorithm is capable of describing the behavior of a deterministic solver $f_{sim}(x)$, there must be a kriging parameter set β which provides optimal fidelity. However, we must keep in mind that we only have a finite amount of anchor points n to work with, so we can only find an estimate $\hat{\beta}$ of β. [16]

In the case of a noisy solver $f_{sim}(z,r)$, the regression will smooth out some of the solver noise [17, 16], while producing greater residuals than in the deterministic case. This does not imply bad quality of the metamodel but rather highlights the deterministic nature of f_{meta}. The criteria after how many simulation runs the metamodel should be finalized is not obvious here. A possible criterion is to track the mean residuals over the number of anchor points n and stop the process when stagnation is reached. It is however not guaranteed that this point will coincide with an acceptable quality of $\hat{\beta}$.

2. Adaptive Sampling

Choosing the anchor points with Latin Hypercube sampling (LHS) [18, 19] allows a sufficient coverage of the parameter space, while avoiding undesired sampling effects at a smaller number of anchor points [20]. However, DEM simulations are computationally expensive, so adaptive sampling, similar to [21], was performed to further reduce the number of solver calls.

The general topology (i.e. global trends) of the metamodel can be estimated quite well in an exploration phase with relatively coarse sampling. In a refinement iteration, we can add anchor points in the interesting regions of the metamodel (i.e. where the predicted error Δt_{rg} between simulation w and experiment u is low) and recalculate the metamodel. With the refined information on promising zones, we can then repeat the refinement for several iterations until stagnation is reached or the maximum computation budget is spent.

3. Optimization

Kriging models are smooth. Therefore, fast gradient based approaches can be used for optimization [21, 22]. The implementation of the Lagrangian NLPQL solver of optiSLang was used due to its numerical performance and accuracy [23, 24].

Validation

There are two sources for errors in the calibration process: numerical (insufficient metamodel quality) or systematic measurement errors and shortcomings in the DEM model. To exclude both, two separate validation steps were performed.

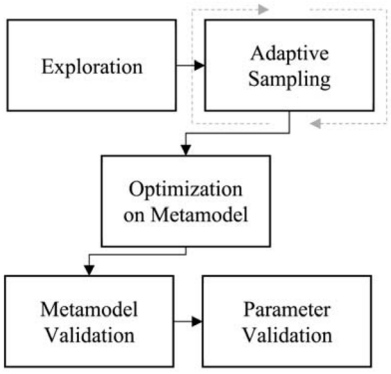

Figure 4. Calibration Workflow in optiSLang.

	Iteration 1 (Exploration)	Iteration 3	Iteration 10	Iteration 20	Average per Iteration*
n_{rnd}	290	379	693	1162	46
cost	11.8 h	15.4 h	28.2 h	47.3 h	1.9 h
n_{cnst}	289	378	698	1143	45
cost	9.1 h	11.5 h	21.3 h	34.9 h	1.4 h

Table 3. Number of anchor points n and total computational cost of the calibration in the drop test at different iterations | *after iteration 1.

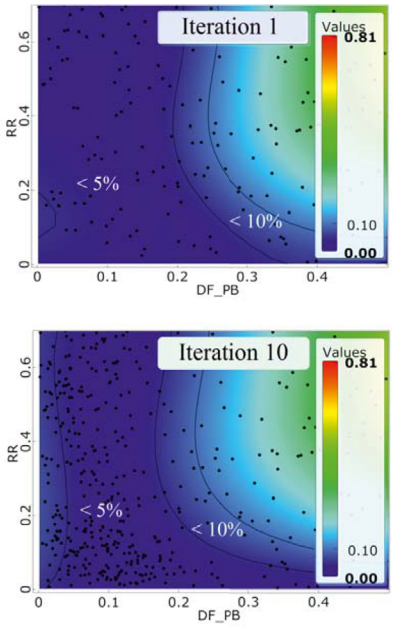

Figure 5. 2-dimensional projection of the 7-dimensional metamodel for Δt_{rg} in % in relation to the two most influential parameters (RR and μd,P-B) at iteration 1 (Exploration) and 10 (RIC).

1. Metamodel Validation

In order to ensure the prediction capability of the metamodel, a set of m validation simulation runs were performed at the supposed minimum x_{opt} and their results w1, w2 ...,w_m were averaged. The difference $e_{opt} = \hat{w}_{opt} - \overline{w}_{opt}$ is a teller for the reliability of the metamodel at that point. If the error is unacceptably high, more anchor points should be added to increase the accuracy of $\hat{\beta}$.

2. Parameter Validation

To verify that the obtained parameter set x_{opt} was viable outside the calibration scenario, validation simulations were performed in the respective scenarios shown in Table 1 (see p. 7) The results were obtained from m averaged simulation runs.

3. Randomness

In real life, the filling of the containers is a random process that cannot be reproduced in the next run, resulting in a partially random initial condition (RIC) of the bulk. This randomness is a physical property of the processes, influencing the outcome of the experiment.

The simulations were designed to account for that randomness, so a random and flat particle bed was created in the simulations before release. This added computational cost of around 37 seconds to the runtime of 110 seconds per run on average (34%). Furthermore, the RIC increases solver noise.

Both increased cost and solver noise are undesirable from an engineering standpoint, while it is unclear if the physical randomness actually plays a significant role and if the additional effort is justified. In order to determine whether the implementation of the physical randomness is actually necessary, we also performed the calibration with an arbitrary but constant initial state (CIC).

/ Results

Table 3 shows the number of anchor points (simulated parameter sets) over the iterations. Figure 5 (see next page) shows a projection of a graphical representation of the metamodel after different iterations 1 and 10. The parameters found to be the most influential on the portion range t_{rg} were μd,p−b and RR. All other parameters are held constant near their respective optimum for low DEM model error. We observe only a slight change in the topology of the metamodel between Iteration 1 and 10. This suggests that the sampling could be stopped after iteration 1.

However, to gain insight into the quality of the prediction of the metamodel, we also must assess the residuals e of Δt_{rg}. Figure 6 shows the local residuals e of the metamodel in the same range as Figure 5. We find that uncertainty is quite high at iteration 1, especially in the area of low predicted errors Δt_{rg}. This implies a bad estimate . After increasing the number of anchor points to more than twice the original count, at iteration 10, residuals were significantly lower, especially in the interesting areas of the metamodel.

Figure 7 shows the relationship between the residuals e in regions of low predicted errors Δt_{rg} and iteration number for the entire parameter space. Stagnation begins after iteration 3, which suggests that adding samples does not improve the metamodel anymore [16].

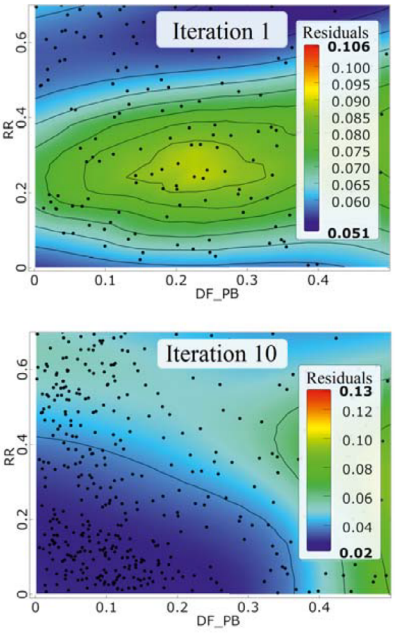

Figure 6. 2-dimensional projection of the local residuals e of Δt_{rg} for iteration 1 (exploration) and iteration 10 (RIC).

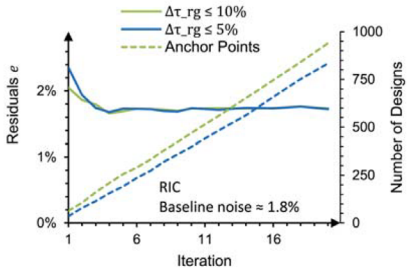

Figure 7. Residuals e of t_{rg} for the areas of the metamodel with low predicted DEM model errors Δt_{rg} over iterations (RIC). The respective number of iterations used to calculate the residuals are shown as dotted lines.

In the next step, the minimum error min (Δt_{rg}) was determined on the metamodel with the NLPQL optimizer. The runtime was < 1 min. The metamodel was then validated at the supposed minimum x_{opt} according to Table 1, showing a very good match (Figure 8). This confirms that the metamodel is indeed of high quality.

The optimized parameter set x_{opt} was then used for the two validation trials as shown in Table 1. The results are shown in Figure 8. We find that the calibrated model exhibits a high fidelity in reproducing the experimental results. An overview over the accuracy of the DEM models is presented in Table 4.

The entire calibration process was repeated with a constant initial condition (CIC) before the drop. The results are shown in Figure 8. We obtain a nearly equally good result as in the case with the RIC. This suggests that the physical randomness was not crucial for the accuracy of the metamodel. This however could only be true for the particular initial condition chosen here.

/ Conclusion

We found that the selected drop test is a suitable experimental approach for DEM model calibration, yielding low prediction errors of a maximum of 2%. The calibration was repeated without physical noise, which yielded an equally good result. This suggests the conclusion that physical noise is not relevant for the calibration, however it still needs to be proven whether this is true for all initial conditions or only some.

/ Author

St. Kirsch (Robert Bosch Packaging Technology B.V.)

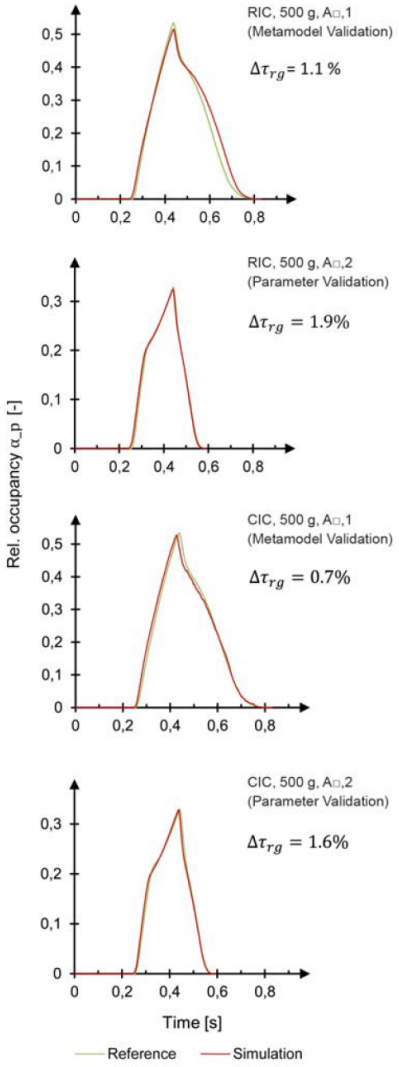

Figure 8. Results of calibration in the drop test (RIC and CIC), validation of metamodel after optimization and parameter validation in the drop test with A□,2 > A□,1.

m=20	Metamodel Validation	Parameter Validation		
		500 g, A□,1	500 g, A□,2	700 g, A□,1
Drop, RIC	$\Delta\tau_{rg}$=1.1%	-	$\Delta\tau_{rg}$=1.9%	$\Delta\tau_{rg}$<0.1%
Drop, CIC	$\Delta\tau_{rg}$=0.7%	-	$\Delta\tau_{rg}$=1.6%	$\Delta\tau_{rg}$=0.7%

Table 4. Actual error of t_{rg} for metamodel validation and for the eventual calibrated parameters x for the random initial condition (RIC) and constant initial condition (CIC). Simulations were performed m=20 times and their results averaged.

/ References

[1] O. R. Walton et al., “Viscosity, granular temperature, and stress calculations for shearing assemblies of inelastic, frictional disks”, Journal of Rheology, 1986.

[2] S. Luding, “Collisions & Contacts Between Two Particles”, Physics of Dry Granular Media, vol. 350, pp. 285-304, 1998.

[3] S. Kirsch et al., “Simulation of Vertical Filling Processes of Granular Foods for typical Retail Amounts”, in 9th Conference Processing Machines and Packaging Technology, 2018.

[4] L. Benvenuti et al., “Identification of DEM simulation parameters by Artificial Neural Networks and bulk experiments”, Powder Technology, vol. 291, pp. 456-465, 04 2016.

[5] C. J. Coetzee, “Review: Calibration of the discrete element method”, Powder Technology, vol. 310, pp. 104-142, 04 2017.

[6] T. Gröger et al., “On the numerical calibration of discrete element models for the simulation of bulk solids”, in 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering, Elsevier BV, 2006, pp. 533-538.

[7] T. Shinbrot, “Granular chaos and mixing: Whirled in a grain of sand”, Chaos, vol. 25, 09 2015.

[8] Y. Xu, et al., “Effects of material properties on granular flow in a silo using DEM simulation”, Particulate Science and Technology, vol. 20, pp. 109-124, 04 2002.

[9] D. Markauskas et al., “Investigation of rice grain flow by multi-sphere particle model with rolling resistance”, Granular Matter, vol. 13, pp. 143-148, 07 2010.

[10] C. González-Montellano et al., “Determination of the mechanical properties of maize grains and olives required for use in DEM simulations”, Journal of Food Engineering, vol. 111, pp. 553-562, 08 2012.

[11] X. Huang, “Exploring critical-state behaviour using DEM”, 2014.

[12] M. Frank et al., “Simulation-based optimization of geometry and motion of a vertical tubular bag machine”, Sächsische Landesbibliothek, 03 2016.

[13] M. W. Johnstone, “Calibration of DEM models for granular materials using bulk physical tests”, 2010.

[14] H. Q. Do et al., “A calibration framework for discrete element model parameters using genetic algorithms”, Advanced Powder Technology, vol. 29, pp. 1393-1403, 06 2018.

[15] M. Rackl et al., “A methodical calibration procedure for discrete element models”, Powder Technology, vol. 307, 02 2017.

[16] J. P. C. Kleijnen, “Simulation Optimization through Regression or Kriging Metamodels”, Discussion Paper, 05 2017.

[17] T. Most et al., “Sensitivity analysis using the Metamodel of Optimal Prognosis”, in Weimar Optimization and Stochastic Days, 2011.

[18] M. D. McKay et al., “A Comparison of three methods for selecting values of input variables in the analysis of output from a computer code”, Technometrics, vol. 21, pp. 239-245, 05 1979.

[19] J. C. Helton et al., “Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems”, Reliability Engineering and System Safety, vol. 81, pp. 23-69, 2003.

[20] M. Rackl et al., “Efficient calibration of discrete element material model parameters using Latin hypercube sampling and kriging”, in ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering, 2016.

[21] R. K. Rathore et al., “An Adaptive Approach for Single Objective Optimization”, Int. Journal of Engineering Research and Applications, vol. 4, pp. 737-746, 02 2014.

[22] F. Jurecka, “Automated metamodeling for efficient multi-disciplinary optimisation of complex automotive structures”, in 7th European LS-DYNA Conference, Salzburg, Austria, 2009.

[23] K. Schittkowski, “NLPQL: A fortran subroutine solving constrained nonlinear programming problems”, Annals of Operations Research, vol. 5, pp. 485-500, Jun 1986.

[24] Y.-h. Dai et al., “A Sequential Quadratic Programming Algorithm with Non-Monotone Line Search”, Pacific Journal of Optimization, vol. 4, 2008.

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
U.S.A.
724.746.3304
ansysinfo@ansys.com

If you've ever seen a rocket launch, flown on an airplane, driven a car, used a computer, touched a mobile device, crossed a bridge or put on wearable technology, chances are you've used a product where Ansys software played a critical role in its creation. Ansys is the global leader in engineering simulation. We help the world's most innovative companies deliver radically better products to their customers. By offering the best and broadest portfolio of engineering simulation software, we help them solve the most complex design challenges and engineer products limited only by imagination.

Visit www.ansys.com for more information.