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 Fig. 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 is available in different DEM implementations. The model used here is the linear hysteresis model developed by Walton and Braun [1, 2] (Fig. 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.

Experiment

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

<table>
<thead>
<tr>
<th>Sample mass</th>
<th>Inner tube area</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>500g</td>
<td>$A_{1}=76 \text{ cm}^2$</td>
<td>Calibration</td>
</tr>
<tr>
<td>700g</td>
<td>$A_{2}=100 \text{ cm}^2$</td>
<td>Validation</td>
</tr>
</tbody>
</table>

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.
Table 2: Calibration parameters recorded. The difference between these residence times set

The goal of model calibration is to identify the parameter calibration in the drop test at different iterations | *after iteration 1

The procedure was implemented in an automated calibration workflow (Fig. 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.

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 10² GPa [8]. The calibration parameters x (Table 2) were friction coefficients μ, respectively for the static (sticking) and the dynamic (sliding) case and the coefficient of restitution e. 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 parameter of 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, 13].

Parameter | Material and Scenario | Symbol
--- | --- | ---
Friction | Particles – Particles | Static | \( x_1 = \mu_{s,p-p} \)
| | Dynamic | \( x_2 = \mu_{d,p-p} \)
Restitution | Particles – Particles | Static | \( x_3 = e_{s,p-p} \)
| | Dynamic | \( x_4 = e_{d,p-p} \)
Rolling Resistance | Particles | Static | \( x_5 = \beta_{p} \)
| | Dynamic | \( x_6 = \psi \)

Table 2: Calibration parameters

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 [21].

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.

1. Metamodeling

Kleijnen [26] 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 \( \hat{w} \) of the metamodel \( f_{meta} \)

\[
w = f_{sim}(x,r) = f_{meta}(x) + \epsilon
\]

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 an kriging parameter set \( \beta \) which provides optimal fidelity. However, we must keep in mind, that we only have a finite amount of anchor points x to work with, so we can only find an estimate of \( \beta \) [16].

In the case of a noisy solver \( f_{sim}(x,r) \), 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 simulations run the metamodel will be finalized is not obvious here. A possible criterion is to track the mean residuals over the number of anchor points and stop when stagnation is reached. It is however not guaranteed that this point will coincide with an acceptable quality of \( \beta \).

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 \( \Delta \tau_{rg} \) 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 NLQPSolver of optiSlang was used due to its numerical performance and accuracy [23, 24].

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.

1. Metamodel Validation

In order to ensure the prediction capability of the metamodel, a set of validation simulation runs were performed at the supposed minimum \( x_{opt} \) and their results \( w_1, w_2 ... w_n \) were averaged. The difference \( e_{opt} = |w_{opt} - w_{opt}| \) 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 \( \beta \).

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 2 (see p. 7). The results were obtained from w 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 (RK) 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. Fig. 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 \( \mu_{d,p-p} \) and \( \beta \). All other parameters are held constant near their respective optimum for low DEM model errors. 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 $\Delta \tau_{rg}$. Fig. 6 shows the local residuals $\epsilon$ of the metamodel in the same range as Fig. 5. We find that uncertainty is quite high at iteration 1, especially in the area of low predicted errors $\Delta \tau_{rg}$. This implies a bad estimate $\hat{\theta}$. After increasing the number of anchor points to more than twice the original count, at iteration 2 anchor points were significantly lower, especially in the interesting areas of the metamodel. Fig. 7 shows the relationship between the residuals $\epsilon$ in regions of low predicted errors $\Delta \tau_{rg}$ 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].

In the next step the minimum error min $\Delta \tau_{rg}$ was determined on the metamodel with the NLQL optimizer. The runtime was $1 \text{ min}$. The metamodel was then validated at the supposed minimum $\Delta \tau_{rg}$ according to Table 1, showing a very good match (Fig. 8). This confirms that the metamodel is indeed of high quality.

The optimized parameter set $\hat{\theta}_{opt}$ was then used for the validation of the entire model. The results are shown in Fig. 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 Fig. 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.