Application of Powder Simulation to Powder Metallurgy

Jun OTSUKA*, Tatsuya SAITO, and Hidehiko MISHIMA

In order to meet the demand for high dimensional accuracy in powder metallurgy products, we have been developing powder simulation techniques to support the design of the production process. Powder simulations provide a macroscopic interpretation of powder behavior by calculating the motion of individual particles that constitute the powder. In this paper, the powder simulation is applied to the milling process of raw powder and the powder feeding process of metal powder. Although the milling efficiency can be predicted from the collision energy of the balls in a ball mill, it is necessary to reproduce the motion of each ball with complicated collision process to predict the collision energy accurately. In this study, we have developed an analysis technique to predict the change of the collision energy when the milling conditions are changed, in order to improve the performance of the milling process. We have also developed an analytical technique to visualize the powder behavior in the powder feeding process to clarify the mechanism of the filling variation in the die.

Keywords: powder metallurgy, ball mill, powder feeding, discrete element method, supercomputer

1. Introduction

Powder metallurgy is a technique used to manufacture products by placing metal powder in a mold for compression and consolidation, followed by sintering at high temperature. Since it cuts machining processes substantially and enables mass production of products with high dimensional accuracy, it is a critical technique in the formed and fabricated material industry. In the design of powder metallurgy materials, achieving a uniform particle size distribution of metal powders is specifically necessary. For example, this is a challenge when making highly strong and tenacious cemented carbide cutting tools. Therefore, it is important to control particle size in the raw powder material milling process. Moreover, even if a metal powder has been prepared as intended in a material design, it is impossible to impart the intended functionality or reliability unless a stable manufacturing process is established. Therefore, important factors also include the process of filling a mold with a metal powder, mold design, and pressing conditions. However, to optimize manufacturing conditions through experiments, large expenses and a large amount of time are required for trial and error. As a solution to this challenge, we promote the use of simulations. This paper reports on examples of the use of simulations modeling raw powder material milling and metal powder feeding processes.

The discrete element method (DEM)[1] is a promising technique of simulating powder behavior. The technique tracks the motions of individual particles constituting a powder and represents powder behavior accurately.[2] Figure 1 shows the algorithm of this computation technique. DEM predicts motions of all particles according to acting forces and based on Newton's equations of motion. It often involves particle approximation with spheres for computation purposes, to simply calculate repulsions taking place during collisions between particles. Although it uses a simple model, DEM is potentially usable for an extensive range of applications. We have analyzed the applicability of DEM to a number of powder metallurgy processes and tested simulations of processes such as feeding,[3] milling,[4],[5] blending, and powder feeding.

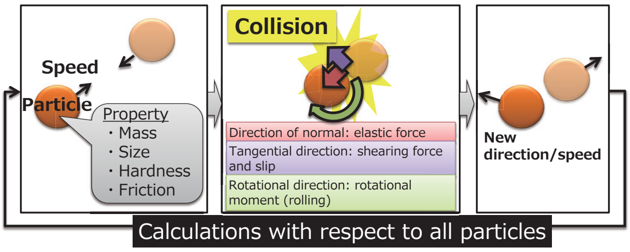

Speed
Particle
Property
· Mass
· Size
· Hardness
· Friction
Collision
Direction of normal: elastic force
Tangential direction: shearing force and slip
Rotational direction: rotational moment (rolling)
New direction/speed
Calculations with respect to all particles

Fig. 1. DEM-based simulation algorithm for powder particle behavior[11]

A major challenge associated with the application of DEM to actual processes is the magnitude of calculation cost arising from the enormous number of particles involved. One example is the use of an actual particle size for simulation when analyzing powder behavior that occurs during a powder feeding process. The immense number of particles would make it impossible to complete the computation within a realistic amount of time even with a computer built to the latest specifications. As a solution to this difficulty, one common practice is using a larger-than-actual particle size, which is known as a coarse-grained model.[6] It is a method of reducing the calculation cost based, for example, on an idea that the number of particles can be reduced to one-eighth by doubling the particle size without altering the volumetric sum of all particles constituting the powder. However, the powder container and mold have an influence on powder behavior, which is termed *wall effects*.[7] Therefore, in powder feeding simulations, a substantial change in the dimensional ratio of the particles to the powder container or mold would make it impossible to accurately express powder behavior. For this reason, it is unavoidable to use the immense number of particles for computation when simulating an actual powder container and mold. Consequently, we used the open-source software “LAMMPS improved for general granular and granular heat transfer simulations (LIGGGHTS*1).”[8]-[10]

LIGGGHTS enables the engineer to use multiple CPUs for shared particle calculations, which is known as parallel computation. It can perform computation of an immense number of particles commensurate with the number of CPUs.

This paper contains texts and figures cited from “Application of Discrete Element Method to Powder Metallurgy Using Open Source Software”(11) on pp. 607–610, No. 7, Vol. 84, of the Journal of JSPE with the permission of the copyright holder Japan Society for Precision Engineering.

2. Example Ⅰ : Simulation of Milling

2-1 Milling performance prediction method

The main raw materials of cemented carbide are powder carbides and metal powders, which are milled in a wet ball mill. Ball mills are designed to place slurry (a mixture of powder and liquid) and balls in a container and rotate to mill the raw materials through mechanical forces produced by collisions between balls. To achieve a desired particle size distribution through milling, milling conditions need to be optimized. Parameters such as ball conditions (size, amount of charge, and hardness), mill speed, and milling duration should be taken into consideration. This section presents examples of analyzing the effects of varying ball sizes and mill speeds on milling performance.

Figure 2 illustrates a ball mill, of which structures relating to slurry inflow and discharge were omitted for ease of understanding. The ball mill container contains balls and the material to be milled. The number of collisions between balls is far greater than the number of collisions between the ball and the wall of the container. Therefore, milling performance can be predicted by evaluating the energy of ball-to-ball collisions.(12),(13) The following equation is used to calculate the collision energy E_w:

$$E_w = \frac{1}{W}\sum_{j=1}^{n}\frac{1}{2}mv_j^2 \quad \cdots\cdots (1)$$

where W is the weight of the material to be milled, n is the number of collisions, m is the mass of a ball, and v_j is the relative speed of the ball at the time of collision with respect to another ball or a wall. The motion of the ball is resisted by the presence of the material to be milled. However, directly factoring the material to be milled into calculation is difficult due to the amount of calculation time; therefore, instead, a technique was devised to use friction for approximation of the motion of the ball hampered by the material to be milled. Using the actual number of balls for the simulation was beyond the capacity of the analysis server owned in house. Therefore, the simulation used the large computer system of the Cybermedia Center of Osaka University.

Milling performance is predicted by checking collision energy analysis results against milling efficiency measurement results. The first step is to analyze collision energy using existing milling condition measurements to ascertain correlations. Then, collision energy is analyzed with respect to the milling conditions for which milling performance is to be predicted. Milling efficiency can be predicted through interpolation or extrapolation based on the abovementioned correlations.(12),(13)

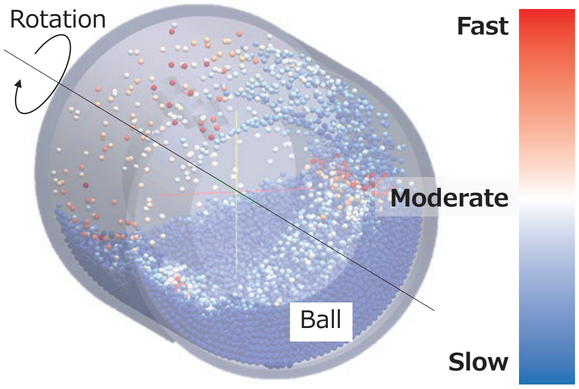


Fig. 2. Ball mill simulation; balls are colored according to speeds(11)

2-2 Milling analysis execution environment

The performance of LIGGGHTS (version 2.2.4) was lower than the execution performance of general applications, as revealed through the computation of the original source code on a supercomputer. Therefore, we improved the performance by revising the source code without making a major change. First, using a profiler function, the execution time of each function was measured to identify bottlenecks that required a large amount of execution time. Next, to eliminate the bottlenecks, the computational processes of the functions that had some problems were rewritten into a form tailored for ease of computation by the supercomputer. For example, repetitive loops that included plural computational processes were divided for simplification through individual loop configurations. As a result, a 3.4-fold speed improvement was achieved,(5) enabling large models to be computed to handle a desired number of balls.

2-3 Fluid resistance modeling

In a wet ball mill, the kinetic energy of the motions of particles decreases due to fluid resistance exerted by slurry. A method has been reported(14) to express this interaction with high accuracy, which is to calculate the slurry flow by means of numerical fluid analysis and couple it with a DEM simulation. However, many balls and internal structures, including the inflow and discharge ports, are present in the mill. Therefore, the high magnitude of computational load makes it impossible to consider an actual mill structure in a fluid analysis. As a solution, we adopted a technique that incorporated approximate representation of the effect of slurry flow into the calculation. This technique takes into account the buoyant force F_B exerted by the slurry onto the ball and the fluid resistance F_D, as external forces, by determining the slurry flow field through fluid analysis.(12) The buoyant force F_B and the fluid resistance F_D are expressed by the following equations:

$$F_B = gV_B\rho_s \quad \cdots\cdots (2)$$

$$F_D = C_d A \rho_s \frac{u_r^2}{2} \quad \cdots\cdots (3)$$

where V_B is the ball volume, C_d is a coefficient of resistance, A is the projected area, ρ_s is the slurry density, and u_r is the relative speed of the fluid and ball. It was necessary to assign a force to the fluidic drag expressed by Equation (3), based on an accurate flow rate distribution. However, LIGGGHTS was designed to assign a uniform force. Under this condition, it would be impossible to express the distribution of flow rates that differed according to the locations in the mill. As a solution, we changed the source code to incorporate a flow rate distribution and implemented external force calculation processes expressed by Equations (2) and (3).

2-4 Milling analysis results

This section first describes the distribution of ball-to-ball collision energy in the mill, evaluation of which became possible through the ball mill simulation. Next, this section presents examples of application of the simulation to the study of effects of different mill conditions. The examples show the results of collision frequency evaluation as the effects of varying rotational speeds of the mill and ball sizes. This paper refrains from predicting milling performance.

The mill was divided into five regions appropriately in the longitudinal and radial directions. Figure 3 presents the results of a collision energy analysis conducted on balls present in each region. First, fluid resistance was ignored to understand the extent of effects of the mill geometry on collision energy distribution. In this case, collision energy differed from region to region. This suggests that, in some regions of the mill, balls moved with relative ease, and in other regions, they did not. Next, with fluid resistance taken into account, collision energy decreased. Moreover, the amount of decrease in collision energy caused by fluid resistance differed from region to region in the mill.

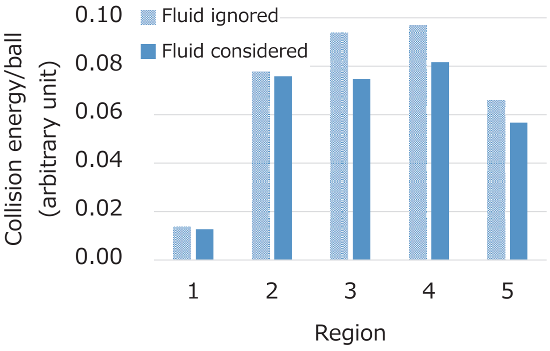


Fig. 3. Ball collision energy[11]

Next, collision frequencies at varying rotational speeds of the mill and ball sizes were analyzed, which are shown in Fig. 4. The results reveal that, at higher rotational speeds, the number of collisions was proportional to the ball size, while, at lower rotational speeds, the relationship between the number of collisions and ball size deviated from simple proportionality. This is inferred that small balls enter the internal structures of the mill, resulting in complex ball motions.

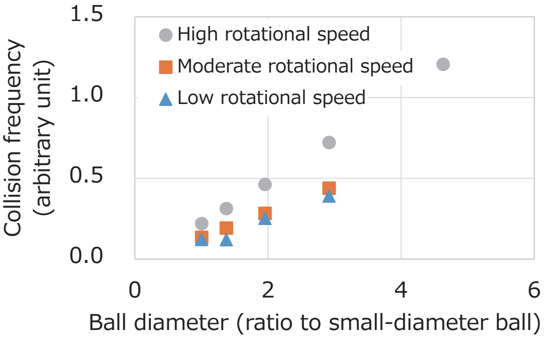


Fig. 4. Relationship between the number of collisions and rotational speed[11]

The correlations were investigated between collision energy determined through analysis and milling efficiency determined based on measurements. The investigation results revealed a trend similar to the relationships observed with wet mills reported by Gudin et al.[15] This proved the viability of predicting milling performance considering the features of wet mills even in the complex mill structure that we studied. The analysis results will be used to optimize milling conditions in the future.

3. Example Ⅱ: Powder Feeding Simulation

3-1 Method of visualizing powder feeding

In the powder feeding process, a powder container moves back and forth over a mold, and the cavity (concave portions of the mold) is filled with the metal powder dropping due to gravity. For improved dimensional accuracy, ensuring that the compact (compression-molded metal powder in a mold) has a uniform density is important. Consequently, filling performance is a factor as important as mold geometry, pressing conditions, and lubricant. One challenge associated with improving filling performance is to reduce weight variation of the powder filled into the cavity. However, factors involved in the variation are not limited to powder container motion and other powder feeding conditions. Environmental factors also have substantial effects. Accordingly, it is difficult to conduct experiments with a high level of reproducibility. Optimizing experimental powder feeding conditions is not an easy task. As a means of optimizing powder feeding conditions, we have developed an analysis technique intended to make visible powder conditions in powder container and cavity through DEM simulations.

A DEM simulation-based analysis of particle behavior during a process of entry from a powder container into a cavity produces location, speed, and exerted force information for all particles. Visualization software applied to displaying constantly varying locations of particles can create an animated film of powder behavior. By coloring particles with respect to a physical quantity of interest, it becomes possible to have a detailed understanding of changes in the state of particles in a powder feeding process.

3-2 Powder feeding analysis execution environment

To improve filling performance, performing many powder feeding simulations using different powder feeding conditions, such as the motion and geometry of the powder container, is necessary. To accelerate analysis work, we made analysis processes automatically executable.

The powder feeding process analysis flow comprised of the following three steps: i) DEM simulation condition setting, ii) DEM computation, and iii) evaluation and visualization of computation results. At first, the person in charge of analysis processed these steps for each condition, in which the speed of the analysis flow was limited by human capability. To make the flow automatic, Python, a programming language, was used. In step i), rewriting input files was made possible by character string replacement available in Python. In step ii), the computation was implemented by executing Linux instructions in Python. In step iii), for visualization, the visualization software ParaView*2 (16) was used, and scripts that rendered the visualization process were created in Python. Automatic processing was realized by connecting these processes and executing successively by means of looping in Python. Thus, constructing a powder feeding analysis system while reducing the number of development man-hours, using an easy-to-learn programing language and existing visualization software, became possible.

3-3 Powder feeding process analysis results

By applying DEM to a powder feeding process, it became possible to make powder behavior in powder feeding process visible, as illustrated in Fig. 5. The powder shown in Fig. 5 is colored according to elevations observed before filling. Coloring enables one to know where the particles filling the cavity were in the powder container. The analysis results revealed that particles at the lower front of the powder container filled the front portion of the cavity, while particles at the upper front of the powder filled the back portion of the cavity. This behavior is consistent with the results observed with an actual machine, proving success in powder behavior visualization. The analysis results will be used in the future to optimize the powder feeding process for improved filling performance.

4. Conclusion

This paper narrated powder simulation application examples, performing computation pertaining to a ball mill and powder feeding process. In the ball mill simulation, evaluating milling performance with good accuracy, using a supercomputer to enable analysis of an enormous number of balls and considering the effects of fluid on ball motion, became possible. Regarding the powder feeding simulation, this paper presented the results of making powder feeding behavior visible, after describing automation of analysis work. A future task is to improve analysis accuracy taking environmental conditions into consideration and to optimize manufacturing processes with the help of powder simulation as a virtual experiment tool. We intend to build a system for proposing optimal process conditions by developing a database of simulation results and utilizing artificial intelligence (AI) technology.

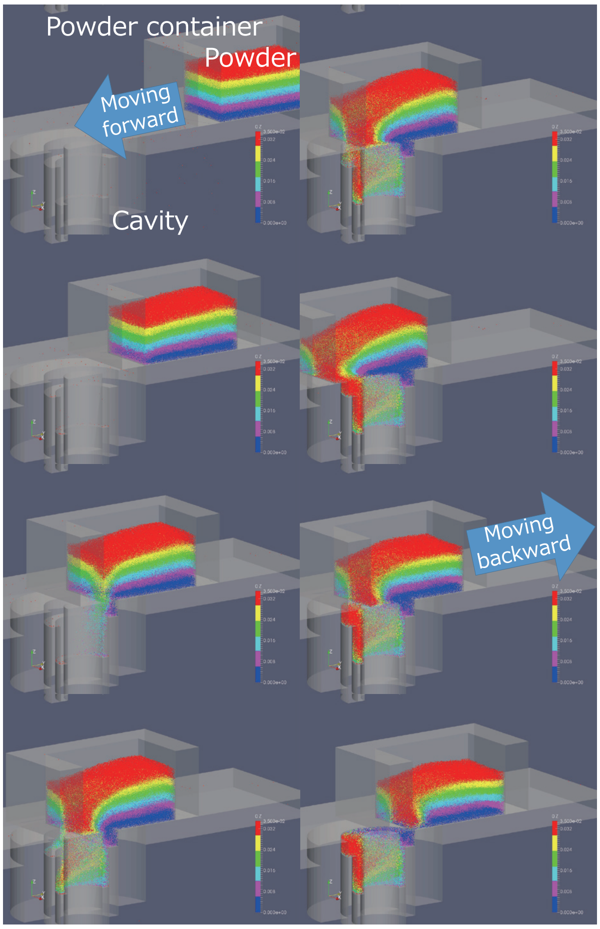


Fig. 5. Powder feeding simulation results. Coloring of particles indicates their elevations observed before filling(11)

5. Acknowledgments

Part of the research achievements presented in this paper was produced using the supercomputer SX-ACE of the Cybermedia Center of Osaka University.

Technical Terms

∗1 LIGGGHTS: Open-source discrete element method software. LIGGGHTS is distributed in accordance with the GNU General Public License.

∗2 ParaView: Open-source data visualization software.

References

(1) P. A. Cundall, O. D. L. Strack, A discrete numerical model for granular assemblies, Géotechnique, **29** (1979) 47

(2) The Society of Powder Technology, Japan (Ed.), Funtai shimyurēshon nyuumon: Konpyūta de fun Tai gijutsu o souzousuru, (1998) Sangyo Tosho

(3) J. Otsuka, Application of OpenMP Parallelism to Powder and Granular Simulations Using Discrete Element Method, HPCS2012, Presentation materials

(4) J. Otsuka, Prediction of Grindability of Ball Mill Using Distinct Element Method Simulator LIGGGHTS, Open CAE Symposium 2014, Presentation materials

(5) J. Otsuka, Large scale simulation of ball mil by particle-fluid coupled analysis, HPCI User Report, hp140085 (2015)

(6) M. Sakai et al, Verification and validation of a coarse grain model of the DEM in a bubbling fluidized bed, Chemical Engineering Journal, **244**, 15 (2014) 33

(7) W. A. Gray, The Packing of Solid Particles, Chapman and Hall (1968) 71

(8) C. Kloss, C. Goniva, A. Hager, S. Amberger, and S. Pirker, Models, algorithms and validation for opensource DEM and CFD-DEM, Progress in Computational Fluid Dynamics, **12**, Nos. 2/3, (2012) 140–152

(9) S. Chikashi, Proceedings of the Design Engineering Society of Japan, 2016, fall season, 175.

(10) S. Yamanaka, S. Yoshida, T. Yamada, Numerical simulation of die filling of metal powder using discrete element method, Proceedings of the Conference on Computational Engineering and Science 19, 3p, 2014-06

(11) J. Otsuka, T Saito, Application of Discrete Element Method to Powder Metallurgy with Open Source Software, JSPE, 84 (2018) 607

(12) R. Soda, J. Kano, F. Saito, Analysis of Effect of Pin Configuration on Beads Motion in a Stirred Mill by DEM, J. Soc. Powder Technol, Japan, **46** (2009) 180

(13) M. Kimura, M. Narumi, T. Kobayashi, Design Method of Ball Mill by Discrete Element Method, SUMITOMO KAGAKU (2007) 31

(14) Y. Tsuji, T. Kawaguchi, T. Tanaka, Discrete particle simulation of two-dimensional fluidized bed, Powder Technol., **77** (1993) 79

(15) D. Gudin, J. Kano, F. Saito, Correlation between the Grinding Rate Constant and the Impact Energy of Beads during Wet Bead Milling, J. Chem. Eng. Jpn., **40** (2007) 980

(16) A. Henderson: ParaView Guide A Parallel Visualization Application. Kitware Inc., (2007)

Contributors

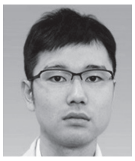
The lead author is indicated by an asterisk (∗).

J. OTSUKA*

- Assistant Manager, Analysis Technology Research Center

T. SAITO

- Advanced Materials Laboratory
  Dr. Eng.



H. MISHIMA

- Group Manager, Analysis Technology Research Center