A Particle System Approach to Simulating Construction Sprays: Shotcrete, Painting, and Abrasive Blasting

Mohammad Reza Yazdi Samadi[1] **, Ralf Waspe**[1] **, Ali Muhammad**[1] **, Christian Schlette**[1]

[1]SDU Center for Large Structure Production, Maersk Mc-Kinney Moller Institute,
University of Southern Denmark, Odense, Denmark

mry@mmmi.sdu.dk,

Abstract -

Spraying processes, such as shotcrete, painting, and abrasive blasting, are critical to construction but remain labor-intensive, hazardous, and challenging to automate. Realistic and computationally efficient simulation models are crucial for supporting robotic process planning, training, and control. However, existing approaches are typically process-specific and lack generality across different material systems. This study presents a unified, physics-informed particle system-based simulation that can represent diverse spraying processes within a single computational model. The formulation integrates stochastic adhesion and rebound mechanisms with process-dependent material parameters, enabling the reproduction of characteristic behaviors—from cohesion and detachment in shotcrete to smooth deposition in painting and particle scattering in abrasive blasting. The framework operates in near real-time and has been quantitatively validated for the shotcrete process using published experimental datasets, while additional spraying processes such as painting and abrasive blasting are demonstrated qualitatively to illustrate the generality of the modeling approach. The results demonstrate that only a limited set of parameter adjustments is required to emulate distinct process dynamics, establishing a scalable foundation for robotic spraying and trajectory optimization.

Keywords -

Physics-Based Simulation; Construction Sprays; Particle System; Robotic Spraying

1 Introduction

A wide range of spraying processes is used in construction, including surface protection and coatings, sprayed concrete (shotcrete), insulation and sealant foams, spray painting, adhesive sprays, and abrasive blasting. All of these processes are highly dynamic: material is ejected from the nozzle and projected onto a target surface, requiring the operator to both control deposition and continuously adjust based on visual feedback. Despite their importance, most of these processes are inherently hazardous—exposing workers to dust, rebound, chemical agents, or physical strain. Combined with a severe labor shortage and a declining skilled workforce in the construction sector [1], the need for robotics and automation to support, augment, or partially replace human operators becomes urgent.

To achieve human-level expertise autonomously, a robotic system must first acquire a reliable understanding of the underlying spraying process. Developing such understanding requires robust simulation models. Shotcrete provides a compelling example: it is a complex and skill-intensive process that requires operators to undergo rigorous training and certification. However, physical shotcrete tests are costly, wasteful, and unsustainable, whereas simulation-based training (e.g., [2]) offers a safer and more scalable alternative. Beyond training, simulation models enable technical analysis, waste reduction, and process optimization, provided they are sufficiently reliable and accurate.

The main contribution of this study is the extension of our previously developed shotcrete simulation model [3] to a broader class of construction spraying processes. The underlying particle-based framework has been quantitatively validated for shotcrete deposition against experimental data [4, 5]. Building on this foundation, the present work investigates whether the same physics-based simulation architecture can be generalized to other spraying applications, including painting and abrasive blasting.

Unlike prior work that focused exclusively on concrete spraying, we show that these processes can be represented within a unified particle-based framework by adjusting a limited set of material and process parameters governing adhesion, cohesion, and rebound behavior. This enables the reproduction of distinct spraying regimes while preserving a consistent simulation architecture. The additional processes considered in this study are, therefore, intended to demonstrate the generality of the framework through parameter adaptation, rather than to provide independent quantitative validation.

The remainder of this article is organized as follows. Section 2 reviews the state of the art in simulation technologies and robotic advancements relevant to each process. A theoretical overview of particle methods is provided in

Section 3. Section 4 details our simulation model, including the parameter settings and ranges for each process. The corresponding results are presented in Section 5, followed by a discussion and concluding remarks in Section 6.

2 Background

Spraying processes have long been the subject of engineering analysis and robotics research. From concrete placement to protective coatings and abrasive blasting, each domain has developed its own simulation methods and automation strategies. However, most of the existing work has remained domain-specific, with few attempts at unifying frameworks that can handle multiple processes under a standard methodology. This section reviews the state of the art in simulation in three representative applications: shotcrete, spray painting, and abrasive blasting.

2.1 Sprayed concrete

Shotcrete simulation has attracted considerable attention due to the complex interactions between aggregates, mortar, and substrate surfaces. Early work applied the Discrete Element Method (DEM) to explicitly represent aggregate and mortar phases, modeling collision, rebound, and packing effects in sprayed layers [6]. DEM-based formulations have also been extended to capture rheological behavior and yield-stress effects of fresh concrete mixtures by combining particle kinematics with macroscopic rheology models [7, 8].

Coupled Computational Fluid Dynamics–Discrete Element Method (CFD–DEM) approaches later emerged to approximate both the granular and fluid aspects of fresh concrete [9]. Nevertheless, DEM and CFD–DEM remain computationally intensive and struggle to meet real-time constraints. Velez et al. [10] traded physical realism for speed (via ray-casting adhesion) and proposed a real-time training simulator. More recently, particle system-based models have been proposed to strike a balance between realism and efficiency [4], introducing a physics-informed particle system capable of reproducing spray dynamics and cohesive failure phenomena. A voxel-based model [11], capable of modeling material evolution in real time, further underscores the growing interest in bridging the divide between computational efficiency and fidelity. These developments highlight a spectrum of approaches, from high-fidelity but slow solvers to simplified, high-performance models.

2.2 Spray painting

Spray painting is among the most extensively studied spraying processes, primarily due to its widespread use in the automotive and manufacturing industries [12]. Simulation models have been developed to analyze droplet transport, deposition, and film formation, typically by resolving the airflow field using CFD [13]. However, accurately predicting paint thickness on complex geometries remains a challenge and requires the development of advanced coupled CFD models.

Coupling CFD with a Lagrangian particle-tracking approach enables the prediction of droplet trajectories and the final film thickness. Earlier, Böttner and Sommerfeld [14] developed a comprehensive CFD–DEM model for electrostatic powder painting, combining a finite-volume approach for the airflow field with discrete particle tracking and the solution of the Laplace equation to incorporate electrostatic effects. Later, Fogliati et al. [15] employed the time-averaged Navier–Stokes equations, comprising the k–ϵ turbulence model, to predict deposition outcomes across various geometries and operating conditions.

In robotics, automated spray-painting systems often integrate simulation or predictive modeling into offline path-planning and optimization pipelines. These systems aim to achieve uniform coating thickness, minimize material waste, and improve process efficiency through model-based trajectory optimization [16, 17].

2.3 Abrasive blasting

Abrasive blasting (sandblasting) is a jet-cleaning process used for surface preparation, roughening, or coating removal. It involves high-velocity particles impacting a target surface, typically modeled as a purely particulate flow with negligible adhesion. Classical simulations of abrasive blasting have predominantly relied on the DEM. To achieve greater physical realism, Longmore et al. [18] employed DEM and represented each granule as a tetrahedral lattice of four particles, implicitly capturing static friction through the interlocking of neighboring granules. Alternative approaches have utilized CFD to analyze the influence of parameters such as particle size, blasting pressure, and nozzle distance [19].

Across all three domains, notable progress has been achieved in either high-fidelity process simulation or task-specific robotic automation. However, the literature remains fragmented: most simulation tools are tailored to a single material system, and their formulations are not readily transferable across different spraying processes. This limitation motivates the present work, which seeks to unify shotcrete application, spray painting, and abrasive blasting within a single, particle-based simulation framework—distinguished only by their process parameters.

3 Particle methods

Particle methods are a broad family of mesh-free numerical techniques in which a continuum or system is modeled through discrete entities—"particles"—whose motion and interactions collectively approximate the behavior of the underlying process. In contrast to grid-based (Eulerian) schemes, particle methods adopt a Lagrangian viewpoint that follows material motion, thereby reducing numerical diffusion during advection, simplifying the treatment of free surfaces and interfaces, and naturally handling complex boundaries and heterogeneous materials without the need for complex mesh generation. These properties make particle methods well-suited to highly dynamic processes, such as sprays, splashes, fragmentation, and other multiphase phenomena. These advantages, however, come at a cost: particle methods generally demand greater computational resources, careful implementation and tuning, and extensive validation to achieve predictive reliability.

Following the formalization proposed by [20], particle methods can be described in terms of five foundational concepts:

1. **Representation:** the basic unit or object that is being manipulated (particle, point, droplet).
2. **Interaction:** the ability of particles to affect one another (e.g., via collisions, constraints, or forces).
3. **Neighborhood:** the criterion that determines when and with which neighbors interactions occur.
4. **Evolution:** changes to a particle's state governed by external physics, independent of neighbors.
5. **Iteration:** the system's progression through discrete timesteps.

They further distinguished three levels of description for particle methods:

- A `general algorithmic framework`, including the data structures and functions common across particle-based approaches.
- A `particle method instance`, which specifies the problem or application context to which the method is applied.
- A `state transition function`, which governs how an instance evolves from its initial state to later states through repeated application of the algorithm.

This layered definition separates the abstract algorithmic design from application-specific implementations, providing a general foundation for particle-based simulation in a wide range of domains.

4 Simulation model

The proposed simulation model is implemented as a modular, physics-informed particle system in `C++`. Designed for generality, the same core framework can simulate a wide range of spraying processes—shotcrete, painting, sandblasting, and others—by modifying only material parameters and process-specific plugins. Figure 1 presents a Unified Modeling Language (UML) class diagram that organizes the framework into a particle emitter, particle data structure, surfaces, obstacles, and a physics engine for integration and collision handling.

At the core lies the `Swarm`, which manages all active and inactive particles. Each particle is represented by a lightweight structure containing its position, velocity, and lifetime, which are updated at discrete timesteps (default: $\Delta t = 0.01$ s). The swarm handles activation, deactivation, and efficient culling of particles outside the domain, enabling scalability to tens of thousands of particles and supporting real-time simulation.

Particle generation is handled by the `Emitter`, derived from an abstract interface. Emitters are parameterized by particle spawn rate, velocity range, shape, aperture, and angular spread, allowing them to replicate the behavior of different nozzle types. For example, paint spraying can be represented by high emission rates with a point emitter (near-zero aperture) for air spray guns, a rectangular emitter for a fan-shaped airless spray guns, or an annulus for rotary bell atomizers. Sandblasting, by contrast, is modeled with a circular emitter, coarser particles than spray paint, and higher initial velocities. Each emitter initializes particles with randomized attributes sampled from empirical distributions.

Within the modular framework, particles emitted by the `Emitter` are managed by the `Swarm` and updated under external forces. Each particle is represented as a geometric, massless point with optional user-defined attributes (e.g., shape, size, color) that affect only visualization. Motion evolves under external forces such as gravity, aerodynamic drag, or viscoelastic effects; in the simplest configuration, trajectories are purely ballistic. The `PhysicsEngine` advances motion explicitly using forward Euler integration at discrete timesteps, ensuring deterministic, computationally efficient updates suitable for real-time use. For collision handling, trajectories are approximated as straight-line segments per timestep, and intersections are tested against geometric obstacle primitives (planes, cylinders, boxes, spheres) using algebraic methods [21]. Impacts are then resolved through specular reflection with attenuation by a coefficient of restitution, rather than by DEM-style contact solvers. This abstraction deliberately omits particle–particle interactions to maintain scalability and real-time performance while retaining sufficient fidelity for spray-process simulation.

Whether a particle adheres or rebounds upon impact is a stochastic process. In shotcrete, a rebound probability $p_r(\theta)$ is computed from the particle's impact angle θ rel-

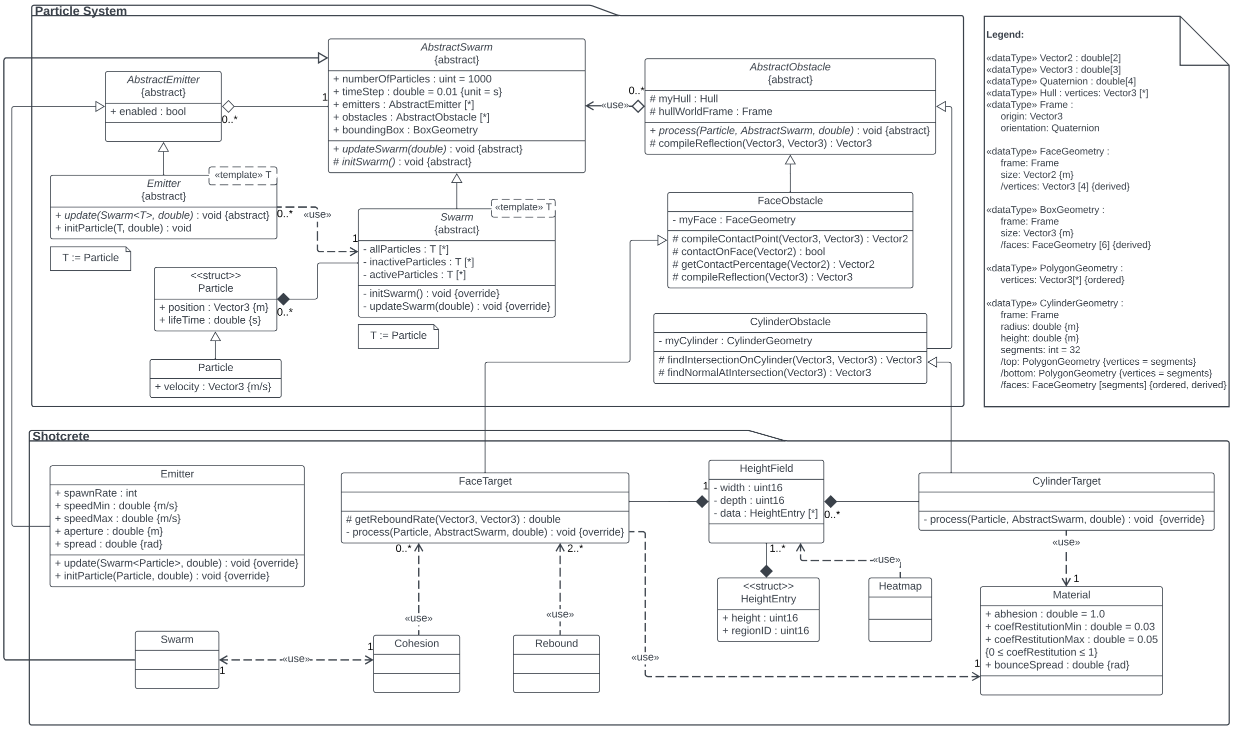

Figure 1. Unified Modeling Language (UML) class diagram of the particle system library, together with an example shotcrete extension. The model specifies abstract and concrete classes for particle emission, swarm management, and obstacle handling, conforming to the UML 2.5.1 specification.

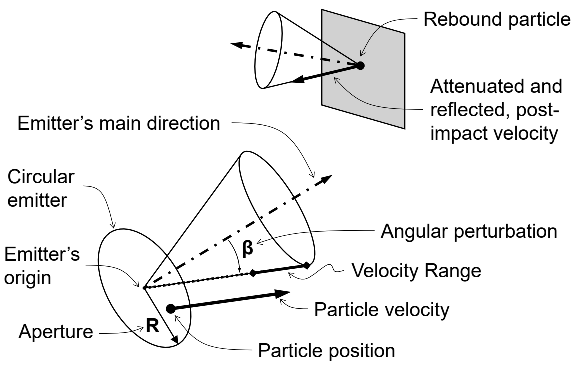

Figure 2. Circular particle emitter model. Particles are stochastically initialized within the emitter aperture and assigned velocity vectors perturbed from the primary emission axis. Perturbations are constrained by angular and magnitude bounds, thereby controlling the spread of the emission. Post-impact dynamics are represented by a reflected direction about the local surface normal, with the speed attenuated to account for energy dissipation.

ative to the surface normal using an empirical polynomial fit to Melbye's curves [22]. A uniform variate $u \sim U[0, 1]$ is drawn and scaled by a material-specific *abhesion* (i.e., anti-adhesion) factor $\alpha \in [0, 1]$; the outcome is obtained by comparing this draw with $p_r(\theta)$. For shotcrete, we use $\alpha = 1$, so the angle–dependent probability governs adhesion/rebound; for paint, we set $\alpha = 0$, suppressing rebound so that all impacts adhere. For abrasive blasting, we do *not* increase α; instead, we bypass the angle curve by setting $p_r(\theta) \equiv 1$, which forces every impact to rebound regardless of angle. Rebounding particles leave with the specularly reflected direction and an attenuated speed given by the coefficient of restitution, with a small random scatter added to match experimental spread.

Surfaces and obstacles are implemented through an abstraction hierarchy. The `AbstractObstacle` interface provides collision detection and reflection utilities, with concrete implementations such as `CylinderObstacle` and `FaceObstacle`. Targets like `FaceTarget` extend this functionality by recording deposition into a `HeightField`, a grid structure that tracks layer thickness and identifiers for cells lacking adequate support. This enables both visualization and quantitative analysis (e.g., thickness uniformity, rebound ratio), supported by a

`Heatmap` utility for graphical overlays.

Cohesion—a key sprayability parameter for achieving satisfactory in-situ properties of shotcrete mixes [23]—quantifies mesoscale structural stability and is modeled through post-processing of the heightfield. At user-defined intervals, cells that exceed local support thresholds are flagged and clustered into connected regions, identifying potential cohesive failure zones. This approach provides a computationally efficient surrogate for more detailed rheological modeling, enabling real-time analysis of failure risk while omitting costly particle–particle or fluid–structure interactions. In painting, analogous effects such as dripping or sagging can be represented but are generally negligible, whereas abrasive blasting does not exhibit cohesive failure at all. A summary of the default parameters for these three processes is provided in Table 1.

The recommended heightfield resolution depends on the target size; nevertheless, higher values such as 1024×1024 are generally advised. Since the simulation model decouples resolution from runtime performance, increasing resolution improves the accuracy of adhesion (i.e., material accumulation) and cohesion analysis without incurring significant computational penalties [4].

The simulation timestep Δt was selected in the range of 10^{-4}–10^{-2} s to balance accuracy and throughput. To avoid missed collisions, Δt must be small enough that the maximum particle displacement per step is less than the smallest obstacle. A default timestep of 10^{-3} s is recommended, with adjustments possible depending on performance and accuracy requirements.

The model achieved near real-time performance without employing common acceleration strategies, despite the availability of effective and well-established techniques such as spatial hashing [25] and data-parallel computation [26]; in typical configurations involving on the order of 10^4-–10^5 active particles, real-time execution was achieved on standard computing hardware, consistent with previously reported benchmarks [3, 4]. Incorporating such methods in future work could further extend scalability to simulations involving millions of particles.

5 Results

As described in Section 4, emission directions are sampled from probability distributions constrained by the spray cone angle. Figure 3 compares two sampling strategies: a uniform distribution and a truncated normal distribution. The uniform distribution produces a stronger concentration of particles near the spray cone axis, whereas the normal distribution allows the spread to be adjusted via its mean (μ) and standard deviation (σ). In earlier work [4], we reproduced on-target accumulation profiles of wet-mix shotcrete reported by Ginouse and Jolin [27] by setting $\mu = 0.255$ and $\sigma = 0.102$. The simulation achieved errors of approximately **5%**, **3%**, and **2.8%** for maximum absolute, root-mean-square, and mean absolute error, respectively, relative to empirical data.

Figure 4 demonstrates four representative runs of the same robotic spraying task under different process models and obstacle configurations, highlighting the simulator's flexibility and the qualitative effects of process physics and geometry. Panel (a) illustrates shotcrete with reinforcements, showcasing pronounced shadowing and underdeposition behind reinforcement bars. Rebounded particles are visible in flight and are deposited on the ground heightfield, while cohesive buildup and detachment are possible. Panel (b)—shotcrete without reinforcements—exhibits a largely radially symmetric accumulation pattern with far fewer localized voids, demonstrating how obstacles strongly affect coverage even for identical nozzle kinematics. Panel (c) depicts spray painting, displaying near-complete immediate adhesion and a uniformly covered surface, consistent with high-adhesion, no rebound, and negligible sagging. Abrasive blasting is illustrated in Panel (d), which contrasts sharply: adhesion is suppressed, nearly all particles rebound with significant scattering, and no material accumulates on the target; the heatmap therefore represents hit coverage and suggests erosion intensity.

All simulations were executed using process parameters derived from Table 1; specifically, cohesion and the angle-dependent rebound model were enabled for the shotcrete cases, rebound was suppressed for the painting, and forced

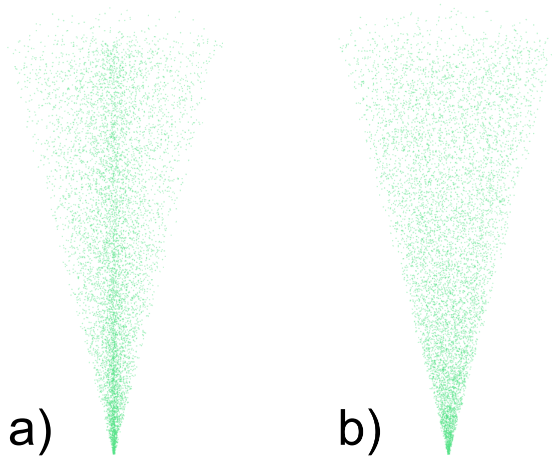

Figure 3. Comparison of particle emission patterns when directions are sampled from **a)** a uniform distribution and **b)** a truncated normal distribution ($\mu = 0.255$, $\sigma = 0.102$). The normal distribution offers tunable control over spread, enabling closer alignment with empirical spray patterns. All other parameters were held constant.

Table 1. Default simulation parameters for three spraying processes.

Parameter (units)	Shotcrete	Spray painting	Abrasive blasting
Emitter type	Circular	Point / rectangular / annulus	Circular
Particle spawn rate (particles/s)	10^4	10^4–10^5	10^3–10^5
Particle size*	3.0 (aggregate)	0.05 (droplet)	0.5 (grain)
Particle velocity (m/s)†	20–30 (wet-mix)	20–60	100–200
Emitter aperture (mm)	32–76	0.5–2.0 (point emitter)	3.2–19.1
Spray cone angle (°)	9–18	15–120+	15
Abhesion α	1.0 (angle-based)	0.0 (rebound suppressed)	N/A (bypassed, forced rebound)
Rebound model	Angle-dependent polynomial	N/A	forced rebound ($p_r = 1$)
Coefficient of restitution	0.03–0.05	N/A	0.6–0.9
Cohesion enabled?	Yes	No (optional drip)	No
Scattering spread	Small angular noise	Negligible	Moderate–large
Notes	High cohesion, angle rebound	Near-instant adhesion	Elastic impacts, erosion

* Particle size refers to the rasterized diameter (in pixels) of points drawn with `GL_POINTS`, stored as a floating-point value.
† Particle velocity is highly affected by input airflow, air pressure, and aggregate size [24].

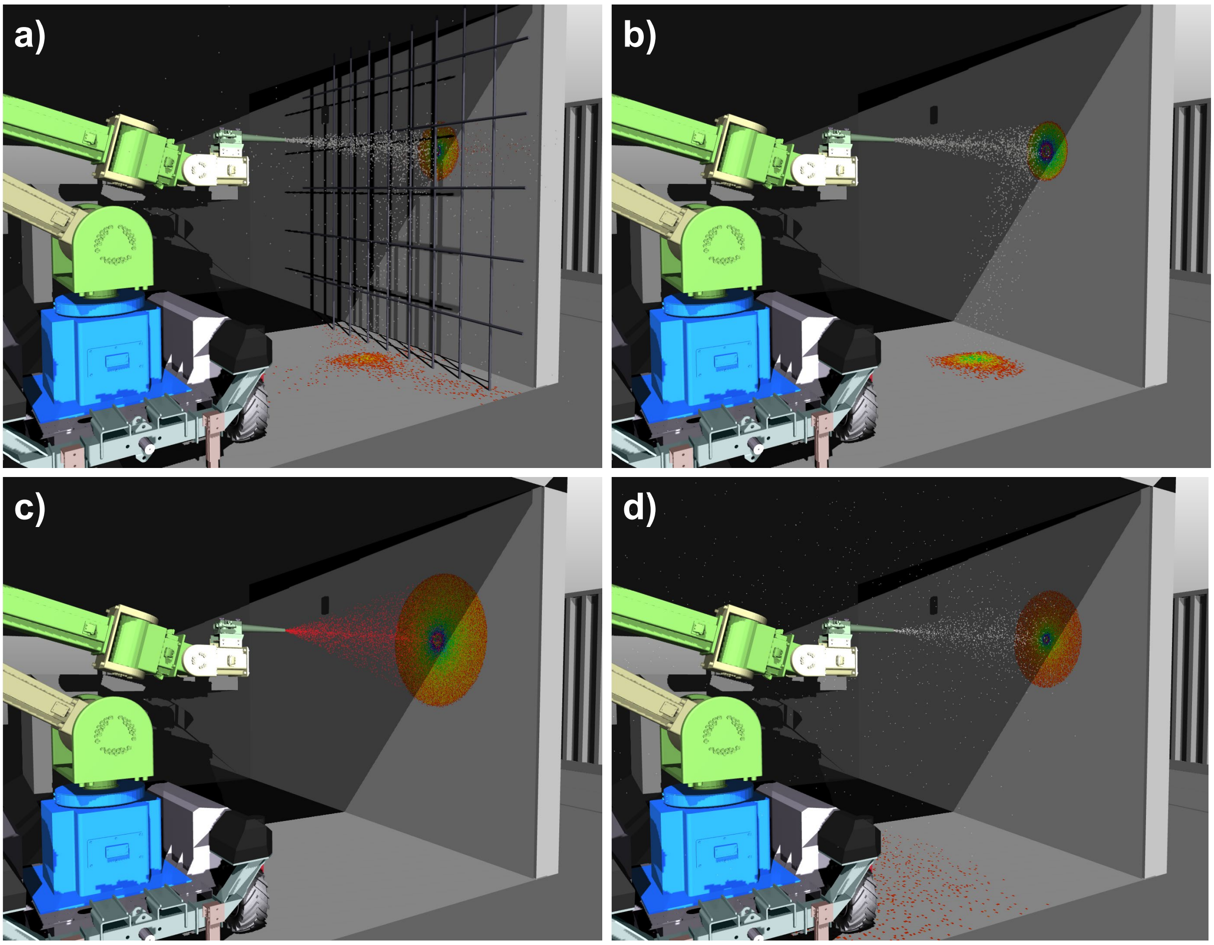

Figure 4. Comparison of robotic spraying simulations across four processes: **a)** shotcrete with reinforcement elements, **b)** shotcrete deposition without reinforcement, **c)** spray painting, and **d)** abrasive blasting, where adhesion is disabled, and all particles rebound with high scattering. Parameters for each scenario were selected according to Table 1.

rebound (high restitution and scatter) was used for blasting. These visual comparisons demonstrate that changing only plugin parameters (while keeping the same emitter/nozzle kinematics) produces distinct, physically interpretable outcomes.

6 Discussion and conclusion

The results confirm that the unified framework can simulate characteristic behaviors in different spraying processes. For instance, in the shotcrete configuration, the model successfully captures cohesive failure and material detachment phenomena, which are absent in the painting mode. Conversely, in painting mode, high adhesion and low rebound parameters produce a smooth surface finish; however, sagging may still occur. In abrasive blasting, the system exhibits purely particulate scattering without accumulation. These qualitative distinctions are consistent with the expected physical behavior of each process, supporting the generality of the proposed model. In this context, variations in physical sprayer characteristics such as flow rate, nozzle geometry, and supply pressure are implicitly captured through emitter parameters (particle spawn rate, velocity distribution, and angular spread), whose influence on deposition behavior has been previously analyzed for the shotcrete process [4], while further process-specific validation under controlled operating conditions remains a direction for future work.

A key advantage lies in the modular software architecture. The same core simulation engine operates across all modes, with behavior determined solely by parameter sets that define material and process properties. This demonstrates the feasibility of our central objective: to enable a single, physics-informed particle system to emulate diverse spraying processes by varying only their governing parameters. Such an approach can be directly embedded into robotic programming environments or Digital Twins, enabling engineers to switch between simulation-based design and control processes seamlessly.

The automation potential of this unified approach is substantial. A robotic cell could employ simulation-in-the-loop optimization to refine spray patterns until predicted deposition or coverage meets desired criteria. Similarly, robotic abrasive blasting systems could test and evaluate different nozzle trajectories in simulation to maximize cleaning efficiency while minimizing material loss. Because the simulation executes in near real time, it also provides a practical foundation for machine learning–based process control or operator training in virtual environments.

The current formulation also excludes several classes of physical phenomena, including particle–particle interactions in dense flow regimes, complex fluid behaviors such as splashing and droplet coalescence, and post-deposition volumetric evolution such as lateral flow or expansion (e.g., in foam-based materials). As a result, the model is most appropriate for particle-dominated spraying processes where deposition is primarily governed by particle–surface interactions, including adhesion, rebound, and accumulation, and where a heightfield representation provides a sufficient approximation of material buildup. This limitation also prevents simulating processes characterized by significant post-deposition evolution, such as expanding foam. Transitioning to a three-dimensional cellular automaton (3D CA) would substantially enhance geometric expressiveness and enable simulation of time-dependent material behavior, including lateral motion and post-impact spreading. Overall, while the framework is robust and extensible, advancing its fidelity will require volumetric modeling, physically grounded parameter calibration, and broader experimental validation. These developments will further strengthen its applicability as a tool for robotic construction, process optimization, and virtual skill assessment across diverse spraying applications.

References

[1] C. Klein and J. Smith. Addressing labour and skills shortages in a fast-changing economy. *Documents de travail du Département des Affaires économiques de l'OCDE*, 2024. doi:10.1787/757311cb-en.

[2] Edvirt — training for the mining and tunneling industry. online: `https://www.edvirt.com/`, Accessed: 25/9/2025.

[3] M. R. Yazdi Samadi, R. Waspe, and C. Schlette. Toward shotcrete process simulation to support robotic operation. In *Proceedings of the International Symposium on Automation and Robotics in Construction*, pages 707–713, Lille, France, 2024. doi:10.22260/ISARC2024/0092.

[4] M. R. Yazdi Samadi, R. Waspe, and C. Schlette. Physics-based particle system modeling of shotcrete process for robotic placement. *Construction Robotics*, 9(30), 2025. doi:10.1007/s41693-025-00172-y.

[5] M. R. Yazdi Samadi, R. Wu, S. Gholami, R. Waspe, A. Muhammad, A. Billard, and C. Schlette. From human to height-field: Predictive shotcrete simulation with a physics-informed particle system. In *Proceedings of the 22nd International Conference on Advanced Robotics (ICAR)*, pages 826–832, San Juan, Argentina, 2025. doi:10.1109/ICAR65334.2025.11338646.

[6] U. C. Puri and T. Uomoto. Characterization of distinct element modeling parameters for fresh concrete

and its application in shotcrete simulations. *Journal of materials in civil engineering*, 14:137–144, 2002. doi:10.1061/ ASCE!0899-1561 2002!14:2 137!

[7] K. Anjneya and A. Deb. A review of the discrete element method for modelling the rheology of fresh concrete. In *Recent Advances in Computational and Experimental Mechanics, Vol—I*, pages 81–95, Singapore, 2022. doi:10.1007/978-981-16-6738-1_7.

[8] V. Mechtcherine, A. Gram, K. Krenzer, J.-H. Schwabe, S. Shyshko, and N. Roussel. Simulation of fresh concrete flow using Discrete Element Method (DEM): theory and applications. *Materials and structures*, 47:615–630, 2014. doi:10.1617/s11527-013-0084-7.

[9] G. Ma, H. Ma, and Z. Sun. Simulation of two-phase flow of shotcrete in a bent pipe based on a CFD–DEM coupling model. *Applied Sciences*, 7:3530, 2022. doi:10.3390/app12073530.

[10] G. Velez, L. Matey, A. Amundarain, Á. Suescun, J. Marín, and C. de Dios. Modeling of shotcrete application for use in a real-time training simulator. *Computer-Aided Civil and Infrastructure Engineering*, 28(6):465–480, 2013. doi:10.1111/j.1467-8667.2012.00788.x.

[11] S. Yamakawa, K. Vazquez-Santiago, Y. Xia, H. Ogura, S. Yamamoto, R. Nakanishi, and K. Shimada. Concrete spray 3D printing simulator for nozzle trajectory planning. *IEEE Robotics and Automation Letters*, 2025. doi:10.1109/lra.2025.3615038.

[12] M. R. Pendar, F. Rodrigues, J. C. Páscoa, and R. Lima. Review of coating and curing processes: Evaluation in automotive industry. *Physics of Fluids*, 34(10):101301, 2022. doi:10.1063/5.0109376.

[13] Y. Chen, W. Chen, B. Li, G. Zhang, and W. Zhang. Paint thickness simulation for painting robot trajectory planning: a review. *Industrial robot*, 44(5): 629–638, 2017. doi:10.1108/IR-07-2016-0205.

[14] C. U. Böttner and M. Sommerfeld. Numerical calculation of electrostatic powder painting using the Euler/Lagrange approach. *Powder technology*, 125(2):206–216, 2002. doi:10.1016/S0032-5910(01)00508-3.

[15] M. Fogliati, D. Fontana, M. Garbero, M. Vanni, and G. Baldi. CFD simulation of paint deposition in an air spray process. *Journal of Coatings Technology Research*, 3(2):117–125, 2006. doi:10.1007/s11998-006-0014-5.

[16] W. Chen, D. Zhao, and B. Zhang. Path planning for spray painting robot of workpiece surfaces. *Mathematical problems in engineering*, 2013:1–6, 2013. doi:10.1155/2013/659457.

[17] P. J. From and J. T. Gravdahl. A real-time algorithm for determining the optimal paint gun orientation in spray paint applications. *IEEE transactions on automation science and engineering*, 7(4):803–816, 2010. doi:10.1109/TASE.2009.2033567.

[18] J. P. Longmore, P. Marais, and M. M. Kuttel. Towards realistic and interactive sand simulation: A GPU-based framework. *Powder technology*, 235:983–1000, 2013. doi:10.1016/j.powtec.2012.10.056.

[19] X. Yu, X. Wei, X. Huang, and L. Zhang. Study on the modeling method and influencing parameters of sandblasting process for blade grinding. *Advances in Materials Science and Engineering*, 2022:1–19, 2022. doi:10.1155/2022/7905927.

[20] J. Pahlke and I. F. Sbalzarini. A unifying mathematical definition of particle methods. *IEEE Open Journal of the Computer Society*, 4:97–108, 2023. doi:10.1109/OJCS.2023.3254466.

[21] D. H. Eberly. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*. Elsevier, Amsterdam, 2007. doi:10.1201/b18212-12.

[22] T. Melbye. Sprayed concrete for rock support. MBT International, 2001.

[23] N. Trussell and S. Jacobsen. Review of sprayability of wet sprayed concrete. *Nordic Concrete Research*, 63(2):21–41, 2020. doi:10.2478/ncr-2020-0016.

[24] N. Ginouse and M. Jolin. Experimental and numerical investigation of particle kinematics in shotcrete. *Journal of Materials in Civil Engineering*, 26: 06014023, 2014. doi:10.1061/(ASCE)MT.1943-5533.0001092.

[25] S. Lefebvre and H. Hoppe. Perfect spatial hashing. *ACM transactions on graphics*, 25(3):579–588, 2006. doi:10.1145/1141911.1141926.

[26] K. Sims. Particle animation and rendering using data parallel computation. *SIGGRAPH Computer Graphics*, 24:405–413, 1990. doi:10.1145/97880.97923.

[27] N. Ginouse and M. Jolin. Mechanisms of placement in sprayed concrete. *Tunnelling and Underground Space Technology*, 58:177–185, 2016. doi:10.1016/j.tust.2016.05.005.