Contents lists available at ScienceDirect

Additive Manufacturing

journal homepage: www.elsevier.com/locate/addma

Research paper

A differentiable material point method solver for the modeling, simulation and analysis of extrusion-flow process for 3D concrete printing

Wenchang Zhang [a], Chuanqi Liu [b], Yiwei Weng [c], Tianju Xue [a,*], Jidong Zhao [a]

[a] Department of Civil and Environmental Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China
[b] State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, 100190, China
[c] Department of Building and Real Estate, The Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Keywords:
3D concrete printing
Material point method
Differentiable programming
Sensitivity analysis

ABSTRACT

Recent years have witnessed rapid development in 3D Concrete Printing (3DCP), which offers a highly automated construction process with significant reductions in material, labor, and time costs. However, deviations from the original design often occur as a result of the complex extrusion-flow behavior of fresh concrete and the associated layer deformations, which are primarily caused by insufficient understanding of key process parameters. In this study, a differentiable material point method (MPM) solver is developed for the modeling and analysis of 3DCP processes, enhanced with reverse-mode automatic differentiation techniques built upon the discrete adjoint method to enable end-to-end derivative computations. The extrusion-flow process of 3DCP is modeled as a weakly compressible Bingham fluid. For single-layer deposition under various printing conditions, the maximum deviations in predicted width and height remain within 20% of experimental measurements. Despite the increased deviations for multi-layer structures due to cumulative effects, the framework provides a robust foundation for simulating the 3DCP process. Beyond its predictive capabilities, the differentiable MPM framework exhibits high efficiency in sensitivity analysis. As demonstrated in a case study of sensitivity with respect to eight process parameters, the framework reduced the computational time to just 60% of that required by the central difference method. This efficient gradient computation provides quantitative insight into 3DCP mechanics and offers a pathway for optimizing process parameters to improve final product quality.

1. Introduction

In recent years, 3D Concrete Printing (3DCP) has attracted significant attention in the construction industry due to its potential to revolutionize traditional methods [1–3]. The most widely adopted technology in this field is extrusion-based 3DCP, whose process chain typically consists of production, transportation/pumping, extrusion and deposition stages [4]. As illustrated in Fig. 1, this printing process involves coordinated interaction between a printing nozzle and a build platform. Fresh concrete is initially extruded onto the platform, while the nozzle follows predetermined trajectories at a controlled velocity to precisely position material deposits. After completing the specific layout of an individual layer, the nozzle elevates to deposit new concrete at the top layers. This layer-by-layer process continues until the designed structure is constructed. Throughout this process, concrete undergoes a critical phase transformation: initially exhibiting viscous fluid behavior to facilitate extrusion and flow, then progressively developing solid-like characteristics to withstand the pressure from the upper concrete. As an advanced automated technology, 3DCP has the potential to significantly reduce the cost of time, materials, and labor.

While 3DCP can offer substantial advantages, further investigation is necessary to promote its industry adoption. Current studies for 3DCP primarily address two aspects: structural failure and extrusion flow [5–10]. In structural failure analysis, significant progress has been made in understanding both the structural instabilities [11] that can occur during the printing process, such as elastic buckling and plastic collapse, and the mechanical behavior of the final hardened product. For instance, sophisticated numerical frameworks have been developed to simulate intricate crack propagation using extended finite element multiscale methods [12]. The extrusion flow process focuses more on the extruded flow behavior of fresh concrete and can be regarded as the basis for 3DCP, as it directly determines the layer geometry and structure, which in turn affects the structural stability and the mechanical properties of the final product [13,14], and has received extensive attention. Numerous experimental studies have been

* Corresponding author.
E-mail address: cetxue@ust.hk (T. Xue).

https://doi.org/10.1016/j.addma.2026.105126
Received 10 August 2025; Received in revised form 12 December 2025; Accepted 13 February 2026
Available online 19 February 2026

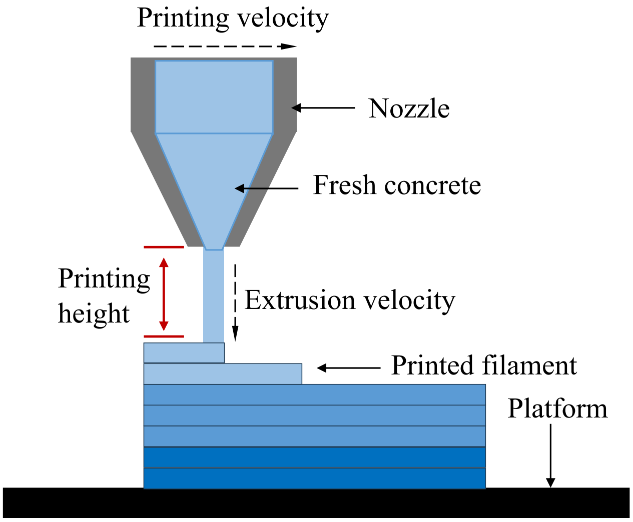

Fig. 1. Schematic diagram of 3DCP process.

conducted and relevant mechanisms have been identified [15–18]. Although these experiments yield valuable insights, their high cost and complexity pose significant limitations-particularly when it comes to systematically exploring how process parameters affect printing outcomes. Meanwhile, numerical modeling and simulations have the potential to provide a more cost-effective alternative for predicting printed processes and investigating process-parameter relationships, drawing increasing attention [19–24].

The finite volume method, a popular numerical method in computational fluid dynamics (CFD), is adopted in [25] to successfully replicate experimentally observed cross-sectional profiles under diverse printing conditions. Subsequent studies extend this mesh-based approach to study various printable concrete materials and printing boundary conditions [26–29]. However, the finite volume method faces challenges in accurately tracking the concrete-air interface, as additional interface-capturing algorithms-such as the volume-of-fluid (VOF) [30] or level set methods [31]-must be implemented. These algorithms not only increase computational complexity, but can also introduce numerical diffusion or mass loss, ultimately reducing the accuracy of interface representation. As an alternative, particle-based numerical methods have been employed for extrusion flow simulation. These methods make it easier to track the free surface without the need for complex interface-tracking algorithms. For instance, discrete element method (DEM), widely applied for fresh concrete flow, has been employed to model the buildability and flow characteristics of printable concrete mixtures [32]. However, the core formulation of DEM is predicated on discrete contact detection, which presents fundamental obstacles to efficient gradient-based sensitivity analysis and optimization. Recently, smoothed particle hydrodynamics (SPH) has been used to create a 2D model to predict the final cross-section shape of 3DCP [33]. The particle finite element method (PFEM) has also been applied in [4,34,35] to simulate the flow extrusion process of fresh concrete. Nevertheless, both methods can involve relatively high computational costs: SPH may require expensive global neighbor searches for particle interaction detection, while PFEM often relies on computationally intensive mesh regeneration. These issues highlight the urgent need for a more suitable numerical method that can effectively simulate fluid flow in the extrusion process and, in the future, can be possibly extended to solid structural modeling as well.

The material point method (MPM) offers a promising solution. MPM is a continuum-based particle method that extends the particle-in-cell (PIC) method used in computational fluid dynamics to computational solid dynamics [36]. Relying on Lagrangian material points that carry physical information and a background Eulerian grid for discretizing continuous fields, MPM is well-suited to handling large deformation problems while avoiding mesh distortion [37–39]. As a result, MPM has been widely used in scenarios where traditional mesh-based methods struggle, such as simulation of landslides and collapses [40–43]. More recently, it has been adopted to simulate the slump test of fresh concrete [44]. The good agreement between the calculated results and the experimental data demonstrates its ability to simulate complex fluid behavior. Therefore, it is promising to use MPM for the modeling and simulation of 3DCP, an effort that has received rather limited attention. Following its initial development, many technical means have been proposed to improve the family of material point methods [45–49]. Among the variants of MPM, the moving least squares material point method (MLS-MPM), proposed by Hu [50], stands out due to its accurate physical field reconstruction, hence is chosen to be implemented as the foundation of this work to simulate the 3D concrete printing process.

While a suite of powerful numerical tools, ranging from commercial software like Abaqus CEL and FLOW-3D to advanced academic methods such as PFEM variants, have demonstrated considerable success in providing high-fidelity forward predictions of the 3DCP process [4,25,51], an efficient and quantitative method to evaluate the influence of process parameters on printing outcomes is underexplored. Cominal et al. [25] conduct a parametric study by varying the process parameters in Flow-3D to investigate how the velocity ratio (V/U), determined by the printing velocity (V) and extrusion velocity (U), influence the cross-sectional width of the printed single layer, showing that higher V/U ratios result in narrower deposited layers. Reinold et al. [4] systematically vary yield stress and plastic viscosity to study their effects on substrate deformation in a series of PFEM simulations. The results demonstrate that materials with lower yield stress and higher viscosity are more prone to plastic yielding. These parametric studies typically require repeated simulation executions with different parameter settings, which can be computationally expensive and inefficient. In contrast, automatic differentiation (AD), which is the cornerstone of the modern machine learning field, provides the possibility for efficient and accurate sensitivity evaluation [52]. Compared with direct finite difference method, AD yields highly accurate derivatives up to machine precision, avoiding the truncation error inherent in finite difference method. Furthermore, AD enhances computational efficiency in high-dimensional scenarios, as reverse-mode AD calculates gradients for all input parameters through a single forward pass followed by a single backward pass, opposed to the finite difference method, which necessitate multiple separate forward evaluations. In contrast to manual adjoint method, AD offers significant automation, requiring minimal code modifications and eliminating the need for problem-specific derivation of adjoint equations. These benefits have motivated the development of differentiable physical solvers, which can efficiently and accurately compute the gradients of simulation outputs with respect to input parameters, facilitating parametric studies, sensitivity analyses, and inverse problem solving. For example, differentiable solvers have been applied to inverse problems in fluid-particle interactions [53] and complex porous media flows [54]. While automatic differentiation has shown significant promise in various scientific and engineering fields, its ability for automatic sensitivity analysis in 3DCP remains largely unexplored. However, the application of AD to MPM-based 3DCP simulations can be challenging. Two primary obstacles stand out: (1) the prohibitive memory costs of storing the entire computational graph for long-duration simulations, and (2) the algorithmic non-smoothness introduced by conditional logic in process modeling, such as nozzle-material interaction.

Accordingly, this paper introduces a differentiable MPM framework capable of simulating the extrusion flow process and efficiently computing its sensitivities. The developed framework addresses the aforementioned challenges by incorporating two key components: a memory-efficient checkpointing strategy and a smoothed, differentiable approximation for boundary treatment. To provide a clear understanding of this study, the paper is organized as follows. Section 2 presents the methodology used for building the differentiable solver for 3DCP simulation. This section first introduces the general framework, including the constitutive model for fresh concrete, the MPM discretization scheme, and the memory-efficient differentiable program development. Subsequently, 3DCP-specific modeling is presented, covering the boundary treatments of the printing process and the smoothing approximation for sensitivity computation. Section 3 details the 3DCP simulations conducted under various printing conditions. The simulation results are compared against experimental data to validate the accuracy of the simulation model. Section 4 leverages the validated model to perform an automatic sensitivity analysis, investigating the influence of printing parameters on the final printed geometry. The reliability of the differentiable solver is substantiated by the agreement between the computed sensitivities and established findings in the literature. Furthermore, a multi-parameter study is introduced to demonstrate the computational efficiency of AD, which highlights its potential to 3DCP process design. Section 5 concludes the application of developed numerical solver in 3DCP and outlines future work.

2. Differentiable material point method and 3DCP simulation model

In this section, the constitutive model of fresh concrete and the numerical technique used for the differentiable MLS-MPM solver development are first introduced. The specific modeling of the 3DCP extrusion process and the requisite treatment for its automatic differentiation process are subsequently discussed.

2.1. Modeling of fresh concrete

During the printing process, fresh concrete is deposited and gradually spreads across the printing platform or onto the previous printed layer. This process can be described by the conservation equations for mass and momentum [25]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0, \tag{1}$$

$$\rho \frac{d\boldsymbol{v}}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{b}, \tag{2}$$

where ρ is the density of the fluid, $\boldsymbol{v}$ is the velocity, $d\boldsymbol{v}/dt$ is the acceleration, $\boldsymbol{\sigma}$ is the Cauchy stress, and $\boldsymbol{b}$ is the body force. The Cauchy stress can be decomposed into two parts for fluid:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}, \tag{3}$$

where p is the pressure and $\boldsymbol{\tau}$ is the shear stress. Fresh concrete can be considered incompressible, with pressure typically updated using semi-implicit or iterative methods, which are computationally expensive. To improve the efficiency, fresh concrete is treated as a weakly compressible fluid in this paper and corresponding equation of state is introduced to calculate the pressure [33]:

$$p = K(\frac{\rho}{\rho_0} - 1), \tag{4}$$

where K is the bulk modulus, ρ is the density at current time, and ρ_0 is the initial density. The second part of the Cauchy stress, $\boldsymbol{\tau}$ is calculated as follow for a weakly compressible Newtonian fluid:

$$\boldsymbol{\tau} = 2\mu \boldsymbol{D}', \tag{5}$$

where μ is the viscosity of the fluid and $\boldsymbol{D}'$ is the deviatoric strain rate tensor, obtain from the symmetric part of the velocity gradient:

$$\boldsymbol{D}' = \frac{1}{2}\left(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^{\mathsf{T}}\right) - \frac{1}{3}(\nabla \cdot \boldsymbol{v})\mathbf{I}. \tag{6}$$

Since fresh concrete is typically regarded as a Bingham fluid [55–58], the above formula must be modified accordingly. Fig. 2 illustrates the stress–strain rate relationship of Newtonian fluid, Bingham fluid, and its bi-viscous model approximation. As shown, ideal Bingham fluid remains rigid until the shear stress exceeds the yield stress. To capture this behavior, a bi-viscous model is employed [25]:

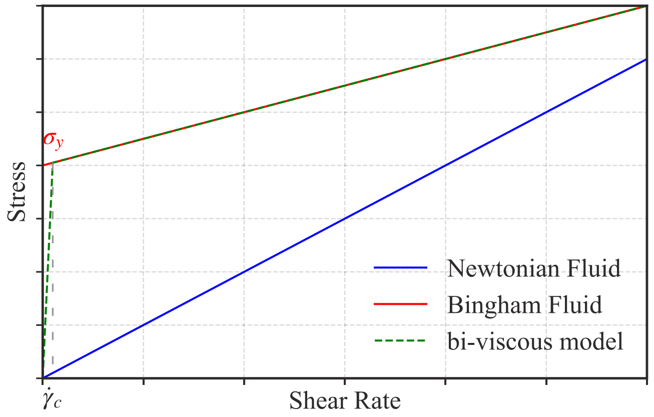

Fig. 2. The stress–strain rate relationship of Newtonian, Bingham fluid and its bi-viscous approximation.

$$\boldsymbol{\tau} = 2\mu_{\mathrm{a}}\boldsymbol{D}' = \begin{cases} 2(\frac{\tau_y}{\dot{\gamma}} + \mu)\boldsymbol{D}', & \text{if } \dot{\gamma} > \dot{\gamma}_{\mathrm{c}}, \\ 2(\frac{\tau_y}{\dot{\gamma}_{\mathrm{c}}} + \mu)\boldsymbol{D}' & \text{if } \dot{\gamma} \leq \dot{\gamma}_{\mathrm{c}}, \end{cases} \tag{7}$$

where μ_{a} is the apparent viscosity, μ is the constant viscosity after yielding, $\dot{\gamma}$ is the magnitude of the deviatoric strain rate tensor with $\dot{\gamma} = \sqrt{2\boldsymbol{D}' : \boldsymbol{D}'}$, and $\dot{\gamma}_{\mathrm{c}}$ is a critical shear rate to prevent the apparent viscosity μ_{a} from increasing infinitely as the shear rate $\dot{\gamma}$ decreases. As shown in Fig. 2, the bi-viscous model has a good approximation for Bingham fluids when the strain rate is greater than the critical shear rate. In this paper, the value of $\dot{\gamma}_{\mathrm{c}}$ is fixed at 0.01, which means the maximum viscosity is the sum of the constant viscosity after yielding and one hundred times the yield stress.

2.2. Moving least squares material point method

For the completeness of the presentation, the procedures of the MPM discretization are described here with essential details. Eqs. (1) and (2), along with proper boundary conditions and a constitutive model, can describe the flow behavior during the printing process. In this paper, the MLS-MPM is used to numerically solve these equations.

The basic procedure of MLS-MPM is shown in Fig. 3. In MLS-MPM, continuous objects are discretized by particles, which carry necessary physical properties, such as mass, volume, velocity and stress. At the beginning of each time step, this information is transferred to grid nodes (P2G) based on the moving least squares approximation. Here, grid nodes only receive information from particles within a certain region, referred to the support domain. The size of this support domain depends on the weight function used in the mapping scheme. Next, the momentum equation at each grid node is solved explicitly to update the nodal accelerations and velocities. The velocities of nodes are also adjusted to satisfy boundary conditions at this stage (Grid operation). These updated nodal quantities are then mapped back to the particles, updating their positions and physical quantities (G2P). Finally, all grid information is erased to start the next time step.

Following the MLS-MPM process, particles interact with each other through nearby grid nodes. The particle-grid transfer poses challenges to parallel simulation due to data racing and load balancing, particularly on modern hardware like GPUs. For example, geometrically neighboring particles need to simultaneously contribute to the same grid nodes during the P2G process, creating write hazards. In this paper,

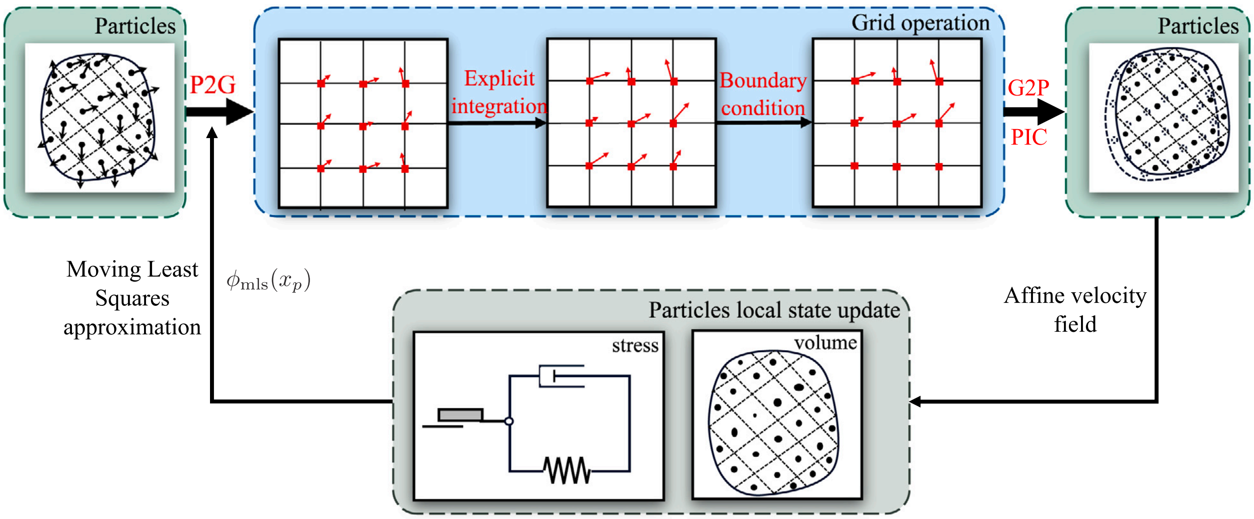

Fig. 3. The computational scheme of MLS-MPM.

the MLS-MPM solver is developed in Taichi, a domain-specific language designed for high-performance, parallel computing [59]. Taichi provides automatic GPU acceleration, allowing for the development of highly parallel programs without dealing with low-level GPU implementation, which enables the developed solver to achieve high efficiency on modern GPUs.

During the flow process, the mass of the particle remains constant, ensuring that the conservation of mass is automatically satisfied. The momentum equation is solved on the Eulerian grid. The semi-discretized form of the momentum equation at each grid node i is (details of the discretization can be found in Appendix A):

$$m_i \boldsymbol{a}_i = \boldsymbol{f}_i^{int} + \boldsymbol{f}_i^{ext}, \tag{8}$$

where $\boldsymbol{a}_i$ is nodal acceleration. m_i, $\boldsymbol{f}_i^{int}$, and $\boldsymbol{f}_i^{ext}$ are nodal mass, internal force, and external force, which are mapped from particle quantities (P2G):

$$m_i = \sum_{p=1}^{n_p} m_p \xi_{ip}, \tag{9}$$

$$\begin{aligned} \boldsymbol{f}_i^{int} &= -\sum_{p=1}^{n_p} \frac{4}{\Delta x^2} \xi_{ip} (\boldsymbol{\sigma}_p \cdot (\boldsymbol{x}_i - \boldsymbol{x}_p)) V_p, \\ \boldsymbol{f}_i^{ext} &= \sum_{p=1}^{n_p} \xi_{ip} \boldsymbol{t}_p V_p h^{-1} + \sum_{p=1}^{n_p} \xi_{ip} m_p \boldsymbol{b}_p, \end{aligned} \tag{10}$$

where n_p is the total number of particles, ξ is the weighting function used to construct the MLS shape functions (as detailed in Appendix A). The subscript p denotes the function value associated with particle p. m_p is the particle mass, ξ_{ip} is the weight of particle p on node i, V_p is the particle volume, $\boldsymbol{t}_p$ is the traction vector on particle p and h is the boundary layer thickness. The quadratic B-spline function is employed here as the weighting function ξ [60]. In three dimensions, the function is constructed from the tensor product of 1D quadratic B-spline kernels W_2. The 1D kernel is defined over a normalized coordinate $s = |x|/\Delta x$, where Δx is the grid spacing:

$$W_2(s) = \begin{cases} \frac{3}{4} - s^2, & 0 \le s < \frac{1}{2}, \\ \frac{1}{2}\left(\frac{3}{2} - s\right)^2, & \frac{1}{2} \le s < \frac{3}{2}, \\ 0, & \frac{3}{2} \le s. \end{cases} \tag{11}$$

The value of the function associated with node i evaluated at a particle position $\boldsymbol{x}_p$ is then given by:

$$\xi_i(\boldsymbol{x}_p) = W_2(|s_x|) W_2(|s_y|) W_2(|s_z|), \tag{12}$$

where $\boldsymbol{s} = (\boldsymbol{x}_p - \boldsymbol{x}_i)/\Delta x = (s_x, s_y, s_z)$ is the normalized relative position vector. The support of this 3D weighting function is a cubic region with a side length of $3\Delta x$, meaning each particle interacts with a 3 × 3 × 3 block of neighboring grid nodes.

Nodal velocity can be updated by explicitly solve Eq. (8):

$$\boldsymbol{v}_i^{new} = \boldsymbol{v}_i + \Delta t \cdot \boldsymbol{a}_i = \boldsymbol{v}_i + \Delta t \cdot \frac{\boldsymbol{f}_i^{int} + \boldsymbol{f}_i^{ext}}{m_i}. \tag{13}$$

Furthermore, a velocity mapping scheme between particle and grid node is necessary for MPM. The MLS-MPM and the affine particle-in-cell (APIC) [61] share the same mapping scheme:

Particle to grid (P2G):

$$(m\boldsymbol{v})_i = \sum_{p=1}^{n_p} m_p \xi_{ip} (\boldsymbol{v}_p + \boldsymbol{C}_p (\boldsymbol{x}_i - \boldsymbol{x}_p)), \tag{14}$$

Grid to particle (G2P):

$$\begin{aligned} \boldsymbol{v}_p &= \sum_{i=1}^{n_i} \xi_{ip} \boldsymbol{v}_i, \\ \boldsymbol{D}_p &= \sum_{i=1}^{n_i} \xi_{ip} (\boldsymbol{x}_i - \boldsymbol{x}_p)(\boldsymbol{x}_i - \boldsymbol{x}_p)^T, \\ \boldsymbol{C}_p &= (\sum_{i=1}^{n_i} \xi_{ip} \boldsymbol{v}_i (\boldsymbol{x}_i - \boldsymbol{x}_p)^T) \boldsymbol{D}_p^{-1}, \end{aligned} \tag{15}$$

where $\boldsymbol{C}_p$ denotes the particle velocity gradient, which is reconstructed using the moving least squares approximation with a linear polynomial basis.

Once the particle velocity $\boldsymbol{v}_p$ and velocity gradient $\boldsymbol{C}_p$ are obtained from the grid, the final step within a time increment is to update the particle state, which involves the stress, position, and volume.

The particle position and volume are first updated:

$$\boldsymbol{x}_p^{new} = \boldsymbol{x}_p + \Delta t \cdot \boldsymbol{v}_p, \tag{16}$$

$$V_p^{new} = V_p (1 + \Delta t \cdot (\nabla \cdot \boldsymbol{v}_p)). \tag{17}$$

With the volume updated to V_p^{new}, the new particle density $\rho_p^{new} = m_p / V_p^{new}$ is determined, which allows for the calculation of the pressure component of stress using the equation of state (Eq. (4)). Concurrently, the deviatoric strain rate tensor $\boldsymbol{D}_p'$ is computed from $\boldsymbol{C}_p$, which in turn yields the shear stress $\boldsymbol{\tau}_p$ via the Bingham model (Eq. (7)). The total Cauchy stress $\boldsymbol{\sigma}_p^{new}$ is then assembled by combining the updated pressure and shear stress components.

Compared to traditional MPM, MLS-MPM eliminates the need to compute the gradient of the shape function when calculating the nodal internal force and particle velocity gradient, reducing memory requirements. Furthermore, the mapping scheme, which is the same as APIC, improves the conservation of angular momentum, which is a crucial factor in the simulation of Bingham fluids [62].

2.3. Memory-efficient differentiable programming

The MLS-MPM is equipped with differentiable programming techniques to enable not only forward simulation, but also automatic gradient calculation for the more challenging task of sensitivity analysis. Since the proposed explicit MLS-MPM for 3DCP simulation requires a significant number of time integration steps, a memory-efficient differentiable programming scheme is proposed based on a checkpointing algorithm.

Differentiable programming is based on automatic differentiation (AD) [52] methods. The basic observation is that computer calculation consists of a series of elemental arithmetic operations and functions. By applying the chain rule repeatedly, AD is able to obtain the gradients of a complex function with simple derivative calculations. There are two primary modes to apply the chain rule: forward mode and reverse mode. In this paper, reverse mode is adopted to compute the derivatives. Unlike forward mode, which computes derivatives alongside the forward simulation, reverse mode first completes the entire forward computation, storing intermediate results as needed, and then traverses the computational graph backward to evaluate derivatives. This mode enables the efficient computation of derivatives of a scalar output with respect to multiple input parameters in a single backward propagation, which is particularly advantageous for high-dimensional problems, e.g., sensitivities of multiple parameters can be found all at once.

From the perspective of automatic differentiation, the MLS-MPM algorithm shown in Section 2.2 can be expressed as:

$$\boldsymbol{u}^{n+1} = f(\boldsymbol{u}^n, \boldsymbol{\theta}), \quad \boldsymbol{u}^0 = \boldsymbol{b}, \tag{18}$$

where $\boldsymbol{u}$ is the state vector encapsulating all elementary information of the particles in each time step: position, velocity, deformation gradient, and velocity gradient. Forward operator f represents the entire computational process, consisting of the particle-to-grid (P2G) mapping, grid operations, and grid-to-particle (G2P) updates. Given the explicit time integration scheme used in the developed solver, the discrete adjoint method is employed for the reverse mode differentiation.

The outcome of 3DCP (e.g., the cross-section of printed concrete) can be described with an objective function $J(\boldsymbol{u}^N)$ that relies on the final state of the printing process, whose gradient with respect to $\boldsymbol{\theta}$ can be expressed using the chain rule:

$$\frac{dJ}{d\boldsymbol{\theta}} = \frac{dJ}{d\boldsymbol{u}^N}\frac{d\boldsymbol{u}^N}{d\boldsymbol{\theta}}. \tag{19}$$

Since every state $\boldsymbol{u}^i$ depends on previous state $\boldsymbol{u}^{i-1}$ and $\boldsymbol{\theta}$, with chain rule, Eq. (19) can be further expressed as:

$$\begin{aligned}\frac{dJ}{d\boldsymbol{\theta}} &= \frac{dJ}{d\boldsymbol{u}^N}\left(\frac{\partial\boldsymbol{u}^N}{\partial\boldsymbol{\theta}} + \frac{\partial\boldsymbol{u}^N}{\partial\boldsymbol{u}^{N-1}}\left(\frac{\partial\boldsymbol{u}^{N-1}}{\partial\boldsymbol{\theta}} + \frac{\partial\boldsymbol{u}^{N-1}}{\partial\boldsymbol{u}^{N-2}}\left(\cdots + \frac{\partial\boldsymbol{u}^1}{\partial\boldsymbol{u}^0}\frac{d\boldsymbol{u}^0}{d\boldsymbol{\theta}}\right)\right)\right)\\ &= \frac{dJ}{d\boldsymbol{u}^N}\frac{\partial\boldsymbol{u}^N}{\partial\boldsymbol{\theta}} + \frac{dJ}{d\boldsymbol{u}^N}\frac{\partial\boldsymbol{u}^N}{\partial\boldsymbol{u}^{N-1}}\frac{\partial\boldsymbol{u}^{N-1}}{\partial\boldsymbol{\theta}} + \frac{dJ}{d\boldsymbol{u}^N}\frac{\partial\boldsymbol{u}^N}{\partial\boldsymbol{u}^{N-1}}\frac{\partial\boldsymbol{u}^{N-1}}{\partial\boldsymbol{u}^{N-2}}\frac{\partial\boldsymbol{u}^{N-2}}{\partial\boldsymbol{\theta}} + \cdots\\ &\quad + \frac{dJ}{d\boldsymbol{u}^N}\frac{\partial\boldsymbol{u}^N}{\partial\boldsymbol{u}^{N-1}}\frac{\partial\boldsymbol{u}^{N-1}}{\partial\boldsymbol{u}^{N-2}}\cdots\frac{\partial\boldsymbol{u}^1}{\partial\boldsymbol{u}^0}\frac{d\boldsymbol{u}^0}{d\boldsymbol{\theta}}.\end{aligned} \tag{20}$$

By introducing the set of adjoint variables [63] $\lambda_{\boldsymbol{b}}$ and $\lambda_{\boldsymbol{a}}$:

$$\begin{aligned}\lambda_{\boldsymbol{b}}^{n-1} &= \lambda_{\boldsymbol{b}}^{n}\frac{\partial\boldsymbol{u}^n}{\partial\boldsymbol{u}^{n-1}}, & \lambda_{\boldsymbol{b}}^{N} &= \frac{dJ}{d\boldsymbol{u}^N},\\ \lambda_{\boldsymbol{a}}^{n-1} &= \lambda_{\boldsymbol{b}}^{n}\frac{\partial\boldsymbol{u}^n}{\partial\boldsymbol{\theta}} + \lambda_{\boldsymbol{a}}^{n}, & \lambda_{\boldsymbol{a}}^{N} &= \boldsymbol{0},\end{aligned} \tag{21}$$

it can be shown that the quantity of interest in Eq. (20) exactly matches $\lambda_{\boldsymbol{a}}^0$, i.e., $\frac{dJ}{d\boldsymbol{\theta}} = \lambda_{\boldsymbol{a}}^0$. In other words, the derivative $dJ/d\boldsymbol{\theta}$ can be obtained by backwardly evaluating adjoint variables $\lambda_{\boldsymbol{b}}^i$ and $\lambda_{\boldsymbol{a}}^i$ from N to 0.

While the above derivation provides a theoretical foundation for the reverse-mode differentiation in MPM, its direct application in large-scale numerical simulations is often impractical due to the prohibitive memory requirements. Unlike forward simulation, which only requires information from the previous time step, reverse-mode differentiation demands access to the states and intermediate variables at all time steps to compute gradients. The high requirement on memory poses a significant challenge for large-scale simulations on GPUs, where memory is limited (e.g., 48 GB for the NVIDIA Quadro RTX8000 GPU used in the conducted experiments). This issue is further exacerbated when the weakly compressible assumption is used, which necessitates small time increment. To make the reverse-mode AD practically feasible, a memory-efficient differentiable programming scheme based on checkpointing algorithms is employed.

Checkpointing algorithms are memory-saving techniques that recompute required intermediate variables during the differentiation process, rather than storing all of them [64]. Without checkpointing, reverse-mode AD in MPM would keep the initial and final states of all particles as well as all intermediate nodal variables from every time step. With the checkpointing scheme designed for MPM [65], the entire simulation process with N time steps can be divided into m segments, each consisting of N/m time steps. During the forward simulation, particle states and nodal information of the intermediate N/m time steps are discarded in each segment. Only the initial and final particle states need to be retained. The final particle state of one segment serves as the initial particle state for the next. As the forward simulation of each segment depends only on its initial state, this storage scheme does not impede the forward computation.

For the reverse-mode differentiation process, the adjoint variables in Eq. (21) need to be calculated from the N-th time step to obtain the final gradient. Fig. 4 demonstrates the backward propagation process. Because the objective function is an explicit function of the final stored particle state $\boldsymbol{u}^N$, AD is able to directly calculate the adjoint variables $\lambda_{\boldsymbol{b}}^N$ and $\lambda_{\boldsymbol{a}}^N$. In contrast, AD cannot immediately update other adjoint variables in the last segment (e.g., $\lambda_{\boldsymbol{b}}^{N-1}$, $\lambda_{\boldsymbol{a}}^{N-1}$, $\lambda_{\boldsymbol{b}}^{N-2}$, and $\lambda_{\boldsymbol{b}}^{N-2}$) since the stored values in the last segment are only particle states $\boldsymbol{u}^{N-N/m}$ and $\boldsymbol{u}^N$. Hence, the forward simulation of the last segment has to be executed again with stored particle state $\boldsymbol{u}^{N-N/m}$ to recover the required intermediate particle states and node variables for AD. To save memory, these intermediate particle states and node variables will be discarded after evaluating the adjoint variables $\lambda_{\boldsymbol{a}}^{N-N/m}$ and $\lambda_{\boldsymbol{b}}^{N-N/m}$. Then, AD repeats the same procedure for the preceding segment until the final adjoint variable $\lambda_{\boldsymbol{a}}^0$ is updated, which involves forward simulation to recompute all intermediate variable values, reverse-mode differentiation, and discarding all intermediate variables. In this way, the amount of required memory is $\mathcal{O}(m + N/m)$ (i.e., m particle states and N/m intermediate particle states), as opposed to the original $\mathcal{O}(N)$. In addition, the original requirement can be scaled down to $O(s)$, by setting $s = \sqrt{N}$, making reverse-mode AD feasible for large-scale physical simulations.

2.4. 3DCP simulation model

Having introduced the constitutive model for fresh concrete (Section 2.1) and the MLS-MPM algorithm (Section 2.2), we now detail the specific 3DCP simulation model. To simulate the 3DCP process within the MLS-MPM framework, it is essential to numerically represent the concrete extrusion from the nozzle. In particle-based methods, the extrusion can be modeled by adding new particles to the computational domain at appropriate time intervals. Given the weak compressibility of the material, the concrete can be assumed to be extruded through the nozzle at a constant velocity. The nozzle is considered to be instantly refilled after the previously extruded concrete is fully expelled.

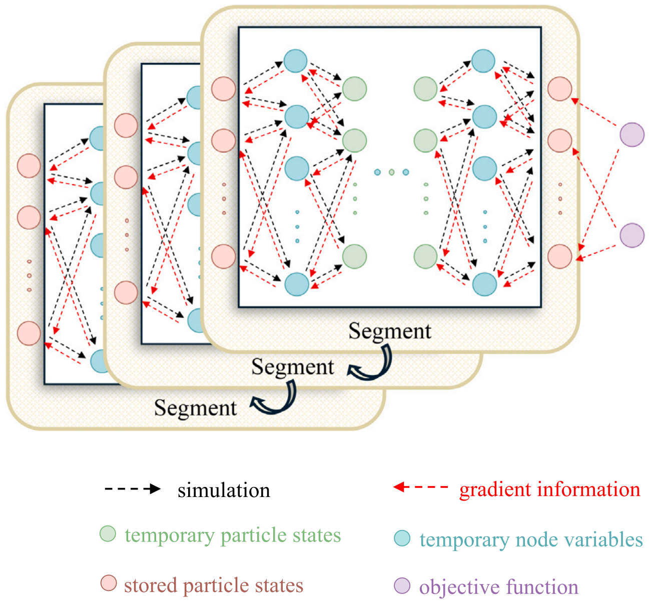

Fig. 4. The reverse-mode differentiation process with checkpointing scheme.

Consequently, the time interval for adding new particles is determined by $h_{\text{nozzle}}/v_{\text{extrusion}}$, where h_{nozzle} is the nozzle height and $v_{\text{extrusion}}$ is the extrusion velocity. In real-world printing, concrete is continuously pumped into the nozzle, forcing out the concrete inside. To replicate this interaction in MLS-MPM, an extrusion velocity is applied to the grid nodes closest to the highest concrete particle in the nozzle at each time step.

The implementation is illustrated in Fig. 5, which also defines the global coordinate system used throughout the 3DCP study. The Z-axis is oriented vertically upwards, opposite to the direction of gravity. The nozzle moves in the printing direction, which is defined as the X-axis, with a prescribed printing velocity. All cross-sectional profile results presented in Section 3 are consequently taken in the Y-Z plane, which is perpendicular to the printing direction. The printing nozzle is not explicitly modeled. Instead, at each time step, its position is updated based on the printing time and velocity. Grid cells whose center points lie within the nozzle radius are identified as inside the nozzle. If the current time matches the printing time interval, new particles will be introduced in these cells with two initial velocity components: a printing velocity along the printing direction, and an extrusion velocity perpendicular to the nozzle. In this paper, the nozzle height is set equal to the grid spacing, Δx.

Besides adding new particles, appropriate boundary conditions should be applied to accurately simulate the 3DCP process. The nozzle moves continuously during the actual printing process, affecting the velocity of the concrete inside the nozzle. Hence, the grid nodes inside the nozzle are assigned the printing velocity at each time step, as shown in Fig. 5. Additionally, the boundary condition between the extruded concrete and the printing platform is set to a no-slip condition, consistent with previous simulations [25,28,33,34]. Table 1 provides a short summary of the algorithm executed at each time step: First, the background grid is reset at the beginning of the step to facilitate the subsequent particle-to-grid (P2G) transfer. Next, the nodal velocity is updated using an explicit time integration scheme. Appropriate boundary conditions are then applied to the nodes based on their positions relative to the nozzle. Finally, the nodal velocity is mapped back to the particles to advance their states.

2.5. The incorrect gradient issue and a smoothing method

Notably, a challenge arises when employing automatic differentiation to compute the gradients of objective functions with respect to the printing velocity within the simulation model from Section 2.4. The reason is that the independent variable-printing velocity-appears in the conditional expression of the “if”statement, which determines the relative position of the nodes and the nozzle. Here, the position of the nozzle is updated based on its initial state and the printing velocity. While AD can track the execution path of each branch within an “if” statement, it does not consider the conditional expression that determines which branch is taken. As a result, unmodified AD code cannot fully capture the effect of the printing velocity on the objective function in this scenario, which is undesired.

Table 1
Summary of the 3DCP simulation algorithm.

```
Algorithm
Reset grid quantities: $m_i = 0$; $(m\boldsymbol{v})_i = 0$; $\boldsymbol{f}_i^{\text{ext}} = 0$; $\boldsymbol{f}_i^{\text{int}} = 0$
Mapping from particles to nodes (P2G)
    Using Eq. (9), compute nodal mass
    Using Eq. (10), compute nodal force
    Using Eq. (14), compute nodal momentum
End mapping
Updating nodal velocity
    Compute nodal velocity: $\boldsymbol{v}_i = (m\boldsymbol{v})_i/m_i$
    Using Eq. (13), update nodal velocity
Applying boundary condition
    If node is inside the nozzle:
        $\boldsymbol{v}_i = \boldsymbol{v}_{\text{printing}}$
        If node is closest to the concrete in $z$ direction:
            $\boldsymbol{v}_i \mathrel{+}= \boldsymbol{v}_{\text{extrusion}}$
    If node is inside the printing platform:
        $\boldsymbol{v}_i = \boldsymbol{0}$
Update particles (G2P)
    Using Eqs. (15), (16), update particle velocities, velocity gradients, and particle positions
    Using Eqs. (17), (4), (6), (5), (3), update stress and volume
End
```

This issue can be better explained using a simplified case of a step function f that increases in its trend but always has a zero derivative. Consider a sequence of real numbers $\{x_i\}_{i=1}^{N}$, where $x_i = i \cdot \Delta x$, with $\Delta x = 10^{-4}$ and $N = 10000$. a function is defined:

$$f(\theta) = \sum_{i=1}^{N} x_i \cdot \mathbf{1}_{x_i<\theta}, \tag{22}$$

where θ is a threshold parameter, and $\mathbf{1}_{x_i<\theta}$ is the indicator function, which equals 1 if $x_i < \theta$, and 0 otherwise. The function $f(\theta)$ represents the sum of all x_i values that are less than the threshold θ. When AD is used to compute the gradient $df/d\theta$, the final result is zero, because the threshold parameter does not appear in the execution path. However, to capture the effect of the threshold parameter over the summation result in a continuous manner, a non-zero gradient would be much useful instead a vanishing gradient.

To alleviate this issue, a strategy is proposed that replaces the “if” statement in such cases with a smooth function approximation. The key idea is that a smooth function can be incorporated into the computational graph – thus enabling automatic differentiation – while also approximating the behavior of an ”if” statement. A sigmoid-like polynomial is used:

$$f(t) = \begin{cases} 1, & t \le 0, \\ 1 - t^2(3 - 2t), & 0 < t < 1, \\ 0, & t \ge 1, \end{cases} \tag{23}$$

where

$$t = \text{clip}\left(\frac{|x - x_{\text{nozzle}}| - \text{bc}_0}{\text{bc}_1 - \text{bc}_0},\ 0,\ 1\right), \tag{24}$$

and $\text{clip}(a, 0, 1) = \min(\max(a, 0), 1)$ restricts a to $[0, 1]$. The parameters bc_0 and bc_1 define the inner and outer bounds of the transition region near the nozzle boundary, respectively. The transition width is controlled by $\delta = \text{bc}_1 - \text{bc}_0$; a larger δ results in a smoother and wider transition. Fig.

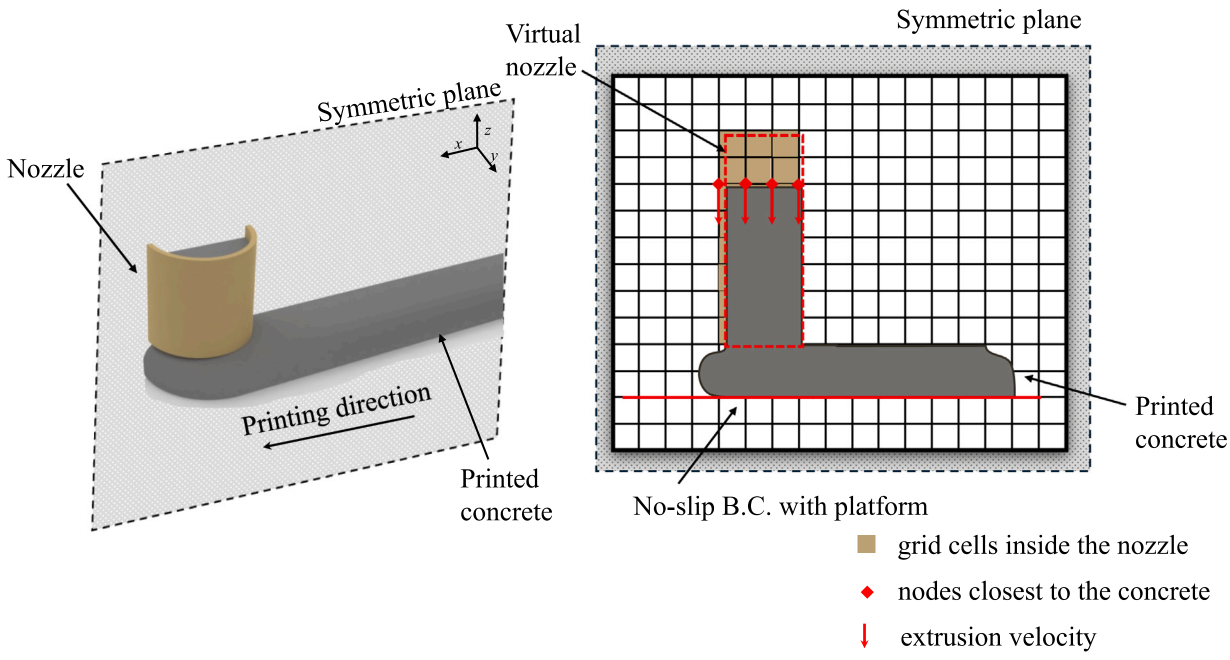

Fig. 5. Numerical settings of MLS-MPM for simulating 3DCP.

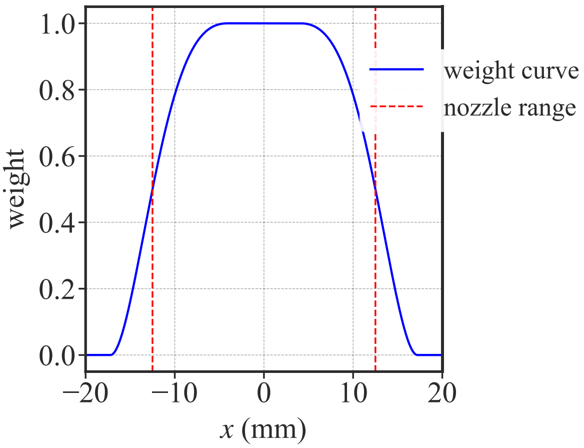

Fig. 6. Smooth function used in 3DCP differentiation process.

Table 2
Modification of the 3DCP simulation algorithm.

```
Algorithm
Reset grid quantities: $m_i = 0$; $(m\boldsymbol{v})_i = 0$; $\boldsymbol{f}_i^{ext} = 0$; $\boldsymbol{f}_i^{int} = 0$
P2G: Same as Algorithm 1 in Table 1
Updating nodal velocity: Same as the algorithm shown in Table 1
Applying boundary condition
    Calculating a weight $w$ based on the
      distance from the node to the nozzle center:
    $w = f_{smooth}(x_{node} - x_{nozzle})$
    $\boldsymbol{v}_i = w \cdot \boldsymbol{v}_{\text{printing}} + (1 - w) \cdot \boldsymbol{v}_i$
    If node is closest to the concrete in $z$ direction:
        $\boldsymbol{v}_i \mathrel{+}= w \cdot \boldsymbol{v}_{\text{extrusion}}$
    If node is inside the printing platform:
        $\boldsymbol{v}_i = \boldsymbol{0}$
G2P: Same as Algorithm 1 in Table 1
```

6 illustrates the weight distribution for $\delta = 0.01$. The red line indicates the nozzle boundary. Inside this region, the weight remains close to 1, while it rapidly decays to 0 outside. With the smooth function, the modified algorithm is presented in Table 2, based on the algorithm shown in Table 1.

3. 3D concrete printing simulation result

Before its application to the 3DCP simulation, the accuracy of the developed MLS-MPM solver is first verified. The verification is accomplished through three numerical benchmarks, for which the detailed setups and validation results are presented in Appendix B.

Having established the credibility of the solver, this section validates the complete 3DCP simulation model. To this end, several printing cases with different printable parameters [25] are simulated. The material properties and printing parameters for these cases are listed in Tables 3 and 4, respectively. The computational domain measures $0.5\ \text{m}\times0.1\ \text{m}\times0.03\ \text{m}$. And eight particles are initially evenly distributed in each cell. To ensure the accuracy and reliability of the simulation results, a convergence study considering both grid size and time step is first conducted using the printing parameters from Case 1. A grid independence study is first performed with three different grid sizes: 2 mm, 1.5 mm, and 1 mm, while the time step is held constant at 5×10^{-7} s. As shown in Fig. 7(a), three grid resolutions produce similar cross-sectional profiles, indicating that the simulation results are approaching grid convergence. Although the differences are minor, the 1 mm grid shows a marginal refinement in capturing the curvature of the profile. Then a time step sensitivity analysis is conducted with the grid resolution fixed at 1 mm. Three time steps, 1×10^{-6} s, 5×10^{-7} s, and 2.5×10^{-7} s, are evaluated. The results, presented in Fig. 7(b), demonstrate consistency, with all three simulations yielding nearly identical profiles, which confirms that the solution is converged with respect to the time step in the tested range. Given that both studies show good convergence, the final parameters are 1 mm for grid size and 5×10^{-7} s for time step.

With the grid resolution and time step established, the simulation model is validated against experimental data for various printing conditions. Fig. 8(a) presents the velocity and pressure distribution at $t = 3$ s for case 1. Consistent with [25], most deposited concrete slows to near-zero velocity, while the newly extruded concrete continues to flow. The flow behavior is governed by the pressure gradient around the moving nozzle. Due to the confinement of deposited concrete, a

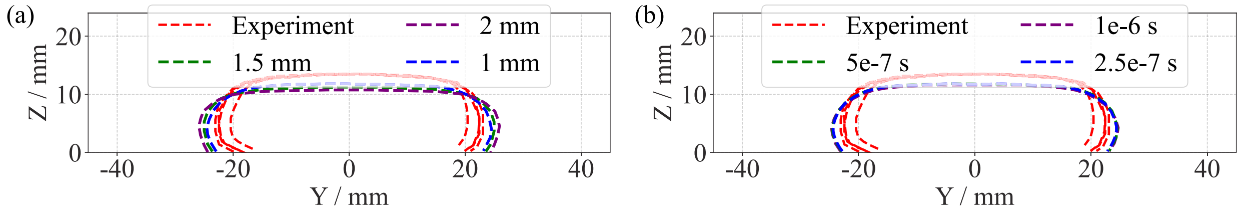

Fig. 7. Convergence study for grid resolution and time step: (a) comparison of cross-sectional profiles obtained with different grid resolutions (1 mm, 1.5 mm, and 2 mm). (b) comparison of profiles obtained with different time steps (1×10^{-6} s, 5×10^{-7} s, and 2.5×10^{-7} s). Both plots are compared against experimental data from [25].

high-pressure region develops at the rear of the nozzle, which creates a pressure gradient that drives the material toward the lower-pressure region at the front. Under this pressure distribution, the velocity field achieves its maximum at the nozzle front. The velocity and pressure distribution demonstrates the buildability required in 3DCP: the material should be easily extrudable yet stabilize quickly. Fig. 8(b)–(f) illustrate the simulated cross-section shapes under different printing parameters, in comparison with the reported results [25], which include both experimental and numerical data. In the reported simulations, the generalized Newtonian fluid (GNF) model and the elasto-visco-plastic (EVP) fluid model are employed. To provide a quantitative validation, the relative errors between the simulation and the experimental results are summarized in Table 5. In this analysis, the width is defined as the maximum extent of the cross-section in the y-direction, and the height is the maximum z-coordinate value. The results show that all calculated errors remain below 20%, indicating a good overall agreement with the experimental data. And a consistent underestimation is observed in the layer height, which constitutes the primary source of discrepancy. A possible reason for this discrepancy is the weakly compressible assumption, which may reduce the total volume, reflected by the height underprediction of the printed layer. Although the assumption has been adopted in previous studies [4,33,35], it is not fully consistent with the nearly incompressible nature of concrete, and a quasi-incompressible model may provide better agreement with experimental results. Overall, the results confirm that the MLS-MPM-based simulation model can effectively simulate the 3DCP process and accurately capture its key physical features. The simulation is executed on a NVIDIA Quadro RTX8000 GPU, and for each case it requires approximately 23 h, with a $500 \times 300 \times 100$ grid, a total number of approximately 800,000 particles, and 1×10^7 time integration steps. Engineering printed structures often consist of dozens or even hundreds of layers. Therefore, it is crucial to develop reduced-order models to lower the cost of the full-order MPM simulations, which is left for future work.

As a representative case of multi-layer printing, a three-layer concrete structure is numerically printed. Each layer is 200 mm long, and the printing parameters match those in Case 1. Fig. 9(a) shows the velocity distribution during the printing process. The velocity profile of the top layer resembles that of a single-layer print. However, in the second layer, the region directly beneath the nozzle has a non-zero velocity due to the additional pressure from the extruded concrete. In contrast, other areas experience only self-weight and the load from upper layers, which suggests that this nozzle-induced pressure is significantly greater than gravity. Fig. 9(b), (c) and (d) show the cross-sectional shapes after printing one, two, and three layers, along with the literature results from [35,66]. A notable phenomenon is that the MLS-MPM shows good agreement with the experimental data in the case of one-layer printing. When the second and third layers are numerically printed, the MLS-MPM results deviate more from experimental data, while the EVP fluid model remains accurate. A similar trend is observed in the simulation results of [35], where the Bingham fluid model and the weakly compressible assumption are also employed within a PFEM solver to simulate the multi-layer printing process. One possible reason is that the weakly compressible assumption is employed in the MLS-MPM simulation model, which may not only lead to greater deformation but also result in the accumulation of errors, especially as the number of printed layers increases. Another reason is that the EVP model exhibits elastic response under stresses below the yield threshold, allowing it to resist deformation. In contrast, the pseudo-rigid body behavior of Bingham model relies on high viscosity, causing it to continue deforming under shear forces.

Overall, the proposed MLS-MPM approach yields satisfactory results in the single-layer and multi-layer printing cases under various process parameters. These numerical tests demonstrate the accuracy and reliability of MLS-MPM in the forward simulation tasks of 3DCP, paving the way for the more challenging inverse sensitivity analysis.

Table 3
Material parameters for printing cases, adapted from [25].

Parameter	Value
Density (kg/m^3)	2100.0
Yield Stress (Pa)	630.0
Viscosity (Pa s)	7.5
Bulk Modulus (Pa)	1.0×10^6

Table 4
Printing parameters for Cases 1–5, adapted from [25].

Parameter	Case 1	Case 2	Case 3	Case 4	Case 5
Printing velocity (mm/s)	30.0	30.0	40.0	30.0	50.0
Extrusion velocity (mm/s)	33.6	38.2	31.6	26.7	33.4
Nozzle height (mm)	12.5	7.5	12.5	17.5	12.5

Table 5
Relative errors between simulation and experimental results from [25] for cases 1–5.

Error type	Case 1	Case 2	Case 3	Case 4	Case 5
Width error (%)	10.99	3.00	7.64	11.68	14.99
Height error (%)	−12.40	−9.64	−5.33	−11.37	−18.89
Cross section error (%)	4.92	5.59	2.51	2.83	4.24

4. Automatic sensitivity analysis

This section presents automatic sensitivity analyses conducted based on the differentiable MLS-MPM solver. Compared to traditional empirical parametric studies and finite-difference-based sensitivity analyses, the differentiable numerical framework provides a more accurate and efficient quantification of how printing parameters influence the printing results, laying the foundation for possibly solving inverse problems such as parameter identification and process optimization. The accuracy of the automatic gradient computations of the developed solver is first validated through several numerical benchmarks. Then, automatic sensitivity analyses are conducted for the 3DCP process to demonstrate the ability of the proposed solver to quantify the effects of key printing parameters.

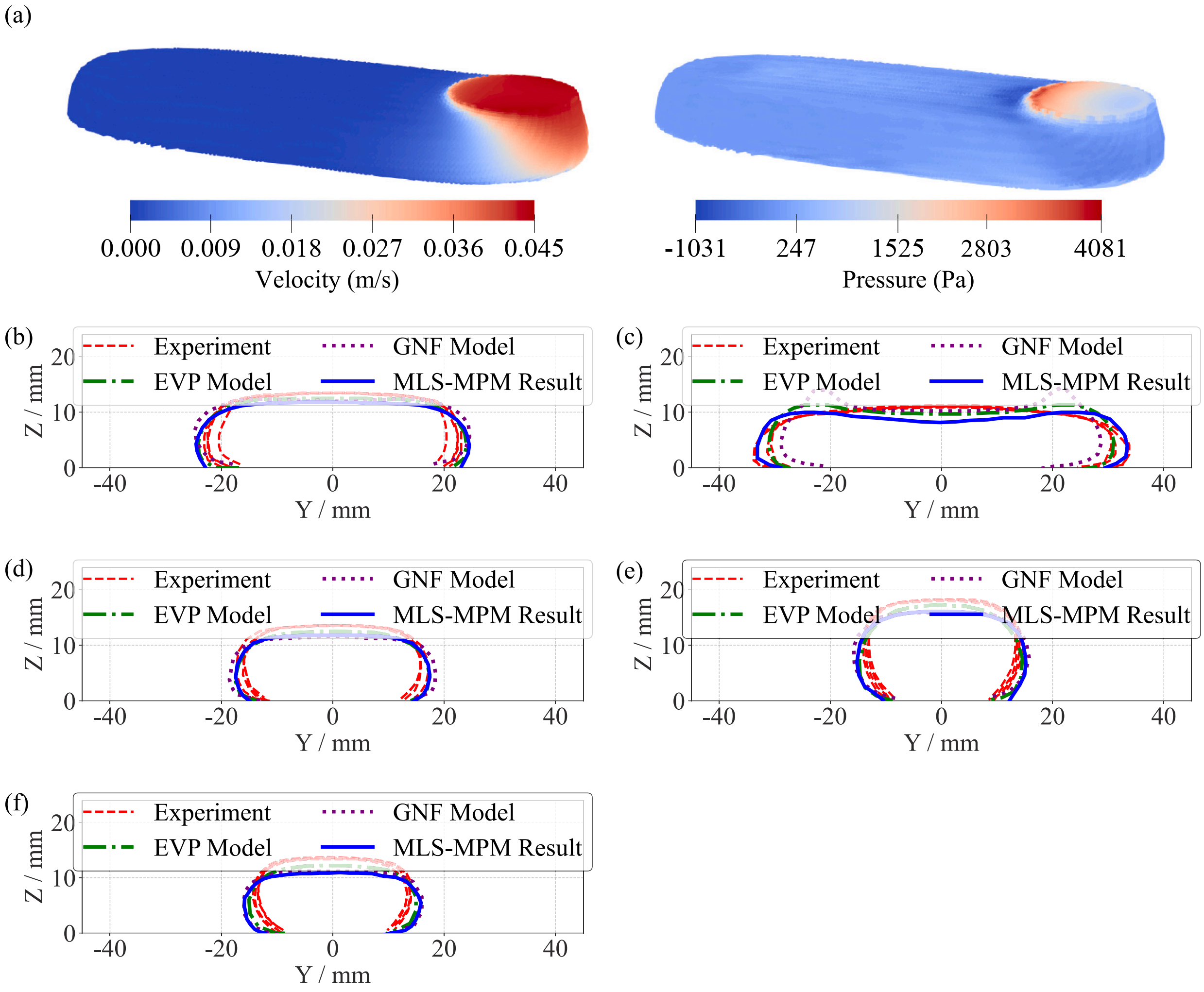

Fig. 8. Velocity, pressure distribution, and cross-section shapes with different printing parameters: (a) final velocity and pressure distribution; (b) case 1; (c) case 2; (d) case 3; (e) case 4; (f) case 5. Blue line: MLS-MPM simulation result; dashed line: simulation and experimental results from [25], where GNF is the generalized Newtonian fluid model and EVP is the elasto-visco-plastic fluid model.

4.1. Sensitivity analysis for numerical benchmarks

To verify the reliability of the automatic differentiation feature of the solver, two numerical cases from Appendix B are examined.

The first case is the Poiseuille flow. The forward simulation in Appendix B.1 demonstrates that the velocity profile is parabolic, with the maximum value occurring at the center. Therefore, the quantity of interest is the gradient of the maximum velocity at $t = 1$ s with respect to the driving force. Despite the large number of time increments (1×10^5 time steps), the developed solver is able to complete the reverse-mode differentiation process, owing to the checkpointing technique detailed in Section 2.3.

To assess the correctness of the automatic differentiation results, two approaches are used. First, the derivative computed by AD is compared with that obtained via the central difference (CD) method by evaluating the relative error:

$$\text{Relative Error} = \frac{|D_{\text{AD}} - D_{\text{CD}}|}{|D_{\text{CD}}|} \times 100\%, \tag{25}$$

where D_{AD} and D_{CD} represent the derivatives from automatic differentiation and the central difference method, respectively. In this case, the derivative derived from the analytical velocity solution [67] is used in place of the central difference derivative in Eq. (25) for the comparison with the AD result. The resulting relative error is 0.6%, indicating a high level of accuracy.

The second approach is the Taylor remainder convergence test, which has been adopted to verify the derivative result in several previous works [68–70]. For a perturbation $\delta\theta$, the first-order Taylor expansion predicts second-order convergence for the remainder:

$$r_{\text{first}} = \left| J(\theta + h\,\delta\theta) - J(\theta) - h\,\frac{dJ}{d\theta}\cdot\delta\theta \right| \to 0 \quad \text{at } O(h^2). \tag{26}$$

Here, $\delta\theta$ is a unit vector defining the direction of the perturbation, and h is a dimensionless scaling factor. The term $h\,\delta\theta$ represents an absolute perturbation applied to the parameter θ. The numerical value of h is equivalent to the magnitude of this absolute perturbation in the corresponding physical units. For the Poiseuille flow case, the parameter of interest is the driving force. Accordingly, $\delta\theta$ represents a unit perturbation of 1 N/m^2. The scaling factors h used here are 10^{-4}, 10^{-5}, 10^{-6}, and 10^{-7}, successively reduced by an order of magnitude. These values of h directly correspond to absolute perturbations of 10^{-4} N/m^3, 10^{-5} N/m^3, etc., applied to the baseline driving force. Accordingly, the remainders are expected to decrease by two orders of magnitude, corresponding to a straight line with a slope of 2 on a log–log plot. Fig. 10 displays the remainder trend. The remainder curve is not parallel to the second-order reference. This result is expected, since analytically the maximum velocity should have a linear relation with driving force, resulting in a theoretically zero Taylor remainder. In this case, the observed remainder is on the order of 10^{-15}, which is within double-precision machine accuracy and can be regarded as zero.

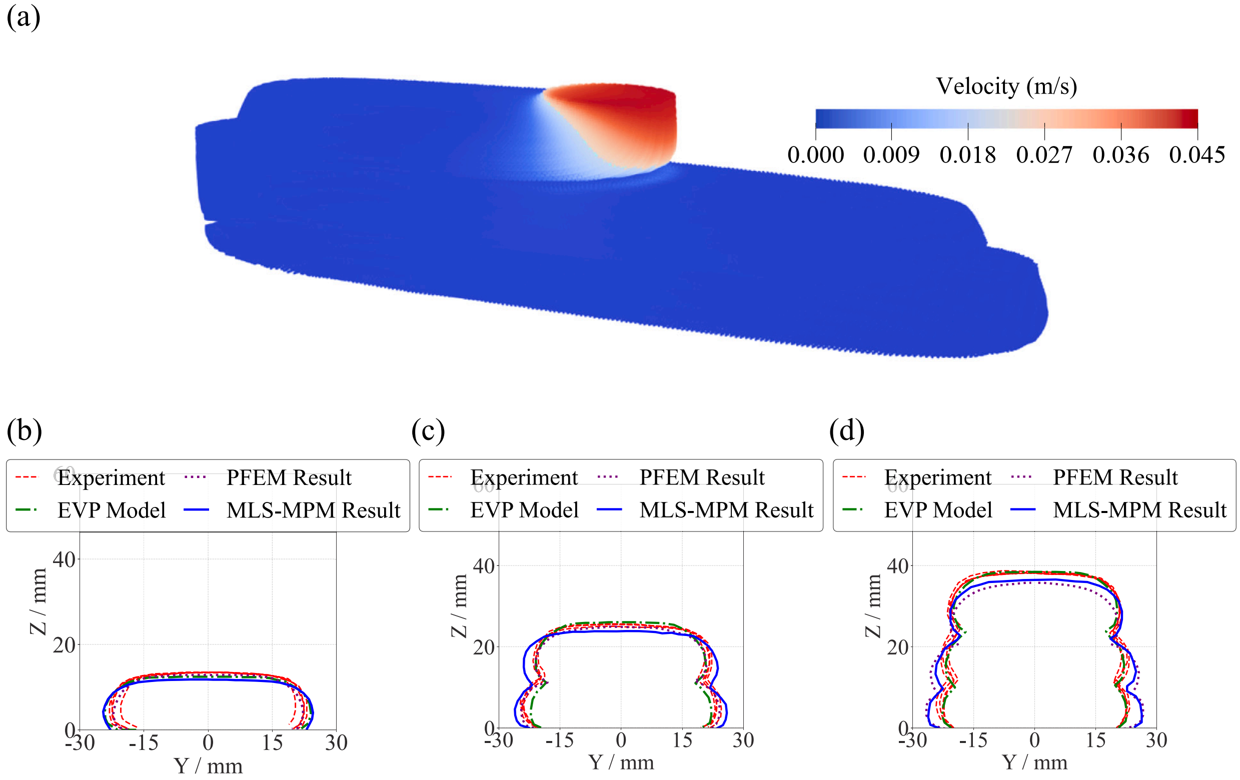

Fig. 9. Simulation result of multi-layer printing: (a) velocity distribution; (b) cross-section shape of first layer; (c) cross-section shape of two layers; (d) cross-section shape of three layers. Blue line: MLS-MPM simulation result; red line: experimental result from [66]; green line: simulation result from [66] with EVP being the elasto-visco-plastic fluid model; purple line: simulation result from [35] with PFEM being the particle finite element method.

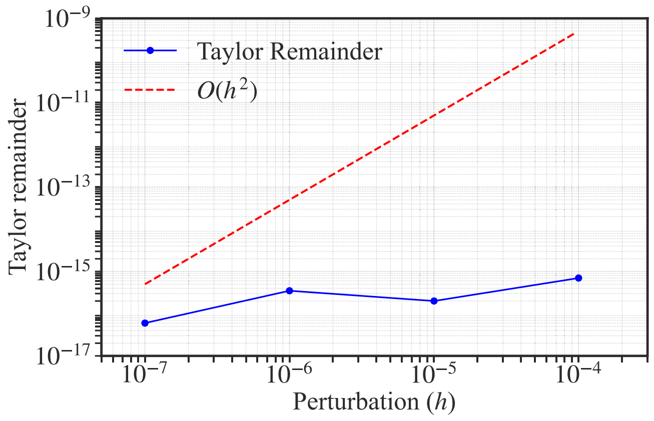

Fig. 10. Taylor remainder test result of Poiseuille Flow.

The second numerical example is the slump test from Appendix B.3. The gradient of interest here is the derivative of the maximum displacement in the x direction at $t = 5$ s (5×10^5 time steps) with respect to the yield stress, a key rheological parameter for viscoplastic flow. For reverse-mode differentiation, mix C is used, with a yield stress of 130 Pa.

Fig. 11(a) presents the relative error between the AD and central difference derivatives. The relative error remains at a very low level across different step sizes, confirming the accuracy of the AD implementation. Fig. 11(b) illustrates the Taylor remainder obtained for different values of the perturbation h. Considering the order of magnitude of the yield stress, the perturbations used are 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}. As h decreases, the remainder decreases proportionally to the square of the change in perturbation, verifying the correctness of the computed gradients. These results underscore the effectiveness and robustness of automatic differentiation for sensitivity analysis in complex non-Newtonian flow problems.

4.2. Sensitivity analysis for 3DCP

Printing parameters play a crucial role in the extrusion flow process. With a validated forward simulation model, automatic differentiation enables an efficient and quantitative evaluation of these effects. To illustrate this ability, case 5 in Section 3 is considered. A key geometric feature, the cross-sectional width of the printed concrete (defined as the maximum width), is chosen as the objective function. Printing velocity and extrusion velocity are treated as the independent variables affecting the objective function.

As the printing velocity appears in the conditional expression of the “if”statement in the simulation model, the smoothing technique detailed in Section 2.5 is employed to calculate the gradient. The final computed gradients with respect to the printing velocity and extrusion velocity are −0.43634 s and 0.69724 s, respectively. Fig. 12 shows the relative error between the AD and central difference derivatives from the smoothed simulation model. The consistently low relative error for various step sizes demonstrates the capability of automatic differentiation to provide highly accurate gradients for the given differentiable model. In addition, the sign of the derivatives further reflects the underlying physical mechanisms. Increasing the printing velocity will lead to a decrease in width, as reflected by the negative derivative. The reason is that at higher printing velocities, the nozzle moves faster and allows less time for the extruded concrete to flow under the applied pressure, causing the material to better retain its original shape. This trend is consistent with previous parametric studies [25]. In contrast, the width increases with extrusion velocity, as indicated by the positive derivative. Higher extrusion velocity leads to greater pressure at the nozzle, delivering more material and resulting in a wider printed track.

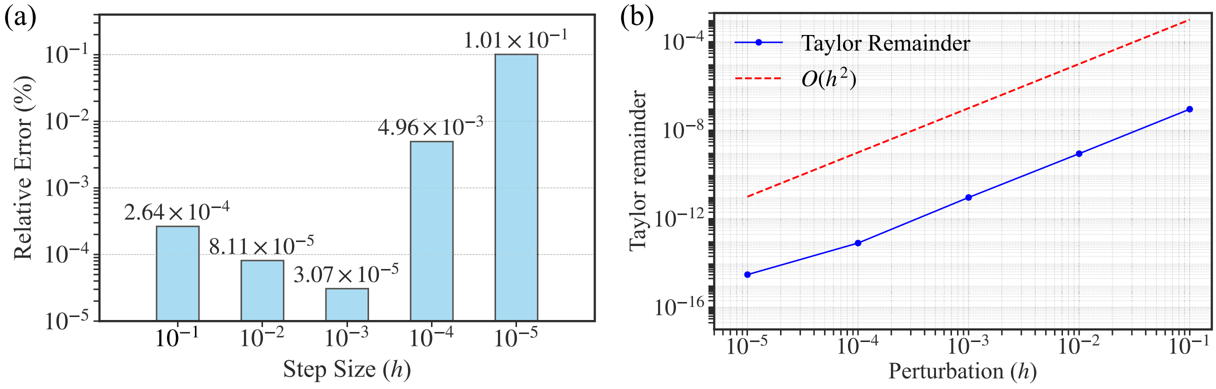

Fig. 11. Automatic differentiation validation of slump test: (a) relative error; (b) Taylor remainder test.

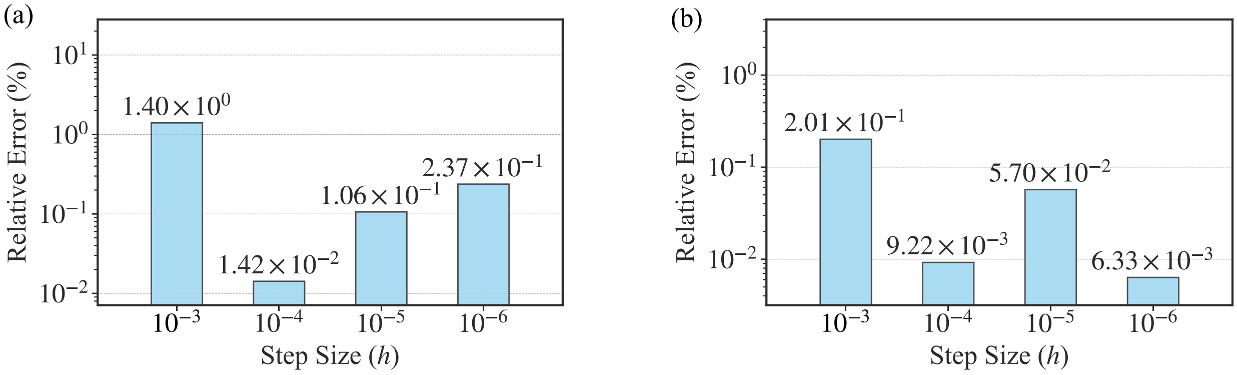

Fig. 12. Automatic differentiation validation of 3DCP: (a) printing velocity; (b) extrusion velocity.

To demonstrate the necessity and reliability of the smoothing method, a comprehensive comparison against the original model with the "if" statement is performed and the results are compiled in Table 6. First, the smoothing approximation has a negligible impact on the predicted result of the forward simulation, as demonstrated by the agreement with experimental data in Fig. 13. Because the forward simulation results are preserved, the sensitivities of parameters not directly involved in the "if" statement remain largely unaffected, which is confirmed by the gradient of extrusion velocity. Indeed, the gradient from the smooth approximation is in good agreement with the gradient values obtained from both the central difference method and a direct AD application on the original "if" statement model, showing a relative difference of about 4%.

The situation is entirely different for the printing velocity. A naive application of AD to the original model fails to capture the full contribution of the printing velocity which appears in the conditional expression of the "if" statement, yielding an incorrect gradient of 0.30384 s. In contrast, the smooth approximation rectifies the fundamental issue, providing a gradient of −0.43634 s. The corrected value aligns well with the central difference result, demonstrating the accuracy of the smooth function.

A sensitivity analysis is performed to evaluate the efficiency of AD for multi-parameter problems in 3DCP. As illustrated in Fig. 14, the strand width in the initial region deviates from the final, stable value. For optimal part quality and geometric accuracy, the strand width throughout the transient region ought to be identical to the final, stable value. Accordingly, an objective function is defined as the sum of the widths at three equidistant cross-sections, measured at 8 mm, 16 mm, and 24 mm downstream from the leading edge of the printed strand. The design variables are the printing and extrusion velocities, which are parameterized as piecewise constant functions over four uniform 0.75 s intervals spanning the initial 3 s of the process. The specific values for these velocity parameters are adopted from case 1, as presented in Section 3. The gradients of the objective function with respect to these eight variables are calculated using AD and the central difference method.

Table 6
Comparison of gradients from three methods: AD with the smooth function, Central Difference with an "if" statement, and AD with an "if" statement.

Method	Printing Velocity (s)	Extrusion Velocity (s)
AD (with the smooth function)	−0.43634	0.69724
Central Difference		
$h = 10^{-3}$	−0.43532	0.66977
$h = 10^{-4}$	−0.42551	0.67098
$h = 10^{-5}$	−0.42575	0.67135
AD (with an "if" statement)	0.30384	0.67131

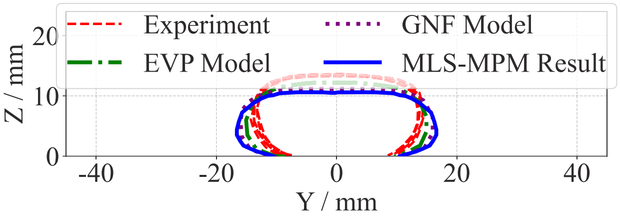

Fig. 13. Validation of the smooth approximation: comparison with experimental and previous simulation results from [25].

Fig. 15 presents the relative error, which is generally low, with a maximum value below 2%. While providing comparable accuracy, AD requires only 14.4 h for a single forward and backward pass, whereas the central difference method takes approximately 24 h to perform the 16 forward simulations needed for the same 8 variables.

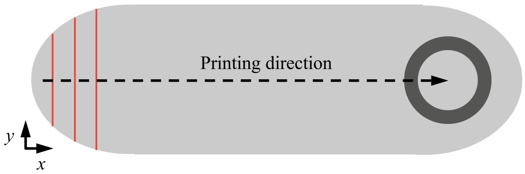

Fig. 14. Top-down schematic of 3DCP.

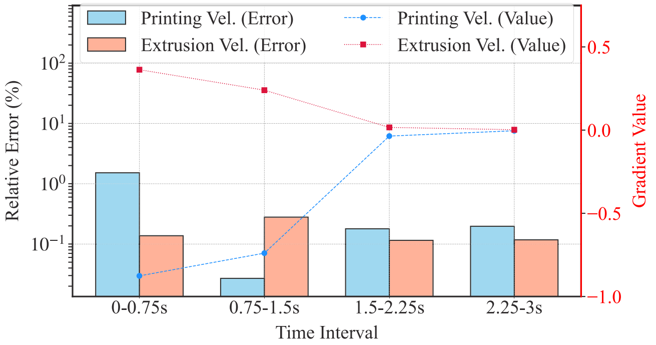

Fig. 15. Automatic differentiation result and validation about the width in the initial region.

And the computational cost of AD is largely independent of the number of parameters, making its efficiency gain even more pronounced as the complexity of the problem increases. Fig. 15 also illustrates the gradient values obtained from AD. The magnitude of the gradients, for both printing and extrusion velocity, approaches zero for later time intervals. The reason is that process parameters in later time intervals (e.g., 2.25–3 s) have a diminishing influence on the strand width in the initial deposition region (measured from 8 mm to 24 mm).

These results show that the developed numerical framework can accurately and efficiently evaluate the influence of printing parameters on the final outcome, demonstrating the potential of the differentiable MLS-MPM solver in efficiently solving inverse problems in 3DCP, such as material property identification and process parameter optimization, which will be conducted as future work.

5. Conclusions

This work presents a differentiable material point method framework designed for the simulation and sensitivity analysis of 3DCP. The main outcome is an efficient, gradient-enabled solver, which accurately computes the sensitivity of the final printed geometry with respect to process parameters. The forward simulation capability of the solver is validated against both classical fluid dynamics benchmarks and experimental data for single-layer 3DCP deposition.

Despite its capabilities, the framework has notable limitations that must be acknowledged. The most significant is the excessive computational cost inherent in high-fidelity simulations of this nature. Furthermore, the accuracy of the model diverges from experimental results in multi-layer printing scenarios, which is primarily attributed to the current weakly compressible Bingham constitutive model.

However, the potential of this differentiable framework lies in its application to solve complex inverse problems that are currently intractable or prohibitively expensive. A prime example is accurately determining concrete material parameters, such as yield stress and viscosity, which play a noticeable role in the extrusion process. By providing efficient access to gradients, our framework has the potential to calibrating these complex rheological parameters by inversely matching simulated filament shapes to experimental ones. Furthermore, the approach could be extended to inverse process design, enabling the automatic determination of optimal printing parameters to achieve specific target geometries.

Building on this work, our future efforts will proceed in two primary, grounded directions to address the aforementioned limitations. First, to enhance model fidelity and address the divergence in multi-layer simulations, the priority is to incorporate more advanced constitutive models. We will implement an elasto-visco-plastic fluid model within the differentiable solver to better predict the multi-layer printing result. Second, to accelerate computation, a reduced-order model will be developed. The observation that deposited regions far from the nozzle exhibit highly similar and stable physical states presents a clear opportunity for model reduction. We will explore techniques to replace these computationally expensive regions with an efficient surrogate, drastically reducing simulation time without sacrificing accuracy near the critical nozzle-deposition zone, thus enabling the simulation of industrial-scale printing scenarios. By pursuing these targeted improvements, we aim to develop a robust, fast, and physically accurate differentiable framework that can significantly accelerate the design, optimization, and adoption of 3D Concrete Printing technology.

CRediT authorship contribution statement

Wenchang Zhang: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Chuanqi Liu:** Writing – review & editing, Methodology. **Yiwei Weng:** Writing – review & editing, Methodology. **Tianju Xue:** Writing – review & editing, Supervision, Software, Resources, Methodology, Investigation, Funding acquisition, Conceptualization. **Jidong Zhao:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the support from the Young Collaborative Research Grant (YCRG) by the Research Grants Council of Hong Kong (Project No. C6002-24Y).

Appendix A. Derivation of the MLS-MPM formulation

This appendix provides a detailed derivation of the MLS-MPM method. The starting point is the weak form of Eq. (2), which can be obtained by multiplying a test function $\delta \boldsymbol{u}$ and integrating over the object domain:

$$\int_V \rho \boldsymbol{a} \cdot \delta \boldsymbol{u} \, dV = -\int_V \boldsymbol{\sigma} : (\nabla \delta \boldsymbol{u}) \, dV + \int_S \boldsymbol{t} \cdot \delta \boldsymbol{u} \, dS + \int_V \rho \boldsymbol{b} \cdot \delta \boldsymbol{u} \, dV, \quad \text{(A.1)}$$

where integration by parts has been applied and $\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$ is the traction applied on the boundary.

Like most MPM variants, particle quadrature is adopted in MLS-MPM and particles serve as integration points. The integrals over the domain are approximated by summing the integrand evaluated at the particle locations, weighted by the volumes of the particles:

$$\sum_{p=1}^{n_p} m_p \boldsymbol{a}_p \cdot \delta \boldsymbol{u}_p = -\sum_{p=1}^{n_p} \boldsymbol{\sigma}_p : (\nabla \delta \boldsymbol{u}_p) V_p + \sum_{p=1}^{n_p} \boldsymbol{t}_p \cdot \delta \boldsymbol{u}_p V_p h^{-1} + \sum_{p=1}^{n_p} m_p \boldsymbol{b}_p \cdot \delta \boldsymbol{u}_p, \quad \text{(A.2)}$$

where n_p is the total number of the particles, the subscript p denotes the function value at the particle position $\boldsymbol{x}_p$, V_p is the particle volume, and h is the boundary layer thickness. Further simplification requires space discretization with a shape function. In MLS-MPM, the MLS shape function [50] for node i, constructed around particle p, is employed for any point $\boldsymbol{x}$ in space:

$$\boldsymbol{\Phi}_i(\boldsymbol{x}) = \xi_i(\boldsymbol{x}_p)\boldsymbol{P}^T(\boldsymbol{x}-\boldsymbol{x}_p)\boldsymbol{M}^{-1}(\boldsymbol{x}_p)\boldsymbol{P}(\boldsymbol{x}_i-\boldsymbol{x}_p), \tag{A.3}$$

where $\boldsymbol{\Phi}_i$ is the MLS shape function of node i and ξ is the weighting function of the moving least squares approximation. $\boldsymbol{P}$ is the linear polynomial basis, which is $[1, x, y, z]$ in three dimensional case. $\boldsymbol{x}_p$ denotes the position of particle p, while $\boldsymbol{x}_i$ represents the position of node i. The moment matrix $\boldsymbol{M}$ is:

$$\boldsymbol{M} = \sum_i \xi_i(\boldsymbol{x}_p)\begin{bmatrix} 1 & (\boldsymbol{x}_i-\boldsymbol{x}_p)^T \\ (\boldsymbol{x}_i-\boldsymbol{x}_p) & (\boldsymbol{x}_i-\boldsymbol{x}_p)(\boldsymbol{x}_i-\boldsymbol{x}_p)^T \end{bmatrix}. \tag{A.4}$$

The quadratic B-spline function, detailed in Section 2.2, is employed as the weighting function ξ [60]. As a result, the structure of the moment matrix $\boldsymbol{M}$ is significantly simplified. In general, any weighting function satisfying the properties of partition of unity ($\sum_i \xi_i = 1$) and linear consistency ($\sum_i \xi_i(\boldsymbol{x}_i - \boldsymbol{x}_p) = \boldsymbol{0}$) reduces $\boldsymbol{M}$ to a block-diagonal form, with a 1 in the top-left entry and zero off-diagonal blocks. Specifically, the employed quadratic B-spline kernel simplifies $\boldsymbol{M}$ further into a fully diagonal matrix, as demonstrated in [50]:

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{4}\Delta x^2 & 0 & 0 \\ 0 & 0 & \frac{1}{4}\Delta x^2 & 0 \\ 0 & 0 & 0 & \frac{1}{4}\Delta x^2 \end{bmatrix}. \tag{A.5}$$

With the MLS shape function, the acceleration and test function value at particle p can be expressed by nodal values:

$$\begin{aligned} \boldsymbol{a}_p &= \sum_{i=1}^{n_i} \boldsymbol{\Phi}_{ip}\boldsymbol{a}_i, \\ \delta\boldsymbol{u}_p &= \sum_{i=1}^{n_i} \boldsymbol{\Phi}_{ip}\delta\boldsymbol{u}_i, \end{aligned} \tag{A.6}$$

where n_i is the total number of grid nodes and $\boldsymbol{\Phi}_{ip}$ is the MLS shape function value of node i at the particle position $\boldsymbol{x}_p$, $\boldsymbol{\Phi}_i(\boldsymbol{x}_p)$. Due to the simple form of the moment matrix $\boldsymbol{M}$ and the linear polynomial basis $\boldsymbol{P}$, the following relation holds:

$$\boldsymbol{\Phi}_{ip} = \xi_i(\boldsymbol{x}_p)\boldsymbol{P}^T(\boldsymbol{x}_p-\boldsymbol{x}_p)\boldsymbol{M}^{-1}(\boldsymbol{x}_p)\boldsymbol{P}(\boldsymbol{x}_i-\boldsymbol{x}_p) = \xi_i(\boldsymbol{x}_p). \tag{A.7}$$

With Eqs. (A.3), (A.6), (A.7) and the arbitrariness of the vector-valued test function, Eq. (A.2) can be simplified to:

$$\begin{aligned} \sum_{p=1}^{n_p}\sum_{j=1}^{n_i} m_p\xi_{ip}\xi_{jp}\boldsymbol{a}_j = &-\sum_{p=1}^{n_p}\frac{4}{\Delta x^2}\xi_{ip}\boldsymbol{\sigma}_p\cdot(\boldsymbol{x}_i-\boldsymbol{x}_p)V_p \\ &+\sum_{p=1}^{n_p}\xi_{ip}\boldsymbol{t}_p V_p h^{-1} + \sum_{p=1}^{n_p}\xi_{ip}m_p\boldsymbol{b}_p. \end{aligned} \tag{A.8}$$

Eq. (A.8) can also be written as:

$$m_{ij}\boldsymbol{a}_j = \boldsymbol{f}_i^{int} + \boldsymbol{f}_i^{ext}, \tag{A.9}$$

where m_{ij}, $\boldsymbol{f}_i^{int}$, and $\boldsymbol{f}_i^{ext}$ are nodal mass, internal force, and external force:

$$\begin{aligned} m_{ij} &= \sum_{p=1}^{n_p} m_p\xi_{ip}\xi_{jp}, \\ \boldsymbol{f}_i^{int} &= -\sum_{p=1}^{n_p}\frac{4}{\Delta x^2}\xi_{ip}(\boldsymbol{\sigma}_p\cdot(\boldsymbol{x}_i-\boldsymbol{x}_p))V_p, \\ \boldsymbol{f}_i^{ext} &= \sum_{p=1}^{n_p}\xi_{ip}\boldsymbol{t}_p V_p h^{-1} + \sum_{p=1}^{n_p}\xi_{ip}m_p\boldsymbol{b}_p. \end{aligned} \tag{A.10}$$

All nodal mass m_{ij} form the consistent mass matrix. In this study, the consistent mass matrix is simplified to a lumped mass matrix, which is a diagonal matrix to improve computational efficiency. The diagonal terms are given by:

$$m_i = \sum_{j=1}^{n_i} m_{ij} = \sum_{p=1}^{n_p} m_p\xi_{ip}. \tag{A.11}$$

Table B.1
Material properties for different concrete mixes, adapted from [44].

	Yield Stress (Pa)	Viscosity (Pa s)
Mix A	54.0	35.8
Mix B	61.0	29.0
Mix C	130.0	30.2

Appendix B. Numerical benchmarks

In this appendix, several numerical examples, including Poiseuille flow, dam break and concrete slump test, are conducted to demonstrate the capability of the developed MLS-MPM solver in solving fluid mechanics problems involving both Newtonian and non-Newtonian flow behaviors.

B.1. Poiseuille flow

The first numerical example is the plane Poiseuille flow, where a viscous fluid flows through a long, straight tube with a constant circular cross-section under a pressure difference. A two-dimensional model, shown in Fig. B.1(a), is used to represent the flow. The computational domain measures 1 mm × 1 mm and is discretized into a 100 × 100 grid. Each grid cell contains four particles, totaling 40,000 particles.

Since the pipe length is typically greater than its radius in Poiseuille flow, a periodic boundary condition is applied at the inlet and outlet. The upper and lower boundaries impose a no-slip boundary condition, ensuring the velocity at the pipe wall remains zero. Water is used as the working fluid, with a density of $1000\ \mathrm{kg/m^3}$ and a viscosity of $0.001\ \mathrm{Pa\cdot s}$. To account for weak compressibility, a bulk modulus of $1\times10^5\ \mathrm{Pa}$ is applied. Instead of directly imposing a pressure difference, a body force $F = 8\ \mathrm{mm/s^2}$ is added in the x-direction, mimicking the pressure gradient. Gravity is neglected in this example.

The total flow time is $1\ \mathrm{s}$ with a time step of $1\times10^{-5}\ \mathrm{s}$, which is enough to allow the flow to evolve from rest to a steady state. Fig. B.1(b) presents the final velocity distribution, where the maximum velocity is observed at the center region, while zero velocity is maintained near the pipe wall. Fig. B.1(c) plots the horizontal velocity against y-coordinates at different time steps. The velocity profiles exhibit a parabolic shape, closely matching the analytical solution [67], confirming the accuracy of MLS-MPM for this fluid mechanics benchmark.

B.2. Dam break

The dam break is a well-known benchmark for mesh-free methods [71–74]. In this scenario, a viscous fluid initially at rest is confined by a dam. When the dam is removed, the fluid flows freely under gravity, creating a free surface flow problem. A 3D geometrical model, consistent with Lobovsky's experiment [75], is established, as shown in Fig. B.2(a). The computational domain, matching the tank dimensions, measures $1.6\ \mathrm{m}\times0.8\ \mathrm{m}\times0.25\ \mathrm{m}$, while the initial fluid block occupies $0.6\ \mathrm{m}\times0.6\ \mathrm{m}\times0.15\ \mathrm{m}$. The domain is discretized into regular cubic cells of size $0.01\ \mathrm{m}\times0.01\ \mathrm{m}\times0.01\ \mathrm{m}$. Eight particles are evenly distributed in each cell, totaling $432{,}000$ particles.

All tank walls impose a slip boundary condition, meaning the normal velocity is zero, while the tangential velocity remains unchanged. The fluid properties and time step are the same as in the previous example. Gravity is considered as the sole driving force ($g = 9.8\ \mathrm{m/s^2}$), and the total time is $0.5\ \mathrm{s}$, which is sufficient for the fluid to spread across the entire bottom wall. To validate accuracy, the normalized time-front position curve is compared with experimental and existing simulation results [75–77]. Fig. B.2(b) shows that the MLS-MPM solution for the normalized front with respect to the normalized time aligns well with previous numerical and experimental results. Fig. B.2(c) illustrates the

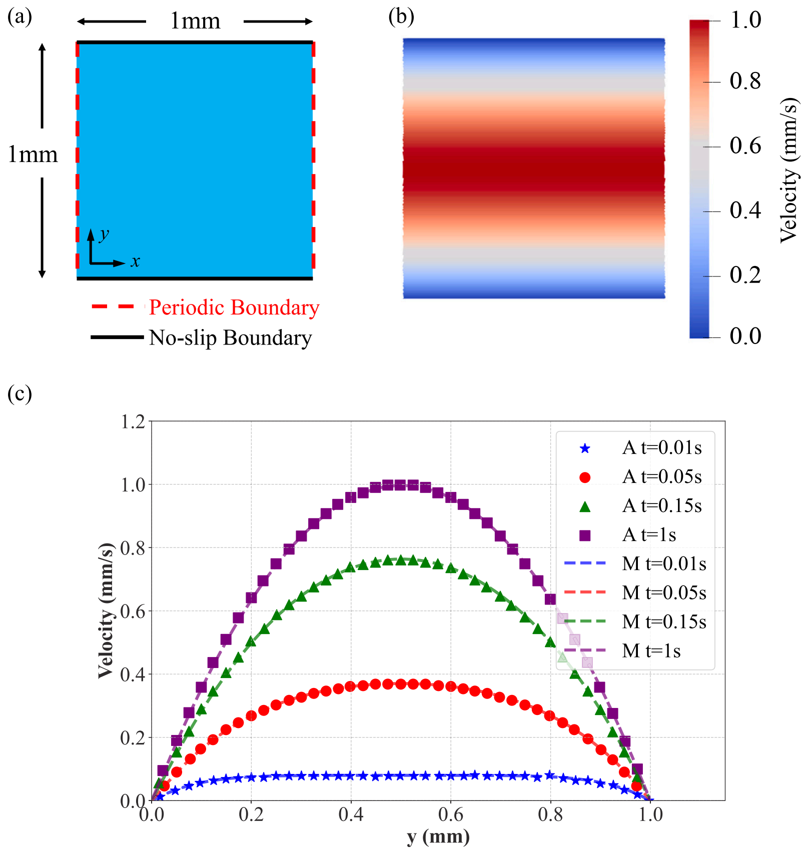

Fig. B.1. Poiseuille flow: (a) geometric model; (b) final velocity distribution; (c) comparison of simulation and analytical results at different times: 'A' represents the analytical solution and 'M' represents the MLS-MPM result.

velocity distribution and front position during the flow process. The maximum velocity occurs at the flow front and the front shape closely matches the experimental results.

B.3. Concrete slump test

To further test the ability of MLS-MPM for simulating non-Newtonian fluids, e.g., fresh concrete as concerned in this research, a numerical simulation for the slump test is conducted. The slump test is a widely used method in engineering to evaluate the fluidity of concrete [78]. The test process is shown in Fig. B.3(a). Engineers fill a cone mold with fresh concrete and lift it quickly. The distance that the concrete spreads before stopping indicates its flow capacity.

A 3D computational domain of size $1\ \mathrm{m} \times 1\ \mathrm{m} \times 0.4\ \mathrm{m}$ is created. In this model, the cone mold is ignored, as is common in several previous studies [79–81], and the initial concrete shape is directly represented as a cone with an upper radius of 0.05 m, a lower radius of 0.1 m, and a height of 0.3 m. The domain is discretized into regular cubic cells of size $0.01\ \mathrm{m}\times0.01\ \mathrm{m}\times0.01\ \mathrm{m}$, with eight particles per cell, totaling 44,096 particles. The bottom boundary is set as a no-slip condition, meaning the velocity at the bottom remains zero. The Bingham fluid constitutive model is employed here, and the material properties for three different concrete mixtures are summarized in Table B.1, as reported in [44]. The bulk modulus is 1×10^6 Pa and the time step size is 1×10^{-5} s.

Fig. B.3(b) illustrates the simulated slump process for material C. In the beginning, the concrete retains its cone shape. Once released, it collapses rapidly under gravity, with the maximum velocity occurring at the top. As the flow progresses, the velocity decreases due to the no-slip boundary condition and viscous resistance. Since Eq. (7) provides only an approximate representation of the Bingham fluid constitutive model (especially for rigid behavior below the yield stress), the velocity does not reduce to zero [82]. A relatively small velocity, $1\times10^{-4}\ \mathrm{m/s}$, is selected as the critical velocity threshold, below which the yield stress is assumed to balance gravity, marking the end of the flow process.

To validate the results, Fig. B.3(c) compares the simulated spread distance with an existing MPM simulation and experimental data from [44]. The red lines represent the experimental spread distance. The MLS-MPM results closely match the experimental measurements; even for Mix B, which shows the largest visual gap, the relative error is only 7%. Notably, while three different calibrated friction coefficients are used in the MPM simulation, the MLS-MPM simulation consistently applies a no-slip boundary condition across all cases. The consistent application of the no-slip boundary condition highlights the improved conservation in MLS-MPM, which is particularly important for accurately simulating high-viscosity fluids like Bingham fluids.

Data availability

Data will be made available on request.

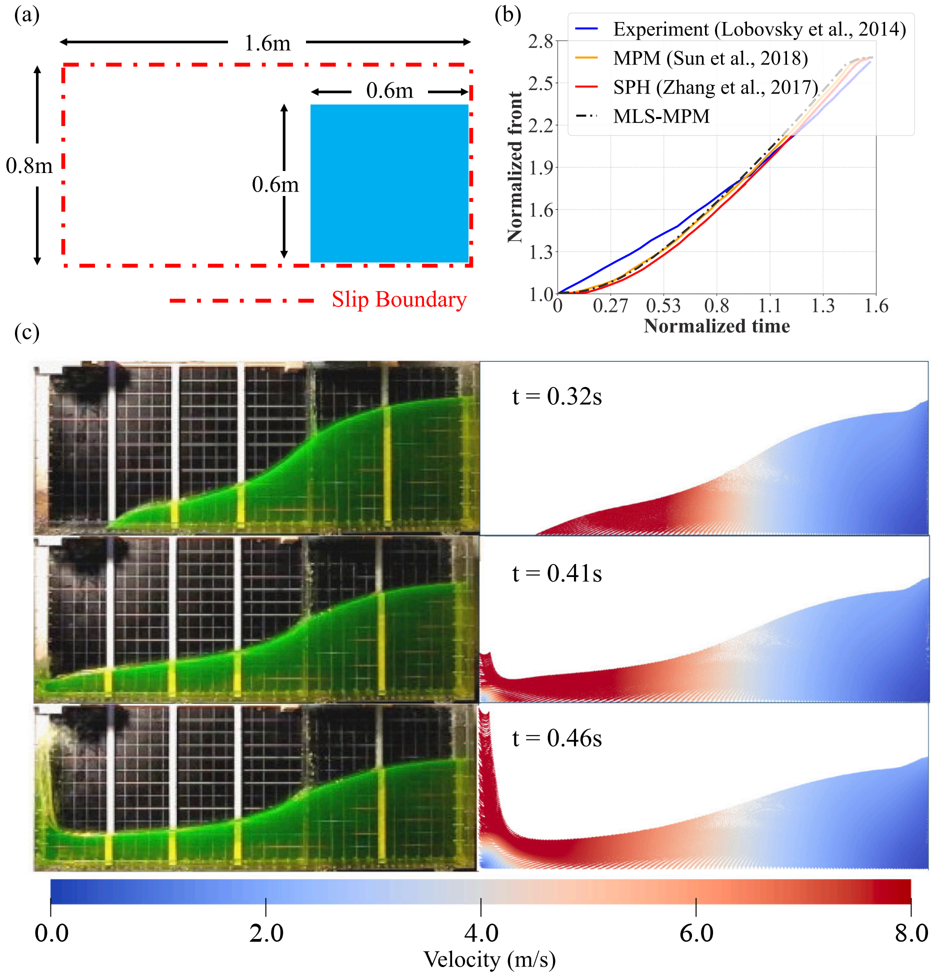

Fig. B.2. Dam break: (a) geometric model; (b) comparison of the MLS-MPM simulation results, the simulation and experimental data from [75–77]; (c) free surface shape at different times.

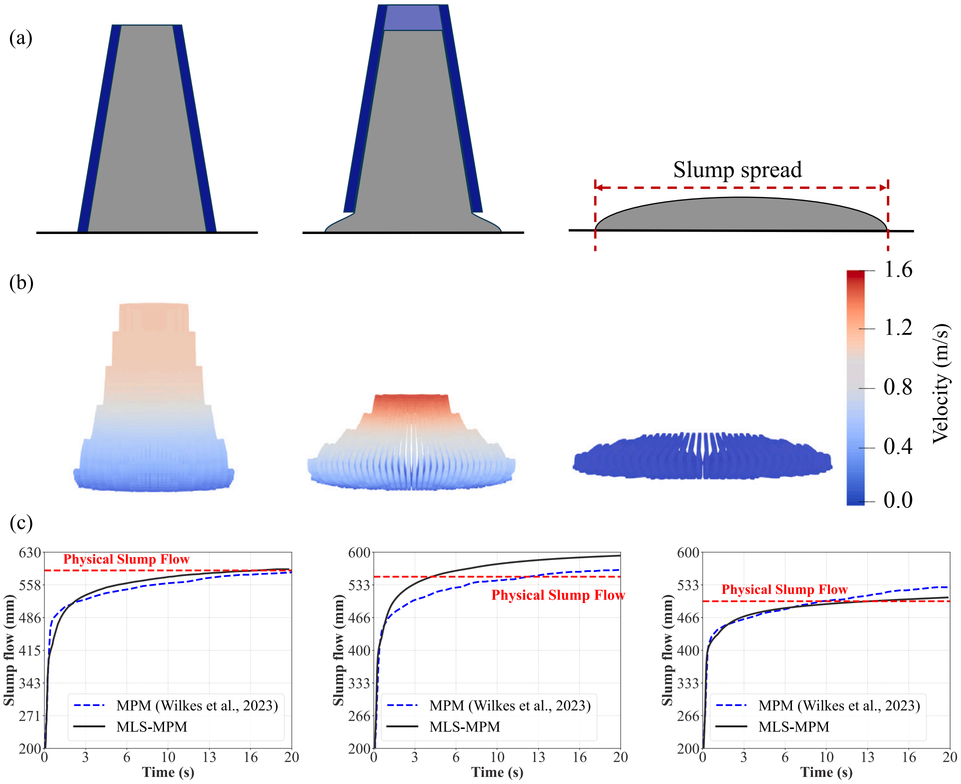

Fig. B.3. Slump test: (a) slump test diagram; (b) shape and velocity distribution for material C at t = 0.1 s, 0.2 s, 0.3 s; (c) comparison of the MLS-MPM simulation results, the simulation and experimental slump spread data from [44], for materials A, B, and C.

References

[1] H. Hassan, E. Rodriguez-Ubinas, A. Al Tamimi, E. Trepci, A. Mansouri, K. Almehairbi, Towards innovative and sustainable buildings: A comprehensive review of 3D printing in construction, Autom. Constr. 163 (2024) 105417, http://dx.doi.org/10.1016/j.autcon.2024.105417.

[2] G.H. Ahmed, N.H. Askandar, G.B. Jumaa, A review of largescale 3DCP: Material characteristics, mix design, printing process, and reinforcement strategies, Structures 43 (2022) 508–532, http://dx.doi.org/10.1016/j.istruc.2022.06.068.

[3] P. Ambily, S.K. Kaliyavaradhan, N. Rajendran, Top challenges to widespread 3D concrete printing (3DCP) adoption–a review, Eur. J. Env. Civ. Eng. 28 (2) (2024) 300–328, http://dx.doi.org/10.1080/19648189.2023.2213294.

[4] J. Reinold, V.N. Nerella, V. Mechtcherine, G. Meschke, Extrusion process simulation and layer shape prediction during 3D-concrete-printing using the particle finite element method, Autom. Constr. 136 (2022) 104173, http://dx.doi.org/10.1016/j.autcon.2022.104173.

[5] T. Pan, R. Guo, Y. Jiang, X. Ji, R. Qi, Flow and deformation behaviors of cementitious materials through nozzles with different geometric parameters: Experimental and numerical approaches, Constr. Build. Mater. 360 (2022) 129605, http://dx.doi.org/10.1016/j.conbuildmat.2022.129605.

[6] V. Nguyen-Van, B. Panda, G. Zhang, H. Nguyen-Xuan, P. Tran, Digital design computing and modelling for 3-D concrete printing, Autom. Constr. 123 (2021) 103529, http://dx.doi.org/10.1016/j.autcon.2020.103529.

[7] V. Mechtcherine, F.P. Bos, A. Perrot, W.L. Da Silva, V. Nerella, S. Fataei, R.J. Wolfs, M. Sonebi, N. Roussel, Extrusion-based additive manufacturing with cement-based materials–production steps, processes, and their underlying physics: a review, Cem. Concr. Res. 132 (2020) 106037, http://dx.doi.org/10.1016/j.cemconres.2020.106037.

[8] M. Maroszek, M. Rudziewicz, A. Hutyra, P. Dziura, M. Hebda, Evaluation of 3D concrete printing extrusion efficiency, Appl. Sci. 14 (24) (2024) 11866, http://dx.doi.org/10.3390/app142411866.

[9] F.Z. Oulkhir, I. Akhrif, M. El Jai, 3D concrete printing success: an exhaustive diagnosis and failure modes analysis, Prog. Addit. Manuf. 10 (1) (2025) 517–559, http://dx.doi.org/10.1007/s40964-024-00638-5.

[10] S. Kanagasuntharam, S. Ramakrishnan, S. Muthukrishnan, J. Sanjayan, Effect of magnetorheological additives on the buildability of 3D concrete printing, J. Build. Eng. 74 (2023) 106814, http://dx.doi.org/10.1016/j.jobe.2023.106814.

[11] A.S. Suiker, R.J. Wolfs, S.M. Lucas, T.A. Salet, Elastic buckling and plastic collapse during 3D concrete printing, Cem. Concr. Res. 135 (2020) 106016, http://dx.doi.org/10.1016/j.cemconres.2020.106016.

[12] V. Nguyen-Van, P. Tran, J. Liu, M. Van Tran, Y.M. Xie, Extended finite element multiscale modelling for crack propagation in 3D-printed fibre-reinforced concrete, Addit. Manuf. 81 (2024) 104019, http://dx.doi.org/10.1016/j.addma.2024.104019.

[13] L. He, W.T. Chow, H. Li, Effects of interlayer notch and shear stress on interlayer strength of 3D printed cement paste, Addit. Manuf. 36 (2020) 101390, http://dx.doi.org/10.1016/j.addma.2020.101390.

[14] V. Nguyen-Van, J. Liu, S. Li, G. Zhang, H. Nguyen-Xuan, P. Tran, Modelling of 3D-printed bio-inspired bouligand cementitious structures reinforced with steel fibres, Eng. Struct. 274 (2023) 115123, http://dx.doi.org/10.1016/j.engstruct.2022.115123.

[15] G. Ma, Z. Li, L. Wang, Printable properties of cementitious material containing copper tailings for extrusion based 3D printing, Constr. Build. Mater. 162 (2018) 613–627, http://dx.doi.org/10.1016/j.conbuildmat.2017.12.051.

[16] J. Wang, P. Zhang, J. Feng, J. Cai, Experimental study and design approach on 3D printed concrete walls under eccentric axial compression, Case Stud. Constr. Mater. 20 (2024) e02892, http://dx.doi.org/10.1016/j.cscm.2024.e02892.

[17] O. Ahi, Ö. Ertunç, Z.B. Bundur, Ö. Bebek, Automated flow rate control of extrusion for 3D concrete printing incorporating rheological parameters, Autom. Constr. 160 (2024) 105319, http://dx.doi.org/10.1016/j.autcon.2024.105319.

[18] B.M. Birru, A.U. Rehman, J.-H. Kim, Comparative analysis of structural build-up in one-component stiff and two-component shotcrete-accelerated set-on-demand mixtures for 3D concrete printing, Virt. Phys. Prototyp. 19 (1) (2024) e2359625, http://dx.doi.org/10.1080/17452759.2024.2359625.

[19] T. Ooms, G. Vantyghem, R. Van Coile, W. De Corte, A parametric modelling strategy for the numerical simulation of 3D concrete printing with complex geometries, Addit. Manuf. 38 (2021) 101743, http://dx.doi.org/10.1016/j.addma.2020.101743.

[20] J. Liu, V. Nguyen-Van, B. Panda, K. Fox, A. du Plessis, P. Tran, Additive manufacturing of sustainable construction materials and form-finding structures: a review on recent progresses, 3D Print. Addit. Manuf. 9 (1) (2022) 12–34, http://dx.doi.org/10.1089/3dp.2020.03.

[21] M. van den Heever, F. Bester, J. Kruger, G. van Zijl, Numerical modelling strategies for reinforced 3D concrete printed elements, Addit. Manuf. 50 (2022) 102569, http://dx.doi.org/10.1016/j.addma.2021.102569.

[22] J. Ge, G. Fu, J.H.S. Almeida Jr., Y. Jin, D. Sun, Thermal effect in CFRP machining: Temperature field characteristics, heat generation mechanism and thermal damage management, Compos. Struct. (2025) 118845, http://dx.doi.org/10.1016/j.compstruct.2025.118845.

[23] V. Nguyen-Van, S. Li, J. Liu, K. Nguyen, P. Tran, Modelling of 3D concrete printing process: A perspective on material and structural simulations, Addit. Manuf. 61 (2023) 103333, http://dx.doi.org/10.1016/j.addma.2022.103333.

[24] H. Liu, T. Ding, J. Xiao, V. Mechtcherine, Buildability prediction of 3D–printed concrete at early-ages: A numerical study with Drucker–Prager model, Addit. Manuf. 55 (2022) 102821, http://dx.doi.org/10.1016/j.addma.2022.102821.

[25] R. Comminal, W.R.L. da Silva, T.J. Andersen, H. Stang, J. Spangenberg, Modelling of 3D concrete printing based on computational fluid dynamics, Cem. Concr. Res. 138 (2020) 106256, http://dx.doi.org/10.1016/j.cemconres.2020.106256.

[26] Y. Wei, S. Han, S. Yu, Z. Chen, Z. Li, H. Wang, W. Cheng, M. An, Parameter impact on 3D concrete printing from single to multi-layer stacking, Autom. Constr. 164 (2024) 105449, http://dx.doi.org/10.1016/j.autcon.2024.105449.

[27] K. El Abbaoui, I. Al Korachi, M. El Jai, B. Šeta, M.T. Mollah, 3D concrete printing using computational fluid dynamics: Modeling of material extrusion with slip boundaries, J. Manuf. Process. 118 (2024) 448–459, http://dx.doi.org/10.1016/j.jmapro.2024.03.042.

[28] W. Cui, H. Sun, J. Zhou, S. Wang, X. Shi, Y. Tao, Geometric quality evaluation of three-dimensional printable concrete using computational fluid dynamics, Front. Struct. Civ. Eng. 18 (7) (2024) 963–976, http://dx.doi.org/10.1007/s11709-024-1080-4.

[29] Y. Tao, G. De Schutter, K. Van Tittelboom, Int. Conf. 3D Constr. Print., Springer, 2023, pp. 69–75, http://dx.doi.org/10.1007/978-3-031-64269-2_9.

[30] Y. Saade, M. Jalaal, A. Prosperetti, D. Lohse, Crown formation from a cavitating bubble close to a free surface, J. Fluid Mech. 926 (2021) A5, http://dx.doi.org/10.1017/jfm.2021.676.

[31] A. Yang, S. Chen, L. Yang, X. Yang, An upwind finite volume method for incompressible inviscid free surface flows, Comput. & Fluids 101 (2014) 170–182, http://dx.doi.org/10.1016/j.compfluid.2014.06.013.

[32] E. Ramyar, G. Cusatis, Discrete fresh concrete model for simulation of ordinary, self-consolidating, and printable concrete flow, J. Eng. Mech. 148 (2) (2022) 04021142, http://dx.doi.org/10.1061/(ASCE)EM.1943-7889.0002059.

[33] H. Yu, W. Zhang, B. Yin, W. Sun, A. Akbar, Y. Zhang, K. Liew, Modeling extrusion process and layer deformation in 3D concrete printing via smoothed particle hydrodynamics, Comput. Methods Appl. Mech. Engrg. 420 (2024) 116761, http://dx.doi.org/10.1016/j.cma.2024.116761.

[34] J. Reinold, G. Meschke, A mixed u–p edge-based smoothed particle finite element formulation for viscous flow simulations, Comput. Mech. 69 (4) (2022) 891–910, http://dx.doi.org/10.1007/s00466-021-02119-w.

[35] G. Rizzieri, L. Ferrara, M. Cremonesi, Numerical simulation of the extrusion and layer deposition processes in 3D concrete printing with the particle finite element method, Comput. Mech. 73 (2) (2024) 277–295, http://dx.doi.org/10.1007/s00466-023-02367-y.

[36] X. Zhang, Z. Chen, Y. Liu, The Material Point Method: a Continuum-Based Particle Method for Extreme Loading Cases, Academic Press, 2016, http://dx.doi.org/10.1016/B978-0-12-407716-4.00010-7.

[37] Z. Wikeckowski, The material point method in large strain engineering problems, Comput. Methods Appl. Mech. Engrg. 193 (39–41) (2004) 4417–4438, http://dx.doi.org/10.1016/j.cma.2004.01.035.

[38] L. Beuth, Z. Wikeckowski, P. Vermeer, Solution of quasi-static large-strain problems by the material point method, Int. J. Numer. Anal. Methods Geomech. 35 (13) (2011) 1451–1465, http://dx.doi.org/10.1002/nag.965.

[39] C. Zhou, Q. Zhong, X. Zhou, X. Wu, S. Chen, Research advances in large deformation analysis and applications of the material point method, Appl. Sci. 15 (12) (2025) 6617, http://dx.doi.org/10.3390/app15126617.

[40] K. Feng, Material Point Method for Large Deformation Analyses of Coseismic Landslides, Hong Kong University of Science and Technology (Hong Kong), 2021, http://dx.doi.org/10.14711/thesis-991012994506803412.

[41] F. Ceccato, A. Yerro, G. Di Carluccio, Simulating landslides with the material point method: Best practices, potentialities, and challenges, Eng. Geol. (2024) 107614, http://dx.doi.org/10.1016/j.enggeo.2024.107614.

[42] F. Wu, W. Sun, X. Li, Y. Guan, M. Dong, Material point method-based simulation of dynamic process of soil landslides considering pore fluid pressure, Int. J. Numer. Anal. Methods Geomech. 47 (13) (2023) 2385–2404, http://dx.doi.org/10.1002/nag.3581.

[43] Y. Zheng, J. Li, T. Zhu, J. Li, Experimental and MPM modelling of widened levee failure under the combined effect of heavy rainfall and high riverine water levels, Comput. Geotech. 184 (2025) 107259, http://dx.doi.org/10.1016/j.compgeo.2025.107259.

[44] C. Wilkes, K. Kumar, G. Biscontin, Investigating the thixotropic behaviour of tremie concrete using the slump-flow test and the material point method, Cem. Concr. Compos. 143 (2023) 105222, http://dx.doi.org/10.1016/j.cemconcomp.2023.105222.

[45] S.G. Bardenhagen, E.M. Kober, et al., The generalized interpolation material point method, Comput. Model. Eng. Sci. 5 (6) (2004) 477–496, http://dx.doi.org/10.3970/cmes.2004.005.477.

[46] M. Steffen, R.M. Kirby, M. Berzins, Analysis and reduction of quadrature errors in the material point method (MPM), Internat. J. Numer. Methods Engrg. 76 (6) (2008) 922–948, http://dx.doi.org/10.1002/nme.2360.

[47] A. Sadeghirad, R.M. Brannon, J. Burghardt, A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations, Internat. J. Numer. Methods Engrg. 86 (12) (2011) 1435–1456, http://dx.doi.org/10.1002/nme.3110.

[48] D. Sulsky, M. Gong, Improving the Material-Point Method, Springer International Publishing, 2016, pp. 217–240, http://dx.doi.org/10.1007/978-3-319-39022-2_10.

[49] Y. Gan, Z. Sun, Z. Chen, X. Zhang, Y. Liu, Enhancement of the material point method using B-spline basis functions, Internat. J. Numer. Methods Engrg. 113 (3) (2018) 411–431, http://dx.doi.org/10.1002/nme.5620.

[50] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, C. Jiang, A moving least squares material point method with displacement discontinuity and two-way rigid body coupling, ACM Trans. Graph. 37 (4) (2018) 1–14, http://dx.doi.org/10.1145/3197517.3201293.

[51] V. Nguyen-Van, H. Nguyen-Xuan, B. Panda, P. Tran, 3D concrete printing modelling of thin-walled structures, Struct. 39 (2022) 496–511, http://dx.doi.org/10.1016/j.istruc.2022.03.049.

[52] J.R. Martins, A. Ning, Engineering Design Optimization, Cambridge University Press, 2021, http://dx.doi.org/10.1017/9781108980647.

[53] M.G. Alhashim, K. Hausknecht, M.P. Brenner, Control of flow behavior in complex fluids using automatic differentiation, Proc. Natl. Acad. Sci. 122 (8) (2025) e2403644122, http://dx.doi.org/10.1073/pnas.24036441.

[54] Q. Wang, K. Kumar, An inverse analysis of fluid flow through granular media using differentiable lattice Boltzmann method, J. Rock Mech. Geotech. Eng. 16 (6) (2024) 2077–2090, http://dx.doi.org/10.1016/j.jrmge.2024.03.010.

[55] Z. Xu, Z. Li, F. Jiang, Numerical approach to pipe flow of fresh concrete based on MPS method, Cem. Concr. Res. 152 (2022) 106679, http://dx.doi.org/10.1016/j.cemconres.2021.106679.

[56] W. Wu, W. Chen, G. Lu, J. Wang, G. Tian, B. Xu, C. Deng, Research on rheological behavior of fresh concrete single-cylinder pumping based on SPH-DEM, Sci. Rep. 13 (1) (2023) 20828, http://dx.doi.org/10.1038/s41598-023-45702-2.

[57] I.K. Kang, T.Y. Shin, J.H. Kim, Unbiased rheological properties determined by adversarial training with Bingham equation, Cem. Concr. Compos. (2025) 105943, http://dx.doi.org/10.1016/j.cemconcomp.2025.105943.

[58] Y. Cai, Q.-f. Liu, M. Chen, Q.X. Xiong, B. Šavija, A CFD-based numerical model for predicting the slump and slump flow of fresh concrete from a rheological perspective, Constr. Build. Mater. 458 (2025) 138501, http://dx.doi.org/10.1016/j.conbuildmat.2024.138501.

[59] Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, F. Durand, Taichi: a language for high-performance computation on spatially sparse data structures, ACM Trans. Graph. 38 (6) (2019) 1–16, http://dx.doi.org/10.1145/3355089.3356506.

[60] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, A. Selle, A material point method for snow simulation, ACM Trans. Graph. 32 (4) (2013) 1–10, http://dx.doi.org/10.1145/2461912.2461948.

[61] C. Jiang, C. Schroeder, A. Selle, J. Teran, A. Stomakhin, The affine particle-in-cell method, ACM Trans. Graph. 34 (4) (2015) 1–10, http://dx.doi.org/10.1145/2766996.

[62] H. Negishi, M. Kondo, H. Amakawa, S. Obara, R. Kurose, Bingham fluid simulations using a physically consistent particle method, J. Fluid Sci. Technol. 18 (4) (2023) http://dx.doi.org/10.1299/jfst.2023jfst0035, JFST0035–JFST0035.

[63] W. Xu, K. Yang, Y. Zhang, S. Sun, S. Mao, T. Xue, DiffLiB: High-fidelity differentiable modeling of lithium-ion batteries and efficient gradient-based parameter identification, 2025, http://dx.doi.org/10.48550/arXiv.2504.20674, arXiv preprint arXiv:2504.20674.

[64] B. Dauvergne, L. Hascoët, The data-flow equations of checkpointing in reverse automatic differentiation, in: Int. Conf. Comput. Sci., Springer, 2006, pp. 566–573, http://dx.doi.org/10.1007/11758549_78.

[65] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, F. Durand, DiffTaichi: Differentiable programming for physical simulation, 2019, http://dx.doi.org/10.48550/arXiv.1910.00935, arXiv preprint arXiv:1910.00935. URL https://arxiv.org/abs/1910.00935. Preprint.

[66] J. Spangenberg, W.R.L. Da Silva, R. Comminal, M.T. Mollah, T.J. Andersen, H. Stang, Numerical simulation of multi-layer 3D concrete printing, RILEM Tech. Lett. 6 (2021) 119–123, http://dx.doi.org/10.21809/rilemtechlett.2021.142.

[67] L.D.G. Sigalotti, J. Klapp, E. Sira, Y. Meleán, A. Hasmy, SPH simulations of time-dependent poiseuille flow at low Reynolds numbers, J. Comput. Phys. 191 (2) (2003) 622–638, http://dx.doi.org/10.1016/S0021-9991(03)00343-7.

[68] T. Xue, S. Liao, Z. Gan, C. Park, X. Xie, W.K. Liu, J. Cao, JAX-FEM: A differentiable GPU-accelerated 3D finite element solver for automatic inverse design and mechanistic data science, Comput. Phys. Comm. 291 (2023) 108802, http://dx.doi.org/10.1016/j.cpc.2023.108802.

[69] A. Niewiarowski, S. Adriaenssens, R.M. Pauletti, Adjoint optimization of pressurized membrane structures using automatic differentiation tools, Comput. Methods Appl. Mech. Engrg. 372 (2020) 113393, http://dx.doi.org/10.1016/j.cma.2020.113393.

[70] T. Xue, S. Mao, Mapped shape optimization method for the rational design of cellular mechanical metamaterials under large deformation, Internat. J. Numer. Methods Engrg. 123 (10) (2022) 2357–2380, http://dx.doi.org/10.1002/nme.6941.

[71] C. Hu, M. Sueyoshi, Numerical simulation and experiment on dam break problem, J. Mar. Sci. Appl. 9 (2) (2010) 109–114, http://dx.doi.org/10.1007/s11804-010-9075-z.

[72] L. Fu, Y.-C. Jin, Simulating velocity distribution of dam breaks with the particle method, J. Hydraul. Eng. 140 (10) (2014) 04014048, http://dx.doi.org/10.1061/(ASCE)HY.1943-7900.0000915.

[73] X. Zhao, D. Liang, M. Martinelli, Numerical simulations of dam-break floods with MPM, Procedia Eng. 175 (2017) 133–140, http://dx.doi.org/10.1016/j.proeng.2017.01.041.

[74] Y. Shi, N. Guo, Z. Yang, GeoTaichi: A taichi-powered high-performance numerical simulator for multiscale geophysical problems, Comput. Phys. Comm. 301 (2024) 109219, http://dx.doi.org/10.1016/j.cpc.2024.109219.

[75] L. Lobovskỳ, E. Botia-Vera, F. Castellana, J. Mas-Soler, A. Souto-Iglesias, Experimental investigation of dynamic pressure loads during dam break, J. Fluids Struct. 48 (2014) 407–434, http://dx.doi.org/10.1016/j.jfluidstructs.2014.03.009.

[76] C. Zhang, X. Hu, N.A. Adams, A weakly compressible SPH method based on a low-dissipation Riemann solver, J. Comput. Phys. 335 (2017) 605–620, http://dx.doi.org/10.1016/j.jcp.2017.01.027.

[77] Z. Sun, H. Li, Y. Gan, H. Liu, Z. Huang, L. He, Material point method and smoothed particle hydrodynamics simulations of fluid flow problems: a comparative study, Prog. Comput. Fluid Dyn. 18 (1) (2018) 1–18, http://dx.doi.org/10.1504/PCFD.2018.089497.

[78] D.E. Dixon, J.R. Prestrera, G.R. Burg, S.A. Chairman, E.A. Abdun-Nur, S.G. Barton, L.W. Bell, S.J. Blas Jr., R.L. Carraquillo, P.M. Carraquillo, et al., Standard practice for selecting proportions for normal heavyweight, and mass concrete (ACI 211.1-91) reapproved 1997, Rep. By ACI Comm 211 (1991) 1–38.

[79] Y. Li, J. Mu, Z. Wang, Y. Liu, H. Du, Numerical simulation on slump test of fresh concrete based on lattice Boltzmann method, Cem. Concr. Compos. 122 (2021) 104136, http://dx.doi.org/10.1016/j.cemconcomp.2021.104136.

[80] N. Roussel, Correlation between yield stress and slump: comparison between numerical simulations and concrete rheometers results, Mater. Struct. 39 (4) (2006) 501–509, http://dx.doi.org/10.1617/s11527-005-9035-2.

[81] N. Roussel, C. Stéfani, R. Leroy, From mini-cone test to abrams cone test: measurement of cement-based materials yield stress using slump tests, Cem. Concr. Res. 35 (5) (2005) 817–822, http://dx.doi.org/10.1016/j.cemconres.2004.07.032.

[82] N. Roussel, A. Gram, M. Cremonesi, L. Ferrara, K. Krenzer, V. Mechtcherine, S. Shyshko, J. Skocec, J. Spangenberg, O. Svec, et al., Numerical simulations of concrete flow: A benchmark comparison, Cem. Concr. Res. 79 (2016) 265–271, http://dx.doi.org/10.1016/j.cemconres.2015.09.022.