

# Taylor particle-in-cell transfer and kernel correction for material point method

Keita Nakamura<sup>\*</sup>, Satoshi Matsumura, Takaaki Mizutani

*Port and Airport Research Institute, 3-1-1 Nagase, Yokosuka, Kanagawa 239-0826, Japan*

Received 7 August 2022; received in revised form 16 October 2022; accepted 16 October 2022

Available online xxxx

## Abstract

The material point method (MPM) has been extensively used to solve problems involving large displacements and deformations. The standard MPM formulation, which adopts fluid-implicit-particle (FLIP) transfer, is less dissipative but also less stable than particle-in-cell (PIC) transfer. The affine PIC (APIC) transfer introduces the affine velocity and was developed to realize stable simulations while overcoming dissipations of the angular momentum in PIC. This paper presents Taylor-PIC (TPIC) transfer, which is a type of APIC transfer that combines the affine velocity based on the first-order Taylor series approximation and PIC transfer. TPIC is simple (only the particle velocity gradient is required) and inherits the key advantages of the original APIC, such as less dissipation and stability. Although TPIC does not conserve angular momentum, in contrast to APIC, the velocity gradient is preserved in the transfer between particles and the grid. This velocity gradient contains the angular information in its skew-symmetric component, which allows TPIC to adequately describe angular motion, similar to APIC.

Furthermore, the MPM can cause stress oscillations near boundaries when boundary conditions are explicitly imposed, e.g., when the boundary grid velocity (or momentum) is set to zero. When affine-type transfers are used, this instability is exacerbated, and simulations can easily fail. Therefore, we propose a kernel correction method based on the weighted least squares for particles near boundaries. The proposed TPIC transfer and kernel correction are validated through five types of simulations, each using a different material—linear elastic, von Mises, Newtonian fluid, and Drucker–Prager. In the simulations, incorrect results due to stress oscillations are observed even for small deformations. Applying the corrected kernels successfully removes these spurious oscillations, and the results are consistent with the analytical solutions. Moreover, the numerical results confirm the accuracy and robustness of the proposed TPIC transfer.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

**Keywords:** Material point method; Particle-in-cell; Fluid-implicit-particle; Affine particle-in-cell; Kernel correction; Boundary condition

## 1. Introduction

Meshfree methods have been extensively used for numerical computations and analyses in various engineering fields. A major advantage of these methods is that they can simulate extremely large deformations while avoiding mesh distortions occurring in the finite element method (FEM). The material point method (MPM) [1,2] is an

<sup>\*</sup> Corresponding author.

E-mail addresses: [nakamura-ke@p.mpat.go.jp](mailto:nakamura-ke@p.mpat.go.jp) (K. Nakamura), [matsumura-s@p.mpat.go.jp](mailto:matsumura-s@p.mpat.go.jp) (S. Matsumura), [mizutani-t@p.mpat.go.jp](mailto:mizutani-t@p.mpat.go.jp) (T. Mizutani).

extensively used particle-based meshfree method, which was developed as an extension of the particle-in-cell (PIC) method [3] and the fluid-implicit-particle (FLIP) method [4,5]. In the MPM, a continuum body is discretized into particles that move through the background mesh. Information pertaining to the physical quantities of the particles, such as the velocity and stress, is transferred to the background mesh. The mesh, which is used to solve the momentum equation, is then deformed according to the updated velocities at the nodes. Subsequently, the particle information is updated according to the deformation of the mesh. At the end of the timestep, the deformed mesh is reset to its original state.

Because the accuracy of MPM simulations depends strongly on the kernel functions, many studies have focused on interpolation schemes. For example, the original MPM [1] suffers from a cell-crossing problem owing to the lack of smoothness in the kernel function. In [6], this problem was solved with the introduction of the generalized interpolation material point (GIMP) method. As an extension of the GIMP method, the convected particle domain interpolation technique was developed [7]. Interpolations based on B-splines have also been extensively studied [8–13]. In addition, the total Lagrangian MPM [14] was proposed to address the cell-crossing instability, low order of convergence, and numerical fracture associated with the conventional MPM.

Another important problem regarding the accuracy of the MPM is the information transfer between the particles and the grid. Two major approaches exist for grid-to-particle transfer: (1) PIC transfer, in which the particle velocity is directly interpolated from the grid velocity, and (2) FLIP transfer, in which the particle velocity is updated using grid acceleration. The original MPM uses FLIP transfer because PIC transfer is highly dissipative [1,5]. However, the FLIP approach is relatively unstable. Therefore, a blended velocity obtained using the PIC and FLIP transfers [15] is also extensively used. In addition to grid-to-particle transfer, studies have focused on improving the particle-to-grid transfer. In [16], a velocity projection method that uses the velocity gradient was proposed. Jiang et al. [17] developed a similar method using affine velocity fields combined with PIC transfer. One of their most significant observations was that the affine velocity could resolve the dissipative nature of PIC transfer without the loss of computational stability. This affine PIC (APIC) transfer successfully conserves angular momentum and has been applied in various simulations [17,18]. Hu et al. [19] proposed the moving least squares MPM (MLS-MPM), which is a generalization of the affine transfer algorithm. Although the APIC transfer satisfies the conservation of the angular momentum, the kinematic energy is still not conserved, which was improved considerably by a polynomial PIC method [20].

As in the case of APIC transfer, the combination of particle-to-grid and grid-to-particle transfers affects considerably the accuracy and stability of simulations. Although the original APIC transfer has been extensively studied [17,21], the velocity projection presented by P. C. Wallstedt [16], which also forms an affine velocity field, has not been adequately investigated for PIC transfer. Thus, one of the primary objectives of the present study is to evaluate the performance of the combination of the velocity projection and PIC approach. As this velocity projection is fundamentally a first-order Taylor series approximation of the particle velocity in space, this combination is named Taylor-PIC (TPIC) transfer to distinguish it from the original APIC transfer. We analyzed the differences among the FLIP, PIC, APIC, and TPIC transfers by evaluating the dissipation (i.e., loss of physical information) according to their mathematical formulations.

In APIC transfer, the B-spline functions are used as the kernel functions, and the peaks of the B-spline functions are centered on the grid nodes, as presented by Steffen et al. [8], in contrast to other recursive B-spline MPMs [9–14]. However, these B-spline functions cause unstable behavior because of stress oscillations near the boundaries. When affine-type transfers such as APIC or TPIC transfers are used, this instability worsens, resulting in the failure of simulations. Schulz and Sutmann [22] proposed a method that used image particles to reduce these artifacts, which has been applied to geotechnical problems [23]. Herein, we present a kernel correction method based on the weighted least squares for reducing the stress oscillation.

The remainder of this paper is organized as follows. Section 2 presents the MPM formulation, including FLIP, APIC, and TPIC transfers. Section 3 describes the mathematical differences among these types of transfers. Section 4 presents the kernel correction method. In Section 5, the proposed formulations are discussed based on five types of numerical simulations. Finally, Section 6 presents the conclusions.

The notation and symbols used in the paper are as follows: boldface letters denote vectors and matrices; the symbol ‘ $\cdot$ ’ denotes a single contraction of adjacent indices (e.g.,  $\mathbf{a} \cdot \mathbf{b} = a_i b_i$ ,  $\mathbf{c} \cdot \mathbf{a} = c_{ij} a_j$ , and  $\mathbf{c} \cdot \mathbf{d} = c_{ij} d_{jk}$ ); the symbol ‘ $:$ ’ denotes a double contraction of adjacent indices (e.g.,  $\mathbf{A} : \mathbf{B} = A_{ij} B_{ij}$  and  $\mathbf{C} : \mathbf{A} = C_{ijkl} A_{kl}$ ); the symbol ‘ $\otimes$ ’ denotes a tensor product (e.g.,  $\mathbf{a} \otimes \mathbf{b} = a_i b_j$ ); and the symbol ‘ $\times$ ’ denotes a cross product.

## 2. Material point method (MPM) formulation

### 2.1. Governing equations

In the problem domain  $\Omega$ , the conservation of momentum in Eulerian form is expressed as

$$\rho \dot{\mathbf{v}} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b}, \quad (1)$$

where  $\rho$  represents the density,  $\mathbf{b}$  represents the body force per unit mass,  $\boldsymbol{\sigma}$  represents the Cauchy stress, and the superposed dot denotes the material derivative. The boundary conditions are as follows:

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \partial\Omega_t, \quad (2)$$

$$\mathbf{v} = \bar{\mathbf{v}} \quad \text{on } \partial\Omega_v, \quad (3)$$

where the superposed bar denotes the prescribed boundary values, and  $\mathbf{n}$  is the unit normal to the body surface  $\partial\Omega$ .

Thus, the weak form of the conservation of momentum equation becomes

$$\int_{\Omega} \boldsymbol{\eta} \cdot \rho \dot{\mathbf{v}} \, dV = - \int_{\Omega} \nabla \boldsymbol{\eta} : \boldsymbol{\sigma} \, dV + \int_{\Omega} \boldsymbol{\eta} \cdot \rho \mathbf{b} \, dV + \int_{\partial\Omega_t} \boldsymbol{\eta} \cdot \bar{\mathbf{t}} \, dA, \quad (4)$$

where  $\boldsymbol{\eta}$  denotes the vector-valued test function. The conservation of mass is automatically satisfied in the MPM [1].

### 2.2. Space discretization

From Eq. (4), the continuum domain is divided into local domains  $\Omega_p$  as follows:

$$\sum_p \int_{\Omega_p} \boldsymbol{\eta} \cdot \rho \dot{\mathbf{v}} \, dV = - \sum_p \int_{\Omega_p} \nabla \boldsymbol{\eta} : \boldsymbol{\sigma} \, dV + \sum_p \int_{\Omega_p} \boldsymbol{\eta} \cdot \rho \mathbf{b} \, dV + \sum_p \int_{\partial\Omega_{tp}} \boldsymbol{\eta} \cdot \bar{\mathbf{t}} \, dA. \quad (5)$$

The velocity  $\mathbf{v}$  and vector-valued test function  $\boldsymbol{\eta}$  can be interpolated from the nodal velocity  $\mathbf{v}_i$  and nodal test function  $\boldsymbol{\eta}_i$  using the kernel function  $\phi_i(\mathbf{x})$  as follows:

$$\mathbf{v} = \sum_i \phi_i(\mathbf{x}) \mathbf{v}_i, \quad (6)$$

$$\boldsymbol{\eta} = \sum_i \phi_i(\mathbf{x}) \boldsymbol{\eta}_i. \quad (7)$$

Thus, in the local domain,  $\Omega_p$ , Eq. (5) becomes

$$\sum_i \boldsymbol{\eta}_i \cdot \sum_j \left[ \int_{\Omega_p} \phi_i \phi_j \rho \, dV \right] \dot{\mathbf{v}}_j = \sum_i \boldsymbol{\eta}_i \cdot \left[ - \int_{\Omega_p} \nabla \phi_i : \boldsymbol{\sigma} \, dV + \int_{\Omega_p} \phi_i \rho \mathbf{b} \, dV + \int_{\partial\Omega_{tp}} \phi_i \bar{\mathbf{t}} \, dA \right]. \quad (8)$$

Considering the arbitrariness of  $\boldsymbol{\eta}_i$ ,

$$\sum_j m_{ij} \dot{\mathbf{v}}_j = \mathbf{f}_i^{\text{int}} + \mathbf{f}_i^{\text{ext}}, \quad (9)$$

where

$$m_{ij} = \sum_p \int_{\Omega_p} \phi_i \phi_j \rho \, dV = \sum_p m_p \phi_{ip} \phi_{jp}, \quad (10)$$

$$\mathbf{f}_i^{\text{int}} = - \sum_p \int_{\Omega_p} \nabla \phi_i : \boldsymbol{\sigma} \, dV = - \sum_p V_p \nabla \phi_{ip} \cdot \boldsymbol{\sigma}_p, \quad (11)$$

$$\mathbf{f}_i^{\text{ext}} = \sum_p \left[ \int_{\Omega_p} \phi_i \rho \mathbf{b} \, dV + \int_{\partial\Omega_{tp}} \phi_i \bar{\mathbf{t}} \, dA \right] = \sum_p \phi_{ip} m_p \mathbf{b}_p + \bar{\mathbf{f}}_i. \quad (12)$$

In Eqs. (11)–(12), we adopt  $\phi_{ip} = \phi_i(\mathbf{x}_p)$ ;  $m_{ij}$  is referred to as the consistent mass matrix. In the MPM, a lumped mass matrix is used instead of a consistent mass matrix to simplify the computations [1]. The lumped mass matrix

is a diagonal matrix in which each diagonal entry is the corresponding row sum of the consistent matrix, which assumes

$$m_i = \sum_p \int_{\Omega_p} \phi_i \rho \, dV = \sum_p m_p \phi_{ip}. \quad (13)$$

### 2.3. Time discretization

Applying forward Euler time discretization to Eq. (9) and using the lumped mass matrix in Eq. (13) yields

$$m_i^n \frac{\mathbf{v}_i^{n+1} - \mathbf{v}_i^n}{\Delta t} = \mathbf{f}_i^n, \quad (14)$$

where  $\mathbf{f}_i^n = \mathbf{f}_i^{\text{int},n} + \mathbf{f}_i^{\text{ext},n}$ . The superscript  $n$  denotes the timestep, and  $\Delta t = t^{n+1} - t^n$ .

### 2.4. Particle-to-grid transfer

In the MPM, the particle velocity should be transferred to the background grid. This is accomplished using Shepard's partition of unity method [24], assuming a mass-weighted function, as in the original MPM [1]:

$$\mathbf{v}_i^n = \frac{1}{m_i^n} \sum_p \phi_{ip}^n m_p \mathbf{v}_p^n. \quad (15)$$

In APIC transfer [17], the affine velocity field  $\tilde{\mathbf{v}}_p^n$  is introduced as

$$\mathbf{v}_i^n = \frac{1}{m_i^n} \sum_p \phi_{ip}^n m_p \tilde{\mathbf{v}}_p^n(\mathbf{x}_i^n). \quad (16)$$

The velocity  $\tilde{\mathbf{v}}_p^n(\mathbf{x}_i^n)$  is expressed as

$$\tilde{\mathbf{v}}_p^n(\mathbf{x}_i^n) = \mathbf{v}_p^n + \mathbf{B}_p^n \cdot (\mathbf{D}_p^n)^{-1} \cdot (\mathbf{x}_i^n - \mathbf{x}_p^n), \quad (17)$$

where

$$\mathbf{D}_p^n = \sum_i \phi_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n) \otimes (\mathbf{x}_i^n - \mathbf{x}_p^n). \quad (18)$$

The matrix  $\mathbf{B}_p^n$ , which indicates the additional particle quantity evaluated in the grid-to-particle transfer (Section 2.6), contains the angular momentum information.

TPIC transfer also uses the affine velocity field  $\tilde{\mathbf{v}}_p^n$  in Eq. (16); however, the velocity is defined using the first-order Taylor series approximation of the particle velocity  $\mathbf{v}_p^n$  at  $\mathbf{x}_i^n$ , as follows:

$$\tilde{\mathbf{v}}_p^n(\mathbf{x}_i^n) = \mathbf{v}_p^n + \nabla \mathbf{v}_p^n \cdot (\mathbf{x}_i^n - \mathbf{x}_p^n), \quad (19)$$

where  $\nabla \mathbf{v}_p^n$  represents the velocity gradient of the particle. This velocity projection was first proposed by P. C. Wallstedt [16]. However, the projection has not been investigated for PIC transfer. One of our contributions is to reveal that the combination of this projection and PIC transfer (i.e., TPIC) can preserve the velocity gradient, as described in Section 3. Its performance is evaluated based on numerical simulations in Section 5.

### 2.5. Grid velocity update

Because the momentum equation is solved on the grid in the MPM, the grid velocity can be updated using Eq. (14) as

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{a}_i^n, \quad (20)$$

where  $\mathbf{a}_i^n$  represents the grid acceleration, which is defined as

$$\mathbf{a}_i^n = \mathbf{f}_i^n / m_i^n. \quad (21)$$

## 2.6. Grid-to-particle transfer

In the original MPM, the particle velocity is updated using FLIP [4,5] transfer, i.e.,

$$\mathbf{v}_p^{\text{FLIP},n+1} = \mathbf{v}_p^n + \Delta t \sum_i \phi_{ip}^n \mathbf{a}_i^n, \quad (22)$$

rather than PIC [3] transfer, i.e.,

$$\mathbf{v}_p^{\text{PIC},n+1} = \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1}. \quad (23)$$

FLIP transfer can reduce the severe damping of the motion in PIC transfer; however, it can result in unstable behavior. In [15], a linearly blended velocity of the PIC and FLIP methods was proposed, as follows:

$$\mathbf{v}_p^{\text{Blend},n+1} = \xi \mathbf{v}_p^{\text{FLIP},n+1} + (1 - \xi) \mathbf{v}_p^{\text{PIC},n+1}, \quad (24)$$

where  $\xi = 0.95$  was used in [25].

Jiang et al. [17] proposed a novel APIC transfer that combines the affine velocity field and PIC transfer (Eq. (23)). It resolves the dissipative nature of PIC transfer by introducing the affine velocity field, as described in Section 2.4. In APIC transfer, the additional particle quantity  $\mathbf{B}_p^n$  should be updated as

$$\mathbf{B}_p^{n+1} = \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1} \otimes (\mathbf{x}_i^n - \mathbf{x}_p^n). \quad (25)$$

Introducing  $\mathbf{B}_p$  satisfies the conservation of angular momentum [17].

TPIC transfer also uses PIC transfer to update the particle velocity, as expressed in Eq. (23). In any transfer method, the velocity gradient in Eq. (19) should be evaluated to obtain the deformation gradient, as discussed in Section 2.7. Thus, TPIC requires no additional quantities such as  $\mathbf{B}_p$  in APIC.

## 2.7. Other particle states

Other particle states are updated similarly, regardless of the transfer approach. The velocity gradient can be obtained as

$$\nabla \mathbf{v}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} \otimes \nabla \phi_{ip}^n. \quad (26)$$

By using the updated velocity gradient, the deformation gradient can be updated as

$$\mathbf{F}_p^{n+1} = (\mathbf{I} + \Delta t \nabla \mathbf{v}_p^{n+1}) \cdot \mathbf{F}_p^n. \quad (27)$$

When the Jaumann rate is selected for the objective stress tensor, the updated stress becomes

$$\boldsymbol{\sigma}_p^{n+1} = \boldsymbol{\sigma}_p^n + \mathbf{T} : \Delta \mathbf{d}_p + (\Delta \boldsymbol{\omega}_p \cdot \boldsymbol{\sigma}_p^n - \boldsymbol{\sigma}_p^n \cdot \Delta \boldsymbol{\omega}_p), \quad (28)$$

where  $\mathbf{T}$  is the tangent modulus tensor of the material model,  $\mathbf{d}$  represents the rate of deformation tensor, and  $\boldsymbol{\omega}$  is the spin tensor.

$$\Delta \mathbf{d}_p = \frac{1}{2} \left[ \nabla \mathbf{v}_p^{n+1} + (\nabla \mathbf{v}_p^{n+1})^\top \right] \Delta t \quad (29)$$

$$\Delta \boldsymbol{\omega}_p = \frac{1}{2} \left[ \nabla \mathbf{v}_p^{n+1} - (\nabla \mathbf{v}_p^{n+1})^\top \right] \Delta t \quad (30)$$

Finally, the position of the particle can be updated as

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1}. \quad (31)$$

In the PIC transfer, Eq. (31) can be simplified as

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}. \quad (32)$$

The numerical implementation for each type of transfer is summarized in Fig. 1.

**1. Grid mass**

$$m_i^n = \sum_p m_p \phi_{ip}^n \quad (\text{Eq. (13)})$$

**2. Grid velocity**

$$\text{FLIP: } \mathbf{v}_i^n = \frac{\sum_p \phi_{ip}^n m_p \mathbf{v}_p^n}{m_i^n} \quad (\text{Eq. (15)})$$

$$\text{APIC: } \mathbf{v}_i^n = \frac{\sum_p \phi_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{B}_p^n \cdot (\mathbf{D}_p^n)^{-1} \cdot (\mathbf{x}_i^n - \mathbf{x}_p^n))}{m_i^n} \quad (\text{Eqs. (16) and (17)})$$

$$\text{TPIC: } \mathbf{v}_i^n = \frac{\sum_p \phi_{ip}^n m_p (\mathbf{v}_p^n + \nabla \mathbf{v}_p^n \cdot (\mathbf{x}_i^n - \mathbf{x}_p^n))}{m_i^n} \quad (\text{Eqs. (16) and (19)})$$

**3. Grid force**

$$\mathbf{f}_i^n = - \sum_p V_p^n \nabla \phi_{ip}^n \cdot \boldsymbol{\sigma}_p^n + \sum_p \phi_{ip}^n m_p \mathbf{b}_p + \bar{\mathbf{f}}_i \quad (\text{Eqs. (11) and (12)})$$

**4. Update grid velocity**

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{a}_i^n = \mathbf{v}_i^n + \Delta t \frac{\mathbf{f}_i^n}{m_i^n} \quad (\text{Eqs. (20) and (21)})$$

**5. Update particle velocity**

$$\text{FLIP: } \mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t \sum_i \phi_{ip}^n \mathbf{a}_i^n \quad (\text{Eq. (22)})$$

$$\text{APIC \& TPIC: } \mathbf{v}_p^{n+1} = \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1} \quad (\text{Eq. (23)})$$

**5\*. Update additional particle quantity (APIC)**

$$\mathbf{B}_p^{n+1} = \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1} \otimes (\mathbf{x}_i^n - \mathbf{x}_p^n) \quad (\text{Eq. (25)})$$

**6. Update particle position**

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1} \quad (\text{Eq. (31)})$$

**Fig. 1.** Summary of numerical implementations for FLIP, APIC and TPIC transfers.

**Remark 1.** In the MLS-MPM with the linear polynomial basis [19], the velocity gradient  $\nabla \mathbf{v}_p^{n+1}$  is reduced to the affine velocity matrix  $\mathbf{B}_p^{n+1} \cdot (\mathbf{D}_p^n)^{-1}$  as

$$\nabla \mathbf{v}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} \otimes \nabla \hat{\phi}_{ip}^n = \left[ \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1} \otimes (\mathbf{x}_i - \mathbf{x}_p^n) \right] \cdot (\mathbf{D}_p^n)^{-1} = \mathbf{B}_p^{n+1} \cdot (\mathbf{D}_p^n)^{-1}, \quad (33)$$

where  $\hat{\phi}_{ip}^n$  is the MLS shape function [19,26]. Thus, APIC and TPIC are reduced to the MLS-MPM formulation when the MLS shape function is applied.

### 2.8. Critical timestep criterion

The Courant–Friedrichs–Lewy (CFL) condition [27] must be considered for obtaining a stable solution in explicit time integration. The speed of sound for the elastic material of a solid can be calculated as

$$c = \sqrt{\frac{K + \frac{4}{3}G}{\rho}}, \quad (34)$$

where  $K$  represents the bulk modulus,  $G$  represents the shear modulus, and  $\rho$  represents the density. Thus, the critical timestep is expressed as

$$\Delta t_{\text{cr}} = \frac{h}{\max_p (c_p + \|\mathbf{v}_p\|)}, \quad (35)$$

where  $h$  represents the grid spacing, and  $c_p$  and  $\mathbf{v}_p$  represent the sound speed and velocity of particle  $p$ , respectively.

For an incompressible Newtonian fluid, the effect of viscous diffusion is considered in the following critical timestep [28]:

$$\Delta t_{\text{cr}} = \min \left( \frac{h}{\max_p (c_p + \|\mathbf{v}_p\|)}, \frac{h^2}{\nu} \right), \quad (36)$$

where  $\nu = \mu/\rho$  ( $\mu$ : dynamic viscosity) represents the kinematic viscosity.

In explicit time integration schemes, the Courant number,  $C_r \leq 1$ , should satisfy

$$\Delta t = C_r \Delta t_{\text{cr}}. \quad (37)$$

### 3. Differences among transfers

In this section, the dissipation (i.e., loss of physical information) in the particle-to-grid and grid-to-particle transfers is evaluated to determine the differences among the FLIP, PIC, APIC, and TPIC transfers. We assume that the kernel functions satisfy the partition of unity and linear field reproduction conditions, which are expressed, respectively, as

$$\sum_i \phi_{ip} = 1, \quad (38)$$

$$\sum_i \phi_{ip} \mathbf{x}_i = \mathbf{x}_p. \quad (39)$$

Their spatial gradient forms are expressed as follows:

$$\sum_i \nabla \phi_{ip} = \mathbf{0}, \quad (40)$$

$$\sum_i \mathbf{x}_i \otimes \nabla \phi_{ip} = \mathbf{I}, \quad (41)$$

where  $\mathbf{I}$  is the second-order identity tensor.

Assuming that a single particle exists at  $\mathbf{x}_p$  with velocity  $\mathbf{v}_p$  and gradient  $\nabla \mathbf{v}_p$ , the velocity can be transferred to the grid using Eq. (15) as

$$\mathbf{v}_i^n = \frac{\phi_{ip}^n m_p \mathbf{v}_p^n}{\phi_{ip}^n m_p} = \mathbf{v}_p^n. \quad (42)$$

The summations  $\sum_p$  in Eqs. (13) and (15) vanish because of the assumption that only one particle exists. If no external forces exist, we can set  $\mathbf{v}_i^{n+1} = \mathbf{v}_i^n$ , implying that  $\mathbf{v}_i^{n+1} = \mathbf{v}_p^n$ . Thus, both the FLIP (Eq. (22)) and PIC (Eq. (23)) methods can preserve the particle velocity  $\mathbf{v}_p^n$  as follows:

$$\mathbf{v}_p^{\text{FLIP},n+1} = \mathbf{v}_p^{\text{PIC},n+1} = \mathbf{v}_p^n. \quad (43)$$

Herein, we use  $\mathbf{a}_i^n = \mathbf{0}$  for FLIP transfer and Eq. (38) for PIC transfer. However, the particle velocity gradient  $\nabla \mathbf{v}_p$  is dissipated as follows:

$$\nabla \mathbf{v}_p = \mathbf{v}_p^n \otimes \sum_i \nabla \phi_{ip}^n = \mathbf{0}, \quad (44)$$

where Eq. (40) is used. Although both the FLIP and PIC transfers lose the particle velocity gradient, FLIP reduces the numerical dissipation because it considers the grid acceleration as expressed in Eq. (22).

In APIC transfer, the particle velocity is transferred to the grid as follows:

$$\mathbf{v}_i^n = \mathbf{v}_p^n + \mathbf{B}_p^n \cdot (\mathbf{D}_p^n)^{-1} \cdot (\mathbf{x}_i^n - \mathbf{x}_p^n). \quad (45)$$

The updated particle velocity can be obtained using Eq. (23) as follows:

$$\begin{aligned} \mathbf{v}_p^{n+1} &= \mathbf{v}_p^n \sum_i \phi_{ip}^n + \mathbf{B}_p^n \cdot (\mathbf{D}_p^n)^{-1} \cdot \sum_i \phi_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n) \\ &= \mathbf{v}_p^n, \end{aligned} \quad (46)$$

where Eqs. (38) and (39) are used. By contrast, the updated particle velocity gradient is not fully preserved as

$$\begin{aligned} \nabla \mathbf{v}_p^{n+1} &= \mathbf{v}_p^n \otimes \sum_i \nabla \phi_{ip}^n + \mathbf{B}_p^n \cdot (\mathbf{D}_p^n)^{-1} \cdot \sum_i (\mathbf{x}_i^n - \mathbf{x}_p^n) \otimes \nabla \phi_{ip}^n \\ &= \mathbf{B}_p^n \cdot (\mathbf{D}_p^n)^{-1}, \end{aligned} \quad (47)$$

where Eqs. (40) and (41) are used. Although the particle velocity gradient is not preserved, the angular momentum is conserved in APIC transfer, as described by Jiang et al. [17].

In contrast to APIC transfer, the TPIC method can preserve the velocity gradient  $\nabla \mathbf{v}_p$  instead of conserving the angular momentum. From Eq. (19), the grid velocity can be expressed as

$$\mathbf{v}_i^n = \mathbf{v}_p^n + \nabla \mathbf{v}_p^n \cdot (\mathbf{x}_i^n - \mathbf{x}_p^n). \quad (48)$$

Thus, the updated particle velocity  $\mathbf{v}_p$  and its gradient  $\nabla \mathbf{v}_p$  can be obtained assuming  $\mathbf{v}_i^{n+1} = \mathbf{v}_i^n$ , as follows:

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n \sum_i \phi_{ip}^n + \nabla \mathbf{v}_p^n \cdot \sum_i \phi_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n) = \mathbf{v}_p^n, \quad (49)$$

$$\nabla \mathbf{v}_p^{n+1} = \mathbf{v}_p^n \sum_i \nabla \phi_{ip}^n + \nabla \mathbf{v}_p^n \cdot \sum_i (\mathbf{x}_i^n - \mathbf{x}_p^n) \otimes \nabla \phi_{ip}^n = \nabla \mathbf{v}_p^n, \quad (50)$$

where Eq. (23) is used. Because the velocity gradient possesses the angular information in its skew-symmetric component, TPIC transfer can adequately describe the angular motion, as indicated by the numerical examples presented in Section 5.3 and Section 5.4.

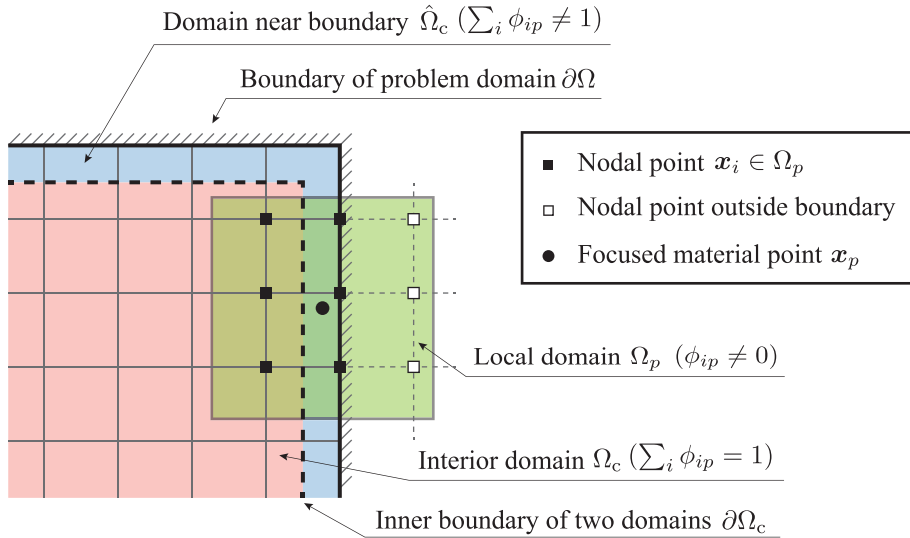
MLS-MPM [19] can satisfy the conservation of angular momentum, in addition to preserving the velocity gradient, because  $\nabla \mathbf{v}_p^n = \mathbf{B}_p^n \cdot (\mathbf{D}_p^n)^{-1}$ . However, the shape of the MLS-MPM kernel gradient differs from that of the original kernel gradient, as described in Section 4.

The preservation of the velocity gradient proved in this section is not generally satisfied when there are more particles. However, one can observe that if the velocity is distributed linearly in the space (i.e., the velocity gradient is constant in space), TPIC again holds this preservation (because the first-order Taylor series approximation is used) and the velocity gradient is never lost in the transfer process. This characteristic of the transfer achieves less dissipative but stable simulations.

#### 4. Kernel correction

MPM can cause stress oscillations near boundaries when boundary conditions are imposed explicitly, e.g., when the bound grid velocity (or momentum) is set to zero. When affine-type transfers such as APIC or TPIC are used,





**Fig. 2.** Condition in which a particle is near the boundary (quadratic B-spline kernel case).

this instability worsens, resulting in simulation failure. In [22], a method was proposed in which image particles are used to reduce these artifacts. In the present study, we address this problem by modifying the kernel such that both the partition of unity and linear field reproduction conditions are satisfied, as expressed in Eqs. (38)–(41).

These conditions are typically satisfied without corrections but not when the particle is near a boundary, as shown in Fig. 2. We found that such inconsistencies cause stress oscillations near boundaries and yield incorrect results as also reported in [22]. Although the instability problem can occur for any kernel function, the following quadratic B-spline function is used in the present study:

$$w(\xi) = \begin{cases} -\|\xi\|^2 + \frac{3}{4} & 0 \leq |\xi| < \frac{1}{2} \\ \frac{1}{2}\|\xi\|^2 - \frac{3}{2}\|\xi\| + \frac{9}{8} & \frac{1}{2} \leq |\xi| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |\xi| \end{cases}, \quad (51)$$

$$\phi_{ip} = \phi_i(\mathbf{x}_p) = w\left(\frac{\mathbf{x}_p - \mathbf{x}_i}{h}\right) w\left(\frac{\mathbf{y}_p - \mathbf{y}_i}{h}\right) w\left(\frac{\mathbf{z}_p - \mathbf{z}_i}{h}\right), \quad (52)$$

where  $h$  represents the grid spacing.

As shown in Fig. 2, the partition of unity condition (Eq. (38)) is not satisfied before modifications when the particle is near the boundary. In [8], the kernels for the boundary nodes and the nodes next to the boundary were modified (Fig. 3(a)). This method can satisfy the partition of unity condition (Eq. (38)); however, the linear field reproduction condition (Eq. (39)) is still not satisfied.

To satisfy Eqs. (38) and (39), the kernel  $\phi_i(\mathbf{x})$  is modified as follows:

$$\hat{\phi}_i(\mathbf{x}) = C(\mathbf{x}, \mathbf{x}_i) \phi_i(\mathbf{x}_p). \quad (53)$$

The correction function  $C$  is constructed using the weighted least squares (WLS) scheme as

$$C(\mathbf{x}, \mathbf{x}_i) = \mathbf{P}(\mathbf{x} - \mathbf{x}_p) \mathbf{M}(\mathbf{x}_p)^{-1} \mathbf{P}^\top(\mathbf{x}_i - \mathbf{x}_p), \quad (54)$$

where the matrix  $\mathbf{M}$  is expressed as

$$\mathbf{M}(\mathbf{x}_p) = \sum_i \phi_{ip} \mathbf{P}^\top(\mathbf{x}_i - \mathbf{x}_p) \mathbf{P}(\mathbf{x}_i - \mathbf{x}_p). \quad (55)$$

The linear polynomial basis  $\mathbf{P}$  is expressed in three dimensions as

$$\mathbf{P}(\mathbf{x}) = [1, x, y, z], \quad m = 4. \quad (56)$$

Next, we represent the inverse of the moment matrix  $\mathbf{M}^{-1}$  using the following block notation:

$$\mathbf{M}^{-1} = \begin{bmatrix} C_1 & \mathbf{C}_2^\top \\ \mathbf{C}_2 & \mathbf{C}_3 \end{bmatrix}, \quad (57)$$

where  $C_1$  is a scalar,  $\mathbf{C}_2$  is a vector with a length of  $d$  ( $d$  represents the dimension of the problem), and  $\mathbf{C}_3$  is a  $d \times d$  matrix. Finally, the corrected kernel can be simplified based on Eq. (53) as

$$\hat{\phi}_{ip} = \hat{\phi}_i(\mathbf{x}_p) = C(\mathbf{x}_p, \mathbf{x}_i)\phi_i(\mathbf{x}_p) = (C_1 + \mathbf{C}_2 \cdot (\mathbf{x}_i - \mathbf{x}_p))\phi_{ip}. \quad (58)$$

We note that, in the above formulation, the material point  $\mathbf{x}_p$  is fixed in Eqs. (53) and (54), i.e., the WLS scheme. However, if the point  $\mathbf{x}_p$  is considered a variable, the scheme becomes the MLS scheme. This difference affects the kernel gradient formulation [29].

**Remark 2.** Using corrected kernel  $\hat{\phi}_i(\mathbf{x})$ , a continuous function  $u(\mathbf{x})$  can be approximated from its nodal values  $u_i = u(\mathbf{x}_i)$  as

$$u(\mathbf{x}) = \sum_i \hat{\phi}_i(\mathbf{x})u_i. \quad (59)$$

When  $u(\mathbf{x}) = \mathbf{P}(\mathbf{x} - \mathbf{x}_p)$ , one can be obtained from Eqs. (53)–(55), and (59) as

$$\mathbf{P}(\mathbf{x} - \mathbf{x}_p) = \mathbf{P}(\mathbf{x} - \mathbf{x}_p)\mathbf{M}(\mathbf{x}_p)^{-1} \sum_i \phi_{ip}\mathbf{P}^\top(\mathbf{x}_i - \mathbf{x}_p)\mathbf{P}(\mathbf{x}_i - \mathbf{x}_p) = \mathbf{P}(\mathbf{x} - \mathbf{x}_p). \quad (60)$$

Thus, if the linear polynomial basis  $\mathbf{P}$  (Eq. (56)) is used, the corrected kernel  $\hat{\phi}_i(\mathbf{x})$  satisfies the partition of unity (Eq. (38)) and the linear field reproduction condition (Eq. (39)).

**Remark 3.** When the WLS or MLS scheme is used in particle methods, the moment matrix  $\mathbf{M}$  in Eq. (54) may be singular or ill-conditioned, and the matrix is not invertible due to the small number of sample points (typically, particles). This can be avoided by diagonalizing the moment matrix (e.g., [30]). However, the moment matrix  $\mathbf{M}$  in the presented WLS scheme is always invertible because the scheme uses nodal points as sample points. In particular, when the linear polynomial (Eq. (56)) is used, the minimum number of surrounding nodal points  $2d$  (the number of nodal points of an element, where  $d$  represents the dimension of the problem) are always equal to or larger than the required number of sample points  $m = d + 1$ .

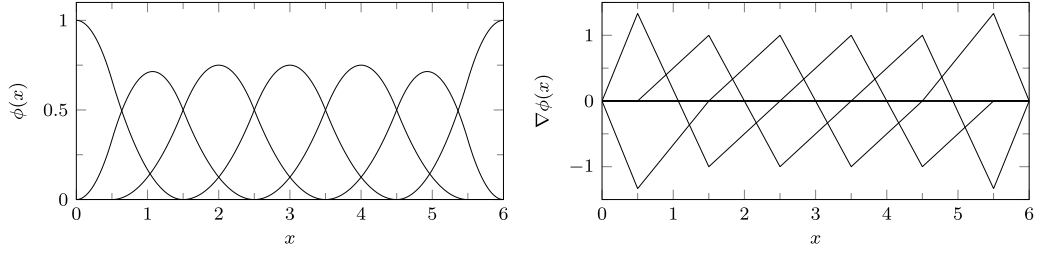
The least squares approach has been extensively used for particle methods, such as the reproducing kernel method [31], smooth particle hydrodynamics [32–34], and MPM [19,26,35]. In this study, we use this technique to improve the numerical stability near the boundary in MPM. As described by Liu et al. [31],  $C_1 = 1$  and  $\mathbf{C}_2 = \mathbf{0}$  when the particle is not close to the domain boundary, whereas  $C_1 \neq 1$  and  $\mathbf{C}_2 \neq \mathbf{0}$  when the particle is close to the domain boundary. Thus, we can use Eq. (58) only for the particles near boundaries, as follows:

$$\Phi_i(\mathbf{x}_p) = \begin{cases} \phi_{ip} & \text{if } \mathbf{x}_p \in \Omega_c \\ \hat{\phi}_{ip} & \text{otherwise (if } \mathbf{x}_p \in \hat{\Omega}_c). \end{cases} \quad (61)$$

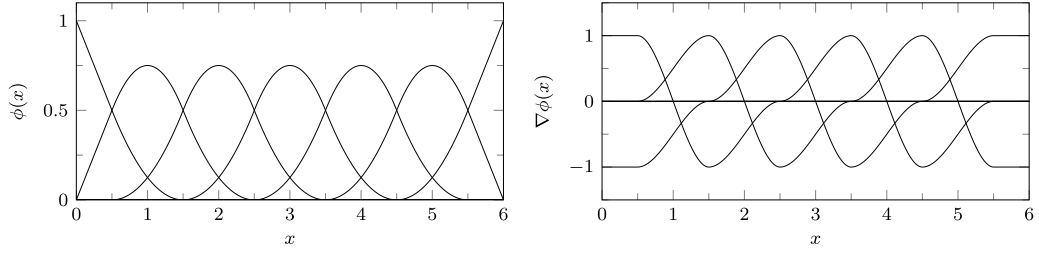
Differentiating Eq. (53) with respect to  $\mathbf{x}$  yields [26]

$$\nabla \hat{\phi}_{ip} = \nabla \hat{\phi}_i(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_p} = (\mathbf{C}_2 + \mathbf{C}_3 \cdot (\mathbf{x}_i - \mathbf{x}_p))\phi_{ip}. \quad (62)$$

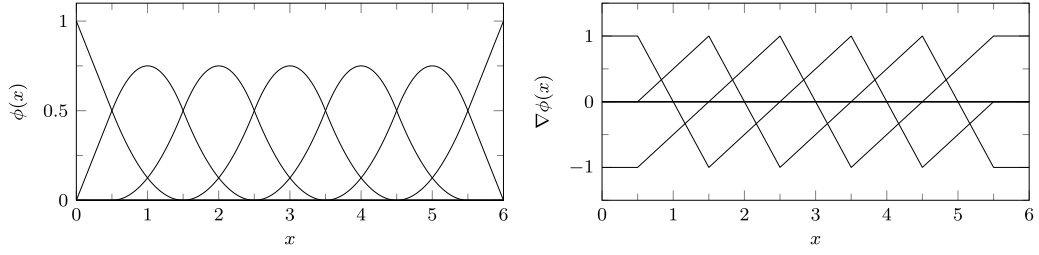
This corrected kernel gradient satisfies Eqs. (40) and (41) even near the boundary, in contrast to the kernel gradient proposed by Steffen et al. [8]. However, its shape becomes different from that of the original kernel gradient  $\nabla \phi_{ip}$  as shown Fig. 3(b). We examined the effect of this small difference through a simple simulation, as described in Section 5.2. To retain the shape of the kernel gradient, the MLS scheme should be used instead of the WLS scheme, as shown in Fig. 3(c). However, the gradient of the kernel corrected using MLS introduces computational complexity because  $\mathbf{M}(\mathbf{x})^{-1}\mathbf{P}(\mathbf{x}_i - \mathbf{x})$  in Eq. (54) must be differentiated with respect to  $\mathbf{x}$  [29,31].



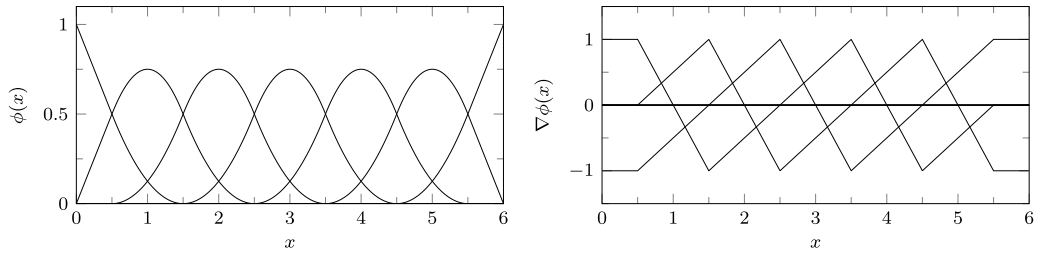
(a) Steffen et al. [8]



(b) MLS-MPM [19, 26]



(c) Kernel correction based on the MLS.



(d) Kernel correction based on the WLS.

**Fig. 3.** Quadratic B-spline functions.

A simple solution involves the use of the original  $\nabla\phi_{ip}$  when the particle is not close to the boundary, as follows:

$$\nabla\hat{\phi}_i(\mathbf{x}_p) = \begin{cases} \nabla\phi_{ip} & \text{if } \mathbf{x}_p \in \Omega_c \\ \nabla\hat{\phi}_{ip} & \text{otherwise (if } \mathbf{x}_p \in \hat{\Omega}_c). \end{cases} \quad (63)$$

Herein, Eq. (63) needs to be a continuous function for consistency, which implies that  $\nabla\phi_i(x_p) = \nabla\hat{\phi}_i(x_p)$  for all  $x_p \in \chi_j$  in one dimension, where  $\chi_j$  denotes the knots of the B-spline function on the inner boundary  $\partial\Omega_c$  in Fig. 2.

This can be easily proved for the quadratic B-spline function using  $C_3 = \frac{4}{h^2} \mathbf{I}$  and the cubic B-spline function using  $C_3 = \frac{3}{h^2} \mathbf{I}$  (the simplified  $C_3$  was given in [17]). As kernel functions in more than one dimension are constructed by products of one-dimensional functions, as expressed in Eq. (52), the continuity is preserved in the multidimensional problem. Eqs. (61) and (63) produce the functions presented in Fig. 3(d).

**Remark 4.** When  $\sum_i \phi_{ip} = 1$  is satisfied, i.e., the particle is not close to the boundary, the WLS scheme in Eq. (62) is reduced to the MLS-MPM formulation [19], which is also described by Nakamura et al. [26]. Thus, for simplicity, we refer to the formulation applying WLS in the entire domain as MLS-MPM (Fig. 3(b)).

According to Eqs. (61) and (63), the WLS kernel correction needs to be applied only when the particle is close to a boundary. Thus, the computational cost of the presented WLS kernel correction is insignificant. In the case of the quadratic B-spline function, MLS (Fig. 3(c)) and WLS (Fig. 3(d)) produce the same results in the domain  $\hat{\Omega}_c$ ; however, they are generally different. As an example, the cubic B-spline function is shown in Fig. 4. Fig. 4 shows slight differences in the gradient functions; for example, the kernel gradient of MLS (Fig. 4(c)) first increases and then decreases from the left boundary, whereas that of WLS (Fig. 4(d)) only decreases at the beginning.

## 5. Numerical simulations

The proposed TPIC transfer and kernel correction were validated based on five types of numerical simulations. In the first two examples, the proposed kernel correction method was validated using elastic and elastoplastic materials. In these simulations, the deformations were kept relatively small, because large deformations can easily result in unstable behavior without kernel correction. In the last three simulations, we validated the TPIC transfer method, including cases of large particle movements and deformations. Moreover, when the B-spline functions were used without the kernel correction, we applied the modification proposed by Steffen et al. [8] (Fig. 3(a)).

### 5.1. Simple shear simulation with linear elastic material

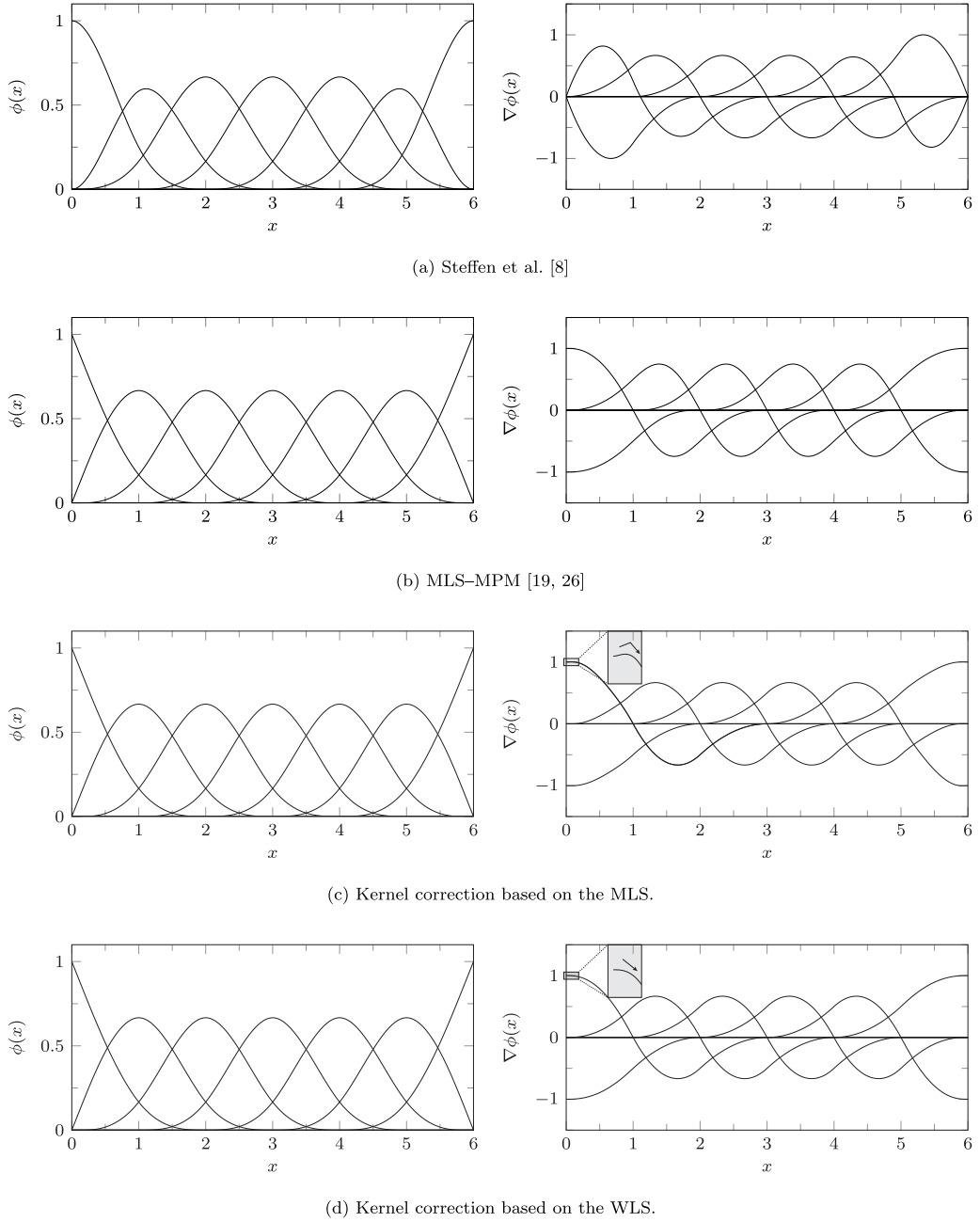
In the first example, we sheared the linear elastic material within a small-strain regime to validate the kernel correction method. As shown in Fig. 5, the sample is a linear elastic material with dimensions 1 m  $\times$  1 m, sandwiched between two no-slip plates. The bottom plate was fixed, and the top plate could slide horizontally (along the  $x$ -axis) at a velocity of 1 mm/s until the total horizontal displacement reached 0.1 mm. These boundary conditions were imposed on the boundary grid, directly modifying the nodal velocity  $\mathbf{v}_i^{n+1}$ . We performed a finite element simulation and considered the results as the analytical solution (Fig. 5).

The material parameters of the linear elastic material were the Young's modulus ( $E = 10^3$  MPa), Poisson's ratio ( $\nu = 0.333$ ), and density ( $\rho = 10^3$  kg/m<sup>3</sup>). The simulations were performed using the FLIP transfer (used in the original MPM [1]), APIC transfer [17], TPIC transfer, and MLS-MPM [19,26]. To analyze the effectiveness of the proposed kernel correction method, we used these approaches with and without the kernel correction, except for MLS-MPM (the kernel function for the MLS-MPM does not require modifications, as described in Section 4). The quadratic B-spline function was selected as the kernel function. In all the simulations, we applied the Courant number,  $C_r = 1$ , in Eq. (35) for the timestep.

Fig. 6 shows the von Mises stress distributions. In all cases, the simulations without the kernel correction exhibited stress oscillations near the top and bottom boundaries, which were also observed in a previous study [22]. As shown in Fig. 6, these stress oscillations caused incorrect distributions of the von Mises stress. The simulation results could not be improved by refining the mesh or by applying smaller timesteps in subsequent simulations (results are not presented), because the kernel function does not satisfy Eqs. (39) and (41). However, when the corrected kernel function (Fig. 3(d)) was used, these spurious oscillations were eliminated, and the results were consistent with the FEM results. This improvement could also be observed from Fig. 7 showing the distributions of the relative error computed by

$$\text{Error} = \sqrt{\frac{(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*) : (\boldsymbol{\sigma} - \boldsymbol{\sigma}^*)}{\boldsymbol{\sigma}^* : \boldsymbol{\sigma}^*}} \times 100\%, \quad (64)$$

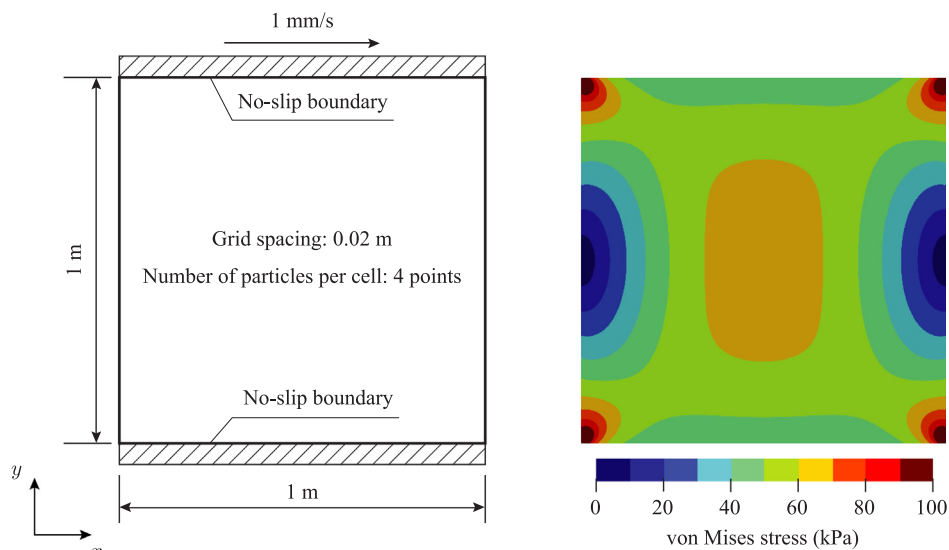
where  $\boldsymbol{\sigma}^*$  is the stress tensor obtained in FEM simulations. As MLS-MPM also satisfied the partition of unity and the linear field reproduction (Eqs. (38)–(41)), it yielded accurate results.

**Fig. 4.** Cubic B-spline functions.

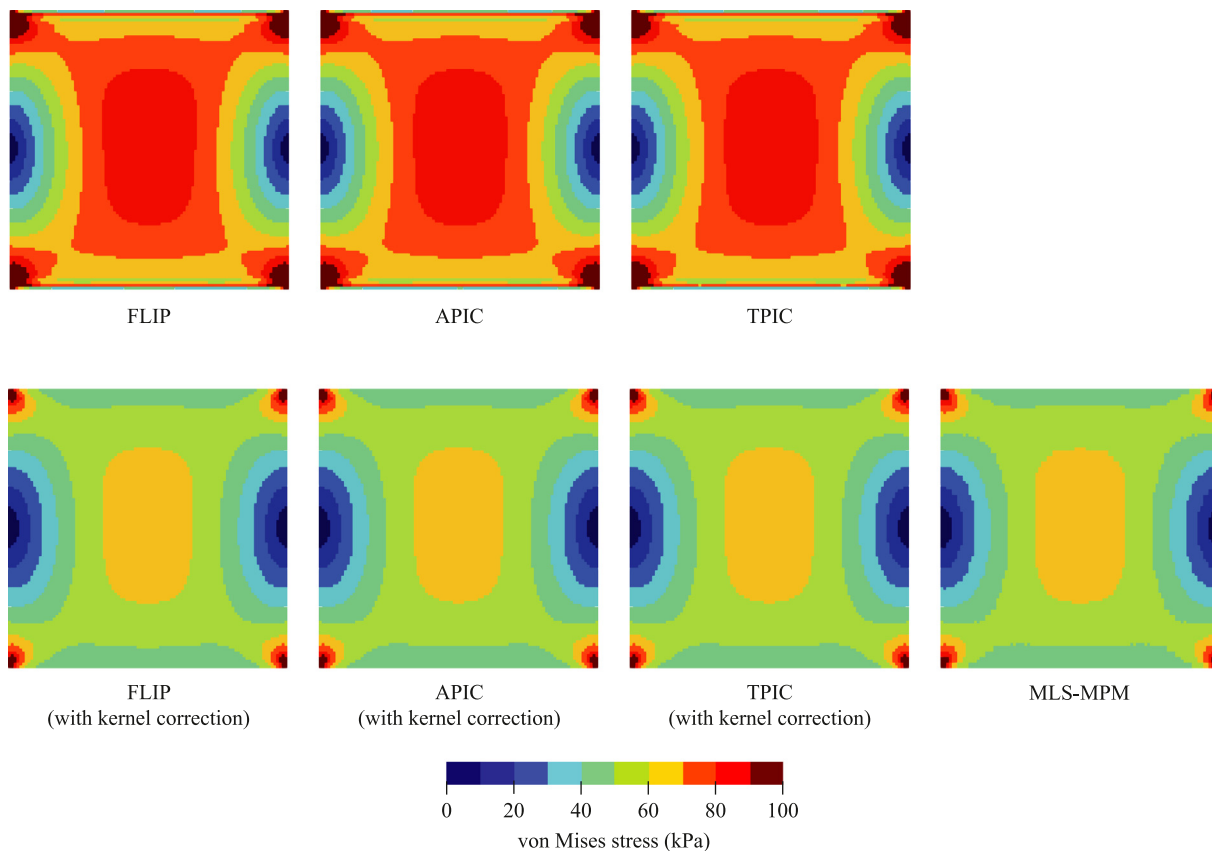
## 5.2. Strip footing

As the material model was purely elastic in the previous simulation, we validated the proposed kernel correction method using an elastoplastic material in a strip footing problem. For the von Mises model, the analytical solution for the bearing capacity was presented by Prandtl as follows [36]:

$$q_u = (2 + \pi)cB, \quad (65)$$



**Fig. 5.** Simple shear simulation with a linear elastic material. The geometry and boundary conditions of the problem (left) and the distributions of the von Mises stress in the FEM (right) are presented.



**Fig. 6.** Distribution of the von Mises stress in the simple shear simulation.

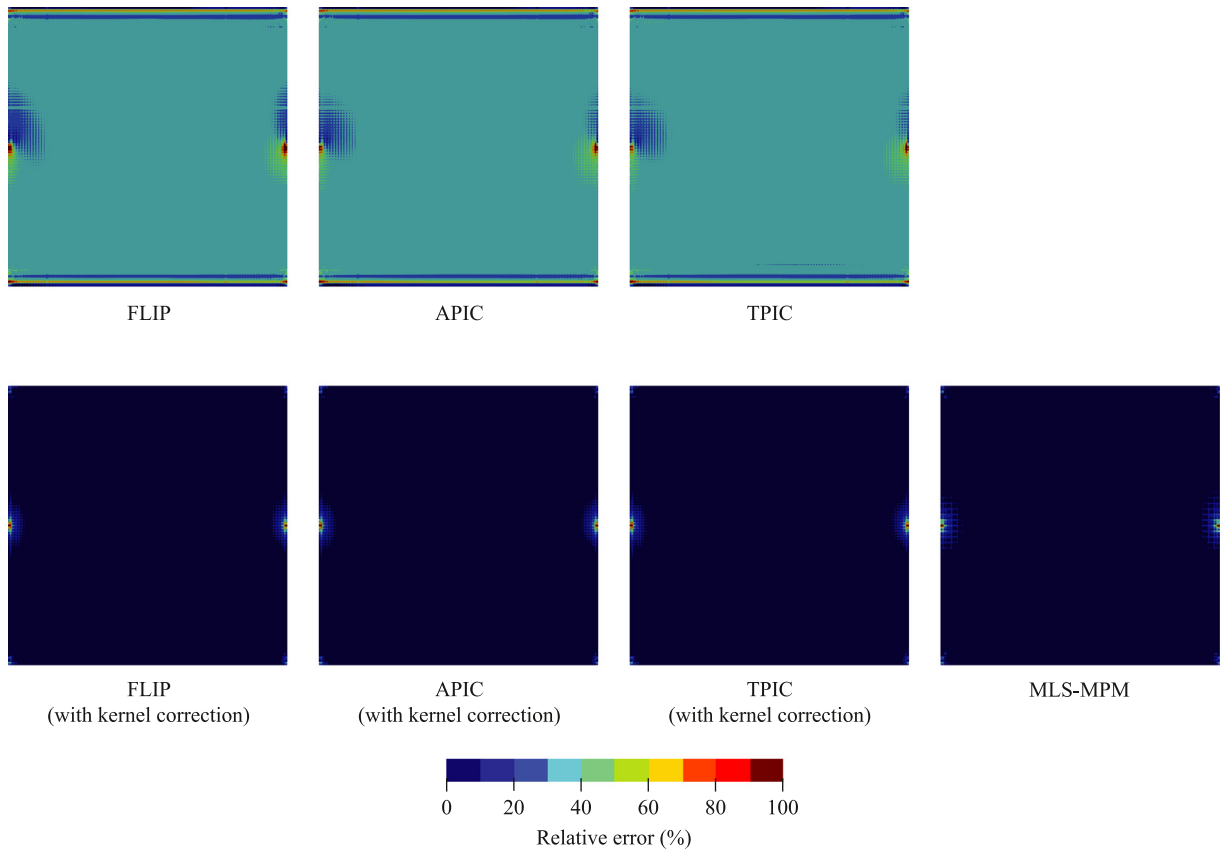


Fig. 7. Distribution of the relative error of the stress tensor in the simple shear simulation.

where  $c$  represents the cohesion of the soil, and  $B$  represents the footing width. The geometry and boundary conditions of the strip footing problem are shown in Fig. 8. As the domain was symmetric, only half of it was simulated. The footing was displaced vertically downward at a velocity of 4 mm/s, and the total displacement was 0.4 mm.

The yield function of the von Mises model is given as

$$\mathcal{F}(\boldsymbol{\sigma}) = \sqrt{3J_2(\boldsymbol{\sigma})} - \sigma_Y = 0, \quad (66)$$

where  $J_2$  is the second deviatoric stress invariant, and  $\sigma_Y > 0$  is the material constant for the yield stress. In the plane-strain condition, the yield stress  $\sigma_Y$  is associated with the cohesion  $c$  of the Mohr–Coulomb failure criterion as follows:

$$\sigma_Y = \sqrt{3}c. \quad (67)$$

For the evolution of the plastic strain, an associative flow rule was employed. The material parameters were the Young's modulus ( $E = 10^3$  MPa), Poisson's ratio ( $\nu = 0.333$ ), density ( $\rho = 10^3$  kg/m<sup>3</sup>), and cohesion ( $c = 10$  kPa).

As in the case of the simple shear simulations (Section 5.1), simulations were conducted using the FLIP, APIC, TPIC transfers with/without the kernel corrections, and the MLS-MPM. The quadratic B-spline function was selected as the kernel function. We used the Courant number,  $C_r = 1$ , in Eq. (35) to determine the timestep in all cases.

Fig. 9 presents the bearing capacity of the strip footing, and Fig. 10 presents the distributions of the deviatoric strain. Evidently, in all cases, large differences from the Prandtl solution are observed, and the slip lines are blurred when the kernel correction is not applied. These results could not be improved considerably with a finer mesh or smaller timesteps in subsequent simulations (results are not presented). Moreover, for the APIC and TPIC transfers, the simulations tended to be unstable near the boundary without the kernel correction, and the simulation failed in

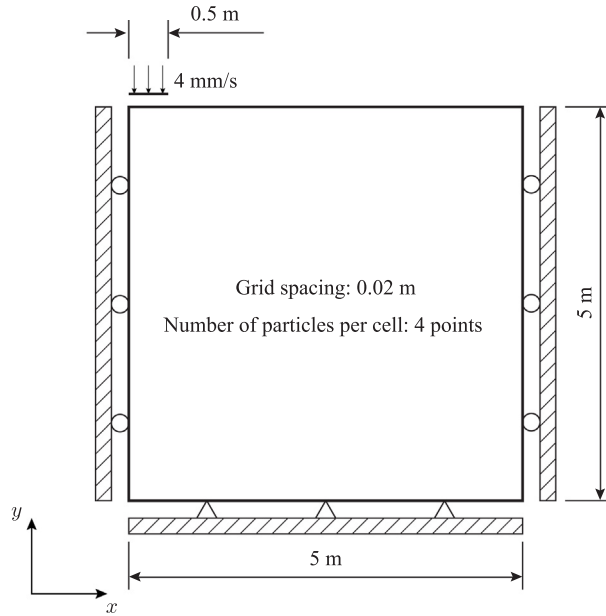


Fig. 8. Geometry and boundary conditions in the strip footing simulation.

the case of APIC transfer. Based on additional investigation, although the APIC transfer simulation could execute with smaller timesteps, the results were inaccurate (approximately identical to those for TPIC transfer without the kernel correction).

As shown in Fig. 9, the corrected kernel function improved considerably the results, which were consistent with the Prandtl solution. Furthermore, in the deviatoric strain distributions (Fig. 10), the slip lines were clearly observed. For the MLS-MPM, the displacement–force relationship was identical to that of the analytical solution. However, the slip line was relatively blurred compared with FLIP, APIC, and TPIC transfers with kernel corrections. This is because in MLS-MPM, the gradient of the kernel function differed from the original gradient; i.e., it became a smooth function, as shown in Fig. 3(b).

### 5.3. Free rotation of a disk

In this section, we validated the TPIC transfer by simulating the free rotation of a disk. As shown in Fig. 11, the sample is a linear elastic disk ( $E = 20$  MPa,  $\nu = 0.3$  and  $\rho = 10^3$  kg/m<sup>3</sup>) with a radius 0.5 m. A pseudo wall, where the boundary condition is not imposed, is also placed to check the effect of the kernel correction near the boundary. The disk is rotated by applying the initial angular velocity  $\omega_0 = 2\pi$  rad/s to each particle. In this case, the initial particle velocity  $\mathbf{v}_0$  is calculated by

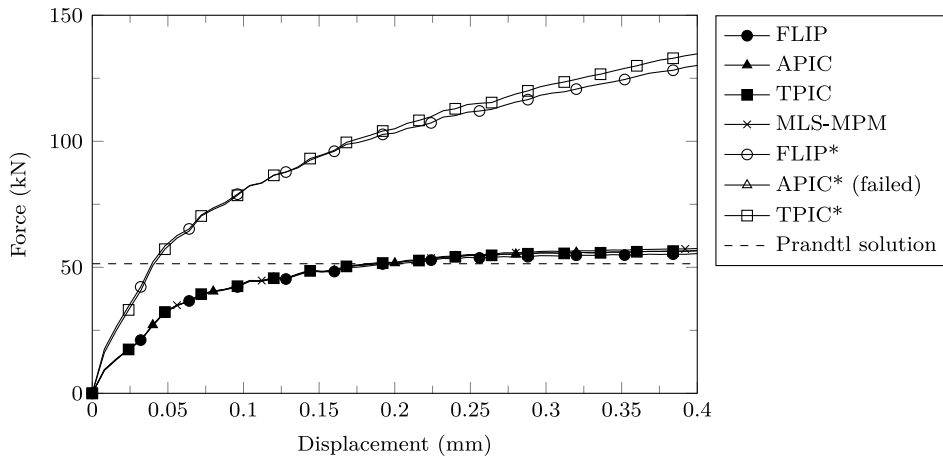
$$\mathbf{v}_0 = \boldsymbol{\omega}_0 \times \mathbf{r}_0. \quad (68)$$

Here,  $\mathbf{r}_0$  is the initial position vector from the center of the disk.

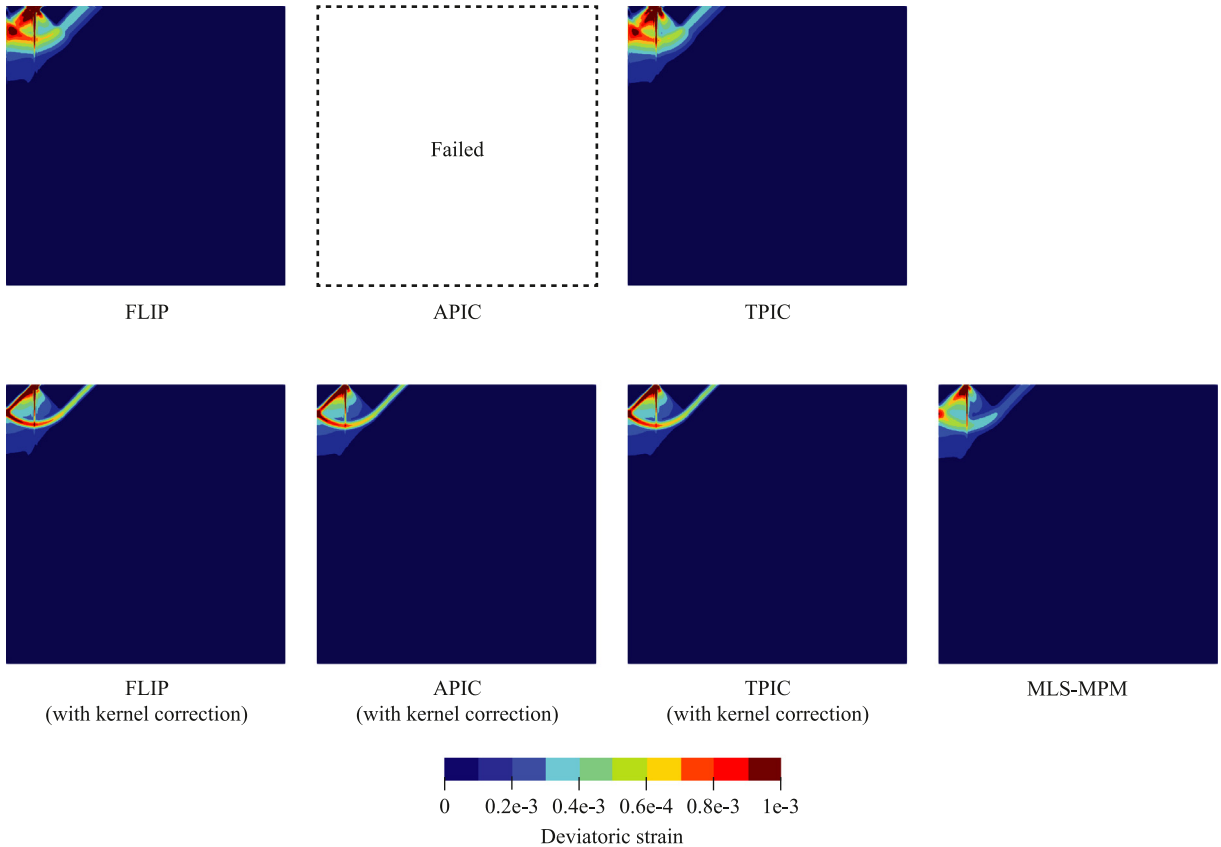
As in the case of previous simulations, simulations were conducted using the FLIP, APIC, TPIC transfers with/without the kernel corrections, and the MLS-MPM. In this simulation, we selected the cubic B-spline function which has a larger domain for the kernel correction than the quadratic one. We used the Courant number,  $C_r = 0.8$ , in Eq. (35) to determine the timestep in all cases.

Fig. 12 shows the distributions of the norm of velocity. The simulations for APIC and TPIC without kernel correction could not be successfully performed due to the unstable behaviors near the boundary. For the FLIP transfer, the simulation could execute, but spurious velocities were observed. Conversely, using the kernel correction



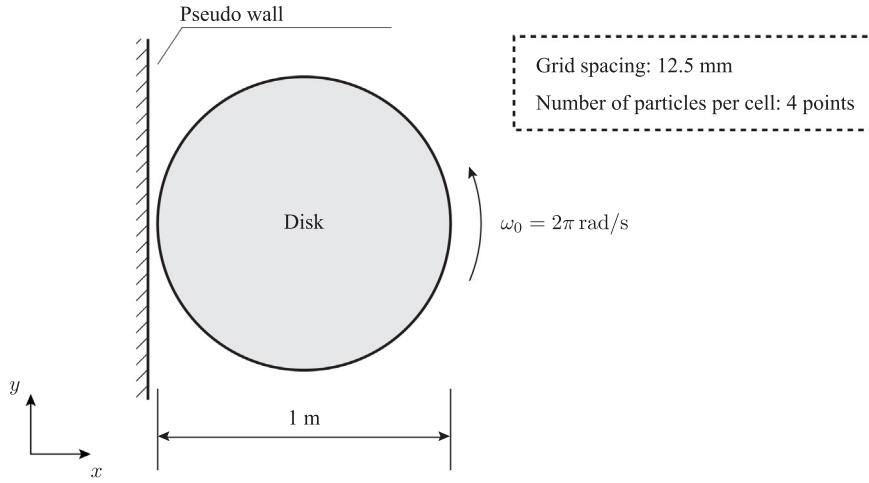


**Fig. 9.** Displacement–force relationship for the FLIP [16], APIC [17], TPIC (present study), and MLS-MPM [19,26] approaches in the strip footing simulations. The asterisk denotes “without kernel correction”. The simulation for APIC transfer without kernel correction could not be successfully performed (the simulation was unstable).

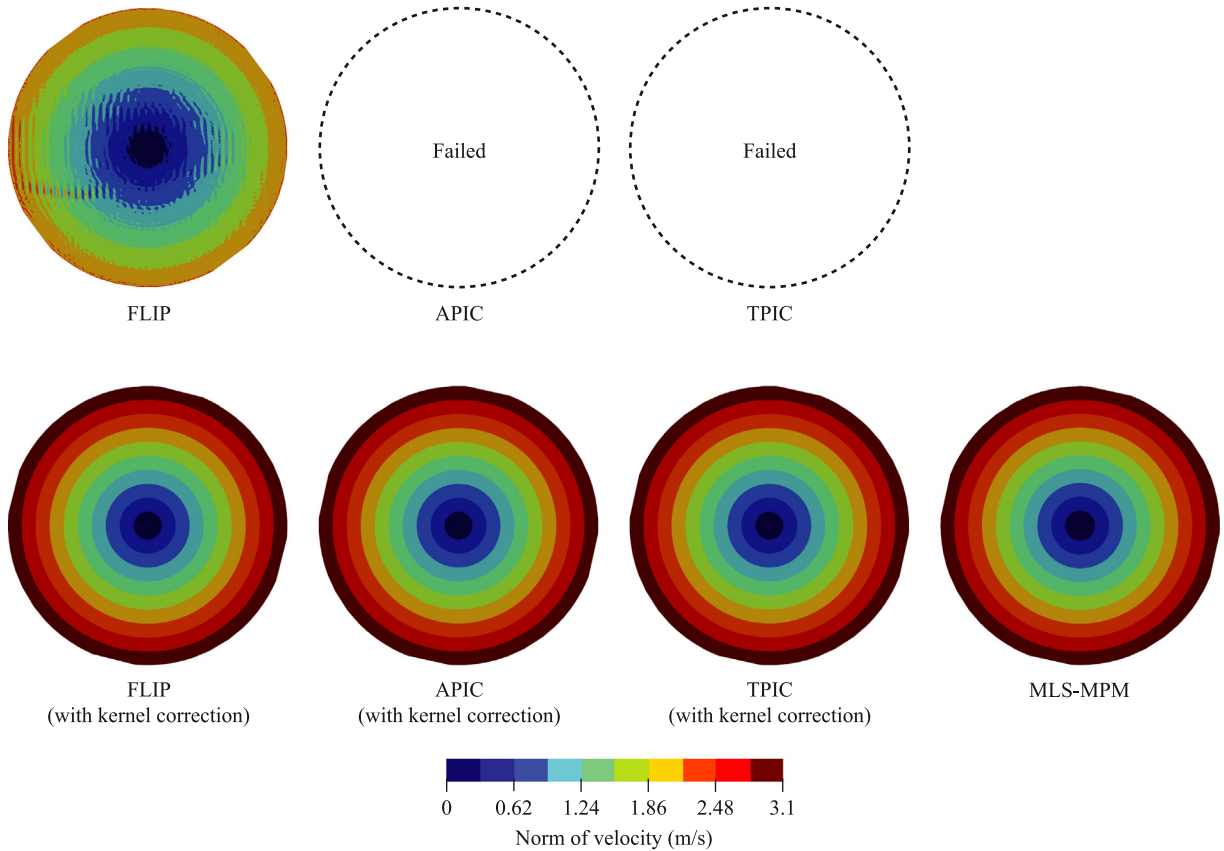


**Fig. 10.** Distribution of the deviatoric strain in the strip footing simulation.

improved considerably the results for all cases. This could also be verified by the distributions of the relative error of the angular velocity  $|\omega - \omega_0|/|\omega_0|$  as shown Fig. 13. From Fig. 13, the angular velocity for FLIP was slightly noisy compared with APIC, TPIC and MLS-MPM.

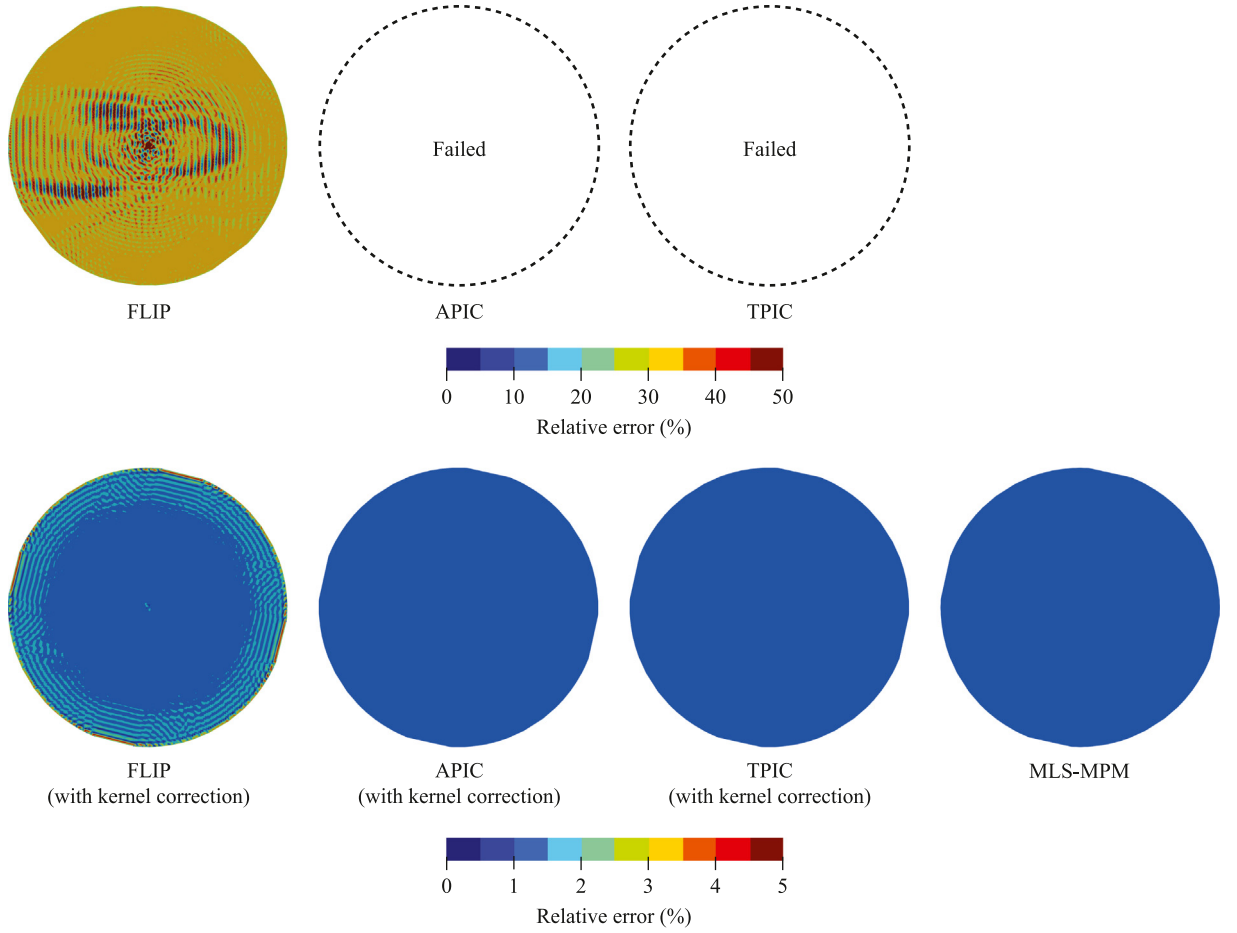


**Fig. 11.** Geometry and boundary conditions in the free rotation simulation.



**Fig. 12.** Distribution of the norm of velocity in the free rotation simulation.

Subsequently, we checked the angular momentum by computing  $\sum_p \mathbf{x}_p \times m_p \mathbf{v}_p$ . In Fig. 14, the angular momentum was normalized by its analytical value  $I\omega_0$ , where  $I$  is the moment of the inertia of the disk. Although the angular momentum is not strictly conserved in the TPIC transfer, no significant loss was observed in this



**Fig. 13.** Distribution of relative error of the angular velocity in the free rotation simulation.

example. Fig. 15 shows the kinematic energy  $\sum_p m_p \mathbf{v}_p \cdot \mathbf{v}_p / 2$  normalized by its analytical value  $I \omega_0^2 / 2$ . In contrast to the angular momentum, the slight dissipations were observed in all cases.

#### 5.4. Dam-break

The performance of TPIC transfer in dam-break simulations was analyzed. As discussed in Section 5.2, the APIC and TPIC transfers were unstable in the simulations without kernel corrections, even for small deformations; thus, the simulations involving material movement and large deformations, such as dam-break, failed immediately. Therefore, we used the corrected kernel functions for all cases. The geometry and boundary conditions of the simulations are shown in Fig. 16. The free-slip condition was imposed on all boundaries.

We used nearly incompressible Newtonian fluids as the fluid medium. The stress-strain relationship for the Newtonian fluids is given by

$$\boldsymbol{\sigma} = -p\mathbf{I} - \frac{2}{3}\mu\text{tr}(\mathbf{d})\mathbf{I} + 2\mu\mathbf{d}, \quad (69)$$

where  $\mathbf{I}$  is the second-order identity tensor,  $\mu$  represents the dynamic viscosity,  $\mathbf{d}$  represents the rate of deformation tensor, and  $p$  represents the fluid pressure, which can be determined using an equation of state (EOS). In this study,

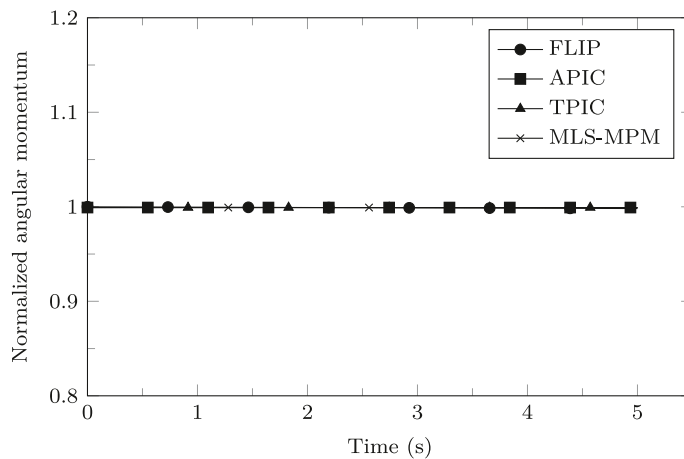


Fig. 14. Normalized angular momentum in the free rotation simulation.

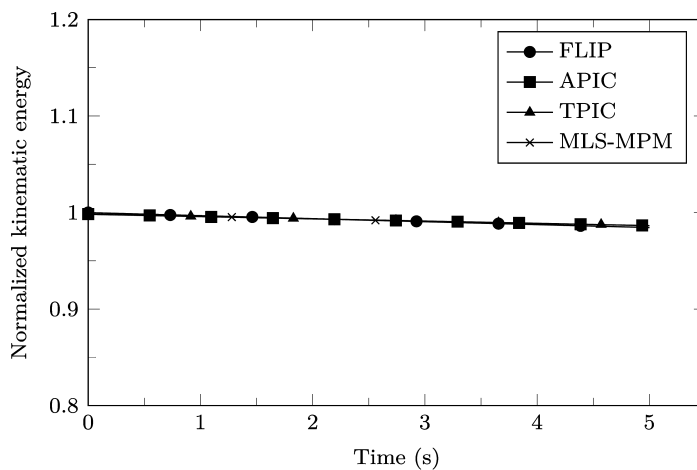


Fig. 15. Normalized kinematic energy in the free rotation simulation.

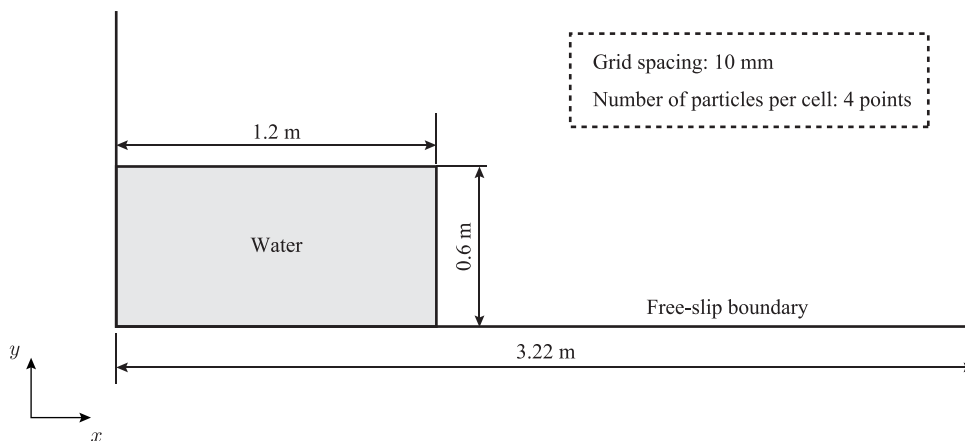


Fig. 16. Geometry and boundary conditions in the dam-break simulation.

Morris's EOS [37] was used, as follows:

$$p = c^2(\rho - \rho_{\text{ref}}), \quad (70)$$

where  $c$  represents the speed of sound, and  $\rho_{\text{ref}}$  represents the reference density. We used  $\mu = 1.01 \times 10^{-3}$  Pa s,  $c = 50$  m/s, and  $\rho_{\text{ref}} = 10^3$  kg/m<sup>3</sup> in all the simulations. The sound speed was selected to achieve the incompressibility of the fluid [38].

The simulations were performed using the FLIP, APIC, and TPIC transfers with kernel corrections. For the kernel function, the quadratic B-spline and the GIMP functions [6] were applied. We used the Courant number,  $C_r = 0.1$  (refer to Eq. (36)), to determine the timestep in all cases.

Fig. 17 shows the fluid pressure distributions in the dam-break simulations. As the FLIP transfer tended to be unstable, pressure oscillations occurred from the beginning of the simulation for both the quadratic B-spline and GIMP kernel functions. This spurious oscillation broke the thin fluid structures, as indicated by the water on the wall at  $t = 1.45$  s and the waterfront at  $t = 1.75$  s in Fig. 17. In comparison, the APIC and TPIC transfers reduced considerably the pressure oscillations, and a smoother pressure field was observed for both kernel functions. The thin sheet broken under FLIP transfer was preserved in these affine-type transfers. To eliminate completely the pressure oscillations, other techniques, such as the  $v$ - $p$  formulation [38], may be needed.

In these simulations, we observed that the TPIC transfer yielded approximately the same results as the APIC transfer, including the angular motions. As shown in Fig. 18, the TPIC method preserved the vorticity, similar to the APIC method. By contrast, the FLIP method produced noisy results.

### 5.5. Sand column collapse

The sand column collapse simulation is a simple method that can be used to evaluate the performance of TPIC transfer for an elastoplastic material under large deformations. The geometry and boundary conditions of the simulations are shown in Fig. 19.

The following Drucker–Prager model was applied for the material model of the sand:

$$\mathcal{F}(\boldsymbol{\sigma}) = \sqrt{J_2(\boldsymbol{\sigma})} + \alpha I_1(\boldsymbol{\sigma}) - \kappa = 0, \quad (71)$$

where  $J_2$  is the second deviatoric stress invariant,  $I_1$  is the first stress invariant, and  $\alpha$  and  $\kappa$  are the material parameters associated with the cohesion  $c$  and internal friction angle  $\phi$  in the Mohr–Coulomb model, respectively. Under the plane-strain condition, the material parameters are calculated as follows:

$$\alpha = \frac{\tan \phi}{\sqrt{9 + 12 \tan^2 \phi}}, \quad \kappa = \frac{3c}{\sqrt{9 + 12 \tan^2 \phi}}. \quad (72)$$

In this simulation, the following material parameters were used: the Young's modulus ( $E = 10$  MPa), Poisson's ratio ( $\nu = 0.3$ ), internal friction angle ( $\phi = 35^\circ$ ), cohesion ( $c = 0$  Pa), and density ( $\rho = 1.4 \times 10^3$  kg/m<sup>3</sup>). For the plastic flow rule, the following non-associative flow rule was employed:

$$\mathcal{G}(\boldsymbol{\sigma}) = \sqrt{J_2(\boldsymbol{\sigma})} + \beta I_1(\boldsymbol{\sigma}), \quad (73)$$

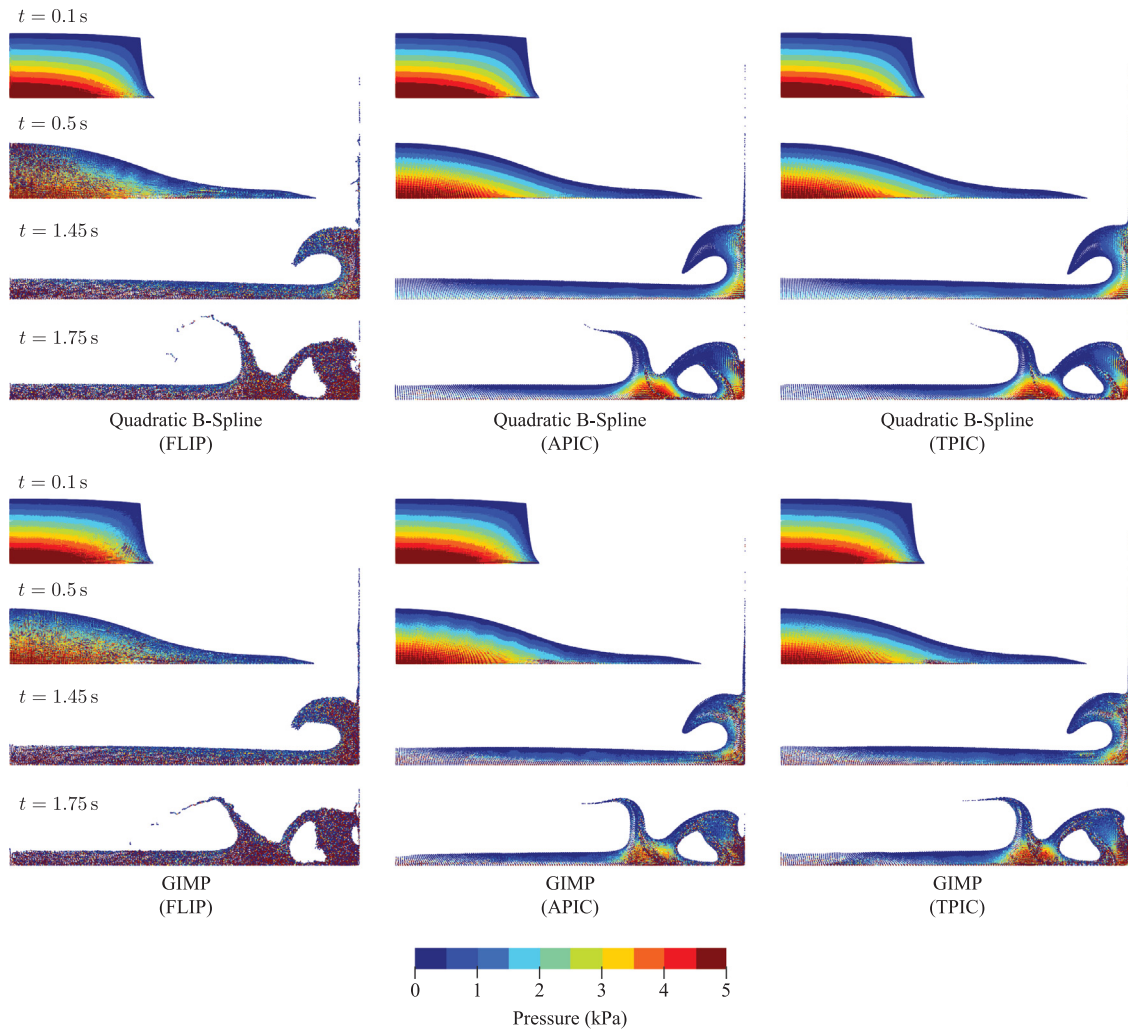
where  $\beta$  is derived from the dilatancy angle  $\psi$  as

$$\beta = \frac{\tan \psi}{\sqrt{9 + 12 \tan^2 \psi}}. \quad (74)$$

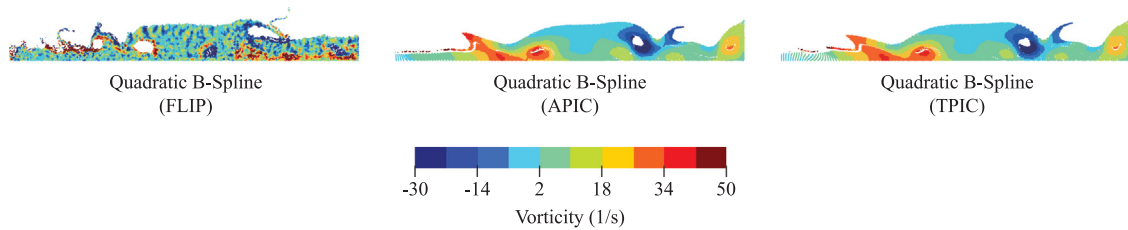
To prevent excessive plastic strain, we used  $\psi = 0^\circ$  in this study.

Similar to the dam-break simulations (Section 5.4), the sand column collapse simulations were performed using the FLIP, APIC, and TPIC transfers with kernel corrections. We selected the quadratic B-spline and GIMP functions [6] as the kernel functions. The simulations failed for the affine-type transfers when the kernel corrections were not applied. For example, Fig. 20 shows unstable behavior in TPIC. Herein, only the results for TPIC transfer are shown, because the APIC transfer simulation failed almost immediately. To determine the timestep, we used the Courant number,  $C_r = 1$  in Eq. (35).

Fig. 21 shows the distribution of the norm of the velocity. The FLIP transfer tended to accumulate spurious velocities because of the stress oscillations, as reported by Klár et al. [18]. The unstable behaviors were suppressed

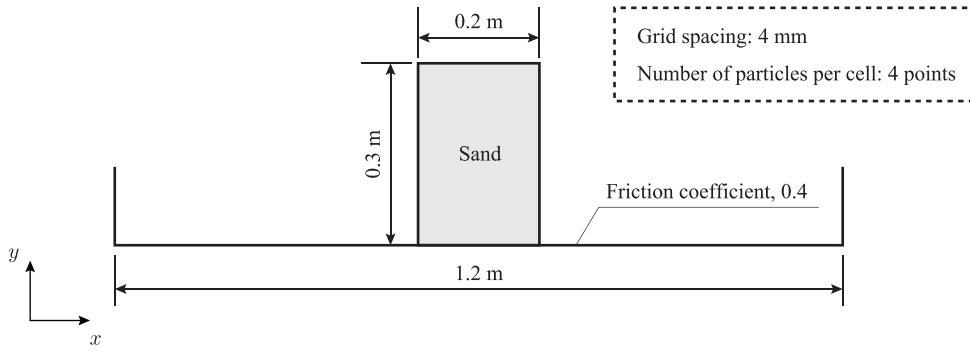


**Fig. 17.** Distribution of the pressure in the dam-break simulation.

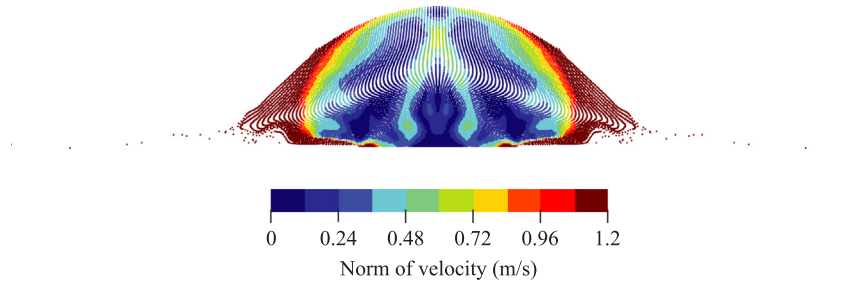


**Fig. 18.** Distribution of the vorticity at  $t = 3.4$  s.

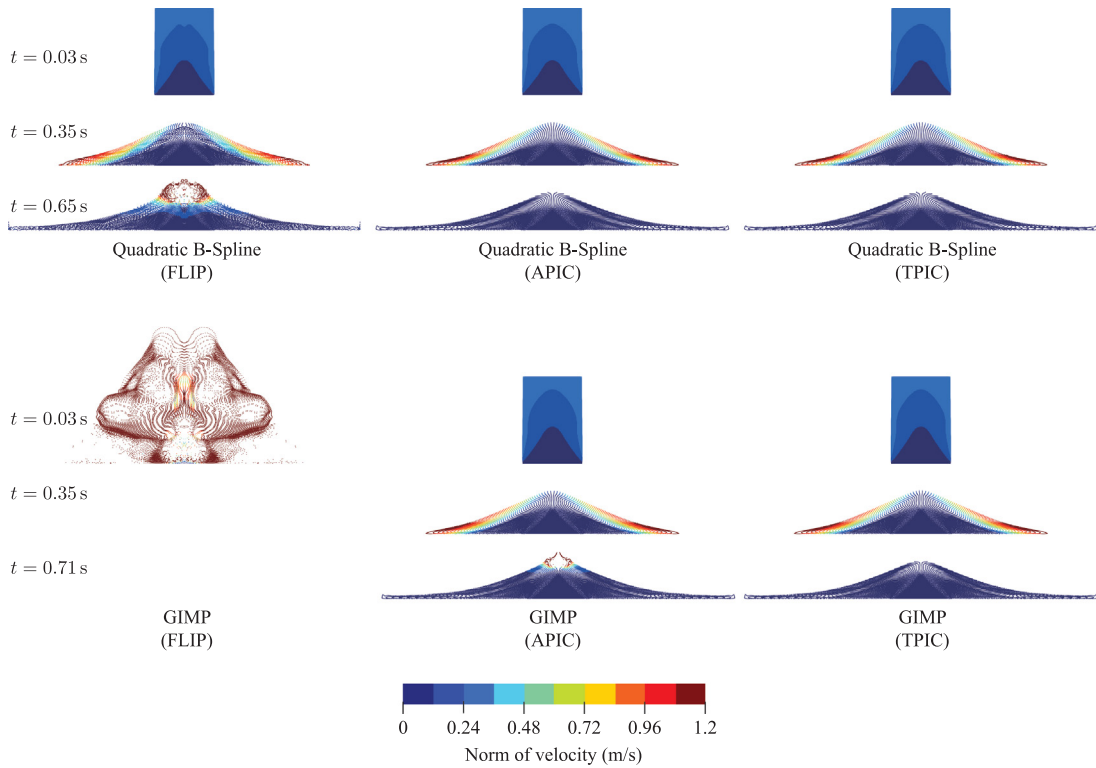
using smaller timesteps. For example, in the FLIP-GIMP case, the simulation ran successfully with a Courant number of approximately  $C_r < 0.75$ . In comparison, this unstable behavior was almost completely eliminated in both the APIC and TPIC transfers, although spurious velocity fluctuations were observed in the case of APIC transfer with the GIMP kernel function at  $t = 0.71$  s.



**Fig. 19.** Geometry and boundary conditions in the sand column collapse simulation.



**Fig. 20.** Unstable behavior without kernel correction at  $t = 0.224$  s. TPIC transfer with a quadratic B-spline function was used in the simulation. The result for the APIC transfer is not presented, as the simulation failed immediately.



**Fig. 21.** Distribution of the norm of the velocity in the sand column collapse simulation.

## 6. Conclusions

This study proposed Taylor-PIC (TPIC) transfer as a type of affine PIC transfer that combined the velocity projection based on the first-order Taylor series approximation and the PIC approach. Although the angular momentum conserved in APIC transfer is not conserved in the TPIC transfer, the velocity gradient is preserved during the transfer between the particles and the grid. Thus, the TPIC transfer can adequately describe angular motion, similar to the APIC method.

However, these affine-type transfers are severely unstable owing to the stress oscillations near the boundaries, and simulations can easily fail. We found that to reduce the spurious oscillations, the kernels should satisfy both the partition of unity ( $\sum_i \phi_i(x) = 1$ ) and linear field reproduction ( $\sum_i \phi_i(x)x_i = x$ ) conditions. Thus, a kernel correction method using the WLS scheme was developed in this study. As the kernel correction needs to be applied only for the particles near boundaries, the computational cost is insignificant.

The proposed TPIC transfer and kernel correction methods were validated using five different types of numerical simulations. We observed that using inconsistent kernel functions produced incorrect results owing to the stress oscillations, even for small deformations. For large-deformation problems, simulations of affine-type transfers immediately failed without the kernel correction. Thus, for many problems, it may be desirable or even necessary to apply the kernel correction. The numerical results verified that the corrected kernels successfully removed the spurious oscillations, and the results satisfactorily agreed with the analytical solutions. Similar to the APIC transfer, the proposed TPIC transfer exhibited the advantages of both the FLIP and PIC methods, i.e., less dissipation and stability, respectively.

In this study, no significant differences were observed between the TPIC and APIC transfer, including the angular motions; however, TPIC transfer loses the conservation of momentum, in contrast to APIC transfer. Therefore, the influence of this difference should be investigated further.

## 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.

## Data availability

No data was used for the research described in the article.

## Acknowledgment

## Funding

This work was supported by a JSPS, Japan KAKENHI grant [Grant Number JP19H00787] provided to the third author.

## References

- [1] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, *Comput. Methods Appl. Mech. Engrg.* 118 (1–2) (1994) 179–196.
- [2] D. Sulsky, S.-J. Zhou, H.L. Schreyer, Application of a particle-in-cell method to solid mechanics, *Comput. Phys. Comm.* 87 (1–2) (1995) 236–252.
- [3] F.H. Harlow, The particle-in-cell computing method for fluid dynamics, *Methods Comput. Phys.* 3 (1964) 319–343.
- [4] J.U. Brackbill, H.M. Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, *J. Comput. Phys.* 65 (2) (1986) 314–343.
- [5] J.U. Brackbill, D.B. Kothe, H.M. Ruppel, FLIP: a low-dissipation, particle-in-cell method for fluid flow, *Comput. Phys. Comm.* 48 (1) (1988) 25–38.
- [6] S.G. Bardenhagen, E.M. Kober, The generalized interpolation material point method, *Comput. Model. Eng. Sci.* 5 (6) (2004) 477–496.
- [7] 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.
- [8] 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.
- [9] R. Tielen, E. Wobbes, M. Möller, L. Beuth, A high order material point method, *Procedia Eng.* 175 (2017) 265–272.



- [10] 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.
- [11] Y. Bing, M. Cortis, T. Charlton, W. Coombs, C. Augarde, B-spline based boundary conditions in the material point method, *Comput. Struct.* 212 (2019) 257–274.
- [12] Y. Yamaguchi, S. Moriguchi, K. Terada, Extended B-spline-based implicit material point method, *Internat. J. Numer. Methods Engrg.* 122 (7) (2021) 1746–1769.
- [13] Z. Sun, Y. Gan, J. Tao, Z. Huang, X. Zhou, An improved quadrature scheme in B-spline material point method for large-deformation problem analysis, *Eng. Anal. Bound. Elem.* 138 (2022) 301–318.
- [14] A. de Vaucorbeil, V.P. Nguyen, C.R. Hutchinson, A total-Lagrangian material point method for solid mechanics problems involving large deformations, *Comput. Methods Appl. Mech. Engrg.* 360 (2020) 112783.
- [15] Y. Zhu, R. Bridson, Animating sand as a fluid, *ACM Trans. Graph.* 24 (3) (2005) 965–972.
- [16] J.E.G. P. C. Wallstedt, Improved velocity projection for the material point method, *CMES Comput. Model. Eng. Sci.* 19 (3) (2007) 223–232.
- [17] C. Jiang, C. Schroeder, A. Selle, J. Teran, A. Stomakhin, The affine particle-in-cell method, *ACM Trans. Graph.* 34 (4) (2015) 1–10.
- [18] G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, J. Teran, Drucker-prager elastoplasticity for sand animation, *ACM Trans. Graph.* 35 (4) (2016) 1–12.
- [19] 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.
- [20] C. Fu, Q. Guo, T. Gast, C. Jiang, J. Teran, A polynomial particle-in-cell method, *ACM Trans. Graph.* 36 (6) (2017) 1–12.
- [21] C. Jiang, C. Schroeder, J. Teran, An angular momentum conserving affine-particle-in-cell method, *J. Comput. Phys.* 338 (2017) 137–164.
- [22] S. Schulz, G. Sutmann, A consistent boundary method for the material point method-using image particles to reduce boundary artefacts, in: *PARTICLES VI: Proceedings of the VI International Conference on Particle-Based Methods: Fundamentals and Applications*, CIMNE, 2019, pp. 522–533.
- [23] M. Kohler, A. Stoecklin, A.M. Puzrin, A MPM framework for large-deformation seismic response analysis, *Can. Geotech. J.* 99 (999) (2022) 1–15.
- [24] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: *Proceedings of the 1968 23rd ACM National Conference*, 1968, pp. 517–524.
- [25] 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.
- [26] K. Nakamura, S. Matsumura, T. Mizutani, Particle-to-surface frictional contact algorithm for material point method using weighted least squares, *Comput. Geotech.* 134 (2021) 104069.
- [27] R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, *IBM J. Res. Dev.* 11 (2) (1967) 215–234.
- [28] S. Shao, E.Y. Lo, Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface, *Adv. Water Resour.* 26 (7) (2003) 787–800.
- [29] X. Zhuang, H. Zhu, C. Augarde, An improved meshless shepard and least squares method possessing the delta property and requiring no singular weight function, *Comput. Mech.* 53 (2) (2014) 343–357.
- [30] Y. Lu, T. Belytschko, L. Gu, A new implementation of the element free Galerkin method, *Comput. Methods Appl. Mech. Engrg.* 113 (3–4) (1994) 397–414.
- [31] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, *Internat. J. Numer. Methods Fluids* 20 (8–9) (1995) 1081–1106.
- [32] J. Bonet, S. Kulasegaram, Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations, *Internat. J. Numer. Methods Engrg.* 47 (6) (2000) 1189–1214.
- [33] J. Bonet, S. Kulasegaram, A simplified approach to enhance the performance of smooth particle hydrodynamics methods, *Appl. Math. Comput.* 126 (2–3) (2002) 133–155.
- [34] R. Vignjevic, J.R. Reveles, J. Campbell, *SPH in a Total Lagrangian Formalism*, Vol. 4, (3) CMC-Tech Science Press, 2006, p. 181.
- [35] Q.A. Tran, M. Berzins, W.T. Solowski, An improved moving least squares method for the material point method, 2019.
- [36] W.-F. Chen, *Limit Analysis and Soil Plasticity*, Elsevier, 2013.
- [37] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, *J. Comput. Phys.* 136 (1) (1997) 214–226.
- [38] Z.-P. Chen, X. Zhang, K.Y. Sze, L. Kan, X.-M. Qiu, vp material point method for weakly compressible problems, *Comput. & Fluids* 176 (2018) 170–181.