Journal of Computational Physics 413 (2020) 109461

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A Nitsche stabilized finite element method for embedded interfaces: Application to fluid-structure interaction and rigid-body contact

B. Liu *, D. Tan

Department of Mechanical Engineering, National University of Singapore, Singapore 119077, Singapore

ARTICLE INFO

Article history:
Received 24 July 2019
Received in revised form 4 April 2020
Accepted 6 April 2020
Available online 10 April 2020

Keywords:
Fluid-structure interaction
Nitsche's method
Ghost penalty method
Projection-based adaptive Gauss quadrature
Finite element

ABSTRACT

In this article, an accurate and robust numerical formulation is presented for the simulation of the fluid-structure interaction in incompressible fluid flow. The incompressible Navier-Stokes equation is discretized with a stabilized finite element framework on the fixed Eulerian grid. Both symmetric and non-symmetric Nitsche's methods are accessed and employed to weakly impose Dirichlet boundary condition along the interface embedded in the element together with the ghost penalty method stabilizing the solution jump across the element edges. An easy-to-implement and robust numerical integration scheme based on a projection approach is proposed. To the author's knowledge, so far, there is no application of a projection-based approach in the field of numerical integration to deal with discontinuities. Therefore, the results presented in this article is considered as a pioneered and novel projection-based approach in the field of numerical integration to deal with embedded discontinuous function. A second-order staggered-partitioned scheme is employed to weakly couple the fluid and structure solvers. A second-order accurate and unconditionally stable time integration scheme is implemented for simulations. Accurate numerical results are obtained in the numerical examples and validation cases, including vortex-induced vibration (VIV), rotation, freely fall and rigid-body contact.

1. Introduction

Fluid-structure interaction (FSI) is frequently encountered in many areas of mechanical, offshore, aerospace and civil engineering. The subtle and fascinating coupling phenomenon between the fluid and the elastically-mounted submerged bodies has attracted attention across diverse research communities. Over decades, a significant amount of efforts had been dedicated to the investigation of mutual responses from both fluid and structures under various flow conditions, e.g., angle of attack, shear flow and turbulent flow, and structure configurations, e.g., side-by-side, tandem, staggered and near-wall. However, large displacements and body-contact of structures often happened in the investigations, which cause a severe mesh distortion and pose difficulties to the numerical formulation. To address these difficulties, we propose an accurate and robust numerical formulation in this work.

* Corresponding author.
E-mail address: a0098961@u.nus.edu (B. Liu).

https://doi.org/10.1016/j.jcp.2020.109461
0021-9991/

Body-fitted formulation A well established technique to deal with FSI problem is arbitrary Lagrangian-Eulerian (ALE) formulation [1–5] using body-fitted meshes. The advantage of ALE-based approach is the position of the structure within fluid domain is timely known and an appropriately-managed fine mesh can be constructed along the submerged interface to accurately approximate the boundary layer. However, large structural deformations and topological changes can significantly distort the meshes and increase the element skewness, which eventually lead to a high computational cost in re-meshing.

Unfitted formulation To overcome these difficulties in ALE formulations, the idea of unfitted formulation was proposed to weakly impose Dirichlet boundary condition along the embedded interfaces over fixed background meshes. The unfitted formulation is an appealing approach in problems involving moving interfaces, e.g., fluid-structure interaction or free surface flows, as well as situations in which efforts are made to eliminate the generation of body-fitted meshes. For example, the immersed boundary (IB) method [6,7] and its variations, e.g., [8–12], are ones among the well-accepted approaches in *unfitted* formulation. Alternative well-established numerical methods under finite element framework are distributed Lagrange multiplier / fictitious domain (DLM/FD) method [13–15] and extended finite element method [16–18].

Embedded interface and cut cell stabilization The concept of distributing Lagrange points along the embedded interfaces was well established and implemented in the aforementioned numerical approaches. However, an appropriate choice of Lagrange multiplier basis space is critical to satisfy the Babŭska-Brezzi (BB) condition [19,20]. Recently, the Nitsche's method [21] gained attention among research communities, because of its advantages, e.g., variationally consistent and no increment in system size. It has been implemented to investigate a number of fluid-structure-interaction (FSI) problems, e.g., [22–28]. The element with an embedded interface is termed as a cut cell. As an interface cuts through it, the cut cell is demarcated into smaller cells, which is termed as integration cells. Some infinitesimal integration cells pose a challenge to the stability of numerical formulation and results into excessive system matrix condition number. Burman (2010) [29] proposed a stabilization technique, ghost penalty method, to alleviate the solution jump across the cut cell. In the present work, we implement the symmetric & non-symmetric Nitsche's method with ghost penalty terms to weakly impose the Dirichlet boundary along the embedded interface.

Integration of cut cell In all unfitted interface formulations, the numerical integration over the cut cell requires a special attention. In this work, we propose a projection-based adaptive Gauss quadrature (PAGQ) scheme. It can be used on either Tessellation or adaptive mesh refinement (AMR) technique. Tessellation [30,31] is a well-established method, in which the cut cell is triangulated or quadrangulated into smaller integration cells. Its advantage is the embedded discontinuity can be accurately captured by aligning with the edge of integration cells. However, Tessellation algorithm becomes complicated in three dimensional space. An alternative is AMR technique, e.g., quatree or smart octree [32,33], which overcomes the difficulty in aligning the integration cell with embedded discontinuities. The primary differences of PAGQ from Tessellation and AMR are (1) no change in finite element (FE) formulation and quadrature rule for the elements with/without embedded discontinuities, (2) the elemental matrices of subdivided integration cell are assembled via transformation in a quadratic form, a projection procedure. In particular we would like to address the following issues: (1) ease-of-implementation in FE formulations (simplicity), (2) capability to produce accurate numerical solution (accuracy) and (3) well-suited for FE formulation (variationally consistency)

The primary contributions of this work include:

- accurate and robust numerical formulation for large displacement and rigid body contact of FSI problem
- stabilized formulation for incompressible Navier-Stokes equations
- symmetric and non-symmetric Nitsche's method for embedded interface
- cut cell stabilization based on ghost penalty method
- projection-based adaptive Gauss quadrature using Tessellation or AMR
- second-order generalized-α time integration for fluid and structure
- second-order staggered-partitioned weakly-coupling FSI scheme

The manuscript is organized as follows. The governing equations and FSI schemes are listed in Sect. 2 at first. Following that, the complete variational formulation of our unfitted FSI solver is shown in Sect. 3. The proposed numerical integration scheme is discussed in Sect. 4. The error analysis is conducted in Sect. 5. Subsequently, numerical examples and validation results are presented in Sect. 6. Finally, we make the concluding remarks in Sect. 7.

2. Governing equations and boundary conditions

2.1. Incompressible Navier-Stokes equations

The implemented incompressible Navier-Stokes equation is shown in Eq. (1),

$$\rho^f \left(\frac{\partial \boldsymbol{u}^f}{\partial t} + \boldsymbol{u}^f \cdot \nabla \boldsymbol{u}^f \right) - \nabla \cdot \boldsymbol{\sigma}\{\boldsymbol{u}^f, p\} = \rho^f \boldsymbol{g}^f \quad \forall \boldsymbol{x} \in \Omega^f(t) \tag{1a}$$

$$\nabla \cdot \boldsymbol{u}^f = 0 \qquad \forall \boldsymbol{x} \in \Omega^f(t) \tag{1b}$$

$$\boldsymbol{u}^f = \tilde{\boldsymbol{u}}^f \qquad \forall \boldsymbol{x} \in \Gamma_D^f(t) \tag{1c}$$

$$\boldsymbol{\sigma}\{\boldsymbol{u}^f, p\} \cdot \boldsymbol{n}^f = \tilde{\boldsymbol{h}}^f \qquad \forall \boldsymbol{x} \in \Gamma_H^f(t) \tag{1d}$$

$$\boldsymbol{u}^f = \boldsymbol{u}_0^f \qquad \forall \boldsymbol{x} \in \Omega^f(0) \tag{1e}$$

where ρ^f, $\boldsymbol{u}^f$, $\boldsymbol{u}_0^f$, $\boldsymbol{g}^f$, $\tilde{\boldsymbol{u}}^f$, $\tilde{\boldsymbol{h}}^f$ and $\boldsymbol{n}^f$ respectively are the fluid density, fluid velocity vector, initial fluid velocity vector, fluid unit body force vector, prescribed fluid velocity, prescribed fluid traction and outward normal vector of fluid domain. The superscript (f) refers to the variables associated with fluid. The spatial domain, Dirichlet and Neumann boundaries respectively are denoted as Ω, Γ_D and Γ_H, where Γ_D and Γ_H are complementary subsets of Γ, $\Gamma = \Gamma_D \cup \Gamma_H$ and $\Gamma_D \cap \Gamma_H = \emptyset$. The Dirichlet and Neumann boundary conditions respectively are imposed along Γ_D and Γ_H as shown below.

$$\boldsymbol{u}^f = \tilde{\boldsymbol{u}}^f \quad \forall \boldsymbol{x} \in \Gamma_D^f(t) \tag{2a}$$

$$\boldsymbol{h}^f = \tilde{\boldsymbol{h}}^f \quad \forall \boldsymbol{x} \in \Gamma_H^f(t) \tag{2b}$$

where $\boldsymbol{h} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$ refers to the surface stresses. $\boldsymbol{\sigma}$ is the Cauchy stress tensor and defined as

$$\boldsymbol{\sigma}\{\boldsymbol{u}^f, p\} = -p\boldsymbol{I} + 2\mu D(\boldsymbol{u}^f) \tag{3a}$$

$$D(\boldsymbol{u}^f) = \frac{1}{2}\left[\nabla \boldsymbol{u}^f + (\nabla \boldsymbol{u}^f)'\right] \tag{3b}$$

The stress tensor is written as the summation of its isotropic and deviatoric tensor ($D(\boldsymbol{u}^f)$) parts, where the prime symbol represents a transpose operator. Here, p, μ and $\boldsymbol{I}$ refer to the fluid pressure, dynamic viscosity and identity matrix respectively.

2.2. Rigid-body dynamics

The equation governing the dynamics of a rigid body is simply implemented as Eq. (4).

$$\begin{aligned} &m^s \boldsymbol{a}^s + \boldsymbol{c}^s \boldsymbol{u}^s + \boldsymbol{k}^s \boldsymbol{d}^s = \boldsymbol{h}^s \quad \forall \boldsymbol{x} \in \Omega^s(t) \\ &\boldsymbol{a}^s = \frac{\partial^2 \boldsymbol{d}^s}{\partial t^2}; \quad \boldsymbol{u}^s = \frac{\partial \boldsymbol{d}^s}{\partial t} \\ &\boldsymbol{c}^s = 2\xi\sqrt{\boldsymbol{k}^s m^s}; \quad \boldsymbol{k}^s = 4\pi^2 \boldsymbol{f}_n^2 m^s \\ &U_r = U/(f_{ny} D); \quad m^s = m^*(0.25\pi D^2 L \rho^f) \end{aligned} \tag{4}$$

where ξ, m^*, $\boldsymbol{f}_n = [f_{nx}, f_{ny}]'$, D and L are the damping ratio, mass ratio, structural frequency vector, diameter of cylinder and spanwise length of cylinder respectively. The superscript (s) denotes the variables of solid. $\boldsymbol{c}^s$ and $\boldsymbol{k}^s$ are the damping and stiffness coefficients respectively. The reduced velocity of cylinder, U_r, is defined based on the structural frequency in the transverse direction, f_{ny}. In the present formulation, it is assumed that the structural frequencies in transverse and streamwise direction are identical, $f_{nx}/f_{ny} = 1.0$. $\boldsymbol{h}^s = [h_x^s, h_y^s]'$ represents the external force exerted on the cylinder surface.

2.3. Interface constraints and fluid-structure interaction

To couple the fluid and the structure dynamics, both the velocity and the traction constraints should be satisfied. The velocity constraint requires the fluid and the structure interfaces align with each other and move at the same velocity in Eq. (5a). On the other hand, the equilibrium of stresses, as shown in Eq. (5b), has to be enforced along the fluid-structure interface Γ^{fs} to satisfy the traction constraint, where $\boldsymbol{n}^f = -\boldsymbol{n}^s$. The superscript (fs) denotes the quantity of a fluid-structure interface.

$$\boldsymbol{u}^f = \boldsymbol{u}^s \quad \forall \boldsymbol{x} \in \Gamma^{fs}(t) \tag{5a}$$

$$\boldsymbol{\sigma}^f \cdot \boldsymbol{n}^f + \boldsymbol{\sigma}^s \cdot \boldsymbol{n}^s = 0 \quad \forall \boldsymbol{x} \in \Gamma^{fs}(t) \tag{5b}$$

$$\Longrightarrow \boldsymbol{h}^f + \boldsymbol{h}^s = 0 \quad \forall \boldsymbol{x} \in \Gamma^{fs}(t)$$

The fluid and structural governing equations can be coupled in either monolithic or staggered-partitioned scheme. In monolithic/fully-implicit scheme [34], the variables of both fluid and structure are solved indiscriminately and simultaneously. The monolithic formulation is robust, stable at relative large time steps, and its solution converges rapidly. Albeit monolithic schemes have the energy conservation property, their computational cost is high and typically require a significant recast in fluid and structural solvers. Moreover, the rigid-body impact and fluid flow take place at different time

scale. It imposes an inefficiency to the monolithic formulation. On the other hand, the staggered-partitioned scheme can be conveniently implemented to existing fluid and structural solvers, in which the fluid flow and structure dynamics are solved separately. The staggered-partitioned schemes can be further classified into strongly-coupled [35–37] or weakly-coupled schemes [38,39]. In this work, a staggered-partitioned, weakly-coupled and second-order accurate scheme in Dettmer & Perić (2013) [38] is implemented. The pseudocode of the implemented staggered scheme in each time step is briefly summarized below

In each time step (t):

1. predict the traction force on the structure: $\boldsymbol{h}_p^s(t+1) = 2\,\boldsymbol{h}^s(t) - \boldsymbol{h}^s(t-1)$
2. find the solution to the structure equation for the next time step, $\boldsymbol{u}^s(t+1)$ and $\boldsymbol{d}^s(t+1)$
3. update the location and velocity of the structure in fluid and find the solution to the Navier-Stokes equation for the next time step, $\boldsymbol{u}^f(t+1)$ and $\boldsymbol{h}^f(t+1)$
4. relax the traction force along interface $\Gamma^{fs}(t+1)$: $\boldsymbol{h}^s(t+1) = -\gamma^{fs}\boldsymbol{h}^f(t+1) + (1-\gamma^{fs})\boldsymbol{h}_p^s(t+1)$

where γ^{fs} is the relaxation parameter. The predicted traction force $\boldsymbol{h}_p^s(t+1)$ is a first-order approximation. For the detailed algorithm of this coupling scheme and the second-order predicted traction force, please refer to Dettmer & Perić (2013) [38].

2.4. Integration in time

The second-order accurate and unconditional stable generalized-α method [40] and [41] are implemented in time integration for both structural equation and Navier-Stokes equations. The detailed formulation for structural equation can be summarized as,

$$\mathbf{d}_{n+1}^s = \mathbf{d}_n^s + \Delta t\mathbf{u}_n^s + \Delta t^2\big(\big(\frac{1}{2}-\beta^s\big)\mathbf{a}_n^s + \beta^s\mathbf{a}_{n+1}^s\big) \tag{6a}$$

$$\mathbf{u}_{n+1}^s = \mathbf{u}_n^s + \Delta t\big((1-\gamma^s)\mathbf{a}_n^s + \gamma^s\mathbf{a}_{n+1}^s\big) \tag{6b}$$

$$\mathbf{d}_{n+\alpha_f^s}^s = (1-\alpha_f^s)\mathbf{d}_n^s + \alpha_f^s\mathbf{d}_{n+1}^s \tag{6c}$$

$$\mathbf{u}_{n+\alpha_f^s}^s = (1-\alpha_f^s)\mathbf{u}_n^s + \alpha_f^s\mathbf{u}_{n+1}^s \tag{6d}$$

$$\mathbf{a}_{n+\alpha_m^s}^s = (1-\alpha_m^s)\mathbf{a}_n^s + \alpha_m^s\mathbf{a}_{n+1}^s \tag{6e}$$

$$\mathbf{F}_{n+\alpha_f^s}^s = (1-\alpha_f^s)\mathbf{F}_n^s + \alpha_f^s\mathbf{F}_{n+1}^s \tag{6f}$$

where $\mathbf{d}_n^s$, $\mathbf{u}_n^s$ and $\mathbf{a}_n^s$ refer to the displacement, velocity and acceleration of cylinder at time $t=n$. The values of α_m^s, α_f^s, γ^s and β^s are defined by Chung & Hulbert (1993) [40] in Eq. (7) to ensure the characteristics of unconditionally stability and second-order accuracy.

$$\alpha_m^s := \frac{2\rho_\infty^s - 1}{\rho_\infty^s + 1}; \qquad \alpha_f^s := \frac{\rho_\infty^s}{\rho_\infty^s + 1} \tag{7a}$$

$$\gamma^s := 0.5 + \alpha_m^s - \alpha_f^s; \qquad \beta^s := 0.25(1 + \alpha_m^s - \alpha_f^s)^2 \tag{7b}$$

Similarly, the second-order accurate and unconditionally stable generalized-α method for Navier-Stokes equation is listed below,

$$\boldsymbol{u}_{n+1}^f = \boldsymbol{u}_n^f + \Delta t\big[(1-\gamma^f)\frac{\partial \boldsymbol{u}_n^f}{\partial t} + \gamma^f\frac{\partial \boldsymbol{u}_{n+1}^f}{\partial t}\big] \tag{8a}$$

$$\boldsymbol{u}_{n+\alpha_f^f}^f = (1-\alpha_f^f)\boldsymbol{u}_n^f + \alpha_f^f\boldsymbol{u}_{n+1}^f \tag{8b}$$

$$\frac{\partial \boldsymbol{u}^f}{\partial t}_{n+\alpha_m^f} = (1-\alpha_m^f)\frac{\partial \boldsymbol{u}_n^f}{\partial t} + \alpha_m^f\frac{\partial \boldsymbol{u}_{n+1}^f}{\partial t} \tag{8c}$$

$$\alpha_m^f := 0.5\frac{3-\rho_\infty^f}{1+\rho_\infty^f}; \quad \alpha_f^f := \frac{1}{1+\rho_\infty^f}; \quad \gamma^f := 0.5 + \alpha_m^f - \alpha_f^f \tag{8d}$$

Here $\rho_\infty^s \in [0,1]$ and $\rho_\infty^f \in [0,1]$ respectively are the spectral radius, which control the amount of numerical high-frequency damping in the temporal schemes. In this work. $\rho_\infty^s = \rho_\infty^f = 0.9$ are chosen for all numerical simulations.

3. Variational form of unfitted stabilized finite element formulation

The outline of the stabilized FE formulation of Navier-Stokes equation with embedded interfaces is summarized in Eq. (9), where $\mathcal{A}^G([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h])$ and $\mathcal{L}^G([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h])$ are the bilinear and linear forms derived from classical Galerkin method. $\mathcal{A}^S([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h])$ attributes to the Petrov-Galerkin formulation, which enables equal approximation function spaces between the velocity and the pressure. $\mathcal{A}^N([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h])$ is the terms of symmetric & non-symmetric Nitsche's methods for weakly imposing Dirichlet boundary condition along an embedded interface. In addition, $\mathcal{A}^{GP}([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h])$ is the ghost penalty terms to minimize the jump of solutions across the edges of cut cells.

$$\begin{aligned}\mathcal{A}^G([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) + \mathcal{A}^S([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) + \mathcal{A}^N([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) \\ + \mathcal{A}^{GP}([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) = \mathcal{L}^G([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h])\end{aligned} \tag{9}$$

The detailed formulations of Eq. (9) are presented in the subsequent sections.

3.1. Stabilized variational form of Navier-Stokes equation

The discrete variational form of the incompressible Navier-Stokes equation in Eq. (1) in a mixed finite element formulation using the classical Galerkin method, as shown in Eq. (10).

$$\begin{aligned}&\mathcal{A}^G([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) = \mathcal{L}^G([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) \\ \Longrightarrow &\int_{\Omega^f(t)} \left[\boldsymbol{v}^f \rho^f \left(\frac{\partial \boldsymbol{u}_h^f}{\partial t} + (\boldsymbol{u}_h^f \cdot \nabla)\boldsymbol{u}_h^f\right) - \nabla \boldsymbol{v}^f \boldsymbol{\sigma}\{\boldsymbol{u}_h^f, p_h\} - \boldsymbol{v}^f \rho^f \boldsymbol{g}^f\right] d\Omega \\ &- \int_{\Gamma_H^f(t)} \boldsymbol{v}^f \tilde{\boldsymbol{h}}_h^f d\Gamma - \int_{\Gamma_{H(out)}^f(t)} \boldsymbol{v}^f (-\mu(\nabla \boldsymbol{u}_h^f)') d\Gamma \\ &+ \int_{\Omega^f(t)} q[\nabla \cdot \boldsymbol{u}_h^f] d\Omega = \boldsymbol{0} \qquad \forall [\boldsymbol{v}^f, q] \in \hat{\boldsymbol{\mathcal{V}}}_h \times \hat{\mathcal{Q}}_h \subset \hat{\boldsymbol{\mathcal{V}}} \times \hat{\mathcal{Q}}\end{aligned} \tag{10}$$

where $[\boldsymbol{v}^f, q]'$ is the vector of test functions for the velocity and pressure of fluid. The boundary integral term $\int \boldsymbol{v}^f (-\mu(\nabla \boldsymbol{u}_h^f)') d\Gamma$ is a correction term proposed by Heywood et al. (1996) [42] for "do-nothing" outflow boundary condition along the outflow boundary Γ_{out}^f. The vector-valued trial and test function spaces $\boldsymbol{\mathcal{V}}$ and $\hat{\boldsymbol{\mathcal{V}}}$ of velocity are defined as

$$\boldsymbol{\mathcal{V}} = \{\boldsymbol{v}^f \in \boldsymbol{\mathcal{H}}^1(\Omega^f(t)) : \boldsymbol{v}^f = \tilde{\boldsymbol{v}}^f \quad \forall \boldsymbol{x} \in \Gamma_D^f(t)\} \tag{11a}$$

$$\hat{\boldsymbol{\mathcal{V}}} = \{\boldsymbol{v}^f \in \boldsymbol{\mathcal{H}}^1(\Omega^f(t)) : \boldsymbol{v}^f = \boldsymbol{0} \quad \forall \boldsymbol{x} \in \Gamma_D^f(t)\} \tag{11b}$$

On the other hand, the scalar-valued trial and test function spaces $\mathcal{Q}$ and $\hat{\mathcal{Q}}$ of pressure are defined as

$$\mathcal{Q} = \{q \in \mathcal{H}^1(\Omega^f(t)) : q = \tilde{p} \quad \forall \boldsymbol{x} \in \Gamma_D^f(t)\} \tag{12a}$$

$$\hat{\mathcal{Q}} = \{q \in \mathcal{H}^1(\Omega^f(t)) : q = 0 \quad \forall \boldsymbol{x} \in \Gamma_D^f(t)\} \tag{12b}$$

where $\boldsymbol{\mathcal{H}}^1$ refers to the Sobolev space, in which $[(\boldsymbol{v}^f)^2, q^2]$ and $[|\nabla \boldsymbol{v}^f|^2, |\nabla q|^2]$ have finite integrals within $\Omega^f(t)$ and allows discontinuous derivatives. Their corresponding discrete function spaces are denoted with subscript (h), e.g., $\hat{\mathcal{Q}}_h$. A residual-based stabilization technique, Petrov-Galerkin method [43–46], in Eq. (13) is implemented to minimize the residual of equation system in a weak/integral sense. Here $\boldsymbol{G}$ and C_I are respectively element cotravariant metric tensor and a positive constant independent upon mesh size [47].

$$\begin{aligned}\mathcal{A}^S([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) = &\sum_{e=1}^{n_{el}} \int_{\Omega^f(t)} \tau_m \left[\rho^f (\boldsymbol{u}_h^f \cdot \nabla)\boldsymbol{v}^f - \mu \nabla^2 \boldsymbol{v}^f + \nabla q\right] \\ &\left[\rho^f \left(\frac{\partial \boldsymbol{u}_h^f}{\partial t} + (\boldsymbol{u}_h^f \cdot \nabla)\boldsymbol{u}_h^f - \boldsymbol{g}^f\right) - \mu \nabla^2 \boldsymbol{u}_h^f + \nabla p_h\right] d\Omega \\ + &\sum_{e=1}^{n_{el}} \int_{\Omega^f(t)} \tau_c \rho^f (\nabla \cdot \boldsymbol{v}^f)(\nabla \cdot \boldsymbol{u}_h^f) d\Omega \qquad \forall [\boldsymbol{v}^f, q] \in \hat{\boldsymbol{\mathcal{V}}}_h \times \hat{\mathcal{Q}}_h \subset \hat{\boldsymbol{\mathcal{V}}} \times \hat{\mathcal{Q}}\end{aligned} \tag{13}$$

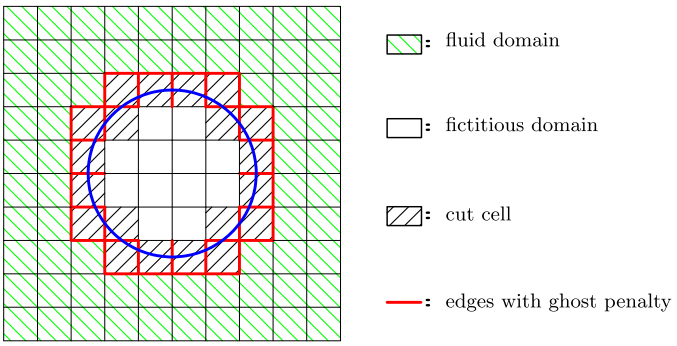


Fig. 1. Schematic diagrams of embedded interface with ghost penalty terms. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\tau_m = \Big[\big(\frac{2\rho^f}{\Delta t}\big)^2 + (\rho^f)^2 \boldsymbol{u}_h^f \cdot \boldsymbol{G}\boldsymbol{u}_h^f + C_I(\mu)^2 \boldsymbol{G} : \boldsymbol{G}\Big]^{-0.5}$$

$$\tau_c = (tr(\boldsymbol{G})\tau_m)^{-1}; \quad \boldsymbol{G} = \frac{\partial \boldsymbol{\xi}'}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}$$

3.2. Nitsche's method

The symmetric & (penalty-free) non-symmetric Nitsche's methods are shown in Eq. (14).

$$\begin{aligned} \mathcal{A}^N([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) &= \gamma_1 \int_{\Gamma^{fs}(t)} \boldsymbol{v}^f (\boldsymbol{u}_h^f - \tilde{\boldsymbol{u}}_h^f) d\Gamma \\ - \int_{\Gamma^{fs}(t)} \boldsymbol{v}^f (\boldsymbol{\sigma}\{\boldsymbol{u}_h^f, p_h\} \cdot \boldsymbol{n}^f) d\Gamma &- \gamma_2 \int_{\Gamma^{fs}(t)} (\boldsymbol{\sigma}\{\boldsymbol{v}^f, q\} \cdot \boldsymbol{n}^f)(\boldsymbol{u}_h^f - \tilde{\boldsymbol{u}}_h^f) d\Gamma \\ \forall [\boldsymbol{v}^f, q] &\in \hat{\boldsymbol{\mathcal{V}}}_h \times \hat{\mathcal{Q}}_h \subset \hat{\boldsymbol{\mathcal{V}}} \times \hat{\mathcal{Q}} \end{aligned} \tag{14}$$

Either the symmetric-variant $\gamma_2 = 1$ or non-symmetric-variant $\gamma_2 = -1$ can be implemented. The penalty term is chosen within an appropriate range $\gamma_1 \in [\mu\frac{10^2}{L}, \mu\frac{10^3}{L}]$ for symmetric-variant based on Benk (2012) [48], where L is the characteristic element length, or $\gamma_1 = 0.0$ for non-symmetric-variant based on Burman (2012) [49]. As long as the numerical solution convergences to the unique solution $\boldsymbol{u}_h^f \approx \tilde{\boldsymbol{u}}_h^f$, the first and third penalty terms vanish.

3.3. Ghost penalty method

The cut cell is demarcated by an embedded interface, e.g., the blue circle in Fig. 1, into a fluid domain and a fictitious domain respectively. The embedded interface is represented by a level-set function $||\boldsymbol{x} - \boldsymbol{x}_c||_{\mathcal{L}_2} - 0.5D$, where $\boldsymbol{x}_c$ and D respectively denote the coordinates of the center of the cylinder and the diameter of a cylinder. The subscript $\mathcal{L}_2$ denotes the Euclidean 2 norm. If the solution to the level-set function is larger than zero, it indicates the fluid domain and vice versa. If the physical part is very small, some basis functions have little support inside the physical domain. It leads to the large system matrix condition numbers. The residual-based ghost penalty method [29] is implemented along the edges of cut cells, the red edges in Fig. 1, to alleviate the jumps of solutions. A comprehensive study of the performance of ghost penalty terms was reported by Dettmer et al., 2016 [24]. The specific terms are listed in Eq. (15).

$$\begin{aligned} \mathcal{A}^{GP}([\boldsymbol{v}^f, q], [\boldsymbol{u}_h^f, p_h]) &= \beta_{gp}^u \mu G_1(\boldsymbol{v}^f, \boldsymbol{u}_h^f) + \beta_{gp}^p \mu^{-1} g_3(q, p_h) \\ \forall [\boldsymbol{v}^f, q] &\in \hat{\boldsymbol{\mathcal{V}}}_h \times \hat{\mathcal{Q}}_h \subset \hat{\boldsymbol{\mathcal{V}}} \times \hat{\mathcal{Q}} \\ g_\phi(q, p_h) &= \sum_{k=1}^{e} l_k^{2(\alpha-1)+\phi} \int_{\Gamma^f(t)} [[\frac{\partial^\alpha q}{\partial \boldsymbol{n}^{f(\alpha)}}]][[\frac{\partial^\alpha p_h}{\partial \boldsymbol{n}^{f(\alpha)}}]] dl_k \\ G_\phi(\boldsymbol{v}^f, \boldsymbol{u}_h^f) &= \sum_{k=1}^{e} \sum_{i=1}^{d} l_k^{2(\alpha-1)+\phi} \int_{\Gamma^f(t)} [[\frac{\partial^\alpha v_{(i)}^f}{\partial \boldsymbol{n}^{f(\alpha)}}]][[\frac{\partial^\alpha u_{h(i)}^f}{\partial \boldsymbol{n}^{f(\alpha)}}]] dl_k \end{aligned} \tag{15}$$

where the penalty parameters is chosen as $\beta_{gp}^{u} = \beta_{gp}^{p} = 0.02$ [24] for the numerical examples in this work. The superscripts (u) and (p) respectively refer to velocity and pressure. The subscript (gp) shows that these terms attribute to ghost penalty terms. e and d respectively are the number of edges of cut cell imposed with ghost penalty terms and the dimension of problem. α, ϕ and l are order of derivative, the notation parameter and element characteristic length respectively. $[[\cdot]]$ denotes a jump of solution across the element edge.

Therefore, the overall numerical formulation of Navier-Stokes equations with embedded interface and the ghost-penalty stabilization is summarized as,

$$
\begin{aligned}
&\int_{\Omega^f(t)} \Big[\boldsymbol{v}^f \rho^f \Big(\frac{\partial \boldsymbol{u}_h^f}{\partial t} + (\boldsymbol{u}_h^f \cdot \nabla)\boldsymbol{u}_h^f\Big) - \nabla \boldsymbol{v}^f \boldsymbol{\sigma}\{\boldsymbol{u}_h^f, p_h\} - \boldsymbol{v}^f \rho^f \boldsymbol{g}^f\Big] d\Omega \\
&\quad - \int_{\Gamma^f_{H(out)}(t)} \boldsymbol{v}^f(-\mu(\nabla \boldsymbol{u}_h^f)')d\Gamma + \int_{\Omega^f(t)} q[\nabla \cdot \boldsymbol{u}_h^f] d\Omega \\
&\quad + \sum_{e=1}^{n_{el}} \int_{\Omega^f(t)} \tau_m \Big[\rho^f (\boldsymbol{u}_h^f \cdot \nabla)\boldsymbol{v}^f - \mu \nabla^2 \boldsymbol{v}^f + \nabla q\Big] \\
&\qquad \Big[\rho^f (\frac{\partial \boldsymbol{u}_h^f}{\partial t} + (\boldsymbol{u}_h^f \cdot \nabla)\boldsymbol{u}_h^f - \boldsymbol{g}^f) - \mu \nabla^2 \boldsymbol{u}_h^f + \nabla p_h\Big] d\Omega \\
&\quad + \sum_{e=1}^{n_{el}} \int_{\Omega^f(t)} \tau_c \rho^f (\nabla \cdot \boldsymbol{v}^f)(\nabla \cdot \boldsymbol{u}_h^f) d\Omega + \gamma_1 \int_{\Gamma^{fs}(t)} \boldsymbol{v}^f (\boldsymbol{u}_h^f - \tilde{\boldsymbol{u}}_h^f) d\Gamma \\
&- \int_{\Gamma^{fs}(t)} \boldsymbol{v}^f (\boldsymbol{\sigma}\{\boldsymbol{u}_h^f, p_h\} \cdot \boldsymbol{n}^f) d\Gamma - \gamma_2 \int_{\Gamma^{fs}(t)} (\boldsymbol{\sigma}\{\boldsymbol{v}^f, q\} \cdot \boldsymbol{n}^f)(\boldsymbol{u}_h^f - \tilde{\boldsymbol{u}}_h^f) d\Gamma \\
&\qquad + \sum_{k=1}^{e} l_k^{2(\alpha-1)+\phi} \int_{\Gamma^f(t)} [[\frac{\partial^\alpha q}{\partial \boldsymbol{n}^{f(\alpha)}}]][[\frac{\partial^\alpha p_h}{\partial \boldsymbol{n}^{f(\alpha)}}]] dl_k \\
&\qquad + \sum_{k=1}^{e} \sum_{i=1}^{d} l_k^{2(\alpha-1)+\phi} \int_{\Gamma^f(t)} [[\frac{\partial^\alpha v_{(i)}^f}{\partial \boldsymbol{n}^{f(\alpha)}}]][[\frac{\partial^\alpha u_{h(i)}^f}{\partial \boldsymbol{n}^{f(\alpha)}}]] dl_k \\
&\qquad = \int_{\Gamma_H^f(t)} \boldsymbol{v}^f \tilde{\boldsymbol{h}}_h^f d\Gamma \qquad \forall [\boldsymbol{v}^f, q] \in \hat{\boldsymbol{\mathcal{V}}}_h \times \hat{\mathcal{Q}}_h \subset \hat{\boldsymbol{\mathcal{V}}} \times \hat{\mathcal{Q}}
\end{aligned} \tag{16}
$$

4. Numerical integration (PAGQ)

In the proposed numerical integration, the cut cell is adaptively subdivided into smaller integration cells, based on the projection approach in a quadratic form transformation [50,51]. The technique of projection in a quadratic form is one of the cornerstones in applied mathematics and widely-accepted in many scientific domains, e.g., machine learning [52–55], reduced order model [56–60], eigenvalue analysis [61] and finite element method [62,63] to name a few. Hence this proposed numerical scheme is termed as a projection-based adaptive Gauss quadrature, PAGQ, in this article. In PAGQ, the FE formulation and Gauss quadrature rule in sub-divided cells are identical to the elements without an embedded interface. Whereas, the assembly procedure is based on the projection approach.

The detailed algorithm of PAGQ is demonstrated using a general FE formulation subsequently. Assuming the domain is discretized by the structured quadrilateral elements in Fig. 2, the corresponding discrete variational form of a general partial differential equation (PDE) of a single-variable problem can be derived as

$$
\begin{aligned}
&\mathcal{A}(\boldsymbol{v}, \boldsymbol{d}_h) = \mathcal{L}(\boldsymbol{v}) \\
&\mathcal{A}(\boldsymbol{v}, \boldsymbol{d}_h) = \int_{\Omega} [\boldsymbol{L}\boldsymbol{v}]' \, \boldsymbol{D} \, [\boldsymbol{L}\boldsymbol{d}_h] d\Omega \\
&\mathcal{L}(\boldsymbol{v}) = \int_{\Omega} [\boldsymbol{v}'\boldsymbol{b}] d\Omega + \int_{\Gamma_H} [\boldsymbol{v}'\tilde{\boldsymbol{h}}_h] d\Gamma
\end{aligned} \tag{17}
$$

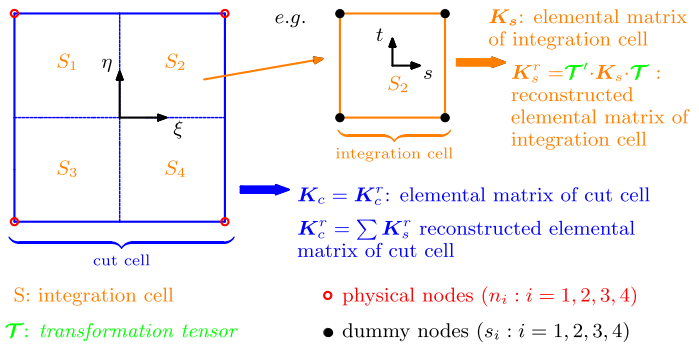


Fig. 2. Illustration of the general concept of PAGQ in a bilinear quadrilateral element.

where $\mathcal{A}(\boldsymbol{v}, \boldsymbol{d}_h)$ and $\mathcal{L}(\boldsymbol{v})$ are respectively bilinear and linear functionals. In Eq. (17), $\boldsymbol{v}$, $\boldsymbol{d}_h$, $\boldsymbol{b}$, $\tilde{\boldsymbol{h}}_h$ and $\boldsymbol{D}$ are the test function, the nodal value, the volume source, the prescribed traction vector and the coefficient matrix respectively. $\boldsymbol{L}$ is a differential operator. The strain matrix is $\boldsymbol{B} = \boldsymbol{LN}$, in which $\boldsymbol{N}$ is a trial function (shape function) vector. In Bubnov-Galerkin method, the test function vector is chosen as trial function vector, $\boldsymbol{v} = \boldsymbol{N}$. Hence the elemental stiffness matrix and force matrix of the cut cell becomes,

$$\boldsymbol{K}_c = \int_\Omega [\boldsymbol{B}_c' \, \boldsymbol{D} \, \boldsymbol{B}_c] d\Omega \tag{18a}$$

$$\boldsymbol{F}_c = \int_\Omega [\boldsymbol{N}_c' \, \boldsymbol{b}] d\Omega + \int_{\Gamma_H} [\boldsymbol{N}_c' \, \tilde{\boldsymbol{h}}_h] d\Gamma \tag{18b}$$

where the subscript (c) refers to the matrices and vectors of a cut cell. The standard Gauss quadrature rule is implemented in each integration cell with respect to its dummy nodes, the corners of each quadrilateral integration cell, as shown in the detailed view of Fig. 2.

Similar to the cut cell, the stiffness matrix and force vector of an integration cell can be formulated as

$$\boldsymbol{K}_s = \int_\Omega [\boldsymbol{B}_s' \, \boldsymbol{D} \, \boldsymbol{B}_s] d\Omega \tag{19a}$$

$$\boldsymbol{F}_s = \int_\Omega [\boldsymbol{N}_s' \, \boldsymbol{b}] d\Omega + \int_{\Gamma_H} [\boldsymbol{N}_s' \, \tilde{\boldsymbol{h}}_h] d\Gamma \tag{19b}$$

where the subscript (s) refers to the matrices of an integration cell. The second term on the right-hand side of Eq. (19b) is a line integral. As demonstrated in Fig. 2, the matrices of integration cells are mapped via a transformation tensor $\mathcal{T}$ and assembled to form the reconstructed elemental matrices of a cut cell, e.g., $\boldsymbol{K}_c^r$, where the superscript (r) denotes a reconstructed matrix.

Two equivalent computational sequences, Algorithm 1 and 2, are applicable, where the superscript (as) denotes an assembled matrix based on the assembly procedure in the standard FE formulation.

Algorithm 1 Transformation-assembly.

1: sub-divide cut cell into integration cells
2: **for** i=no. of integration cell **do**
3: construct $\boldsymbol{T}$ for ith integration cell, Eq. (21a)
4: Gaussian quadrature for ith integration cell
5: $\boldsymbol{K}_s^r = \boldsymbol{T}' \cdot \boldsymbol{K}_s \cdot \boldsymbol{T}$, Eq. (23a) and (23b)
6: sum up as $\boldsymbol{K}_c^r += \boldsymbol{K}_s^r$, Eq. (24a) and (24b)
7: **end for**

4.1. Transformation tensor

A detailed construction procedure of $\mathcal{T}$ is demonstrated in Fig. 3. For instance, one would like to approximate the scalar values V at Gauss points inside an integration cell, the solid-circles in Fig. 3, from the dummy nodes, s_j. Simultaneously, the values on dummy nodes can be approximated from the physical nodes, n_i. As a result, the scalar value at a Gauss point $\boldsymbol{x}_k$, $V(\boldsymbol{x}_k)$, can be approximated from the physical nodes as follows.

Algorithm 2 Assembly-transformation.

1: sub-divide cut cell into integration cells
2: construct $\boldsymbol{T}^{as}$ (rectangular tensor) for all integration cells, Eq. (21a)
3: **for** i=no. of integration cell **do**
4: Gaussian quadrature for ith integration cell
5: assembly as $\boldsymbol{K}_s^{as}$
6: **end for**
7: $\boldsymbol{K}_c^r = \boldsymbol{T}^{as'} \cdot \boldsymbol{K}_s^{as} \cdot \boldsymbol{T}^{as}$, Eq. (25a) and (25b)

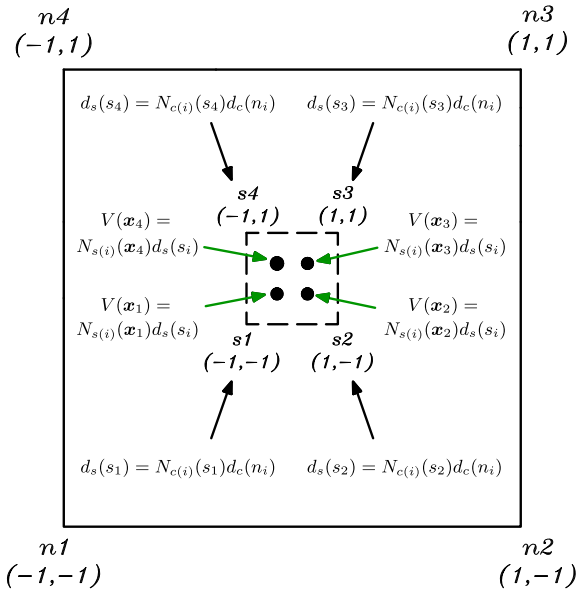


Fig. 3. Mapping procedure between integration cell and cut cell.

$$\begin{aligned} V(\boldsymbol{x}_k) &= N_{s(j)}(\boldsymbol{x}_k)N_{c(i)}(s_j)d_c(n_i) \\ &= \tilde{N}_i(\boldsymbol{x}_k)d_c(n_i) \end{aligned} \tag{20}$$

where $\tilde{\boldsymbol{N}}(\boldsymbol{x})$ is a composed trial function vector and the Einstein summation convention is used. The $\boldsymbol{N}_s$ and $\boldsymbol{N}_c$ respectively are the shape function matrices associated with the dummy nodes and physical nodes. Hence a transformation tensor $\mathcal{T}$ can be defined in Eq. (21a). The column j of $\boldsymbol{T}$ refers to the weights from a physical node n_j to the dummy nodes s_k of a cut cell. Therefore, $\tilde{\boldsymbol{N}}$ can be accordingly re-casted in Eq. (21b).

$$\mathcal{T}_{kj} = N_{c(j)}(s_k) \tag{21a}$$

$$\tilde{N}_{ij} = N_{s(k)}(\boldsymbol{x}_i)\mathcal{T}_{kj} \tag{21b}$$

Hence the corresponding matrix form of $\boldsymbol{T}$ can be written as

$$\begin{aligned} \mathcal{T} &= [\boldsymbol{N}'_{c(\boldsymbol{n})}(s_1), \boldsymbol{N}'_{c(\boldsymbol{n})}(s_2), ..., \boldsymbol{N}'_{c(\boldsymbol{n})}(s_i)]' \\ &= \begin{bmatrix} N_{c(n_1)}(s_1) & N_{c(n_2)}(s_1) & \dots & N_{c(n_j)}(s_1) \\ N_{c(n_1)}(s_2) & N_{c(n_2)}(s_2) & \dots & N_{c(n_j)}(s_2) \\ \vdots & \vdots & \ddots & \vdots \\ N_{c(n_1)}(s_i) & N_{c(n_2)}(s_i) & \dots & N_{c(n_j)}(s_i) \end{bmatrix} \end{aligned} \tag{22}$$

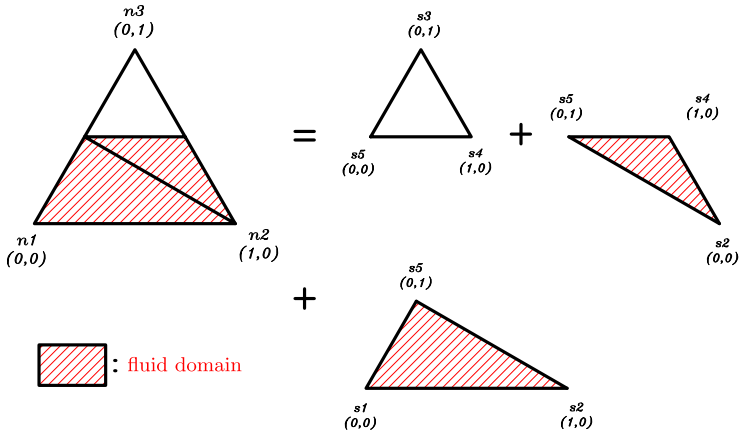
Subsequently the $\boldsymbol{T}$ is used to map the elemental matrices between the bases of the integration cells and their cut cell, as shown in Eq. (23).

$$K^r_{s(ij)} = \mathcal{T}_{ki}K_{s(kl)}\mathcal{T}_{lj} \tag{23a}$$

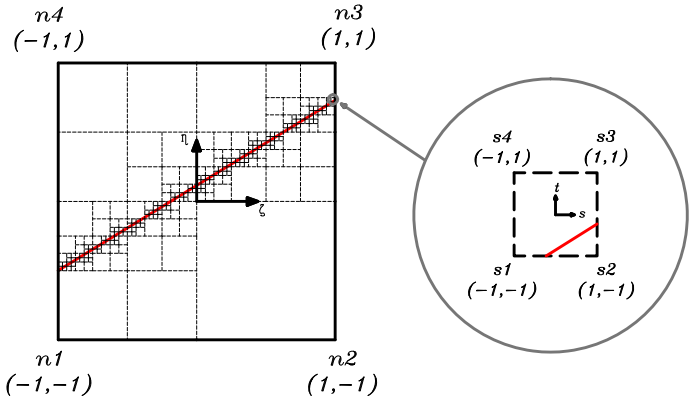
$$F^r_{s(i)} = \mathcal{T}_{ki}F_{s(k)} \tag{23b}$$

where $K^r_{s(ij)}$ and $F^r_{s(i)}$ are the reconstructed elemental matrices of an integration cell in the component form.

Subsequently, the reconstructed elemental matrices of a cut cell is simply formed by a summation operation, as shown in Eq. (24).



(a)



(b)

Fig. 4. Implementation of PAGQ: (a) Tessellation; (b) AMR.

$$K^r_{c(ij)} = \sum_{n=1}^{en} K^r_{s(ij)}(n) = \sum_{n=1}^{en} \mathcal{T}_{ki}(n) K_{s(kl)}(n) \mathcal{T}_{lj}(n) \tag{24a}$$

$$F^r_{c(i)} = \sum_{n=1}^{en} F^r_{s(i)}(n) = \sum_{n=1}^{en} \mathcal{T}_{ki}(n) F_{s(k)}(n) \tag{24b}$$

where *en* parameter is the total number of integration cells in a cut cell. The above demonstrates the computational procedures in Algorithm 1. The assembly procedure can be performed before transformation operation, as shown in Algorithm 2. The corresponding transformation operation in Algorithm 2 is shown in Eq. (25).

$$K^r_{c(ij)} = \mathcal{T}^{as}_{ki} K^{as}_{s(kl)} \mathcal{T}^{as}_{lj} \tag{25a}$$

$$F^r_{c(i)} = \mathcal{T}^{as}_{ki} F^{as}_{s(k)} \tag{25b}$$

where $\mathcal{T}^{as}$ is a rectangular transformation tensor in this case, in which the number of rows and columns are respectively the total number of dummy nodes and physical nodes. Therefore, PAGQ can be implemented via either Tessellation or AMR, as demonstrated in Fig. 4a and Fig. 4b respectively. In terms of Tessellation in Fig. 4a, the embedded discontinuity is precisely aligned with the edge of integration cells and the embedded interface is computed in line integrals. For the AMR in Fig. 4b, the integration cells are continuously refined around the embedded interface. In such a way, a large portion of the cut cell is precisely integrated without embedded discontinuities. In the infinitesimal integration cells cut by the interface, a line integral is reformulated in Eq. (26) based on Rätz & Voigt, A. (2006) [64], Li et al. (2009) [65] and Nguyen et al. (2017) [66].

$$\int_{\Omega} f(\boldsymbol{x}) d\Omega = \int_{\mathcal{K}} f(\boldsymbol{x}) H d\Omega \approx \int_{\mathcal{K}} f(\boldsymbol{x}) \phi d\Omega \tag{26a}$$

$$\int_{\Gamma} f(\boldsymbol{x})d\Gamma = \int_{\mathcal{K}} f(\boldsymbol{x})\delta_{\Gamma}d\Omega \approx \int_{\mathcal{K}} f(\boldsymbol{x})|\nabla\phi|d\Omega \tag{26b}$$

$$\int_{\Gamma} f(\boldsymbol{x})\cdot\boldsymbol{n}d\Gamma = \int_{\mathcal{K}} f(\boldsymbol{x})\cdot\boldsymbol{n}\delta_{\Gamma}d\Omega \approx -\int_{\mathcal{K}} f(\boldsymbol{x})\cdot\nabla\phi d\Omega \tag{26c}$$

where $f(\boldsymbol{x})$, δ_{Γ} and ϕ are respectively a general scalar function, a Dirac delta function and a phase-field function. For the detailed discussion of this reformulation, please refer to Nguyen et al. (2017) [66].

4.2. Characteristics of PAGQ

In this section, the characteristics of PAGQ are discussed: (1) the partition of unity property, (2) the adaptive Gauss quadrature (GQ), (3) the projection in a quadratic form and (4) the order reduction approximation.

Partition of unity property: Partition of unity is one of the fundamental properties in FE approximation. It can be simply proven the composed trial function $\tilde{\boldsymbol{N}}$ in Sect. 4.1 satisfies the partition of unity property in Eq. (27).

$$\sum_{i}\tilde{N}_i(\alpha) = \sum_{i,j} N_j(\alpha)N_i(\beta_j) = \sum_{i} N_i(\alpha) = 1.0 \tag{27}$$

Adaptive Gauss quadrature: The mathematical derivation in Eq. (28) shows the reconstructed elemental matrices, e.g., $\boldsymbol{K}_c^r$, exactly recovers the elemental matrices obtained by standard Gauss quadrature of a cut cell, e.g., $\boldsymbol{K}_c$.

$$\begin{aligned}
\boldsymbol{K}_c^r &= \sum_{n=1}^{en}\boldsymbol{\mathcal{T}}'(n)\cdot\boldsymbol{K}_s(n)\cdot\boldsymbol{\mathcal{T}}(n)\\
&= \sum_{n=1}^{en}\boldsymbol{\mathcal{T}}'(n)\cdot\int_{\Omega}[\boldsymbol{B}_s'(n)\ \boldsymbol{D}\ \boldsymbol{B}_s(n)]d\Omega\cdot\boldsymbol{\mathcal{T}}(n)\\
&= \sum_{n=1}^{en}\int_{\Omega}[\tilde{\boldsymbol{N}}'(n)\boldsymbol{L}'\ \boldsymbol{D}\ \boldsymbol{L}\tilde{\boldsymbol{N}}(n)]d\Omega\\
&= \sum_{n=1}^{en}\sum_{g=1}^{gp}[\tilde{\boldsymbol{B}}'(n,g)\ \boldsymbol{D}\ \tilde{\boldsymbol{B}}(n,g)|\boldsymbol{J}(n,g)|W(g)]\\
&= \sum_{k=1}^{en\cdot gp}[\tilde{\boldsymbol{B}}'(k)\ \boldsymbol{D}\ \tilde{\boldsymbol{B}}(k)|\boldsymbol{J}(k)|W(k)]\\
&= \int_{\Omega}\tilde{\boldsymbol{B}}'\ \boldsymbol{D}\ \tilde{\boldsymbol{B}}d\Omega\\
&= \boldsymbol{K}_c
\end{aligned} \tag{28}$$

$\tilde{\boldsymbol{B}}(k) = \boldsymbol{L}\tilde{\boldsymbol{N}}(k)$, $\boldsymbol{J}(k)$ and $W(k)$ are the composed strain vector, the Jacobian matrix and the Gauss integration weights respectively. The value of $en\cdot gp$ is the total number of the Gauss integration points within a cut cell. en and gp respectively are the number of integration cells and the number of Gauss points within an integration cell.

It should be noted that $\boldsymbol{B}_s$ is computed over a continuous function space in an integration cell. On the other hand, its solutions are approximated from the nodes of its parent element, the cut cell, through the quadratic transformation. It guarantees an accurate approximation of the gradients within the integration cell. Therefore, PAGQ can be considered as an adaptive variation of the standard Gauss quadrature numerical integration.

Quadratic form: It is known that a matrix, e.g., $\boldsymbol{K}_s$ in Eq. (29), can be mapped back to its own basis function space using its unit basis vectors, e.g., $\boldsymbol{e}_1 = [1,0,0]'$ in Cartesian coordinate system of $\mathbb{R}^3$.

$$K_{s(i,j)} = \boldsymbol{e}_i'\cdot\boldsymbol{K}_s\cdot\boldsymbol{e}_j \tag{29}$$

Similarly, it can be projected to other basis function spaces, provided an appropriate transformation tensor is defined. In PAGQ, $\boldsymbol{\mathcal{T}}$ is constructed based on its trial functions in Eq. (21a), such that the nodal values and their residuals are re-projected in the basis function space of the cut cell. Because the FE formulation results into a symmetric matrix system,[1]

[1] In Navier-Stokes equation, the resultant stiffness matrix $\boldsymbol{K}$ can be subdivided into symmetric matrix blocks.

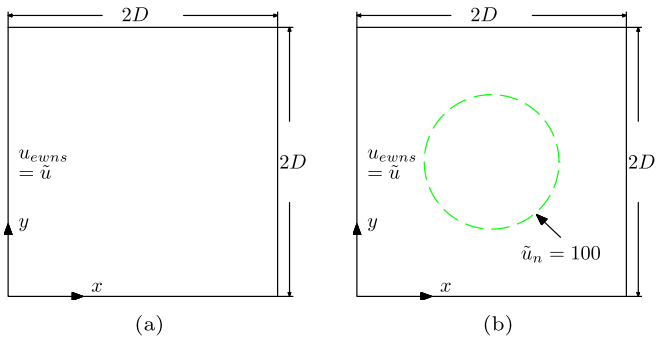


Fig. 5. Schematic diagram of convergence analysis: (a) Uniformly refinement by PAGQ; (b) Embedded discontinuity along the green circle. Non-harmonic Dirichlet boundary condition is applied along the east, west, north and south boundaries u_{ewns}.

which can always be transformed into a quadratic form, the proposed PAGQ is mathematically-robust and consistent with FE formulation.

Order reduction: PAGQ can be considered as an order reduction technique too, which projects a high dimensional matrix to a lower dimensional one with an appropriate transformation tensor $\mathcal{T}$. In this lower dimensional space, the basis vector is the trial function in FE formulation. In Algorithm 2, the matrices of integration cells are assembled at first. The resultant $\boldsymbol{K}_s^{as}$ represents the approximation of solutions around this portion of embedded interface with enough large number of degree-of-freedoms (DoFs). In this case, $\mathcal{T}^{as}$ is defined as a rectangular transformation matrix.

5. Convergence analysis

The convergence analysis of the proposed numerical integration scheme PAGQ is presented in this section. The Poisson's equation in Eq. (30) is employed and discretized over a two-dimensional square domain Ω with quadrilateral elements. The non-harmonic Dirichlet boundary condition is imposed along the computational boundary Γ.

$$\begin{aligned} -\Delta u &= f \quad \forall \boldsymbol{x} \in \Omega \\ u &= \tilde{u} \quad \forall \boldsymbol{x} \in \Gamma \end{aligned} \tag{30}$$

The discrete variational form of Eq. (30) can be written as

$$\int_{\Omega} \nabla v \cdot \nabla u_h \, d\Omega = \int_{\Omega} v f \, d\Omega + \int_{\Gamma} v \nabla u_h \cdot \boldsymbol{n} \, d\Gamma \quad \forall v \in \mathcal{Q}_h \subset \mathcal{Q} \tag{31}$$

where v is the test function. The convergence analysis is conducted in a two-dimensional square domain of size $\boldsymbol{x} \in [(-2D, 2D), (-2D, 2D)]$ in Fig. 5, where $D = 50$ in this case is the diameter of a cylinder and represents the characteristic length.[2] The prescribed values $\tilde{u}$ is imposed along its boundary. The computational domain is uniformly discretized with quadrilateral elements in different resolutions, e.g., 5×5, 10×10 and 320×320. The proposed PAGQ numerical integration scheme is applied uniformly in the computational domain with five times of refinement. A designated test problem with an exact solution is employed to check the convergence error. The following manufactured function in Eq. (32) is taken as the exact solution. By substituting Eq. (32) into Eq. (30), it can be computed that the value of the source term should be $f = 0.02\pi^2 \cos(0.01\pi x)\cos(0.01\pi y)$.

$$u_e(\boldsymbol{x}) = 100\cos(0.01\pi x)\cos(0.01\pi y) \tag{32}$$

In Fig. 5a, the elements are uniformly refined five times by the proposed PAGQ. Subsequently, the obtained convergence rate is compared with the standard Gauss quadrature rule. In Fig. 5b, an embedded discontinuity $\tilde{u}_n = 100$ is weakly imposed using symmetric & non-symmetric Nitsche's methods along the green circle of diameter $D = 50$. Its convergence rate is compared with those of the standard Gauss quadrature rule and PAGQ without embedded discontinuities. The numerical formulation of a Poisson's equation with an embedded interface is shown in Eq. (33).

$$\int_{\Omega} \nabla v \cdot \nabla u_h \, d\Omega + \frac{r_1}{h_e} \int_{\Gamma} v(u - \tilde{u}) \, d\Gamma - \int_{\Gamma} v \nabla u \cdot \boldsymbol{n} \, d\Gamma$$

[2] The characteristic length is scaled up in convergence analysis, such that the minimum size of the integration cell after five times of refinement is in the order of magnitude 1×10^{-3}.

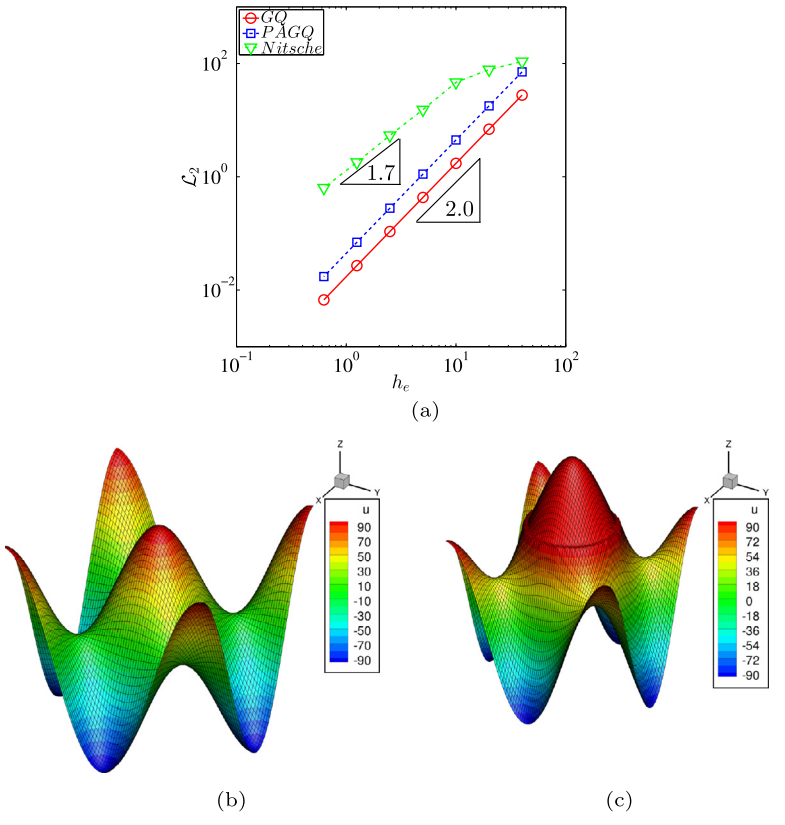


Fig. 6. Convergence analysis of a Poisson's equation with respect to Eulerian grid refinement: (a) Convergence rates of PAGQ and symmetric & non-symmetric Nitsche's method; (b) Contour plot of PAGQ; (c) Contour plot of PAGQ with embedded interface.

$$= r_2 \int_\Gamma \nabla v \cdot \boldsymbol{n}(u - \tilde{u})\, d\Gamma + \int_\Omega v f\, d\Omega + \int_\Gamma v \nabla u_h \cdot \boldsymbol{n}\, d\Gamma \quad \forall v \in \mathcal{Q}_h \subset \mathcal{Q} \tag{33}$$

where $r_1 = 1000$ and h_e respectively are the penalty parameter and the element size. In the symmetric and non-symmetric Nitsche's method, the value of r_2 is 1 and -1 respectively. The obtained convergence rates are presented in Fig. 6a, where $\mathcal{L}_2$ and h_e denote the Euclidean 2 norm and the element length respectively. $\mathcal{L}_2$ norm is computed based on Eq. (34), in which $\boldsymbol{E}$ and φ are the relative error vector and measured quantity respectively. The superscript (n) and subscript (ref) respectively denote the number of background nodes along a side and the reference solution. In the assessment of standard Gauss quadrature (GQ) and PAGQ, the exact solution is taken as the reference solution. To assess the implemented PAGQ with Nitsche's method, the solution to the case with the highest mesh resolution is chosen as the reference solution. The order of convergence is computed based on $\boldsymbol{E}$ in Eq. (35).

$$||\boldsymbol{E}^n_\varphi||_{\mathcal{L}_2} = \sqrt{(\boldsymbol{E}^n_\varphi)' \cdot \boldsymbol{E}^n_\varphi}\ ; \qquad E^n_\varphi(i) = \varphi(i) - \varphi_{ref}(\boldsymbol{x}(i)) \tag{34}$$

$$\text{order} = \frac{log(||\boldsymbol{E}^n_\varphi||_{\mathcal{L}_2}/||\boldsymbol{E}^{2n}_\varphi||_{\mathcal{L}_2})}{log(h^n_e/h^{2n}_e)} \tag{35}$$

In Fig. 6a, it shows that the convergence of PAGQ is optimal and the convergence rate is exactly 2.0. Its corresponding contour plot is shown in Fig. 6b. When an embedded interface is weakly imposed in the domain using either symmetric or non-symmetric Nitsche's method, the convergence rate of PAGQ becomes sub-optimal, approximately 1.7. No difference is observed between the convergence rates of the symmetric & non-symmetric Nitsche's method. The observed sub-optimal convergence rate agrees with the findings reported by Schilinger et al. (2016) [25] and Burman (2012) [49]. The imposed discontinuity can be observed prominently from the discontinuous contour of u in the center of the domain in Fig. 6c.

The same Poisson's equation and problem setup are used to assess the accuracy of PAGQ by comparing the $\mathcal{L}_2$ norms between the standard Gauss quadrature (GQ), the tessellation, PAGQ numerical integration schemes. The results in Table 1 shows that the accuracy of the proposed PAGQ is similar to the classical tessellation numerical integration scheme, but higher than the standard GQ scheme.

Table 1
Error introduced in the approximated solution by different integration schemes.

Resolution	Standard GQ	Tessellation	PAGQ
10×10	6.9114	14.5474	17.8756
20×20	1.7269	3.6869	4.4630
40×40	0.4317	0.9227	1.1153
80×80	0.1079	0.2309	0.2788

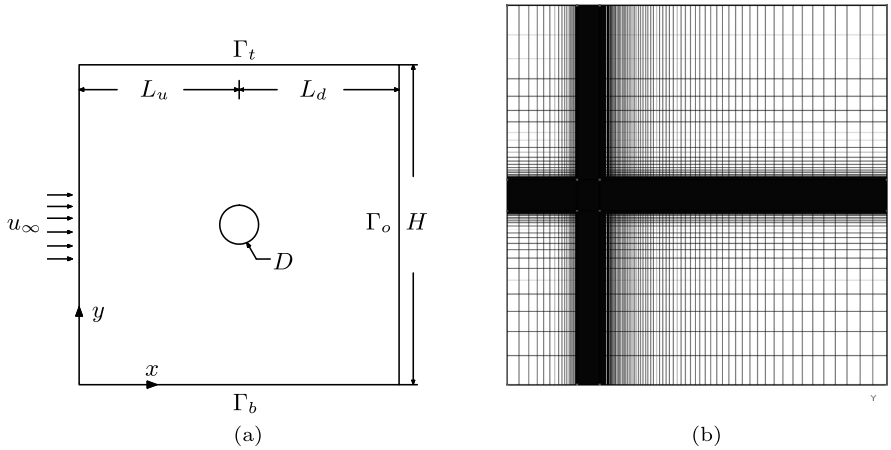


Fig. 7. A cylinder in two-dimensional flow: (a) the schematic diagram of a cylinder; (b) the resolution of the mesh used for the simulation of a cylinder in two-dimensional flow.

Table 2
Flow around a stationary circular cylinder: re-circulation length (L/D) and drag coefficient (C_d) for $Re = 20$ and 40.

		L/D	C_d
$Re = 20$	Tritton [67]	–	2.22
	Coutanceau and Bouard [68]	0.73	–
	Calhoun [69]	0.91	2.19
	Russell and Wang [70]	0.94	2.13
	Li et al. [71]	0.931	2.062
	Present (T)	0.94	2.171
	Present (A)	0.927	2.149
$Re = 40$	Tritton [67]	–	1.48
	Coutanceau and Bouard [68]	1.89	–
	Calhoun [69]	2.18	1.62
	Russell and Wang [70]	2.29	1.60
	Li et al. [71]	2.24	1.569
	Present (T)	2.27	1.608
	Present (A)	2.22	1.561

6. Numerical examples and validations

In this section, a number of numerical examples are presented to assess the accuracy and robustness of the derived numerical formulations. The performed simulations are (a) flow past a stationary cylinder, (b) flow past a rotating cylinder, (c) flow past a freely-vibrating cylinder, (d) a freely-falling particle, (e) six freely-falling particles and (f) flow past a stationary sphere. In all numerical examples, both symmetric & (penalty-free) non-symmetric Nitsche's methods are assessed. There is no significant discrepancy of accuracy between the symmetric & non-symmetric Nitsche's methods is observed in the numerical examples. The Tessellation-based PAGQ and AMR-based PAGQ are validated with literature and used interchangeably. In particular, the AMR-based PAGQ is used in the case of a stationary sphere in three-dimensional flow.

6.1. Flow past a stationary cylinder

The flow past a stationary cylinder at $Re \leq 200$, is a classical benchmark example. Its schematic diagram is shown in Fig. 7a, where $u_\infty = 1.0$, $D = 1.0$, $L_u = 50D$, $L_d = 50D$ and $H = 100D$ denote the free stream velocity, the diameter of cylinder, the upstream length, the downstream length and the height of computational domain. The mesh used for the

Table 3
Flow around a fixed circular cylinder: mean drag coefficient (C_d^{mean}), r.m.s. lift coefficient (C_l^{rms}) and Strouhal number (St) for $Re = 100$ and 200.

		C_d^{mean}	C_l^{rms}	S_t
$Re = 100$	Braza et al. [72]	1.364	±0.25	–
	Liu et al. [73]	1.350	±0.339	0.164
	Calhoun [69]	1.330	±0.298	0.175
	Russell and Wang [70]	1.380	±0.300	0.169
	Li et al. [71]	1.301	±0.324	0.167
	Kadapa et al. [74]	1.390	±0.339	0.166
	Present (T)	1.365	±0.301	0.164
	Present (A)	1.334	±0.313	0.164
$Re = 200$	Braza et al. [72]	1.40	±0.75	–
	Liu et al. [73]	1.310	±0.69	0.192
	Calhoun [69]	1.172	±0.594	0.202
	Russell and Wang [70]	1.390	±0.50	0.195
	Li et al. [71]	1.307	±0.419	0.192
	Kadapa et al. [74]	1.42	±0.594	0.194
	Present (T)	1.372	±0.648	0.194
	Present (A)	1.365	±0.655	0.194

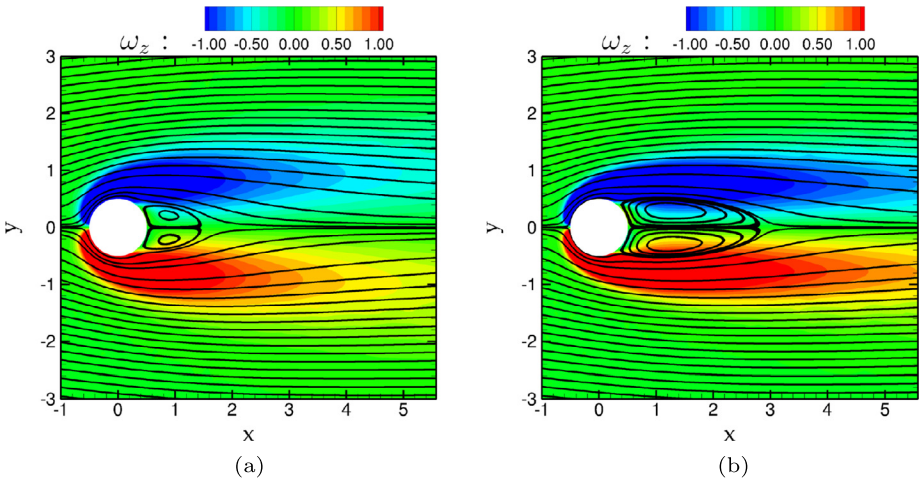


Fig. 8. ω_z contour and streamline plot of a fixed circular cylinder: (a) $Re = 20$; (b) $Re = 40$.

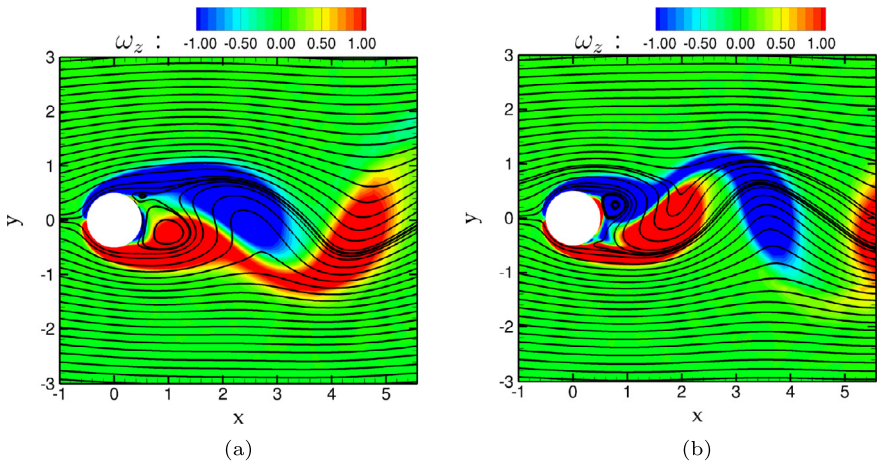


Fig. 9. ω_z contour and streamline plot of a fixed circular cylinder: (a) $Re = 100$; (b) $Re = 200$.

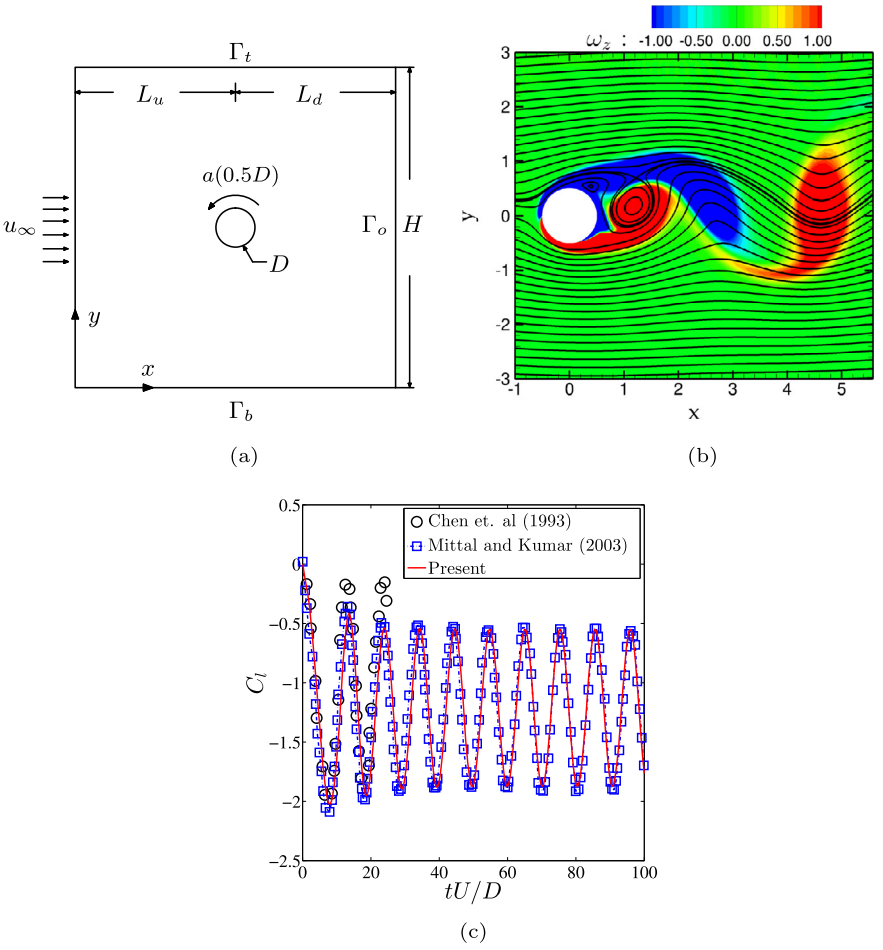


Fig. 10. Rotating cylinder in cross-flow at $Re = 200$ and $a = 1.0$: (a) Schematic diagram; (b) ω_z contour and streamline plot; (c) Time trace of lift coefficient.

simulation of a stationary, rotating or vibrating cylinder in Sect. 6.1, 6.2 and 6.3 respectively is shown in Fig. 7b. The spatial domain is discretized with quadrilateral elements. The total number of grids is 63984. Traction free boundary condition is imposed on the domain boundaries Γ_o, Γ_t and Γ_b. The fluid density $\rho^f = 1.0$, dynamic viscosity $\mu = 0.01$ and time step $dt = 0.01$ are chosen for the simulations. The obtained numerical results using both Tessellation (T) and AMR (A) are compared with literature in Table 2 and 3. The obtained numerical results from the Tessellation-based PAGQ and AMR-based PAGQ agree well with literature. The corresponding contour of z-component vorticity ω_z are plotted in Fig. 8 and 9.

6.2. Flow past a rotating cylinder

To simulate a rotating cylinder, a prescribed velocity $\tilde{\boldsymbol{u}}$ is imposed along the embedded interface. Its schematic diagram is shown in Fig. 10a. The value of $\tilde{\boldsymbol{u}}$ is computed as $[a(0.5D)]\boldsymbol{n}$, where $a = 1.0$, $D = 1.0$ and $\boldsymbol{n}$ respectively are the angular velocity, the diameter of cylinder and the unit directional vector of velocity. The proposed AMR-based PAGQ is chosen and applied for the simulation of a rotating cylinder. Its contour of ω_z is plotted in Fig. 10b. The time trace of lift coefficient agrees well with results from literature [75,76], as shown in Fig. 10c.

6.3. Flow past a freely-vibrating cylinder

In this section, the Tessellation-based PAGQ is used in the benchmark examples of two types of vibrating cylinder, e.g., transversely-vibrating (1-DoFs) cylinder in Fig. 11a and freely-vibrating (2-DoFs) cylinder in x and y directions in Fig. 11b. For the transversely-vibrating cylinder cases, $Re = 100$, $m^* = 10.0$ (mass ratio), $\zeta = 0.01$ (damping ratio) and $U_r \in [3,8]$ (reduced velocity) are chosen to set up the simulations. The obtained numerical results in Fig. 12a show a good agreement with literature [77,78]. In freely-vibrating cylinder case, the cylinder can vibrate in both x and y directions. A representative case at $Re = 150$, $m^* = 2.55$, $\zeta = 0.0$ and $U_r = 5.0$ is chosen for the purpose of validation. Its trajectory results in Fig. 12b match well with literature [78]. The contours of ω_z for representative cases are plotted in Fig. 12c and 12d respectively.

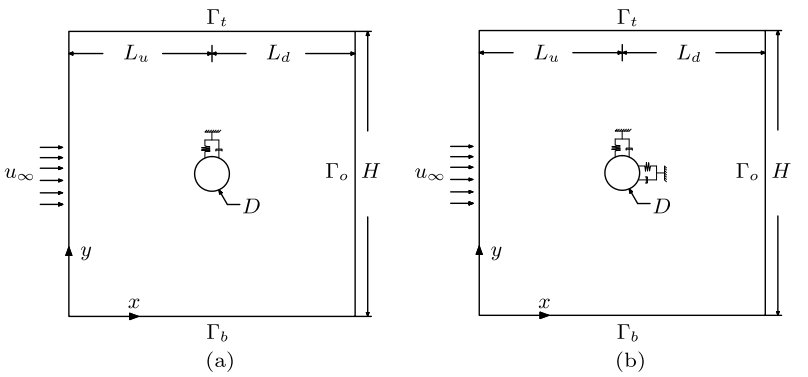


Fig. 11. Schematic diagrams of vibrating cylinder in cross-flow: (a) a transverse-vibrating (1-DoFs) cylinder; (d) a freely-vibrating (2-DoFs) cylinder.

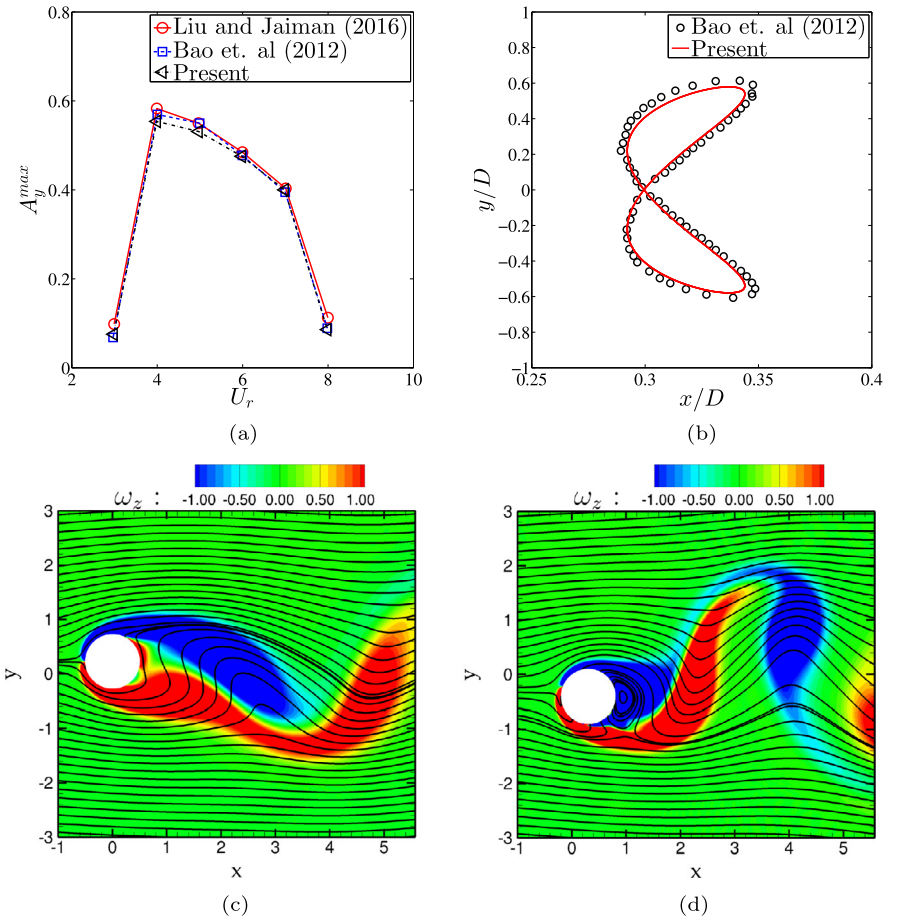


Fig. 12. Vibrating cylinder in cross-flow: (a,c) $Re = 100$, $m^* = 10.0$, $\zeta = 0.01$, $U_r = 7.0$ and vibrating in y direction; (b,d) $Re = 150$, $m^* = 2.55$, $\zeta = 0.0$, $U_r = 5.0$ and vibrating in x and y directions.

6.4. Freely-falling: a single particle

The sedimentation is a classical benchmark example for fictitious domain methods. In this example, the derived numerical formulation with the Tessellation-based PAGQ is used to simulate a circular particle freely-falling in the calm water under the gravitational force in an incompressible fluid. The particle is accelerated at rest and subsequently achieves a terminal velocity $\boldsymbol{u}_t^s$. The chosen parameters in the simulation are $m^* = 1.25$, $\rho^f = 1000$, $\mu = 0.01$ and $D = 0.25$.

The schematic diagram is shown in Fig. 13a. The subscript (e), (w) and (s) denotes the east, west and south wall boundary respectively. "no-slip" boundary condition is imposed on the east, west and south walls $\tilde{\boldsymbol{u}}_{ews}^f = \mathbf{0.0}$. Traction free

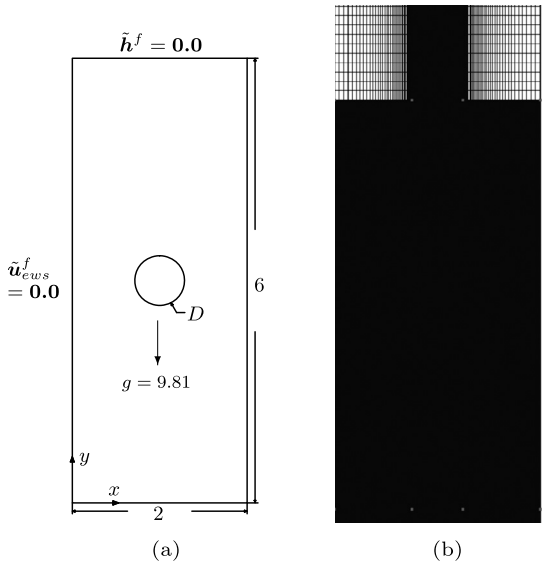


Fig. 13. A freely-falling circular particle: (a) the schematic diagram; (b) the resolution of the mesh used for the simulation of a freely-falling circular particle. The north n boundary is traction-free and the "no-slip" boundary condition is imposed along the rest of boundaries.

boundary condition is imposed on the output as $\tilde{\boldsymbol{h}}^f = \mathbf{0.0}$. The resolution of the mesh is presented in Fig. 13b. The spatial domain is discretized with totally 7.16656×10^5 quadrilateral elements. The time step used in this numerical example is $dt = 0.001$. The particle falls from the rest at $[x, y] = [1, 4]$. The contour of ω_z is plotted in Fig. 14a. The numerical results are compared with literature [79,80] in Fig. 14b and 14c. The obtained numerical results match well with literature.

6.5. Freely-falling: six particles

Similar to Sect. 6.4, the derived numerical formulation with Tessellation-based PAGQ is applied for the simulation of six particles freely-falling under the gravity. The schematic diagram is shown in Fig. 15a. The applied boundary conditions are identical to Sect. 6.4. However, this problem is different from the example of a single free-falling particle, because of the rigid-body contact and the complex interaction between particles and walls. Because no experimental results are available, this example is barely a qualitative demonstration. The resolution of the mesh is shown in Fig. 15b. The domain is discretized with 1.210944×10^6 structured quadrilateral elements. The time step taken is $dt = 0.001$. The width and height of domain are $x/D = [-3D, 3D]$, $y/D = [1D, -7D]$ respectively, where $D = 1.0$ is particle diameter. The top layer particles are rest at $x/D = 0$ at $t = 0$. The fluid density, dynamic viscosity, mass ratio respectively are $\rho^f = 1.0$, $\mu = 0.01$ and $m^* = 1.1$.

A repulsive force contact model proposed by Wan & Turek (2006) [79] is implemented in this work, which generates repulsive forces and ensures no penetration among particle and wall. The detailed algorithm of the implemented rigid-body contact model in viscous fluid is presented in Eq. (36),

$$\boldsymbol{F}^p_{i,j} = \begin{cases} 0 & \forall \quad |\boldsymbol{d}_{i,j}| > D + \varsigma \\ \frac{1}{\varepsilon_p}\boldsymbol{d}_{i,j}(D + \varsigma - |\boldsymbol{d}_{i,j}|)^2 & \forall \quad D \leqslant |\boldsymbol{d}_{i,j}| \leqslant D + \varsigma \\ \frac{1}{\varepsilon'_p}\boldsymbol{d}_{i,j}(D - |\boldsymbol{d}_{i,j}|) & \forall \quad |\boldsymbol{d}_{i,j}| \leqslant D \end{cases} \tag{36a}$$

$$\boldsymbol{F}^w_i = \begin{cases} 0 & \forall \quad |\boldsymbol{d}^*_i| > D + \varsigma \\ \frac{1}{\varepsilon_w}\boldsymbol{d}^*_i(D + \varsigma - |\boldsymbol{d}^*_i|)^2 & \forall \quad D \leqslant |\boldsymbol{d}^*_i| \leqslant D + \varsigma \\ \frac{1}{\varepsilon'_w}\boldsymbol{d}^*_i(D - |\boldsymbol{d}^*_i|) & \forall \quad |\boldsymbol{d}^*_i| \leqslant D \end{cases} \tag{36b}$$

For particle-particle collision, the impact force $\boldsymbol{F}^p_{i,j}$ is modeled in Eq. (36a), where $\varepsilon_p \approx (\Delta h)^2$ and $\varepsilon'_p \approx \Delta h$ are the stiffness parameters for the particle-particle collision in different regimes, and Δh is the element length. The value of $\boldsymbol{d}_{i,j}$ refers to the center-center vector between particle i and j. The center-center distance between circular particles is $|\boldsymbol{d}_{i,j}|$. In this article, the particles are assumed to have an identical diameter, D. The value of ς represents the buffer range between particles to generate the repulsive forces and avoid the penetration. Based on the recommendation by Wan & Turek (2006) [79], the range of the buffer range is chosen as $\varsigma \in [0.5\Delta h, 2.5\Delta h]$. On the other hand, the particle-wall impact force $\boldsymbol{F}^w_i$ is modeled in Eq. (36b), where $\varepsilon_w = 0.5\varepsilon_p$ and $\varepsilon'_w = 0.5\varepsilon'_p$ refer to the stiffness parameters for particle-wall collision. In particle-wall collision, it is assumed that there are imaginary particles of diameter D underneath the wall and forming the wall sur-

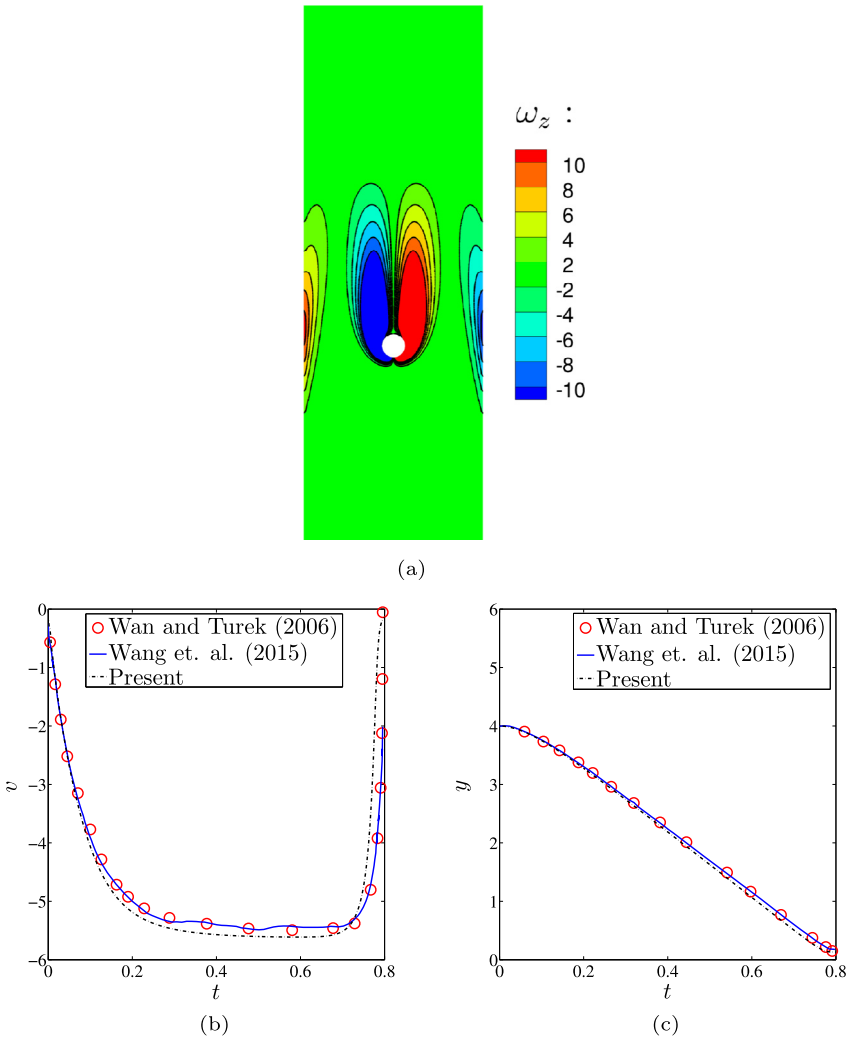


Fig. 14. A free-falling circular particle: (a) ω_z contour plot at $t = 0.4$; (c) time trace of y-component velocity; (d) time trace of y-component displacement.

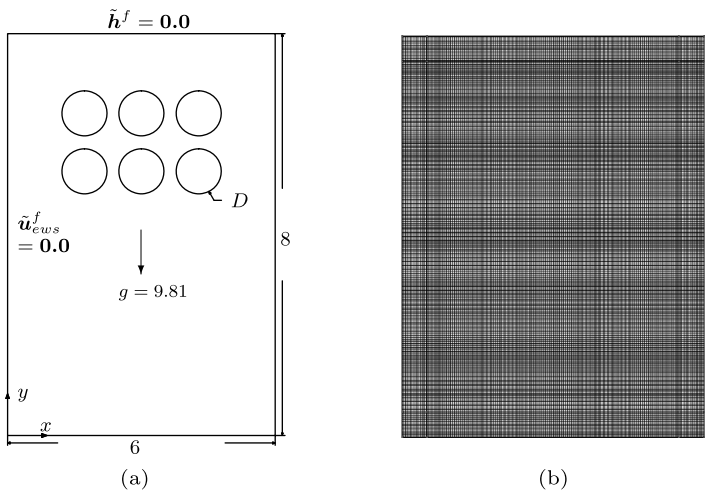


Fig. 15. Six freely-falling particles: (a) schematic diagram; (b) the resolution of the mesh used for the simulation of six freely-falling circular particles. The north n boundary is traction-free and the "no-slip" boundary condition is imposed along the rest of boundaries.

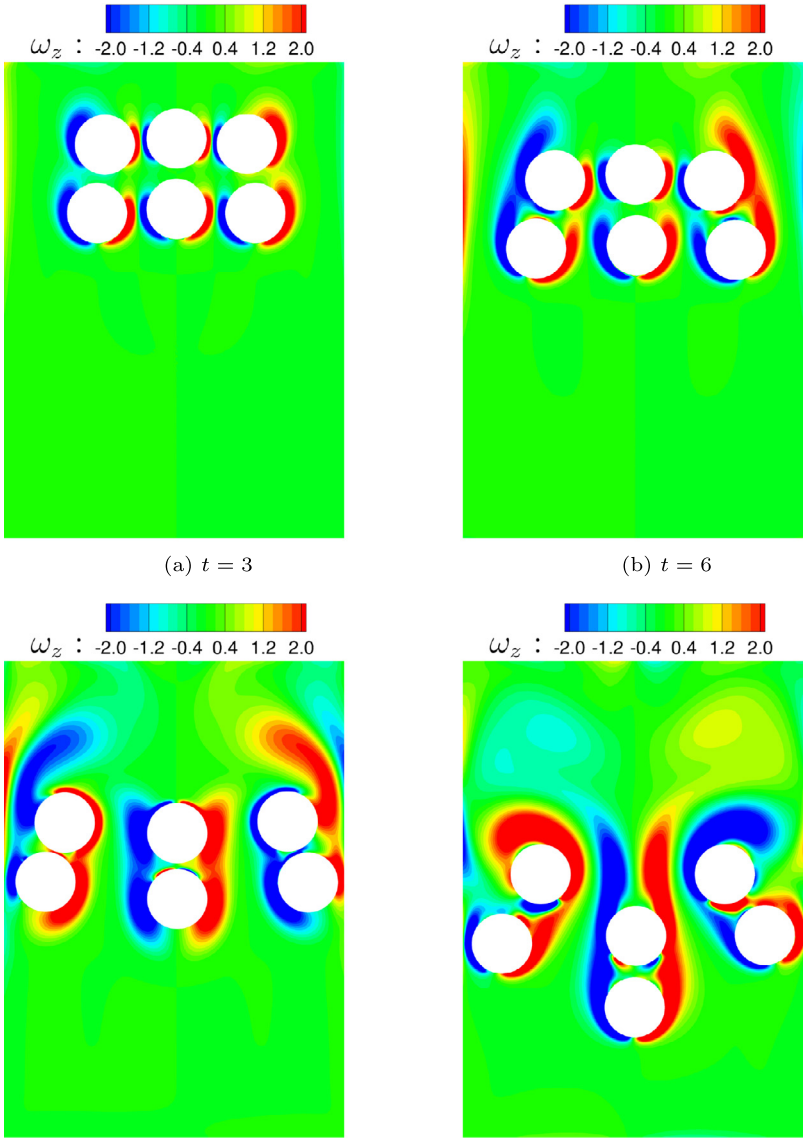


Fig. 16. ω_z contour of six free falling particles (I).

face. The center-center distance between a particle i and its nearest imaginary particle is denoted as $\boldsymbol{d}_i^*$. In simulations of multiple colliding particles, the resultant impact force $\boldsymbol{F}_i$ on particle i is computed in Eq. (37).

$$\boldsymbol{F}_i = \sum_{j=1, j\neq i}^{N} \boldsymbol{F}_{i,j}^{p} + \boldsymbol{F}_i^{w} \tag{37}$$

where N is the number of the particles colliding together. In this case, six particles are released simultaneously and fall under gravity. Complex vortex wakes are generated as particles interacting with each other and falling through the channel in Fig. 16 and 17. Eventually, all particles are rest at the bottom of the channel and the vortex wakes vanish.

6.6. Flow past a stationary sphere

The flow past a stationary sphere is simulated using the derived numerical formulations with AMR-based PAGQ in this article. The computational domain is $12D \times 8D \times 8D$, where $D = 1.0$ is the diameter of sphere. The computational domain is discretized with 2.904×10^6 structured quadrilateral elements, as shown in Fig. 18a. The center of cylinder is at location [0.0, 0.0, 0.0]. The Reynolds number, the fluid density, the inlet velocity and the time step are respectively $Re = 300$, $\rho^f = 1.0$,

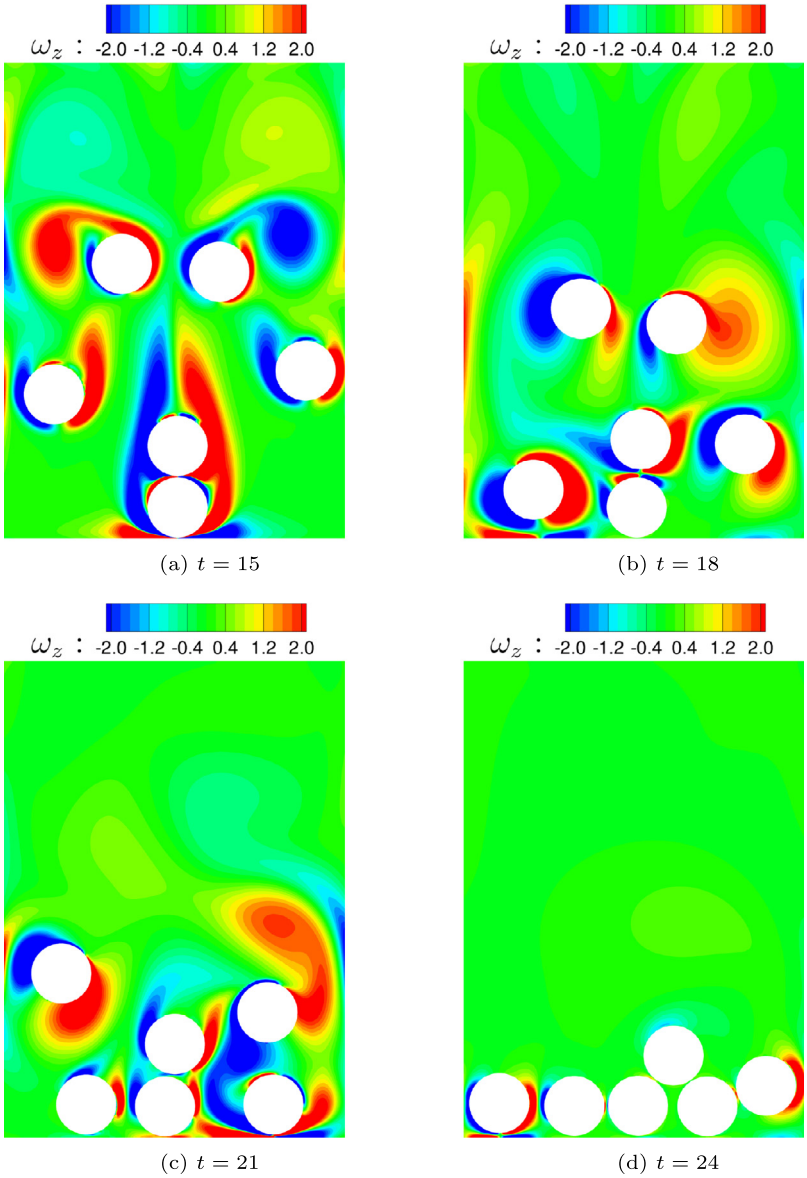


Fig. 17. ω_z contour of six free falling particles (II).

$u_\infty = 1.0$ and $dt = 0.05$. The iso-surface of vortex wake based on Q-criterion is plotted in Fig. 18b. The pressure contour on y-z plane is shown in Fig. 18c, in which a prominent pressure discontinuity is captured along the embedded surface of the sphere. The drag coefficient of the sphere at $Re = 300$ is compared with the results reported by Liska & Colonius (2017) [81], as shown in Fig. 18d

7. Conclusion

An accurate and robust numerical formulation for the application of FSI problem with large displacement and rigid-body contact was presented. This scheme was formulated based on a stabilized finite element formulation with Nitsche's methods. The Dirichlet boundary condition was weakly imposed along an embedded interface using either symmetric or (penalty-free) non-symmetric Nitsche's method together with ghost penalty stabilization. A projection-based adaptive Gauss quadrature numerical integration scheme (PAGQ) was proposed for the numerical integration with embedded discontinuity. The convergence analysis and validations were assessed in detail. Accurate numerical results were obtained in various numerical examples based on the proposed numerical formulations.

The key advantages of the present numerical formulations are (1) robustness for relative large time steps, large displacement and rigid-body contact for two and three dimensional simulations under finite element framework; (2) accurate and

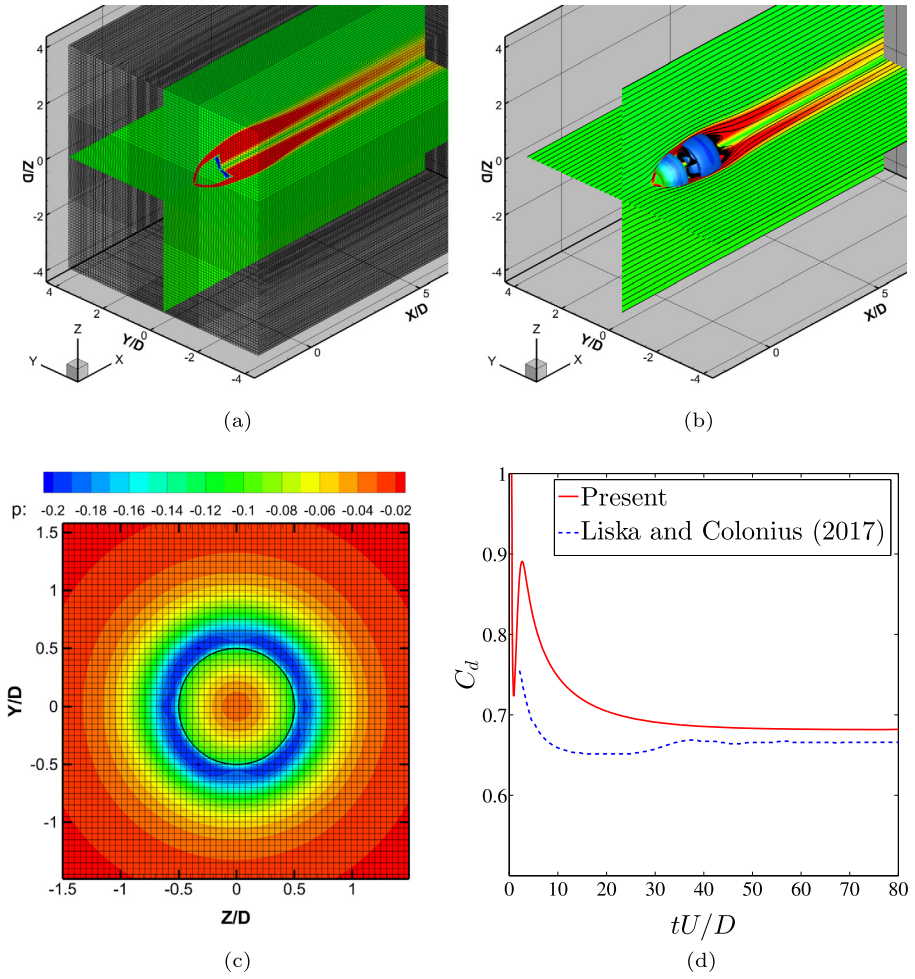


Fig. 18. Flow past a stationary sphere at $Re = 300$: (a) the three-dimensional mesh resolution $\boldsymbol{x} \in [(-2, 10), (-4, 4), (-4, 4)]$; (b) the iso-surface of vortex cluster (Q-criterion) and the contours of ω_y on $x - z$ plane and ω_z on $x - y$ plane crossing the sphere's center at $tU/D = 80$; (c) the pressure contour on $y - z$ plane crossing the sphere's center at $tU/D = 80$; (d) the time trace of drag coefficient.

staggered coupling of the fluid and structure solvers for the simulations involving rigid body contact in viscous fluid flow; (3) easy-to-implement numerical integration scheme (PAGQ) for the cut cells with embedded interface; (4) application of PAGQ in three-dimensional simulation with embedded interfaces.

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.

Acknowledgements

The first author would like to thank for the financial support from National Research Foundation Singapore through Keppel-NUS Corporate Laboratory. The conclusions put forward reflect the views of the authors alone, and not necessarily those of the institutions.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jcp.2020.109461.

References

[1] C. Hirt, A.A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.

[2] T. Belytschko, J.M. Kennedy, Computer models for subassembly simulation, Nucl. Eng. Des. 49 (1–2) (1978) 17–38.
[3] T. Belytschko, J.M. Kennedy, D. Schoeberle, Quasi-Eulerian finite element formulation for fluid-structure interaction, J. Press. Vessel Technol. 102 (1) (1980) 62–69.
[4] T.J. Hughes, W.K. Liu, T.K. Zimmermann, Lagrangian-Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Eng. 29 (3) (1981) 329–349.
[5] J. Donea, S. Giuliani, J.P. Halleux, An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions, Comput. Methods Appl. Mech. Eng. 33 (1) (1982) 689–723, https://doi.org/10.1016/0045-7825(82)90128-1.
[6] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
[7] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252, https://doi.org/10.1016/0021-9991(77)90100-0.
[8] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517, https://doi.org/10.1017/S0962492902000077.
[9] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier-Stokes equations, SIAM J. Sci. Comput. 25 (3) (2003) 832–856, https://doi.org/10.1137/S1064827502414060.
[10] X. Wang, W.K. Liu, Extended immersed boundary method using FEM and RKPM, Comput. Methods Appl. Mech. Eng. 193 (12–14) (2004) 1305–1321.
[11] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[12] C. Shu, N. Liu, Y.T. Chew, A novel immersed boundary velocity correction–lattice Boltzmann method and its application to simulate flow past a circular cylinder, J. Comput. Phys. 226 (2) (2007) 1607–1622.
[13] R. Glowinski, T.W. Pan, J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 112 (1–4) (1994) 133–148.
[14] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiph. Flow 25 (5) (1999) 755–794.
[15] R. Van Loon, P.D. Anderson, F.P. Baaijens, F.N. Van de Vosse, A three-dimensional fluid–structure interaction method for heart valve modelling, C. R. Mecanique 333 (12) (2005) 856–866.
[16] A. Gerstenberger, W.A. Wall, An extended finite element method/Lagrange multiplier based approach for fluid–structure interaction, Comput. Methods Appl. Mech. Eng. 197 (19) (2008) 1699–1714.
[17] U.M. Mayer, A. Popp, A. Gerstenberger, W.A. Wall, 3d fluid–structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach, Comput. Mech. 46 (1) (2010) 53–67.
[18] F. Alauzet, B. Fabrèges, M.A. Fernández, M. Landajuela, Nitsche-XFEM for the coupling of an incompressible fluid with immersed thin-walled structures, Comput. Methods Appl. Mech. Eng. 301 (2016) 300–335.
[19] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers (Revue française d'automatique, informatique, recherche opérationnelle), Anal. Numér. 8 (R2) (1974) 129–151.
[20] F. Brezzi, K.J. Bathe, A discourse on the stability conditions for mixed finite element formulations, Comput. Methods Appl. Mech. Eng. 82 (1–3) (1990) 27–57.
[21] J. Nitsche, Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, in: Abhandlungen aus dem, in: Mathematischen Seminar der Universität Hamburg, vol. 36, Springer, 1971, pp. 9–15.
[22] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, Appl. Numer. Math. 62 (4) (2012) 328–341.
[23] A. Massing, M.G. Larson, A. Logg, M.E. Rognes, A stabilized Nitsche fictitious domain method for the Stokes problem, J. Sci. Comput. 61 (3) (2014) 604–628.
[24] W.G. Dettmer, C. Kadapa, D. Perić, A stabilised immersed boundary method on hierarchical b-spline grids, Comput. Methods Appl. Mech. Eng. 311 (2016) 415–437.
[25] D. Schillinger, I. Harari, M.C. Hsu, D. Kamensky, S.K. Stoter, Y. Yu, Y. Zhao, The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements, Comput. Methods Appl. Mech. Eng. 309 (2016) 625–652.
[26] C. Kadapa, W.G. Dettmer, D. Perić, A stabilised immersed boundary method on hierarchical b-spline grids for fluid–rigid body interaction with solid–solid contact, Comput. Methods Appl. Mech. Eng. 318 (2017) 242–269.
[27] Z. Zou, W. Aquino, I. Harari, Nitsche's method for Helmholtz problems with embedded interfaces, Int. J. Numer. Methods Eng. 110 (7) (2017) 618–636.
[28] C. Kadapa, W.G. Dettmer, D. Perić, A stabilised immersed framework on hierarchical b-spline grids for fluid-flexible structure interaction with solid–solid contact, Comput. Methods Appl. Mech. Eng. 335 (2018) 472–489.
[29] E. Burman, Ghost penalty, C. R. Math. 348 (21–22) (2010) 1217–1220.
[30] G.R. Liu, Mesh Free Methods: Moving Beyond the Finite Element Method, CRC Press, 2002.
[31] T. Belytschko, R. Gracie, G. Ventura, A review of extended/generalized finite element methods for material modeling, Model. Simul. Mater. Sci. Eng. 17 (4) (2009) 043001.
[32] H. Samet, Applications of Spatial Data Structures: Computer Graphics, Image Processing, and Gis, Addison-Wesley, Reading, Mass, 1990.
[33] M.D. Berg, O. Cheong, M.V. Kreveld, M. Overmars, Computational Geometry: Algorithms and Applications, Springer-Verlag TELOS, 2008.
[34] F.J. Blom, A monolithical fluid-structure interaction algorithm applied to the piston problem, Comput. Methods Appl. Mech. Eng. 167 (3–4) (1998) 369–391.
[35] W. Dettmer, D. Perić, A computational framework for fluid–rigid body interaction: finite element formulation and applications, Comput. Methods Appl. Mech. Eng. 195 (13–16) (2006) 1633–1666.
[36] R.K. Jaiman, M.Z. Guan, T.P. Miyanawala, Partitioned iterative and dynamic subgrid-scale methods for freely vibrating square-section structures at subcritical Reynolds number, Comput. Fluids 133 (2016) 68–89.
[37] C. Kadapa, W.G. Dettmer, D. Perić, A fictitious domain/distributed Lagrange multiplier based fluid–structure interaction scheme with hierarchical B-spline grids, Comput. Methods Appl. Mech. Eng. 301 (2016) 1–27.
[38] W.G. Dettmer, D. Perić, A new staggered scheme for fluid–structure interaction, Int. J. Numer. Methods Eng. 93 (1) (2013) 1–22.
[39] A. Placzek, J.F. Sigrist, A. Hamdouni, Numerical simulation of an oscillating cylinder in a cross-flow at low Reynolds number: forced and free oscillations, Comput. Fluids 38 (1) (2009) 80–100.
[40] J. Chung, G.M. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method, J. Appl. Mech. 60 (2) (1993) 371–375.
[41] K.E. Jansen, C.H. Whiting, G.M. Hulbert, A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method, Comput. Methods Appl. Mech. Eng. 190 (3) (2000) 305–319.
[42] J.G. Heywood, R. Rannacher, S. Turek, Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 22 (5) (1996) 325–352.
[43] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1–3) (1982) 199–259.
[44] F. Shakib, T.J.R. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 89 (1–3) (1991) 141–219.

[45] T.E. Tezduyar, S. Mittal, S.E. Ray, R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, Comput. Methods Appl. Mech. Eng. 95 (2) (1992) 221–242.
[46] L.P. Franca, S.L. Frey, Stabilized finite element methods: II. The incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 99 (2) (1992) 209–233.
[47] I. Harari, T.J. Hughes, What are c and h?: inequalities for the analysis and design of finite element methods, Comput. Methods Appl. Mech. Eng. 97 (2) (1992) 157–192.
[48] J. Benk, Immersed boundary methods within a PDE toolbox on distributed memory systems, Ph.D. thesis, Universitätsbibliothek der TU München, 2012.
[49] E. Burman, A penalty-free nonsymmetric Nitsche-type method for the weak imposition of boundary conditions, SIAM J. Numer. Anal. 50 (4) (2012) 1959–1981.
[50] J. Gregory, Quadratic Form Theory and Differential Equations, vol. 152, Elsevier, 1981.
[51] R.D. Cook, Concepts and Applications of Finite Element Analysis, 1981.
[52] V.N. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Netw. 10 (5) (1999) 988–999.
[53] J.A.K. Suykens, Least Squares Support Vector Machines, World Scientific, River Edge, N.J, Singapore, 2002.
[54] M.N. Murty, R. Raghava, Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks, Springer, 2016.
[55] B. Liu, Y. Jin, A. Magee, L. Yiew, S. Zhang, System identification of Abkowitz Model for ship maneuvering motion based on ε-support vector regression, in: International Conference on Offshore Mechanics and Arctic Engineering, vol. 58844, American Society of Mechanical Engineers, 2019, V07AT06A067.
[56] J.L. Lumley, The structure of inhomogeneous turbulence, in: A.M. Yaglom, V.I. Tatarski (Eds.), Atmospheric Turbulence and Wave Propagation, 1967, pp. 166–178.
[57] J.L. Lumley, Stochastic Tools in Turbulence, Courier Corporation, 2007.
[58] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28.
[59] M.R. Jovanović, P.J. Schmid, J.W. Nichols, Sparsity-promoting dynamic mode decomposition, Phys. Fluids 26 (2) (2014) 024103.
[60] B. Liu, R.K. Jaiman, Dynamics and stability of gap-flow interference in a vibrating side-by-side arrangement of two circular cylinders, J. Fluid Mech. 855 (2018) 804–838.
[61] J.H. Wilkinson, The Algebraic Eigenvalue Problem, vol. 662, Oxford Clarendon, 1965.
[62] S.G. Mikhlin, Variational Methods in Mathematical Physics, vol. 50, Pergamon Press, 1964 [distributed by Macmillan, New York].
[63] A. Ern, J.L. Guermond, Theory and Practice of Finite Elements, vol. 159, Springer Science & Business Media, 2013.
[64] A. Rätz, A. Voigt, PDE's on surfaces—a diffuse interface approach, Commun. Math. Sci. 4 (3) (2006) 575–590.
[65] X. Li, J. Lowengrub, A. Rätz, A. Voigt, Solving PDEs in complex geometries: a diffuse domain approach, Commun. Math. Sci. 7 (1) (2009) 81.
[66] L.H. Nguyen, S.K.F. Stoter, M. Ruess, M.A.S. Uribe, D. Schillinger, The diffuse Nitsche method: Dirichlet constraints on phase-field boundaries, Int. J. Numer. Methods Eng. 113 (4) (2018) 601–633.
[67] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, J. Fluid Mech. 6 (4) (1959) 547–567.
[68] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (2) (1977) 231–256.
[69] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, J. Comput. Phys. 176 (2) (2002) 231–275.
[70] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, J. Comput. Phys. 191 (1) (2003) 177–205.
[71] Z. Li, R.K. Jaiman, B.C. Khoo, An immersed interface method for flow past circular cylinder in the vicinity of a plane moving wall, Int. J. Numer. Methods Fluids 81 (10) (2016) 611–639.
[72] M. Braza, P.H.H.M. Chassaing, H.H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid Mech. 165 (1986) 79–130.
[73] C. Liu, X. Zheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, J. Comput. Phys. 139 (1) (1998) 35–57.
[74] C. Kadapa, W.G. Dettmer, D. Perić, A fictitious domain/distributed Lagrange multiplier based fluid–structure interaction scheme with hierarchical b-spline grids, Comput. Methods Appl. Mech. Eng. 301 (2016) 1–27.
[75] Y.M. Chen, Y.R. Ou, A.J. Pearlstein, Development of the wake behind a circular cylinder impulsively started into rotatory and rectilinear motion, J. Fluid Mech. 253 (1993) 449–484.
[76] S. Mittal, B. Kumar, Flow past a rotating cylinder, J. Fluid Mech. 476 (2003) 303–334.
[77] B. Liu, R.K. Jaiman, Interaction dynamics of gap flow with vortex-induced vibration in side-by-side cylinder arrangement, Phys. Fluids 28 (12) (2016) 127103.
[78] Y. Bao, C. Huang, D. Zhou, J. Tu, Z. Han, Two-degree-of-freedom flow-induced vibrations on isolated and tandem cylinders with varying natural frequency ratios, J. Fluids Struct. 35 (2012) 50–75.
[79] D. Wan, S. Turek, Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method, Int. J. Numer. Methods Fluids 51 (5) (2006) 531–566.
[80] Y. Wang, C. Shu, C.J. Teo, J. Wu, An immersed boundary-lattice Boltzmann flux solver and its applications to fluid–structure interaction problems, J. Fluids Struct. 54 (2015) 440–465.
[81] S. Liska, T. Colonius, A fast immersed boundary method for external incompressible viscous flows using lattice Green's functions, J. Comput. Phys. 331 (2017) 257–279.