Stability evaluation of high-order splitting method for incompressible flow based on discontinuous velocity and continuous pressure

Liyang Xu\textsuperscript{1}, Xinhai Xu\textsuperscript{1,2}, Xiaoguang Ren\textsuperscript{1,2}, Yunrui Guo\textsuperscript{1}, Yongquan Feng\textsuperscript{1} and Xuejun Yang\textsuperscript{1,2}

Abstract
In this work, we deal with high-order solver for incompressible flow based on velocity correction scheme with discontinuous Galerkin discretized velocity and standard continuous approximated pressure. Recently, small time step instabilities have been reported for pure discontinuous Galerkin method, in which both velocity and pressure are discretized by discontinuous Galerkin. It is interesting to examine these instabilities in the context of mixed discontinuous Galerkin–continuous Galerkin method. By means of numerical investigation, we find that the discontinuous Galerkin–continuous Galerkin method shows great stability at the same configuration. The consistent velocity divergence discretization scheme helps to achieve more accurate results at small time step size. Since the equal order discontinuous Galerkin–continuous Galerkin method does not satisfy inf-sup stability requirement, the instability for high Reynolds number flow is investigated. We numerically demonstrate that fine mesh resolution and high polynomial order are required to obtain a robust system. With these conclusions, discontinuous Galerkin–continuous Galerkin method is able to achieve high-order spatial convergence rate and accurately simulate high Reynolds flow. The solver is tested through a series of classical benchmark problems, and efficiency improvement is proved against pure discontinuous Galerkin scheme.

Keywords
Discontinuous Galerkin, incompressible flow, high-order discretization, velocity correction method, HopeFOAM

Date received: 11 May 2018; accepted: 24 April 2019

Handling Editor: Roslinda Nazar

Introduction
Discontinuous Galerkin method (DGM) is one of the most potential high-order discretization method among the state-of-the-art methods, such as finite difference methods and finite volume method (FVM). DGM has attracted lots of attention from both academic and industry community for the easiness to achieve high-order spacial convergence rate, geometrical flexibility, numerical stability, and good extensibility. On the contrary, the numerical simulation of incompressible Navier–Stokes (INS) equations is a key issue in the research of DGM but still far from completeness. This article is devoted to discussing the stability DGM coupled with continuous Galerkin (CG) in the simulation of INS problems.

\textsuperscript{1}State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, China
\textsuperscript{2}Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, China

Corresponding author:
Xinhai Xu, Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing 100091, China.
Email: xuxinhai@nudt.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (http://www.creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
For the simulation of INS equations, coupled solver solves all components of velocity and pressure in a monolithic manner, which require the solution of saddle point problem and non-linear iteration. Due to the complexity, the coupled solver has only been applied in small scale academic cases. An alternative choice is splitting method proposed by Guermond et al.\textsuperscript{1} and Shahbazi et al.\textsuperscript{2} to solve the strong coupling of pressure and velocity. Contrast to the coupled solver, a Poisson and Helmholtz equation are solved sequentially for pressure and velocity, which shows to be much more effective in many applications. Splitting methods are also widely used in the context of DGM discretization.

In recent years, several kinds of instabilities have been reported for splitting methods when employed with DGM. Ferrer and Willden\textsuperscript{3} show the small time step instability in the simulation of unsteady Stokes and vortex flow with this scheme. They find that a minimum time step $\Delta t(h, k)$ is required to achieve long-time stable simulation. This idea is extended by Ferrer et al.\textsuperscript{4} showing that the minimum time step size is constrained by $C_b h^2 k^{-3}$ for both continuous and DGM. A series of papers are published trying to find out the reasons behind and remedy these problems. Krank et al.\textsuperscript{5} attribute the instability to the non-zero velocity divergence and a div-div stabilization approach is proposed. Similar work has also been done by Joshi et al.\textsuperscript{6} to wipe the spurious divergence at element interfaces. Emamy et al.\textsuperscript{7} propose a modified version of projection scheme to reduce the influence of time step size. Steinmoeller et al.\textsuperscript{8} also review the instabilities of DG in simulating under-resolved and small viscosity cases. And they suggest the instabilities are related to the inf-sup instability of DGM. Recently, Fehn et al.\textsuperscript{9} propose that the small time instability is due to the discretization of velocity divergence and pressure gradient. In the numerical investigation, they find that a stable and robust method can be obtained using integration by parts.

In the theory of finite element, a mixed-order continuous velocity and discontinuous pressure are usually chosen to solve the INS problem, which exhibit good numerical behavior. But considering moderate-to-high Reynolds flow, discontinuous velocity helps to keep stability when convection term becomes dominant. As a supplement to pure DG (both of velocity and pressure are discretized with DGM), Botti and Di Pietro\textsuperscript{10} present a pressure-correction scheme for the INS equations, which discrete velocity with standard DGM while pressure with CG. A major preliminary of this work is that the procedure of solving the pressure during the splitting method is the most expensive stage, while approximation of the pressure with CG can significantly reduce computational cost compared to monolithic pure DG. This is much more favorable for three-dimensional (3D) simulations. Meanwhile, the stability for convection dominant problem is also inherited. This scheme has been validated against a large set of classical benchmarks and shows great computational efficiency improvement.

However, the instabilities encountered in pure DG method for small time step size and under-resolved cases may also exhibit in mixed DG-CG method. To our best knowledge, the evaluation on the stability of DG-CG method for INS equations has not been published. In this article, we develop a high-order INS solver based on velocity correction scheme, in which mixed equal order DG-CG method is used. The stabilities of DG-CG method for small time step size and small viscosity are numerically investigated. This work is crucial to the employment of DG-CG method and may also shed light on the remedy of instabilities in pure DG scheme. Major innovations of this article include the following:

- The behavior of DG-CG method for small time step instability is investigated. On the simulation of Pearson vortex problem, DG-CG method shows great stability on the regimes where pure DG method by Hesthaven and Warburton\textsuperscript{11} reported to be unstable. In addition, the consistent velocity divergence discretization scheme helps to achieve more accurate results at small time step size.
- The spatial stability of DG-CG method in low viscosity situations is numerically investigated through comprehensive experiments for steady state Navier–Stokes equations. We find that sufficient fine mesh and high polynomial order help to make DG-CG scheme more robust and accurate for high Reynolds number flow.
- The DG-CG solver is validated through a series of classical problems. With the guidance of our conclusions, stabilities for relative high Reynolds flow and efficiency against pure DG scheme have been proved.

The remaining parts of this article are organized as follows: We start by introducing the governing equations and velocity correction steps. In the “Spatial discretization” section, the spatial discretization is presented in detail. “Numerical results” section numerically investigate the small time and spatial stability through a serious of classical benchmark problems. The temporal and spatial convergence rate and computational efficiency are also examined. Finally, this article in concluded in the “Conclusion” section.
Methodology

Governing equations

Here, we consider the unsteady INS equations in conservation form defined on physical domain $\Omega$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = - \nabla p + \nu \nabla^2 \mathbf{u}$$

(1)

$$\nabla \cdot \mathbf{u} = 0$$

(2)

where $\mathbf{u} = (u_1, \ldots, u_d)^T$ is the velocity, $p$ indicates the kinematic pressure, and $\nu$ is the kinematic viscosity. The definition of convection flux $\mathbf{F}$ reads $\mathbf{F}(\mathbf{u}) = \mathbf{u} \otimes \mathbf{u}$.

For a particular problem, the initial conditions are given as $\mathbf{u}(t = 0) = \mathbf{u}_0$ and $p(t = 0) = p_0$ in $\Omega$, where $\mathbf{u}_0$ and $p_0$ also have to fulfill the governing equations constraints. The Dirichlet conditions defined on $\partial \Omega^D$ and Neumann conditions on $\partial \Omega^N$ are prescribed as

$$\mathbf{u} = \mathbf{g}_u \text{ on } \partial \Omega^D \times [0, T]$$

(3)

$$\nu \nabla \mathbf{u} \cdot \mathbf{n} - p \cdot \mathbf{n} = 0 \text{ on } \partial \Omega^N \times [0, T]$$

(4)

with the constraints $\partial \Omega^D \cap \partial \Omega^N = \emptyset$ and $\partial \Omega^D \cap \partial \Omega^N = \emptyset$, respectively. Here, $\mathbf{n}$ denotes the outward normal vectors. Since the gradient instead of absolute value of pressure makes sense in INS, the resultant value of $p$ can be adjusted by imposing $p = g_p$ on $\partial \Omega^N$. In case of pure Dirichlet boundary conditions, two additional constraints are needed, with $\int_{\partial \Omega} \mathbf{g} \cdot \mathbf{n} dx = 0$ to satisfy the continuity equation and $\int_{\partial \Omega} p dx = 0$ to guarantee a unique pressure solution.

Velocity correction method and temporal discretization

The velocity correction method combined with discontinuous Galerkin (DG) has been investigated by several works, including Hesthaven and Warburton,11 Ferrer et al.,3 and Emamy et al.7 Here, the fixed semi-explicit dual-splitting schemes are used to solve equations (1) and (2). Within this scheme, the INS equations are split into three distinct substeps which can be solved successively.

In the first substep, equation (1) is advanced using Adams–Bashforth time scheme with the nonlinear convection term treated explicitly

$$\frac{\gamma_0 \tilde{\mathbf{u}} - \sum_{i=0}^{J-1} \alpha_i \mathbf{u}^{n-i}}{\Delta t} = - \sum_{i=0}^{J-1} \beta_i F(\mathbf{u}^{n-i})$$

(5)

where $J = 1, 2$ corresponding to different temporal order. The value of constant time integration coefficients can refer to Karniadakis et al.12 However, this temporal scheme should be performed from $J = 1$ and increase the order successively, for it can not start automatically when $J > 1$.

The second substep is pressure projection, in which the intermediate velocity is updated with the following equation

$$\frac{\gamma_0 \tilde{\mathbf{u}} - \tilde{\mathbf{u}}}{\Delta t} = - \nabla p^{n+1}$$

(6)

By projecting the velocity into weakly divergence free space, equation (6) turns into a Poisson equation for pressure at time $t^{n+1}$ as

$$- \nabla^2 p^{n+1} = - \frac{\gamma_0}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}$$

(7)

Specially, Neumann boundary conditions should be applied on $\Omega^D$ for pressure to keep consistent with governing equations. Multiplying the momentum equation (1) with face normal, we obtain the formulation of pressure gradient given as

$$\gamma_0 \tilde{\mathbf{u}} \cdot \nabla p^{n+1} = - \sum_{i=0}^{J-1} \beta_i \tilde{\mathbf{u}}$$

(8)

$$\cdot \left( \frac{\partial \mathbf{u}^{n-i}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}^{n-i}) + \nu \nabla \times \omega^{n-i} \right)$$

where $\omega = \nabla \times \mathbf{u}$ is the vorticity. It should be noted that only the solenoidal part of viscosity is accounted for instead of original formulation, yielding the constraint of divergence free. Then, intermediate velocity can be updated with $p^{n+1}$ by

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}} - \frac{\Delta t}{\gamma_0} \nabla p^{n+1}$$

(9)

The final substep takes the viscosity into consideration and solves an implicit Helmholtz equation

$$- \nabla^2 \mathbf{u}^{n+1} + \frac{\gamma_0}{\nu \Delta t} \mathbf{u} = \frac{\gamma_0}{\nu \Delta t} \tilde{\mathbf{u}}$$

(10)

Spatial discretization

DG and CG space

The space discretization deployed here is based on the combination of DG and standard CG method. This choice benefits from the stability of DG on handling convection term and efficiency of CG in solving the projection step.

The computational domain denoted by $\Omega_k = \bigcup_{k=1}^{K} D^k$ is a numerical approximation of $\Omega$, which consists of $K$ geometric compatible non-overlapping cells $D^k$. Theoretically, the shape of $D^k$ can be arbitrary. Every interior face $\partial D^k \cap \partial D^l$ contains information from two adjacent cells, where the current
cell is marked by a symbol “−” while adjacent cell by “+.” With these notations, the average of a scalar or vector variable \( u \) on a face is defined as

\[
\{\{u\}\} = \frac{u^- + u^+}{2}
\]  

(11)

The discontinuity of variables across adjacent cell faces may be expressed in similar formulation

\[
[[u]] = \mathbf{n}^- u^- + \mathbf{n}^+ u^+, \quad \{[u]\} = \mathbf{n}^- \cdot u^- + \mathbf{n}^+ \cdot u^+
\]  

(12)

To discrete in either continuous or discontinuous formulation, approximate solution \( u_h \) in cell \( k \) is represented by a linear combination of basis functions

\[
u_h^k(x,t) = \sum_{n=1}^{N_p} \tilde{u}^k_n(t) \psi_n(x), \quad x \in D^k
\]

where \( N_p \) is the number of basis under polynomial order \( N \) and \( \psi_n \) is the modal basis while \( \tilde{u}^k_n \) is the nodal basis. Piece-wise polynomial on each cell composite is the entire solution, namely

\[
u(x,t) \simeq u_h(x,t) = \sum_{k} u_h^k(x,t)
\]

(14)

The choice of basis function is critical to space discretization. Here, the discontinuous and continuous basis function space is defined as

\[
DG^N(\Omega_h) = \left\{ \nu_h \in L^2(\Omega_h) : \nu_h|_{D_k} \in \mathbb{P}_N(D_k) \right\}
\]

\[
CG^N(\Omega_h) = \left\{ q_h \in \mathcal{C}^0(\Omega_h) : q_h|_{D_k} \in \mathbb{P}_N(D_k) \right\}
\]

(15)

Following the method introduced in the work by Hesthaven and Warburton, a set of hierarchical orthogonal basis is constructed by \( L^2 \) norm-based Gram–Schmidt method. For high-dimensional cases, expansion of basis function can be obtained by the tensor product of one-dimensional polynomial instead of generating specific function space.

Till now, we have an arbitrary order modal basis to represent the continuous or discontinuous function space. Although they are equivalent mathematically, the nodal basis is more favorable for space discretization for its convenience on handling boundary conditions. By providing the position of the nodal point, namely, Legendre–Gauss–Lobatto (LGL) quadrature points, the Lagrangian nodal basis can be derived from a modal basis. The transformation matrix between nodal and modal basis is called Vandermond matrix. LGL points are essential to keep the matrix well conditioned. Since the basis functions talked above are defined on each cell piece-wisely, they maintain discontinuity on cell interfaces. However, to satisfy \( C^0 \) constraints in a continuous framework, the variable value of two adjacent cells should be the same on cell interfaces.

**Variational form of discretization terms**

**Convective term.** The convective term is in the momentum equation of the first substep. To increase stability for convection dominant flow, upwind DG method is used in the discretization of the convective term. Integrating by parts and using Lax–Friedrichs flux, the discrete formulation of the convective term can be derived as

\[
(v_h \cdot \nabla \cdot \mathcal{F})_{D_k} = (v_h \cdot \nabla \cdot \mathcal{F}(u_h))_{D_k} - \frac{1}{2} \left( v_h \cdot \left[ \mathcal{F}(u_h) \right] \right)_{\partial D_k} + \frac{\lambda}{2} \left( v_h \cdot \left[ u_h \right] \right)_{\partial D_k}
\]

(16)

where \( \lambda \) refers to maximum eigenvalue of respective flux Jacobian. In this case, \( \lambda \) is approximated by the maximum velocity magnitude along flux direction on entire \( \partial D_k \). Boundary conditions are weakly imposed by given corresponding \( u^+ \) value on \( \partial \Omega_h \), which is set using a mirror principle

\[
u^+_h + \nu^-_h = \bar{g}_u \text{ on } \partial \Omega_D, \quad \nu^+_h = \nu^-_h \text{ on } \partial \Omega_N
\]

(17)

**Laplacian term.** The Laplacian term in pressure projection step is discretized with CG method. Since variables are continuous across interfaces in the framework of CG, flux integration can be ignored for internal faces. The variational formulation is given as

\[
\left( \nabla \cdot \nabla q_h, q_h \right)_{D_k} = - \left( \nabla q_h, \nabla p_h \right)_{D_k} + \left( q_h, \mathbf{n} \cdot \nabla p_h \right)_{\partial D_k \cap \partial \Omega_N}
\]

(18)

where the pressure gradient on Dirichlet boundaries is calculated by equation (8). Reference values on Neumann conditions are imposed by altering discretized linear system explicitly.

In the third viscosity substep, Laplacian term is discretized within DG framework using internal penalty formulation following methods used by Hesthaven and Warburton, which helps to avoid spurious jumps on interfaces and provides sufficient stabilization. For the \( x \) component of velocity \( u_h \), the variational formulation of Laplacian term reads
\begin{align}
(\nabla \cdot u_h, q_h)_{D_h} &= (\nabla v_h, \nabla u_h)_{D_h} - \frac{1}{2} (\hat{n} \cdot \nabla v_h, \hat{n} \cdot [u_h])_{\partial D_h} \\
&
- (v_h \cdot \hat{n} \cdot \{\nabla u_h\})_{\partial D_h \cap \partial \Omega} + (\hat{n} \cdot \nabla v_h, [u_h])_{\partial D_h} \\
&
- (\hat{n} \cdot \nabla v_h, u_h^i)_{\partial D_h \cap \partial \Omega} + (\hat{n} \cdot \nabla v_h, u_D)_{\partial D_h \cap \partial \Omega} \\
&
+ (\hat{n} \cdot \nabla v_h, u_h^e)_{\partial D_h \cap \partial \Omega} - (\hat{n} \cdot \nabla v_h, u_D)_{\partial D_h \cap \partial \Omega} \\
&
- (v_h \cdot \hat{n} \cdot \nabla v_h)_{\partial D_h \cap \partial \Omega}. 
\end{align}

(19)

The penalty parameter \( \tau^k \) for each element is defined as

\[
\tau^k = \frac{(N + 1)^2}{h^d} 
\]

(20)
in which \( \tau \) is an empirical constant value and \( h \) is the mesh length scale of cell \( k \). In the following context, \( \tau \) is set to a default value of 20 taken from work by Hesthaven and Warburton.\(^{11} \)

**Gradient and divergence term.** According to the work by Fehn et al.,\(^9 \) the implementations of gradient and divergence term are critical to avoid small time instability. So, we shall illustrate the discretization of these two terms in detail, and show the differences between various methods.

The implementation of our DG-CG method is similar to the work by Botti and Di Pietro.\(^{10} \) The gradient of pressure is calculated explicitly by derivative of basis function as shown in equation (21).

\[
\nabla p_h = \sum_{i=1}^{N_i} \rho_h \frac{dq_h}{dx}
\]

(21)

Divergence of intermediate velocity uses integration by parts to include information from neighbor cells and boundary conditions, as shown in equation (22). This is to make the divergence term consistent with viscosity substep (equation (19)).

\[
(\nabla \cdot u_h, q_h)_{D_h} = - (\nabla q_h, u_h)_{D_h} + (q_h, u_h^i \cdot \hat{n})_{\partial D_h} \\
+ (q_h, u_h^e \cdot \hat{n})_{\partial D_h \cap \partial \Omega} + (q_h, g^a \cdot \hat{n})_{\partial D_h \cap \partial \Omega}
\]

(22)

The approximation of velocity on cell interfaces are evaluated by central flux

\[
\dot{u}_h^i = \{u_h^i\}
\]

Imposing the boundary conditions illustrated in equation (17), the final formulation of divergence term is derived as

\[
(\nabla \cdot u_h, q_h)_{D_h} = - (\nabla q_h, u_h)_{D_h} + (q_h, \{u_h^i\} \cdot \hat{n})_{\partial D_h} \\
+ (q_h, u_h^i \cdot \hat{n})_{\partial D_h \cap \partial \Omega} + (q_h, g^a \cdot \hat{n})_{\partial D_h \cap \partial \Omega}
\]

(23)

where \( g^a \) is the fixed value of velocity for Dirichlet boundaries.

To give a more direct illustration of the implementation, the corresponding pseudocode in Matlab style for two-dimensional case is provided as follows:

1 % compute gradient of pressure
2 pr = Dr * p; ps = Ds * p; 
3 gradPx = rx * pr + sx * ps; 
4 gradPy = ry * pr + sy * ps; 
5 % compute divergence of velocity
6 ur = Drw * Ux; us = Dsw * Ux; 
7 vr = Drw * Uy; vs = Dsw * Uy; 
8 divU = -(rx * ur + sx * us + ry * vr + ... sy * vs) + LIFT * (Fs rage * fluxU);

In the pseudocode, \( Dr \) and \( Ds \) indicate the derivative matrices in strong form, while \( Drw \) and \( Dsw \) are the derivative matrices in week form. \( rx, sx, ry, sy \) are the Jacobian coefficients. \( LIFT \) is the facial integration coefficients times inverse of mass matrix, \( Fscale \) is facial Jacobian coefficients, and variables with prefix \( flux \) are central flux for pressure or velocity. The calculation of pressure gradient uses only the information in local cell while velocity divergence needs one layer of neighbor cells.

To examine the small time stability of DG-CG method, it is necessary to find a reference. The method by Hesthaven and Warburton\(^{11} \) is a widely used pure DG method, but reported to be unstable for small time step in the work by Krantz et al.\(^5 \) In HW’s method, both pressure gradient and velocity divergence are calculated explicitly, as shown in the following equation

\[
(\nabla \cdot u_h, q_h)_{D_h} = - (\nabla q_h, u_h)_{D_h} + (q_h, \{u_h^i\} \cdot \hat{n})_{\partial D_h} \\
+ (q_h, u_h^i \cdot \hat{n})_{\partial D_h \cap \partial \Omega} + (q_h, g^a \cdot \hat{n})_{\partial D_h \cap \partial \Omega}
\]

(24)

The corresponding pseudo-code is listed below.

1 % compute gradient of pressure
2 pr = Dr * p; ps = Ds * p; 
3 gradPx = rx * pr + sx * ps; 
4 gradPy = ry * pr + sy * ps; 
5 % compute divergence of velocity
6 ur = Drw * Ux; us = Dsw * Ux; 
7 vr = Drw * Uy; vs = Dsw * Uy; 
8 divU = -(rx * ur + sx * us + ry * vr + ... sy * vs) + LIFT * (Fs rate * fluxU);

As argued by Fehn et al.,\(^9 \) the DG formulation of velocity divergence and pressure gradient term are the essential reasons for the small time instability. He recommends that integration by parts coupled with a suitable definition of boundary conditions should be used to keep stable and robust. The major difference
from HW’s method is performing integration by parts for both pressure gradient and velocity divergence, which are consistent with schemes used in pressure projection and velocity correction substeps. In Fehn’s method, the discretization of velocity divergence is almost the same as the one used in our DG-CG method (equation (22)). But the pressure gradient term is different, which is discretized with integration by parts. We can obtain the variational formulation as

\[
(\nabla p_h, q_h)_{D_b} = - (\nabla \cdot (q_h, p_h))_{D_b} + (q_h, \{ p_h \})_{\partial D} \quad \text{(25)}
\]

where \( g_p \) is the fixed value of pressure gradient for Neumann boundaries. The pseudo-code is listed below.

\[
\begin{align*}
1 & \text{ % compute gradient of pressure} \\
2 & pr = Drw^* p; ps = Dsw^* p; \\
3 & \text{gradPx} = -(rx.pr + sx.ps) + \ldots \\
4 & \text{LIFT} (\text{Fsacle.*fluxPx}); \\
5 & \text{gradPy} = -(ry.pr + sy.ps) + \ldots \\
6 & \text{LIFT} (\text{Fsacle.*fluxPy}); \\
7 & \text{ % compute divergence of velocity} \\
8 & ur = Drw^* ux; us = Dsw^* ux; \\
9 & vr = Drw^* uy; vs = Dsw^* uy; \\
10 & \text{DivU} = -(rx.ur + sx.us + ry.vr + \ldots) \\
11 & \text{sy.vs}) + \text{LIFT} (\text{Fsacle.*fluxU});
\end{align*}
\]

It should be noted that the pressure in DG-CG method is continuous across element interfaces. That is to say, applying integration by parts for pressure gradient term in DG-CG method will not bring much difference. In the view of pressure gradient and velocity divergence, the DG-CG method is nearly the same as the one used in Fehn et al.\(^9\) As Fehn claims that not applying integration by parts will lead to small time instability for pure DG method, one may ask whether the same phenomenon will happen in our mixed DG-CG method. Thus, we will also examine the DG-CG method using inconsistent velocity divergence scheme, as illustrated in equation (24). This method will be short for DG-CG-I in the later context.

### Numerical results

The solver is implemented within the open-source framework HopeFOAM,\(^13\) which aims at bringing high-order methods to official OpenFOAM based on second-order finite volume (Jasak et al.\(^{14}\) Yang et al.\(^{15}\) Wang et al.\(^{16}\)). The released HopeFOAM-0.1 provides high-order DGM, as well as high-order curved mesh, making it capable of achieving accurate space discretization. It inherits the user interfaces from OpenFOAM, including domain specified language, control dictionary, and boundary condition settings. Thus, developing high-order solver and cases within HopeFOAM is relatively convenient.

The linear system solver in HopeFOAM is based on PETSc,\(^{17}\) which is a library intended for solving large-scale scientific applications modeled by partial differential equations. It supports most of the commonly used linear solver and preconditioner for the sparse matrix. In the following context, the ilu preconditioned GMRES Krylov iterative method is used.

### Pearson vortex

Pearson vortex simulation is an unsteady INS problem and frequently used to demonstrate the small time step instability. The solution is constituted by making temporal derivatives in momentum equation equal with viscous term and non-linear convective term balance with pressure gradient. The analytical solution is given by

\[
\begin{align*}
\rho u &= - \sin(2\pi y)e^{-\sqrt{4\pi^2 t}} \\
\rho v &= \sin(2\pi x) e^{-\sqrt{4\pi^2 t}} \\
\rho p &= - \cos(2\pi x) \cos(2\pi y) e^{-\sqrt{8\pi^2 t}}
\end{align*}
\]

The computational domain is shown in Figure 1 enclosed by four inlet and four outlet boundaries. Dirichlet condition is enforced on inlets and Neumann condition on outlets for velocity, in which the boundary values are set by analytical solution. We choose viscosity \( \nu = 0.025 \) with corresponding \( Re = 40 \). The relative \( L^2 \) error is calculated at time \( T = 1.0 \) to evaluate the simulation results.

### Small time instability analysis

Since the small time instability is frequently reported on under-resolved cases, the simulation is performed on a mesh composed of 14 unstructured tetrahedral cells. Equal order polynomials are used for both pressure and velocity for all cases, with \( N = 1, N = 2, \) and \( N = 4 \). The time step is chosen...
to start from $2^{-1}$ to $2^{-16}$ decreased by a ratio of 0.5. To make a direct comparison, different methods discussed in the previous context are all performed on the test, including our DG-CG, DG-CG_I, the method used by Hesthaven and Warburton, and the one proposed by Fehn et al.

The results in Figure 2 reproduce the phenomenon reported for Hesthaven and Fehn’s method. For the HW’s method, instabilities exist in all cases and the relative error increases dramatically at small time step size. For Fehn’s method, integration by parts in pressure gradient and velocity divergence terms successfully remedy the instability for velocity. The relative error of velocity reduces to a stable value, which is dominated by spatial error.

The equal order DG-CG and DG-CG_I methods show great stability at small time step for both velocity and pressure at all polynomial orders. Similar relative $L_2$ error is obtained for these two methods for polynomial order 2 or 4. However, a significant increase of $L_2$ error of pressure is observed in Figure 2(b) for $N = 1$, while the increase in DG-CG is nearly not noticeable. This indicates that consistent velocity divergence scheme helps to obtain more accurate results for DG-CG method when the spatial resolution is extremely low. In our following context, the consistent velocity divergence scheme is used in our DG-CG method.

It should be noted that the $L^2$ error of pressure from Fehn’s method is larger than DG-CG method in extremely low spatial resolution, namely, $N = 1$ and $N = 2$. Fehn pointed out that the increase of $L^2$ error is an indication of inf-sup instabilities. Although equal order DG-CG method has not been proved to satisfy the inf-sup condition, its $L^2$ error is relatively smaller and bounded at small time step. Comparing to pure DG method, DG-CG method is observed to be much more accurate for the small time step.

**Eigenvalue spectrum analysis.** The eigenvalue spectrum analysis of the propagation matrix is performed to examine temporal stability of DG-CG method. Similar analysis has also been used in Leriche et al. for spectral element collocation schemes and in Fehn et al. for pure DG scheme. To perform this, the source term is discarded, harmonic boundary conditions are used for velocity, and the convective term is neglected. Then, the velocity $U_{n+1}$ can be derived by propagation matrix and $U_n$, which reads

$$U_{n+1} = AU_n$$

To compute $A$, INS equations are treated as a whole system and the pressure is eliminated as an auxiliary variable represented by velocity. The stable criteria for this system is given as

$$\max \lambda(A) \leq 1$$

The max eigenvalues as a function of time step size are calculated on the problem defined in the previous subsection. Figure 3 shows the results of DG-CG method and pure DG by Hesthaven. For our DG-CG method, all eigenvalues are less than 1 and within the stable region. When the time step size comes to the limit $\Delta t \to 0$, the temporal derivative is dominating and max eigenvalue tends to 1. Thus, we can draw a conclusion that DG-CG method has formed a stable temporal
iterative system. On the contrary, the max eigenvalue of HW’s pure DG method exceeds 1 at a small time step size, which indicates an unstable algebraic system. This is consistent with our findings from Figure 2.

**Spatial stability in low viscosity situation**

In this section, the spatial stability of equal order DG-CG method in low viscosity situation is analyzed. The behavior of equal-order DG-CG method at high Reynolds number is what we care about. In addition, how to keep stable is quite important when we try to apply the DG-CG method.

This test is carried on Poiseuille flow case. In the Pearson vortex case, the viscosity term is set to balance with the convective term especially. Therefore, solving of pressure does not depend on viscosity. Thus, Pearson vortex is not suitable to investigate the instability caused by low viscosity. In Poiseuille flow, the viscous equals with pressure gradient term in the final steady state. Notably, the flow pattern of Poiseuille flow is common in computational fluid dynamics (CFD). This is a steady INS problem in two dimensions with an analytical solution

\[
\begin{aligned}
    u &= U_{\text{max}} \left( 1 - \frac{y^2}{L^2} \right) \\
    \nu &= 0 \\
    p &= 2U_{\text{max}} \frac{\nu(L + x)}{L^2}
\end{aligned}
\]

The computational domain \( \Omega = [-L, L]^2 \) with \( L = 1 \). The velocity profile is a hyperbolic equation with max value \( U_{\text{max}} = 2m/s \) along the x axis. The inflow and up and down walls are set with Dirichlet boundary conditions, while outflow part is prescribed as Neumann boundary conditions. The flow starts from zero and ends at final time \( T = 30s \), which ensures the flow has reached a steady state. The simulations are performed with equal order DG-CG method with consistent velocity divergence scheme, and the temporal scheme is set to \( J = 2 \).

First, the simulations are carried out with \( \nu = 5 \times 10^{-4} \) to examine the instability under high Reynolds number. Results are shown in Figures 4 and 4. Simulation results of Poiseuille flow under different mesh scales, including (a) \( h = 1.0 \), (b) \( h = 0.5 \), and (c) \( h = 0.25 \). Cases are set with dynamic viscosity \( \nu = 5 \times 10^{-4} \) and polynomial order \( N = 3 \).
beginning of the simulation, which is dominated by convection. Time step size $\delta t = 10^{-3}$ and $\delta t = 10^{-5}$ get very close $L^2$ error. However, slight increase can still be found in cases with $\delta t = 10^{-5}$, which may relate to the inf-sup instability. When polynomial order $N = 4$, this increase seems more notable. Overall, the error introduced by a small time step size is bounded and do not affect the convergent. Otherwise, $\delta t = 10^{-5}$ is too small for the simulation, and setting the step size under CFL condition seems to be a good choice.

At last, we can conclude that equal order DG-CG method may fail for high Reynolds number cases under low spatial resolution. But with sufficient mesh resolution and polynomial order, coupling with proper time step size (CFL condition), accurate simulation results can still be achieved.

### 3D Beltrami flow and convergence test

This 3D case is developed by Ethier and Steinman\textsuperscript{22} for benchmarking purposes, and is a counterpart of 2D Taylor vortex problem. This problem is defined on domain $\Omega = [-1,1]^3$. The exact solution of velocity and pressure is given as

$$
\begin{align*}
\mathbf{u} &= - a [ e^{ay} \sin(ax + dz) + e^{az} \cos(ax + dy) ] e^{-ndt}\sin(x) \\
\mathbf{v} &= - a [ e^{ay} \sin(ax + dz) + e^{az} \cos(ax + dy) ] e^{-ndt}\sin(y) \\
w &= - \frac{a}{2} [ e^{2ax} + 2 \sin(ax + dy) \cos(az + dx) e^{(y+z)} \\
+ e^{2ay} + 2 \sin(ay + dz) \cos(ax + dy) e^{(x+z)} ] e^{-ndt}
\end{align*}
$$

(30)

Here, the parameters are defined as dynamic viscosity $\nu = 1.0$, $a = \pi/4$, and $d = \pi/2$. To examine the spatial convergence, uniform structural meshes with $4^3$, $8^3$, and $16^3$ elements are used.

The point-wise error in $L_2$ norm of velocity and pressure in $T = 1$ s is gathered in Figure 6. The slope between error and grid scale indicates the convergence order. Both velocity and pressure have reached theoretical convergence rate $h^N$ or even higher, compatible with pure DG discretization results mentioned in Hesthaven and Warburton.\textsuperscript{11}

To reduce the influence of error introduced by spatial discretization, the polynomial order is set with $N = 5$. A reverse Cuthill–McKee reorder method preconditioned LU-factor direct linear solver is chosen to solve the linear system. Temporal integration schemes with $J = 1$ and $J = 2$ are performed. Figure 7 shows the error in $L_2$ norm of velocity and pressure. For $J = 1$ case, the convergence rate is $\Delta t$, while $J = 2$ scheme reaches optimal convergence rate of $\Delta t^2$. It should be noted that in the case of $J = 2$, the error does not continue to reduce when $\Delta t < 2 \times 10^{-4}$ since it is dominated by spacial accuracy.
Table 1. Relative $L^2$ error of pressure at $t = 30$ in the simulation of Poiseuille flow, influences of polynomial orders, time step sizes, dynamic viscosities, and mesh scales are compared.

| $N$ | $h = 0.5$ | | $h = 0.25$ | | $h = 0.125$ | |
|-----|-----------|-----|-----------|-----|-----------|-----|
|     | $dt = 10^{-1}$ | $dt = 10^{-3}$ | $dt = 10^{-5}$ | $dt = 10^{-1}$ | $dt = 10^{-3}$ | $dt = 10^{-5}$ |
| $\nu = 0.05$ | | | | | | |
| 1   | 2.1201    | 3.224 | 3.2331 | 0.4028 | 0.7083 | 0.7249 | NAN | 0.1832 | 0.1964 |
| 2   | 0.1257    | 0.0943 | 0.0952 | NAN | 0.0017 | 0.0018 | NAN | 1.54×$10^{-4}$ | 1.59×$10^{-4}$ |
| 3   | NAN       | 0.032 | 0.0323 | 6.2×$10^{-12}$ | 1.8×$10^{-10}$ | 2.8×$10^{-10}$ | NAN | 1.0×$10^{-11}$ | 2.9×$10^{-10}$ |
| 4   | NAN       | 6.9×$10^{-10}$ | 1.52×$10^{-8}$ | 8.2×$10^{-12}$ |
| $\nu = 0.0005$ | | | | | | |
| 1   | NAN       | 243.06 | 242.88 | NAN | 243.06 | 242.88 | NAN | 1.1459 |
| 2   | NAN       | 26.706 | 26.673 | NAN | 3.0002 | 3.023 | 3.0002 | 3.023 |
| 3   | NAN       | 4.2288 | 4.2224 | NAN | 0.2054 | 0.2027 | 0.2054 | 0.2027 |
| 4   | NAN       | 6.9×$10^{-10}$ | 1.52×$10^{-8}$ | 3.0×$10^{-10}$ | 9.02×$10^{-8}$ |

NAN means this case cannot get convergent in the simulation.

Figure 6. Point-wise error in $L_2$ norm of (a) velocity and (b) pressure. Slope indicates the space convergence rate.

Figure 7. Results of 3D Beltrami flow simulation: (a) Temporal convergence rate of velocity and pressure using $J=1$ and $J=2$ and (b) Snapshot of velocity magnitude at $T=1$s.
Here, we consider two-dimensional lid-driven cavity flow with relative high Reynolds number. The computational domain is a unit square. Upper bound (the lid) is enforced with a constant velocity $u = (1, 0)$, while no-slip boundary conditions are set on other sides. The dynamic viscosity is configured with $\nu = 10^{-3}$, yielding a Reynolds number for the flow of $Re = 1000$. Fifth-order DG-CG space discretization and $J_2$ time integration scheme are applied on a triangular mesh with 3606 cells. The spatial resolution is sufficient to avoid low viscosity related instability and obtain accurate results. The simulation advances until reaching a steady state.

Figure 8(a) shows the streamline of steady-state cavity. Two vortices appear at the bottom corner, which is consistent with the literature. The profile of $x$ and $y$ components of velocity along vertical and horizontal center line is drawn in Figure 8(b). The simulation results (the lines) fit well with the reference value (dots). With special care on the mesh resolution and discretization order, DG-CG method can simulate relative high Reynolds problem and keep stable.

### Cavity flow

Here, we consider two-dimensional lid-driven cavity flow with relative high Reynolds number. The computational domain is a unit square. Upper bound (the lid) is enforced with a constant velocity $u = (1, 0)$, while no-slip boundary conditions are set on other sides. The dynamic viscosity is configured with $\nu = 10^{-3}$, yielding a Reynolds number for the flow of $Re = 1000$. Fifth-order DG-CG space discretization and $J_2$ time integration scheme are applied on a triangular mesh with 3606 cells. The spatial resolution is sufficient to avoid low viscosity related instability and obtain accurate results. The simulation advances until reaching a steady state.

Figure 8(a) shows the streamline of steady-state cavity. Two vortices appear at the bottom corner, which is consistent with the literature. The profile of $x$ and $y$ components of velocity along vertical and horizontal center line is drawn in Figure 8(b). The simulation results (the lines) fit well with the reference value (dots). With special care on the mesh resolution and discretization order, DG-CG method can simulate relative high Reynolds problem and keep stable.

### 3D laminar flow past cylinder

This is a very challenging case with unsteady time-varying inflow profile. As a simplification of practical applications, lots of papers have used this case to verify the 3D simulation accuracy of CFD code. For the benchmark problem, a cylinder locates at the middle line of a channel, while no-slip boundary conditions are employed on the walls. The detailed geometry configuration can refer to Schäfer et al.\textsuperscript{24} and John\textsuperscript{25} The inlet velocity is set with

$$U(x, y, z) = \frac{16U_m y \sin \left(\frac{\pi x}{4}\right)(H-y)(H-z)}{H^4}$$

(31)

where $U_m = 2.25m/s$. The simulation starts with zero inflow at $t = 0$ and finishes at $t = 8s$. Guided by the findings in “Pearson vortex” and “Spatial stability in low viscosity situation” sections, the simulation is successfully carried out by DG-CG method with up to fifth-polynomial order. Although using a coarse mesh composed of 50,000 tetrahedral, results are nearly converged at third order due to the high-order method. Special attentions are paid on the cylinder boundary, where high-order curved elements are employed on the nearest cells. Figure 9 depicts the velocity field at $t = 2.4s$ using DG-CG method with polynomial order $N = 3$, which is indistinguishable from results with $N = 4, 5$.

The maximum drag coefficient and minimum lift coefficient are major comparing criteria for this case. Results obtained by DG-CG and pure DG method under different polynomial orders are listed in Table 2. Accurate results are gained under both methods according to the reference values in Bayraktar et al.\textsuperscript{26} It is worth to mention that results by our high order methods with $N = 3$ are as accurate as OpenFOAM (second-order FVM) on a mesh with 3 million cells. DG-CG method uses about 33% degree of freedom (Dof) than FVM to reach similar accuracy.

---

**Figure 8.** Results of lid driven cavity flow on mesh with 3606 cells on $Re = 1000$ and $N=5$: (a) stream line and (b) profile of velocity components along vertical and horizontal center line. The lines in figure indicate simulation results while dots is reference value.
Performance evaluation

DG method is preferred for its high-order convergence rate, stability, and high parallel efficiency while coming along with the disadvantages of extraordinary computation and memory overhead. This has been a major factor in obstructing its spread and application on complex problems. In the splitting method, solving of pressure takes up the major part of computational overhead. The solver with discontinuous velocity and continuous pressure has been examined to achieve high-order spacial convergence rate in previous sections. Moreover, pressure discrete by CG has the potential to be more efficient comparing to the discontinuous method. In this section, the performance of equal order DG-CG and pure DG method are analyzed under ilu preconditioned GMRES solver with the Pearson vortex case. The test ends at final time $T = 1.0s$ with time step set by CFL condition. Although it is a very simple case, results can indicate the general tendency and provide references for other cases.

The performance statistics of DG-CG and pure DG method are collected in Table 3, including the iterative number ($n_{it}$), computational time, and the number of degree of freedom ($nDof$) for each substep. Notation “pressure-cg” indicates the substep of solving pressure in DG-CG method, “pressure-dg” is a similar substep in pure DG method, and “velocity-dg” is the substep of solving velocity correction in both methods. Similar to the LU direct solver cases, DG-CG method has fewer Dofs and need less iterative steps to get convergence comparing to pure DG method. The speedup reaches up to 4 times at 896 test case while 6 times at 3584 test case, which is more significant at high-order scenario.

### Table 2. Maximum drag coefficient and minimum lift coefficient obtained by DG-CG and pure DG method under different polynomial order.

| $N$ | DG-CG | Pure DG |
|-----|--------|---------|
|     | $C_{D_{\text{max}}}$ | $C_{L_{\text{min}}}$ | $C_{D_{\text{max}}}$ | $C_{L_{\text{min}}}$ |
| 2   | 3.2305 | -0.009683 | 3.2131 | -0.009675 |
| 3   | 3.2877 | -0.010897 | 3.2576 | -0.010782 |
| 4   | 3.2945 | -0.010999 | 3.2859 | -0.010877 |
| 5   | 3.2971 | -0.011002 | 3.2901 | -0.010929 |

DG: discontinuous Galerkin; CG: continuous Galerkin.

### Table 3. The ilu preconditioned GMRES solver profiling for DG-CG method and pure DG method.

| $N$ | pressure-cg | pressure-dg | velocity-dg |
|-----|-------------|-------------|-------------|
|     | nDof | n_{it} | Time (s) | nDof | n_{it} | Time (s) | nDof | n_{it} | Time (s) |
| 896 test grid | | | | | | | | |
| 2   | 1857 | 14 | 9.88 | 5376 | 18 | 28.52 | 5376 | 7 | 18.68 |
| 3   | 4129 | 16 | 42.58 | 8960 | 29 | 148 | 8960 | 8 | 58.16 |
| 4   | 7297 | 19 | 133.4 | 13,440 | 36 | 1028 | 13,440 | 10 | 172.3 |
| 5   | 11,361 | 22 | 365.5 | 18,816 | 43 | 2745 | 18,816 | 10 | 354.9 |
| 3584 test grid | | | | | | | | |
| 2   | 7297 | 23 | 110.3 | 21,504 | 37 | 470 | 21,504 | 8 | 121.7 |
| 3   | 16,321 | 29 | 526.3 | 35,840 | 57 | 5570 | 35,840 | 9 | 462 |
| 4   | 28,929 | 36 | 1916 | 53,760 | 71 | 18195 | 53,760 | 11 | 1374 |
| 5   | 45,121 | 41 | 5612 | 75,264 | 86 | 50099 | 75,264 | 11 | 2875 |

DG: discontinuous Galerkin; CG: continuous Galerkin; nDof: number of degree of freedom.
Conclusion

In this article, we analyzed in detail the stability of the splitting method with discontinuous velocity and continuous pressure. Results are also compared to pure DG method by Hesthaven and Warburton\textsuperscript{11} and Fehn et al.\textsuperscript{9} For the small time step size problem, we first reproduce the instability with HW’s method on the simulation of Pearson vortex, while our DG-CG method remains stable at the same settings. As the time step size decreases continuously, the relative $L^2$ error of pressure increase litter. In addition, the consistent velocity divergence discretization scheme helps to get more accurate results, especially for low polynomial order cases. Second, we demonstrate the instabilities of equal order DG-CG method in high Reynolds number cases. Through comprehensive experiments of Poiseuille flow covering various viscosities, time step sizes, mesh scales, and polynomial orders, we numerically show that sufficient polynomial order and mesh helps remedy the instability and obtain accurate results for high Reynolds flow.

In the work of Botti and Di Pietro,\textsuperscript{10} the accuracy and performance of high-order polynomial are not discussed, where the highest pair is DG(3)-CG(2). In the stability analysis, we numerically show that higher polynomial order can lead to a more robust scheme in low viscosity situations. By the example of 3D Beltrami flow, up to sixth-order spatial convergence rate and second-order temporal convergence rate have been demonstrated. The cavity case is run on relative high Reynolds number flow condition, which confirms the stability of DG-CG method. At last, we check out the performance of DG-CG method at high polynomial order with pure DG. Results show that DG-CG method is more efficient than pure DG method, to be specific, a speedup of 2.5 to 6 times on ilu preconditioned GMRES solver, which meets our initial expectation to develop mixed DG-CG solver.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was funded through the National Natural Science Foundation of China (Grant No. 61872380, 61802426), National Key Research and Development Program of China (No. 2016YFB0201301), and Science Challenge Project (No. TZ2016002).

ORCID iD

Liyang Xu https://orcid.org/0000-0003-0932-643X

References

1. Guermond JL, Minev P and Shen J. An overview of projection methods for incompressible flows. Comp Meth Appl Mech Eng 2006; 195: 6011–6045.
2. Shahbazi K, Fischer PF and Ethier CR. A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations. J Comput Phys 2007; 222: 391–407.
3. Ferrer E and Willden R. A high order discontinuous Galerkin finite element solver for the incompressible Navier–Stokes equations. Comput Fluid 2011; 46: 224–230.
4. Ferrer E, Moxey D, Willden R, et al. Stability of projection methods for incompressible flows using high order pressure-velocity pairs of same degree: continuous and discontinuous Galerkin formulations. Commun Comput Phys 2014; 16: 817–840.
5. Krank B, Fehn N, Wall WA, et al. A high-order semi-explicit discontinuous Galerkin solver for 3D incompressible flow with application to DNS and LES of turbulent channel flow. J Comput Phys 2017; 348: 634–659.
6. Joshi SM, Di Giacomo PJ, Steinmoeller DT, et al. A post-processing technique for stabilizing the discontinuous pressure projection operator in marginally-resolved incompressible inviscid flow. Comput Fluid 2016; 139: 120–129.
7. Emamy N, Kummer F, Mrosek M, et al. Implicit-explicit and explicit projection schemes for the unsteady incompressible Navier–Stokes equations using a high-order dG method. Comput Fluid 2017; 154: 285–295.
8. Steinmoeller DT, Stastna M and Lamb K. A short note on the discontinuous Galerkin discretization of the pressure projection operator in incompressible flow. J Comput Phys 2013; 251: 480–486.
9. Fehn N, Wall WA and Kronbichler M. On the stability of projection methods for the incompressible Navier–Stokes equations based on high-order discontinuous Galerkin discretizations. J Comput Phys 2017; 351: 392–421.
10. Botti L and Di Pietro DA. A pressure-correction scheme for convection-dominated incompressible flows with discontinuous velocity and continuous pressure. J Comput Phys 2011; 230: 572–585.
11. Hesthaven JS and Warburton T. Nodal discontinuous Galerkin methods, Vol. 54. 2008, http://link.springer.com/10.1007/978-0-387-72067-8.
12. Karniadakis GE, Israeli M and Orszag SA. High-order spatial discretizations for convection-dominated incompressible flows. J Comput Phys 2006; 195: 6011–6045.
13. HopeFOAM EG. HopeFOAM-0.1, 2017, https://github.com/HopeFOAM/HopeFOAM.
14. Jasak H, Jemcov A and Tukovic Z. OpenFOAM: A C++ library for complex physics simulations. In: International workshop on coupled methods in numerical dynamics, Dubrovnik, Croatia, 9–21 September 2007, pp.1–20.
15. Yang WJ, Yi W, Ren XG, et al. Toward large scale parallel computer simulation of viscoelastic fluid flow: a study of benchmark flow problems. J Non-Newton Fluid Mech 2015; 222: 82–95.
16. Wang Q, Xu XH, Ye S, et al. WI-USHER: a grid-based parallel algorithm for particle insertion in hybrid
atomistic-continuum method. *Adv Mech Eng*. Epub ahead of print 13 February 2017. DOI: 10.1177/16871407691895.

17. Balay S, Abhyankar S, Adams MF, et al. PETSc web page, 2016, http://www.mcs.anl.gov/petsc

18. Girault V and Raviart PA. *Finite element methods for Navier-Stokes equations: theory and algorithms*, Vol. 5. Berlin: Springer Science & Business Media, 2012.

19. Leriche E, Perchat E, Labrosse G, et al. Numerical evaluation of the accuracy and stability properties of high-order direct stokes solvers with or without temporal splitting. *J Sci Comput* 2006; 26: 25–43.

20. Lederer PL, Linke A, Merdon C, et al. Divergence-free reconstruction operators for pressure-robust stokes discretizations with continuous pressure finite elements. *SIAM J Numer Anal* 2017; 55: 1291–1314.

21. Linke A. On the role of the helmholtz decomposition in mixed methods for incompressible flows and a new variational crime. *Comput Meth Appl Mech Eng* 2014; 268: 782–800.

22. Ethier CR and Steinman D. Exact fully 3D Navier–Stokes solutions for benchmarking. *Int J Num Meth Fluid* 1994; 19: 369–375.

23. Erturk E, Corke TC and Gökçöl C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int J Num Meth Fluid* 2005; 48: 747–774.

24. Schäfer M, Turek S, Durst F, et al. Benchmark computations of laminar flow around a cylinder. In: Hirschel EH (ed.) *Flow simulation with high-performance computers II*. Cham: Springer, 1996, pp.547–566.

25. John V. On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3D flow around a cylinder. *Int J Num Meth Fluid* 2006; 50: 845–862.

26. Bayraktar E, Mierka O and Turek S. Benchmark computations of 3D laminar flow around a cylinder with CFX, openFoam and FeatFlow. *Int J Comput Sci Eng* 2012; 7: 253–266.