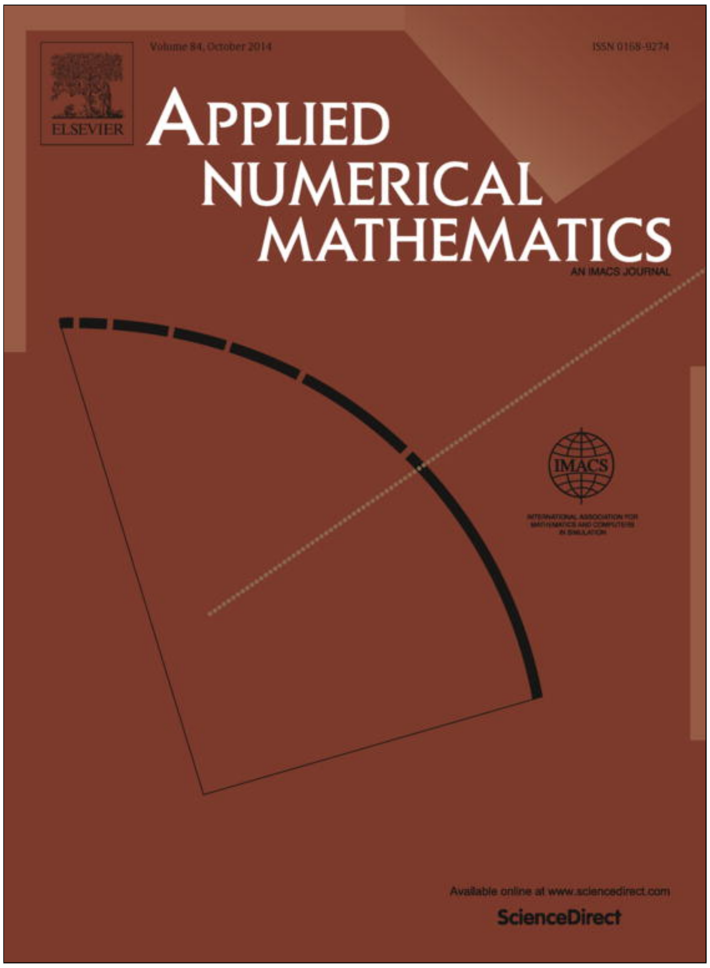
Volume 84, October 2014
ISSN 0168-9274
ELSEVIER
APPLIED
NUMERICAL
MATHEMATICS
AN IMACS JOURNAL
IMACS
INTERNATIONAL ASSOCIATION FOR MATHEMATICS AND COMPUTERS IN SIMULATION
Available online at www.sciencedirect.com
ScienceDirect

Applied Numerical Mathematics 84 (2014) 66–79

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Parallel spectral-element direction splitting method for incompressible Navier–Stokes equations [☆]

Lizhen Chen [a,c], Jie Shen [a,b,*], Chuanju Xu [a], Li-Shi Luo [c]

[a] Fujian Provincial Key Laboratory on Mathematical Modeling and Scientific Computing and School of Mathematical Sciences, Xiamen University, 361005 Xiamen, China
[b] Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA
[c] Beijing Computational Science Research Center, 100084, Beijing, China

ARTICLE INFO

Article history:
Received 25 February 2013
Received in revised form 4 October 2013
Accepted 1 May 2014
Available online 14 June 2014

Keywords:
Navier–Stokes equations
Direction splitting
Spectral-element methods
Parallel algorithm

ABSTRACT

An efficient parallel algorithm for the time dependent incompressible Navier–Stokes equations is developed in this paper. The time discretization is based on a direction splitting method which only requires solving a sequence of one-dimensional Poisson type equations at each time step. Then, a spectral-element method is used to approximate these one-dimensional problems. A Schur-complement approach is used to decouple the computation of interface nodes from that of interior nodes, allowing an efficient parallel implementation. The unconditional stability of the full discretized scheme is rigorously proved for the two-dimensional case. Numerical results are presented to show that this algorithm retains the same order of accuracy as a usual spectral-element projection type schemes but it is much more efficient, particularly on massively parallel computers.

1. Introduction

The alternating direction method was developed in the 1950s for solving elliptic and parabolic equations (cf. [14]). It received new attention recently due to its suitability for parallel computing. In particular, Guermond and Minev [3] (see also [7]) proposed a direction splitting method which combines the pressure-stabilization method (cf. [17,8]) and the alternating direction technique for the time discretization of the incompressible Navier–Stokes equations. This scheme leads to a sequence of one-dimensional problems at each time step, making it very efficient and amenable for parallel computing. In [5] and [6], the authors implemented this direction splitting scheme with a finite difference method in space, and showed that the algorithm can achieve a high level of parallelism. In [1], the authors developed a direction splitting scheme with a hybrid (one domain) spectral method in space, and proved that the scheme is unconditional stable. However, the (one domain) spectral method is not suitable for massive parallel computers and for more general domains, so its applicability to large scale simulations is quite limited.

The main purpose of this paper is to develop a stable direction splitting scheme with a spectral-element discretization in space that allows for efficient parallel implementation and is applicable for more general domains. The construction of the stable full discretized scheme is by no means a trivial extension of the spectral method in [1]. In order to prove the unconditional stability, we have to construct a new spectral-element method for the velocity while using the classical spectral-element method for the pressure.

[☆] This research is partially supported by National Natural Science Foundation of China (Grant numbers 11071203, 11371298 and 91130002).

[*] Corresponding author at: Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA.

http://dx.doi.org/10.1016/j.apnum.2014.05.010
0168-9274/© 2014 IMACS. Published by Elsevier B.V. All rights reserved.

The spectral-element method (cf. [13]) combines advantages of the high-order accuracy of the spectral methods and the geometrical flexibility of the finite element methods. It has become a major tool in scientific computing, particularly for computational fluid dynamics (see, for instance, [2,10] and the references therein). However, well-developed parallel spectral-element codes such as NEK5000 (cf. [18]) and NekTar (cf. [9]) do not really take advantage of the situations where the domains are simple separable domains. We note, however, that an efficient parallel spectral-element algorithm for separable elliptic problems is developed in [11]. The algorithm is based on a matrix decomposition approach which reduces multi-dimensional problems to a sequence of one-dimensional problems. However, it requires the computation and storing of a very large eigenmatrix which limits the size of problems that it can handle. The method we propose in this paper will lead to, at each time step, a sequence of one dimensional problems that can be solved more efficiently in parallel and does not require storing a large eigenmatrix as required in [11].

The outline of the paper is as follows. In the next section we describe in detail the spectral-element direction splitting scheme. Then, we present a Schur-compliment approach for solving one-dimensional problems, and describe how to implement the whole algorithm in parallel in Section 3. We prove in Section 4 the unconditional stability of the full discretized scheme. Finally, we present some numerical results and discussions in Section 5.

2. Spectral-element direction splitting scheme

We consider the time-dependent Navier–Stokes equations:

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} - \nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{f}, & \text{in } \Omega\times(0,T], \\ \nabla\cdot\boldsymbol{u} = 0, & \text{in } \Omega\times[0,T], \\ \boldsymbol{u}|_{\partial\Omega} = \boldsymbol{0}, & \text{in } [0,T], \\ \boldsymbol{u}|_{t=0} = \boldsymbol{u}_0, & \text{in } \Omega, \end{cases} \tag{2.1}$$

where ν is the kinematic viscosity coefficient, $\boldsymbol{u}$ and p are the velocity vector and the pressure, respectively. For the sake of simplicity, we shall restrict our attention in this paper to $\Omega = (0,R)^d$, $d = 2,3$, although the method can be easily extended to other separable domains or more general domains consisting of unions of separable domains.

2.1. The direction splitting scheme

We consider the following direction splitting scheme proposed in [3].

Setting $\boldsymbol{u}^0 = \boldsymbol{u}|_{t=0}$, $p^{-\frac{1}{2}} = p|_{t=0}$ and $\phi^{-\frac{1}{2}} = 0$. Then, for $n \ge 0$, we look for $(\boldsymbol{\eta}^{n+1}, \boldsymbol{u}^{n+1}, p^{n+\frac{1}{2}})$ as follows:

- **Predictor for velocity.**

$$\frac{\boldsymbol{\xi}^{n+1} - \boldsymbol{u}^n}{\Delta t} - \nu\Delta\boldsymbol{u}^n + \nabla p^{*,n+\frac{1}{2}} + \mathbf{NL}(\boldsymbol{u}^n, \boldsymbol{u}^{n-1}) = \boldsymbol{f}^{n+\frac{1}{2}}, \quad \boldsymbol{\xi}^{n+1}|_{\partial\Omega} = \boldsymbol{0}, \tag{2.2}$$

where

$$\begin{aligned} & p^{*,n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n-\frac{1}{2}}, \\ & \mathbf{NL}(\boldsymbol{u}^n, \boldsymbol{u}^{n-1}) = \frac{3}{2}(\boldsymbol{u}^n\cdot\nabla)\boldsymbol{u}^n - \frac{1}{2}(\boldsymbol{u}^{n-1}\cdot\nabla)\boldsymbol{u}^{n-1}; \end{aligned} \tag{2.3}$$

- **Direction splitting for velocity.**

$$\begin{aligned} & \frac{\boldsymbol{\eta}^{n+1} - \boldsymbol{\xi}^{n+1}}{\Delta t} - \frac{1}{2}\nu\partial_{xx}(\boldsymbol{\eta}^{n+1} - \boldsymbol{u}^n) = \boldsymbol{0}, \quad \boldsymbol{\eta}^{n+1}|_{x=0,R} = \boldsymbol{0}, \\ & \frac{\boldsymbol{\zeta}^{n+1} - \boldsymbol{\eta}^{n+1}}{\Delta t} - \frac{1}{2}\nu\partial_{yy}(\boldsymbol{\zeta}^{n+1} - \boldsymbol{u}^n) = \boldsymbol{0}, \quad \boldsymbol{\zeta}^{n+1}|_{y=0,R} = \boldsymbol{0}, \\ & \frac{\boldsymbol{u}^{n+1} - \boldsymbol{\zeta}^{n+1}}{\Delta t} - \frac{1}{2}\nu\partial_{zz}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) = \boldsymbol{0}, \quad \boldsymbol{u}^{n+1}|_{z=0,R} = \boldsymbol{0}; \end{aligned} \tag{2.4}$$

- **Direction splitting for pressure.**

$$\begin{aligned} & \psi^{n+\frac{1}{2}} - \partial_{xx}\psi^{n+\frac{1}{2}} = -\frac{\nabla\cdot\boldsymbol{u}^{n+1}}{\Delta t}, \quad \partial_x\psi^{n+\frac{1}{2}}|_{x=0,R} = 0, \\ & \varphi^{n+\frac{1}{2}} - \partial_{yy}\varphi^{n+\frac{1}{2}} = \psi^{n+\frac{1}{2}}, \quad \partial_y\varphi^{n+\frac{1}{2}}|_{y=0,R} = 0, \\ & \phi^{n+\frac{1}{2}} - \partial_{zz}\phi^{n+\frac{1}{2}} = \varphi^{n+\frac{1}{2}}, \quad \partial_z\phi^{n+\frac{1}{2}}|_{z=0,R} = 0; \end{aligned} \tag{2.5}$$

- **Pressure update.**

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n+\frac{1}{2}} - \chi\nu\nabla\cdot\left(\frac{1}{2}\left(\boldsymbol{u}^{n+1}+\boldsymbol{u}^{n}\right)\right), \tag{2.6}$$

where the parameter $\chi \in (0,1)$. The two-dimensional version of the algorithm is obtained by skipping the last substeps in (2.4) and (2.5) and setting $\boldsymbol{u}^{n+1} = \boldsymbol{\eta}^{n+1}$, $\phi^{n+\frac{1}{2}} = \varphi^{n+\frac{1}{2}}$, respectively.

We observe that each of the substeps in (2.4) and (2.5) are a sequence of one-dimensional elliptic problems. Therefore, the overall scheme is extremely efficient.

The convergence of the above scheme, in the absence of nonlinear terms, was studied in [7].

2.2. Two spectral-element methods for one-dimensional problems

We observe from the direction splitting scheme above that we have to solve, repeatedly at each time, the following one-dimensional Poisson type equation:

$$\alpha u - \beta\partial_x^2 u = f, \quad x \in \Lambda = (0,R) \tag{2.7}$$

with $u(0) = u(R) = 0$ or $u'(0) = u'(R) = 0$.

We partition the interval Λ into K non-overlapping subdomains $\Lambda^k = (a_{k-1}, a_k)$, $k = 1, 2, \cdots, K$, where $0 = a_0 < a_1 < \cdots < a_K = R$, and denote $h_k = a_k - a_{k-1}$.

Let P_N be the space of polynomials of degree less or equal than N. We define the spectral-element spaces

$$\begin{aligned} X_{\mathcal{N}} &= \left\{u_{\mathcal{N}} \in C(\Lambda) : u_{\mathcal{N}}\,|_{\Lambda^k} \in P_N\left(\Lambda^k\right),\ k = 1, 2, \cdots, K\right\}, \\ X_{\mathcal{N}}^0 &= \left\{u_{\mathcal{N}} \in X_{\mathcal{N}} : u_{\mathcal{N}}(0) = u_{\mathcal{N}}(R) = 0\right\}, \\ \mathbb{P}_N &= (P_N)^d, \qquad \mathbb{X}_{\mathcal{N}} = (X_{\mathcal{N}})^d, \qquad \mathbb{X}_{\mathcal{N}}^0 = \left(X_{\mathcal{N}}^0\right)^d, \end{aligned} \tag{2.8}$$

where $\mathcal{N}$ denotes the integer pair (N, K).

Let $\{x_p, \omega_p\}_{0\le p\le N}$ be the set of Legendre–Gauss–Lobatto (LGL) points and weights on $\bar{I} = [-1, 1]$, and $\{x_p^k, \omega_p^k\}_{0\le p\le N}$ be the set of mapped collocation points in Λ^k by $\{x_p, \omega_p\}_{0\le p\le N}$.

Let $(\cdot,\cdot)$ be the usual inner product in $L^2(\Lambda)$, and let $(\cdot,\cdot)_{\mathcal{N},\Lambda} = \sum_{k=1}^K (\cdot,\cdot)_{\mathcal{N},\Lambda^k}$, where $(\cdot,\cdot)_{\mathcal{N},\Lambda^k}$ is the discrete inner product associated with the LGL quadrature on Λ^k:

$$(\varphi, \psi)_{\mathcal{N},\Lambda^k} = \sum_{p=0}^{N} \varphi\left(x_p^k\right)\psi\left(x_p^k\right)\omega_p^k, \quad \forall\varphi, \psi \in X_{\mathcal{N}}. \tag{2.9}$$

Then, the classical spectral-element method for (2.7) with $u(0) = u(R) = 0$ is as follows: Find $u_{\mathcal{N}} \in X_{\mathcal{N}}^0$, such that

$$b(u_{\mathcal{N}}, v_{\mathcal{N}}) := \alpha(u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda} + \beta(\partial_x u_{\mathcal{N}}, \partial_x v_{\mathcal{N}})_{\mathcal{N},\Lambda} = (f, v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad \forall v_{\mathcal{N}} \in X_{\mathcal{N}}^0. \tag{2.10}$$

On the other hand, the classical spectral-element method for (2.7) with $u'(0) = u'(R) = 0$ reads: Find $u_{\mathcal{N}} \in X_{\mathcal{N}}$, such that

$$b(u_{\mathcal{N}}, v_{\mathcal{N}}) := \alpha(u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda} + \beta(\partial_x u_{\mathcal{N}}, \partial_x v_{\mathcal{N}})_{\mathcal{N},\Lambda} = (f, v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad \forall v_{\mathcal{N}} \in X_{\mathcal{N}}. \tag{2.11}$$

Next, we describe a new spectral-element method for (2.7). The new method makes use an extension of the discrete derivative defined in the piecewise polynomial spaces, which will be crucial for the proof of the stability of the proposed scheme in Section 4. We start by introducing a modified derivative, which will be used to define the new spectral-element method. The modified derivative, denoted by $\tilde{\partial}_x$, is defined as follows: for any $v_{\mathcal{N}} \in X_{\mathcal{N}}$, $\tilde{\partial}_x v_{\mathcal{N}}$ is the piecewise polynomial determined by, for all $k = 1, \ldots, K$,

$$\tilde{\partial}_x v_{\mathcal{N}}\left(x_p^k\right) = \begin{cases} \left[\partial_x v_{\mathcal{N}}\left(x_N^{k-1}\right)\omega_N^{k-1} + \partial_x v_{\mathcal{N}}\left(x_0^k\right)\omega_0^k\right] / \left(\omega_N^{k-1} + \omega_0^k\right), & p = 0, \\ \partial_x v_{\mathcal{N}}\left(x_p^k\right), & p = 1, \ldots, N-1, \\ \left[\partial_x v_{\mathcal{N}}\left(x_N^k\right)\omega_N^k + \partial_x v_{\mathcal{N}}\left(x_0^{k+1}\right)\omega_0^{k+1}\right] / \left(\omega_N^k + \omega_0^{k+1}\right), & p = N, \end{cases} \tag{2.12}$$

with an obvious modification when $k = 1$ and $k = K$. In the above, the traditional derivative ∂_x is defined on each element and discontinuous across the element interface points. On the other hand, the modified derivative $\tilde{\partial}_x$ at an element interface point is defined as a weighted average of the two values from the neighboring elements, and is continuous across the element interface.

This new derivative has the following obvious properties:

$$\tilde{\partial}_x v_{\mathcal{N}} \in X_{\mathcal{N}}, \quad \text{if } v_{\mathcal{N}} \in X_{\mathcal{N}}. \tag{2.13}$$

$$\tilde{\partial}_x v_{\mathcal{N}} \equiv \partial_x v_{\mathcal{N}}, \quad \text{if } v_{\mathcal{N}} \in C^1(\Lambda) \cap X_{\mathcal{N}}. \tag{2.14}$$

Some additional properties of $\tilde{\partial}_x$ are given below.

Lemma 2.1. *The following identities hold:*

$$(\tilde{\partial}_x u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda} = (\partial_x u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad \forall u_{\mathcal{N}}, v_{\mathcal{N}} \in X_{\mathcal{N}}; \tag{2.15}$$

$$(\tilde{\partial}_x u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda} = -(u_{\mathcal{N}}, \tilde{\partial}_x v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad \forall u_{\mathcal{N}}, v_{\mathcal{N}} \in X^0_{\mathcal{N}}; \tag{2.16}$$

$$-\left(\tilde{\partial}^2_x u_{\mathcal{N}}, v_{\mathcal{N}}\right)_{\mathcal{N},\Lambda} = (\tilde{\partial}_x u_{\mathcal{N}}, \tilde{\partial}_x v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad \forall u_{\mathcal{N}}, v_{\mathcal{N}} \in X^0_{\mathcal{N}}, \tag{2.17}$$

where $\tilde{\partial}^2_x u_N := \tilde{\partial}_x(\tilde{\partial}_x u_N)$.

Proof. (2.15) can be proved by a direct calculation.

For (2.16), by using (2.15) and the exactness of the Legendre–Gauss–Lobatto quadrature, we have, for all $u_{\mathcal{N}}, v_{\mathcal{N}} \in X^0_{\mathcal{N}}$,

$$(\tilde{\partial}_x u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda} = (\partial_x u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda} = -(u_{\mathcal{N}}, \partial_x v_{\mathcal{N}})_{\mathcal{N},\Lambda} = -(u_{\mathcal{N}}, \tilde{\partial}_x v_{\mathcal{N}})_{\mathcal{N},\Lambda}.$$

(2.17) is a direct consequence of (2.16) and (2.13). □

Remark 2.1. Since in general $\partial_x v_{\mathcal{N}} \notin X_{\mathcal{N}}$, consequently,

$$(\tilde{\partial}_x u_{\mathcal{N}}, \tilde{\partial}_x v_{\mathcal{N}})_{\mathcal{N},\Lambda} \neq (\partial_x u_{\mathcal{N}}, \partial_x v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad u_{\mathcal{N}}, v_{\mathcal{N}} \in X_{\mathcal{N}},$$

$$(\tilde{\partial}_x u_{\mathcal{N}}, \partial_x v_{\mathcal{N}})_{\mathcal{N},\Lambda} \neq (\partial_x u_{\mathcal{N}}, \partial_x v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad u_{\mathcal{N}}, v_{\mathcal{N}} \in X_{\mathcal{N}}.$$

The new spectral-element method for (2.7) with $u(0) = u(R) = 0$ is: Find $u_{\mathcal{N}} \in X^0_{\mathcal{N}}$, such that

$$\tilde{b}(u_{\mathcal{N}}, v_{\mathcal{N}}) := \alpha(u_{\mathcal{N}}, v_{\mathcal{N}})_{\mathcal{N},\Lambda} + \beta(\tilde{\partial}_x u_{\mathcal{N}}, \tilde{\partial}_x v_{\mathcal{N}})_{\mathcal{N},\Lambda} = (f, v_{\mathcal{N}})_{\mathcal{N},\Lambda}, \quad \forall v_{\mathcal{N}} \in X^0_{\mathcal{N}}. \tag{2.18}$$

In the next subsection, we shall use (2.11) and (2.18) to construct our direction splitting scheme. We will defer to the next section on how to solve the related linear systems.

2.3. A spectral-element method for the direction splitting scheme

While it is straightforward to discretize the direction splitting scheme (2.4)–(2.5) by using a finite difference method (cf. [3]), it requires special considerations if one wants to use a high-order spectral type method. In fact, in order to maintain the unconditional stability of the semi-discretized scheme, a hybrid spectral-collocation and spectral-Galerkin method was developed in [1]. In this paper, we construct the direction splitting scheme by using a spectral-element method for the spatial discretization based on the weak formulation with the LGL quadrature.

To fix the idea, we shall only consider the three-dimensional case. We first subdivide Ω into K^3 equal-size subdomains $\Omega = \bigcup_{1\le i,j,k\le K} \Omega_{ijk}$. Each subdomain Ω_{ijk} will be mapped to the reference domain $\hat{\Omega} = (-1,1)^d$.

Let $\hat{\Sigma}$ be the set of LGL points in $\hat{\Omega}$, i.e., $\hat{\Sigma} = \{(x_p, y_q, z_r) : 0 \le p,q,r \le N\}$ where $\{y_p = z_p = x_p\}_{0\le p\le N}$. We denote the set of mapped collocation points in Ω_{ijk} by Σ_{ijk}. Then, the set of all collocation points on Ω is

$$\Sigma_{all} = \bigcup_{1\le i,j,k\le K} \Sigma_{ijk} = \bigcup_{1\le i,j,k\le K} \left\{\left(x^i_p, y^j_q, z^k_r\right) : 0 \le p,q,r \le N\right\}.$$

The set of all interior collocation points on Ω is denoted by

$$\Sigma^0_{all} = \bigcup_{1\le i,j,k\le K} \Sigma_{ijk} = \bigcup_{1\le i,j,k\le K} \left\{\left(x^i_p, y^j_q, z^k_r\right) : 1 \le p,q,r \le N-1\right\}.$$

Denote the spectral-element approximation space by

$$Y_{\mathcal{N}} = \left\{u \in C(\Omega) : u|_{\Omega_{ijk}} \in P_N(\Omega_{ijk}),\ 1 \le i,j,k \le K\right\},$$

$$Y^0_{\mathcal{N}} = \{u \in Y_{\mathcal{N}} : u|_{\partial\Omega} = 0\},$$

$$\mathbb{Y}_{\mathcal{N}} = (Y_{\mathcal{N}})^d, \qquad \mathbb{Y}^0_{\mathcal{N}} = \left(Y^0_{\mathcal{N}}\right)^d. \tag{2.19}$$

We define the interpolation operator $\mathbb{I}_{\mathcal{N}} : C(\Omega) \to Y_{\mathcal{N}}$ by

$$(\mathbb{I}_{\mathcal{N}} f)(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \Sigma_{all}. \tag{2.20}$$

We propose a spectral-element method for (2.4)–(2.5) based on a weak formulation with LGL quadrature.

We initialize the algorithm by setting $\boldsymbol{u}^0_{\mathcal{N}} = \mathbb{I}_{\mathcal{N}} \boldsymbol{u}_0$, $p^{-\frac{1}{2}}_{\mathcal{N}} = \mathbb{I}_{\mathcal{N}} p_0$, $\phi^{-\frac{1}{2}}_{\mathcal{N}} = 0$.

- **Predictor for velocity.**
  We start by computing a predictor for the velocity, using a spectral element approach with the 3D-LGL quadrature: Find $\boldsymbol{\xi}^{n+1}_{\mathcal{N}} \in \mathbb{Y}^0_{\mathcal{N}}$ s.t.

$$\begin{aligned}&\left(\frac{\boldsymbol{\xi}^{n+1}_{\mathcal{N}} - \boldsymbol{u}^n_{\mathcal{N}}}{\Delta t}, \boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega} + \nu\big(\tilde{\nabla} \boldsymbol{u}^n_{\mathcal{N}}, \tilde{\nabla} \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} + \big(\nabla p^{*,n+\frac{1}{2}}_{\mathcal{N}}, \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} + \big(\mathbf{NL}^{n+1}\big(\boldsymbol{u}^n_{\mathcal{N}}, \boldsymbol{u}^{n-1}_{\mathcal{N}}\big), \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}\\ &\quad = \big(\boldsymbol{f}^{n+\frac{1}{2}}, \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}, \quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{Y}^0_{\mathcal{N}},\end{aligned} \tag{2.21}$$

  where $(\cdot,\cdot)_{\mathcal{N},\Omega}$ is the 3D-LGL quadrature defined on Σ_{all}, $\tilde{\nabla} := (\tilde{\partial}_x, \tilde{\partial}_y, \tilde{\partial}_z)^T$, and

$$\begin{aligned}&p^{*,n+\frac{1}{2}}_{\mathcal{N}}(\boldsymbol{x}) = p^{n-\frac{1}{2}}_{\mathcal{N}}(\boldsymbol{x}) + \phi^{n-\frac{1}{2}}_{\mathcal{N}}(\boldsymbol{x}),\\ &\mathbf{NL}_{\mathcal{N}}\big(\boldsymbol{u}^n_{\mathcal{N}}, \boldsymbol{u}^{n-1}_{\mathcal{N}}\big) = \frac{3}{2}\big(\boldsymbol{u}^n_{\mathcal{N}} \cdot \tilde{\nabla}\big)\boldsymbol{u}^n_{\mathcal{N}} - \frac{1}{2}\big(\boldsymbol{u}^{n-1}_{\mathcal{N}} \cdot \tilde{\nabla}\big)\boldsymbol{u}^{n-1}_{\mathcal{N}}, \quad \forall \boldsymbol{x} \in \Sigma_{all}.\end{aligned} \tag{2.22}$$

- **Direction splitting for velocity.**
  We now propose a direction splitting using the one-dimensional spectral element approach proposed in Subsection 2.2. For each $q, r = 0, 1, \cdots, N$ and $j, k = 1, \cdots, K$, find $\boldsymbol{\eta}^{n+1}_{\mathcal{N}}(\cdot, y^j_q, z^k_r) \in \mathbb{X}^0_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\left(\frac{\boldsymbol{\eta}^{n+1}_{\mathcal{N}} - \boldsymbol{u}^n_{\mathcal{N}}}{\Delta t}\big(\cdot, y^j_q, z^k_r\big), \boldsymbol{v}_{\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda} + \frac{1}{2}\nu\big(\tilde{\partial}_x\big(\boldsymbol{\eta}^{n+1}_{\mathcal{N}} - \boldsymbol{u}^n_{\mathcal{N}}\big)\big(\cdot, y^j_q, z^k_r\big), \tilde{\partial}_x \boldsymbol{v}_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}\\ &\quad = \left(\frac{\boldsymbol{\xi}^{n+1}_{\mathcal{N}} - \boldsymbol{u}^n_{\mathcal{N}}}{\Delta t}\big(\cdot, y^j_q, z^k_r\big), \boldsymbol{v}_{\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda}, \quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{X}^0_{\mathcal{N}};\end{aligned} \tag{2.23}$$

  For each $p, r = 0, 1, \cdots, N$ and $i, k = 1, \cdots, K$, find $\boldsymbol{\zeta}^{n+1}_{\mathcal{N}}(x^i_p, \cdot, z^k_r) \in \mathbb{X}^0_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\left(\frac{\boldsymbol{\zeta}^{n+1}_{\mathcal{N}} - \boldsymbol{\eta}^{n+1}_{\mathcal{N}}}{\Delta t}\big(x^i_p, \cdot, z^k_r\big), \boldsymbol{v}_{\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda} + \frac{1}{2}\nu\big(\tilde{\partial}_y\big(\boldsymbol{\zeta}^{n+1}_{\mathcal{N}} - \boldsymbol{u}^n_{\mathcal{N}}\big)\big(x^i_p, \cdot, z^k_r\big)_{\mathcal{N},\Lambda}, \tilde{\partial}_y \boldsymbol{v}_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda} = \mathbf{0},\\ &\quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{X}^0_{\mathcal{N}};\end{aligned} \tag{2.24}$$

  For each $p, q = 0, 1, \cdots, N$ and $i, j = 1, \cdots, K$, find $\boldsymbol{u}^{n+1}_{\mathcal{N}}(x^i_p, y^j_q, \cdot) \in \mathbb{X}^0_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\left(\frac{\boldsymbol{u}^{n+1}_{\mathcal{N}} - \boldsymbol{\zeta}^{n+1}_{\mathcal{N}}}{\Delta t}\big(x^i_p, y^j_q, \cdot\big), \boldsymbol{v}_{\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda} + \frac{1}{2}\nu\big(\tilde{\partial}_z\big(\boldsymbol{u}^{n+1}_{\mathcal{N}} - \boldsymbol{u}^n_{\mathcal{N}}\big)\big(x^i_p, y^j_q, \cdot\big), \tilde{\partial}_z \boldsymbol{v}_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda} = \mathbf{0},\\ &\quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{X}^0_{\mathcal{N}}.\end{aligned} \tag{2.25}$$

- **Direction splitting for pressure.**
  As for the velocity, we perform the direction splitting for the pressure as follows:
  For each $q, r = 0, 1, \cdots, N$ and $j, k = 1, \cdots, K$, find $\psi^{n+\frac{1}{2}}_{\mathcal{N}}(\cdot, y^j_q, z^k_r) \in X_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\big(\psi^{n+\frac{1}{2}}_{\mathcal{N}}\big(\cdot, y^j_q, z^k_r\big), w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda} + \big(\partial_x \psi^{n+\frac{1}{2}}_{\mathcal{N}}\big(\cdot, y^j_q, z^k_r\big), \partial_x w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}\\ &\quad = \left(-\frac{\nabla \cdot \boldsymbol{u}^{n+1}_{\mathcal{N}}\big(\cdot, y^j_q, z^k_r\big)}{\Delta t}, w_{\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda}, \quad \forall w_{\mathcal{N}} \in X_{\mathcal{N}};\end{aligned} \tag{2.26}$$

  For each $p, r = 0, 1, \cdots, N$ and $i, k = 1, \cdots, K$, find $\varphi^{n+\frac{1}{2}}_{\mathcal{N}}(x^i_p, \cdot, z^k_r) \in X_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\big(\varphi^{n+\frac{1}{2}}_{\mathcal{N}}\big(x^i_p, \cdot, z^k_r\big), w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda} + \big(\partial_y \varphi^{n+\frac{1}{2}}_{\mathcal{N}}\big(x^i_p, \cdot, z^k_r\big), \partial_y w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}\\ &\quad = \big(\psi^{n+\frac{1}{2}}_{\mathcal{N}}\big(x^i_p, \cdot, z^k_r\big), w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}, \quad \forall w_{\mathcal{N}} \in X_{\mathcal{N}};\end{aligned} \tag{2.27}$$

For each $p, q = 0, 1, \cdots, N$ and $i, j = 1, \cdots, K$, find $\phi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i, y_q^j, \cdot) \in X_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\big(\phi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i, y_q^j, \cdot), w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda} + \big(\partial_z\phi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i, y_q^j, \cdot), \partial_z w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}\\ &\quad = \big(\varphi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i, y_q^j, \cdot), w_{\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}, \quad \forall w_{\mathcal{N}} \in X_{\mathcal{N}}.\end{aligned} \tag{2.28}$$

- **Pressure update.**

Find $p_{\mathcal{N}}^{n+\frac{1}{2}} \in Y_{\mathcal{N}}$, s.t.

$$\begin{aligned}\big(p_{\mathcal{N}}^{n+\frac{1}{2}}, q_{\mathcal{N}}\big)_{\mathcal{N},\Omega} &= \big(p_{\mathcal{N}}^{n-\frac{1}{2}}, q_{\mathcal{N}}\big)_{\mathcal{N},\Omega} + \big(\phi_{\mathcal{N}}^{n+\frac{1}{2}}, q_{\mathcal{N}}\big)_{\mathcal{N},\Omega}\\ &\quad - \frac{1}{2}\chi\nu\big(\nabla\cdot\big(\boldsymbol{u}_{\mathcal{N}}^{n+1} + \boldsymbol{u}_{\mathcal{N}}^{n}\big), q_{\mathcal{N}}\big)_{\mathcal{N},\Omega}, \quad \forall q_{\mathcal{N}} \in Y_{\mathcal{N}}.\end{aligned} \tag{2.29}$$

The above spectral element direction splitting scheme results in a sequence of one-dimensional problems (2.11), for the velocity components, and (2.18) for the pressure. The different choice of spectral-element methods used here appears to be necessary to establish the stability of the overall scheme.

3. Parallel spectral element method

We describe below a parallel implementation of the spectral-element direction splitting scheme presented in the previous section.

3.1. Schur-compliment approach for one-dimensional spectral-elements

We first describe a fast algorithm for solving the classical spectral-element systems (2.10) and (2.11).

We first consider (2.10) and construct a set of basis functions for $X_{\mathcal{N}}^0$. Transform the subdomain Λ^k to the reference domain $\hat{\Lambda} = (-1, 1)$ with

$$\hat{x}_k := \hat{x}_k(x) = \frac{2}{h_k}x - \frac{a_k + a_{k-1}}{h_k}, \quad \forall x \in \Lambda^k,$$

and denote

$$\mathring{V}_N^k := P_N\big(\Lambda^k\big) \cap H_0^1\big(\Lambda^k\big), \quad k = 1, 2, \cdots, K.$$

Then, a set of ortho-normal basis in $H_0^1(\Lambda^k)$ for $\mathring{V}_N^k, k = 1, 2, \cdots, K$, is (cf. [15]):

$$\phi_j^k(x) = \begin{cases}\frac{1}{\sqrt{4j+6}}(L_j(\hat{x}_k) - L_{j+2}(\hat{x}_k)), & x \in \Lambda^k,\\ 0, & \text{others,}\end{cases} \quad j = 0, 1, \cdots, N-2, \tag{3.1}$$

where $L_j(x)$ is the Legendre polynomial of degree j. Clearly,

$$\mathring{V}_N^k = \operatorname{span}\big\{\phi_0^k(x), \phi_1^k(x), \cdots, \phi_{N-2}^k(x)\big\}, \quad k = 1, 2, \cdots, K.$$

We define the space consisting of all interior basis functions:

$$\mathring{X}_{\mathcal{N}} = \big\{v;\, v_{|\Lambda^k} \in \mathring{V}_N^k, k = 1, 2, \cdots, K\big\}.$$

For the basis functions at the nodes $\{a_j\}_{j=1}^{K-1}$, we define

$$\varphi_k(x) = \begin{cases}\frac{1}{2}(L_0(\hat{x}_k) + L_1(\hat{x}_k)), & x \in \Lambda^k,\\ \frac{1}{2}(L_0(\hat{x}_{k+1}) - L_1(\hat{x}_{k+1})), & x \in \Lambda^{k+1},\\ 0, & \text{otherwise.}\end{cases} \tag{3.2}$$

It is then clear that

$$X_{\mathcal{N}}^0 = \mathring{X}_{\mathcal{N}} \oplus \operatorname{span}\{\varphi_1, \varphi_2, \cdots, \varphi_{K-1}\}.$$

With this set of basis functions, the system (2.10) leads to a coupled linear system. We shall now use the Schur-compliment procedure to decouple the computation of interface nodes from the interior nodes.

- **Pre-computation:** We construct the orthogonal complement of the space $\mathring{X}_{\mathcal{N}}$ with respect to the bilinear form $b(\cdot,\cdot)$. Let $\mathring{\varphi}_1, \mathring{\varphi}_2, \cdots, \mathring{\varphi}_{K-1} \in \mathring{X}_{\mathcal{N}}$ be the solutions of the following problems:

$$b(\mathring{\varphi}_k, \mathring{v}_{\mathcal{N}}) = -b(\varphi_k, \mathring{v}_{\mathcal{N}}), \quad \forall \mathring{v}_{\mathcal{N}} \in \mathring{X}_{\mathcal{N}},\ k = 1, 2, \cdots, K-1. \tag{3.3}$$

Then, by construction, we have

$$b(\mathring{\varphi}_k + \varphi_k, v_{\mathcal{N}}) = 0, \quad \forall v_{\mathcal{N}} \in \mathring{X}_{\mathcal{N}}. \tag{3.4}$$

Hence, setting $\Theta_k = \mathring{\varphi}_k + \varphi_k$, $k = 1, 2, \cdots, K-1$, we find that the orthogonal complement, in the sense of (3.4), of $\mathring{X}_{\mathcal{N}}$ in $X_{\mathcal{N}}$ is given by $\mathring{X}_{\mathcal{N}}^{\perp} = \operatorname{span}\{\Theta_1, \Theta_2, \cdots, \Theta_{K-1}\}$.
Next, we compute the Schur-complement matrix S with $S_{jk} = b(\Theta_k, \Theta_j)$ for $1 \le j,k \le K-1$. Since $\operatorname{supp}(\varphi_k) = \Lambda^k \cup \Lambda^{k+1}$, it is clear from (3.3) that $\operatorname{supp}(\mathring{\varphi}_k) = \Lambda^k \cup \Lambda^{k+1}$. Hence, we have $\operatorname{supp}(\Theta_k) = \Lambda^k \cup \Lambda^{k+1}$ which indicates that S is a symmetric tridiagonal matrix.
- **Step 1:** Find the values at the nodes $\{a_j\}_{j=1}^{K-1}$ by solving the following Schur-complement problem:

$$\sum_{k=1}^{K-1} b(\Theta_k, \Theta_j) u_{\mathcal{N}}(a_k) = (f, \Theta_j), \quad j = 1, 2, \cdots, K-1. \tag{3.5}$$

Note that the above is equivalent to a matrix system with the symmetric tridiagonal matrix S.
- **Step 2:** Solve a sequence of subproblem in Λ^k for $k = 1, 2, \cdots, K$.
Find $\mathring{u}_{\mathcal{N}} \in \mathring{X}_{\mathcal{N}}$, such that

$$b(\mathring{u}_{\mathcal{N}}, \mathring{v}_{\mathcal{N}}) = (f, \mathring{v}_{\mathcal{N}}), \quad \forall \mathring{v}_{\mathcal{N}} \in \mathring{X}_{\mathcal{N}}. \tag{3.6}$$

We now derive the matrix system of (3.6) using the basis functions constructed above. We denote

$$\begin{aligned}
&\mathring{u}_{\mathcal{N}}(x)_{|\Lambda^k} = \sum_{i=0}^{N-2} \hat{u}_i^k \phi_i^k(x), \qquad U^k = \left(\hat{u}_1^k, \hat{u}_2^k, \cdots, \hat{u}_{N-2}^k\right)^T;\\
&F^k = \left(f_1^k, f_2^k, \cdots, f_{N-2}^k\right)^T \quad \text{with } f_j^k = \left(f, \phi_i^k\right)_{\Lambda^k};\\
&A^k = \left(A_{ji}^k\right)_{0\le i,j\le N-2} \quad \text{with } A_{ji}^k = \left(\phi_i^{k'}, \phi_j^{k'}\right)_{\Lambda^k};\\
&B^k = \left(B_{ji}^k\right)_{0\le i,j\le N-2} \quad \text{with } B_{ji}^k = \left(\phi_i^k, \phi_j^k\right)_{\Lambda^k}.
\end{aligned} \tag{3.7}$$

Plugging the expansion for $\mathring{u}_{\mathcal{N}}$ in (3.6) and taking $\mathring{v}_{\mathcal{N}} = \varphi_j^k$, we obtain the following linear systems:

$$\left(\alpha B^k + \beta A^k\right) U^k = F^k, \quad k = 1, 2, \cdots, K. \tag{3.8}$$

It is clear that the matrices B^k (resp. A^k) are proportional to the 1-D mass matrix B (resp. stiffness matrix A) on the unit interval, and by the orthogonality of the Legendre polynomials, it can be easily verified that A^k is diagonal and B^k is tridiagonal [15].
- **Step 3:** The solution of (2.10) is obtained through the expression

$$u_{\mathcal{N}} = \mathring{u}_{\mathcal{N}} + \sum_{k=1}^{K-1} u_{\mathcal{N}}(a_k)\Theta_k.$$

Indeed, for all $v_{\mathcal{N}} \in X_{\mathcal{N}}^0$, there exist $\mathring{v}_{\mathcal{N}} \in \mathring{X}_{\mathcal{N}}$ and $v_{\mathcal{N}}^{\perp} \in \mathring{X}_{\mathcal{N}}^{\perp}$ such that $v_{\mathcal{N}} = \mathring{v}_{\mathcal{N}} + v_{\mathcal{N}}^{\perp}$, and consequently

$$\begin{aligned}
b(u_{\mathcal{N}}, v_{\mathcal{N}}) &= b\left(\mathring{u}_{\mathcal{N}} + \sum_{k=1}^{K-1} u_{\mathcal{N}}(a_k)\Theta_k, \mathring{v}_{\mathcal{N}} + v_{\mathcal{N}}^{\perp}\right)\\
&= b(\mathring{u}_{\mathcal{N}}, \mathring{v}_{\mathcal{N}}) + b\left(\mathring{u}_{\mathcal{N}}, v_{\mathcal{N}}^{\perp}\right) + \sum_{k=1}^{K-1} u_{\mathcal{N}}(a_k) b(\Theta_k, \mathring{v}_{\mathcal{N}}) + \sum_{k=1}^{K-1} u_{\mathcal{N}}(a_k) b\left(\Theta_k, v_{\mathcal{N}}^{\perp}\right)\\
&= (f, \mathring{v}_{\mathcal{N}}) + \left(f, v_{\mathcal{N}}^{\perp}\right) = (f, v_{\mathcal{N}}).
\end{aligned}$$

In the above, we have used the fact that $b(\mathring{u}_{\mathcal{N}}, v_{\mathcal{N}}^{\perp}) = 0$ and $b(\Theta_k, \mathring{v}_{\mathcal{N}}) = 0$ due to the orthogonality. Thus $\mathring{u}_{\mathcal{N}} + \sum_{k=1}^{K-1} u_{\mathcal{N}}(a_k)\Theta_k$ is the solution of (2.10).

In summary, after pre-computing the orthogonal complement of $\mathring{X}_{\mathcal{N}}$ and the Schur-complement matrix, then for each right hand side f, the above algorithm decomposes the problem (2.10) into a set of *local* problems (3.6) in each subinterval which can be solved independently and in parallel, and a *global* tridiagonal Schur-complement problem (3.5).

To deal with (2.11), we just need to replace $X^0_{\mathcal{N}}$ by $X_{\mathcal{N}}$, and add two more basis functions related to the nodes a_0 and a_k as follows:

$$\varphi_0(x) = \begin{cases} \frac{1}{2}(L_0(\hat{x}_1) - L_1(\hat{x}_1)), & x \in \Lambda^1, \\ 0, & \text{otherwise}, \end{cases}$$
$$\varphi_K(x) = \begin{cases} \frac{1}{2}(L_0(\hat{x}_K) + L_1(\hat{x}_K)), & x \in \Lambda^K, \\ 0, & \text{otherwise}. \end{cases}$$

The above procedure can be directly extended to (2.18). We only need to replace the bilinear form $b(\cdot,\cdot)$ by $\tilde{b}(\cdot,\cdot)$. Since the orthogonality of the basis functions $\{\phi^k_j, j = 0, 1, \ldots, N-2\}$, defined in (3.1), is no longer valid for $\tilde{b}(\cdot,\cdot)$, it is more convenient to use the Lagrangian nodal basis for $\mathring{V}^k_N$.

3.2. Parallel implementation

As demonstrated in [3], the scheme (2.4)–(2.6) with a finite difference discretization is extremely well suited for parallel computation. We are going to see next that, by using the spectral-element methods described above, the scheme (2.21)–(2.29) is equally well-suited for parallel computation.

We take the 3-D case with K^3 subdomains as an example, and describe a parallel algorithm using Message Passing Interface (MPI) on K^3 processors.

We associate each subdomain Ω_{ijk} with a processor $P_{(i,j,k)}$, for all $1 \le i, j, k \le K$, and store the right-hand side function values f^{ijk}_{pqr} and solution values u^{ijk}_{pqr} with $0 \le p, q, r \le N$ on the processor $P_{(i,j,k)}$.

The Schur-complement in the x-, y- and z-direction will be stored in $P_{(1,j,k)}$, $P_{(j,1,k)}$, and $P_{(j,k,1)}$, respectively for each $j, k = 1, \cdots K$.

Since, at each time step, our direction splitting method leads to a sequence of one dimensional sub-problems, we only need to discuss the parallel implementation for the one-dimensional equation (2.10).

In order to reduce communications, we decompose the nodal basis functions φ_k, defined in (3.2), as $\varphi_k = \varphi^l_k + \varphi^r_k$ with

$$\varphi^l_k(x) = \begin{cases} \frac{1}{2}(L_0(\hat{x}_k) + L_1(\hat{x}_k)), & x \in \Lambda^k, \\ 0, & \text{otherwise}, \end{cases} \tag{3.9}$$

and

$$\varphi^r_k(x) = \begin{cases} \frac{1}{2}(L_0(\hat{x}_{k+1}) - L_1(\hat{x}_{k+1})), & x \in \Lambda^{k+1}, \\ 0, & \text{otherwise}. \end{cases} \tag{3.10}$$

Accordingly, we set $\Theta^l_k = \mathring{\varphi}_k + \varphi^l_k$ and $\Theta^r_k = \mathring{\varphi}_{k+1} + \varphi^r_k$.

We now summarize the parallel algorithm for (2.10) on K processors below.

Pre-computation: Compute the matrices B^k, A^k in (3.7) on the k-th processor, $k = 1, 2, \cdots, K$. Fix the first processor as the main processor, construct and store the Schur-complement matrix S, with $S_{jk} = b(\Theta_k, \Theta_j)$ for $1 \le j, k \le K-1$, on this processor.

Then for each right-hand side function f, we perform:

- **Step 1:** Compute the data (f, Θ^l_k) and (f, Θ^r_{k-1}) on the k-th processor and send it to the first processor where the Schur-complement is stored. Then, solve the Schur-complement problem on the first processor:
$$\sum_{k=1}^{K-1} b(\Theta_k, \Theta_j) u_{\mathcal{N}}(a_k) = \left(f, \Theta^l_j\right) + \left(f, \Theta^r_j\right) = (f, \Theta_j), \quad j = 1, 2, \cdots, K-1. \tag{3.11}$$
Send $u_{\mathcal{N}}(a_{k-1}), u_{\mathcal{N}}(a_k)$ back to the k-th processor.
- **Step 2:** On each processor k, compute F^k defined in (3.7), and solve (3.8) to get the solution at the interior of Λ^k, $\mathring{u}^k_{\mathcal{N}}$. Finally, we obtain the solution in Λ^k by setting $u^k_{\mathcal{N}} = \mathring{u}^k_{\mathcal{N}} + u_{\mathcal{N}}(a_{k-1})\Theta^r_{k-1} + u_{\mathcal{N}}(a_k)\Theta^l_k$ on each processor.

Note that the total amount of communications for solving each one-dimensional sub-problem consists of receiving/sending two values from/to each processor to/from the first processor. So the algorithm is well suited for parallel computation.

4. A proof of stability

To simplify the presentation, we shall only consider the two-dimensional case and show that the full discretized scheme without the no-linear term and with $\boldsymbol{f} = 0$ and $\chi = 0$ is unconditionally stable.

We will need some more notations. We define the bilinear form

$$a_{\mathcal{N}}(p_{\mathcal{N}},q_{\mathcal{N}})=(p_{\mathcal{N}},q_{\mathcal{N}})_{\mathcal{N}}+(\nabla p_{\mathcal{N}},\nabla q_{\mathcal{N}})_{\mathcal{N}}+(\partial_{xy}p_{\mathcal{N}},\partial_{xy}q_{\mathcal{N}})_{\mathcal{N}},\quad \forall p_{\mathcal{N}},q_{\mathcal{N}}\in Y_{\mathcal{N}},\tag{4.1}$$

and the associated norm $\|\cdot\|_{A_{\mathcal{N}}}:=a_{\mathcal{N}}(\cdot,\cdot)^{1/2}$. It is obvious that:

$$a_{\mathcal{N}}(\cdot,\cdot)\text{ is symmetric, and }\|\nabla q_{\mathcal{N}}\|_{\mathcal{N}}\le\|\cdot\|_{A_{\mathcal{N}}},\quad \forall q_{\mathcal{N}}\in Y_{\mathcal{N}}.\tag{4.2}$$

We define the difference operator δ by

$$\delta p_{\mathcal{N}}^{n+\frac{1}{2}}:=p_{\mathcal{N}}^{n+\frac{1}{2}}-p_{\mathcal{N}}^{n-\frac{1}{2}},\qquad \delta^2 p_{\mathcal{N}}^{n+\frac{1}{2}}:=p_{\mathcal{N}}^{n+\frac{1}{2}}-2p_{\mathcal{N}}^{n-\frac{1}{2}}+p_{\mathcal{N}}^{n-\frac{3}{2}},$$

and use $\bar{\boldsymbol{u}}_{\mathcal{N}}$ to denote the sequence whose generic term is $\bar{\boldsymbol{u}}_{\mathcal{N}}^{n+\frac{1}{2}}:=\frac{1}{2}(\boldsymbol{u}_{\mathcal{N}}^{n+1}+\boldsymbol{u}_{\mathcal{N}}^{n})$.

Without the no-linear term and with $\boldsymbol{f}=0$ and $\chi=0$, the scheme (2.21)–(2.29) in the two-dimensional case reads:

- **Direction splitting for velocity.**
  Find $\boldsymbol{\xi}_{\mathcal{N}}^{n+1}\in\mathbb{Y}_{\mathcal{N}}^{0}$, s.t.

$$\left(\frac{\boldsymbol{\xi}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t},\boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega}+\nu\big(\tilde\nabla\boldsymbol{u}_{\mathcal{N}}^{n},\tilde\nabla\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}+\big(\nabla p_{\mathcal{N}}^{*,n+\frac{1}{2}},\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}=\boldsymbol{0},\quad \forall\boldsymbol{v}_{\mathcal{N}}\in\mathbb{Y}_{\mathcal{N}}^{0}\tag{4.3}$$

  with

$$p_{\mathcal{N}}^{*,n+\frac{1}{2}}(\boldsymbol{x})=2p_{\mathcal{N}}^{n-\frac{1}{2}}(\boldsymbol{x})-p_{\mathcal{N}}^{n-\frac{3}{2}}(\boldsymbol{x}),\quad \forall\boldsymbol{x}\in\Sigma_{all};$$

  For each $q=0,1,\cdots,N$ and $j=1,\cdots,K$, find $\boldsymbol{\eta}_{\mathcal{N}}^{n+1}(\cdot,y_q^j)\in\mathbb{X}_{\mathcal{N}}^{0}$, s.t.

$$\begin{aligned}&\left(\frac{\boldsymbol{\eta}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t}(\cdot,y_q^j),\boldsymbol{v}_{1,\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda}+\frac{1}{2}\nu\big(\tilde\partial_x\big(\boldsymbol{\eta}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\big)(\cdot,y_q^j),\tilde\partial_x\boldsymbol{v}_{1,\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}\\&\quad=\left(\frac{\boldsymbol{\xi}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t}(\cdot,y_q^j),\boldsymbol{v}_{1,\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda},\quad \forall\boldsymbol{v}_{1,\mathcal{N}}\in\mathbb{X}_{\mathcal{N}}^{0};\end{aligned}\tag{4.4}$$

  For each $p=0,1,\cdots,N$ and $i=1,\cdots,K$, find $\boldsymbol{u}_{\mathcal{N}}^{n+1}(x_p^i,\cdot)\in\mathbb{X}_{\mathcal{N}}^{0}$, s.t.

$$\begin{aligned}&\left(\frac{\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{\eta}_{\mathcal{N}}^{n+1}}{\Delta t}(x_p^i,\cdot),\boldsymbol{v}_{2,\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda}+\frac{1}{2}\nu\big(\tilde\partial_y\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\big)(x_p^i,\cdot),\tilde\partial_y\boldsymbol{v}_{2,\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}=\boldsymbol{0},\\&\forall\boldsymbol{v}_{2,\mathcal{N}}\in\mathbb{X}_{\mathcal{N}}^{0}.\end{aligned}\tag{4.5}$$

- **Direction splitting for pressure.**
  For each $q=0,1,\cdots,N$ and $j=1,\cdots,K$, find $\psi_{\mathcal{N}}^{n+\frac{1}{2}}(\cdot,y_q^j)\in X_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\big(\psi_{\mathcal{N}}^{n+\frac{1}{2}}(\cdot,y_q^j),q_{1,\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}+\big(\partial_x\psi_{\mathcal{N}}^{n+\frac{1}{2}}(\cdot,y_q^j),\partial_x q_{1,\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}\\&\quad=\left(-\frac{\nabla\cdot\boldsymbol{u}_{\mathcal{N}}^{n+1}(\cdot,y_q^j)}{\Delta t},q_{1,\mathcal{N}}(\cdot)\right)_{\mathcal{N},\Lambda},\quad \forall q_{1,\mathcal{N}}\in X_{\mathcal{N}};\end{aligned}\tag{4.6}$$

  For each $p=0,1,\cdots,N$ and $i=1,\cdots,K$, find $\varphi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i,\cdot)\in X_{\mathcal{N}}$, s.t.

$$\begin{aligned}&\big(\phi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i,\cdot),q_{2,\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}+\big(\partial_y\phi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i,\cdot),\partial_y q_{2,\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda}\\&\quad=\big(\psi_{\mathcal{N}}^{n+\frac{1}{2}}(x_p^i,\cdot),q_{2,\mathcal{N}}(\cdot)\big)_{\mathcal{N},\Lambda},\quad \forall q_{2,\mathcal{N}}\in X_{\mathcal{N}}.\end{aligned}\tag{4.7}$$

- **Pressure update.**
  Find $p_{\mathcal{N}}^{n+\frac{1}{2}}\in Y_{\mathcal{N}}$, s.t.

$$\big(p_{\mathcal{N}}^{n+\frac{1}{2}},q_{\mathcal{N}}\big)_{\mathcal{N},\Omega}=\big(p_{\mathcal{N}}^{n-\frac{1}{2}},q_{\mathcal{N}}\big)_{\mathcal{N},\Omega}+\big(\phi_{\mathcal{N}}^{n+\frac{1}{2}},q_{\mathcal{N}}\big)_{\mathcal{N},\Omega},\quad \forall q_{\mathcal{N}}\in Y_{\mathcal{N}}.\tag{4.8}$$

The following lemma can be proved by using essentially the same arguments as in Lemma 3.1 of [1].

Lemma 4.1. *The pressure $\phi_{\mathcal{N}}^{n+\frac{1}{2}} \in Y_{\mathcal{N}}(\Omega)$ given by the splitting scheme (4.6)–(4.7) satisfies:*

$$a_{\mathcal{N}}\big(\phi_{\mathcal{N}}^{n+\frac{1}{2}}, q_{\mathcal{N}}\big) = \left(-\frac{\nabla\cdot \boldsymbol{u}_{\mathcal{N}}^{n+1}}{\Delta t}, q_{\mathcal{N}}\right)_{\mathcal{N},\Omega}, \quad \forall q_{\mathcal{N}} \in Y_{\mathcal{N}}(\Omega), \tag{4.9}$$

where $a_{\mathcal{N}}(\cdot,\cdot)$ is defined in (4.1).

Our main theoretical result is the following:

Theorem 4.1. *The solution to (4.3)–(4.8) with $\boldsymbol{f} = \boldsymbol{0}$ and $\chi = 0$ is unconditionally stable and satisfies the following dissipative law for all $n \ge 0$:*

$$\begin{aligned}&\left(\|\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2 + \frac{\Delta t\nu}{2}\|\tilde{\nabla}\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2 + \Delta t^2\|p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2 + \frac{\Delta t^2\nu^2}{4}\|\tilde{\partial}_{xy}\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2\right) + \frac{\Delta t\nu}{2}\|\tilde{\nabla}(\boldsymbol{u}_{\mathcal{N}}^{n+1} + \boldsymbol{u}_{\mathcal{N}}^{n})\|_{\mathcal{N}}^2\\ &\quad\le \left(\|\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2 + \frac{\Delta t\nu}{2}\|\tilde{\nabla}\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2 + \Delta t^2\|p_{\mathcal{N}}^{n-\frac{1}{2}}\|_{A_{\mathcal{N}}}^2 + \frac{\Delta t^2\nu^2}{4}\|\tilde{\partial}_{xy}\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2\right).\end{aligned}$$

Proof. For any $\boldsymbol{v}_{2,\mathcal{N}}(y) \in \mathbb{X}_{\mathcal{N}}^0$, multiplying (4.4) by $\boldsymbol{v}_{2,\mathcal{N}}(y_q^j)$, then performing the LGL quadrature in the y-direction yield: $\forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{Y}_{\mathcal{N}}^0$,

$$\left(\frac{\boldsymbol{\eta}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t}, \boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega} + \frac{\nu}{2}\big(\tilde{\partial}_x\big(\boldsymbol{\eta}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n}\big), \tilde{\partial}_x\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} = \left(\frac{\boldsymbol{\xi}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t}, \boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega}. \tag{4.10}$$

Then combining (4.10) and (4.3) leads to

$$\begin{aligned}&-\frac{\nu}{2}\big(\tilde{\partial}_x\big(\boldsymbol{\eta}_{\mathcal{N}}^{n+1} + \boldsymbol{u}_{\mathcal{N}}^{n}\big), \tilde{\partial}_x\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} - \nu\big(\tilde{\partial}_y\boldsymbol{u}_{\mathcal{N}}^{n}, \tilde{\partial}_y\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} - \big(\nabla p_{\mathcal{N}}^{*,n+\frac{1}{2}}, \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}\\ &\quad= \left(\frac{\boldsymbol{\eta}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t}, \boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega}, \quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{Y}_{\mathcal{N}}^0.\end{aligned} \tag{4.11}$$

Similarly, for any $\boldsymbol{v}_{1,\mathcal{N}}(x) \in \mathbb{X}_{\mathcal{N}}^0$, multiplying (4.5) by $\boldsymbol{v}_{1,\mathcal{N}}(x_p^i) \in X_{\mathcal{N}}^0$, and performing the LGL quadrature in the x-direction yield: $\forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{Y}_{\mathcal{N}}^0$,

$$\left(\frac{\boldsymbol{u}_{\mathcal{N}}^{n+1} - \boldsymbol{\eta}_{\mathcal{N}}^{n+1}}{\Delta t}, \boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega} + \frac{1}{2}\nu\big(\tilde{\partial}_y\big(\boldsymbol{u}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n}\big), \tilde{\partial}_y\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} = \boldsymbol{0}. \tag{4.12}$$

Summing (4.12) with (4.11) gives

$$\begin{aligned}&-\frac{\nu}{2}\big(\tilde{\partial}_x\big(\boldsymbol{\eta}_{\mathcal{N}}^{n+1} + \boldsymbol{u}_{\mathcal{N}}^{n}\big), \tilde{\partial}_x\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} - \frac{\nu}{2}\big(\tilde{\partial}_y\big(\boldsymbol{u}_{\mathcal{N}}^{n+1} + \boldsymbol{u}_{\mathcal{N}}^{n}\big), \tilde{\partial}_y\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} - \big(\nabla p_{\mathcal{N}}^{*,n+\frac{1}{2}}, \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}\\ &\quad= \left(\frac{\boldsymbol{u}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t}, \boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega}, \quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{Y}_{\mathcal{N}}^0.\end{aligned} \tag{4.13}$$

Now we subtract (4.13) from (4.11) to obtain

$$\big(\boldsymbol{\eta}_{\mathcal{N}}^{n+1}, \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} = \big(\boldsymbol{u}_{\mathcal{N}}^{n+1}, \boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega} + \frac{\Delta t\nu}{2}\big(\tilde{\partial}_y\big(\boldsymbol{u}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n}\big), \tilde{\partial}_y\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}, \quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{Y}_{\mathcal{N}}^0. \tag{4.14}$$

Note that both functions $\boldsymbol{\eta}_{\mathcal{N}}^{n+1}$ and $\boldsymbol{u}_{\mathcal{N}}^{n+1} + \frac{\Delta t\nu}{2}\tilde{\partial}_y(\boldsymbol{u}_{\mathcal{N}}^{n+1} - \boldsymbol{u}_{\mathcal{N}}^{n})$ are continuous piecewise polynomials of degree $\le N$ with respect to x, and vanishing at the two sides $x = 0, R$. Thus from (4.14), we obtain

$$\begin{aligned}\big(\boldsymbol{\eta}_{\mathcal{N}}^{n+1}(x,\cdot), \boldsymbol{v}_{2,\mathcal{N}}\big)_{\mathcal{N},\Lambda} &= \big(\boldsymbol{u}_{\mathcal{N}}^{n+1}(x,\cdot), \boldsymbol{v}_{2,\mathcal{N}}\big)_{\mathcal{N},\Lambda}\\ &\quad+ \frac{\Delta t\nu}{2}\big(\tilde{\partial}_y\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}(x,\cdot) - \boldsymbol{u}_{\mathcal{N}}^{n}(x,\cdot)\big), \tilde{\partial}_y\boldsymbol{v}_{2,\mathcal{N}}\big)_{\mathcal{N},\Lambda}, \quad \forall x \in \Lambda,\ \forall \boldsymbol{v}_{2,\mathcal{N}} \in \mathbb{X}_{\mathcal{N}}^0,\end{aligned} \tag{4.15}$$

where $(\cdot,\cdot)_{\mathcal{N},\Lambda}$ means 1D-LGL quadrature in the y-direction.

Taking the second derivative of both sides of (4.15) with respect to x, then rewriting back the resulting equation into the weak form by using Lemma 2.1, we get

$$\begin{aligned}\big(\tilde{\partial}_x\boldsymbol{\eta}_{\mathcal{N}}^{n+1}(x,\cdot), \tilde{\partial}_x\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Lambda} &= \big(\tilde{\partial}_x\boldsymbol{u}_{\mathcal{N}}^{n+1}(x,\cdot), \tilde{\partial}_x\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Lambda}\\ &\quad+ \frac{\Delta t\nu}{2}\big(\tilde{\partial}_{xy}\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}(x,\cdot) - \boldsymbol{u}_{\mathcal{N}}^{n}(x,\cdot)\big), \tilde{\partial}_{xy}\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Lambda}, \quad \forall \boldsymbol{v}_{\mathcal{N}} \in \mathbb{Y}_{\mathcal{N}}^0.\end{aligned} \tag{4.16}$$

Now we plug (4.16) into (4.13) to get

$$\left(\frac{\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}}{\Delta t},\boldsymbol{v}_{\mathcal{N}}\right)_{\mathcal{N},\Omega}+\nu\big(\tilde{\nabla}\bar{\boldsymbol{u}}_{\mathcal{N}}^{n+\frac{1}{2}},\tilde{\nabla}\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}+\big(\nabla p_{\mathcal{N}}^{*,n+\frac{1}{2}},\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}+\frac{\Delta t\nu^2}{4}\big(\tilde{\partial}_{xy}\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\big),\tilde{\partial}_{xy}\boldsymbol{v}_{\mathcal{N}}\big)_{\mathcal{N},\Omega}$$
$$=\boldsymbol{0},\quad\forall\boldsymbol{v}_{\mathcal{N}}\in\mathbb{Y}_{\mathcal{N}}^{0}.\tag{4.17}$$

Taking $\boldsymbol{v}_{\mathcal{N}}=\boldsymbol{u}_{\mathcal{N}}^{n+1}$ in (4.17), then using the identity, $2(a-b,a)=\|a\|^2+\|a-b\|^2-\|b\|^2$, we find

$$\begin{aligned}&\|\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2+\|\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2+\frac{\Delta t\nu}{2}\big(\|\tilde{\nabla}\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2+\|\tilde{\nabla}\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}+\boldsymbol{u}_{\mathcal{N}}^{n}\big)\|_{\mathcal{N}}^2\big)\\&\quad+2\Delta t\big(\nabla p_{\mathcal{N}}^{*,n+\frac{1}{2}},\boldsymbol{u}_{\mathcal{N}}^{n+1}\big)_{\mathcal{N}}+\frac{\Delta t^2\nu^2}{4}\|\tilde{\partial}_{xy}\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2+\frac{\Delta t^2\nu^2}{4}\|\tilde{\partial}_{xy}\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\big)\|_{\mathcal{N}}^2\\&=\|\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2+\frac{\Delta t\nu}{2}\|\tilde{\nabla}\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2+\frac{\Delta t^2\nu^2}{4}\|\tilde{\partial}_{xy}\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2.\end{aligned}\tag{4.18}$$

On the other hand, from (4.9) and (4.3), the function $p_{\mathcal{N}}^{n+\frac{1}{2}}-p_{\mathcal{N}}^{n-\frac{1}{2}}\in Y_{\mathcal{N}}(\Omega)$ satisfies: for $n\geq 0$,

$$a_{\mathcal{N}}\big(p_{\mathcal{N}}^{n+\frac{1}{2}}-p_{\mathcal{N}}^{n-\frac{1}{2}},q_{\mathcal{N}}\big)=-\frac{1}{\Delta t}\big(\nabla\cdot\boldsymbol{u}_{\mathcal{N}}^{n+1},q_{\mathcal{N}}\big)_{\mathcal{N}},\quad\forall q_{\mathcal{N}}\in Y_{\mathcal{N}}(\Omega).\tag{4.19}$$

Taking $q_{\mathcal{N}}=2\Delta t^2p_{\mathcal{N}}^{*,n+\frac{1}{2}}:=2\Delta t^2\big(2p_{\mathcal{N}}^{n-\frac{1}{2}}-p_{\mathcal{N}}^{n-\frac{3}{2}}\big)$ in the above, and using the fact that $a_{\mathcal{N}}(\cdot,\cdot)$ is symmetric, we have

$$\begin{aligned}&-2\Delta t\big(\nabla\cdot\boldsymbol{u}_{\mathcal{N}}^{n+1},p_{\mathcal{N}}^{*,n+\frac{1}{2}}\big)_{\mathcal{N}}\\&=2\Delta t^2a_{\mathcal{N}}\big(\delta p_{\mathcal{N}}^{n+\frac{1}{2}},2p_{\mathcal{N}}^{n-\frac{1}{2}}-p_{\mathcal{N}}^{n-\frac{3}{2}}\big)\\&=2\Delta t^2a_{\mathcal{N}}\big(\delta p_{\mathcal{N}}^{n+\frac{1}{2}},p_{\mathcal{N}}^{n+\frac{1}{2}}\big)+2\Delta t^2a_{\mathcal{N}}\big(\delta p_{\mathcal{N}}^{n+\frac{1}{2}},-\delta p_{\mathcal{N}}^{n+\frac{1}{2}}+\delta p_{\mathcal{N}}^{n-\frac{1}{2}}\big)\\&=\Delta t^2\big(\|p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2-\|p_{\mathcal{N}}^{n-\frac{1}{2}}\|_{A_{\mathcal{N}}}^2+\|\delta p_{\mathcal{N}}^{n-\frac{1}{2}}\|_{A_{\mathcal{N}}}^2-\|\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2\big).\end{aligned}\tag{4.20}$$

Next we seek to bound the term $\|\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2$. Applying the time increment operator δ to (4.19) and taking $q_{\mathcal{N}}=\Delta t\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}$ in the resulting equation, we obtain

$$\begin{aligned}\Delta t\|\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2&=-\big(\nabla\cdot\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\big),\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\big)_{\mathcal{N}}\\&=\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n},\nabla\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\big)_{\mathcal{N}}\\&\leq\|\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}\|\nabla\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{\mathcal{N}}.\end{aligned}$$

Then by using the coercivity (4.2), we obtain

$$\Delta t\|\nabla\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{\mathcal{N}}^2\leq\|\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}\|\nabla\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}},$$

which furthermore results in

$$\Delta t^2\|\delta^2p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2\leq\|\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2.$$

Inserting this bound into (4.20) gives

$$\Delta t^2\big(\|p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2+\|\delta p_{\mathcal{N}}^{n-\frac{1}{2}}\|_{A_{\mathcal{N}}}^2-\|p_{\mathcal{N}}^{n-\frac{1}{2}}\|_{A_{\mathcal{N}}}^2\big)\leq\|\boldsymbol{u}_{\mathcal{N}}^{n+1}-\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2-2\Delta t\big(\nabla\cdot\boldsymbol{u}_{\mathcal{N}}^{n+1},p_{\mathcal{N}}^{*,n+\frac{1}{2}}\big)_{\mathcal{N}}.$$

Then, combining the above estimate with (4.18), noticing that

$$\big(\nabla\cdot\boldsymbol{u}_{\mathcal{N}}^{n+1},p_{\mathcal{N}}^{*,n+\frac{1}{2}}\big)_{\mathcal{N}}=-\big(\nabla p_{\mathcal{N}}^{*,n+\frac{1}{2}},\boldsymbol{u}_{\mathcal{N}}^{n+1}\big)_{\mathcal{N}},$$

and dropping some unnecessary positive terms, we arrive at

$$\begin{aligned}&\|\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2+\frac{\Delta t\nu}{2}\|\tilde{\nabla}\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2+\Delta t^2\|p_{\mathcal{N}}^{n+\frac{1}{2}}\|_{A_{\mathcal{N}}}^2+\frac{\Delta t^2\nu^2}{4}\|\tilde{\partial}_{xy}\boldsymbol{u}_{\mathcal{N}}^{n+1}\|_{\mathcal{N}}^2+\frac{\Delta t\nu}{2}\|\tilde{\nabla}\big(\boldsymbol{u}_{\mathcal{N}}^{n+1}+\boldsymbol{u}_{\mathcal{N}}^{n}\big)\|_{\mathcal{N}}^2\\&\leq\|\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2+\frac{\Delta t\nu}{2}\|\tilde{\nabla}\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2+\Delta t^2\|p_{\mathcal{N}}^{n-\frac{1}{2}}\|_{A_{\mathcal{N}}}^2+\frac{\Delta t^2\nu^2}{4}\|\tilde{\partial}_{xy}\boldsymbol{u}_{\mathcal{N}}^{n}\|_{\mathcal{N}}^2.\end{aligned}$$

Thus the proof is completed. □

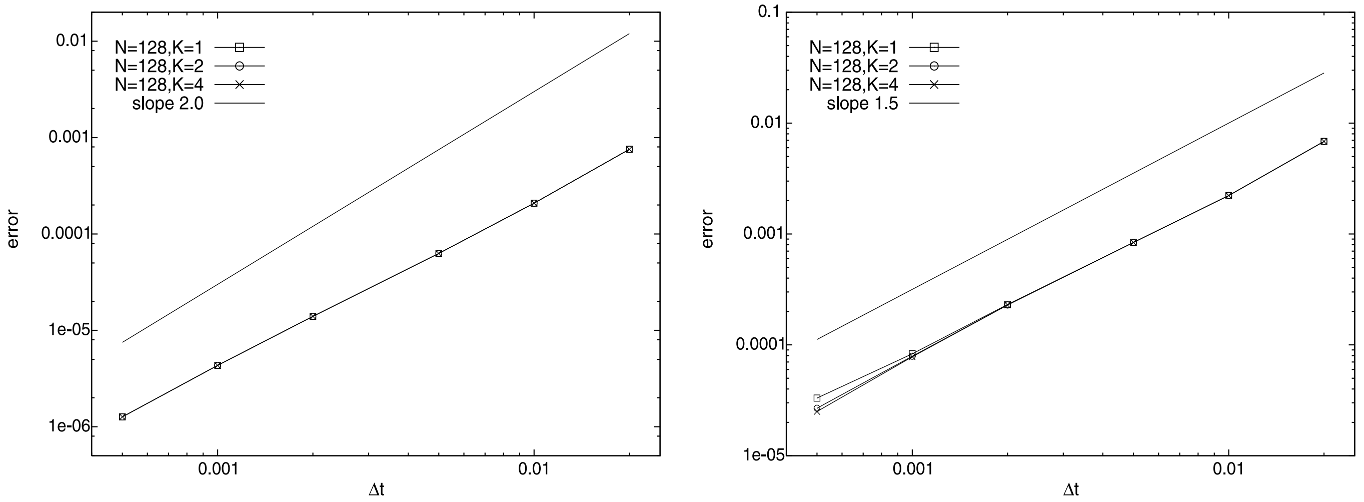

Fig. 5.1. Velocity (left) and pressure (right) error in L^2-norm as a function of Δt.

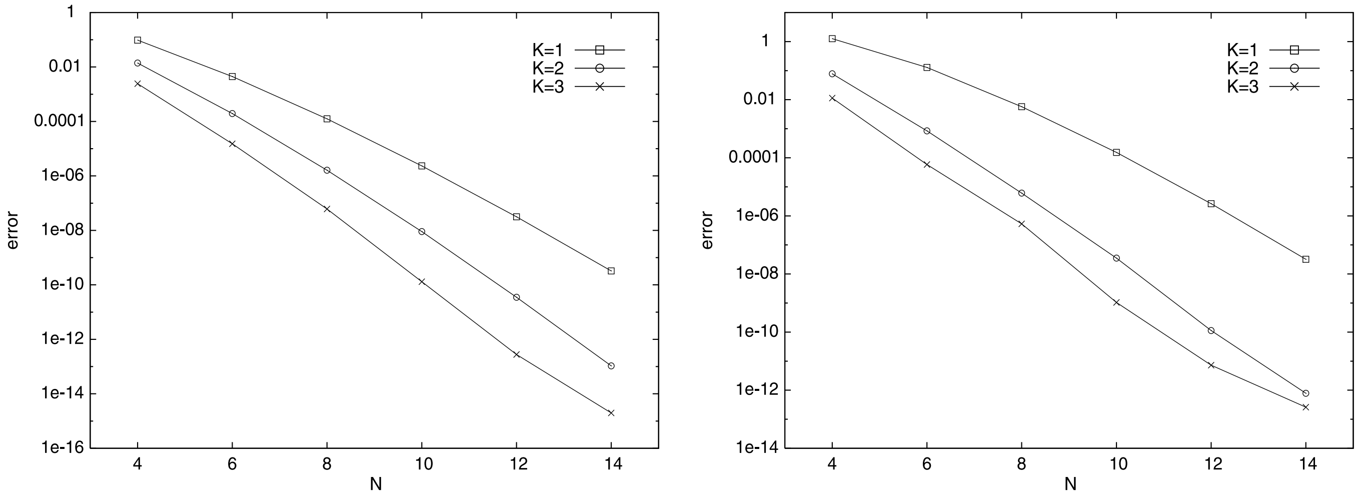

Fig. 5.2. Velocity (left) and pressure (right) error in L^2-norm as a function of N for several K.

5. Numerical results and discussions

We present below some numerical experiments to verify the stability, accuracy and the parallel efficiency of the proposed scheme with the fixed parameter $\chi = 0.5$.

Example 5.1 *(Time accuracy, spectral accuracy, and parallel efficiency in 2-D).* We consider the 2D time-dependent Navier–Stokes equations (2.1) with the exact solution:

$$\begin{cases} \boldsymbol{u}(x,y) = \pi \sin t\big(\sin^2 \pi x \sin 2\pi y, -\sin 2\pi x \sin^2 \pi y\big), \\ p(x,y,t) = \sin t \cos \pi x \sin \pi y. \end{cases} \tag{5.1}$$

First we investigate the time accuracy, i.e., the direction splitting error. We plot, in Fig. 5.1, the L^2-velocity errors (the left figure) and the L^2-pressure errors (the right figure) at $T = 1$ w.r.t. Δt in log–log scale, with several different (N, K) large enough such that the spatial discretization errors are negligible compared with the time discretization errors. The figures show that our algorithm is of second order accuracy in the L^2-norm for the velocity and 1.7 order accuracy in the L^2-norm for the pressure. These results are clearly consistent with Theorem 3.1 [3], which concluded that the splitting error is of second order for the velocity and one half order for pressure in L^2-norm respectively.

Then we investigate the space accuracy by checking the convergence behavior of numerical solutions with respect to the polynomial degrees N. We plot, in semi-log scale, the L^2-velocity errors in Fig. 5.2 (left) and the L^2-pressure errors (right) at $T = 1$ with $\Delta t = 0.0001$ for several different K. As expected, the errors show an exponential decay, since in this semi-log representation one observes that the error variations are linear versus the degrees of polynomial N.

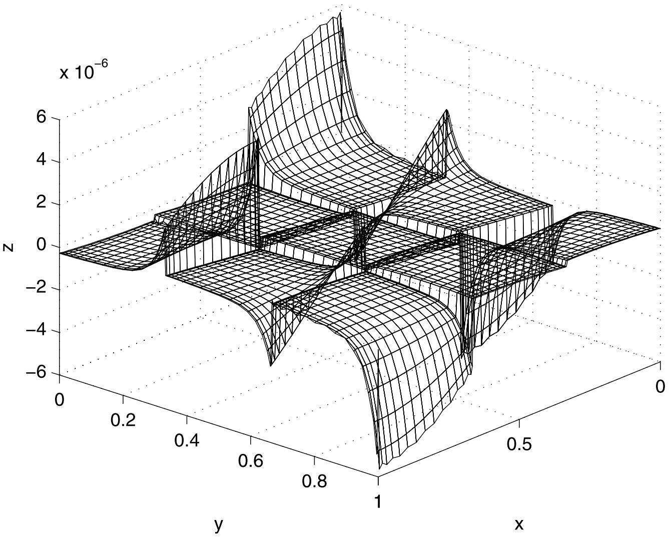

Fig. 5.3. Divergence of the computed solution with $N = 16$, $K = 3$, $\Delta t = 0.0001$ at $t = 1$.

Table 5.1
Parallel performance: CPU time (in second) and parallel efficiency per time step.

N	32		64		128		256	
# procs	running time	parallel efficiency	running time	parallel efficiency	running time	parallel efficiency	running time	parallel efficiency
1 × 1	0.001691	–	0.006879	–	0.03710	–	0.2225	–
4 × 4	0.001934	0.8714	0.007587	0.9066	0.03934	0.9410	0.2342	0.9500
8 × 8	0.002185	0.7736	0.007923	0.8681	0.04027	0.9192	0.2366	0.9403
16 × 16	0.003064	0.5517	0.009535	0.7214	0.04080	0.9095	0.2414	0.9218
32 × 32	0.004842	0.3491	0.01212	0.5675	0.04804	0.7724	0.2653	0.8386

We plot, in Fig. 5.3, the divergence of the computed solution at $t = 1$. One observes that the error of the divergence at the subdomain interfaces and domain boundary is noticeably larger but the overall error of the divergence is satisfactorily small.

Before we move to the parallel efficiency, note that we used both P_N for the velocity and pressure approximation. It is well-known that $P_N \times P_N$ does not satisfy the inf–sup condition for the Stokes problem. While this choice does not affect the stability for the splitting/projection schemes and convergence rate for the velocity, it does affect the convergence rate when Δt is very small, we refer to [4] (cf. also [1]) for additional discussion in this regard.

Finally we examine the parallel efficiency of the scheme. We report in Table 5.1 the parallel efficiency in weak scaling. Here the weak scaling means that the problem size assigned to each processor stays constant and additional processors are used to solve a larger total problem. One observes from Table 5.1 that as we increase N, polynomial degrees in each direction of each subdomain, the parallel efficiency increases; and as we increase the number of processors (so the number of subdomains), the parallel efficiency decreases slightly.

Example 5.2 *(Parallel efficiency in 3-D).* We now consider the 3-D case with the following exact solution:

$$\begin{cases} \boldsymbol{u}(x,y,z,t) = \pi \sin t \begin{pmatrix} \sin^2 \pi x \sin 2\pi y \sin 2\pi z \\ \sin 2\pi x \sin^2 \pi y \sin 2\pi z \\ -2 \sin 2\pi x \sin 2\pi y \sin^2 \pi z \end{pmatrix}, \\ p(x,y,z,t) = \sin t \cos \pi x \cos \pi y \sin \pi z. \end{cases} \tag{5.2}$$

In Table 5.2, we report the parallel efficiency in weak scaling for the three dimensional case. One observes essentially the same behaviors as in the two dimensional case with a slightly decreased parallel efficiency.

6. Summary

We developed a spectral-element direction splitting scheme for the incompressible Navier–Stokes equations, and presented an efficient parallel implementation. Our numerical results indicated that the proposed scheme enjoys the same accuracy as the (one-domain) spectral direction splitting scheme in [1], while achieving a high level of parallel efficiency in weak scaling for both two and three dimensional problems.

Table 5.2
Parallel performance: CPU time (in second) and parallel efficiency in 3-D.

N	16		32		64		128	
# procs	running time	parallel efficiency	running time	parallel efficiency	running time	parallel efficiency	running time	parallel efficiency
$1 \times 1 \times 1$	0.01667	–	0.1141	–	1.0305	–	11.2867	–
$4 \times 4 \times 4$	0.02058	0.8101	0.1326	0.8608	1.3743	0.7498	13.8449	0.8152
$6 \times 6 \times 6$	0.02478	0.6729	0.1417	0.8057	1.4103	0.7307	14.2834	0.7902
$8 \times 8 \times 8$	0.02865	0.5819	0.1575	0.7247	1.4643	0.7037	15.1162	0.7467
$10 \times 10 \times 10$	0.03090	0.5396	0.1647	0.6929	1.4736	0.6992	16.7231	0.6749

We believe that the method presented in this paper have great potential for large scale CFD simulations. It can also be extended to treat multiphase flows and/or complex fluids for which the Navier–Stokes solver plays an important role. Consider, for example, the Allen–Cahn or Cahn–Hilliard Navier–Stokes phase-field models for the two-phase incompressible and immiscible flows (cf. [16,12]), where the governing system consists of an Allen–Cahn or Cahn–Hilliard phase equation and Navier–Stokes equations coupled together by the transport term in the phase equation and the surface tensional term in the momentum equation. If we use a semi-implicit time discretization where all linear terms are treated implicitly and all nonlinear terms explicitly, it leads to a sequence of second-order Poisson type equations[1] to solve at each time step. Hence, the parallel algorithm presented in this paper can be directly applied.

Acknowledgement

The authors would like to thank Jean-Luc Guermond and Peter Minev for stimulating discussions and for providing us with their finite difference parallel code.

References

[1] L. Chen, J. Shen, C. Xu, Spectral direction splitting schemes for the incompressible Navier–Stokes equations, East Asian J. Appl. Math. 1 (2011) 215–234.
[2] M.O. Deville, P.F. Fischer, E.H. Mund, High-order Methods for Incompressible Fluid Flow, vol. 9, Cambridge Univ. Press, 2002.
[3] Jean-Luc Guermond, Peter D. Minev, A new class of fractional step techniques for the incompressible Navier–Stokes equations using direction splitting, C. R. Math. Acad. Sci. Paris 348 (9–10) (2010) 581–585.
[4] J.L. Guermond, P. Minev, Jie Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (2006) 6011–6045.
[5] J.L. Guermond, P.D. Minev, A new class of fractional step techniques for the incompressible Navier–Stokes equations using direction splitting, C. R. Math. 348 (9) (2010) 581–585.
[6] J.L. Guermond, P.D. Minev, Start-up flow in a three-dimensional lid-driven cavity by means of a massively parallel direction splitting algorithm, Int. J. Numer. Methods Fluids 68 (2012) 856–871.
[7] J.L. Guermond, P.D. Minev, A.J. Salgado, Convergence analysis of a class of massively parallel direction splitting algorithms for the Navier–Stokes equations, preprint, 2011.
[8] J.L. Guermond, Jie Shen, On the error estimates for the rotational pressure-correction projection methods, Math. Comput. 73 (248) (2004) 1719–1737 (electronic).
[9] G. Karniadakis, S.J. Sherwin, Spectral/hp Element Methods for CFD, Oxford University Press, USA, 1999.
[10] George E. Karniadakis, Spencer J. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, Oxford University Press, 2005.
[11] Y.Y. Kwan, J. Shen, An efficient direct parallel spectral-element solver for separable elliptic problems, J. Comput. Phys. 225 (2) (2007) 1721–1735.
[12] Chun Liu, Jie Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, Physica D 179 (3–4) (2003) 211–228.
[13] Anthony T. Patera, Fast direct Poisson solvers for high-order finite element discretizations in rectangularly decomposable domains, J. Comput. Phys. 65 (2) (August 1986) 474–480.
[14] D.W. Peaceman, H.H. Rachford Jr., The numerical solution of parabolic and elliptic differential equations, J. Soc. Ind. Appl. Math. 3 (1955) 28–41.
[15] J. Shen, Efficient spectral-Galerkin method I. Direct solvers of second-and fourth-order equations using Legendre polynomials, SIAM J. Sci. Comput. 15 (6) (1994) 1489–1508.
[16] Jie Shen, Haijun Yu, Efficient spectral sparse grid methods and applications to high dimensional elliptic problems, SIAM J. Sci. Comput. 32 (2010) 3228–3250.
[17] L.J.P. Timmermans, P.D. Minev, F.N. Van De Vosse, An approximate projection scheme for incompressible flow using spectral elements, Int. J. Numer. Methods Fluids 22 (1996) 673–688.
[18] H.M. Tufo, P.F. Fischer, Terascale spectral element algorithms and implementations, in: Supercomputing, ACM/IEEE 1999 Conference, IEEE, 1999, p. 68.
[19] P. Yue, J.J. Feng, C. Liu, J. Shen, A diffuse-interface method for simulating two-phase flows of complex fluids, J. Fluid Mech. 515 (2004) 293–317.

[1] For the Cahn–Hilliard model, the fourth-order equation can be rewritten as a coupled second-order system which can be further reduced to two decoupled second-order equations (cf. [19]).