

Journal Pre-proof

A Bound-Preserving High Order Scheme for Variable Density Incompressible Navier-Stokes Equations

Maojun Li, Yongping Cheng, Jie Shen and Xiangxiong Zhang

PII: S0021-9991(20)30680-X
DOI: <https://doi.org/10.1016/j.jcp.2020.109906>
Reference: YJCPH 109906

To appear in: *Journal of Computational Physics*

Received date: 13 June 2020
Revised date: 22 September 2020
Accepted date: 4 October 2020

Please cite this article as: M. Li, Y. Cheng, J. Shen et al., A Bound-Preserving High Order Scheme for Variable Density Incompressible Navier-Stokes Equations, *Journal of Computational Physics*, 109906, doi: <https://doi.org/10.1016/j.jcp.2020.109906>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier.



Highlights

- This is a high order accurate and strictly bound-preserving scheme for variable density incompressible flows.
- The spatial discretization includes a high order discontinuous Galerkin method for density and continuous finite element method for velocity field.
- A constant Poisson equation for stream function is solved efficiently to enforce local incompressibility. To recover pressure, the variable coefficient pressure Poisson equation with Neumann boundary conditions is solved by Laplacian preconditioned conjugate gradient method.
- The time discretization is a third order implicit-explicit (IMEX) SSP Runge-Kutta method. When applied to density evolution, it is equivalent to the third order TVD Runge-Kutta method thus bound-preserving is achieved through convex combination of forward Euler steps. On the other hand, diffusion term in the momentum equation is treated implicitly thus small time steps are avoided.

A Bound-Preserving High Order Scheme for Variable Density Incompressible Navier-Stokes Equations

Maojun Li^a, Yongping Cheng^b, Jie Shen^c, Xiangxiong Zhang^{c,*}

^a*School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, P.R. China*

^b*College of Mathematics and Statistics, Chongqing University, Chongqing, 401331, P.R. China*

^c*Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA*

Abstract

For numerical schemes to the incompressible Navier-Stokes equations with variable density, it is a critical property to preserve the bounds of density. A bound-preserving high order accurate scheme can be constructed by using high order discontinuous Galerkin (DG) methods or finite volume methods with a bound-preserving limiter for the density evolution equation, with any popular numerical method for the momentum evolution. In this paper, we consider a combination of a continuous finite element method for momentum evolution and a bound-preserving DG method for density evolution. Fully explicit and explicit-implicit strong stability preserving Runge-Kutta methods can be used for the time discretization for the sake of bound-preserving. Numerical tests on representative examples are shown to demonstrate the performance of the proposed scheme.

Keywords: Variable density incompressible flows, Navier-Stokes equations, discontinuous Galerkin method, bound-preserving scheme, finite element method.

1. Introduction

The Navier-Stokes equations governing incompressible viscous flows with variable density are given as

$$\begin{cases} \rho_t + \mathbf{u} \cdot \nabla \rho = 0, & (1.1a) \\ \rho(\mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u}) + \nabla p - \nabla \cdot (\mu(\rho) \nabla \mathbf{u}) = \mathbf{f}, & (1.1b) \\ \nabla \cdot \mathbf{u} = 0, & (1.1c) \end{cases}$$

where the unknowns are the density ρ , the velocity field \mathbf{u} and the pressure p . The coefficient $\mu(\rho)$ models the dynamic viscosity, and \mathbf{f} represents the external force, e.g., gravity. For simplicity, we consider the following initial and boundary conditions on a two-dimensional bounded domain Ω :

$$\begin{cases} \rho(\mathbf{x}, 0) = \rho_0(\mathbf{x}), \quad \rho(\mathbf{x}, t)|_{\Gamma} = a(\mathbf{x}, t), \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{u}(\mathbf{x}, t)|_{\Gamma} = \mathbf{0}, \end{cases} \quad (1.1d)$$

where $\Gamma = \partial\Omega$ is boundary of Ω . We note that no initial and boundary conditions are needed for the pressure p which can be viewed as a Lagrange multiplier to enforce the incompressibility condition (1.1c). We refer to [22] for the mathematical theory on the well-posedness of (1.1).

For Navier-Stokes equations with constant density, the numerical schemes have been well studied, e.g., see [8] and [10] and the references therein. However, the Navier-Stokes equations with variable density (1.1) involves a density equation which is a purely convection equation thus oscillations may occur near the sharp interface. It is crucial to maintain the physical bound of density to ensure numerical stability. The

*Corresponding author.

Email addresses: limj@uestc.edu.cn (Maojun Li), cyp@cqu.edu.cn (Yongping Cheng), shen@math.purdue.edu (Jie Shen), zhan1966@purdue.edu (Xiangxiong Zhang)

main objective of this paper is to construct a high order bound-preserving scheme for this variable-density incompressible fluid system.

For solving variable-density incompressible Navier-Stokes equations (1.1), established numerical methods for incompressible flows can be used. One popular approach is the fractional time-stepping method or implicit projection method [5, 10], which can effectively handle the difficulties associated with the incompressibility constraint. This approach has been developed and used in [2, 3, 11, 12, 21, 25], among others, for incompressible fluid flows with variable density. In [11] Guermond and Quartapelle gave the stability analysis for any projection type scheme with variable density. Nonetheless, the variable density introduces considerable difficulties for the construction and analysis of accurate and stable projection type schemes. On the other hand, the system (1.1) can also be solved directly by a Runge-Kutta method with finite element method or a non-variational method such as spectral-collocation method for an equivalent pressure Poisson equation formulation. In [17] Johnston and Liu presented such a second order finite difference scheme based on local pressure boundary conditions for time-dependent viscous incompressible flows with variable density for moderate to large Reynolds number simulations.

For a scalar convection problem, a high order finite volume or a high order DG scheme can be easily rendered bound-preserving using a simple bound-preserving limiter [32, 33, 34, 35]. Thus density can be ensured bound-preserving if using a high order bound-preserving DG method for (1.1a). For (1.1b), we can also use a DG method, which however induces unnecessarily more degree of freedoms than a continuous finite element method. So we will explore a combination of a bound-preserving discontinuous Galerkin method solving the density equation and a finite element method solving momentum equation for variable-density incompressible Navier-Stokes equations. For instance, for using Q^k elements on rectangular meshes, DG can be implemented as a nodal DG scheme [13, 19] and continuous finite element method can be implemented as a variational finite difference scheme [20], both of which have degree of freedoms defined on Gauss-Lobatto points of the rectangular cells thus can be easily combined. For the sake of bound-preserving, a high order strong stability preserving (SSP) time discretization [9] must be used for the time discretization. If considering moderate to large Reynolds number flows, i.e., convection-dominated flows, a fully explicit SSP Runge-Kutta method is suitable. For large viscosity flows, fully explicit time discretization will impose small time steps for the sake of linear stability. In this case, implicit-explicit (IMEX) SSP Runge-Kutta schemes in [24] can be used to both ensure bound-preserving of density and to avoid small time steps for low Reynolds number flows.

The rest of the paper is organized as follows. In Section 2, we introduce variable density incompressible Navier-Stokes equations and equivalent pressure Poisson equation formulations. The numerical scheme is presented in Section 3. In Section 4, we perform representative numerical tests to explore the performance of the proposed scheme. Concluding remarks are given in Section 5.

2. Mathematical formulation

One of the challenges for numerically solving the incompressible Navier-Stokes equations is how to ensure a divergence-free flow field and recover the pressure from the velocity. Following [17], we first take the divergence of the momentum equation (1.1b), then along with the incompressibility constraint (1.1c) we get

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \nabla \cdot \left(-(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla \cdot (\mu(\rho) \nabla \mathbf{u}) + \frac{\mathbf{f}}{\rho} \right). \quad (2.1)$$

This equation gives the evolution of p provided that we know the evolution of ρ and \mathbf{u} . A proper boundary condition for p is needed for solving (2.1). A natural candidate is given by the normal component of the momentum equation (1.1b) along Γ :

$$\frac{\partial p}{\partial \mathbf{n}} \Big|_{\Gamma} = [\nabla \cdot (\mu(\rho) \nabla (\mathbf{u} \cdot \mathbf{n})) + \mathbf{f} \cdot \mathbf{n}] \Big|_{\Gamma}. \quad (2.2)$$

On the other hand, with suitable boundary conditions, the exact solution of (1.1a) satisfies the maximum principle $\rho(x, y, t) \in [m, M]$, for all (x, y, t) , where $m = \min_{x,y} \rho_0(x, y)$ and $M = \max_{x,y} \rho_0(x, y)$. For

discontinuous solutions or solutions containing sharp gradient regions, it is preferable to solve the following equivalent conservative form

$$\rho_t + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (2.3)$$

rather than the non-conservative form (1.1a). However, the solution to (2.3) is not necessarily bound-preserving unless the velocity field is incompressible, i.e., $\nabla \cdot \mathbf{u} = 0$. It is usually much easier to construct a bound-preserving scheme solving the non-conservative form (1.1a) but the conservation would be difficult to preserve. For solving the conservative form (2.3) coupled with $\nabla \cdot \mathbf{u} = 0$, a high order accurate DG scheme with a bound-preserving limiter was developed in [32, 34].

Therefore, we will consider solving an equivalent pressure Poisson equation formulation of (1.1):

$$\begin{cases} \rho_t + \nabla \cdot (\rho \mathbf{u}) = 0, & (2.4a) \\ \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla p - \frac{1}{\rho} \nabla \cdot (\mu(\rho) \nabla \mathbf{u}) = \frac{1}{\rho} \mathbf{f}, & (2.4b) \\ \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \nabla \cdot \left(-(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla \cdot (\mu(\rho) \nabla \mathbf{u}) + \frac{\mathbf{f}}{\rho} \right), & (2.4c) \\ \nabla \cdot \mathbf{u} = 0, & (2.4d) \end{cases}$$

with the following initial and boundary conditions for ρ , \mathbf{u} and p :

$$\begin{cases} \rho(\mathbf{x}, 0) = \rho_0(\mathbf{x}), \quad \rho(\mathbf{x}, t)|_{\Gamma} = a(\mathbf{x}, t), \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{u}(\mathbf{x}, t)|_{\Gamma} = 0, \\ \frac{\partial p}{\partial \mathbf{n}}|_{\Gamma} = [\nabla \cdot (\mu(\rho) \nabla (\mathbf{u} \cdot \mathbf{n})) + \mathbf{f} \cdot \mathbf{n}]|_{\Gamma}. \end{cases} \quad (2.4e)$$

3. Numerical methods

In this section, we describe the numerical methods for the system (2.4). For simplicity, we only consider a rectangular domain Ω , discretized by Ω_h consisting of uniform rectangular cells. Extension to nonuniform rectangular meshes is straightforward. Consider polynomial approximation spaces on rectangular cells

$$V_h^k = \{v : v|_K \in Q^k(K), \forall K \in \Omega_h\}, \quad W_h^k = V_h^k \cap C_0(\Omega_h), \quad W_{0,h}^k = \{u|u \in W_h^k, u|_{\Gamma} = 0\},$$

$$\mathbf{V}_h^k = \{\mathbf{u} = (u, v) | u, v \in V_h^k\}, \quad \mathbf{W}_{0,h}^k = \{\mathbf{u} = (u, v) | u, v \in W_{0,h}^k\},$$

where Q^k refers to the space of tensor products of 1D polynomials of degree k . A bound-preserving discontinuous Galerkin scheme [32, 35, 34] will be used for the density evolution (2.4a) and a finite element method [20] will be used for the velocity evolution (2.4b) and pressure (2.4c).

3.1. A high order accurate DG scheme for the density evolution

We first consider a high order accurate DG scheme for the equation (2.4a), following the approach of constructing bound-preserving high order schemes for passive convection with an incompressible velocity field in [32, 35, 34]. For given velocity field $\mathbf{u}_h \in \mathbf{V}_h^k$, on a rectangular cell K , the DG solution $\rho_h \in V_h^k$ satisfies

$$\int_K \partial_t \rho_h \phi_h dx dy - \int_K \rho_h \mathbf{u}_h \cdot \nabla \phi_h dx dy + \sum_{e \in \partial K} \int_e \widehat{\rho_h \mathbf{u}_h \cdot \mathbf{n}} \phi_h ds = 0, \quad \forall \phi_h \in V_h^k, \quad (3.1)$$

where \mathbf{n} denotes the unit outward normal vector to ∂K and $\widehat{\rho_h \mathbf{u}_h \cdot \mathbf{n}}$ is the numerical flux.

To construct a bound-preserving scheme as in [32, 35, 34], it would be convenient to require the velocity field \mathbf{u}_h to satisfy two constraints:

1. The incompressibility $\nabla \cdot \mathbf{u}_h = 0$ holds everywhere inside the cell K .
2. The normal velocity $\mathbf{u}_h \cdot \mathbf{n}$ is continuous across cell boundaries ∂K .

These two constraints can be easily satisfied if the stream function ψ_h is available. Namely, if we compute the velocity field by $\mathbf{u}_h = \langle u_h, v_h \rangle = \left\langle -\frac{\partial \psi_h}{\partial y}, \frac{\partial \psi_h}{\partial x} \right\rangle$ from some stream function $\psi_h \in W_{0,h}^k$. Notice that we have the normal velocity $\mathbf{u}_h \cdot \mathbf{n} = \left\langle -\frac{\partial \psi_h}{\partial y}, \frac{\partial \psi_h}{\partial x} \right\rangle \cdot \mathbf{n} = \frac{\partial \psi_h}{\partial \tau}$ where τ denote the unit vector tangent to ∂K , thus $\mathbf{u}_h \cdot \mathbf{n}$ is continuous across any element boundary ∂K since ψ_h is continuous. For now we assume $\psi_h \in W_{0,h}^k$ is given and we will discuss how to obtain it in Section 3.5.

Assume the rectangular cell K can be denoted as $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$, then we can also refer to the cell K as (i, j) cell. Let $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ and $\Delta y = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$. At time level n , in (i, j) cell, let $\rho_{i-\frac{1}{2},j}^+(y)$, $\rho_{i+\frac{1}{2},j}^-(y)$, $\rho_{i,j-\frac{1}{2}}^+(x)$, and $\rho_{i,j+\frac{1}{2}}^-(x)$ denote the traces of the DG polynomial $\rho_{i,j}(x, y)$ on the left, right, bottom and top edges, respectively. On the left, right, bottom and top edges, $\mathbf{u}_h \cdot \mathbf{n}$ is $u_{i-\frac{1}{2},j}(y)$, $u_{i+\frac{1}{2},j}(y)$, $v_{i,j-\frac{1}{2}}(x)$, and $v_{i,j+\frac{1}{2}}(x)$, respectively. Since the normal velocity is continuous, we can define the Lax-Friedrichs flux as:

$$\widehat{\rho_h \mathbf{u}_h \cdot \mathbf{n}} = h (\rho_h^-, \rho_h^+, \mathbf{u}_h \cdot \mathbf{n}) = \frac{1}{2} [\mathbf{u}_h \cdot \mathbf{n} (\rho_h^+ + \rho_h^-) - a (\rho_h^+ - \rho_h^-)], \quad (3.2)$$

where a can be taken as the maximum of $|\mathbf{u}_h \cdot \mathbf{n}|$ either locally or globally, corresponding to a local Lax-Friedrichs flux or a global Lax-Friedrichs flux. If we simply take $a = |\mathbf{u}_h \cdot \mathbf{n}|$, then it is exactly the same as the upwind flux used in [23, 32].

By setting the test function $\phi_h \equiv 1$, we obtain the scheme satisfied by the cell average $\bar{\rho}_K = \bar{\rho}_{i,j} = \frac{1}{\Delta x \Delta y} \iint_K \rho(x, y) dx dy$, i.e., the scheme (3.1) with $\phi_h \equiv 1$ becomes,

$$\frac{d}{dt} \bar{\rho}_{i,j} + \frac{1}{\Delta x \Delta y} \sum_{e \in \partial K} \int_e \widehat{\rho_h \mathbf{u}_h \cdot \mathbf{n}} ds = 0.$$

With the forward Euler time discretization, the cell average scheme in the DG method is

$$\begin{aligned} \bar{\rho}_{i,j}^{n+1} = \bar{\rho}_{i,j}^n &- \frac{\Delta t}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left[h \left(\rho_{i+\frac{1}{2},j}^-(y), \rho_{i+\frac{1}{2},j}^+(y), u_{i+\frac{1}{2},j}(y) \right) - h \left(\rho_{i-\frac{1}{2},j}^-(y), \rho_{i-\frac{1}{2},j}^+(y), u_{i-\frac{1}{2},j}(y) \right) \right] dy \\ &- \frac{\Delta t}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[h \left(\rho_{i,j+\frac{1}{2}}^-(x), \rho_{i,j+\frac{1}{2}}^+(x), v_{i,j+\frac{1}{2}}(x) \right) - h \left(\rho_{i,j-\frac{1}{2}}^-(x), \rho_{i,j-\frac{1}{2}}^+(x), v_{i,j-\frac{1}{2}}(x) \right) \right] dx. \end{aligned} \quad (3.3)$$

The integrals in (3.3) can be computed exactly. Since all the integrands are single variable polynomials of degree at most $2k$, the integrals in (3.3) are equal to the L -point Gauss quadrature if $L \geq k+1$. In a DG scheme with polynomial basis of degree k , $L \geq k+1$ must be used, see [7]. Let $S_i^x = \{x_i^\beta : \beta = 1, \dots, L\}$ denote the Gauss quadrature points on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and $S_j^y = \{y_j^\beta : \beta = 1, \dots, L\}$ denote the Gauss quadrature points on $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. Let w_β denote the corresponding quadrature weights on the interval $[-\frac{1}{2}, \frac{1}{2}]$ so that $\sum_{\beta=1}^L w_\beta = 1$. We will use the subscript β to denote the values at the Gauss quadrature points, for instance, $\rho_{i+\frac{1}{2},\beta}^- = \rho^-(x_{i+\frac{1}{2}}, y_j^\beta)$. Substituting the integrals by the L -point Gauss quadrature in (3.3), we obtain the mathematically equivalent expression

$$\begin{aligned} \bar{\rho}_{i,j}^{n+1} = \bar{\rho}_{i,j}^n &- \lambda_1 \sum_{\beta=1}^L w_\beta \left[h \left(\rho_{i+\frac{1}{2},\beta}^-, \rho_{i+\frac{1}{2},\beta}^+, u_{i+\frac{1}{2},\beta} \right) - h \left(\rho_{i-\frac{1}{2},\beta}^-, \rho_{i-\frac{1}{2},\beta}^+, u_{i-\frac{1}{2},\beta} \right) \right] \\ &- \lambda_2 \sum_{\beta=1}^L w_\beta \left[h \left(\rho_{i,\beta,j+\frac{1}{2}}^-, \rho_{i,\beta,j+\frac{1}{2}}^+, v_{i,\beta,j+\frac{1}{2}} \right) - h \left(\rho_{i,\beta,j-\frac{1}{2}}^-, \rho_{i,\beta,j-\frac{1}{2}}^+, v_{i,\beta,j-\frac{1}{2}} \right) \right], \end{aligned} \quad (3.4)$$

where $\lambda_1 = \frac{\Delta t}{\Delta x}$ and $\lambda_2 = \frac{\Delta t}{\Delta y}$.

3.2. The bound-preserving property in the high order DG method

In this subsection we focus on the first order forward Euler time discretization. Bound-preserving high order time discretizations will be discussed in Section 3.6. We show that the cell average $\bar{\rho}_{i,j}^{n+1}$ in (3.4) is a monotonically increasing function with respect to some quadrature point values of the DG polynomial $\rho_{i,j}(x, y)$ in K , thus it is possible to obtain a bound-preserving scheme.

For completeness, we include a detailed discussion, which is slightly different from those in [32, 34] but essentially the same. Let N be the smallest integer such that $2N - 3 \geq k$ then the N -point Gauss-Lobatto quadrature rule is exact for polynomials of degree k . Let $\hat{S}_i^x = \{\hat{x}_i^\alpha : \alpha = 1, \dots, N\}$ and $\hat{S}_j^y = \{\hat{y}_j^\alpha : \alpha = 1, \dots, N\}$ denote the Gauss-Lobatto quadrature points on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ respectively. Let \hat{w}_α denote the quadrature weights on the interval $[-\frac{1}{2}, \frac{1}{2}]$ respectively. We will use subscript α to denote the evaluation at the Gauss-Lobatto quadrature points, for instance, $\rho_{\alpha,\beta} = \rho(\hat{x}_i^\alpha, \hat{y}_j^\beta)$.

After replacing the integrals by quadrature rules, the cell average $\bar{\rho}_{i,j}^n$ can be written as

$$\begin{aligned} \bar{\rho}_{i,j}^n &= \frac{1}{\Delta x \Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \rho_{i,j}(x, y) dx dy = \frac{1}{\Delta x \Delta y} \sum_{\beta=1}^L \sum_{\alpha=1}^N w_\beta \hat{w}_\alpha \rho_{\alpha,\beta}(\hat{x}_i^\alpha, \hat{y}_j^\beta) \Delta x \Delta y \\ &= \sum_{\beta=1}^L \sum_{\alpha=1}^N w_\beta \hat{w}_\alpha \rho_{\alpha,\beta} = \sum_{\beta=1}^L \sum_{\alpha=2}^{N-1} w_\beta \hat{w}_\alpha \rho_{\alpha,\beta} + \sum_{\beta=1}^L w_\beta \hat{w}_1 \rho_{\beta,j-\frac{1}{2}}^+ + \sum_{\beta=1}^L w_\beta \hat{w}_N \rho_{\beta,j+\frac{1}{2}}^-, \end{aligned}$$

and similarly we have

$$\bar{\rho}_{i,j}^n = \sum_{\alpha=1}^N \sum_{\beta=1}^L \hat{w}_\alpha w_\beta \rho_{\alpha,\beta} = \sum_{\alpha=2}^{N-1} \sum_{\beta=1}^L \hat{w}_\alpha w_\beta \rho_{\alpha,\beta} + \sum_{\beta=1}^L \hat{w}_1 w_\beta \rho_{i-\frac{1}{2},\beta}^+ + \sum_{\beta=1}^L \hat{w}_N w_\beta \rho_{i+\frac{1}{2},\beta}^-.$$

Let a_1 and a_2 be the global maximum of $|u|$ and $|v|$ respectively, and let $\mu_1 = \frac{a_1 \lambda_1}{a_1 \lambda_1 + a_2 \lambda_2}$ and $\mu_2 = \frac{a_2 \lambda_2}{a_1 \lambda_1 + a_2 \lambda_2}$, then $\mu_1 + \mu_2 = 1$. We can decompose the cell average as

$$\begin{aligned} \bar{\rho}_{i,j}^n &= \mu_1 \sum_{\alpha=1}^N \sum_{\beta=1}^L \hat{w}_\alpha w_\beta \rho_{\alpha,\beta} + \mu_2 \sum_{\beta=1}^L \sum_{\alpha=1}^N w_\beta \hat{w}_\alpha \rho_{\alpha,\beta} \\ &= \mu_1 \sum_{\alpha=2}^{N-1} \sum_{\beta=1}^L \hat{w}_\alpha w_\beta \rho_{\alpha,\beta} + \mu_1 \sum_{\beta=1}^L \hat{w}_1 w_\beta \rho_{i-\frac{1}{2},\beta}^+ + \mu_1 \sum_{\beta=1}^L \hat{w}_N w_\beta \rho_{i+\frac{1}{2},\beta}^- \\ &\quad + \mu_2 \sum_{\beta=1}^L \sum_{\alpha=2}^{N-1} w_\beta \hat{w}_\alpha \rho_{\alpha,\beta} + \mu_2 \sum_{\beta=1}^L w_\beta \hat{w}_1 \rho_{\beta,j-\frac{1}{2}}^+ + \mu_2 \sum_{\beta=1}^L w_\beta \hat{w}_N \rho_{\beta,j+\frac{1}{2}}^-. \end{aligned} \quad (3.5)$$

Notice that, for any $\lambda > 0$, under the constraint $\lambda a \leq 1$, the numerical flux (3.2) satisfies

$$\begin{aligned} \frac{\partial}{\partial \rho^+} [\rho^+ + \lambda h(\rho^-, \rho^+, |\mathbf{u} \cdot \mathbf{n}|)] &= 1 + \lambda \frac{|\mathbf{u} \cdot \mathbf{n}| - a}{2} \geq 0, & \frac{\partial}{\partial \rho^-} [\rho^+ + \lambda h(\rho^-, \rho^+, |\mathbf{u} \cdot \mathbf{n}|)] &= \lambda \frac{|\mathbf{u} \cdot \mathbf{n}| + a}{2} \geq 0, \\ \frac{\partial}{\partial \rho^-} [\rho^- - \lambda h(\rho^-, \rho^+, |\mathbf{u} \cdot \mathbf{n}|)] &= 1 - \lambda \frac{|\mathbf{u} \cdot \mathbf{n}| + a}{2} \geq 0, & \frac{\partial}{\partial \rho^+} [\rho^- - \lambda h(\rho^-, \rho^+, |\mathbf{u} \cdot \mathbf{n}|)] &= \lambda \frac{a - |\mathbf{u} \cdot \mathbf{n}|}{2} \geq 0. \end{aligned}$$

Plugging (3.2) and (3.5) into (3.4), we get

$$\begin{aligned}
\bar{\rho}_{i,j}^{n+1} &= \bar{\rho}_{i,j}^n - \lambda_1 \sum_{\beta=1}^L w_\beta \left[h\left(\rho_{i+\frac{1}{2},\beta}^-, \rho_{i+\frac{1}{2},\beta}^+, u_{i+\frac{1}{2},\beta}\right) - h\left(\rho_{i-\frac{1}{2},\beta}^-, \rho_{i-\frac{1}{2},\beta}^+, u_{i-\frac{1}{2},\beta}\right) \right] \\
&\quad - \lambda_2 \sum_{\beta=1}^L w_\beta \left[h\left(\rho_{\beta,j+\frac{1}{2}}^-, \rho_{\beta,j+\frac{1}{2}}^+, v_{\beta,j+\frac{1}{2}}\right) - h\left(\rho_{\beta,j-\frac{1}{2}}^-, \rho_{\beta,j-\frac{1}{2}}^+, v_{\beta,j-\frac{1}{2}}\right) \right] \\
&= \mu_1 \sum_{\alpha=2}^{N-1} \sum_{\beta=1}^L \hat{w}_\alpha w_\beta \rho_{\alpha,\beta} + \mu_2 \sum_{\beta=1}^L \sum_{\alpha=2}^{N-1} w_\beta \hat{w}_\alpha \rho_{\beta,\alpha} \\
&\quad + \mu_1 \sum_{\beta=1}^L w_\beta \hat{w}_1 \left[\rho_{i-\frac{1}{2},\beta}^+ + \frac{\lambda_1}{\mu_1 \hat{w}_1} h\left(\rho_{i-\frac{1}{2},\beta}^-, \rho_{i-\frac{1}{2},\beta}^+, u_{i-\frac{1}{2},\beta}\right) \right] \\
&\quad + \mu_1 \sum_{\beta=1}^L w_\beta \hat{w}_N \left[\rho_{i+\frac{1}{2},\beta}^- - \frac{\lambda_1}{\mu_1 \hat{w}_N} h\left(\rho_{i+\frac{1}{2},\beta}^-, \rho_{i+\frac{1}{2},\beta}^+, u_{i+\frac{1}{2},\beta}\right) \right] \\
&\quad + \mu_2 \sum_{\beta=1}^L w_\beta \hat{w}_1 \left[\rho_{\beta,j-\frac{1}{2}}^+ + \frac{\lambda_2}{\hat{w}_1 \mu_2} h\left(\rho_{\beta,j-\frac{1}{2}}^-, \rho_{\beta,j-\frac{1}{2}}^+, v_{\beta,j-\frac{1}{2}}\right) \right] \\
&\quad + \mu_2 \sum_{\beta=1}^L w_\beta \hat{w}_N \left[\rho_{\beta,j+\frac{1}{2}}^- - \frac{\lambda_2}{\hat{w}_N \mu_2} h\left(\rho_{\beta,j+\frac{1}{2}}^-, \rho_{\beta,j+\frac{1}{2}}^+, v_{\beta,j+\frac{1}{2}}\right) \right], \tag{3.6}
\end{aligned}$$

which is a monotonically increasing function with respect to all the point values $\rho_{\alpha,\beta}$, $\rho_{\beta,\alpha}$, $\rho_{\beta,j\pm\frac{1}{2}}^\pm$ and $\rho_{i\pm\frac{1}{2},\beta}^\pm$ under the CFL conditions $\frac{\lambda_1 a_1}{\hat{w}_1 \mu_1} \leq 1$ and $\frac{\lambda_2 a_2}{\hat{w}_1 \mu_2} \leq 1$, i.e.,

$$\max |u| \frac{\Delta t}{\Delta x} + \max |v| \frac{\Delta t}{\Delta y} \leq \hat{w}_1 = \hat{w}_N = \frac{1}{N(N-1)}. \tag{3.7}$$

The bound-preserving property in a high order DG method (3.1) can be stated as the following theorem.

Theorem 1. *Under the CFL condition (3.7), the cell average $\bar{\rho}_{i,j}^{n+1}$ computed in the (3.3) is a monotonically increasing function with respect to the point values $\rho(\hat{x}_i^\alpha, y_j^\beta)$ and $\rho(x_i^\beta, \hat{y}_j^\alpha)$ for all $\alpha = 1, \dots, N$ and $\beta = 1, \dots, L$ and all i, j . Furthermore, if $\rho(\hat{x}_i^\alpha, y_j^\beta) \in [m, M]$ and $\rho(x_i^\beta, \hat{y}_j^\alpha) \in [m, M]$ for all $\alpha = 1, \dots, N$ and $\beta = 1, \dots, L$ and all i, j , then $\bar{\rho}_{i,j}^{n+1} \in [m, M]$ provided that $\nabla \cdot \mathbf{u} = 0$ holds inside the (i, j) cell.*

Proof. We only discuss the lower bound and the upper bound can be similarly discussed. First, by setting $\rho \equiv m$ in (3.5), we have

$$m = \mu_1 \sum_{\alpha=1}^N \sum_{\beta=1}^L \hat{w}_\alpha w_\beta m + \mu_2 \sum_{\beta=1}^L \sum_{\alpha=2}^{N-1} w_\beta \hat{w}_\alpha m.$$

Since $\bar{\rho}_{i,j}^{n+1}$ computed in the (3.3) is a monotonically increasing function with all the point values, $\bar{\rho}_{i,j}^{n+1}$ should be greater than or equal to the right hand side of (3.6) with all point values replaced by m . Thus

we have

$$\begin{aligned}
\bar{\rho}_{i,j}^{n+1} &\geq \mu_1 \sum_{\alpha=2}^{N-1} \sum_{\beta=1}^L \hat{w}_\alpha w_\beta m + \mu_2 \sum_{\beta=1}^L \sum_{\alpha=2}^{N-1} w_\beta \hat{w}_\alpha m \\
&\quad + \mu_1 \sum_{\beta=1}^L w_\beta \hat{w}_1 \left[m + \frac{\lambda_1}{\mu_1 \hat{w}_1} m u_{i-\frac{1}{2},\beta} \right] + \mu_1 \sum_{\beta=1}^L w_\beta \hat{w}_N \left[m - \frac{\lambda_1}{\mu_1 \hat{w}_N} m u_{i+\frac{1}{2},\beta} \right] \\
&\quad + \mu_2 \sum_{\beta=1}^L w_\beta \hat{w}_1 \left[m + \frac{\lambda_2}{\hat{w}_1 \mu_2} m v_{\beta,j-\frac{1}{2}} \right] + \mu_2 \sum_{\beta=1}^L w_\beta \hat{w}_N \left[m - \frac{\lambda_2}{\hat{w}_N \mu_2} m v_{\beta,j+\frac{1}{2}} \right] \\
&= m + m \left[\lambda_1 \sum_{\beta=1}^L w_\beta u_{i-\frac{1}{2},\beta} - \lambda_1 \sum_{\beta=1}^L w_\beta u_{i+\frac{1}{2},\beta} + \lambda_2 \sum_{\beta=1}^L w_\beta v_{\beta,j-\frac{1}{2}} - \lambda_2 \sum_{\beta=1}^L w_\beta v_{\beta,j+\frac{1}{2}} \right] \\
&= m + m \frac{\Delta t}{\Delta x \Delta y} \left[\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i-\frac{1}{2}}, y) dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+\frac{1}{2}}, y) dy + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(x, y_{j-\frac{1}{2}}) dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(x, y_{j+\frac{1}{2}}) dx \right] \\
&= m - m \frac{\Delta t}{\Delta x \Delta y} \iint_K \nabla \cdot \langle u, v \rangle dx dy \\
&= m,
\end{aligned}$$

where we have used Gauss quadrature for the line integrals of polynomial integrands and the Divergence Theorem. \square

Remark 1. The divergence free constraint $\nabla \cdot \langle u, v \rangle = 0$ is trivially satisfied on the cell K since we assume the velocity is computed by $\langle u_h, v_h \rangle = \left\langle -\frac{\partial \psi_h}{\partial y}, \frac{\partial \psi_h}{\partial x} \right\rangle$ for some stream function $\psi_h \in W_{0,h}^k$.

Remark 2. If one is interested in preserving a special lower bound $m = 0$, i.e., positivity-preserving, then the result above also holds without the incompressibility constraint $\nabla \cdot \langle u, v \rangle = 0$.

3.3. A simple high order accurate bound-preserving limiter

Let \otimes denote the tensor product, for instance, $S_i^x \otimes S_j^y = \{(x, y) : x \in S_i^x, y \in S_j^y\}$. Now we define the set $S_{i,j}$ as

$$S_{i,j} = (\hat{S}_i^x \otimes S_j^y) \cup (S_i^x \otimes \hat{S}_j^y). \quad (3.8)$$

At time level n , given the DG polynomial $\rho_{i,j}^n(x, y)$ on the (i, j) cell with a bounded cell average $\bar{\rho}_{i,j}^n \in [m, M]$, by Theorem 1 we need $\rho_{i,j}^n(x, y) \in [m, M]$ for any $(x, y) \in S_{i,j}$ to ensure $\bar{\rho}_{i,j}^{n+1} \in [m, M]$. The simple bound-preserving limiter in [32] is to modify $\rho_{i,j}^n(x, y)$ into a new polynomial $\tilde{\rho}_{i,j}^n(x, y)$:

$$\tilde{\rho}_{i,j}^n(x, y) = \theta (\rho_{i,j}^n(x, y) - \bar{\rho}_{i,j}^n) + \bar{\rho}_{i,j}^n, \quad \theta = \min \left\{ \left| \frac{m - \bar{\rho}_{i,j}^n}{m_{i,j} - \bar{\rho}_{i,j}^n} \right|, \left| \frac{M - \bar{\rho}_{i,j}^n}{M_{i,j} - \bar{\rho}_{i,j}^n} \right|, 1 \right\}, \quad (3.9)$$

$$m_{i,j} = \min_{(x,y) \in S_{i,j}} \rho_{i,j}^n(x, y), \quad M_{i,j} = \max_{(x,y) \in S_{i,j}} \rho_{i,j}^n(x, y). \quad (3.10)$$

Then $\tilde{\rho}_{i,j}^n(x, y) \in [m, M]$ for all $(x, y) \in S_{i,j}$ and the cell average of $\tilde{\rho}_{i,j}^n(x, y)$ is still $\bar{\rho}_{i,j}^n$. Moreover, this limiter does not destroy high order accuracy if $\rho_{i,j}^n(x, y)$ is high order approximation to any smooth function bounded by m and M , see the appendix in [31]. The limiter (3.9) should be used for all (i, j) cells.

3.4. A finite element method for the velocity and pressure evolution

For solving the momentum equation (2.4b) on a rectangular domain, we can use a continuous finite element method following the method with pressure p treated explicitly in time in the momentum equation.

We focus on the first order forward Euler time discretization. High order time discretizations will be discussed in Section 3.6. Let ρ^n , \mathbf{u}^n and p^n denote numerical solutions of the flow variables at time t_n . We

rewrite the diffusion term in momentum equation (2.4b) into a divergence form then compute $\mathbf{u}^{n+1} \in \mathbf{W}_{0,h}^k$ by discretizing the equation with a finite element method as follows

$$\left\langle \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t}, \mathbf{v} \right\rangle_{\Omega_h} + \left\langle (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \nabla \frac{1}{\rho^n} \cdot (\mu(\rho^n) \nabla \mathbf{u}^n) + \frac{\nabla p^n - \mathbf{f}^n}{\rho^n}, \mathbf{v} \right\rangle_{\Omega_h} + \left\langle \frac{1}{\rho^n} \mu(\rho^n) \nabla \mathbf{u}^n, \nabla \mathbf{v} \right\rangle_{\Omega_h} = 0, \forall \mathbf{v} \in \mathbf{W}_{0,h}^k, \quad (3.11)$$

where $\langle u, v \rangle_{\Omega}$ denotes the L^2 inner product of u and v on Ω_h .

Notice that ρ^n or ρ^{n+1} computed by the DG method is not single valued across the cell boundaries. To map point values of ρ from a DG mesh to a continuous finite element mesh, we can simply define ρ as average of values from two neighbor cells at edge centers, and as average of values from four neighbour cells at interior knots.

After ρ^{n+1} , \mathbf{u}^{n+1} are computed, we can update the pressure $p^{n+1} \in W_h^k$ by solving the variable-coefficient pressure Poisson equation (2.4c) with the Neumann boundary condition by a finite element method, which is given by

$$\left\langle \frac{1}{\rho^{n+1}} \nabla p^{n+1}, \nabla v \right\rangle_{\Omega_h} = \langle f_p^{n+1}, v \rangle_{\Omega_h} + \left\langle \frac{1}{\rho^{n+1}} g^{n+1}, v \right\rangle_{\Gamma}, \quad \forall v \in W_h^k, \quad (3.12)$$

where $g^{n+1} = [\nabla \cdot (\mu(\rho^{n+1}) \nabla (\mathbf{u}^{n+1} \cdot \mathbf{n})) + \mathbf{f}^{n+1} \cdot \mathbf{n}]|_{\Gamma}$ is the boundary condition and

$$f_p^{n+1} = \nabla \cdot \left((\mathbf{u}^{n+1} \cdot \nabla) \mathbf{u}^{n+1} - \frac{1}{\rho^{n+1}} \nabla \cdot (\mu(\rho^{n+1}) \nabla \mathbf{u}^{n+1}) - \frac{\mathbf{f}^{n+1}}{\rho^{n+1}} \right).$$

Here for solving the pressure Poisson equation, we regard f_p^{n+1} as a given source term and the derivatives involved in f_p^{n+1} can be computed by finite difference. Since $\mathbf{u}^{n+1}, \rho^{n+1}$ are continuous piecewise Q^k polynomials, they can be equivalently represented as point values at a global grid consisting of all Gauss-Lobatto quadrature points for all cells. Then derivatives of $\rho^{n+1}, \mathbf{u}^{n+1}, \mu$ at this global grid can be approximated by any conventional high order accurate finite difference method.

In practice it is computationally demanding to solve (2.4c) for three reasons. First, since variable coefficient $\frac{1}{\rho^{n+1}}$ is different in each time step, it is preferred to have an efficient assembly of stiffness matrix such as a few compact matrix matrix multiplications in a traditional finite difference scheme. Second, the purely Neumann boundary condition results in an inconsistent linear system $A\mathbf{x} = \mathbf{b}$ for any scheme discretizing (2.4c). To find the least square solution, one needs to project \mathbf{b} onto the column space of A , for which the left null vector of A is needed. Third, condition number of A will deteriorate when discontinuity appears in the coefficient ρ , e.g., for computing large density ratio flows. Let $\mathbf{1}$ denote the constant one vector, then $A\mathbf{1} = \mathbf{0}$ since A approximates a differential operator $\nabla \cdot (\frac{1}{\rho} \nabla)$. Thus the left null vector of A is also $\mathbf{1}$ if A is symmetric. The variational formulation of (3.12) naturally gives a symmetric stiffness matrix, which is one of main practical advantages in finite element method over conventional finite difference schemes for purely Neumann boundary condition since the stiffness matrix in traditional finite difference methods is difficult to symmetrize and the left null vector is expensive to compute. A simple assembly of stiffness matrix for finite element method on rectangular meshes, as efficient as in a conventional finite difference scheme, along with a simple and efficient Laplacian based preconditioner for variable coefficient problem, were discussed in Section 7 of [20]. For the linear system $A\mathbf{x} = \mathbf{b}$ from the scheme (3.12), since $\mathbf{1}$ is the left null vector A , the projection of \mathbf{b} onto column space of A is $\bar{\mathbf{b}} = \mathbf{b} - (\mathbf{b} \cdot \mathbf{1})/(\mathbf{1} \cdot \mathbf{1})\mathbf{1}$. Thus to obtain the least square solution $A\mathbf{x} = \mathbf{b}$, we can use preconditioned conjugate gradient method for the consistent linear system $A\mathbf{x} = \bar{\mathbf{b}}$.

3.5. The stream function

Given \mathbf{u}^n , we need to enforce the incompressibility condition $\nabla \cdot \mathbf{u}^n = 0$ inside each element for the sake of bound-preserving. To this end, we consider the stream function ψ satisfying $\mathbf{u} = \nabla^\perp \psi = (-\psi_y, \psi_x)$ for an incompressible velocity field \mathbf{u} . We consider enforcing the boundary conditions $\mathbf{u} = \mathbf{0}$ on the boundary Γ for which we need both $\psi|_{\Gamma} = 0$ and $\frac{\partial \psi}{\partial \mathbf{n}}|_{\Gamma} = 0$. Enforcing both Dirichlet and Neumann boundary

conditions for Poisson equation will result in an overdetermined linear system. Thus we first find a stream function satisfying a Poisson equation with the Dirichlet boundary condition

$$\Delta \tilde{\psi}^n = \omega^n, \quad \tilde{\psi}^n \Big|_{\Gamma} = 0, \quad (3.13)$$

where $\omega^n = \nabla \times \mathbf{u}^n = v_x^n - u_y^n$ is the vorticity. Then we modify $\tilde{\psi}^n$ to enforce the Neumann boundary condition, for which we need the solution to a constrained least square problem,

$$\min_{\psi^n \in W_{0,h}^k} \sum_{i,j} \sum_{(x,y) \in \hat{S}_i^x \otimes \hat{S}_j^y} |\psi^n(x,y) - \tilde{\psi}^n(x,y)|^2, \quad \frac{\partial \psi^n}{\partial \mathbf{n}} \Big|_{\Gamma} = 0. \quad (3.14)$$

Due to the fact that $\tilde{\psi}, \psi \in W_{0,h}^k$, the least square problem can be decoupled into local least square problems for each boundary cell. For each boundary rectangular cell K , which is also denoted as (i, j) cell, a local least square problem is to find $\psi^n \in Q^k(K)$ by minimizing the distance

$$\sum_{(x,y) \in \hat{S}_i^x \otimes \hat{S}_j^y} |\psi^n(x,y) - \tilde{\psi}^n(x,y)|^2, \quad s.t. \quad \psi^n \Big|_{\partial K \cap \Gamma} = \frac{\partial \psi^n}{\partial \mathbf{n}} \Big|_{\partial K \cap \Gamma} = 0. \quad (3.15)$$

The Q^k basis in each cell can be represented as Lagrangian basis at $(k+1) \times (k+1)$ Gauss-Lobatto points,

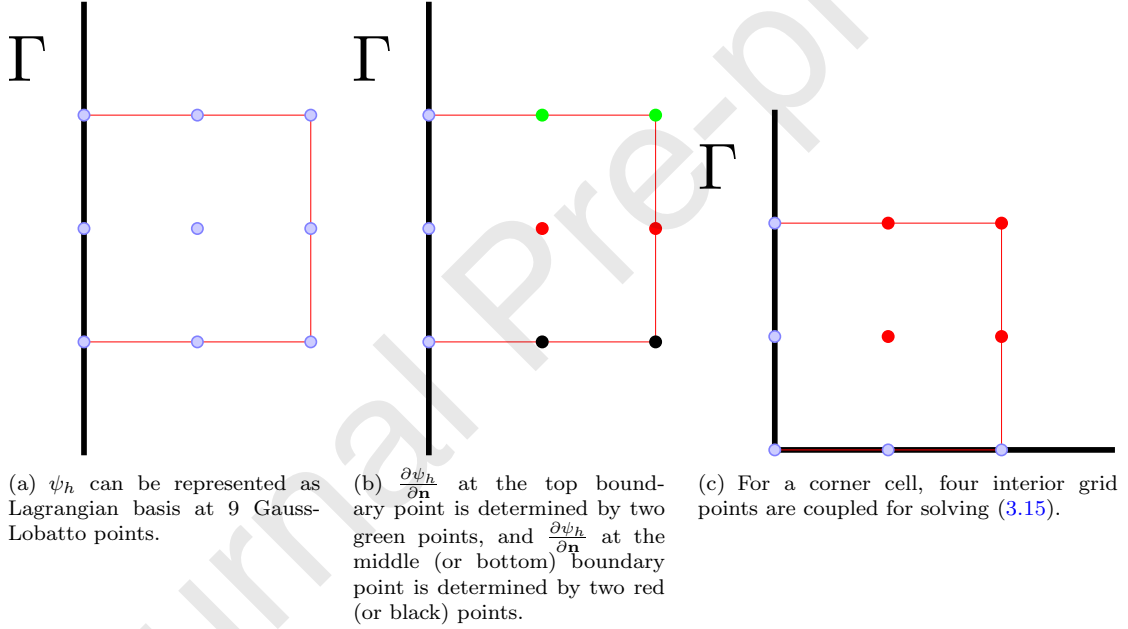


Fig. 3.1: An illustration of the local least square problem for a Q^2 element $\psi_h \in W_{0,h}^k$ on a boundary cell. All boundary grid points have fixed value zero due to the boundary condition.

e.g., Figure 3.1 (a). There are two types of boundary cells: corner cells and non-corner boundary cells. For simplicity, consider Q^2 basis as an example, since boundary point values are already fixed as zero, only four interior points are degree of freedoms in solving (3.15), see Figure 3.1 (c). For $\psi_h \in W_{0,h}^k$ on a non-corner boundary cell K adjacent to the left boundary, $\frac{\partial \psi_h}{\partial \mathbf{n}}$ at a boundary grid point is determined by point values at grid points on the same horizontal line, thus the local least square problem (3.15) can be further decoupled and solved in a line by line fashion, see Figure 3.1 (b). Therefore, the global least square problem can be easily solved without losing global continuity of ψ :

1. First, solve the local least square problems (3.15) for four corner cells.
2. Second, solve the local least square problem (3.15) for non-corner boundary cells in a line by line fashion, excluding the lines included in four corner cells.

3.6. High order time discretizations

3.6.1. Explicit SSP Runge-Kutta methods

For preserving bounds of density, any strong stability preserving (SSP) high order time discretizations [9] can be used instead of the forward Euler time discretization. SSP high order time discretizations are convex combinations of forward Euler steps. Thus if forward Euler can preserve bounds, then so is a SSP time discretization due to the convex combination. For example, we can use the following explicit third order SSP Runge-Kutta method (with the CFL coefficient $c = 1$) for solving

$$u_t = F(u), \quad (3.16)$$

where $F(u)$ is a spatial discretization operator,

$$\begin{aligned} u^{(1)} &= u^n + \Delta t F(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4} \left(u^{(1)} + \Delta t F(u^{(1)}) \right), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3} \left(u^{(2)} + \Delta t F(u^{(2)}) \right). \end{aligned} \quad (3.17)$$

Here, the CFL coefficient c for a SSP time discretization refers to the fact that, if we assume the forward Euler time discretization for solving the equation $u_t = F(u)$ is stable in a norm or preserves bounds/positivity under a time step restriction $\Delta t \leq \Delta t_0$, then the high order SSP time discretization is also stable in the same norm or preserves bounds/positivity under the time step restriction $\Delta t \leq c\Delta t_0$. The bound-preserving limiter (3.9) should be applied to the DG polynomials for ρ in each time stage of the third order SSP Runge-Kutta method.

Notice that the CFL condition (3.7) is only sufficient for preserving bounds of the density in the DG scheme solving (2.4a). Even though (3.7) happens to be sufficient for the linear stability of a DG method solving a convection equation [32], it is not enough for the linear stability of the full scheme. For the stability of the numerical method solving (2.4b), we also need to enforce suitable CFL constraints for an explicit time discretization solving a convection diffusion equation [18]. For example, for forward Euler, in addition to CFL for preserving bounds for density (3.7), we also need $\Delta t = \mathcal{O}(\frac{1}{\mu}\Delta x^2)$ to ensure linear stability for momentum evolution. Thus such a fully explicit time discretization is more suitable for high Reynolds number flows, i.e., when μ is small.

3.6.2. Implicit-explicit SSP Runge-Kutta methods

An alternative to avoid the constraint $\Delta t = \mathcal{O}(\frac{1}{\mu}\Delta x^2)$ is to use an implicit-explicit (IMEX) SSP Runge-Kutta schemes in [24]. For IMEX type time discretizations, stability can be proven under time step constraints in the form of $\Delta t = \mathcal{O}(\sqrt{\mu})$ which is independent of Δx , see [27] and also [28, 29, 30]. We emphasize that an IMEX SSP Runge-Kutta method is not a convex combination of first order IMEX schemes, because such a convex combination is only first order accurate, see [16, 15] and references therein. Consider solving a problem in the form $U_t + F(U) = G(U)$. An IMEX Runge-Kutta scheme consists of applying an implicit discretization to G and an explicit one to F :

$$U^{(i)} = U^n - \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} F(U^{(j)}) + \Delta t \sum_{j=1}^i a_{ij} G(U^{(j)}), \quad (3.18a)$$

$$U^{n+1} = U^n - \Delta t \sum_{i=1}^{\nu} \tilde{w}_i F(U^{(i)}) + \Delta t \sum_{i=1}^{\nu} w_i G(U^{(i)}), \quad (3.18b)$$

where the matrices $\tilde{A} = (\tilde{a}_{ij})$, with $\tilde{a}_{ij} = 0$ for $j \geq i$ and $A = (a_{ij})$ are $\nu \times \nu$ matrices such that the resulting scheme is explicit in F and implicit in G . An IMEX Runge-Kutta scheme is characterized by these two matrices and the coefficient vectors $\tilde{w} = (\tilde{w}_1, \dots, \tilde{w}_{\nu})^T$ and $w = (w_1, \dots, w_{\nu})^T$, thus can be represented by a double Butcher tableau:

$$\begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline & \tilde{w}^T \end{array} \quad \begin{array}{c|c} c & A \\ \hline & w^T \end{array}$$

where the vectors $\tilde{c} = (\tilde{c}_1, \dots, \tilde{c}_s)^T$, $c = (c_1, \dots, c_s)^T$ are defined as

$$\tilde{c}_i = \sum_{j=1}^{i-1} \tilde{a}_{ij}, \quad c_i = \sum_{j=1}^i a_{ij}.$$

If the explicit table coincides with a SSP Runge-Kutta method, then we say the full scheme is an IMEX SSP Runge-Kutta method. For instance, the double Butcher tableau for a third order accurate IMEX SSP Runge-Kutta scheme SSP3(4,3,3) given in [24] is:

0	0	0	0	0	α	α	0	0	0
0	0	0	0	0	0	$-\alpha$	α	0	0
1	0	1	0	0	1	0	$1-\alpha$	α	0
1/2	0	1/4	1/4	0	1/2	β	η	$1/2 - \beta - \eta - \alpha$	α
	0	1/6	1/6	2/3		0	1/6	1/6	2/3

with

$$\alpha = 0.24169426078821, \beta = 0.06042356519705, \eta = 0.12915286960590.$$

If we use the SSP3(4,3,3) scheme above for solving (2.4a) and (2.4b) with $U = (\rho, \mathbf{u})^T$, G representing only the diffusion term $\nabla \cdot (\mu \nabla \mathbf{u})$ and F rerepresenting all other spatial and forcing terms in (3.18), then the time discretization for the variable ρ is equivalent to the third order SSP TVD Runge-Kutta method (3.17). By doing so, bound-preserving is achieved in a third order time solver, and small time steps due to linear stability in a fully explicit time solver for low Reynolds number are avoided, since there is no viscosity term for the density evolution.

Remark 3. When implementing the SSP3(4,3,3) scheme above for solving (2.4a) and (2.4b), the time discretization for density evolution (2.4a) must be implemented in the form of (3.17) so that the bound-preserving limiter can be applied to each forward Euler step.

3.7. Summary of the scheme with implementation details

For the readers' convenience, we summarize below the implementation of the bound-preserving scheme with forward Euler time discretization. For high order Runge-Kutta time discretizations, the implementation for each time stage is similar.

Given $\mathbf{u}^n \in \mathbf{W}_{0,h}^k$, $p^n \in W_h^k$ and $\rho^n \in V_h^k$ with $\bar{\rho}^n \in [m, M]$ at time level n , the variables can be updated as follows.

1. First solve (3.13) then solve local least square problems along boundary cells to find ψ^n . Obtain locally divergence free velocity field by setting $\mathbf{u}^n = \nabla^\perp \psi^n = (-\psi_y^n, \psi_x^n)$. The velocity on the DG method mesh to be used for solving (2.4a) is recovered from $\mathbf{u}^n = \nabla^\perp \psi^n = (-\psi_y^n, \psi_x^n)$. Notice that $u^n, v^n \in V_h^k$ with the following three desired properties trivially satisfied:
 - The boundary condition $\mathbf{u}^n|_\Gamma = \mathbf{0}$.
 - Discrete incompressibility $\nabla \cdot \mathbf{u}^n = 0$ holds inside each cell in the mesh for DG method.
 - Across any inner cell boundary, the normal velocity $\mathbf{u}^n \cdot \mathbf{n}$ is continuous because it is exactly the tangential derivative of a continuous function $\psi^n(x, y)$ along cell boundaries.
2. Apply the bound-preserving limiter (3.9) to density variable so that ρ^n is the desired range $[m, M]$ at suitable points. Compute $\rho^{n+1} \in V_h^k$ by (3.1) with forward Euler time discretization with CFL constraint (3.7). By Theorem 1, bound-preserving of cell averages $\bar{\rho}^{n+1} \in [m, M]$ is ensured.
3. Update \mathbf{u}^{n+1} by (3.11) and update p^{n+1} by (3.12).

Remark 4. For the sake of bound-preserving in Theorem 1, to enforce local divergence free constraint, an extra Poisson equation for stream function (3.13) needs be solved in each time stage. The extra computational cost is marginal though, because inverting Laplacian is much cheaper compared to solving (3.12), especially for problems with large density ratios. For continuous finite element method with Q^k elements, Laplacian can be efficiently inverted by eigenvector method as explained in Section 7.4 in [20].

4. Numerical examples

In this section, we provide some numerical examples to investigate the performance of the proposed scheme. In all examples, Q^2 polynomials were used on uniform rectangular meshes. For the DG method solving density, besides the bound-preserving limiter, a total variation bounded (TVB) limiter [6] is also used for reducing oscillations.

4.1. Accuracy test

We test the accuracy of the numerical scheme by considering a smooth exact solutions in the following form [17]:

$$\begin{cases} \rho(x, y, t) = ((3/4) + (1/4) \sin t)(2 + \cos x \cos y), \\ u(x, y, t) = ((3/4) + (1/4) \sin t)(-\sin^2 x \sin y \cos y), \\ v(x, y, t) = ((3/4) + (1/4) \sin t)(\sin x \cos x \sin^2 y), \\ p(x, y, t) = ((3/4) + (1/4) \sin t)(\cos x \sin y). \end{cases} \quad (4.1)$$

Here we set $\mu = \frac{\pi}{5000}$, and to ensure that (4.1) is an exact solution of (2.4), appropriate forcing functions are applied to the system. The errors on a unit square $\Omega = [0, \pi]m \times [0, \pi]m$ at $t = 0.1$ are shown in Table 4.1, in which we observe around third order accuracy for the explicit SSP Runge-Kutta method. In this test, the time step for the explicit SSP Runge-Kutta method is set as $\Delta t = 0.1\Delta x^2$. The time step for the IMEX SSP Runge-Kutta method is set as $\Delta t = 0.1\Delta x$, with which the explicit SSP Runge-Kutta method is not stable. For IMEX SSP Runge-Kutta method, we observe some obvious order reduction, which could be due to the well known order reduction phenomenon for IMEX Runge-Kutta methods in certain regime [4, 16, 15]. To the best of our knowledge, only for IMEX BDF methods, uniform accuracy can be proven [1, 14]. For this particular accuracy test, the order reduction for the IMEX SSP Runge-Kutta method is more prominent for larger viscosity coefficient μ , and clean third order accuracy can be observed if using a third order IMEX BDF method.

4.2. Rayleigh-Taylor instability

We consider the development of Rayleigh-Taylor instabilities in the viscous regime as documented by Tryggvason in [26]. This problem consists of two layers of fluid initially at rest in the gravity field. The initial perturbed interface between the two fluids is given by $\eta(x) = -0.1d \cos(2\pi x/d)$. The Atwood number is defined as

$$At = \frac{\rho_M - \rho_m}{\rho_M + \rho_m}, \quad (4.2)$$

and the Reynolds number is given by

$$Re = \frac{\rho_m d^{3/2} g^{1/2}}{\mu}, \quad (4.3)$$

where g is the gravitational acceleration. For $t > 0$ the system evolves under the action of a vertical downward gravity field of intensity g , so the source term in the momentum equation is downward and equal to ρg . In the numerical tests, we set $Re = 1000$, $g = -9.80665m/s^2$ and $\rho_m = 1.0kg/m^3$, and we test three different cases with $At = 0.5, 0.75$ and 0.9 . The computational domain is $[-0.5, 0.5]m \times [-2, 1]m$. The time evolution of the density with $At = 0.5$ on three different meshes are shown in Figure 4.1, Figure 4.2 and Figure 4.3. The time evolution of the density with $At = 0.75$ on three different meshes are shown in Figure 4.4, Figure 4.5 and Figure 4.6. The time evolution of the density with $At = 0.9$ on three different meshes are shown in Figure 4.7, Figure 4.8 and Figure 4.9. From the results of these cases, we can observe the similar structure and the global characteristics of the flow in the early stage. At the same time, we found that the heavy fluid falls faster when the low Atwood number becomes larger. In order to verify the performance of the bound-preserving limiter, we show the numerical results obtained from the proposed scheme with and without the bound-preserving limiter Figure 4.10. We can see that the proposed scheme with the bound-preserving limiter maintains the numerical results of density in the domain $[1, 3]$ for $At = 0.5$, in the domain $[1, 7]$ for $At = 0.75$, and in the domain $[1, 19]$ for $At = 0.9$, respectively.

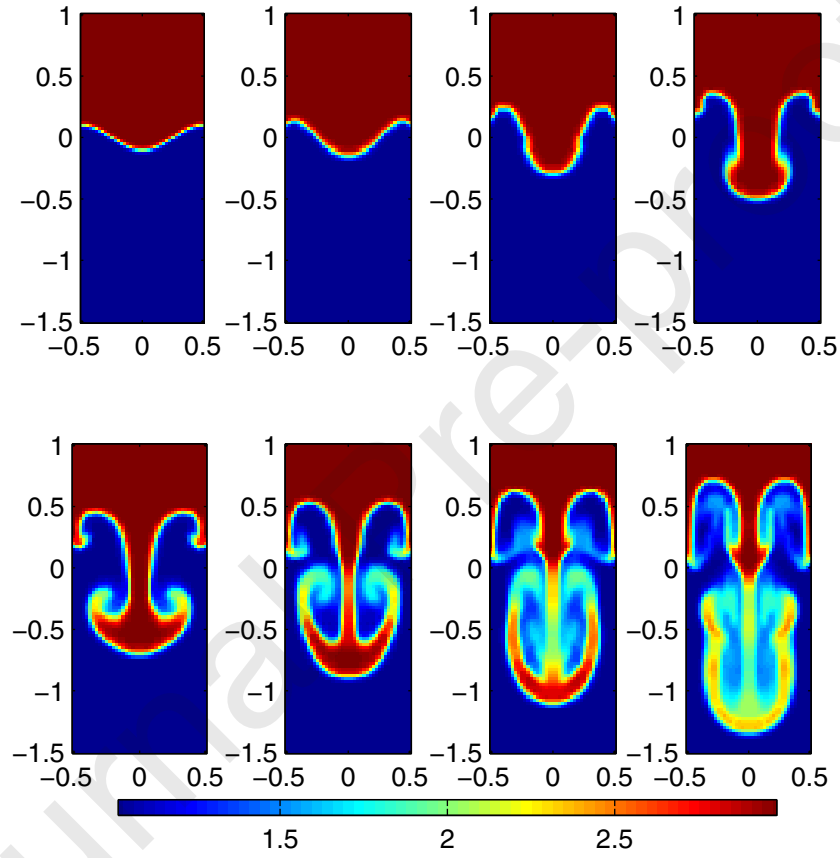


Fig. 4.1: Numerical solution of Rayleigh-Taylor instability with $At = 0.5$ at $t = 0s, 0.2s, 0.4s, 0.6s, 0.8s, 1.0s, 1.2s$ and $1.4s$ (from left to right, from top to bottom). Mesh: 20×50 .

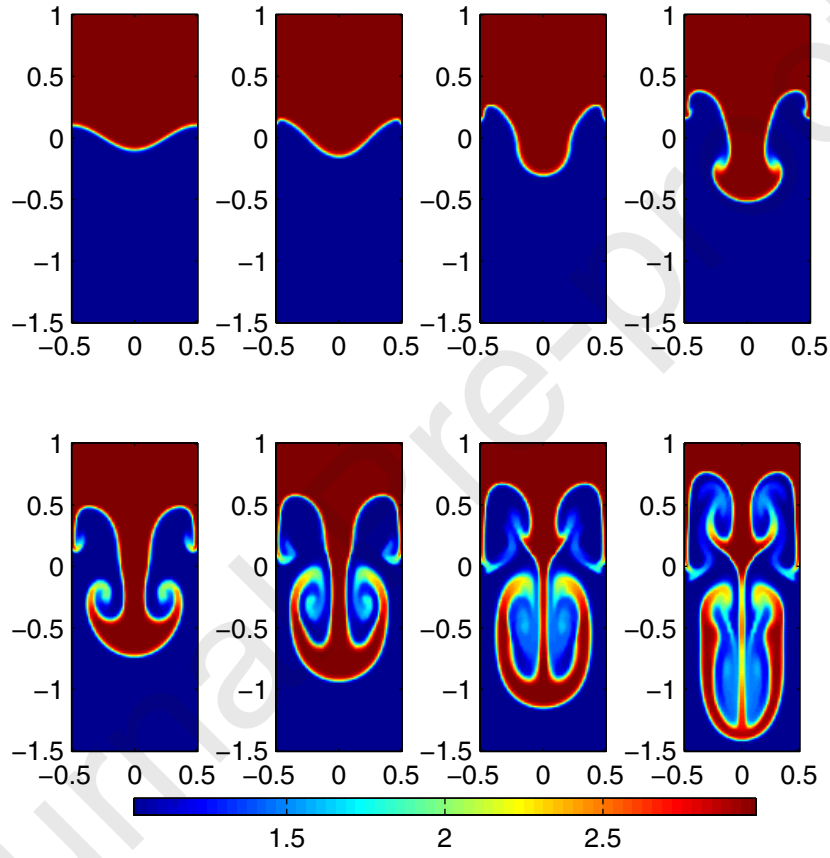


Fig. 4.2: Numerical solution of Rayleigh-Taylor instability with $At = 0.5$ at $t = 0s, 0.2s, 0.4s, 0.6s, 0.8s, 1.0s, 1.2s$ and $1.4s$ (from left to right, from top to bottom). Mesh: 40×100 .

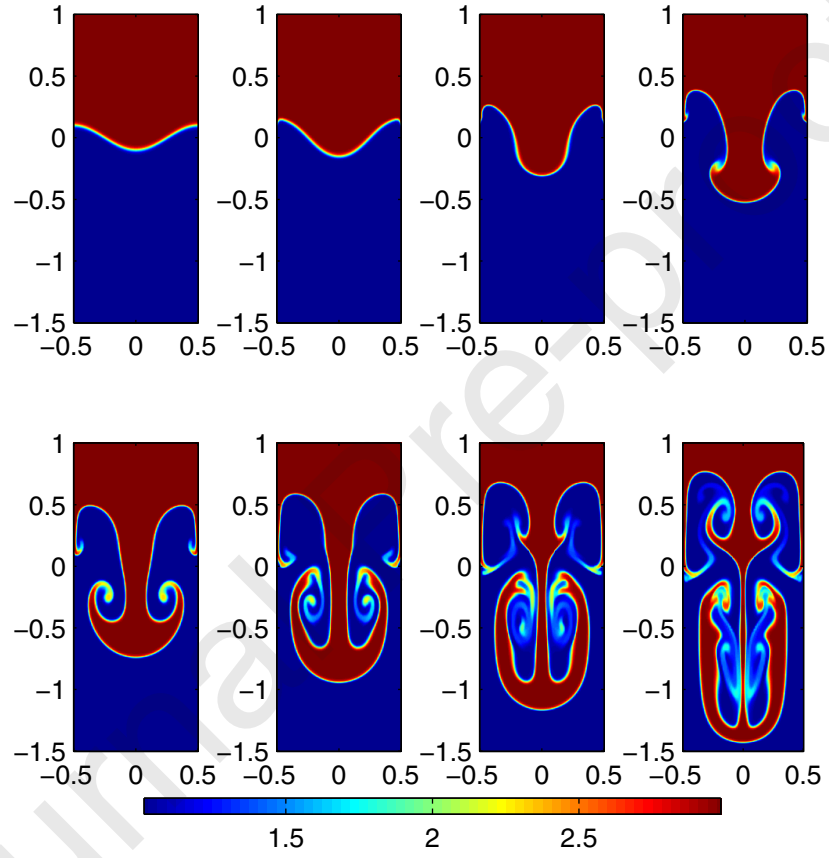


Fig. 4.3: Numerical solution of Rayleigh-Taylor instability with $At = 0.5$ at $t = 0s, 0.2s, 0.4s, 0.6s, 0.8s, 1.0s, 1.2s$ and $1.4s$ (from left to right, from top to bottom). Mesh: 80×200 .

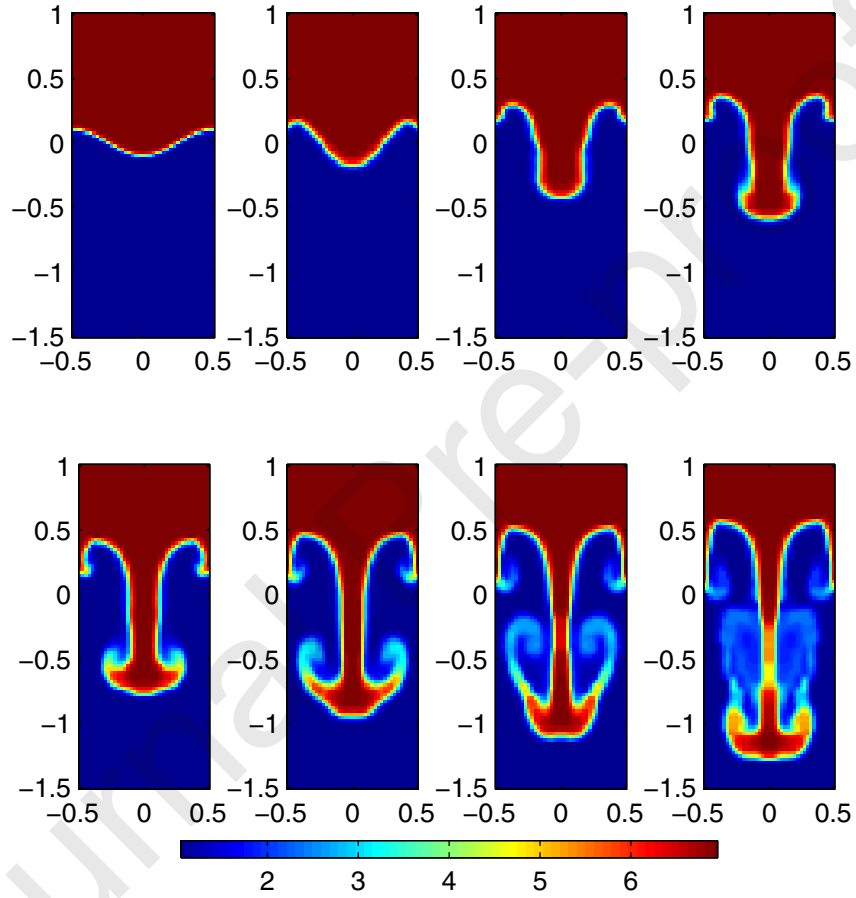


Fig. 4.4: Numerical solution of Rayleigh-Taylor instability with $At = 0.75$ at $t = 0s, 0.2s, 0.4s, 0.5s, 0.6s, 0.7s, 0.8s$ and $0.9s$ (from left to right, from top to bottom). Mesh: 20×50 .

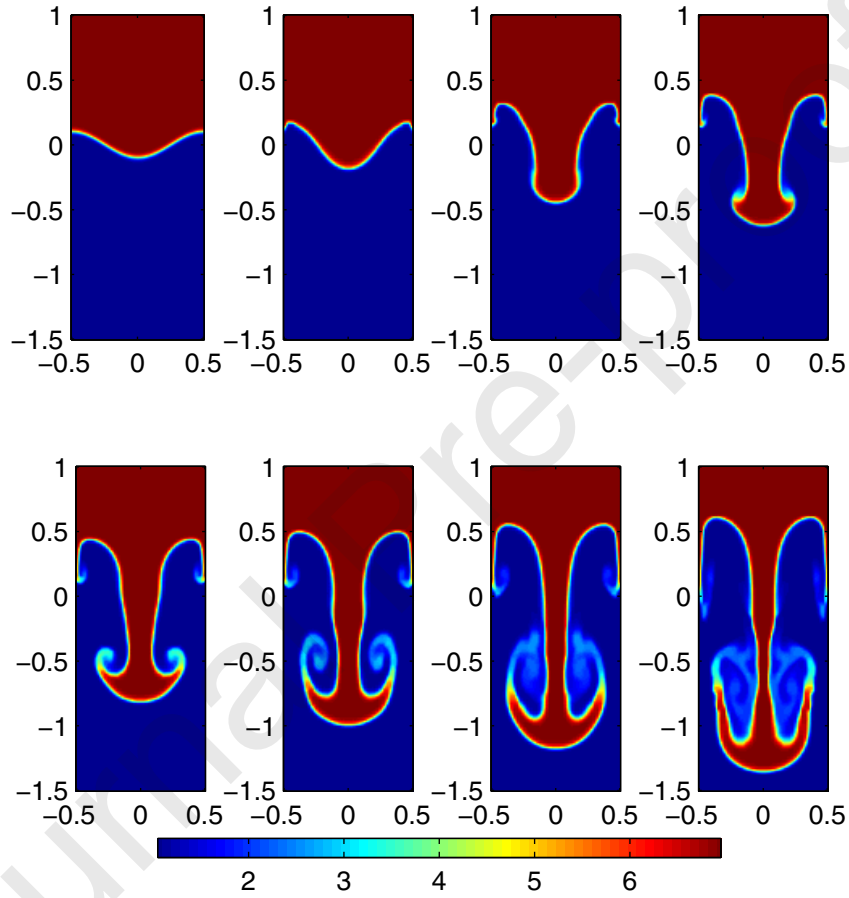


Fig. 4.5: Numerical solution of Rayleigh-Taylor instability with $At = 0.75$ at $t = 0s, 0.2s, 0.4s, 0.5s, 0.6s, 0.7s, 0.8s, 0.9s$ and $0.9s$ (from left to right, from top to bottom). Mesh: 40×100 .

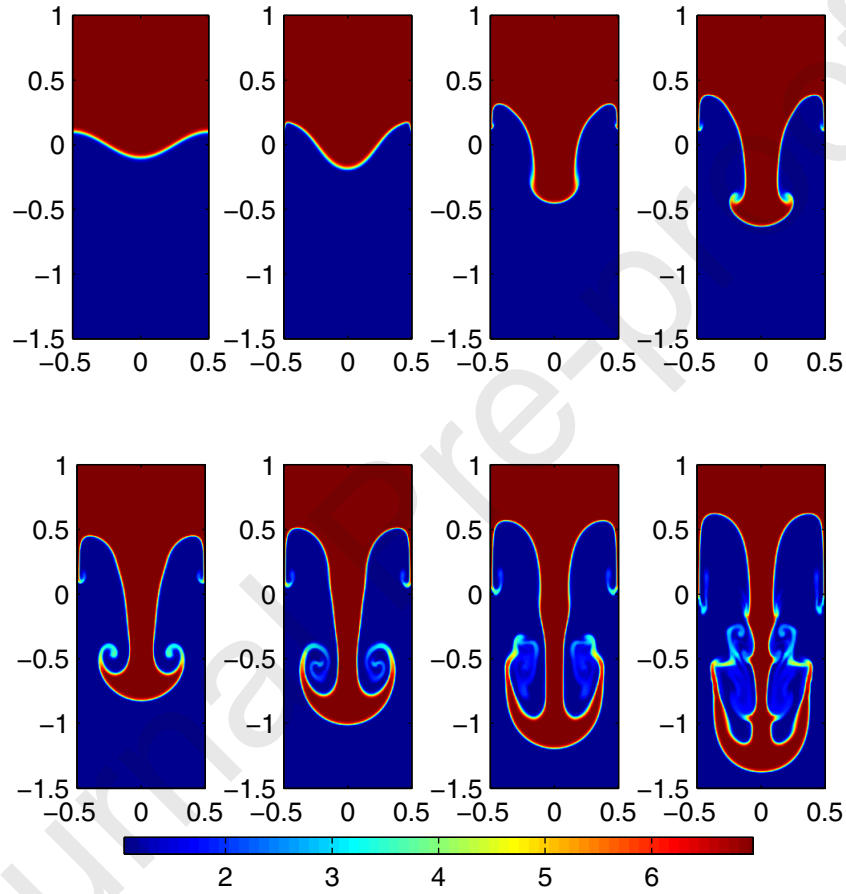


Fig. 4.6: Numerical solution of Rayleigh-Taylor instability with $At = 0.75$ at $t = 0s, 0.2s, 0.4s, 0.5s, 0.6s, 0.7s, 0.8s, 0.9s$ and $0.9s$ (from left to right, from top to bottom). Mesh: 80×200 .

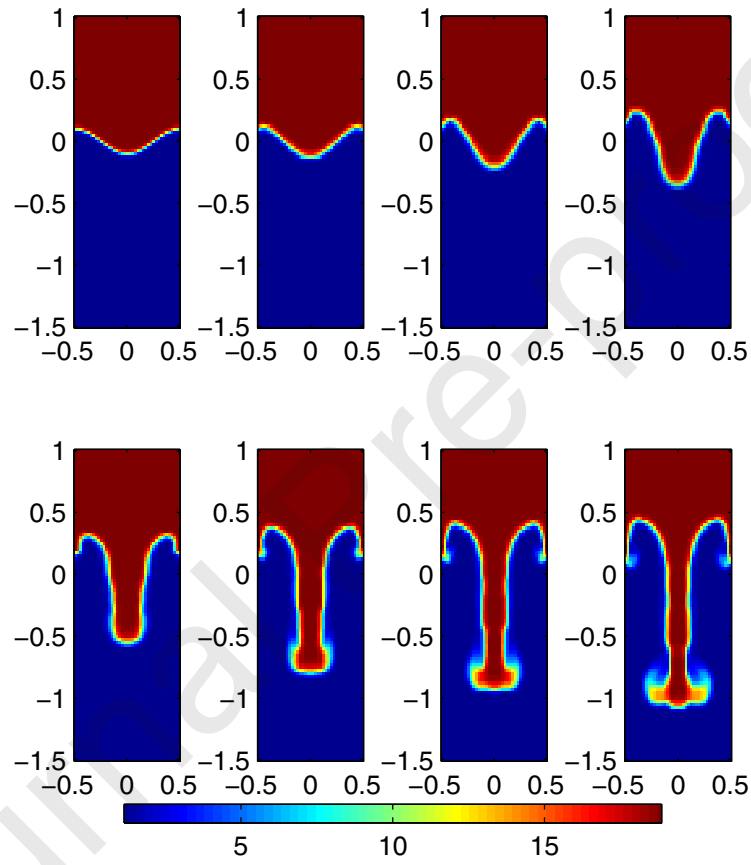


Fig. 4.7: Numerical solution of Rayleigh-Taylor instability with $At = 0.9$ at $t = 0s, 0.1s, 0.2s, 0.3s, 0.4s, 0.5s, 0.55s$ and $0.6s$ (from left to right, from top to bottom). Mesh: 20×50 .

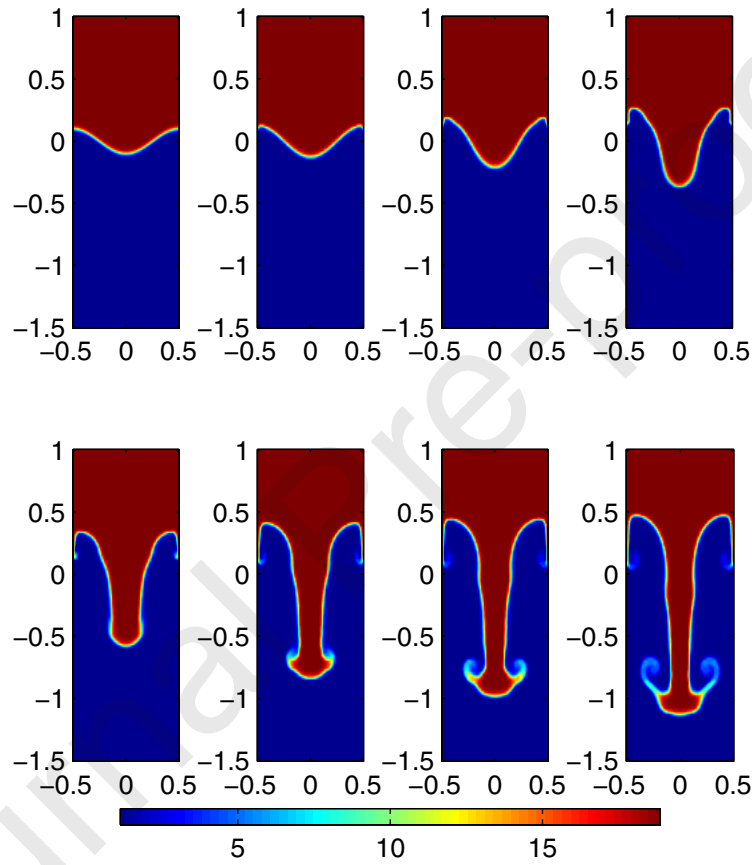


Fig. 4.8: Numerical solution of Rayleigh-Taylor instability with $At = 0.9$ at $t = 0s, 0.1s, 0.2s, 0.3s, 0.4s, 0.5s, 0.55s$ and $0.6s$ (from left to right, from top to bottom). Mesh: 40×100 .

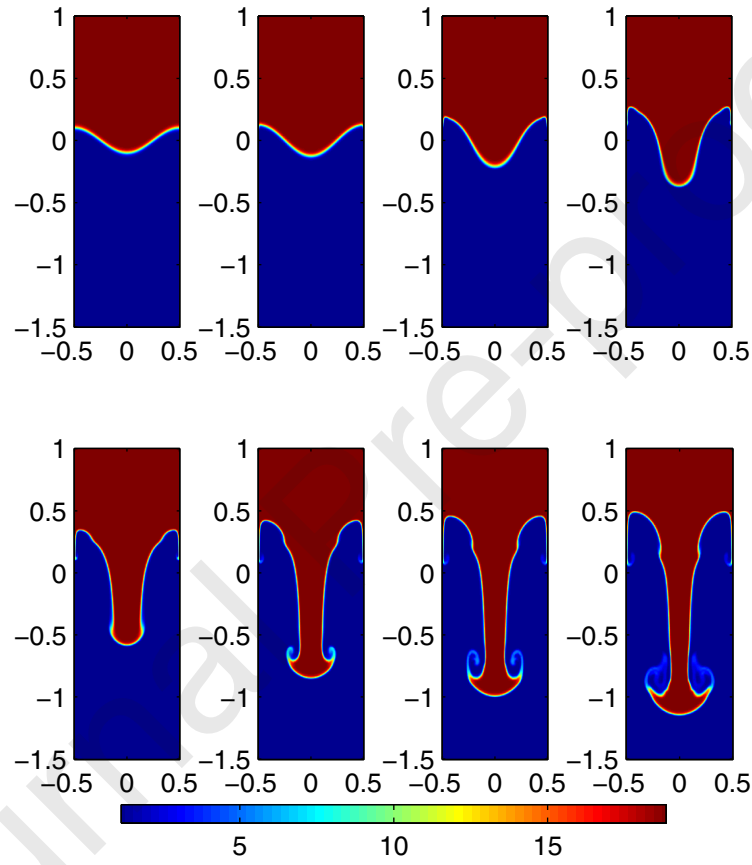


Fig. 4.9: Numerical solution of Rayleigh-Taylor instability with $At = 0.9$ at $t = 0s, 0.1s, 0.2s, 0.3s, 0.4s, 0.5s, 0.55s$ and $0.6s$ (from left to right, from top to bottom). Mesh: 80×200 .

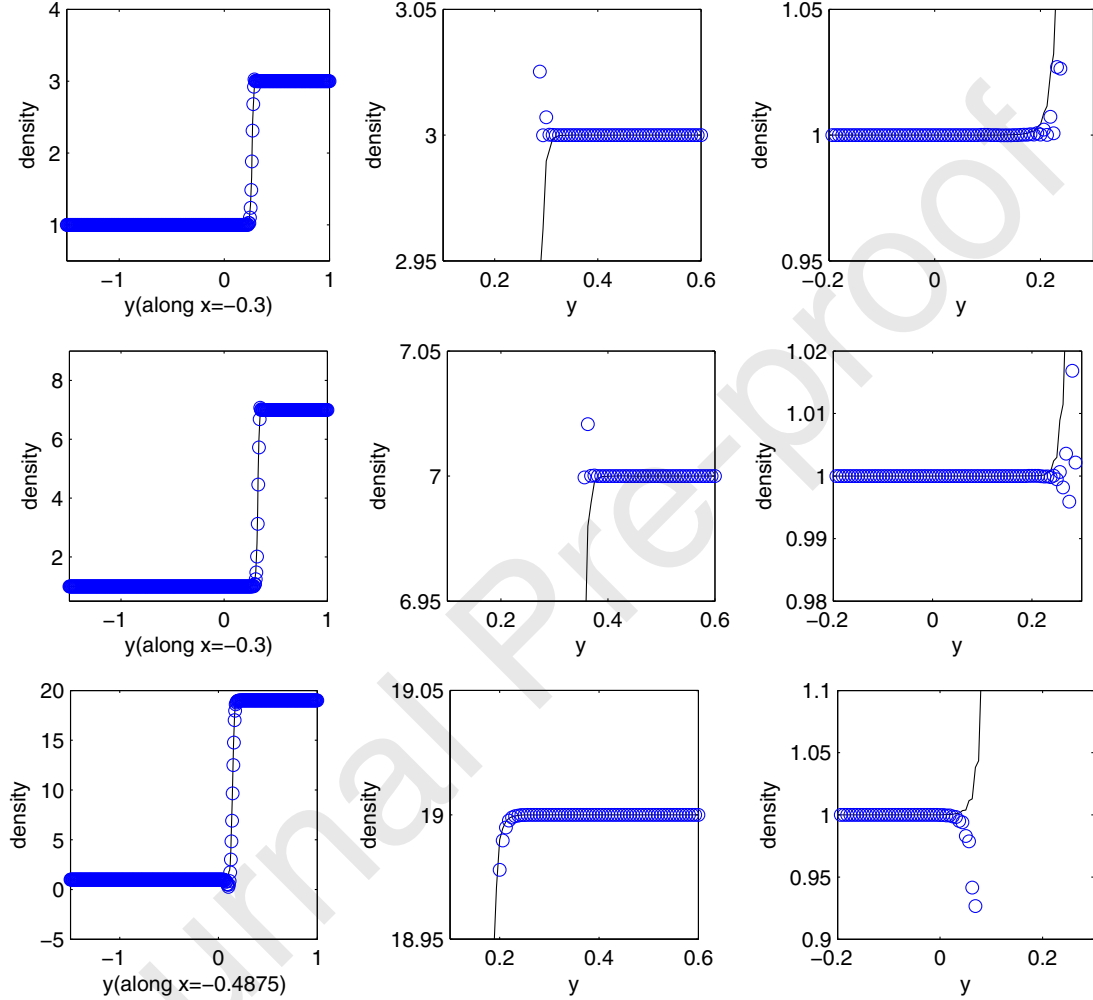


Fig. 4.10: Rayleigh-Taylor instability: comparison between numerical schemes with and without bound-preserving (BP) limiter. Solid line: with BP limiter; Circle: without BP limiter. Top: $At = 0.5$ and $t = 0.5s$. Middle: $At = 0.75$ and $t = 0.5s$. Bottom: $At = 0.9$ and $t = 0.15s$. The right column and middle column are partially enlarged lower and upper parts of the picture on the left column, respectively.

Table 4.1: Errors at time $t = 0.1$ for the scheme using Q^2 polynomials with $N_x \times N_y$ cells with third order Runge-Kutta methods.

$N_x \times N_y$	Explicit SSP Runge-Kutta				IMEX SSP Runge-Kutta			
	L^∞ error	Order	L^1 error	Order	L^∞ error	Order	L^1 error	Order
ρ	10×10	—	4.32e-4	—	6.08e-4	—	4.41e-4	—
	20×20	2.98	5.38e-5	3.01	8.20e-5	2.89	5.88e-5	2.91
	30×30	2.99	1.68e-5	2.87	2.52e-5	2.91	1.94e-5	2.73
	40×40	3.00	7.41e-6	2.85	1.34e-5	2.21	8.96e-6	2.69
	50×50	2.99	3.86e-6	2.92	8.16e-6	2.20	4.89e-6	2.71
	60×60	2.83	2.24e-6	2.98	5.40e-6	2.27	2.97e-6	2.75
	70×70	2.71	1.40e-6	3.05	3.77e-6	2.33	1.94e-6	2.77
u	10×10	—	2.71e-4	—	1.77e-4	—	2.08e-4	—
	20×20	2.95	3.41e-5	2.99	2.52e-5	2.81	3.28e-5	2.66
	30×30	2.96	1.01e-5	3.00	7.60e-6	2.95	9.85e-6	2.97
	40×40	2.98	4.28e-6	2.98	3.12e-6	3.10	4.01e-6	3.12
	50×50	2.93	2.19e-6	3.00	1.54e-6	3.16	1.94e-6	3.25
	60×60	2.95	1.27e-6	2.99	8.58e-7	3.22	1.05e-6	3.36
	70×70	2.94	8.01e-7	2.99	5.15e-7	3.31	6.22e-7	3.42
v	10×10	—	5.98e-5	—	1.77e-4	—	2.12e-4	—
	20×20	2.79	7.06e-6	3.08	2.63e-5	2.75	3.36e-5	2.66
	30×30	2.79	2.21e-6	2.86	8.14e-6	2.89	1.08e-5	2.80
	40×40	2.87	9.68e-7	2.87	3.61e-6	2.82	4.84e-6	2.79
	50×50	2.94	5.07e-7	2.90	1.98e-6	2.70	2.63e-6	2.73
	60×60	2.86	2.99e-7	2.90	1.22e-6	2.63	1.62e-6	2.66
	70×70	2.81	1.91e-7	2.91	8.23e-7	2.57	1.08e-6	2.59

4.3. Falling bubble test

In this subsection, we investigate the capability of our method to work with larger density variations, and consider the falling bubble problem [21]. In this problem, a heavy droplet falls through a light fluid and impacts into the plane surface of the heavy fluid in a cavity. The computational domain is $\Omega = [0, d] \times [0, 2d]$, where $d = 2m$ and the initial density interface takes the form

$$\rho(x, y, 0) = \begin{cases} 100kg/m^3, & \text{if } 0m \leq y \leq 1m \text{ or } 0m \leq r \leq 0.2m, \\ 1kg/m^3, & \text{if } 1m < y \leq 2m \text{ and } r \geq 0.2m, \end{cases}$$

where $r = \sqrt{(x-0.5)^2 + (y-1.75)^2}$. In the test, the gravity term is $\mathbf{f} = (0, \rho g)^\top$ with $g = -1m/s^2$, the viscosity of the fluid is assumed be constant in the whole domain and we have $Re = 1000$. The computational domain is divided into 50×100 cells.

The results are displayed in Figure 4.11. The snapshots show that the droplet travels up through a light fluid and merges with a light fluid below. As the droplet falls, its shape changes a little due to the absence of the surface tension. As the droplet hits the interface, it merges with the heavy fluid below and creates waves on the surface.

4.4. Rising bubble test

In this subsection, we test our numerical scheme by simulating an air bubble rising in water. Since the air and water have different viscosities. The air bubble with radius $0.0025m$ is initially at rest and located at $(0, 0.0075)$. The gravity term is $\mathbf{f} = (0, \rho g)^\top$ with $g = -9.80665m/s^2$. The initial condition for the velocity is set to be zero. We computed the problem in a rectangle of size $[-0.01, 0.01]m \times [0, 0.03]m$. We use the true physical parameters in Table 4.2. For numerical density taking intermediate values, the viscosity is defined as a linear interpolation.

The time evolution of the density field of the air bubble at nine different times $0s, 0.2s, 0.4s, 0.6s, 0.8s$ and $1.0s$ is displayed in Figure 4.12. These results are very close to the results reported in the literature [17, 25].

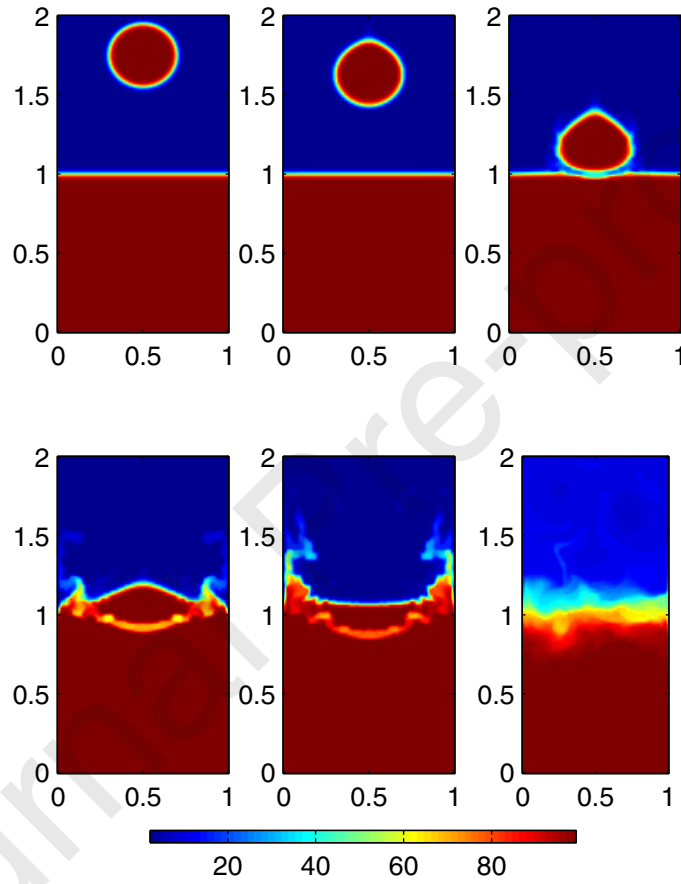


Fig. 4.11: Numerical solution of air bubble rising in water at $t = 0s, 0.5s, 1.1s, 1.3s, 1.5s$ and $20s$ (from left to right, from top to bottom). Mesh: 50×100 .

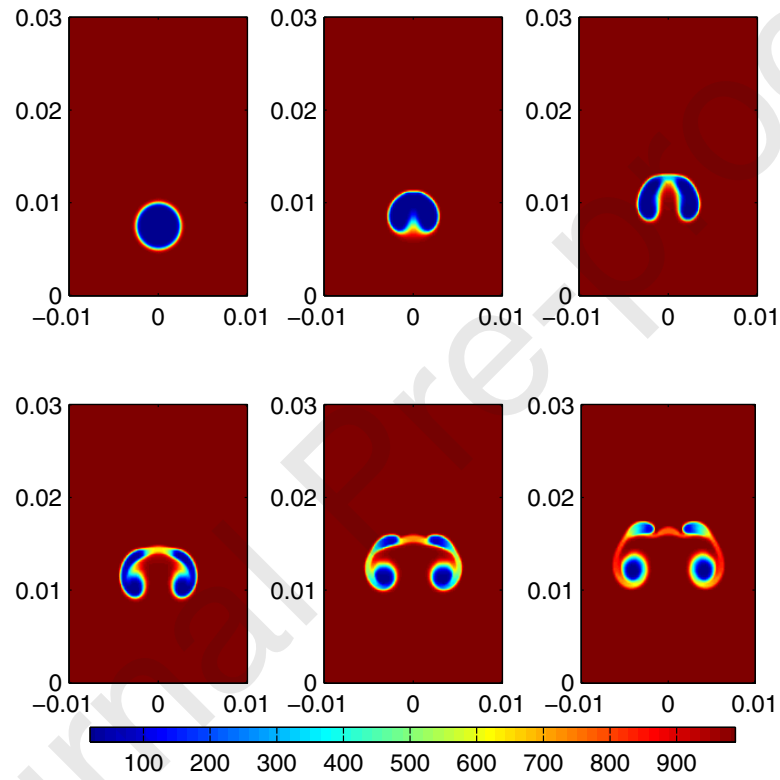


Fig. 4.12: Numerical solution of air bubble rising in water at $t = 0s, 0.02s, 0.04s, 0.06s, 0.08s$ and $0.1s$ (from left to right, from top to bottom). Mesh: 80×120 .

Table 4.2: Physical parameters for an air bubble rising in water.

Physical parameters	Air	Water	Units (MKS)
Density (ρ)	1.161	995.65	kg/m ³
Viscosity (μ)	0.0000186	0.0007977	kg/ms

5. Conclusions

We constructed in this paper a high order accurate numerical method for solving variable density incompressible Navier-Stokes equations in its pressure Poisson equation formulation. The bound-preserving discontinuous Galerkin scheme is used for the density evolution and a continuous finite element method is used for the momentum evolution. Numerical results indicate that the scheme is strictly bound-preserving and well suited for numerical simulation of incompressible flows with variable density.

Acknowledgments

The work of M. Li is partially supported by NSFC grant(Grant Nos. 11701055, 11871139). The work of Y. Cheng is partially supported by the Chongqing University 2016 "Graduate Student Research Innovation Project" (Project No. CYB16042). The work of J. Shen is partially supported by NSF Grant DMS-2012585 and by AFOSR FA9550-20-1-0309. The work of X. Zhang is supported by NSF grant DMS-1913120.

References

- [1] G. Akrivis and C. Lubich. Fully implicit, linearly implicit and implicit–explicit backward difference formulae for quasi-linear parabolic equations. *Numerische Mathematik*, 131(4):713–735, 2015.
- [2] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *Journal of computational Physics*, 142(1):1–46, 1998.
- [3] J. B. Bell and D. L. Marcus. A second-order projection method for variable-density flows. *Journal of Computational Physics*, 101(2):334–348, 1992.
- [4] S. Boscarino. On an accurate third order implicit-explicit Runge–Kutta method for stiff problems. *Applied Numerical Mathematics*, 59(7):1515–1528, 2009.
- [5] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.
- [6] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of computation*, 52(186):411–435, 1989.
- [7] B. Cockburn and C.-W. Shu. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.
- [8] M. O. Deville, P. F. Fischer, and E. H. Mund. *High-order methods for incompressible fluid flow*, volume 9. Cambridge university press, 2002.
- [9] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu. High order strong stability preserving time discretizations. *Journal of Scientific Computing*, 38(3):251–289, 2009.
- [10] J.-L. Guermond, P. Mineev, and J. Shen. An overview of projection methods for incompressible flows. *Computer methods in applied mechanics and engineering*, 195(44):6011–6045, 2006.
- [11] J.-L. Guermond and L. Quartapelle. A projection FEM for variable density incompressible flows. *Journal of Computational Physics*, 165(1):167–188, 2000.

- [12] J.-L. Guermond and A. Salgado. A splitting method for incompressible flows with variable density based on a pressure Poisson equation. *Journal of Computational Physics*, 228(8):2834–2846, 2009.
- [13] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.
- [14] J. Hu and R. Shu. On the uniform accuracy of implicit-explicit backward differentiation formulas (IMEX-BDF) for stiff hyperbolic relaxation systems and kinetic equations. *arXiv preprint arXiv:1912.00559*, 2019.
- [15] J. Hu, R. Shu, and X. Zhang. Asymptotic-preserving and positivity-preserving implicit-explicit schemes for the stiff bgk equation. *SIAM Journal on Numerical Analysis*, 56(2):942–973, 2018.
- [16] J. Hu and X. Zhang. On a class of implicit–explicit runge–kutta schemes for stiff kinetic equations preserving the navier–stokes limit. *Journal of Scientific Computing*, 73(2-3):797–818, 2017.
- [17] H. Johnston and J.-G. Liu. Finite difference schemes for incompressible flow based on local pressure boundary conditions. *Journal of Computational Physics*, 180(1):120–154, 2002.
- [18] H. Johnston and J.-G. Liu. Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term. *Journal of Computational Physics*, 199(1):221–259, 2004.
- [19] D. A. Kopriva. *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*. Springer Science & Business Media, 2009.
- [20] H. Li and X. Zhang. Superconvergence of high order finite difference schemes based on variational formulation for elliptic equations. *Journal of Scientific Computing*, 82(2):36, 2020.
- [21] Y. Li, L. Mei, J. Ge, and F. Shi. A new fractional time-stepping method for variable density incompressible flows. *Journal of Computational Physics*, 242:124–137, 2013.
- [22] P.-L. Lions and T. Moulden. *Mathematical topics in fluid mechanics, Volume 1: Incompressible models*, volume 50. AMERICAN SOCIETY OF MECHANICAL ENGINEERS, 1997.
- [23] J.-G. Liu and C.-W. Shu. A high-order discontinuous Galerkin method for 2D incompressible flows. *Journal of Computational Physics*, 160(2):577–596, 2000.
- [24] L. Pareschi and G. Russo. Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientific computing*, 25(1-2):129–155, 2005.
- [25] J.-H. Pyo and J. Shen. Gauge-Uzawa methods for incompressible flows with variable density. *Journal of Computational Physics*, 221(1):181–197, 2007.
- [26] G. Tryggvason. Numerical simulations of the Rayleigh-Taylor instability. *Journal of Computational Physics*, 75(2):253–282, 1988.
- [27] H. Wang, Y. Liu, Q. Zhang, and C.-W. Shu. Local discontinuous Galerkin methods with implicit-explicit time-marching for time-dependent incompressible fluid flow. *Mathematics of Computation*, 88(315):91–121, 2019.
- [28] H. Wang, C.-W. Shu, and Q. Zhang. Stability and Error Estimates of Local Discontinuous Galerkin Methods with Implicit-Explicit Time-Marching for Advection-Diffusion Problems. *SIAM Journal on Numerical Analysis*, 53(1):206–227, 2015.
- [29] H. Wang, C.-W. Shu, and Q. Zhang. Stability analysis and error estimates of local discontinuous Galerkin methods with implicit–explicit time-marching for nonlinear convection–diffusion problems. *Applied Mathematics and Computation*, 272:237–258, 2016.

- [30] H. Wang, S. Wang, Q. Zhang, and C.-W. Shu. Local discontinuous Galerkin methods with implicit-explicit time-marching for multi-dimensional convection-diffusion problems. *ESAIM: M2AN*, 50(4):1083–1105, 2016.
- [31] X. Zhang. On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations. *Journal of Computational Physics*, 328:301–343, 2017.
- [32] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, 2010.
- [33] X. Zhang and C.-W. Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 2010.
- [34] X. Zhang and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20110153. The Royal Society, 2011.
- [35] X. Zhang, Y. Xia, and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *Journal of Scientific Computing*, 50(1):29–62, 2012.

Declaration of interests

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

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

NONE.

Credit author statement

Maojun Li and Yongping Cheng: Implementation, Writing Original draft preparation. **Jie Shen and Xiangxiong Zhang:** Conceptualization, Methodology, Reviewing and Editing,