UNCONDITIONALLY STABLE, SECOND ORDER, DECOUPLED ENSEMBLE SCHEMES FOR COMPUTING EVOLUTIONARY BOUSSINESQ EQUATIONS

NAN JIANG* AND HUANHUAN YANG †

Abstract. In this report we present two unconditionally stable, second order, decoupled ensemble schemes for computing evolutionary Boussinesq equations: the stabilized scalar auxiliary variable Crank-Nicolson leap-frog ensemble scheme (Stab-SAV-CNLF-En) and the stabilized scalar auxiliary variable BDF2 ensemble scheme (Stab-SAV-BDF2-En). The two ensemble schemes adopt the recently developed stabilized scalar auxiliary variable (SAV) idea and ensemble timestepping to achieve both high efficiency and unconditional stability. Specifically, the stabilized SAV approach makes it possible to devise unconditionally long time stable schemes for which the nonlinear terms and coupling terms in the Boussinesq equations are made fully explicit, leading to linear systems with constant coefficient matrices to be solved after spatial discretization. The ensemble timestepping further improves the efficiency by making the coefficient matrices of all realizations the same, so that efficient block solvers can be applied to solve the corresponding one linear system with multiple right hand sides and thus greatly reduce the computational cost. We prove the proposed schemes are unconditionally stable and present implementation details. Ample numerical tests are performed to show the efficiency and effectiveness of the combined approach.

Key words. Boussinesq equations, ensemble algorithm, SAV, uncertainty quantification

1. Introduction.

Computing an ensemble of fluid flow equations with different initial conditions, boundary conditions or other model parameters are commonly seen in many scientific and engineering applications. For example, ensemble weather forecasting is made daily in Europe and US where an ensemble of flow simulations were run simultaneously with different initial conditions; uncertainty quantification (UQ) in many engineering and geophysical applications requires sampling an uncertain parameter and repeatedly simulating the governing partial differential equations (PDEs) with different sampled parameters. The main challenge in these ensemble simulations is the excessive computational cost, as many realistic flow problems are modeled by complex nonlinear PDEs which already pose great challenge to obtain an accurate numerical solution for just one model run. It is then highly desirable to have an efficient numerical scheme that can compute PDE ensembles with greatly reduced computational cost. To overcome this challenge, Jiang and Layton proposed an ensemble timestepping scheme [14] that runs all the simulations simultaneously and leads to one linear system with multiple right hands instead of multiple linear systems with multiple right hand sides. For such linear systems, block solvers, such as block CG, block GMRES, can be used to significantly reduce the computational cost. This ensemble timestepping idea has been further developed and tested for different flow problems including both nonlinear PDE problems, e.g., Navier-Stokes equations [10, 11, 12, 17, 14, 16, 37, 38, 23, 26, 27], MHD flows [3, 4, 22, 34], natural convection [6, 8, 15], and linear PDE problems, e.g., heat equation [7, 31, 32], Stokes-Darcy equations [13, 20, 21, 18, 19, 25], which have demonstrated that it is highly efficient and suitable for predictive flow simulations and many UQ applications.

In this report we propose and study two unconditionally stable, second order, decoupled ensemble schemes for computing the evolutionary Boussinesq equations. We will incorporate a recently developed scalar auxiliary variable (SAV) approach to treat the nonlinear terms and coupling terms fully explicit with the ensemble timesteping, leading to more efficient ensemble algorithms than the ones in [6, 8, 15], since the coefficient matrices of the corresponding linear systems after full discretization is both sample-index independent and time independent. The SAV approach was first studied in [35, 36] for gradient flows and later adapted to compute other PDE models, e.g., the Navier-Stokes equations [28, 29], natural convection [33]. It introduces extra SAVs and associated differential equations and reformulates the governing PDE system to facilitate the design of numerical schemes that maintain both unconditional stability and computational efficiency. We will prove our proposed ensemble algorithms are long time stable without any time step constraints and demonstrate their efficiency in the numerical tests.

*Department of Mathematics, University of Florida, Gainesville, FL 32611, jiangn@ufl.edu. This author was partially supported by the US National Science Foundation grants DMS-1720001, DMS-2120413, DMS-2143331.

†Corresponding author. Department of Mathematics, Shantou University, Guangdong, China 515063, huan2yang@stu.edu.cn. This author was supported by the Guangdong Basic and Applied Basic Research Foundation (2023A1515030199).

2. Ensemble Algorithms.

Let $u(x,t), p(x,t)$, and $\theta(x,t)$ denote the fluid velocity, pressure, and temperature respectively. We consider the context in which uncertainties may exist for the initial velocity, initial temperature, boundary conditions for velocity and temperature, forcing function and heat source, and J sample points $(u_j^0, \theta_j^0, a_j, b_j, f_j, g_j)$, $j = 1, ..., J$, have been drawn from respective stochastic spaces for each model parameter. The next step is to compute J corresponding solutions (u_j, p_j, θ_j) that satisfy the two-dimensional incompressible Boussinesq equations in an open bounded domain $\Omega \subset \mathbb{R}^2$ given by

$$\left\{\begin{aligned} \partial_t u_j + u_j \cdot \nabla u_j - \tfrac{1}{Re}\Delta u_j + \nabla p_j = Ri \cdot \theta_j \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} + f_j(x,t) &\quad \text{in } \Omega \times (0,T], \\ \nabla \cdot u_j = 0 &\quad \text{in } \Omega \times (0,T], \\ \partial_t \theta_j + u_j \cdot \nabla \theta_j - \tfrac{1}{RePr}\Delta \theta_j = g_j(x,t) &\quad \text{in } \Omega \times (0,T], \\ u_j = a_j(x,t) \text{ and } \theta_j = b_j(x,t) &\quad \text{on } \partial\Omega \times (0,T], \\ u_j = u_j^0(x) \text{ and } \theta_j = \theta_j^0(x) &\quad \text{in } \Omega \times \{0\}. \end{aligned}\right. \tag{2.1}$$

Here Re is the Reynolds number, Pr is the Prandtl number, which represents the ratio of kinematic viscosity to the heat conductivity, and Ri is the Richardson number that accounts for gravitational force and the thermal expansion of the flow.

We now introduce the SAVs which will be added to the Boussinesq equations to form a new equivalent governing PDE system. Define the scalar auxiliary variables $q_j(t)$ and $r_j(t)$ by

$$q_j(t) = exp(-\frac{t}{T}), \qquad r_j(t) = exp(-\frac{t}{T}), \qquad j = 1, \cdots, J. \tag{2.2}$$

Note that the true solutions of these SAVs are the same for all ensemble members, but the numerical solutions will be different for each j. Using the equalities $\int_\Omega (u_j \cdot \nabla) u_j \cdot u_j \, dx = \int_{\partial\Omega} (\vec{n} \cdot u_j) \frac{1}{2} |u_j|^2 \, d\sigma$ and $\int_\Omega (u_j \cdot \nabla) \theta_j \cdot \theta_j \, dx = \int_{\partial\Omega} (\vec{n} \cdot u_j) \frac{1}{2} |\theta_j|^2 \, d\sigma$, these SAVs satisfy the following differential equations.

$$\frac{dq_j}{dt} = -\frac{1}{T} q_j + \frac{1}{exp(-\frac{t}{T})} \int_\Omega (u_j \cdot \nabla) u_j \cdot u_j \, dx - \frac{1}{q_j} \int_{\partial\Omega} (\vec{n} \cdot u_j) \tfrac{1}{2} |u_j|^2 \, d\sigma, \tag{2.3}$$

$$\frac{dr_j}{dt} = -\frac{1}{T} r_j + \frac{1}{exp(-\frac{t}{T})} \int_\Omega (u_j \cdot \nabla) \theta_j \cdot \theta_j \, dx - \frac{1}{r_j} \int_{\partial\Omega} (\vec{n} \cdot u_j) \tfrac{1}{2} |\theta_j|^2 \, d\sigma. \tag{2.4}$$

Adding these two differential equations to the original Boussinesq equations and using $\frac{q_j(t)}{exp(-\frac{t}{T})} = 1$ and $\frac{r_j(t)}{exp(-\frac{t}{T})} = 1$ as the coefficients for the nonlinear terms, we have a new governing system that is equivalent to (2.1).

$$\left\{\begin{aligned} \partial_t u_j + \tfrac{q_j(t)}{exp(-\frac{t}{T})} u_j \cdot \nabla u_j - \tfrac{1}{Re}\Delta u_j + \nabla p_j = Ri \cdot \theta_j \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} + f_j(x,t) &\quad \text{in } \Omega \times (0,T], \\ \nabla \cdot u_j = 0 &\quad \text{in } \Omega \times (0,T], \\ \partial_t \theta_j + \tfrac{r_j(t)}{exp(-\frac{t}{T})} u_j \cdot \nabla \theta_j - \tfrac{1}{RePr}\Delta \theta_j = g_j(x,t) &\quad \text{in } \Omega \times (0,T], \\ \tfrac{dq_j}{dt} = -\tfrac{1}{T} q_j + \tfrac{1}{exp(-\frac{t}{T})} \textstyle\int_\Omega (u_j \cdot \nabla) u_j \cdot u_j \, dx - \tfrac{1}{q_j} \textstyle\int_{\partial\Omega} (\vec{n} \cdot a_j) \tfrac{1}{2} |a_j|^2 \, d\sigma &\quad \text{in } (0,T], \\ \tfrac{dr_j}{dt} = -\tfrac{1}{T} r_j + \tfrac{1}{exp(-\frac{t}{T})} \textstyle\int_\Omega (u_j \cdot \nabla) \theta_j \cdot \theta_j \, dx - \tfrac{1}{r_j} \textstyle\int_{\partial\Omega} (\vec{n} \cdot a_j) \tfrac{1}{2} |b_j|^2 \, d\sigma &\quad \text{in } (0,T]. \end{aligned}\right. \tag{2.5}$$

For this equivalent governing system we propose two unconditionally stable, second order ensemble schemes for fast computation of Boussinesq flow ensembles. The SAV schemes are known to have relatively low accuracy for complex nonlinear flow problems, so we incorporate a stabilization technique from [24] in the proposed schemes to stabilize the flows and increase the accuracy of the schemes.

Let $t^n = n\Delta t$, $n = 0, 1, 2, \cdots, N$, where $N = T/\Delta t$, denote a uniform partition of the interval $[0, T]$, and h the mesh size of the chosen spatial discretization. The stabilized SAV ensemble scheme based on Crank–Nicolson leap-frog (CNLF) that we propose is then given by

Algorithm 2.1 (Stab-SAV-CNLF-En).

(Sub-problem 1) Given u_j^{n-1}, u_j^n, and θ_j^n, find the velocity u_j^{n+1} such that

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + \frac{q_j^{n+1} + q_j^{n-1}}{2exp(-\frac{t^n}{T})} (u_j^n \cdot \nabla) u_j^n - \frac{1}{Re} \Delta \frac{u_j^{n+1} + u_j^{n-1}}{2} + \nabla \frac{p_j^{n+1} + p_j^{n-1}}{2} \tag{2.6}$$

$$-\alpha h\Delta(u_j^{n+1}-u_j^{n-1}) = Ri\cdot\theta_j^n\cdot\begin{pmatrix}0\\1\end{pmatrix}+f_j^n,$$

$$\nabla\cdot u_j^{n+1}=0, \tag{2.7}$$

$$\frac{q_j^{n+1}-q_j^{n-1}}{2\Delta t} = -\frac{1}{T}\frac{q_j^{n+1}+q_j^{n-1}}{2}+\frac{1}{exp(-\frac{t^n}{T})}\int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot\frac{u_j^{n+1}+u_j^{n-1}}{2}\,dx-\frac{2c_j^n}{q_j^{n+1}+q_j^{n-1}}, \tag{2.8}$$

where $c_j^n=\int_{\partial\Omega}(\vec{n}\cdot a_j^n)\frac{1}{2}|a_j^n|^2\,d\sigma$, *and* $\alpha>0$ *is a tuning stabilization parameter.*

(Sub-problem 2) Given u_j^n *and* $\theta_j^{n-1},\theta_j^n$, *find* θ_j^{n+1} *such that*

$$\frac{\theta_j^{n+1}-\theta_j^{n-1}}{2\Delta t}-\frac{1}{RePr}\Delta\frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}+\frac{r_j^{n+1}+r_j^{n-1}}{2exp(-\frac{t^n}{T})}u^n\cdot\nabla\theta_j^n-\beta h\Delta(\theta_j^{n+1}-\theta_j^{n-1})=g_j^n, \tag{2.9}$$

$$\frac{r_j^{n+1}-r_j^{n-1}}{2\Delta t} = -\frac{1}{T}\frac{r_j^{n+1}+r_j^{n-1}}{2}+\frac{1}{exp(-\frac{t^n}{T})}\int_\Omega (u_j^n\cdot\nabla)\theta_j^n\cdot\frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}\,dx-\frac{2d_j^n}{r_j^{n+1}+r_j^{n-1}}, \tag{2.10}$$

where $d_j^n=\int_{\partial\Omega}(\vec{n}\cdot a_j^n)\frac{1}{2}|b_j^n|^2\,d\sigma$, *and* $\beta>0$ *is a tuning stabilization parameter.*

The stabilized SAV ensemble scheme based on BDF2 that we propose is given by
Algorithm 2.2 (Stab-SAV-BDF2-En).
(Sub-problem 1) Given u_j^{n-1},u_j^n, *and* $\theta_j^{n-1},\theta_j^n$, *find the velocity* u_j^{n+1} *such that*

$$\begin{aligned}\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t}&+\frac{q_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}((2u_j^n-u_j^{n-1})\cdot\nabla)(2u_j^n-u_j^{n-1})-\frac{1}{Re}\Delta u_j^{n+1}+\nabla p_j^{n+1}\\&-\alpha h\Delta(3u_j^{n+1}-4u_j^n+u_j^{n-1})=Ri\cdot(2\theta_j^n-\theta_j^{n-1})\cdot\begin{pmatrix}0\\1\end{pmatrix}+f_j^{n+1},\end{aligned} \tag{2.11}$$

$$\nabla\cdot u_j^{n+1}=0, \tag{2.12}$$

$$\frac{3q_j^{n+1}-4q_j^n+q_j^{n-1}}{2\Delta t}=-\frac{1}{T}q_j^{n+1}+\frac{1}{exp(-\frac{t^{n+1}}{T})}\int_\Omega((2u_j^n-u_j^{n-1})\cdot\nabla)(2u_j^n-u_j^{n-1})\cdot u_j^{n+1}\,dx-\frac{c_j^{n+1}}{q_j^{n+1}}, \tag{2.13}$$

where $c_j^{n+1}=\int_{\partial\Omega}(\vec{n}\cdot a_j^{n+1})\frac{1}{2}|a_j^{n+1}|^2\,d\sigma$, *and* $\alpha>0$ *is a tuning stabilization parameter.*

(Sub-problem 2) Given u_j^{n-1},u_j^n *and* $\theta_j^{n-1},\theta_j^n$, *find* θ_j^{n+1} *such that*

$$\begin{aligned}\frac{3\theta_j^{n+1}-4\theta_j^n+\theta_j^{n-1}}{2\Delta t}&-\frac{1}{RePr}\Delta\theta_j^{n+1}+\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}(2u_j^n-u_j^{n-1})\cdot\nabla(2\theta_j^n-\theta_j^{n-1})\\&-\beta h\Delta(3\theta_j^{n+1}-4\theta_j^n+\theta_j^{n-1})=g_j^{n+1},\end{aligned} \tag{2.14}$$

$$\frac{3r_j^{n+1}-4r_j^n+r_j^{n-1}}{2\Delta t}=-\frac{1}{T}r_j^{n+1}+\frac{1}{exp(-\frac{t^{n+1}}{T})}\int_\Omega((2u_j^n-u_j^{n-1})\cdot\nabla)(2\theta_j^n-\theta_j^{n-1})\cdot\theta_j^{n+1}\,dx-\frac{d_j^{n+1}}{r_j^{n+1}}, \tag{2.15}$$

where $d_j^{n+1}=\int_{\partial\Omega}(\vec{n}\cdot a_j^{n+1})\frac{1}{2}|b_j^{n+1}|^2\,d\sigma$, *and* $\beta>0$ *is a tuning stabilization parameter.*

One can easily see that all ensemble members share the same constant coefficient matrix in all steps, and thus efficient direct or iterative solvers [5, 9] can be used for fast computation. In these two schemes, the SAVs are coupled together with the primary variables. We will present equivalent, fully decoupled implementation algorithms for both Stab-SAV-CNLF-En and Stab-SAV-BDF2-En in Section 4.

3. Unconditional Stability. In this section, we prove that Algorithm 2.1 and Algorithm 2.2 are both unconditionally long time stable.

Theorem 3.1 (Unconditional Stability of Algorithm 2.1). *Consider the stabilized SAV-CNLF ensemble scheme. With homogeneous Dirichlet boundary conditions, Algorithm 2.1 is unconditionally long time stable, and the following energy inequalities hold. For any $N \geq 2$,*

$$\begin{aligned}
&\|\theta_j^N\|^2 + \|\theta_j^{N-1}\|^2 + |r_j^N|^2 + |r_j^{N-1}|^2 + \Delta t \sum_{n=1}^{N-1} \frac{2}{RePr}\|\nabla \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}\|^2 \\
&\quad + 2\beta h\Delta t\|\nabla\theta_j^N\|^2 + 2\beta h\Delta t\|\nabla\theta_j^{N-1}\|^2 + \Delta t\sum_{n=1}^{N-1}\frac{4}{T}|\frac{r_j^{n+1}+r_j^{n-1}}{2}|^2 \\
&\leq \|\theta_j^1\|^2 + \|\theta_j^0\|^2 + |r_j^1|^2 + |r_j^0|^2 + 2\beta h\Delta t\|\nabla\theta_j^1\|^2 + 2\beta h\Delta t\|\nabla\theta_j^0\|^2 + \Delta t\sum_{n=1}^{N-1} 2C_P^2 RePr\|g_j^n\|^2 \ ,
\end{aligned} \tag{3.1}$$

and

$$\begin{aligned}
&\|u_j^N\|^2 + \|u_j^{N-1}\|^2 + |q_j^N|^2 + |q_j^{N-1}|^2 + \Delta t\sum_{n=1}^{N-1}\frac{2}{Re}\|\nabla\frac{u_j^{n+1}+u_j^{n-1}}{2}\|^2 \\
&\quad + 2\alpha h\Delta t\|\nabla u_j^N\|^2 + 2\alpha h\Delta t\|\nabla u_j^{N-1}\|^2 + \Delta t\sum_{n=2}^{N-1}\frac{4}{T}|\frac{q_j^{n+1}+q_j^{n-1}}{2}|^2 \\
&\leq \|u_j^1\|^2 + \|u_j^0\|^2 + |q_j^1|^2 + |q_j^0|^2 + 2\alpha h\Delta t\|\nabla u_j^1\|^2 + 2\alpha h\Delta t\|\nabla u_j^0\|^2 \\
&\quad + \sum_{n=2}^{N-1} 4C_P^2 Ri^2 Re\Delta t\Bigg(\|\theta_j^1\|^2 + \|\theta_j^0\|^2 + |r_j^1|^2 + |r_j^0|^2 + 2\beta h\Delta t\|\nabla\theta_j^1\|^2 + 2\beta h\Delta t\|\nabla\theta_j^0\|^2 \\
&\quad + \Delta t\sum_{k=1}^{n-1} 2C_P^2 RePr\|g_j^k\|^2\Bigg) + 4C_P^2 Ri^2 Re\Delta t\|\theta_j^1\|^2 + \Delta t\sum_{n=1}^{N-1} 4C_P^2 Re\|f_j^n\|^2.
\end{aligned} \tag{3.2}$$

Proof. Taking the L^2 inner product of (2.6) and (2.9) with $\frac{u_j^{n+1}+u_j^{n-1}}{2}$ and $\frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}$ respectively gives

$$\begin{aligned}
&\frac{1}{4\Delta t}\|u_j^{n+1}\|^2 - \frac{1}{4\Delta t}\|u_j^{n-1}\|^2 + \frac{q_j^{n+1}+q_j^{n-1}}{2exp(-\frac{t^n}{T})}(u_j^n\cdot\nabla u_j^n, \frac{u_j^{n+1}+u_j^{n-1}}{2}) + \frac{1}{Re}\|\nabla\frac{u_j^{n+1}+u_j^{n-1}}{2}\|^2 \\
&\quad -\frac{1}{Re}\int_{\partial\Omega}(\vec{n}\cdot\nabla\frac{u_j^{n+1}+u_j^{n-1}}{2})\cdot\frac{u_j^{n+1}+u_j^{n-1}}{2}d\sigma + \int_{\partial\Omega}(\vec{n}\cdot\frac{u_j^{n+1}+u_j^{n-1}}{2})\frac{p_j^{n+1}+p_j^{n-1}}{2}d\sigma \\
&\quad -\alpha h\int_{\partial\Omega}(\vec{n}\cdot(\nabla u_j^{n+1}-\nabla u_j^{n-1}))\cdot\frac{u_j^{n+1}+u_j^{n-1}}{2}d\sigma + \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right) \\
&\quad = Ri\cdot(\theta_j^n\cdot\begin{pmatrix}0\\1\end{pmatrix}, \frac{u_j^{n+1}+u_j^{n-1}}{2}) + (f_j^n, \frac{u_j^{n+1}+u_j^{n-1}}{2}),
\end{aligned} \tag{3.3}$$

and

$$\begin{aligned}
&\frac{1}{4\Delta t}\|\theta_j^{n+1}\|^2 - \frac{1}{4\Delta t}\|\theta_j^{n-1}\|^2 + \frac{1}{RePr}\|\nabla\frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}\|^2 + \frac{r_j^{n+1}+r_j^{n-1}}{2exp(-\frac{t^n}{T})}(u_j^n\cdot\nabla\theta_j^n, \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}) \\
&-\frac{1}{RePr}\int_{\partial\Omega}(\vec{n}\cdot\nabla\frac{\theta_j^{n+1}+\theta_j^{n-1}}{2})\cdot\frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}d\sigma - \beta h\int_{\partial\Omega}(\vec{n}\cdot(\nabla\theta_j^{n+1}-\nabla\theta_j^{n-1}))\cdot\frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}d\sigma \\
&\quad +\frac{\beta}{2}h\left(\|\nabla\theta_j^{n+1}\|^2 - \|\nabla\theta_j^{n-1}\|^2\right) = (g_j^n, \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}).
\end{aligned} \tag{3.4}$$

The terms on the right hand side can be bounded as follows.

$$\begin{aligned} Ri \cdot (\theta_j^n \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \frac{u_j^{n+1}+u_j^{n-1}}{2}) &\le C_P Ri \|\theta_j^n\| \|\nabla \frac{u_j^{n+1}+u_j^{n-1}}{2}\| \\ &\le \frac{1}{4Re}\|\nabla \frac{u_j^{n+1}+u_j^{n-1}}{2}\|^2 + C_P^2 Ri^2 Re \|\theta_j^n\|^2 \end{aligned} \tag{3.5}$$

$$(f_j^n, \frac{u_j^{n+1}+u_j^{n-1}}{2}) \le C_P \|f_j^n\| \|\nabla \frac{u_j^{n+1}+u_j^{n-1}}{2}\| \le \frac{1}{4Re}\|\nabla \frac{u_j^{n+1}+u_j^{n-1}}{2}\|^2 + C_P^2 Re \|f_j^n\|^2, \tag{3.6}$$

$$(g_j^n, \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}) \le C_P \|g_j^n\| \|\nabla \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}\| \le \frac{1}{2RePr}\|\nabla \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}\|^2 + \frac{1}{2} C_P^2 RePr \|g_j^n\|^2. \tag{3.7}$$

Multiplying (2.8) with $\frac{q_j^{n+1}+q_j^{n-1}}{2}$ and (2.10) with $\frac{r_j^{n+1}+r_j^{n-1}}{2}$ gives

$$\frac{1}{4\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) = -\frac{1}{T}|\frac{q_j^{n+1}+q_j^{n-1}}{2}|^2 + \frac{q_j^{n+1}+q_j^{n-1}}{2exp(-\frac{t^n}{T})}(u_j^n \cdot \nabla u_j^n, \frac{u_j^{n+1}+u_j^{n-1}}{2}) - c_j^n, \tag{3.8}$$

and

$$\frac{1}{4\Delta t}\left(|r_j^{n+1}|^2 - |r_j^{n-1}|^2\right) = -\frac{1}{T}|\frac{r_j^{n+1}+r_j^{n-1}}{2}|^2 + \frac{r_j^{n+1}+r_j^{n-1}}{2exp(-\frac{t^n}{T})}(u_j^n \cdot \nabla \theta_j^n, \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}) - d_j^n. \tag{3.9}$$

For simplicity of presentation, we now assume homogeneous Dirichlet boundary conditions. With the bounds (3.5)-(3.7), adding (3.3) and (3.8), (3.4) and (3.9) gives

$$\begin{aligned} &\frac{1}{4\Delta t}\|u_j^{n+1}\|^2 - \frac{1}{4\Delta t}\|u_j^{n-1}\|^2 + \frac{1}{4\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) + \frac{1}{2Re}\|\nabla \frac{u_j^{n+1}+u_j^{n-1}}{2}\|^2 \\ &+ \frac{\alpha}{2} h \left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right) + \frac{1}{T}|\frac{q_j^{n+1}+q_j^{n-1}}{2}|^2 \le C_P^2 Ri^2 Re \|\theta_j^n\|^2 + C_P^2 Re \Delta t \|f_j^n\|^2 \ , \end{aligned} \tag{3.10}$$

and

$$\begin{aligned} &\frac{1}{4\Delta t}\|\theta_j^{n+1}\|^2 - \frac{1}{4\Delta t}\|\theta_j^{n-1}\|^2 + \frac{1}{4\Delta t}\left(|r_j^{n+1}|^2 - |r_j^{n-1}|^2\right) + \frac{1}{2RePr}\|\nabla \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}\|^2 \\ &+ \frac{\beta}{2} h \left(\|\nabla \theta_j^{n+1}\|^2 - \|\nabla \theta_j^{n-1}\|^2\right) + \frac{1}{T}|\frac{r_j^{n+1}+r_j^{n-1}}{2}|^2 \le \frac{1}{2} C_P^2 RePr \|g_j^n\|^2 \ . \end{aligned} \tag{3.11}$$

Summing up (3.11) from $n = 1$ to $n = N - 1$ and multiplying through by $4\Delta t$ gives

$$\begin{aligned} &\|\theta_j^N\|^2 + \|\theta_j^{N-1}\|^2 + |r_j^N|^2 + |r_j^{N-1}|^2 + \Delta t \sum_{n=1}^{N-1} \frac{2}{RePr}\|\nabla \frac{\theta_j^{n+1}+\theta_j^{n-1}}{2}\|^2 \\ &+ 2\beta h \Delta t \|\nabla \theta_j^N\|^2 + 2\beta h \Delta t \|\nabla \theta_j^{N-1}\|^2 + \Delta t \sum_{n=1}^{N-1} \frac{4}{T}|\frac{r_j^{n+1}+r_j^{n-1}}{2}|^2 \\ &\le \|\theta_j^1\|^2 + \|\theta_j^0\|^2 + |r_j^1|^2 + |r_j^0|^2 + 2\beta h \Delta t \|\nabla \theta_j^1\|^2 + 2\beta h \Delta t \|\nabla \theta_j^0\|^2 + \Delta t \sum_{n=1}^{N-1} 2C_P^2 RePr \|g_j^n\|^2 \ . \end{aligned} \tag{3.12}$$

Summing up (3.10) from $n = 1$ to $n = N - 1$, multiplying through by $4\Delta t$, and using the stability result for the temperature from (3.12) yields

$$\|u_j^N\|^2 + \|u_j^{N-1}\|^2 + |q_j^N|^2 + |q_j^{N-1}|^2 + \Delta t \sum_{n=1}^{N-1} \frac{2}{Re}\|\nabla \frac{u_j^{n+1}+u_j^{n-1}}{2}\|^2 \tag{3.13}$$

$$
\begin{aligned}
&\quad + 2\alpha h\Delta t\|\nabla u_j^N\|^2 + 2\alpha h\Delta t\|\nabla u_j^{N-1}\|^2 + \Delta t\sum_{n=2}^{N-1}\frac{4}{T}|\frac{q_j^{n+1}+q_j^{n-1}}{2}|^2 \\
&\le \|u_j^1\|^2 + \|u_j^0\|^2 + |q_j^1|^2 + |q_j^0|^2 + 2\alpha h\Delta t\|\nabla u_j^1\|^2 + 2\alpha h\Delta t\|\nabla u_j^0\|^2 + \Delta t\sum_{n=1}^{N-1} 4C_P^2 Ri^2 Re\|\theta_j^n\|^2 \\
&\quad + \Delta t\sum_{n=1}^{N-1} 4C_P^2 Re\|f_j^n\|^2 \\
&\le \|u_j^1\|^2 + \|u_j^0\|^2 + |q_j^1|^2 + |q_j^0|^2 + 2\alpha h\Delta t\|\nabla u_j^1\|^2 + 2\alpha h\Delta t\|\nabla u_j^0\|^2 \\
&\quad + \sum_{n=2}^{N-1} 4C_P^2 Ri^2 Re\Delta t\|\theta_j^n\|^2 + 4C_P^2 Ri^2 Re\Delta t\|\theta_j^1\|^2 + \Delta t\sum_{n=1}^{N-1} 4C_P^2 Re\|f_j^n\|^2 \\
&\le \|u_j^1\|^2 + \|u_j^0\|^2 + |q_j^1|^2 + |q_j^0|^2 + 2\alpha h\Delta t\|\nabla u_j^1\|^2 + 2\alpha h\Delta t\|\nabla u_j^0\|^2 \\
&\quad + \sum_{n=2}^{N-1} 4C_P^2 Ri^2 Re\Delta t\Bigg(\|\theta_j^1\|^2 + \|\theta_j^0\|^2 + |r_j^1|^2 + |r_j^0|^2 + 2\beta h\Delta t\|\nabla\theta_j^1\|^2 + 2\beta h\Delta t\|\nabla\theta_j^0\|^2 \\
&\quad + \Delta t\sum_{k=1}^{n-1} 2C_P^2 RePr\|g_j^k\|^2\Bigg) + 4C_P^2 Ri^2 Re\Delta t\|\theta_j^1\|^2 + \Delta t\sum_{n=1}^{N-1} 4C_P^2 Re\|f_j^n\|^2.
\end{aligned}
$$

With homogeneous Dirichlet boundary conditions, q_j^n, r_j^n are real and thus $|q_j^n|, |r_j^n|$ are positive (see Theorem 4.1 and Theorem 4.2).

This concludes the proof. □

Next we prove that Algorithm 2.2 is unconditionally long time stable.

Theorem 3.2 (Unconditional Stability of Algorithm 2.2). *Consider the stabilized SAV-BDF2 ensemble scheme. With homogeneous Dirichlet boundary conditions, Algorithm 2.2 is unconditionally long time stable, and the following energy inequalities hold. For any $N \geq 2$,*

$$
\begin{aligned}
&\|\theta_j^N\|^2 + \|2\theta_j^N - \theta_j^{N-1}\|^2 + \sum_{n=1}^{N-1}\|\theta_j^{n+1} - 2\theta_j^n + \theta_j^{n-1}\|^2 + |r_j^N|^2 + |2r_j^N - r_j^{N-1}|^2 \qquad (3.14) \\
&+ \sum_{n=1}^{N-1}\|r_j^{n+1} - 2r_j^n + r_j^{n-1}\|^2 + \Delta t\sum_{n=1}^{N-1}\frac{2}{RePr}\|\nabla\theta_j^{n+1}\|^2 + \Delta t\sum_{n=1}^{N-1}\frac{4}{T}|r_j^{n+1}|^2 \\
&\quad + 2\beta h\Delta t\|\nabla\theta_j^N\|^2 + 2\beta h\Delta t\|2\nabla\theta_j^N - \nabla\theta_j^{N-1}\|^2 + \Delta t\sum_{n=1}^{N-1} 2\beta h\|\nabla\theta_j^{n+1} - 2\nabla\theta_j^n + \nabla\theta_j^{n-1}\|^2 \\
&\le \|\theta_j^1\|^2 + \|2\theta_j^1 - \theta_j^0\|^2 + |r_j^1|^2 + |2r_j^1 - r_j^0|^2 + 2\beta h\Delta t\|\nabla\theta_j^1\|^2 + 2\beta h\Delta t\|\nabla 2\theta_j^1 - \theta_j^0\|^2 \\
&\quad + \Delta t\sum_{n=1}^{N-1} 2C_P^2 RePr\|g_j^{n+1}\|^2\ ,
\end{aligned}
$$

and

$$
\begin{aligned}
&\|u_j^N\|^2 + \|2u_j^N - u_j^{N-1}\|^2 + \sum_{n=1}^{N-1}\|u_j^{n+1} - 2u_j^n + u_j^{n-1}\|^2 + |q_j^N|^2 + |2q_j^N - q_j^{N-1}|^2 \qquad (3.15) \\
&+ \sum_{n=1}^{N-1}\|q_j^{n+1} - 2q_j^n + q_j^{n-1}\|^2 + \Delta t\sum_{n=1}^{N-1}\frac{2}{Re}\|\nabla u_j^{n+1}\|^2 + \Delta t\sum_{n=2}^{N-1}\frac{4}{T}|q_j^{n+1}|^2 \\
&+ 2\alpha h\Delta t\|\nabla u_j^N\|^2 + 2\alpha h\Delta t\|2\nabla u_j^N - \nabla u_j^{N-1}\|^2 + \Delta t\sum_{n=1}^{N-1} 2\alpha h\|\nabla u_j^{n+1} - 2\nabla u_j^n + \nabla u_j^{n-1}\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq \|u_j^1\|^2 + \|2u_j^1 - u_j^0\|^2 + |q_j^1|^2 + |2q_j^1 - q_j^0|^2 + 2\alpha h\Delta t\|\nabla u_j^1\|^2 + 2\alpha h\Delta t\|2\nabla u_j^1 - \nabla u_j^0\|^2 \\
&\quad + \sum_{n=2}^{N-1} 36C_P^2 Ri^2 Re\Delta t\Bigg(\|\theta_j^1\|^2 + \|2\theta_j^1 - \theta_j^0\|^2 + |r_j^1|^2 + |2r_j^1 - r_j^0|^2 + 2\beta h\Delta t\|\nabla\theta_j^1\|^2 \\
&\quad + 2\beta h\Delta t\|\nabla 2\theta_j^1 - \theta_j^0\|^2 + \Delta t\sum_{k=1}^{n-1} 2C_P^2 RePr\|g_j^{k+1}\|^2\Bigg) + 36C_P^2 Ri^2 Re\Delta t\|\theta_j^1\|^2 + 12C_P^2 Ri^2 Re\Delta t\|\theta_j^0\|^2 \\
&\quad + \Delta t\sum_{n=1}^{N-1} 4C_P^2 Re\|f_j^{n+1}\|^2.
\end{aligned}
$$

Proof. Taking the L^2 inner product of (2.11) and (2.14) with u_j^{n+1} and θ_j^{n+1} respectively gives

$$
\begin{aligned}
&\frac{1}{4\Delta t}\left(\|u_j^{n+1}\|^2 + \|2u_j^{n+1} - u_j^n\|^2\right) - \frac{1}{4\Delta t}\left(\|u_j^n\|^2 + \|2u_j^n - u_j^{n-1}\|^2\right) + \frac{1}{4\Delta t}\|u_j^{n+1} - 2u_j^n + u_j^{n-1}\|^2 \\
&\quad + \frac{q_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}((2u_j^n - u_j^{n-1})\cdot\nabla(2u_j^n - u_j^{n-1}), u_j^{n+1}) + \frac{1}{Re}\|\nabla u_j^{n+1}\|^2 - \frac{1}{Re}\int_{\partial\Omega}(\vec{n}\cdot\nabla u_j^{n+1})\cdot u_j^{n+1}d\sigma \\
&\qquad + \int_{\partial\Omega}(\vec{n}\cdot u_j^{n+1})p_j^{n+1}d\sigma - \alpha h\int_{\partial\Omega}(\vec{n}\cdot(3\nabla u_j^{n+1} - 4\nabla u_j^n + \nabla u_j^{n-1}))\cdot u_j^{n+1}d\sigma \\
&+ \frac{\alpha}{2}h\Bigg(\|\nabla u_j^{n+1}\|^2 + \|2\nabla u_j^{n+1} - \nabla u_j^n\|^2 - \|\nabla u_j^n\|^2 - \|2\nabla u_j^n - \nabla u_j^{n-1}\|^2 + \|\nabla u_j^{n+1} - 2\nabla u_j^n + \nabla u_j^{n-1}\|^2\Bigg) \\
&\qquad = Ri\cdot((2\theta_j^n - \theta_j^{n-1})\cdot\begin{pmatrix}0\\1\end{pmatrix}, u_j^{n+1}) + (f_j^{n+1}, u_j^{n+1}),
\end{aligned}
\tag{3.16}
$$

and

$$
\begin{aligned}
&\frac{1}{4\Delta t}\left(\|\theta_j^{n+1}\|^2 + \|2\theta_j^{n+1} - \theta_j^n\|^2\right) - \frac{1}{4\Delta t}\left(\|\theta_j^n\|^2 + \|2\theta_j^n - \theta_j^{n-1}\|^2\right) + \frac{1}{4\Delta t}\|\theta_j^{n+1} - 2\theta_j^n + \theta_j^{n-1}\|^2 \\
&+ \frac{1}{RePr}\|\nabla\theta_j^{n+1}\|^2 + \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}((2u_j^n - u_j^{n-1})\cdot\nabla(2\theta_j^n - \theta_j^{n-1}), \theta_j^{n+1}) - \frac{1}{RePr}\int_{\partial\Omega}(\vec{n}\cdot\nabla\theta_j^{n+1})\cdot\theta_j^{n+1}d\sigma \\
&\quad - \beta h\int_{\partial\Omega}(\vec{n}\cdot(\nabla 3\theta_j^{n+1} - 4\nabla\theta_j^n + \nabla\theta_j^{n-1}))\cdot\theta_j^{n+1}d\sigma + \frac{\beta}{2}h\Bigg(\|\nabla\theta_j^{n+1}\|^2 + \|\nabla(2\theta_j^{n+1} - \theta_j^n)\|^2 \\
&\qquad - \|\nabla\theta_j^n\|^2 - \|\nabla(2\theta_j^n - \theta_j^{n-1})\|^2 + \|\nabla(\theta_j^{n+1} - 2\theta_j^n + \theta_j^{n-1})\|^2\Bigg) = (g_j^{n+1}, \theta_j^{n+1}).
\end{aligned}
\tag{3.17}
$$

The terms on the right hand side can be bounded as follows.

$$
Ri\cdot(2\theta_j^n - \theta_j^{n-1}\cdot\begin{pmatrix}0\\1\end{pmatrix}, u_j^{n+1}) \leq C_P Ri\|2\theta_j^n - \theta_j^{n-1}\|\|\nabla u_j^{n+1}\| \tag{3.18}
$$

$$
\leq \frac{1}{4Re}\|\nabla u_j^{n+1}\|^2 + C_P^2 Ri^2 Re\|2\theta_j^n - \theta_j^{n-1}\|^2 \leq \frac{1}{4Re}\|\nabla u_j^{n+1}\|^2 + 6C_P^2 Ri^2 Re\|\theta_j^n\|^2 + 3C_P^2 Ri^2 Re\|\theta_j^{n-1}\|^2,
$$

$$
(f_j^{n+1}, u_j^{n+1}) \leq C_P\|f_j^{n+1}\|\|\nabla u_j^{n+1}\| \leq \frac{1}{4Re}\|\nabla u_j^{n+1}\|^2 + C_P^2 Re\|f_j^{n+1}\|^2, \tag{3.19}
$$

$$
(g_j^{n+1}, \theta_j^{n+1}) \leq C_P\|g_j^{n+1}\|\|\nabla\theta_j^{n+1}\| \leq \frac{1}{2RePr}\|\nabla\theta_j^{n+1}\|^2 + \frac{1}{2}C_P^2 RePr\|g_j^{n+1}\|^2. \tag{3.20}
$$

Multiplying (2.13) with q_j^{n+1} and (2.15) with r_j^{n+1} gives

$$
\frac{1}{4\Delta t}\left(|q_j^{n+1}|^2 + |2q_j^{n+1} - q_j^n|^2 - |q_j^n|^2 - |2q_j^n - q_j^{n-1}|^2 + |q_j^{n+1} - 2q_j^n + q_j^{n-1}|^2\right) \tag{3.21}
$$

$$= -\frac{1}{T}|q_j^{n+1}|^2 + \frac{q_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}((2u_j^n - u_j^{n-1}) \cdot \nabla(2u_j^n - u_j^{n-1}), u_j^{n+1}) - c_j^{n+1},$$

and

$$\begin{aligned} &\frac{1}{4\Delta t}\left(|r_j^{n+1}|^2 + |2r_j^{n+1} - r_j^n|^2 - |r_j^n|^2 - |2r_j^n - r_j^{n-1}|^2 + |r_j^{n+1} - 2r_j^n + r_j^{n-1}|^2\right) \\ &= -\frac{1}{T}|r_j^{n+1}|^2 + \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}((2u_j^n - u_j^{n-1}) \cdot \nabla(2\theta_j^n - \theta_j^{n-1}), \theta_j^{n+1}) - d_j^{n+1}. \end{aligned} \tag{3.22}$$

Assuming homogeneous Dirichlet boundary conditions, with the bounds (3.18)-(3.20), adding (3.16) and (3.21), (3.17) and (3.22) gives

$$\begin{aligned} &\frac{1}{4\Delta t}\left(\|u_j^{n+1}\|^2 + \|2u_j^{n+1} - u_j^n\|^2\right) - \frac{1}{4\Delta t}\left(\|u_j^n\|^2 + \|2u_j^n - u_j^{n-1}\|^2\right) + \frac{1}{4\Delta t}\|u_j^{n+1} - 2u_j^n + u_j^{n-1}\|^2 \\ &+ \frac{1}{4\Delta t}\left(|q_j^{n+1}|^2 + |2q_j^{n+1} - q_j^n|^2 - |q_j^n|^2 - |2q_j^n - q_j^{n-1}|^2 + |q_j^{n+1} - 2q_j^n + q_j^{n-1}|^2\right) + \frac{1}{2Re}\|\nabla u_j^{n+1}\|^2 \\ &+ \frac{\alpha}{2}h\Bigg(\|\nabla u_j^{n+1}\|^2 + \|2\nabla u_j^{n+1} - \nabla u_j^n\|^2 - \|\nabla u_j^n\|^2 - \|2\nabla u_j^n - \nabla u_j^{n-1}\|^2 + \|\nabla u_j^{n+1} - 2\nabla u_j^n + \nabla u_j^{n-1}\|^2\Bigg) \\ &+ \frac{1}{T}|q_j^{n+1}|^2 \leq 6C_P^2 Ri^2 Re\|\theta_j^n\|^2 + 3C_P^2 Ri^2 Re\|\theta_j^{n-1}\|^2 + C_P^2 Re\|f_j^{n+1}\|^2 \ , \end{aligned} \tag{3.23}$$

and

$$\begin{aligned} &\frac{1}{4\Delta t}\left(\|\theta_j^{n+1}\|^2 + \|2\theta_j^{n+1} - \theta_j^n\|^2\right) - \frac{1}{4\Delta t}\left(\|\theta_j^n\|^2 + \|2\theta_j^n - \theta_j^{n-1}\|^2\right) + \frac{1}{4\Delta t}\|\theta_j^{n+1} - 2\theta_j^n + \theta_j^{n-1}\|^2 \\ &+ \frac{1}{4\Delta t}\left(|r_j^{n+1}|^2 + |2r_j^{n+1} - r_j^n|^2 - |r_j^n|^2 - |2r_j^n - r_j^{n-1}|^2 + |r_j^{n+1} - 2r_j^n + r_j^{n-1}|^2\right) + \frac{1}{2RePr}\|\nabla\theta_j^{n+1}\|^2 \\ &+ \frac{\beta}{2}h\Bigg(\|\nabla\theta_j^{n+1}\|^2 + \|\nabla(2\theta_j^{n+1} - \theta_j^n)\|^2 - \|\nabla\theta_j^n\|^2 - \|\nabla(2\theta_j^n - \theta_j^{n-1})\|^2 + \|\nabla(\theta_j^{n+1} - 2\theta_j^n + \theta_j^{n-1})\|^2\Bigg) \\ &+ \frac{1}{T}|r_j^{n+1}|^2 \leq \frac{1}{2}C_P^2 RePr\|g_j^{n+1}\|^2 \ . \end{aligned} \tag{3.24}$$

Summing up (3.24) from $n = 1$ to $n = N - 1$ and multiplying through by $4\Delta t$ gives

$$\begin{aligned} &\|\theta_j^N\|^2 + \|2\theta_j^N - \theta_j^{N-1}\|^2 + \sum_{n=1}^{N-1}\|\theta_j^{n+1} - 2\theta_j^n + \theta_j^{n-1}\|^2 + |r_j^N|^2 + |2r_j^N - r_j^{N-1}|^2 \\ &+ \sum_{n=1}^{N-1}\|r_j^{n+1} - 2r_j^n + r_j^{n-1}\|^2 + \Delta t\sum_{n=1}^{N-1}\frac{2}{RePr}\|\nabla\theta_j^{n+1}\|^2 + \Delta t\sum_{n=1}^{N-1}\frac{4}{T}|r_j^{n+1}|^2 \\ &\quad + 2\beta h\Delta t\|\nabla\theta_j^N\|^2 + 2\beta h\Delta t\|2\nabla\theta_j^N - \nabla\theta_j^{N-1}\|^2 + \Delta t\sum_{n=1}^{N-1}2\beta h\|\nabla\theta_j^{n+1} - 2\nabla\theta_j^n + \nabla\theta_j^{n-1}\|^2 \\ &\leq \|\theta_j^1\|^2 + \|2\theta_j^1 - \theta_j^0\|^2 + |r_j^1|^2 + |2r_j^1 - r_j^0|^2 + 2\beta h\Delta t\|\nabla\theta_j^1\|^2 + 2\beta h\Delta t\|\nabla 2\theta_j^1 - \theta_j^0\|^2 \\ &\quad + \Delta t\sum_{n=1}^{N-1}2C_P^2 RePr\|g_j^{n+1}\|^2 \ . \end{aligned} \tag{3.25}$$

Summing up (3.23) from $n = 1$ to $n = N - 1$, multiplying through by $4\Delta t$, and using the stability result for the temperature from (3.25) yields

$$\|u_j^N\|^2 + \|2u_j^N - u_j^{N-1}\|^2 + \sum_{n=1}^{N-1}\|u_j^{n+1} - 2u_j^n + u_j^{n-1}\|^2 + |q_j^N|^2 + |2q_j^N - q_j^{N-1}|^2 \tag{3.26}$$

$$
\begin{aligned}
&+\sum_{n=1}^{N-1}\|q_j^{n+1}-2q_j^n+q_j^{n-1}\|^2+\Delta t\sum_{n=1}^{N-1}\frac{2}{Re}\|\nabla u_j^{n+1}\|^2+\Delta t\sum_{n=2}^{N-1}\frac{4}{T}|q_j^{n+1}|^2\\
&+2\alpha h\Delta t\|\nabla u_j^N\|^2+2\alpha h\Delta t\|2\nabla u_j^N-\nabla u_j^{N-1}\|^2+\Delta t\sum_{n=1}^{N-1}2\alpha h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2\\
&\le \|u_j^1\|^2+\|2u_j^1-u_j^0\|^2+|q_j^1|^2+|2q_j^1-q_j^0|^2+2\alpha h\Delta t\|\nabla u_j^1\|^2+2\alpha h\Delta t\|2\nabla u_j^1-\nabla u_j^0\|^2\\
&\quad+\Delta t\sum_{n=1}^{N-1}\left(24C_P^2Ri^2Re\|\theta_j^n\|^2+12C_P^2Ri^2Re\|\theta_j^{n-1}\|^2\right)+\Delta t\sum_{n=1}^{N-1}4C_P^2Re\|f_j^{n+1}\|^2\\
&\le \|u_j^1\|^2+\|2u_j^1-u_j^0\|^2+|q_j^1|^2+|2q_j^1-q_j^0|^2+2\alpha h\Delta t\|\nabla u_j^1\|^2+2\alpha h\Delta t\|2\nabla u_j^1-\nabla u_j^0\|^2\\
&\quad+\sum_{n=2}^{N-1}36C_P^2Ri^2Re\Delta t\|\theta_j^n\|^2+36C_P^2Ri^2Re\Delta t\|\theta_j^1\|^2+12C_P^2Ri^2Re\Delta t\|\theta_j^0\|^2+\Delta t\sum_{n=1}^{N-1}4C_P^2Re\|f_j^{n+1}\|^2\\
&\le \|u_j^1\|^2+\|2u_j^1-u_j^0\|^2+|q_j^1|^2+|2q_j^1-q_j^0|^2+2\alpha h\Delta t\|\nabla u_j^1\|^2+2\alpha h\Delta t\|2\nabla u_j^1-\nabla u_j^0\|^2\\
&\quad+\sum_{n=2}^{N-1}36C_P^2Ri^2Re\Delta t\Bigg(\|\theta_j^1\|^2+\|2\theta_j^1-\theta_j^0\|^2+|r_j^1|^2+|2r_j^1-r_j^0|^2+2\beta h\Delta t\|\nabla\theta_j^1\|^2\\
&\quad+2\beta h\Delta t\|\nabla 2\theta_j^1-\theta_j^0\|^2+\Delta t\sum_{k=1}^{n-1}2C_P^2RePr\|g_j^{k+1}\|^2\Bigg)+36C_P^2Ri^2Re\Delta t\|\theta_j^1\|^2+12C_P^2Ri^2Re\Delta t\|\theta_j^0\|^2\\
&\quad+\Delta t\sum_{n=1}^{N-1}4C_P^2Re\|f_j^{n+1}\|^2.
\end{aligned}
$$

□

4. Implementation Algorithms. In this section we present the fully decoupled implementation algorithms for Algorithm 2.1 and Algorithm 2.2 respectively, and it will be easier to see the efficiency of the ensemble schemes from here.

We first derive the implementation algorithm for the Stab-SAV-CNLF-En scheme. Let

$$
S_j^{n+1}=\frac{q_j^{n+1}+q_j^{n-1}}{2exp(-\frac{t^n}{T})},\qquad u_j^{n+1}=\hat{u}_j^{n+1}+S_j^{n+1}\breve{u}_j^{n+1},\qquad p_j^{n+1}=\hat{p}_j^{n+1}+S_j^{n+1}\breve{p}_j^{n+1},\tag{4.1}
$$

$$
V_j^{n+1}=\frac{r_j^{n+1}+r_j^{n-1}}{2exp(-\frac{t^n}{T})},\qquad \theta_j^{n+1}=\hat{\theta}_j^{n+1}+V_j^{n+1}\breve{\theta}_j^{n+1}.\tag{4.2}
$$

Then instead of solving (2.6)-(2.10), we solve the following two subproblems for $(\hat{u}_j^{n+1},\hat{p}_j^{n+1},\hat{\theta}_j^{n+1})$, $(\breve{u}_j^{n+1},\breve{p}_j^{n+1},\breve{\theta}_j^{n+1})$ respectively.

$$
\begin{cases}
\dfrac{1}{2\Delta t}\hat{u}_j^{n+1}-\dfrac{1}{2Re}\Delta\hat{u}_j^{n+1}-\alpha h\Delta\hat{u}_j^{n+1}+\dfrac{1}{2}\nabla\hat{p}_j^{n+1}\\
\qquad\qquad=f_j^n+Ri\cdot\theta_j^n\cdot\begin{pmatrix}0\\1\end{pmatrix}+\dfrac{1}{2\Delta t}u_j^{n-1}+\dfrac{1}{2Re}\Delta u_j^{n-1}-\alpha h\Delta u_j^{n-1}-\dfrac{1}{2}\nabla p_j^{n-1},\ \text{in }\Omega\\
\nabla\cdot\hat{u}_j^{n+1}=0,\ \text{in }\Omega\\
\hat{u}_j^{n+1}=a_j^{n+1},\ \text{on }\partial\Omega.
\end{cases}\tag{CNLF sub-problem 1}
$$

$$
\begin{cases}
\dfrac{1}{2\Delta t}\breve{u}_j^{n+1}-\dfrac{1}{2Re}\Delta\breve{u}_j^{n+1}-\alpha h\Delta\breve{u}_j^{n+1}+\dfrac{1}{2}\nabla\breve{p}_j^{n+1}=-(u_j^n\cdot\nabla)u_j^n,\ \text{in }\Omega\\
\nabla\cdot\breve{u}_j^{n+1}=0,\ \text{in }\Omega\\
\breve{u}_j^{n+1}=0,\ \text{on }\partial\Omega.
\end{cases}\tag{CNLF sub-problem 2}
$$

$$\begin{cases} \frac{1}{2\Delta t}\hat{\theta}_j^{n+1} - \frac{1}{2RePr}\Delta\hat{\theta}_j^{n+1} - \beta h\Delta\hat{\theta}_j^{n+1} \\ \qquad\qquad = g_j^n + \frac{1}{2\Delta t}\theta_j^{n-1} + \frac{1}{2RePr}\Delta\theta_j^{n-1} - \beta h\Delta\theta_j^{n-1} \text{ in } \Omega \\ \hat{\theta}_j^{n+1} = b_j^{n+1}, \text{ on } \partial\Omega. \end{cases} \tag{CNLF sub-problem 3}$$

$$\begin{cases} \frac{1}{2\Delta t}\breve{\theta}_j^{n+1} - \frac{1}{2RePr}\Delta\breve{\theta}_j^{n+1} - \beta h\Delta\breve{\theta}_j^{n+1} = -(u_j^n\cdot\nabla)\theta_j^n, \text{ in } \Omega \\ \breve{\theta}_j^{n+1} = 0, \text{ on } \partial\Omega. \end{cases} \tag{CNLF sub-problem 4}$$

Now we need to derive an equation for S_j^{n+1}.

$$S_j^{n+1} = \frac{q_j^{n+1} + q_j^{n-1}}{2exp(-\frac{t^n}{T})} \qquad \Longrightarrow \qquad q_j^{n+1} = 2exp(-\frac{t^n}{T})S_j^{n+1} - q_j^{n-1}. \tag{4.3}$$

Plugging this expression of q_j^{n+1} into (3.8) gives

$$\begin{aligned} &\frac{1}{4\Delta t}\left(4exp(-\frac{2t^n}{T})(S_j^{n+1})^2 - 4q_j^{n-1}exp(-\frac{t^n}{T})S_j^{n+1}\right) \\ &\qquad = -\frac{1}{T}exp(-\frac{2t^n}{T})(S_j^{n+1})^2 + S_j^{n+1}\int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot\frac{\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1} + u_j^{n-1}}{2})dx - c_j^n. \end{aligned} \tag{4.4}$$

We then obtain the equation for S_j^{n+1} as

$$A_j^{n+1}(S_j^{n+1})^2 + B_j^{n+1}S_j^{n+1} + C_j^{n+1} = 0, \tag{CNLF sub-problem 5}$$

where

$$\begin{aligned} A_j^{n+1} &= (\frac{1}{\Delta t} + \frac{1}{T})exp(-\frac{2t^n}{T}) - \frac{1}{2}\int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot\breve{u}_j^{n+1}dx, \\ B_j^{n+1} &= -\frac{q_j^{n-1}}{\Delta t}exp(-\frac{t^n}{T}) - \frac{1}{2}\int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot(\hat{u}_j^{n+1} + u_j^{n-1})dx, \\ C_j^{n+1} &= c_j^n. \end{aligned}$$

This is a scalar quadratic equation with two roots. By the definition of S_j^{n+1}, we should pick the root that is close to 1. In solving sub-problems 1, 2, 3 and 4, all realizations share a constant coefficient matrix that is both time independent and ensemble index independent so that block solvers, such as block CG, block GMRES, can be used to solve the corresponding linear systems fast and efficiently. Sub-problem 5 can be solved quickly since it is a scalar quadratic equations.

After getting $\hat{u}_j^{n+1}, \breve{u}_j^{n+1}$, and S_j^{n+1}, we have $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}$. Similarly, we can derive an equation for V_j^{n+1}.

$$\tilde{A}_j^{n+1}(V_j^{n+1})^2 + \tilde{B}_j^{n+1}V_j^{n+1} + \tilde{C}_j^{n+1} = 0, \tag{CNLF sub-problem 6}$$

where

$$\begin{aligned} \tilde{A}_j^{n+1} &= (\frac{1}{\Delta t} + \frac{1}{T})exp(-\frac{2t^n}{T}) - \frac{1}{2}\int_\Omega (u_j^n\cdot\nabla\theta_j^n)\breve{\theta}_j^{n+1}dx, \\ \tilde{B}_j^{n+1} &= -\frac{r_j^{n-1}}{\Delta t}exp(-\frac{t^n}{T}) - \frac{1}{2}\int_\Omega (u_j^n\cdot\nabla\theta_j^n)(\hat{\theta}_j^{n+1} + \theta_j^{n-1})dx, \end{aligned}$$

$$\tilde{C}_j^{n+1} = d_j^n.$$

THEOREM 4.1. *With homogeneous Dirichlet boundary conditions, there exists a unique solution, which is real, for CNLF sub-problem 5 and 6 respectively. Thus q_j^n and r_j^n are guaranteed to be real for any $n = 0, \cdots, N$, $j = 1, \cdots, J$.*

Proof. With homogeneous Dirichlet boundary conditions, we have $C_j^{n+1} = 0$. So CNLF sub-problem 5 has a unique real solution if $A_j^{n+1} \neq 0$. Testing CNLF sub-problem 2 with $\breve{u}_j^{n+1}$ we have

$$\frac{1}{2\Delta t}\|\breve{u}_j^{n+1}\|^2 + (\frac{1}{2Re} + \alpha h)\|\nabla \breve{u}_j^{n+1}\|^2 = -\int_\Omega (u_j^n \cdot \nabla) u_j^n \cdot \breve{u}_j^{n+1} dx.$$

Thus

$$A_j^{n+1} = (\frac{1}{\Delta t} + \frac{1}{T}) exp(-\frac{2t^n}{T}) + \frac{1}{4\Delta t}\|\breve{u}_j^{n+1}\|^2 + (\frac{1}{4Re} + \frac{1}{2}\alpha h)\|\nabla \breve{u}_j^{n+1}\|^2 > 0.$$

Similarly, we have

$$\tilde{A}_j^{n+1} = (\frac{1}{\Delta t} + \frac{1}{T}) exp(-\frac{2t^n}{T}) + \frac{1}{4\Delta t}\|\breve{\theta}_j^{n+1}\|^2 + (\frac{1}{4RePr} + \frac{1}{2}\beta h)\|\nabla \breve{\theta}_j^{n+1}\|^2 > 0.$$

Since $\tilde{C}_j^{n+1} = 0$, CNLF sub-problem 6 also has a unique solution and it is real.

By the expressions of q_j^{n+1} (4.3), and $r_j^{n+1} = 2exp(-\frac{t^n}{T})V_j^{n+1} - r_j^{n-1}$, we can conclude by indution that q_j^n and r_j^n are real for any $n = 0, \cdots, N$, $j = 1, \cdots, J$. □

We next present the implementation algorithm for the Stab-SAV-BDF2-En scheme. Let

$$S_j^{n+1} = \frac{q_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}, \qquad u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}, \qquad p_j^{n+1} = \hat{p}_j^{n+1} + S_j^{n+1}\breve{p}_j^{n+1}, \tag{4.5}$$

$$V_j^{n+1} = \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}, \qquad \theta_j^{n+1} = \hat{\theta}_j^{n+1} + V_j^{n+1}\breve{\theta}_j^{n+1}. \tag{4.6}$$

Then instead of solving (2.11)-(2.15), we solve the following two subproblems for $(\hat{u}_j^{n+1}, \hat{p}_j^{n+1}, \hat{\theta}_j^{n+1})$, $(\breve{u}_j^{n+1}, \breve{p}_j^{n+1}, \breve{\theta}_j^{n+1})$ respectively.

$$\begin{cases} \frac{3}{2\Delta t}\hat{u}_j^{n+1} - \frac{1}{Re}\Delta \hat{u}_j^{n+1} - 3\alpha h \Delta \hat{u}_j^{n+1} + \nabla \hat{p}_j^{n+1} \\ \qquad\qquad = f_j^{n+1} + Ri \cdot (2\theta_j^n - \theta_j^{n-1}) \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \frac{2}{\Delta t}u_j^n - \frac{1}{2\Delta t}u_j^{n-1} - 4\alpha h \Delta u_j^n + \alpha h \Delta u_j^{n-1}, \text{ in } \Omega \\ \nabla \cdot \hat{u}_j^{n+1} = 0, \text{ in } \Omega \\ \hat{u}_j^{n+1} = a_j^{n+1}, \text{ on } \partial\Omega. \end{cases} \tag{BDF2 sub-problem 1}$$

$$\begin{cases} \frac{3}{2\Delta t}\breve{u}_j^{n+1} - \frac{1}{Re}\Delta \breve{u}_j^{n+1} - 3\alpha h \Delta \breve{u}_j^{n+1} + \nabla \breve{p}_j^{n+1} \\ \qquad\qquad = -((2u_j^n - u_j^{n-1}) \cdot \nabla)(2u_j^n - u_j^{n-1}), \text{ in } \Omega \\ \nabla \cdot \breve{u}_j^{n+1} = 0, \text{ in } \Omega \\ \breve{u}_j^{n+1} = 0, \text{ on } \partial\Omega. \end{cases} \tag{BDF2 sub-problem 2}$$

$$\begin{cases} \frac{3}{2\Delta t}\hat{\theta}_j^{n+1} - \frac{1}{RePr}\Delta \hat{\theta}_j^{n+1} - 3\beta h \Delta \hat{\theta}_j^{n+1} \\ \qquad\qquad = g_j^{n+1} + \frac{2}{\Delta t}\theta_j^n - \frac{1}{2\Delta t}\theta_j^{n-1} - 4\beta h \Delta \theta_j^n + \beta h \Delta \theta_j^{n-1}, \text{ in } \Omega \\ \hat{\theta}_j^{n+1} = b_j^{n+1}, \text{ on } \partial\Omega. \end{cases} \tag{BDF2 sub-problem 3}$$

$$\begin{cases} \frac{3}{2\Delta t}\breve{\theta}_j^{n+1} - \frac{1}{RePr}\Delta\breve{\theta}_j^{n+1} - 3\beta h\Delta\breve{\theta}_j^{n+1} \\ \qquad\qquad\qquad = -((2u_j^n - u_j^{n-1})\cdot\nabla)(2\theta_j^n - \theta_j^{n-1}), \text{ in } \Omega \\ \breve{\theta}_j^{n+1} = 0, \text{ on } \partial\Omega. \end{cases} \tag{BDF2 sub-problem 4}$$

Now we need to derive an equation for S_j^{n+1}.

$$S_j^{n+1} = \frac{q_j^{n+1}}{exp(-\frac{t^{n+1}}{T})} \qquad \Longrightarrow \qquad q_j^{n+1} = exp(-\frac{t^{n+1}}{T})\, S_j^{n+1}. \tag{4.7}$$

Multiplying (2.13) by q_j^{n+1} gives

$$(\frac{3}{2\Delta t} + \frac{1}{T})(q_j^{n+1})^2 + \frac{-4q_j^n + q_j^{n-1}}{2\Delta t} q_j^{n+1} - S_j^{n+1}\int_\Omega ((2u_j^n - u_j^{n-1})\cdot\nabla)(2u_j^n - u_j^{n-1})\cdot u_j^{n+1}\, dx + c_j^{n+1} = 0$$
$$\Longrightarrow (\frac{3}{2\Delta t} + \frac{1}{T})exp(-\frac{2t^{n+1}}{T})\,(S_j^{n+1})^2 + \frac{-4q_j^n + q_j^{n-1}}{2\Delta t} exp(-\frac{t^{n+1}}{T})\, S_j^{n+1}$$
$$- S_j^{n+1}\int_\Omega ((2u_j^n - u_j^{n-1})\cdot\nabla)(2u_j^n - u_j^{n-1})\cdot(\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1})\, dx + c_j^{n+1} = 0.$$

At last, we obtain the equation for S_j^{n+1} as

$$A_j^{n+1}(S_j^{n+1})^2 + B_j^{n+1}S_j^{n+1} + C_j^{n+1} = 0, \tag{BDF2 sub-problem 5}$$

where

$$\begin{aligned} A_j^{n+1} &= (\frac{3}{2\Delta t} + \frac{1}{T})exp(-\frac{2t^{n+1}}{T}) - \int_\Omega ((2u_j^n - u_j^{n-1})\cdot\nabla)(2u_j^n - u_j^{n-1})\cdot\breve{u}_j^{n+1}\, dx, \\ B_j^{n+1} &= \frac{-4q_j^n + q_j^{n-1}}{2\Delta t} exp(-\frac{t^{n+1}}{T}) - \int_\Omega ((2u_j^n - u_j^{n-1})\cdot\nabla)(2u_j^n - u_j^{n-1})\cdot\hat{u}_j^{n+1}\, dx, \\ C_j^{n+1} &= c_j^{n+1}. \end{aligned}$$

Similarly, we can derive an equation for V_j^{n+1}.

$$\tilde{A}_j^{n+1}(V_j^{n+1})^2 + \tilde{B}_j^{n+1}V_j^{n+1} + \tilde{C}_j^{n+1} = 0, \tag{BDF2 sub-problem 6}$$

where

$$\begin{aligned} \tilde{A}_j^{n+1} &= (\frac{3}{2\Delta t} + \frac{1}{T})exp(-\frac{2t^{n+1}}{T}) - \int_\Omega ((2u_j^n - u_j^{n-1})\cdot\nabla)(2\theta_j^n - \theta_j^{n-1})\cdot\breve{\theta}_j^{n+1}\, dx, \\ \tilde{B}_j^{n+1} &= \frac{-4r_j^n + r_j^{n-1}}{2\Delta t} exp(-\frac{t^{n+1}}{T}) - \int_\Omega ((2u_j^n - u_j^{n-1})\cdot\nabla)(2\theta_j^n - \theta_j^{n-1})\cdot\hat{\theta}_j^{n+1}\, dx, \\ \tilde{C}_j^{n+1} &= d_j^{n+1}. \end{aligned}$$

We can also prove

Theorem 4.2. *With homogeneous Dirichlet boundary conditions, there exists a unique solution, which is real, for BDF2 sub-problem 5 and 6 respectively. Thus q_j^n and r_j^n are guaranteed to be real for any $n = 0, \cdots, N$, $j = 1, \cdots, J$.*

5. Algebraic Systems. Let $\{\chi_j^{uu}\}_{j=1}^{N_u}$, $\{\chi_j^{p}\}_{j=1}^{N_p}$, and $\{\chi_j^{\theta}\}_{j=1}^{N_\theta}$ be the basis functions of finite element spaces for approximating the velocity u, pressure p, and temperature θ, respectively. The approximated solutions will be represented by vectors of nodal values, denoted in bold. When a superscript n is applied to a bold vector, it represents the value at time $t_n = n\Delta t$, and a subscript j will be applied to represent the solution for the j-th sample. Let $\mathbf{M}_{uu}$ and $\mathbf{S}_{uu}$ denote the velocity mass matrice and velocity stiffness matrice respectively. Similarly $\mathbf{M}_\theta$ and $\mathbf{S}_\theta$ for temperature. We also define matrices $\mathbf{D}_{up}$, $\mathbf{N}_{uu}(u)$, $\mathbf{C}_{u\theta}$, and $\mathbf{N}_{u\theta}(u)$ whose entries are given as follows

$$[\mathbf{D}_{up}]_{kl} = \int_\Omega \chi_l^p (\nabla \cdot \chi_k^{uu}), \quad [\mathbf{N}_{uu}(u)]_{kl} = \int_\Omega (u \cdot \nabla)\chi_l^{uu} \cdot \chi_k^{uu},$$
$$[\mathbf{C}_{u\theta}]_{kl} = \int_\Omega \chi_l^\theta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \chi_k^{uu}, \quad [\mathbf{N}_{u\theta}(u)]_{kl} = \int_\Omega (u \cdot \nabla \chi_l^\theta)\chi_k^\theta.$$

The proposed schemes will be compared to the standard BDF2 scheme:

Algorithm 5.1 (BDF2).

$$\begin{cases} \dfrac{3u_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t} + ((2u_j^n - u_j^{n-1}) \cdot \nabla) u_j^{n+1} - \dfrac{1}{Re}\Delta u_j^{n+1} + \nabla p_j^{n+1} = Ri(2\theta_j^n - \theta_j^{n-1}) \begin{pmatrix} 0 \\ 1 \end{pmatrix} + f_j^{n+1}, \\ \nabla \cdot u_j^{n+1} = 0, \\ \dfrac{3\theta_j^{n+1} - 4\theta_j^n + \theta_j^{n-1}}{2\Delta t} + (2u_j^n - u_j^{n-1}) \cdot \nabla \theta_j^{n+1} - \dfrac{1}{RePr}\Delta \theta_j^{n+1} = g_j^{n+1} \end{cases}$$

Below we list the forms of algebraic systems of the Stab-SAV-BDF2-En scheme and the BDF2 scheme, for sample $j = 1, \cdots, J$.

- Stab-SAV-BDF2-En:

$$\mathbf{A}_{\text{savbdf2}} \begin{pmatrix} \hat{\mathbf{u}}_j^{n+1} \\ \hat{\mathbf{p}}_j^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{b}}_j^{n+1} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{A}_{\text{savbdf2}} \begin{pmatrix} \breve{\mathbf{u}}_j^{n+1} \\ \breve{\mathbf{p}}_j^{n+1} \end{pmatrix} = \begin{pmatrix} \breve{\mathbf{b}}_j^{n+1} \\ \mathbf{0} \end{pmatrix}, \tag{5.1}$$

$$\mathbf{B}_{\text{savbdf2}} \hat{\boldsymbol{\theta}}_j^{n+1} = \hat{\mathbf{c}}_j^{n+1}, \quad \mathbf{B}_{\text{savbdf2}} \breve{\boldsymbol{\theta}}_j^{n+1} = \breve{\mathbf{c}}_j^{n+1} \tag{5.2}$$

with

$$\mathbf{A}_{\text{savbdf2}} = \begin{pmatrix} \frac{3}{2\Delta t}\mathbf{M}_{uu} + (\frac{1}{Re} + 3\alpha h)\mathbf{S}_{uu} & -\mathbf{D}_{up} \\ -\mathbf{D}_{up}^T & \mathbf{0} \end{pmatrix},$$
$$\mathbf{B}_{\text{savbdf2}} = \frac{3}{2\Delta t}\mathbf{M}_\theta + (\frac{1}{RePr} + 3\beta h)\mathbf{S}_\theta,$$
$$\hat{\mathbf{b}}_j^{n+1} = \mathbf{f}_j^{n+1} + \mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n - \tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1}) + \mathbf{S}_{uu}(4\alpha h \mathbf{u}_j^n - \alpha h \mathbf{u}_j^{n-1}) + Ri\mathbf{C}_{u\theta}(2\boldsymbol{\theta}_j^n - \boldsymbol{\theta}_j^{n-1}),$$
$$\breve{\mathbf{b}}_j^{n+1} = -\mathbf{N}_{uu}(2u_j^n - u_j^{n-1}) \cdot (2\mathbf{u}_j^n - \mathbf{u}_j^{n-1}),$$
$$\hat{\mathbf{c}}_j^{n+1} = \mathbf{g}_j^{n+1} + \mathbf{M}_\theta(\tfrac{2}{\Delta t}\boldsymbol{\theta}_j^n - \tfrac{1}{2\Delta t}\boldsymbol{\theta}_j^{n-1}) + \mathbf{S}_\theta(4\beta h \boldsymbol{\theta}_j^n - \beta h \boldsymbol{\theta}_j^{n-1}),$$
$$\breve{\mathbf{c}}_j^{n+1} = -\mathbf{N}_{u\theta}(2u_j^n - u_j^{n-1}) \cdot (2\boldsymbol{\theta}_j^n - \boldsymbol{\theta}_j^{n-1}).$$

- BDF2:

$$\mathbf{A}_{\text{bdf2}}^{(n),j} \begin{pmatrix} \mathbf{u}_j^{n+1} \\ \mathbf{p}_j^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{h}_j^{n+1} \\ \mathbf{0} \end{pmatrix}, \tag{5.3}$$

$$\mathbf{B}_{\text{bdf2}}^{(n),j} \boldsymbol{\theta}_j^{n+1} = \mathbf{m}_j^{n+1} \tag{5.4}$$

with

$$\begin{aligned}
\mathbf{A}_{\rm bdf2}^{(n),j} &= \begin{pmatrix} \frac{3}{2\Delta t}\mathbf{M}_{uu} + \mathbf{N}_{uu}(2u_j^n - u_j^{n-1}) + \frac{1}{Re}\mathbf{S}_{uu} & -\mathbf{D}_{up} \\ -\mathbf{D}_{up}^T & \mathbf{0} \end{pmatrix}, \\
\mathbf{B}_{\rm bdf2}^{(n),j} &= \frac{3}{2\Delta t}\mathbf{M}_{\theta} + \mathbf{N}_{u\theta}(2u_j^n - u_j^{n-1}) + \frac{1}{RePr}\mathbf{S}_{\theta}, \\
\mathbf{h}_j^{n+1} &= \mathbf{f}_j^{n+1} + \mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n - \tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1}) + Ri\mathbf{C}_{u\theta}(2\boldsymbol{\theta}_j^n - \boldsymbol{\theta}_j^{n-1}), \\
\mathbf{m}_j^{n+1} &= \mathbf{g}_j^{n+1} + \mathbf{M}_{\theta}(\tfrac{2}{\Delta t}\boldsymbol{\theta}_j^n - \tfrac{1}{2\Delta t}\boldsymbol{\theta}_j^{n-1}).
\end{aligned}$$

Note that the matrix $\mathbf{A}_{\rm savbdf2}$ in the Stab-SAV-BDF2-En scheme is fixed at different time steps. If a direct linear solver is taken, then LU decomposition is needed only once. If an iterative linear solver is taken, then a fixed preconditioner can be applied in all time steps. The matrix $\mathbf{A}_{\rm savbdf2}$ is also fixed for different samples, so we can simultaneously solves a single linear system with multiple right hand sides corresponding to different samples. In contrast, the matrix $\mathbf{A}_{\rm bdf2}^{(n),j}$ in the BDF2 scheme changes over j and over time. We need to simulate J different samples one by one, and no common LU decomposition or preconditioner can be used. The advantage of getting matrix $\mathbf{B}_{\rm savbdf2}$ is similar. The Stab-SAV-BDF2-En scheme posses an extra superiority: its discretized linear systems are symmetric, whereas the linear systems for the BDF2 scheme are non-symmetric. To be specific, system (5.2) can be solved efficiently by a block conjugate gradient method, but a block GMRES method should be resorted to for solving (5.4).

6. Numerical Experiments. We will perform numerical experiments to demonstrate the effectiveness and efficiency of the proposed algorithms. To make the presentation short, we only focus on the Stab-SAV-BDF2-En scheme except for the convergence test. In all simulations, the finite element method is used for spatial discretization, and the finite element spaces are (P2, P1, P2) for the velocity/pressure/temperature, respectively. Both algorithms are implemented in Matlab with a data structure based on the iFEM matlab package.

6.1. Convergence test. In this section, we test the convergence rate of the Stab-SAV-CNLF-En and Stab-SAV-BDF2-En algorithms by computing the numerical error between the numerical solution and a given exact solution. Specifically, we consider the Boussinesq equations on $\Omega = (0,1)^2$ and set the model parameters as $Ri = 10, Pr = 1$. A manufactured analytical solution is constructed as in [15]:

$$\begin{aligned}
u(x_1,x_2,t) &= (\pi\sin(t)\sin(2\pi x_2)\sin^2(\pi x_1), -\pi\sin(t)\sin(2\pi x_1)\sin^2(\pi x_2))^T, \\
p(x_1,x_2,t) &= \sin(t)\cos(\pi x_1)\sin(\pi x_2), \\
\theta(x_1,x_2,t) &= \pi\sin(t)\sin(2\pi x_2)\sin^2(\pi x_1) - \pi\sin(t)\sin(2\pi x_1)\sin^2(\pi x_2).
\end{aligned}$$

We will consider an ensemble of $J = 2$ simulations, whose exact solutions are given by

$$\begin{aligned}
u_j(x_1,x_2,t) &= (1+\epsilon_j)u(x_1,x_2,t), \\
p_j(x_1,x_2,t) &= (1+\epsilon_j)p(x_1,x_2,t), \\
\theta_j(x_1,x_2,t) &= (1+\epsilon_j)\theta(x_1,x_2,t),
\end{aligned}$$

where $j = 1,2, \epsilon_1 = 0.01, \epsilon_2 = -0.01$. Dirichlet boundary conditions are imposed on $\partial\Omega$; the boundary conditions, initial conditions, forcing function and heat source are selected to match the prescribed analytical solution.

To examine the temporal convergence rate of each algorithm, we compute approximated solutions five times with four successive mesh refinements and timestep reductions. In particular, we test two typical cases. In the first test, we set $Re = 100$ so that the Reynolds number is relatively small and no stabilization is needed, i.e. $\alpha = 0, \beta = 0$. The initial mesh size is $h = 1/10$ and the relation $\Delta t = h$ is fixed. The solution errors at $T = 0.5$ for $j = 1$, computed by Stab-SAV-CNLF-En and Stab-SAV-BDF2-En, are reported in Table 6.1 and 6.2 respectively. These results confirm that the proposed algorithms are second order in time convergent.

Table 6.1: Errors at $T = 0.5$ and convergence rates of the Stab-SAV-CNLF-En algorithm $(J = 2, j = 1)$ with $\Delta t = h, \alpha = 0, \beta = 0, Re = 100$.

Δt	$\|u_{1,h} - u_1\|_{H^1}$	Rate	$\|p_{1,h} - p_1\|_{L^2}$	Rate	$\|\theta_{1,h} - \theta_1\|_{H^1}$	Rate
1/10	2.655×10^{-1}	-	3.042×10^{-2}	-	1.748×10^{-1}	-
1/20	4.482×10^{-2}	2.57	7.977×10^{-3}	1.93	3.628×10^{-2}	2.27
1/40	8.354×10^{-3}	2.42	2.022×10^{-3}	1.98	8.355×10^{-3}	2.12
1/80	1.827×10^{-3}	2.19	5.073×10^{-4}	1.99	2.026×10^{-3}	2.04
1/160	4.380×10^{-4}	2.06	1.270×10^{-4}	2.00	5.016×10^{-4}	2.01

Table 6.2: Errors at $T = 0.5$ and convergence rates of the Stab-SAV-BDF2-En algorithm $(J = 2, j = 1)$ with $\Delta t = h, \alpha = 0.0, \beta = 0.0, Re = 100$.

Δt	$\|u_{1,h} - u_1\|_{H^1}$	Rate	$\|p_{1,h} - p_1\|_{L^2}$	Rate	$\|\theta_{1,h} - \theta_1\|_{H^1}$	Rate
1/10	2.315×10^{-1}	-	1.512×10^{-2}	-	1.357×10^{-1}	-
1/20	3.942×10^{-2}	2.55	4.869×10^{-3}	1.63	2.472×10^{-2}	2.46
1/40	6.799×10^{-3}	2.54	1.341×10^{-3}	1.86	5.396×10^{-3}	2.20
1/80	1.419×10^{-3}	2.26	3.490×10^{-4}	1.94	1.316×10^{-3}	2.04
1/160	3.378×10^{-4}	2.07	8.894×10^{-4}	1.97	3.305×10^{-4}	1.99

In the second test, we take $Re = 1000$, which is relatively large and stabilization is necessary to guarantee convergence according to our experiments. In this case, we set $\alpha = 0.1, \beta = 0.1$, and fix $\Delta t = 0.2 * h$. The solution errors at $T = 0.5$ for $j = 1$, computed by Stab-SAV-CNLF-En and Stab-SAV-BDF2-En, are reported in Table 6.3 and 6.4 respectively. As one can see, both schemes have second order convergence rate as predicted.

6.2. Marsigli flow. Inspired by the numerical experiments in [30, 1], we check the effectiveness of the Stab-SAV-BDF2-En scheme on simulating the Marsigli flow, which has been known since the observation of Marsigli in 1681. The physical phenomenon is that when two fluids with different densities are separated by a vertical plan, once the partition is removed a motion driven by the gravity will create two currents with opposite moving directions. The flow with lighter density becomes a surface flow, while the other becomes the undercurrent. Because the fluid density difference can be converted into temperature difference with the reverse ratio by the Boussinesq assumption, this physical process can be modelled by the incompressible Boussinesq equations (2.1).

Following [30], we set the domain as $\Omega = (0, 8) \times (0, 1)$ to represent an insulated box with a partition located at $x_1 = 4$. Initially, the temperature is set to be 1.5 in the left half of the box, indicating the lower density flow, and 1 in the right half of the box, indicating the higher density flow. The temperature is subject to an adiabatic boundary condition. The Boussinesq flow is initially at rest, and subject to a no-slip boundary condition. The forcing function $f = 0$ and heat source $g = 0$ are used. The Richardson number and Prandtl number are set to be $Ri = 4, Pr = 1$.

To compare with results in [30], we set $Re = 5000$ and report the corresponding numerical solutions at $t = 2, 4, 6, 8$ in Fig. 6.1. Here the resolution is taken as $h = 1/64, \Delta t = 0.001$. The simulation shows almost idential temperature distributions and flow patterns as given in [30]. The appearance of a surface flow and undercurrent is observed clearly at the right time.

We would also like to show the importance of adding the stabilization term in the SAV approach. For this purpose, we firstly run the simulation with $\Delta t = 0.01, \alpha = 0, \beta = 0$ (i.e. no stabilization), and then compare the results with those obtained by choosing $\alpha = 0.5, \beta = 0.5$. Here we consider $Re = 1000$ which was used for study in [1]. The non-stabilized results are presented in Fig. 6.2, and the stabilized ones are given in Fig. 6.3. As one can see, the non-stabilized solution does not indicate the generation of

Table 6.3: Errors at $T = 0.5$ and convergence rates of the Stab-SAV-CNLF-En algorithm $(J = 2, j = 1)$ with $\Delta t = 0.2 * h, \alpha = 0.1, \beta = 0.1, Re = 1000$.

Δt	$\|u_{1,h} - u_1\|_{H^1}$	Rate	$\|p_{1,h} - p_1\|_{L^2}$	Rate	$\|\theta_{1,h} - \theta_1\|_{H^1}$	Rate
1/10	8.094×10^{-1}	-	1.290×10^{-1}	-	5.259×10^{-1}	-
1/20	2.202×10^{-1}	1.88	3.497×10^{-2}	1.88	1.542×10^{-1}	1.78
1/40	5.312×10^{-2}	2.05	8.944×10^{-3}	1.97	3.726×10^{-2}	2.05
1/80	1.165×10^{-2}	2.19	2.256×10^{-3}	1.99	8.549×10^{-3}	2.12
1/160	2.689×10^{-3}	2.12	5.679×10^{-4}	1.99	2.058×10^{-3}	2.05

Table 6.4: Errors at $T = 0.5$ and convergence rates of the Stab-SAV-BDF2-En algorithm $(J = 2, j = 1)$ with $\Delta t = 0.2 * h, \alpha = 0.1, \beta = 0.1, Re = 1000$.

Δt	$\|u_{1,h} - u_1\|_{H^1}$	Rate	$\|p_{1,h} - p_1\|_{L^2}$	Rate	$\|\theta_{1,h} - \theta_1\|_{H^1}$	Rate
1/10	7.668×10^{-1}	-	1.188×10^{-1}	-	4.967×10^{-1}	-
1/20	2.154×10^{-1}	1.83	3.369×10^{-2}	1.82	1.505×10^{-1}	1.72
1/40	5.251×10^{-2}	2.04	8.802×10^{-3}	1.94	3.680×10^{-2}	2.03
1/80	1.155×10^{-2}	2.18	2.241×10^{-3}	1.97	8.504×10^{-3}	2.11
1/160	2.675×10^{-3}	2.11	5.649×10^{-4}	1.99	2.057×10^{-3}	2.05

surface or undercurrent flow as time progresses. In contrast, the stabilized solution catches very well the temperture distribution and flow pattern reported in [1] at each time level. The generation of a surface flow and a undercurrent is clearly observed.

6.3. Efficiency tests. In this section we show the computational efficiency of Stab-SAV-BDF2-En by two numerical experiments, and compare its performance with the standard BDF2 scheme.

In the first experiment, we investigate the ensemble efficiency with the number J of samples varying from 1 to 100. Analytic solutions are given as in Sec. 6.1 with ϵ_j being random numbers uniformly distributed in $[-0.01, 0.01]$. The physical parameters are set to be $Pr = 1, Ri = 1, Re = 100$. For both the Stab-SAV-BDF2-En and BDF2 schemes, we take $h = 1/64, \Delta t = 0.002$ and run the simulation until $T = 0.5$. For accuracy comparison, we set $\alpha = 0, \beta = 0$ in the Stab-SAV-BDF2-En scheme. In this particular test, we use the block GMRES iterative linear solver [2] to solve for the velocity and pressure, and apply the least-squares commutator preconditioner to speed up convergence. Efficiency of this preconditioned iterative solver has been reported in [24]. We then use the block CG iterative linear solver for the temperature in the Stab-SAV-BDF2-En scheme, whereas block GRMRES for the temperature in the BDF2 scheme. Note that the latter involves non-symmetric linear systems.

Table 6.5 reports the comparison of CPU time and numerical errors computed by the Stab-SAV-BDF2-En and BDF2 schemes with $J = 1, 10, 100$. One can observe that the Stab-SAV-BDF2-En algorithm outperforms the BDF2 scheme as it takes less CPU times while having the same accuracy. The advantage of the Stab-SAV-BDF2-En algorithm becomes more apparent as the ensemble size increases.

The second experiment is to observe efficiency by simulating the differentially heated cavity flow, studied in [30, 15]. This is a realistic problem having many important industrial applications including solar collector, room ventilation, and cooling of electronics chips. Similar to [15], we simulate the differentially heated cavity flow in a rectangular domain $\Omega = (0, 1) \times (0, 4)$ with zero forcing function and heat source. A no-slip boundary condition is imposed for the velocity and adiabatic boundary condition is imposed for the temperature on the horizontal wall. The temperature on the left wall is fixed as constant $\theta = 0.5$, while on the right wall the temperature is always $\theta = -0.5$. In our simulation, the Prandtl number is chosen to be 0.71, the Richardson number is $Ri = 1$, and the Reynolds number is given by $Re = \sqrt{Ra/(PrRi)}$. We consider $J = 1$ and set initial conditions to be zero.

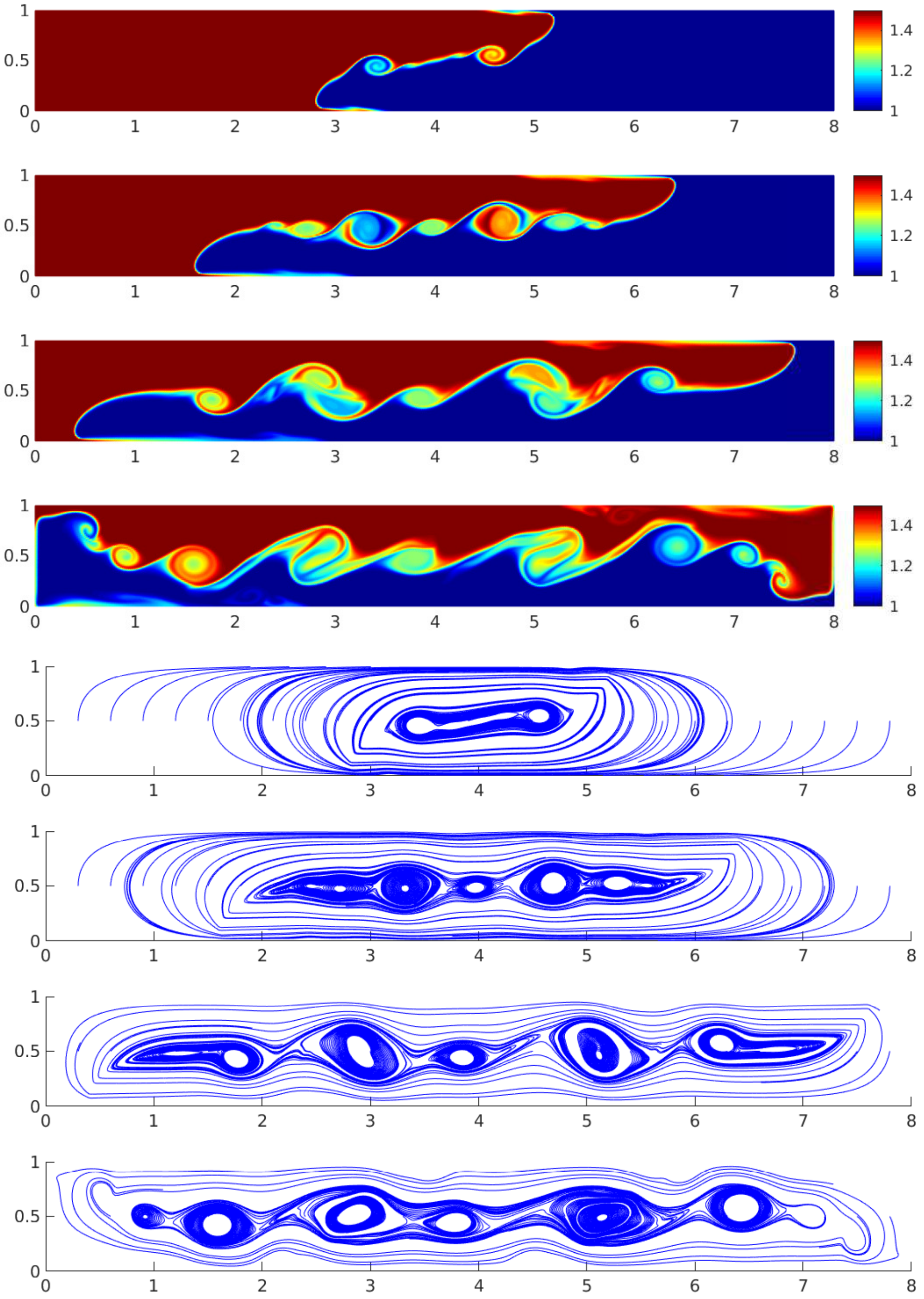

Fig. 6.1: Marsigli flow with $Re = 5000$ simulated by the Stab-SAV-BDF2-En scheme. From top to bottom: temperature at $t = 2, 4, 6, 8$, streamline at $t = 2, 4, 6, 8$.

For both the Stab-SAV-BDF2-En and BDF2 schemes, we take $h = 1/64, \Delta t = 0.002$ and run the simulation until $T = 10$. We still set $\alpha = 0, \beta = 0$ in the Stab-SAV-BDF2-En scheme. Simulations are performed by using both a direct linear solver (LU decomposition) and the preconditioned block GMRES

Table 6.5: CPU time and errors of the solution expectation at $T = 0.5$ with $h = 1/64, \Delta t = 0.002$ and different values of J.

	Stab-SAV-BDF2-En			BDF2		
	$J = 1$	$J = 10$	$J = 100$	$J = 1$	$J = 10$	$J = 100$
$\|\|\mathbb{E}[u_h - u]\|\|_{H^1}$	5.70×10^{-4}	5.68×10^{-4}	5.71×10^{-4}	5.70×10^{-4}	5.68×10^{-4}	5.71×10^{-4}
$\|\|\mathbb{E}[p_h - p]\|\|_{L^2}$	1.07×10^{-4}	1.07×10^{-4}	1.08×10^{-4}	1.07×10^{-4}	1.07×10^{-4}	1.08×10^{-4}
$\|\|\mathbb{E}[\theta_h - \theta]\|\|_{H^1}$	2.89×10^{-4}	2.88×10^{-4}	2.90×10^{-4}	2.89×10^{-4}	2.88×10^{-4}	2.90×10^{-4}
CPU time	530 s	3468 s	28423 s	631 s	6507 s	49003 s

Table 6.6: CPU time for simulating the differentially heated cavity flow with $T = 10, h = 1/64, \Delta t = 0.002$.

	Stab-SAV-BDF2-En	BDF2
CPU time (direct linear solver)	10401 s	97132 s
CPU time (iterative linear solver)	12584 s	44602 s

iterative solver mentioned above. The execution times are reported in Table 6.6. The temperature distributions and streamlines at $t = 5, 10$ are plotted in Fig. 6.4. Form the table and the figure we can see that no matter a direct linear solver or an iterative linear solver is used, the Stab-SAV-BDF2-En algorithm outperforms the BDF2 scheme since it takes much less CPU times while providing almost identical numerical solutions. The patterns in Fig. 6.4 also match well with the results in [15].

7. Conclusions. We proposed two second order ensemble schemes for fast computation of the Boussinesq flow ensembles combining the ensemble timestepping and the SAV approach. Both of the schemes are proved to be unconditionally long time stable without any time step constraints. These ensemble schemes are highly efficient as the coefficient matrices of the linear systems after spatial discretization are both time independent and ensemble index independent, for which special efficient block linear solvers can be used to significantly reduce the computational cost and CPU time. We also presented fully decoupled implementation algorithms for these schemes and explanations on the corresponding algebraic systems. Numerical experiments are performed to demonstrate that our schemes are second order convergent, and have comparable accuracy to standard non-ensemble methods while taking significantly less CPU time. The added stabilization term is very effective in increasing the schemes' accuracy and stability. We have focused on time discretization in this paper. The proposed ensemble algorithms can also be easily combined with nonforming mixed elements and other spatial discretization methods.

REFERENCES

[1] M. Akbas and L.G. Rebholz, *Modular grad-div stabilization for the incompressible non-isothermal fluid flows*, Applied Mathematics and Computation, 393 (2021), 125748.

[2] H. Calandra, S. Gratton, J. Langou, X. Pinel, X. Vasseur, *Flexible Variants of Block Restarted GMRES Methods with Application to Geophysics*, SIAM Journal on Scientific Computing, vol. 34, no. 2, (2012), 714-736.

[3] J. Carter, D. Han and N. Jiang, *Second order, unconditionally stable, linear ensemble algorithms for the Magnetohydrodynamics equations*, Journal of Scientific Computing, 94 (2023), 41.

[4] J. Carter and N. Jiang, *Numerical analysis of a second order ensemble method for evolutionary Magnetohydrodynamics equations at small Magnetic Reynolds number*, Numerical Methods for Partial Differential Equations, 38 (2022), 1407-1436.

[5] Y. T. Feng, D. R. J. Owen and D. Peric, *A block Conjugate Gradient method applied to linear systems with multiple right hand sides*, Comp. Meth. Appl. Mech., 127 (1995), 1-4.

[6] J. Fiordilino, *A second order ensemble timestepping algorithm for natural convection*, SIAM Journal on Numerical Analysis, 56 (2018), 816-837.

[7] J. Fiordilino, *Ensemble time-stepping algorithms for the heat equation with uncertain conductivity*, Numerical Methods for Partial Differential Equations, 34 (2018), 1901-1916.

[8] J. Fiordilino and S. Khankan, *Ensemble timestepping algorithms for natural convection*, International Journal of Numerical Analysis and Modeling, 15 (2018), 524-551.

[9] E. Gallopulos and V. Simoncini, *Convergence of BLOCK GMRES and matrix polynomials*, Lin. Alg. Appl., 247 (1996), 97-119.

[10] M. Gunzburger, T. Iliescu and M. Schneier, *A Leray regularized ensemble-proper orthogonal decomposition method for parameterized convection-dominated flows*, IMA Journal of Numerical Analysis, 40 (2020), 886-913.

[11] M. Gunzburger, N. Jiang and M. Schneier, *An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations*, SIAM Journal on Numerical Analysis, 55 (2017), 286-304.

[12] M. Gunzburger, N. Jiang and Z. Wang, *An efficient algorithm for simulating ensembles of parameterized flow problems*, IMA Journal of Numerical Analysis, 39 (2019), 1180-1205

[13] X. He, N. Jiang and C. Qiu, *An artificial compressibility ensemble algorithm for a stochastic Stokes-Darcy model with random hydraulic conductivity and interface conditions*, International Journal for Numerical Methods in Engineering, 121 (2020), 712-739.

[14] N. Jiang and W. Layton, *An algorithm for fast calculation of flow ensembles*, International Journal for Uncertainty Quantification, 4 (2014), 273-301.

[15] N. Jiang, *A pressure-correction ensemble scheme for computing evolutionary Boussinesq equations*, Journal of Scientific Computing, 80 (2019), 315-350.

[16] N. Jiang and W. Layton, *Numerical analysis of two ensemble eddy viscosity numerical regularizations of fluid motion*, Numerical Methods for Partial Differential Equations, 31 (2015), 630-651.

[17] N. Jiang, S. Kaya and W. Layton, *Analysis of model variance for ensemble based turbulence modeling*, Computational Methods in Applied Mathematics, 15 (2015), 173-188.

[18] N. Jiang, Y. Li and H. Yang, *An artificial compressibility Crank–Nicolson leap-frog method for the Stokes–Darcy model and application in ensemble simulations*, SIAM Journal on Numerical Analysis, 59 (2021), 401-428.

[19] N. Jiang, Y. Li and H. Yang, *A second order ensemble method with different subdomain time steps for simulating coupled surface-groundwater flows*, Numerical Methods for Partial Differential Equations, 38 (2022), 1880-1907.

[20] N. Jiang and C. Qiu, *An efficient ensemble algorithm for numerical approximation of stochastic Stokes-Darcy equations*, Computer Methods in Applied Mechanics and Engineering, 343 (2019), 249-275.

[21] N. Jiang and C. Qiu, *Numerical analysis of a second order ensemble algorithm for numerical approximation of stochastic Stokes-Darcy equations*, Journal of Computational and Applied Mathematics, 406 (2022), 113934.

[22] N. Jiang, and M. Schneier, *An efficient, partitioned ensemble algorithm for simulating ensembles of evolutionary MHD flows at low magnetic Reynolds number*, Numerical Methods for Partial Differential Equations, 34 (2018), 2129-2152.

[23] N. Jiang, A. Takhirov and J. Waters, *Robust SAV-ensemble algorithms for parametrized flow problems with energy stable open boundary conditions*, Computer Methods in Applied Mechanics and Engineering, 392 (2022), 114709.

[24] N. Jiang and H. Yang, *Stabilized scalar auxiliary variable ensemble algorithms for parameterized flow problems*, SIAM Journal on Scientific Computing, 43(4) (2021), A2869-A2896.

[25] N. Jiang and H. Yang, *SAV decoupled ensemble algorithms for fast computation of Stokes-Darcy flow ensembles*, Computer Methods in Applied Mechanics and Engineering, 387 (2021), 114150.

[26] N. Jiang and H. Yang, *Numerical investigation of two second-order, stabilized SAV ensemble methods for the Navier-Stokes equations*, Advances in Computational Mathematics, 48 (2022), 65.

[27] N. Jiang and H. Yang, *Artificial compressibility SAV ensemble algorithms for the incompressible Navier-Stokes equations*, Numerical Algorithms, 92 (2023), 2161-2188.

[28] X. Li and J. Shen, *Error analysis of the SAV-MAC scheme for the Navier-Stokes equations*, SIAM Journal on Numerical Analysis, 58 (2020), 2465-2491.

[29] L. Lin, Z. Yang and S. Dong, *Numerical approximation of incompressible Naiver-Stokes equations based on an auxiliary energy variable*, Journal of Computational Physics, 388 (2019), 1-22.

[30] J. Liu, C. Wang and H. Johnston, *A fourth order scheme for incompressible Boussinesq equations*, Journal of Scientific Computing, 18 (2003), 253-285.

[31] Y. Luo and Z. Wang, *An ensemble algorithm for numerical solutions to deterministic and random parabolic PDEs*, SIAM J. Numer. Anal., 56 (2018), 859-876.

[32] Y. Luo and Z. Wang, *A multilevel Monte Carlo ensemble scheme for random parabolic PDEs*, SIAM Journal on Scientific Computing, 41 (2019), A622-A642.

[33] Y. Li, W. Zhao and W. Zhao, *Optimal convergence of the scalar auxiliary variable finite element method for the natural convection equations*, Journal of Scientific Computing, 93 (2022), Article number 39.

[34] M. Mohebujjaman and L. Rebholz, *An efficient algorithm for computation of MHD flow ensembles*, Computational Methods in Applied Mathematics, 17 (2017), 121-137.

[35] J. Shen and J. Xu, *Convergence and error analysis for the scalar auxiliary variable (SAV) schemes to gradient flows*, SIAM Journal on Numerical Analysis, 56 (2018), 2895-2912.

[36] J. Shen, J. Xu and J. Yang, *The scalar auxiliary variable (SAV) approach for gradient flows*, Journal of Computational Physics, 353 (2018), 407-416.

[37] A. Takhirov, M. Neda, and J. Waters, *Time relaxation algorithm for flow ensembles*, Numerical Methods for Partial Differential Equations, 32 (2016), 757-777.

[38] A. Takhirov and J. Waters, *Ensemble algorithm for parametrized flow problems with energy stable open boundary conditions*, Computational Methods in Applied Mathematics, 20 (2020), 531-554.

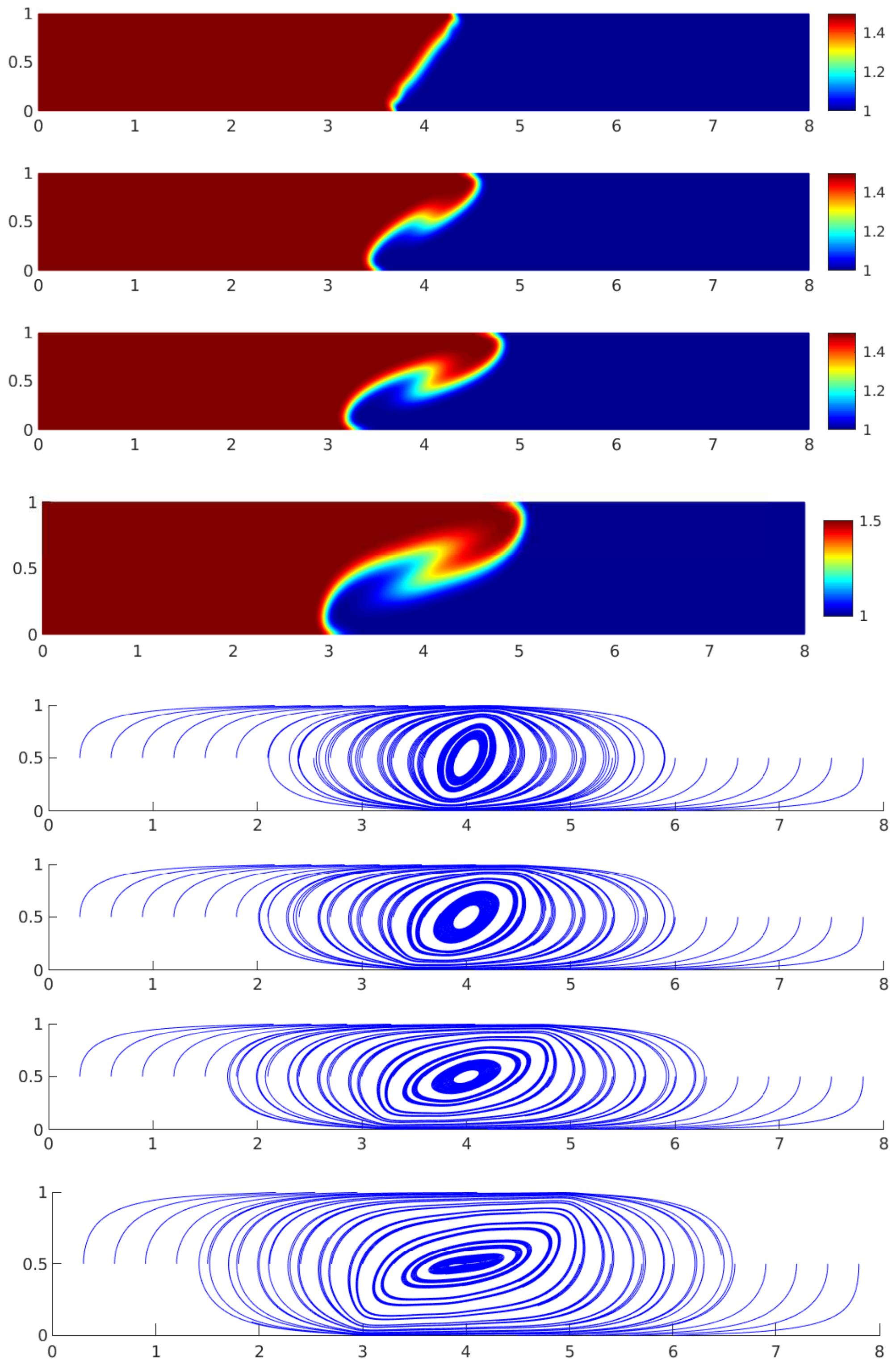

Fig. 6.2: Marsigli flow with $Re = 1000$ simulated by the Stab-SAV-BDF2-En scheme ($\alpha = 0, \beta = 0$). From top to bottom: temperature at $t = 2, 4, 6, 8$, streamline at $t = 2, 4, 6, 8$.

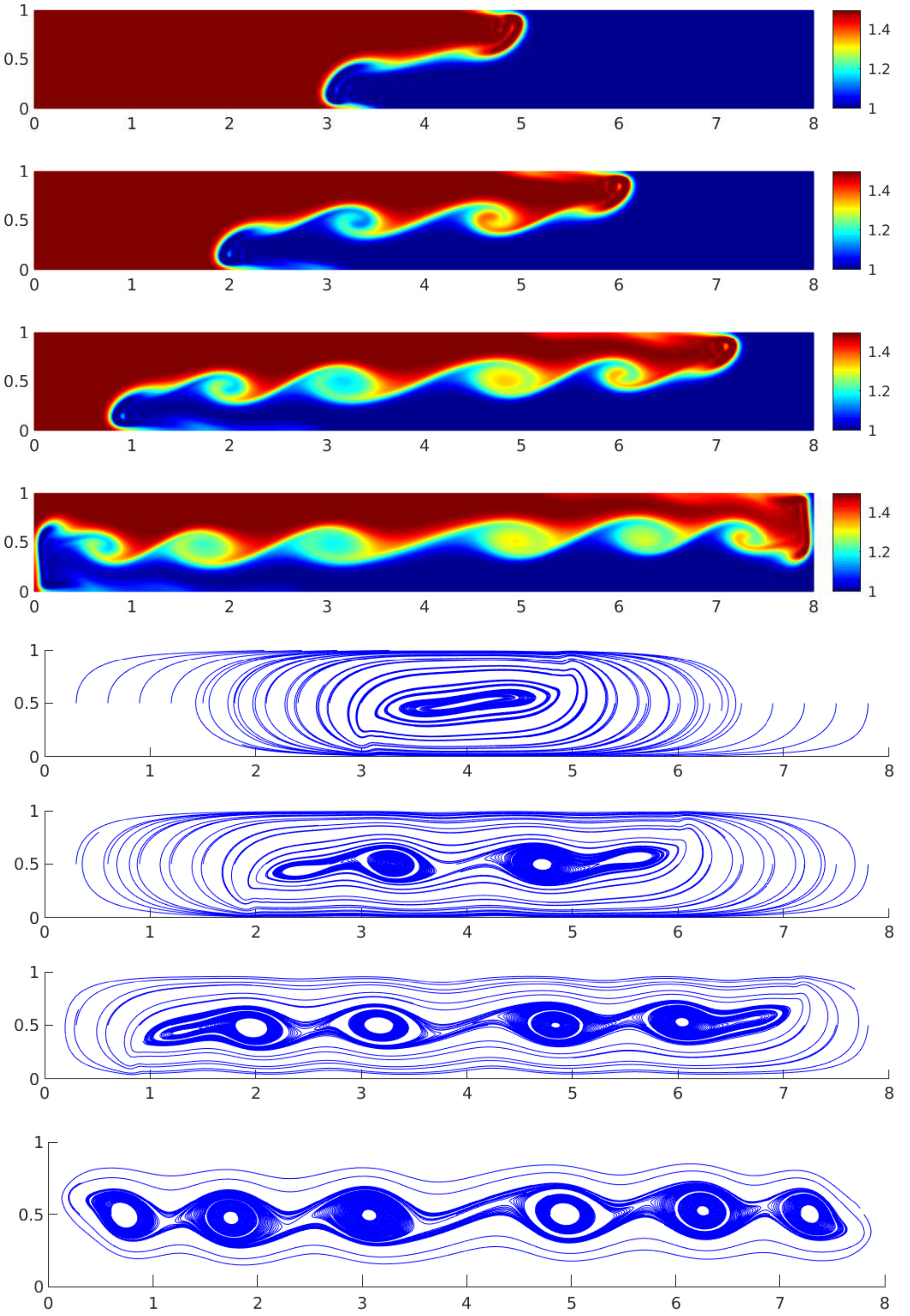

Fig. 6.3: Marsigli flow with $Re = 1000$ simulated by the Stab-SAV-BDF2-En scheme ($\alpha = 0.5, \beta = 0.5$). From top to bottom: temperature at $t = 2, 4, 6, 8$, streamline at $t = 2, 4, 6, 8$.

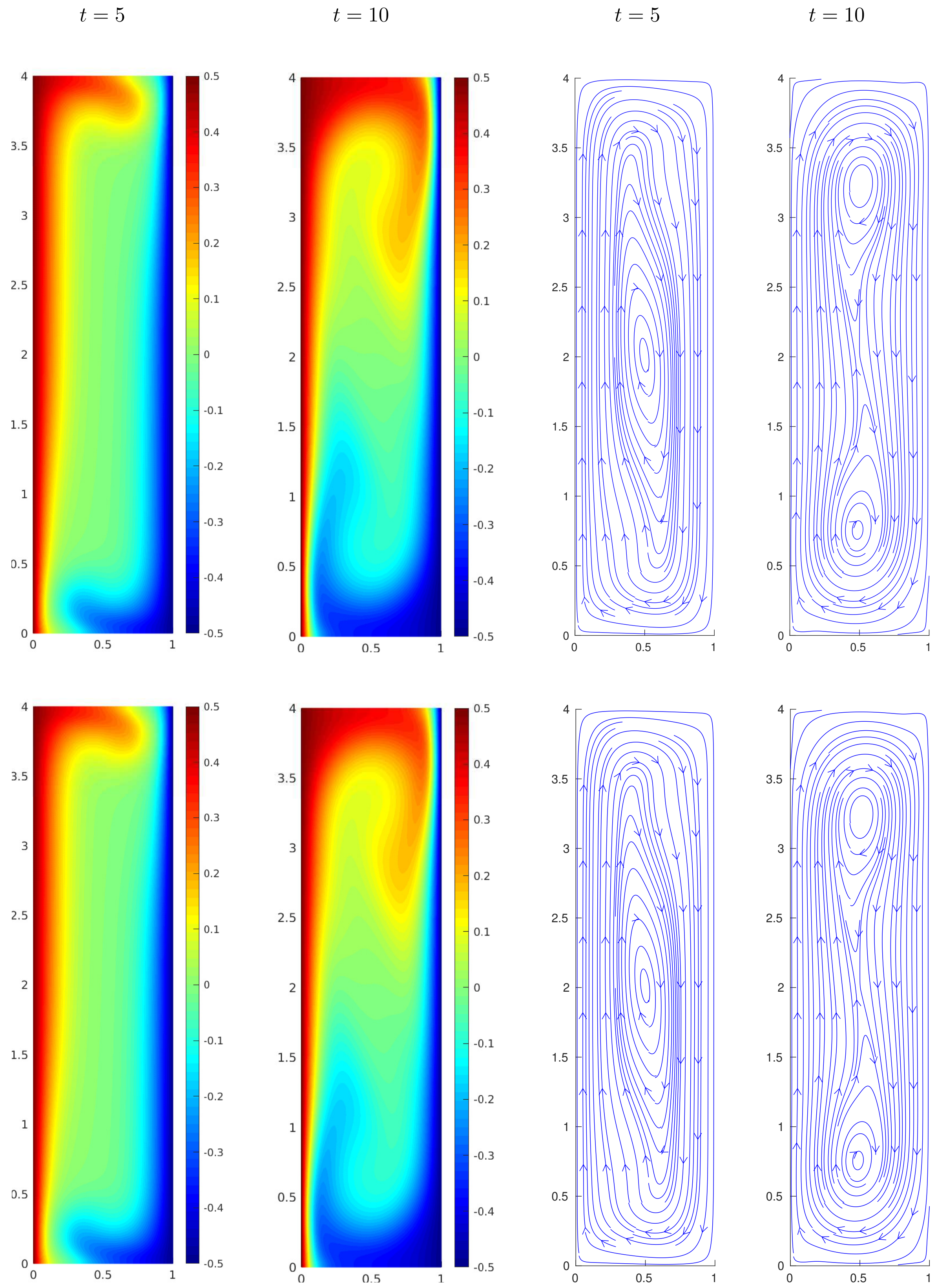

Fig. 6.4: Differentially heated cavity flow simulated by the Stab-SAV-BDF2-En scheme (top) and the BDF2 scheme (bottom). From left to right: temperature at $t = 5, 10$, streamline at $t = 5, 10$.