ARTIFICIAL COMPRESSIBILITY SAV ENSEMBLE ALGORITHMS FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

NAN JIANG* AND HUANHUAN YANG †

Abstract. This report presents two scalar auxiliary variable (SAV) ensemble algorithms based on artificial compressibility (AC) for fast computation of incompressible flow ensembles. We combine and exploit three numerical techniques: ensemble timestepping, SAV, AC, to design extremely efficient and fast algorithms for the computation of a (possibly large) Navier-Stokes flow ensemble. The proposed numerical algorithms feature that 1) all ensemble members share a common *constant* coefficient matrix allowing the use of efficient block solvers to significantly reduce required computational cost; 2) the computation of the velocity and the pressure is decoupled, and the pressure can be updated directly without solving a Possion equation, further reducing the overall computational cost. We prove both algorithms are long time stable under a parameter fluctuation condition, without any timestep constraints. Extensive numerical tests are also presented to demonstrate the efficiency and effectiveness of the ensemble algorithms.

Key words. Navier-Stokes equations, artificial compressibility, ensemble algorithm, uncertainty quantification, scalar auxiliary variable, stabilization

1. Introduction. Geophysical and engineering flow simulations are inevitably subject to uncertainties from the input data as well as model parameters. The need to address uncertainties has been well recognized in the past two decades and there has been an intense interest in modeling and quantifying uncertainties [1, 22, 48, 57–59]. The main challenge in implementing these models and the uncertainty quantification (UQ) methods in industrial applications is the associated high computational cost. Developing efficient UQ methods to reduce the cost has been an active research area with the main goal focused on reducing the sample size required for effective approximation of the stochastic space, e.g., [1, 24, 48, 57]. A different idea for reducing the computational cost was introduced in [27] which builds on the idea of ensemble timestepping and exploits the structure of corresponding linear systems for each realization, making it possible to design highly efficient ensemble simulation algorithms with computational cost not growing linearly with the number of samples. The ensemble algorithms have been extensively tested in the literature [6, 10–12, 16–18, 23, 26, 28, 29, 31, 32, 34, 38, 42, 43, 45, 53, 54] and shown to be able to effectively predict flow statistics in various UQ applications at significantly reduced computational cost.

The scalar auxiliary variable (SAV) approach was introduced in [51,52] for gradient flows and subsequentially adapted to solve the Navier-Stokes equations in [40, 41]. The idea is to introduce into the governing system a new scalar auxiliary variable and its associated differential equation that will facilitate the design of unconditionally stable numerical methods while treating the nonlinear term explicitly. With the nonlinear term explicit in the SAV schemes, the resulting linear system after spatial discretization is constant, which puts forth a new way to design and improve the ensemble algorithms both in efficiency and stability in the literature. In a very recent paper [31] the SAV approach was first combined with the ensemble timesteppng method to construct efficient ensemble simulation algorithms with provable long time stability without any timestep constraints. In this report, we propose to use the artificial compressibilty technique to further cut down the computational cost while maintaining comparable stability with the algorithms studied in [31].

One of the main difficulties in solving the Navier-Stokes equations is the incompressibility constraint and the coupling of the velocity and pressure. The artificial compressibility (AC) method was introduced in the 1960s by Chorin [5], Temam [55,56], Kuznetsov, Vladimirova and Yanenko [35] to relax the incompressibiltiy constraint by introducing a small perturbation which makes it possible to devise efficient numerical methods that decouple the computation of the velocity and pressure. The AC method had been much less studied in the literature, compared with the popular projection methods, until recently [4, 7–9, 13, 19–21, 23, 29, 36, 39, 47, 49]. The outstanding feature of the AC method is that the computation of the velocity and pressure is decoupled while the pressure can be updated directly without solving a Poisson equation, reducing the size of the algebraic linear systems after discretization and avoiding the boundary layers in pressure error due to artificial boundary condition that is commonly required by the projection methods. This work incorporates the AC technique into the stabilized SAV ensemble schemes [31] to design highly efficient ensemble simulation algorithms.

*Department of Mathematics, University of Florida, Gainesville, FL 32611, jiangn@ufl.edu..

†Corresponding author. Department of Mathematics, Shantou University, Guangdong, China 515063, huan2yang@stu.edu.cn.

2. The AC-SAV Ensemble Algorithms.

Model Problem. We consider the setting that the Navier-Stokes equations are subject to uncertainties in the initial conditions, boundary conditions, body forces, and the kinematic viscosity. Assuming a set of parameter samples have been generated using an ensemble based UQ method, e.g., [24, 57], and the next step is to compute the Navier-Stokes equations corresponding to each sample $(u_j^0(x),\, g_j(x,t),\, f_j(x,t),\, \nu_j(x))$, $j = 1, \cdots, J$. Our proposed ensemble algorithms are intended for fast calculation at this step and can be combined with any ensemble based UQ methods to accelerate the UQ procedure and extend its applicability in industrial applications.

The J Navier-Stokes equations on a bounded domain with J slightly different initial conditions, Dirichlet boundary conditions, body forces, and the kinematic viscosity $(u_j^0(x),\ g_j(x,t),\ f_j(x,t),\ \nu_j(x))$, for $j = 1, \cdots, J$, are given as follows.

$$\begin{aligned} \partial_t u_j + (u_j \cdot \nabla) u_j + \nabla \times (\nu_j (\nabla \times u_j)) + \nabla p_j &= f_j, \text{ in } \Omega, \\ \nabla \cdot u_j &= 0, \text{ in } \Omega, \\ u_j &= g_j, \text{ on } \partial\Omega, \\ u_j(x,0) &= u_j^0(x), \text{ in } \Omega. \end{aligned} \tag{2.1}$$

Algorithms. Herein we present two AC-SAV ensemble algorithms for fast computation of the Navier-Stokes equations with J different parameter samples. Standard ensemble methods [27] are usually associated with a possibly restrictive timestep condition that comes from bounding the nonlinear term in the Navier-Stokes equations. The adapting of the SAV approach in developing ensemble algorithms [31] resolved this issue as the nonlinear term will be canceled out in the stability proof with a smart manipulation of the added differential equation for the SAV, leading to fast ensemble algorithms with provable long time stability without any timestep constraints.

We first introduce the scalar auxiliary variables q_j, $j = 1, \cdots, J$, and their associated differential equations following [31]. Define the scalar auxiliary variable $q_j(t)$ by

$$q_j(t) = \sqrt{E(u_j) + \delta}, \tag{2.2}$$

where $E(u_j) = \int_\Omega \frac{1}{2}|u_j|^2\, dx$ is the total kinetic energy of the system and δ is an arbitrary positive constant.

The purpose of introducing this scalar variable is to facilitate the canceling of the fully explicit nonlinear term $\int_\Omega (u_j^n \cdot \nabla) u_j^n \cdot u_j^n\, dx$ in the stability analysis. Bounding this term in standard methods will usually lead to a restrictive CFL condition. Here the idea is to cancel out this nonlinear term directly by placing the term with a different sign in the differential equation for the scalar variable q_j, see (2.5), and avoid the need to bound this nonlinear term.

By the vector identity

$$(u_j \cdot \nabla) u_j \cdot u_j = \frac{1}{2} \nabla \cdot (u_j |u_j|^2) - \frac{1}{2} |u_j|^2 \nabla \cdot u_j, \tag{2.3}$$

and the fact that the velocity u_j is divergence free, using the equality $q_j(t) = \sqrt{E(u_j) + \delta}$, integrating over the flow domain Ω, we have the following equation

$$\frac{1}{2\sqrt{E(u_j) + \delta}} \int_\Omega (u_j \cdot \nabla) u_j \cdot u_j\, dx - \frac{1}{2q_j} \int_{\partial\Omega} (\vec{n} \cdot u_j) \tfrac{1}{2} |u_j|^2\, d\sigma = 0. \tag{2.4}$$

Letting $\bar{\nu}(x) := \frac{1}{J} \sum_{j=1}^J \nu_j(x)$ denote the ensemble mean of the viscosity, taking derivative of $q_j(t)$, using $\nabla \cdot u_j = 0$, and adding the above zero term gives the following ordinary differential equation

$$\begin{aligned} \frac{dq_j}{dt} = \frac{1}{2q_j} \int_\Omega \frac{\partial u_j}{\partial t} \cdot u_j\, dx - \frac{1}{2q_j} \int_\Omega \bar{\nu} (\nabla \cdot u_j)^2\, dx \\ + \frac{1}{2\sqrt{E(u_j) + \delta}} \int_\Omega (u_j \cdot \nabla) u_j \cdot u_j\, dx - \frac{1}{2q_j} \int_{\partial\Omega} (\vec{n} \cdot u_j) \tfrac{1}{2} |u_j|^2\, d\sigma. \end{aligned} \tag{2.5}$$

To derive an unconditionally stable scheme, we need to incorporate the above differential equation into the original Navier-Stokes equations and have a new coupled PDE system. To couple the new variable q_j with the original variables u_j, p_j, we will add a new term $\frac{q_j(t)}{\sqrt{E(u_j)+\delta}}$, which is equal to 1 by definition, in front of the nonlinear term $(u_j \cdot \nabla)u_j$ in the momentum equation of the Navier-Stokes equations to form a coupled system of u_j, p_j, q_j. And this is the reason why we have $\frac{1}{\sqrt{E(u_j)+\delta}}$ in front of the integral of the nonlinear term in (2.5): multiplying both sides of (2.5) by $2q_j$ in the stability proof will yield $\frac{q_j(t)}{\sqrt{E(u_j)+\delta}}(u_j \cdot \nabla)u_j$, which is the same as the nonlinear term from the momentum equation and thus can be canceled out. Note that the nonlinear term in the zero term in (2.5) plays an essential role in the design of the SAV ensemble algorithms and eliminating the requirement of a timestep constraint. Adding (2.5) to the original Navier-Stokes equations forms a new governing system with an added new variable q_j:

$$
\begin{aligned}
&\partial_t u_j + \frac{q_j(t)}{\sqrt{E(u_j)+\delta}}(u_j \cdot \nabla)u_j + \nabla \times (\nu_j(\nabla \times u_j)) + \nabla p_j = f_j(x,t), \qquad (2.6)\\
&\nabla \cdot u_j = 0,\\
&\frac{dq_j}{dt} = \frac{1}{2q_j}\int_\Omega \frac{\partial u_j}{\partial t}\cdot u_j\,dx - \frac{1}{2q_j}\int_\Omega \bar{\nu}(\nabla \cdot u_j)^2\,dx\\
&\qquad\qquad + \frac{1}{2\sqrt{E(u_j)+\delta}}\int_\Omega (u_j \cdot \nabla)u_j \cdot u_j\,dx - \frac{1}{2q_j}\int_{\partial\Omega}(\vec{n}\cdot g_j)\tfrac{1}{2}|g_j|^2\,d\sigma.
\end{aligned}
$$

System (2.6) is equivalent to (2.1), and we will present numerical algorithms that approximate the solution of this new governing system. The essential idea of the ensemble timestepping method is to keep all realizations sharing the same coefficient matrix. So here we need to split the viscosity into two parts: the mean $\bar{\nu}$ and the fluctuation ν_j', and lag the fluctuation term to the previous time levels so that it does not contribute to the coefficient matrix. The ensemble fluctuations are defined as follows.

$$\nu_j'(x) := \nu_j(x) - \bar{\nu}(x), \quad j = 1, \cdots, J.$$

Let $t_n = n\Delta t$, $n = 0, 1, 2, \cdots, N$, where $N = T/\Delta t$, denote a uniform partition of the interval $[0, T]$. We now present two AC-SAV ensemble algorithms for efficient approximation of (2.6). The first order scheme based on the backward Euler (BE) timestepping reads

ALGORITHM 2.1 (AC-SAV-BE). *For $j = 1, 2, \cdots, J$, given u_j^1, p_j^1, q_j^1, for $n = 2, 3, \cdots, N-1$, find $u_j^{n+1}, p_j^{n+1}, q_j^{n+1}$ satisfying*

$$
\begin{aligned}
&\frac{u_j^{n+1}-u_j^n}{\Delta t} + \frac{q_j^{n+1}}{\sqrt{E(u_j^n)+\delta}}(u_j^n \cdot \nabla)u_j^n + \nabla p_j^{n+1} + \nabla \times (\bar{\nu}(\nabla \times u_j^{n+1})) \qquad (2.7)\\
&\qquad\qquad + \nabla \times (\nu_j'(\nabla \times u_j^n)) - \alpha h \Delta(u_j^{n+1} - u_j^n) = f_j^{n+1},\\
&\beta(p_j^{n+1} - p_j^n) + \nabla \cdot u_j^{n+1} = 0, \qquad (2.8)\\
&\frac{q_j^{n+1}-q_j^n}{\Delta t} = \frac{1}{2q_j^{n+1}}\left(\frac{u_j^{n+1}-u_j^n}{\Delta t}, u_j^{n+1}\right) - \frac{1}{2q_j^{n+1}}\int_\Omega \bar{\nu}(\nabla \cdot u_j^{n+1})^2\,dx \qquad (2.9)\\
&\qquad\qquad + \frac{1}{2\sqrt{E(u_j^n)+\delta}}\int_\Omega (u_j^n \cdot \nabla)u_j^n \cdot u_j^{n+1}\,dx - \frac{b_j^{n+1}}{2q_j^{n+1}},
\end{aligned}
$$

where $b_j^{n+1} = \int_{\partial\Omega}(\vec{n}\cdot g_j^{n+1})\frac{1}{2}|g_j^{n+1}|^2\,d\sigma$.

The second order scheme based on the second order backward differentiation formula (BDF2) is given by

ALGORITHM 2.2 (AC-SAV-BDF2). *For $j = 1, 2, \cdots, J$, given $u_j^1, u_j^2, p_j^1, p_j^2, q_j^1, q_j^2$, for $n = 2, 3, \cdots, N-1$, find $u_j^{n+1}, p_j^{n+1}, q_j^{n+1}$ satisfying*

$$\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t}+\frac{q_j^{n+1}}{\sqrt{E(\tilde{u}_j^{n+1})+\delta}}(\tilde{u}_j^{n+1}\cdot\nabla)\tilde{u}_j^{n+1}+\nabla p_j^{n+1} \tag{2.10}$$

$$+\nabla\times(\bar{\nu}(\nabla\times u_j^{n+1}))+\nabla\times(\nu_j'(\nabla\times\tilde{u}_j^{n+1}))-\tfrac{1}{2}\alpha h\Delta(3u_j^{n+1}-4u_j^n+u_j^{n-1})=f_j^{n+1},$$

$$\beta\Delta t(3p_j^{n+1}-4p_j^n+p_j^{n-1})+\nabla\cdot u_j^{n+1}=0, \tag{2.11}$$

$$\frac{3q_j^{n+1}-4q_j^n+q_j^{n-1}}{2\Delta t}=\frac{1}{2q_j^{n+1}}\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},u_j^{n+1}\right)-\frac{1}{2q_j^{n+1}}\int_\Omega\bar{\nu}(\nabla\cdot u_j^{n+1})^2\,dx \tag{2.12}$$

$$+\frac{1}{2\sqrt{E(\tilde{u}_j^{n+1})+\delta}}\int_\Omega(\tilde{u}_j^{n+1}\cdot\nabla)\tilde{u}_j^{n+1}\cdot u_j^{n+1}\,dx-\frac{b_j^{n+1}}{2q_j^{n+1}},$$

where $b_j^{n+1}=\int_{\partial\Omega}(\vec{n}\cdot g_j^{n+1})\frac{1}{2}|g_j^{n+1}|^2\,d\sigma$ *and* $\tilde{u}_j^{n+1}=2u_j^n-u_j^{n-1}$.

Note that the initial pressure p_j^0 is usually unknown and one needs to use a different method, e.g., standard BE or BDF2, to compute p_j^1 and p_j^2 as well as u_j^1 and u_j^2. The values of q_j^1 and q_j^2 can be computed by the definition directly.

In these two algorithms we incorporated a stabilization technique from [31] to increase the accuracy of the SAV schemes. The stabilizations $-\alpha h\Delta(u_j^{n+1}-u_j^n)$ for the AC-SAV-BE scheme and $-\frac{1}{2}\alpha h\Delta(3u_j^{n+1}-4u_j^n+u_j^{n-1})$ for the AC-SAV-BDF2 scheme add artificial viscosity to the system to better condition the linear systems to be solved after spatial discretization, and also add antidiffusion at previous time levels to avoid overdiffusion. They are discretized forms of $-\alpha h\Delta t\Delta(\partial_t u)$ in the first and second order respectively. It was demonstrated in [31] that this stabilization techniques can effectively increase the accuracy of SAV methods.

The AC method is incorporated by adding a small perturbation ϵp_t to the mass conservation equation, and discretized as $\beta\Delta t\frac{p_j^{n+1}-p_j^n}{\Delta t}$ for the AC-SAV-BE scheme and $2\beta\Delta t^2\frac{3p_j^{n+1}-4p_j^n+p_j^{n-1}}{2\Delta t}$ for AC-SAV-BDF2. With (2.8) and (2.11), we can replace p_j^{n+1} in the momentum equation and only solve the equation for u_j^{n+1} avoiding solving a saddle point problem. Then p_j^{n+1} can be updated directly after obtaining u_j^{n+1} without solving an addition Poisson equation which was required by the projection methods.

We will show how to efficiently implement these two ensemble algorithms in Section 4, where details on implementation algorithms and linear solvers will be discussed.

3. Long Time Stability of the AC-SAV Ensemble Algorithms. We next prove the long time stability of both algorithms without any timestep constraints. There is a parameter condition that needs to be satisfied for each algorithm hence limiting the extent of the fluctuation of the viscosity parameter. This condition is not difficult to satisfy in UQ applications where the magnitude of the fluctuation is usually small. Moreover, a larger ensemble can be split into smaller ensembles and the ensemble algorithms can be applied to each smaller ensemble. One should also notice that this parameter condition only exists when computing ensembles. For computing just one realization, both of the two algorithms are unconditionally stable.

We assume $\nu_j(x)\in L^\infty(\Omega)$ with $\nu_j(x)\geq\nu_{j,min}>0$. Define the minimum average $\bar{\nu}_{min}$ and the maximum fluctuation ν'_{max} of the kinematic viscosity as

$$\bar{\nu}_{min}:=\frac{1}{J}\sum_{j=1}^J\nu_{j,min},\quad \nu'_{max}:=\max_j\sup_{x\in\Omega}|\nu_j'(x)|.$$

In the stability proof we will assume q_j^n is real for any $n=0,\cdots,N$, $j=1,\cdots,J$ so that $|q_j^n|$ is nonnegative. In the simulations, if q_j^n ever becomes complex, the simulation is getting unstable and should be stopped. Nevertheless, it is shown in our numerical experiments that the stabilization we incorporated in the algorithms is very effective in increasing the accuracy of the algorithms and preventing the q_j^n from becoming complex.

THEOREM 3.1 (Long Time Stability of AC-SAV-BE). *Assume the parameter condition* $\nu'_{max}<\bar{\nu}_{min}$ *holds and* q_j^n *is real for any* $n=0,\cdots,N$, $j=1,\cdots,J$. *With homogeneous Dirichlet boundary condition,*

Algorithm 2.1 is nonlinearly, long time stable, and the following energy inequality holds

$$
\begin{aligned}
&|q_j^N|^2 + \Delta t \sum_{n=1}^{N-1} |q_j^{n+1} - q_j^n|^2 + \frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^N\|^2 + \frac{\alpha}{2}h\Delta t\|\nabla u_j^N\|^2 + \frac{\beta}{2}\Delta t\|p_j^N\|^2 \\
&+ \frac{\beta}{2}\Delta t \sum_{n=1}^{N-1} \|p_j^{n+1} - p_j^n\|^2 + \frac{\alpha}{2}h\Delta t \sum_{n=1}^{N-1} \|\nabla u_j^{n+1} - u_j^n\|^2 \qquad (3.1)\\
&\le |q_j^1|^2 + \frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^1\|^2 + \frac{\beta}{2}\Delta t\|p_j^1\|^2 + \frac{\alpha}{2}h\Delta t\|\nabla u_j^1\|^2 + \frac{\Delta t}{2(\bar{\nu}_{min} - \nu'_{max})} \sum_{n=1}^{N-1} \|f_j^{n+1}\|_{-1}^2.
\end{aligned}
$$

Proof. Taking the L^2 inner product of (2.7) with u_j^{n+1}, (2.8) with p_j^{n+1} and adding the two equations gives

$$
\begin{aligned}
&\left(\frac{u_j^{n+1} - u_j^n}{\Delta t}, u_j^{n+1}\right) + \frac{q_j^{n+1}}{\sqrt{E(u_j^n) + \delta}} b(u_j^n, u_j^n, u_j^{n+1}) + \int_{\partial\Omega} (\vec{n} \cdot u_j^{n+1}) p_j^{n+1} d\sigma \\
&+ \frac{\beta}{2}\left(\|p_j^{n+1}\|^2 - \|p_j^n\|^2 + \|p_j^{n+1} - p_j^n\|^2\right) + (\bar{\nu}\nabla \times u_j^{n+1}, \nabla \times u_j^{n+1}) - \int_{\partial\Omega} (\vec{n} \cdot (u_j^{n+1} \times (\bar{\nu}\nabla \times u_j^{n+1})))d\sigma \\
&+ (\nu'_j \nabla \times u_j^n, \nabla \times u_j^{n+1}) - \int_{\partial\Omega} (\vec{n} \cdot (u_j^{n+1} \times (\nu'_j \nabla \times u_j^n)))d\sigma - \alpha h \int_{\partial\Omega} (\vec{n} \cdot (\nabla u_j^{n+1} - \nabla u_j^n)) \cdot u_j^{n+1} d\sigma \\
&+ \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^n\|^2 + \|\nabla u_j^{n+1} - \nabla u_j^n\|^2\right) = (f_j^{n+1}, u_j^{n+1}). \qquad (3.2)
\end{aligned}
$$

Multiplying (2.9) with $2q_j^{n+1}$ gives

$$
\begin{aligned}
&\frac{1}{\Delta t}\left(|q_j^{n+1}|^2 - |q_j^n|^2 + |q_j^{n+1} - q_j^n|^2\right) + \left(\bar{\nu}\nabla \cdot u_j^{n+1}, \nabla \cdot u_j^{n+1}\right) \\
&\qquad = \left(\frac{u_j^{n+1} - u_j^n}{\Delta t}, u_j^{n+1}\right) + \frac{q_j^{n+1}}{\sqrt{E(u_j^n) + \delta}} b(u_j^n, u_j^n, u_j^{n+1}) - b_j^{n+1}. \qquad (3.3)
\end{aligned}
$$

Adding (3.2) and (3.3) gives

$$
\begin{aligned}
&\frac{1}{\Delta t}\left(|q_j^{n+1}|^2 - |q_j^n|^2 + |q_j^{n+1} - q_j^n|^2\right) + (\bar{\nu}\nabla \times u_j^{n+1}, \nabla \times u_j^{n+1}) + \left(\bar{\nu}\nabla \cdot u_j^{n+1}, \nabla \cdot u_j^{n+1}\right) \\
&+ \frac{\beta}{2}\left(\|p_j^{n+1}\|^2 - \|p_j^n\|^2 + \|p_j^{n+1} - p_j^n\|^2\right) + \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^n\|^2 + \|\nabla u_j^{n+1} - \nabla u_j^n\|^2\right) \qquad (3.4)\\
&= (f_j^{n+1}, u_j^{n+1}) - (\nu'_j \nabla \times u_j^n, \nabla \times u_j^{n+1}) + \int_{\partial\Omega} (\vec{n} \cdot (u_j^{n+1} \times (\nu'_j \nabla \times u_j^n)))d\sigma \\
&+ \int_{\partial\Omega} (\vec{n} \cdot (u_j^{n+1} \times (\bar{\nu}\nabla \times u_j^{n+1})))d\sigma - \int_{\partial\Omega} (\vec{n} \cdot u_j^{n+1}) p_j^{n+1} d\sigma + \alpha h \int_{\partial\Omega} (\vec{n} \cdot (\nabla u_j^{n+1} - \nabla u_j^n)) \cdot u_j^{n+1} d\sigma - b_j^{n+1}.
\end{aligned}
$$

Assuming homogeneous Dirichlet condition, applying Cauchy-Schwarz inequality gives

$$
\begin{aligned}
&\frac{1}{\Delta t}\left(|q_j^{n+1}|^2 - |q_j^n|^2 + |q_j^{n+1} - q_j^n|^2\right) + \bar{\nu}_{min}\|\nabla \times u_j^{n+1}\|^2 + \bar{\nu}_{min}\|\nabla \cdot u_j^{n+1}\|^2 \\
&+ \frac{\beta}{2}\left(\|p_j^{n+1}\|^2 - \|p_j^n\|^2 + \|p_j^{n+1} - p_j^n\|^2\right) + \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^n\|^2 + \|\nabla u_j^{n+1} - \nabla u_j^n\|^2\right) \\
&\le \|f_j^{n+1}\|_{-1}\|\nabla u_j^{n+1}\| + \nu'_{max}\|\nabla \times u_j^n\|\|\nabla \times u_j^{n+1}\|. \qquad (3.5)
\end{aligned}
$$

Since $u_j = 0$ on $\partial\Omega$, we have

$$
\|\nabla u_j\|^2 = \|\nabla \cdot u_j\|^2 + \|\nabla \times u_j\|^2. \qquad (3.6)
$$

Then using Young's inequalities to the right hand side of (3.5) gives, for any $\beta_1 > 0, \epsilon_1 > 0$,

$$\frac{1}{\Delta t}\left(|q_j^{n+1}|^2-|q_j^n|^2+|q_j^{n+1}-q_j^n|^2\right)+\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2+\frac{\beta}{2}\left(\|p_j^{n+1}\|^2-\|p_j^n\|^2+\|p_j^{n+1}-p_j^n\|^2\right)$$
$$+\frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^n\|^2+\|\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)$$
$$\leq \beta_1\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2+\frac{1}{4\beta_1\bar{\nu}_{min}}\|f_j^{n+1}\|_{-1}^2+\frac{\epsilon_1\nu'_{max}}{2}\|\nabla u_j^{n+1}\|^2+\frac{\nu'_{max}}{2\epsilon_1}\|\nabla u_j^n\|^2. \tag{3.7}$$

As the last two terms all need to be bounded by $\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2$, we want to minimize $\frac{\epsilon_1}{2}+\frac{1}{2\epsilon_1}$ by taking $\epsilon_1=1$. (3.5) then reduces to

$$\frac{1}{\Delta t}\left(|q_j^{n+1}|^2-|q_j^n|^2+|q_j^{n+1}-q_j^n|^2\right)+\left((1-\beta_1)\bar{\nu}_{min}-\nu'_{max}\right)\|\nabla u_j^{n+1}\|^2 \tag{3.8}$$
$$+\frac{1}{2}\nu'_{max}\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^n\|^2\right)+\frac{\beta}{2}\left(\|p_j^{n+1}\|^2-\|p_j^n\|^2+\|p_j^{n+1}-p_j^n\|^2\right)$$
$$+\frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^n\|^2+\|\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)\leq\frac{1}{4\beta_1\bar{\nu}_{min}}\|f_j^{n+1}\|_{-1}^2.$$

If the parameter condition is satisfied, then $\bar{\nu}_{min}-\nu'_{max}>0$. Taking $\beta_1=\frac{1}{2}-\frac{1}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}}>0$, we have

$$(1-\beta_1)\bar{\nu}_{min}-\nu'_{max}=(\frac{1}{2}+\frac{1}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}})\bar{\nu}_{min}-\nu'_{max}=\frac{1}{2}\left(\bar{\nu}_{min}-\nu'_{max}\right)>0. \tag{3.9}$$

(3.8) can then be reduced to

$$\frac{1}{\Delta t}\left(|q_j^{n+1}|^2-|q_j^n|^2+|q_j^{n+1}-q_j^n|^2\right)+\frac{1}{2}\nu'_{max}\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^n\|^2\right)$$
$$+\frac{\beta}{2}\left(\|p_j^{n+1}\|^2-\|p_j^n\|^2+\|p_j^{n+1}-p_j^n\|^2\right)+\frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^n\|^2+\|\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)$$
$$\leq\frac{1}{2(\bar{\nu}_{min}-\nu'_{max})}\|f_j^{n+1}\|_{-1}^2. \tag{3.10}$$

Summing up from $n=1$ to $n=N-1$ and multiplying through by Δt gives

$$|q_j^N|^2+\Delta t\sum_{n=1}^{N-1}|q_j^{n+1}-q_j^n|^2+\frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^N\|^2+\frac{\alpha}{2}h\Delta t\|\nabla u_j^N\|^2+\frac{\beta}{2}\Delta t\|p_j^N\|^2$$
$$+\frac{\beta}{2}\Delta t\sum_{n=1}^{N-1}\|p_j^{n+1}-p_j^n\|^2+\frac{\alpha}{2}h\Delta t\sum_{n=1}^{N-1}\|\nabla u_j^{n+1}-u_j^n\|^2 \tag{3.11}$$
$$\leq|q_j^1|^2+\frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^1\|^2+\frac{\beta}{2}\Delta t\|p_j^1\|^2+\frac{\alpha}{2}h\Delta t\|\nabla u_j^1\|^2+\frac{\Delta t}{2(\bar{\nu}_{min}-\nu'_{max})}\sum_{n=1}^{N-1}\|f_j^{n+1}\|_{-1}^2.$$

□

Next we prove the long time stability of the AC-SAV-BDF2 scheme.

THEOREM 3.2 (Long Time Stability of AC-SAV-BDF2). *Assume the parameter condition $\nu'_{max}<\frac{1}{3}\bar{\nu}_{min}$ holds and q_j^n is real for any $n=0,\cdots,N$, $j=1,\cdots,J$. With homogeneous Dirichlet boundary condition, Algorithm 2.2 is nonlinearly, long time stable, and the following energy inequality holds*

$$|q_j^N|^2+|2q_j^N-q_j^{N-1}|^2+\sum_{n=2}^{N-1}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2+\beta\Delta t^2\left(\|p_j^N\|^2+\|2p_j^N-p_j^{N-1}\|^2\right)$$
$$+\beta\Delta t^2\sum_{n=2}^{N-1}\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|^2+\frac{1}{2}\alpha h\Delta t\left(\|\nabla u_j^N\|^2+\|2\nabla u_j^N-\nabla u_j^{N-1}\|^2\right)$$

$$+\frac{1}{2}\alpha h\Delta t\sum_{n=2}^{N-1}\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2+3\nu'_{max}\Delta t\|\nabla u_j^N\|^2+\nu'_{max}\Delta t\|\nabla u_j^{N-1}\|^2 \tag{3.12}$$
$$\leq |q_j^1|^2+|2q_j^1-q_j^0|^2+\beta\Delta t^2\left(\|p_j^1\|^2+\|2p_j^1-p_j^0\|^2\right)+\frac{1}{2}\alpha h\Delta t\left(\|\nabla u_j^1\|^2+\|2\nabla u_j^1-\nabla u_j^0\|^2\right)$$
$$+3\nu'_{max}\Delta t\|\nabla u_j^1\|^2+\nu'_{max}\Delta t\|\nabla u_j^0\|^2+\frac{\Delta t}{\bar{\nu}_{min}-3\nu'_{max}}\sum_{n=2}^{N-1}\|f_j^{n+1}\|_{-1}^2.$$

Proof. Taking the L^2 inner product of (2.10) with u_j^{n+1}, (2.11) with p_j^{n+1} and adding the two equations gives

$$\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},u_j^{n+1}\right)+\frac{q_j^{n+1}}{\sqrt{E(\tilde{u}_j^{n+1})+\delta}}b(\tilde{u}_j^{n+1},\tilde{u}_j^{n+1},u_j^{n+1})+\int_{\partial\Omega}(\vec{n}\cdot u_j^{n+1})p_j^{n+1}d\sigma$$
$$+\frac{\beta}{2}\Delta t\left(\|p_j^{n+1}\|^2+\|2p_j^{n+1}-p_j^n\|^2\right)-\frac{\beta}{2}\Delta t\left(\|p_j^n\|^2+\|2p_j^n-p_j^{n-1}\|^2\right)+\frac{\beta}{2}\Delta t\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|^2$$
$$+(\bar{\nu}\nabla\times u_j^{n+1},\nabla\times u_j^{n+1})-\int_{\partial\Omega}(\vec{n}\cdot(u_j^{n+1}\times(\bar{\nu}\nabla\times u_j^{n+1})))d\sigma+(\nu'_j\nabla\times\tilde{u}_j^{n+1},\nabla\times u_j^{n+1})$$
$$-\int_{\partial\Omega}(\vec{n}\cdot(u_j^{n+1}\times(\nu'_j\nabla\times\tilde{u}_j^{n+1})))d\sigma-\frac{\alpha}{2}h\int_{\partial\Omega}(\vec{n}\cdot(3\nabla u_j^{n+1}-4\nabla u_j^n+\nabla u_j^{n-1}))\cdot u_j^{n+1}d\sigma$$
$$+\frac{\alpha}{4}h\left(\|\nabla u_j^{n+1}\|^2+\|2\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)-\frac{\alpha}{4}h\left(\|\nabla u_j^n\|^2+\|2\nabla u_j^n-\nabla u_j^{n-1}\|^2\right)$$
$$+\frac{\alpha}{4}h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2=(f_j^{n+1},u_j^{n+1}). \tag{3.13}$$

Multiplying (2.12) with $2q_j^{n+1}$ gives

$$\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2+|2q_j^{n+1}-q_j^n|^2\right)-\frac{1}{2\Delta t}\left(|q_j^n|^2+|2q_j^n-q_j^{n-1}|^2\right)+\frac{1}{2\Delta t}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2 \tag{3.14}$$
$$+\left(\bar{\nu}\nabla\cdot u_j^{n+1},\nabla\cdot u_j^{n+1}\right)=\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},u_j^{n+1}\right)+\frac{q_j^{n+1}}{\sqrt{E(\tilde{u}_j^{n+1})+\delta}}b(\tilde{u}_j^{n+1},\tilde{u}_j^{n+1},u_j^{n+1})-b_j^{n+1}.$$

Adding (3.13) and (3.14) gives

$$\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2+|2q_j^{n+1}-q_j^n|^2\right)-\frac{1}{2\Delta t}\left(|q_j^n|^2+|2q_j^n-q_j^{n-1}|^2\right)+\frac{1}{2\Delta t}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2$$
$$+\frac{\beta}{2}\Delta t\left(\|p_j^{n+1}\|^2+\|2p_j^{n+1}-p_j^n\|^2\right)-\frac{\beta}{2}\Delta t\left(\|p_j^n\|^2+\|2p_j^n-p_j^{n-1}\|^2\right)+\frac{\beta}{2}\Delta t\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|^2$$
$$+(\bar{\nu}\nabla\times u_j^{n+1},\nabla\times u_j^{n+1})+\left(\bar{\nu}\nabla\cdot u_j^{n+1},\nabla\cdot u_j^{n+1}\right)+\frac{\alpha}{4}h\left(\|\nabla u_j^{n+1}\|^2+\|2\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)$$
$$-\frac{\alpha}{4}h\left(\|\nabla u_j^n\|^2+\|2\nabla u_j^n-\nabla u_j^{n-1}\|^2\right)+\frac{\alpha}{4}h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2 \tag{3.15}$$
$$=(f_j^{n+1},u_j^{n+1})-(\nu'_j\nabla\times\tilde{u}_j^{n+1},\nabla\times u_j^{n+1})+\int_{\partial\Omega}(\vec{n}\cdot(u_j^{n+1}\times(\nu'_j\nabla\times\tilde{u}_j^{n+1})))d\sigma$$
$$+\int_{\partial\Omega}(\vec{n}\cdot(u_j^{n+1}\times(\bar{\nu}\nabla\times u_j^{n+1})))d\sigma+\frac{1}{4}\alpha h\int_{\partial\Omega}(\vec{n}\cdot(3\nabla u_j^{n+1}-4\nabla u_j^n+\nabla u_j^{n-1}))\cdot u_j^{n+1}d\sigma$$
$$-\int_{\partial\Omega}(\vec{n}\cdot u_j^{n+1})p_j^{n+1}d\sigma-b_j^{n+1}.$$

In particular, with homogeneous Dirichlet condition the integrals on the boundary $\partial\Omega$ are equal to zero. Using the fact that $\|\nabla\times u_j\|^2+\|\nabla\cdot u_j\|^2=\|\nabla u_j\|^2$, applying Cauchy-Schwarz and Young's inequalities to the right hand side and and using $(2a-b)^2\leq 6a^2+3b^2$ gives, for any $\beta_2>0,\epsilon_2>0$,

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

$$
\begin{aligned}
&+\frac{\beta}{2}\Delta t\left(\|p_j^{n+1}\|^2+\|2p_j^{n+1}-p_j^n\|^2\right)-\frac{\beta}{2}\Delta t\left(\|p_j^n\|^2+\|2p_j^n-p_j^{n-1}\|^2\right)+\frac{\beta}{2}\Delta t\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|^2\\
&+\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2+\frac{\alpha}{4}h\left(\|\nabla u_j^{n+1}\|^2+\|2\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)-\frac{\alpha}{4}h\left(\|\nabla u_j^n\|^2+\|2\nabla u_j^n-\nabla u_j^{n-1}\|^2\right)\\
&+\frac{\alpha}{4}h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2\\
&\le\|f_j^{n+1}\|_{-1}\|\nabla u_j^{n+1}\|+\nu'_{max}\|\nabla\tilde{u}_j^{n+1}\|\|\nabla u_j^{n+1}\| \qquad (3.16)\\
&\le\beta_2\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2+\frac{1}{4\beta_2\bar{\nu}_{min}}\|f_j^{n+1}\|_{-1}^2+\frac{\epsilon_2\nu'_{max}}{2}\|\nabla u_j^{n+1}\|^2+\frac{\nu'_{max}}{2\epsilon_2}\|\nabla\tilde{u}_j^{n+1}\|^2\\
&\le\beta_2\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2+\frac{1}{4\beta_2\bar{\nu}_{min}}\|f_j^{n+1}\|_{-1}^2+\frac{\epsilon_2\nu'_{max}}{2}\|\nabla u_j^{n+1}\|^2+\frac{\nu'_{max}}{2\epsilon_2}\|\nabla\left(2u_j^n-u_j^{n-1}\right)\|^2\\
&\le\beta_2\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2+\frac{1}{4\beta_2\bar{\nu}_{min}}\|f_j^{n+1}\|_{-1}^2+\frac{\epsilon_2\nu'_{max}}{2}\|\nabla u_j^{n+1}\|^2+\frac{3\nu'_{max}}{\epsilon_2}\|\nabla u_j^n\|^2+\frac{3\nu'_{max}}{2\epsilon_2}\|\nabla u_j^{n-1}\|^2.
\end{aligned}
$$

As the last three terms all need to be bounded by $\bar{\nu}_{min}\|\nabla u_j^{n+1}\|^2$, we want to minimize $\frac{\epsilon_2}{2}+\frac{3}{\epsilon_2}+\frac{3}{2\epsilon_2}$ by taking $\epsilon_2=3$. The inequality (3.16) then reduces to

$$
\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2+|2q_j^{n+1}-q_j^n|^2\right)-\frac{1}{2\Delta t}\left(|q_j^n|^2+|2q_j^n-q_j^{n-1}|^2\right)+\frac{1}{2\Delta t}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2 \qquad (3.17)\\
&+\frac{\beta}{2}\Delta t\left(\|p_j^{n+1}\|^2+\|2p_j^{n+1}-p_j^n\|^2\right)-\frac{\beta}{2}\Delta t\left(\|p_j^n\|^2+\|2p_j^n-p_j^{n-1}\|^2\right)+\frac{\beta}{2}\Delta t\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|^2\\
&+\frac{\alpha}{4}h\left(\|\nabla u_j^{n+1}\|^2+\|2\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)-\frac{\alpha}{4}h\left(\|\nabla u_j^n\|^2+\|2\nabla u_j^n-\nabla u_j^{n-1}\|^2\right)\\
&+\frac{\alpha}{4}h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2\\
&+\left((1-\beta_2)\bar{\nu}_{min}-3\nu'_{max}\right)\|\nabla u_j^{n+1}\|^2+\frac{3}{2}\nu'_{max}\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^n\|^2\right)+\frac{1}{2}\nu'_{max}\left(\|\nabla u_j^n\|^2-\|\nabla u_j^{n-1}\|^2\right)\\
&\le\frac{1}{4\beta_2\bar{\nu}_{min}}\|f_j^{n+1}\|_{-1}^2.
\end{aligned}
$$

If the parameter condition is satisfied, then $\bar{\nu}_{min}-3\nu'_{max}>0$. Taking $\beta_2=\frac{1}{2}-\frac{3}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}}>0$, we have

$$
(1-\beta_2)\bar{\nu}_{min}-3\nu'_{max}=(\frac{1}{2}+\frac{3}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}})\bar{\nu}_{min}-3\nu'_{max}=\frac{1}{2}\left(\bar{\nu}_{min}-3\nu'_{max}\right)>0. \qquad (3.18)
$$

(3.17) can then be reduced to

$$
\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2+|2q_j^{n+1}-q_j^n|^2\right)-\frac{1}{2\Delta t}\left(|q_j^n|^2+|2q_j^n-q_j^{n-1}|^2\right)+\frac{1}{2\Delta t}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2 \qquad (3.19)\\
&+\frac{\beta}{2}\Delta t\left(\|p_j^{n+1}\|^2+\|2p_j^{n+1}-p_j^n\|^2\right)-\frac{\beta}{2}\Delta t\left(\|p_j^n\|^2+\|2p_j^n-p_j^{n-1}\|^2\right)+\frac{\beta}{2}\Delta t\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|^2\\
&+\frac{\alpha}{4}h\left(\|\nabla u_j^{n+1}\|^2+\|2\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)-\frac{\alpha}{4}h\left(\|\nabla u_j^n\|^2+\|2\nabla u_j^n-\nabla u_j^{n-1}\|^2\right)\\
&+\frac{\alpha}{4}h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2+\frac{3}{2}\nu'_{max}\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^n\|^2\right)+\frac{1}{2}\nu'_{max}\left(\|\nabla u_j^n\|^2-\|\nabla u_j^{n-1}\|^2\right)\\
&\le\frac{1}{2(\bar{\nu}_{min}-3\nu'_{max})}\|f_j^{n+1}\|_{-1}^2.
\end{aligned}
$$

Summing up from $n=2$ to $n=N-1$ and multiplying through by $2\Delta t$ gives

$$
\begin{aligned}
&|q_j^N|^2+|2q_j^N-q_j^{N-1}|^2+\sum_{n=2}^{N-1}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2+\beta\Delta t^2\left(\|p_j^N\|^2+\|2p_j^N-p_j^{N-1}\|^2\right)\\
&+\beta\Delta t^2\sum_{n=2}^{N-1}\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|^2+\frac{\alpha}{2}h\Delta t\left(\|\nabla u_j^N\|^2+\|2\nabla u_j^N-\nabla u_j^{N-1}\|^2\right)
\end{aligned}
$$

$$+\frac{\alpha}{2}h\Delta t\sum_{n=2}^{N-1}\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2+3\nu'_{max}\Delta t\|\nabla u_j^N\|^2+\nu'_{max}\Delta t\|\nabla u_j^{N-1}\|^2 \tag{3.20}$$
$$\le |q_j^2|^2+|2q_j^2-q_j^1|^2+\beta\Delta t^2\left(\|p_j^2\|^2+\|2p_j^2-p_j^1\|^2\right)+\frac{\alpha}{2}h\Delta t\left(\|\nabla u_j^2\|^2+\|2\nabla u_j^2-\nabla u_j^1\|^2\right)$$
$$+3\nu'_{max}\Delta t\|\nabla u_j^2\|^2+\nu'_{max}\Delta t\|\nabla u_j^1\|^2+\frac{\Delta t}{\bar{\nu}_{min}-3\nu'_{max}}\sum_{n=2}^{N-1}\|f_j^{n+1}\|_{-1}^2.$$

□

4. Implementation Algorithms. In this section we describe the implementation algorithms for fully decoupling u, p, and q in the proposed ensemble algorithms respectively, following the decoupling strategy in [31, 40, 41]. The advantages of the proposed ensemble methods in the aspect of numerical linear algebra is discussed then.

4.1. Implementation of AC-SAV-BE. We will introduce a new scalar S_j^{n+1} to decompose the numerical solution u_j^{n+1} into two parts yielding two sub-problems for the two components $\hat{u}_j^{n+1}$, $\breve{u}_j^{n+1}$ respectively, which do not contain S_j^{n+1}. A separate algebraic equation for S_j^{n+1} will be derived.

Let

$$S_j^{n+1}=\frac{q_j^{n+1}}{\sqrt{E(u_j^n)+\delta}},\qquad u_j^{n+1}=\hat{u}_j^{n+1}+S_j^{n+1}\breve{u}_j^{n+1}. \tag{4.1}$$

Substituting (4.1) into (2.7)-(2.9) and grouping the terms with S_j^{n+1} and those without S_j^{n+1}, we can derive two separate PDE systems for $\hat{u}_j^{n+1}$, $\breve{u}_j^{n+1}$. Then instead of solving (2.7)-(2.8), we solve the following two subproblems for $\hat{u}_j^{n+1}, \breve{u}_j^{n+1}$ respectively.

$$\begin{cases}\dfrac{1}{\Delta t}\hat{u}_j^{n+1}+\nabla\times(\bar{\nu}(\nabla\times\hat{u}_j^{n+1}))-\alpha h\Delta\hat{u}_j^{n+1}-\dfrac{1}{\beta}\nabla(\nabla\cdot\hat{u}_j^{n+1})\\ \quad=f_j^{n+1}+\dfrac{1}{\Delta t}u_j^n-\alpha h\Delta u_j^n-\nabla\times(\nu'_j(\nabla\times u_j^n))-\nabla p_j^n,\ \text{in }\Omega,\\ \hat{u}_j^{n+1}=g_j^{n+1},\ \text{on }\partial\Omega.\end{cases} \tag{AC-SAV-BE sub-problem 1}$$

$$\begin{cases}\dfrac{1}{\Delta t}\breve{u}_j^{n+1}+\nabla\times(\bar{\nu}(\nabla\times\breve{u}_j^{n+1}))-\alpha h\Delta\breve{u}_j^{n+1}-\dfrac{1}{\beta}\nabla(\nabla\cdot\breve{u}_j^{n+1})\\ \quad=-(u_j^n\cdot\nabla)u_j^n,\ \text{in }\Omega,\\ \breve{u}_j^{n+1}=0,\ \text{on }\partial\Omega.\end{cases} \tag{AC-SAV-BE sub-problem 2}$$

In both sub-problems, the coefficients for the unknown $\hat{u}_j^{n+1}$ or $\breve{u}_j^{n+1}$ are all constants and are independent of the ensemble index j. So after spatial discretization, the coefficient matrices for all realizations are the same, and do not change from one time step to another, i.e., the linear system is in the form of

$$A[x_1^{n+1},x_1^{n+1},\cdots,x_J^{n+1}]=[b_1^{n+1},b_2^{n+1},\cdots,b_J^{n+1}].$$

This is a linear system with multiple right hand sides, which has been well studied in the literature and efficient block solvers such as block GMRES can be used to compute it efficiently. We will show in our numerical tests in Section 5 that our ensemble algorithms are extremely fast and comparably accurate, compared with traditional methods that run each realization independently.

We also need to derive an equation for S_j^{n+1}.

$$S_j^{n+1}=\frac{q_j^{n+1}}{\sqrt{E(u_j^n)+\delta}}\qquad\Longrightarrow\qquad q_j^{n+1}=\sqrt{E(u_j^n)+\delta}\,S_j^{n+1}. \tag{4.2}$$

Plugging this expression of q_j^{n+1} into (3.3) gives

$$\begin{aligned}
&\frac{1}{\Delta t}(q_j^{n+1})^2 - \frac{1}{\Delta t}(q_j^n)^2 + \frac{1}{\Delta t}(q_j^{n+1} - q_j^n)^2 + \left(\bar{\nu}\nabla\cdot u_j^{n+1}, \nabla\cdot u_j^{n+1}\right) \\
&\qquad - \left(\frac{u_j^{n+1}-u_j^n}{\Delta t}, u_j^{n+1}\right) - S_j^{n+1}\int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot u_j^{n+1}\,dx + b_j^{n+1} = 0 \\
\Longrightarrow\ &\frac{1}{\Delta t}\left(\sqrt{E(u_j^n)+\delta}\, S_j^{n+1}\right)^2 - \frac{1}{\Delta t}(q_j^n)^2 + \frac{1}{\Delta t}\left(\sqrt{E(u_j^n)+\delta}\, S_j^{n+1} - q_j^n\right)^2 \\
&+ \left(\bar{\nu}\nabla\cdot(\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}), \nabla\cdot(\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1})\right) - \left(\frac{\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1} - u_j^n}{\Delta t}, \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}\right) \\
&- S_j^{n+1}\int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot(\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1})\,dx + b_j^{n+1} = 0.
\end{aligned}$$

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

$$A_j^{n+1}(S_j^{n+1})^2 + B_j^{n+1}S_j^{n+1} + C_j^{n+1} = 0, \qquad \text{(AC-SAV-BE sub-problem 3)}$$

where

$$\begin{aligned}
A_j^{n+1} &= \frac{2}{\Delta t}(E(u_j^n)+\delta) + \left(\bar{\nu}\nabla\cdot\breve{u}_j^{n+1}, \nabla\cdot\breve{u}_j^{n+1}\right) - \left(\frac{\breve{u}_j^{n+1}}{\Delta t}, \breve{u}_j^{n+1}\right) - \int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot\breve{u}_j^{n+1}\,dx, \\
B_j^{n+1} &= -\frac{2}{\Delta t}\sqrt{E(u_j^n)+\delta}\, q_j^n + 2\left(\bar{\nu}\nabla\cdot\hat{u}_j^{n+1}, \nabla\cdot\breve{u}_j^{n+1}\right) - \left(\frac{\breve{u}_j^{n+1}}{\Delta t}, \hat{u}_j^{n+1}\right) - \left(\frac{\hat{u}_j^{n+1}-u_j^n}{\Delta t}, \breve{u}_j^{n+1}\right) \\
&\quad - \int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot\hat{u}_j^{n+1}\,dx, \\
C_j^{n+1} &= \left(\bar{\nu}\nabla\cdot\hat{u}_j^{n+1}, \nabla\cdot\hat{u}_j^{n+1}\right) - \left(\frac{\hat{u}_j^{n+1}-u_j^n}{\Delta t}, \hat{u}_j^{n+1}\right) + b_j^{n+1}.
\end{aligned}$$

In general, 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-problem 1 and sub-problem 2, all realizations have the same constant coefficient matrix for all time steps and therefore can be solved very efficiently. In sub-problem 3, we need to solve each realization separately. But since it is a scalar quadratic equation, it can be solved quickly. After getting $\hat{u}_j^{n+1}, \breve{u}_j^{n+1}$, and S_j^{n+1}, we have $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}$. The pressure p_j^{n+1} can be updated directly without solving a linear system using the formula

$$p_j^{n+1} = p_j^n - \frac{1}{\beta}\nabla\cdot u_j^{n+1}.$$

4.2. Implementation of AC-SAV-BDF2. Similarly, we introduce an intermediate variable S_j^{n+1} and derive three sub-problems for efficient implementation of the AC-SAV-BDF2 ensemble algorithm. Let

$$S_j^{n+1} = \frac{q_j^{n+1}}{\sqrt{E(\tilde{u}_j^{n+1})+\delta}}, \qquad u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}. \tag{4.3}$$

Then instead of solving (2.10)-(2.12), we solve the following two sub-problems for $\hat{u}_j^{n+1}$ and $\breve{u}_j^{n+1}$ respectively.

$$\begin{cases}
\dfrac{3}{2\Delta t}\hat{u}_j^{n+1} + \nabla\times(\bar{\nu}(\nabla\times\hat{u}_j^{n+1})) - \dfrac{3}{2}\alpha h\Delta\hat{u}_j^{n+1} - \dfrac{1}{3\beta\Delta t}\nabla(\nabla\cdot\hat{u}_j^{n+1}) \\
\qquad = f_j^{n+1} + \dfrac{2}{\Delta t}u_j^n - \dfrac{1}{2\Delta t}u_j^{n-1} - \nabla\times(\nu_j'(\nabla\times\tilde{u}_j^{n+1})) \\
\qquad - 2\alpha h\Delta u_j^n + \dfrac{1}{2}\alpha h\Delta u_j^{n-1} - \nabla(\dfrac{4}{3}p_j^n - \dfrac{1}{3}p_j^{n-1}),\ \text{in } \Omega, \\
\hat{u}_j^{n+1} = g_j^{n+1},\ \text{on } \partial\Omega,
\end{cases} \qquad \text{(AC-SAV-BDF2 sub-problem 1)}$$

$$\begin{cases} \dfrac{3}{2\Delta t}\breve{u}_j^{n+1} + \nabla \times (\bar{\nu}(\nabla \times \breve{u}_j^{n+1})) - \dfrac{3}{2}\alpha h \Delta \breve{u}_j^{n+1} - \dfrac{1}{3\beta\Delta t}\nabla(\nabla \cdot \breve{u}_j^{n+1}) \\ \qquad = -(\tilde{u}_j^{n+1} \cdot \nabla)\tilde{u}_j^{n+1}, \text{ in } \Omega, \\ \breve{u}_j^{n+1} = 0, \text{ on } \partial\Omega. \end{cases} \tag{AC-SAV-BDF2 sub-problem 2}$$

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

$$S_j^{n+1} = \frac{q_j^{n+1}}{\sqrt{E(\tilde{u}_j^{n+1}) + \delta}} \qquad \Longrightarrow \qquad q_j^{n+1} = \sqrt{E(\tilde{u}_j^{n+1}) + \delta}\, S_j^{n+1}. \tag{4.4}$$

Multiplying (2.12) with $2q_j^{n+1}$ and then plugging in (4.4) gives

$$\begin{aligned} &\frac{3}{\Delta t}(q_j^{n+1})^2 + \frac{-4q_j^n + q_j^{n-1}}{\Delta t} q_j^{n+1} + \left(\bar{\nu}\nabla \cdot u_j^{n+1}, \nabla \cdot u_j^{n+1}\right) - \left(\frac{3u_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t}, u_j^{n+1}\right) \\ &\quad - S_j^{n+1}\int_\Omega (\tilde{u}_j^{n+1} \cdot \nabla)\tilde{u}_j^{n+1} \cdot u_j^{n+1}\, dx + b_j^{n+1} = 0 \\ \Longrightarrow\ &\frac{3}{\Delta t}(E(\tilde{u}_j^{n+1}) + \delta)(S_j^{n+1})^2 + \frac{-4q_j^n + q_j^{n-1}}{\Delta t}\sqrt{E(\tilde{u}_j^{n+1}) + \delta}\, S_j^{n+1} \\ &\quad + \left(\bar{\nu}\nabla \cdot (\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}), \nabla \cdot (\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1})\right) \\ &\quad - \left(\frac{3(\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}) - 4u_j^n + u_j^{n-1}}{2\Delta t}, \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}\right) \\ &\quad - S_j^{n+1}\int_\Omega (\tilde{u}_j^{n+1} \cdot \nabla)\tilde{u}_j^{n+1} \cdot (\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1})\, dx + b_j^{n+1} = 0. \end{aligned}$$

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

$$A_j^{n+1}(S_j^{n+1})^2 + B_j^{n+1}S_j^{n+1} + C_j^{n+1} = 0, \tag{AC-SAV-BDF2 sub-problem 3}$$

where

$$\begin{aligned} A_j^{n+1} &= \frac{3}{\Delta t}(E(\tilde{u}_j^{n+1}) + \delta) + \left(\bar{\nu}\nabla \cdot \breve{u}_j^{n+1}, \nabla \cdot \breve{u}_j^{n+1}\right) - \left(\frac{3\breve{u}_j^{n+1}}{2\Delta t}, \breve{u}_j^{n+1}\right) - \int_\Omega (\tilde{u}_j^{n+1} \cdot \nabla)\tilde{u}_j^{n+1} \cdot \breve{u}_j^{n+1}\, dx, \\ B_j^{n+1} &= \frac{-4q_j^n + q_j^{n-1}}{\Delta t}\sqrt{E(\tilde{u}_j^{n+1}) + \delta} + 2\left(\bar{\nu}\nabla \cdot \hat{u}_j^{n+1}, \nabla \cdot \breve{u}_j^{n+1}\right) - \left(\frac{3\breve{u}_j^{n+1}}{2\Delta t}, \hat{u}_j^{n+1}\right) \\ &\quad - \left(\frac{3\hat{u}_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t}, \breve{u}_j^{n+1}\right) - \int_\Omega (\tilde{u}_j^{n+1} \cdot \nabla)\tilde{u}_j^{n+1} \cdot \hat{u}_j^{n+1}\, dx, \\ C_j^{n+1} &= \left(\bar{\nu}\nabla \cdot \hat{u}_j^{n+1}, \nabla \cdot \hat{u}_j^{n+1}\right) - \left(\frac{3\hat{u}_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t}, \hat{u}_j^{n+1}\right) + b_j^{n+1}. \end{aligned}$$

After getting u_j^{n+1} by $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}$, the pressure can be updated directly as follows.

$$p_j^{n+1} = \frac{1}{3}(4p_j^n - p_j^{n-1}) - \frac{1}{3\beta\Delta t}\nabla \cdot u_j^{n+1}.$$

4.3. The Algebraic Systems and Linear Solvers. Denote the finite element basis functions for the velocity u and the pressure p by $\{\chi_j^u\}_{j=1}^{N_u}$, $\{\chi_j^p\}_{j=1}^{N_p}$ respectively. We then define matrices $\mathbf{M}_{uu}$, $\mathbf{D}_{uu}$, $\mathbf{D}_{uup}$, $\mathbf{S}_{uu}$, $\mathbf{C}(\nu)$, and $\mathbf{N}(u)$ whose entries are given as follows.

$$[\mathbf{M}_{uu}]_{kl} = \int_\Omega \chi_l^u \cdot \chi_k^u, \quad [\mathbf{D}_{uu}]_{kl} = \int_\Omega (\nabla \cdot \chi_l^u)(\nabla \cdot \chi_k^u), \quad [\mathbf{D}_{uup}]_{kl} = \int_\Omega \chi_l^p(\nabla \cdot \chi_k^u),$$

$$[\mathbf{S}_{uu}]_{kl} = \int_\Omega \nabla\chi_l^u \cdot \nabla\chi_k^u, \quad [\mathbf{C}(\nu)]_{kl} = \int_\Omega \nu(\nabla\times\chi_l^u)\cdot(\nabla\times\chi_k^u), \quad [\mathbf{N}(u)]_{kl} = \int_\Omega (u\cdot\nabla)\chi_l^u\cdot\chi_k^u.$$

In the next section, we will perform several efficiency tests by comparing the AC-SAV-BE and AC-SAV-BDF2 ensemble schemes with the ones not using the AC technique, namely the SAV-BE and SAV-BDF2 ensemble schemes [31]. Below we list the coefficient matrices of the algebraic systems corresponding to above schemes and the standard nonensemble ones, for the realization of sample $j = 1, \cdots, J$.

(1) AC-SAV-BE ensemble:

$$\mathbf{A}_{\text{as1e}} = \tfrac{1}{\Delta t}\mathbf{M}_{uu} + \mathbf{C}(\bar{\nu}) + \alpha h\mathbf{S}_{uu} + \tfrac{1}{\beta}\mathbf{D}_{uu}.$$

(2) SAV-BE ensemble:

$$\mathbf{A}_{\text{s1e}} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}_{uu} + \mathbf{C}(\bar{\nu}) + \alpha h\mathbf{S}_{uu} & -\mathbf{D}_{uup} \\ -\mathbf{D}_{uup}^T & \mathbf{0} \end{pmatrix}.$$

(3) BE nonensemble:

$$\mathbf{A}_{\text{1n}}^{(j,n)} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}_{uu} + \mathbf{N}(u_j^n) + \mathbf{C}(\nu_j) & -\mathbf{D}_{uup} \\ -\mathbf{D}_{uup}^T & \mathbf{0} \end{pmatrix}.$$

(4) AC-SAV-BDF2 ensemble:

$$\mathbf{A}_{\text{as2e}} = \tfrac{3}{2\Delta t}\mathbf{M}_{uu} + \mathbf{C}(\bar{\nu}) + \frac{3}{2}\alpha h\mathbf{S}_{uu} + \tfrac{1}{3\beta\Delta t}\mathbf{D}_{uu}.$$

(5) SAV-BDF2 ensemble:

$$\mathbf{A}_{\text{s2e}} = \begin{pmatrix} \frac{3}{2\Delta t}\mathbf{M}_{uu} + \mathbf{C}(\bar{\nu}) + \frac{3}{2}\alpha h\mathbf{S}_{uu} & -\mathbf{D}_{uup} \\ -\mathbf{D}_{uup}^T & \mathbf{0} \end{pmatrix}.$$

(6) BDF2 nonensemble:

$$\mathbf{A}_{\text{2n}}^{(j,n)} = \begin{pmatrix} \frac{3}{2\Delta t}\mathbf{M}_{uu} + \mathbf{N}(\tilde{u}_j^{n+1}) + \mathbf{C}(\nu_j) & -\mathbf{D}_{uup} \\ -\mathbf{D}_{uup}^T & \mathbf{0} \end{pmatrix}.$$

One should notice that the matrices $\mathbf{A}_{\text{as1e}}$ and $\mathbf{A}_{\text{as2e}}$ are not only in reduced size, as compared with $\mathbf{A}_{\text{s1e}}$ and $\mathbf{A}_{\text{s2e}}$, but also symmetric and positive definite (SPD), so the conjugate gradient (CG) iterative linear solver can be applied. When compared to $\mathbf{A}_{\text{1n}}^{(j,n)}$ and $\mathbf{A}_{\text{2n}}^{(j,n)}$, the matrices $\mathbf{A}_{\text{as1e}}$ and $\mathbf{A}_{\text{as2e}}$ are fixed not only for different samples but also for different time steps, so we can simultaneously achieve all realizations by solving a single linear system with multiple right hand sides (RHSs) corresponding to different samples. Computational efficiency is then guaranteed as redundant information due to linear dependence of multiple residuals can be removed beforehand. In contrast, $\mathbf{A}_{\text{1n}}^{(j,n)}$ and $\mathbf{A}_{\text{2n}}^{(j,n)}$ in the nonensemble methods change over sample index j and time step index n, hence we need to simulate J samples one by one at each time step.

In application for large-scale simulations especially for 3D problems, one can use the block CG method [25, 44, 46] or CG method with vectorization operation, preconditioned by multigrid, to handle the systems associated with $\mathbf{A}_{\text{as1e}}$ and $\mathbf{A}_{\text{as2e}}$ in the AC-SAV ensemble schemes. As for the SAV-BDF2 ensemble scheme, the block GMRES method [3, 14] with the least-squares commutator preconditioning was claimed having good performance [31]. So we keep using this solver for the SAV-BE and SAV-BDF2 ensemble schemes in the efficiency comparison tests to appear. In particular, the preconditioner inside block GMRES is solved by the block CG algorithm with a multigrid preconditioner.

5. Numerical Experiments.

5.1. Tests for convergence rate. To validate the temporal convergence rate of AC-SAV-BE and AC-SAV-BDF2 ensemble schemes, we consider a simple test problem [15] with Green-Taylor vortex solution on a square domain $\Omega = (0,1)^2$. The analytic solution and forcing term are given by

$$\begin{aligned} u &= (-\cos x \sin y, \sin x \cos y)^T g(t), \\ p &= -\frac{1}{4}[\cos(2x) + \cos(2y)] g(t)^2, \\ f(x,y,t) &= [g'(t) + 2\nu g(t)](-\cos x \sin y, \sin x \cos y)^T, \end{aligned}$$

with $g(t) = e^{\nu}\cos(2t)$. Initial and Dirichlet boundary conditions are set to match the prescribed analytic solution. We consider an ensemble computation of J solutions corresponding to

$$\nu_j = \nu_{min}(1+\epsilon_j),\ j = 1, \cdots, J.$$

In this setup, the J realizations have different initial conditions, boundary conditions, and body forces. For efficiency in computation on fine mesh, we only set $J = 3,\ \epsilon_1 = 0,\ \epsilon_2 = 0.2/3,\ \epsilon_3 = 0.4/3$, while much larger J also works.

We will evaluate numerical errors of the velocity in H^1 semi-norm and the pressure in L^2 norm at $T = 2$, computed on successively refined meshs with $h = 1/16, \cdots, 1/128$. Taking $\Delta t = 0.01 * h$, the expected numerical errors of u and p by the AC-SAV-BE ensemble method are both $O(h^2 + \Delta t) = O(h)$. In this test, we set $\alpha = 0.5$, $\beta = 1$. The simulation errors and convergence rates for the first and third samples are reported in Table 5.1 and Table 5.2, for $\nu_{min} = 0.01$ and $\nu_{min} = 0.001$ respectively. As we can see, the AC-SAV-BE ensemble scheme is first-order convergent in time as predicted.

Taking $\Delta t = h$, the expected numerical errors of u and p by the AC-SAV-BDF2 ensemble method are both $O(h^2 + \Delta t^2) = O(\Delta t^2)$. Still setting $\alpha = 0.5$, $\beta = 1$, we report the corresponding results in Table 5.3 and Table 5.4 for $\nu_{min} = 0.01$ and $\nu_{min} = 0.001$ respectively. As predicted, the AC-SAV-BDF2 ensemble scheme has second-order convergence rate in time.

Table 5.1: Errors at $T = 2$ and convergence rates of the AC-SAV-BE ensemble algorithm ($J = 3$) with $\Delta t = 0.01 * h$, $\alpha = 0.5$, $\beta = 1$, $\nu_{min} = 0.01$.

h	$\lvert u_h - u\rvert^{E,1}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,1}_{L^2}$	Rate	$\lvert u_h - u\rvert^{E,3}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,3}_{L^2}$	Rate
1/16	1.32×10^{-3}	—	2.81×10^{-4}	—	1.26×10^{-3}	—	2.83×10^{-4}	—
1/32	6.52×10^{-4}	1.02	1.22×10^{-4}	1.21	6.22×10^{-4}	1.02	1.23×10^{-4}	1.20
1/64	3.26×10^{-4}	1.00	5.66×10^{-5}	1.11	3.11×10^{-4}	1.00	5.70×10^{-5}	1.11
1/128	1.63×10^{-4}	1.00	2.73×10^{-5}	1.05	1.55×10^{-4}	1.00	2.74×10^{-5}	1.05

Table 5.2: Errors at $T = 2$ and convergence rates of the AC-SAV-BE ensemble algorithm ($J = 3$) with $\Delta t = 0.01 * h$, $\alpha = 0.5$, $\beta = 1$, $\nu_{min} = 0.001$.

h	$\lvert u_h - u\rvert^{E,1}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,1}_{L^2}$	Rate	$\lvert u_h - u\rvert^{E,3}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,3}_{L^2}$	Rate
1/16	3.04×10^{-3}	—	2.76×10^{-4}	—	2.92×10^{-3}	—	2.76×10^{-4}	—
1/32	1.53×10^{-3}	0.99	1.19×10^{-4}	1.21	1.46×10^{-3}	1.00	1.19×10^{-4}	1.21
1/64	7.72×10^{-4}	0.99	5.54×10^{-5}	1.11	7.34×10^{-4}	0.99	5.54×10^{-5}	1.11
1/128	3.86×10^{-4}	1.00	2.67×10^{-5}	1.05	3.67×10^{-4}	1.00	2.67×10^{-5}	1.05

5.2. Efficiency tests. We then test the AC efficiency and ensemble efficiency of AC-SAV-BE and AC-SAV-BDF2 schemes using the analytic solution stated in Section 5.1. The values of ν_j are set by taking

$$\nu_j = \nu_{min}(1+\epsilon_j),\ \epsilon_j = 0.2(j-1)/J,\ j = 1, \cdots, J.$$

Table 5.3: Errors at $T = 2$ and convergence rates of the AC-SAV-BDF2 ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0.5$, $\beta = 1$, $\nu_{min} = 0.01$.

Δt	$\lvert u_h - u\rvert^{E,1}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,1}_{L^2}$	Rate	$\lvert u_h - u\rvert^{E,3}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,3}_{L^2}$	Rate
1/16	1.28×10^{-2}	—	3.32×10^{-3}	—	1.13×10^{-2}	—	3.34×10^{-3}	—
1/32	2.75×10^{-3}	2.22	8.12×10^{-4}	2.03	2.50×10^{-3}	2.18	8.17×10^{-4}	2.03
1/64	6.78×10^{-4}	2.02	2.01×10^{-4}	2.01	6.21×10^{-4}	2.01	2.02×10^{-4}	2.01
1/128	1.69×10^{-4}	2.00	5.00×10^{-5}	2.01	1.56×10^{-4}	2.00	5.03×10^{-5}	2.01

Table 5.4: Errors at $T = 2$ and convergence rates of the AC-SAV-BDF2 ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0.5$, $\beta = 1$, $\nu_{min} = 0.001$.

Δt	$\lvert u_h - u\rvert^{E,1}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,1}_{L^2}$	Rate	$\lvert u_h - u\rvert^{E,3}_{H^1}$	Rate	$\lvert p_h - p\rvert^{E,3}_{L^2}$	Rate
1/16	5.70×10^{-2}	—	3.22×10^{-3}	—	5.50×10^{-2}	—	3.22×10^{-3}	—
1/32	1.73×10^{-2}	1.72	7.80×10^{-4}	2.04	1.58×10^{-2}	1.80	7.80×10^{-4}	2.04
1/64	3.13×10^{-3}	2.47	1.91×10^{-4}	2.03	2.86×10^{-3}	2.47	1.92×10^{-4}	2.03
1/128	6.85×10^{-4}	2.19	4.75×10^{-5}	2.01	6.39×10^{-4}	2.16	4.75×10^{-5}	2.01

The space and time resolutions are fixed as $h = \Delta t = 1/64$ and the final time is taken at $T = 10$.

First of all, we compare the performance of the first-order methods, AC-SAV-BE ($\alpha = 1, \beta = 1$) and SAV-BE ($\alpha = 1$), on simulating a single flow, i.e. $J = 1$. This is to show the advantage of using AC technique. Extensive testing is performed by considering various ν_{min} values: $\nu_{min} = 0.1, 0.01, 0.001$. In the efficiency tests, we use the block CG linear solver with the multi-grid preconditioner for the AC-SAV-BE method. As for the SAV-BE scheme, the block GMRES solver with the least-squares commutator preconditioner, which is tested to be efficienct in [31], is taken to handle the algebraic linear systems. Our MATLAB implementation is based on the data structure of iFEM package.

Table 5.5 reports the execution times and solution errors computed by the first-order schemes. It is observed that the AC-SAV-BE scheme outperforms the SAV-BE scheme as it takes much less CPU time but maintains similar accuracy, thanks to the splitting of velocity and pressure by the AC technique. Note that the reason why the reported CPU time for SAV-BE decreases as ν_{min} decreases is that the least-squares commutator preconditioner is more efficient for smaller viscosity, as reported in Table 1 of [31].

Next, we study the ensemble efficiency by varing the number J of realizations from 1 to 100. The execution times of simulations with $J = 1, 10, 100$ by the AC-SAV-BE and AC-SAV-BDF2 ensemble schemes are plotted in Figure 5.1. There, the red dash line is a reference for linear increase of execution time with respect to the sample size J. As we can see, the advantage of the two ensemble algorithms is apparent as the ensemble size increases, the execution time is significantly reduced as compared to individual simulations of J flows. This is because all the realizations in the ensemble method share a common matrix, hence all the J realizations can be achieved simutaneously.

5.3. Long time stability on a double driven cavity flow. We then simulate the two-dimensional driven cavity flow [2], a classical benchmark problem, to test the AC-SAV-BDF2 scheme. To be specific, the flow in the square $\Omega = (0, 1)^2$ is driven by two sides of the boundary: $(u_1, u_2) = (1, 0)$ on $y = 1$ and $(u_1, u_2) = (0, -1)$ on $x = 0$. No-slip boundary condition is imposed on the other parts of the boundary. The initial velocity and external body force are set to be zero. Taking $\nu = 10^{-3}$ and then $\nu = 10^{-4}$, we run simulations until $T = 60$ to study the long time stability of the proposed AC-SAV-BDF2 scheme.

Our numerical experiments are performed with $h = 1/64$, $\Delta t = 0.01, 0.05, 0.002$. The time histories of energy and S_j^n computed with $\alpha = 1, \beta = 1$ in the AC-SAV-BDF2 scheme are plotted in Figure 5.2 for the case $\nu = 10^{-3}$. Notice that S_j^n does converge to one when the simulation is performed using relatively large time steps. Similar results are also obtained for the case $\nu = 10^{-4}$, as shown in Figure 5.3. Figure 5.2 and 5.3 illustrate the long time stability of the AC-SAV-BDF2 scheme. We also state that the AC-SAV-BDF2

Table 5.5: Execution times and solution errors at final time $T = 10$ computed by the first-order schemes with $h = \Delta t = 1/64, J = 1$.

	AC-SAV-BE ($\alpha = 1, \beta = 1$)			SAV-BE ($\alpha = 1$)		
ν_{min}	$\lvert\mathbb{E}[u_h - u]\rvert_{H^1}$	$\lvert\mathbb{E}[p_h - p]\rvert_{L^2}$	Exe time	$\lvert\mathbb{E}[u_h - u]\rvert_{H^1}$	$\lvert\mathbb{E}[p_h - p]\rvert_{L^2}$	Exe time
0.1	9.728×10^{-3}	4.442×10^{-3}	755 s	6.656×10^{-3}	4.948×10^{-3}	19339 s
0.01	2.980×10^{-2}	2.984×10^{-3}	1014 s	3.105×10^{-2}	3.903×10^{-3}	7355 s
0.001	1.246×10^{-1}	3.003×10^{-3}	1108 s	1.124×10^{-1}	3.961×10^{-3}	5736 s

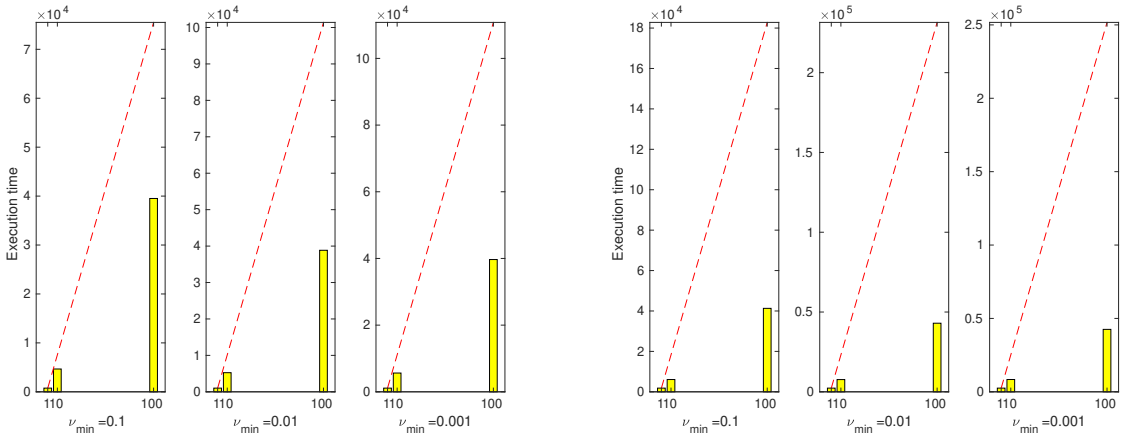


Fig. 5.1: Execution times of simulations with $J = 1, 10, 100$. The red dash line is a reference for linear increase of execution time with respect to the sample size J. The simulations are performed with $T = 10$, $h = \Delta t = 1/64$, $\beta = 1$.

scheme has very loose time step constraints for convergence. For both $\nu = 10^{-3}$ and $\nu = 10^{-4}$, a large $\Delta t = 0.01$ with $\alpha = 1, \beta = 1$ is good enough for stable and reliable simulations.

The velocity fields of the double driven cavity flow at $t = 60$ computed with $\nu = 10^{-3}$ and $\nu = 10^{-4}$ are also plotted in the left and right of Figure 5.4, respectively.

5.4. Flow past cylinder. Our final test is on the two-dimensional flow past cylinder, a classical benchmark problem introduced in Schäfer and Turek [50]. This problem has been widely used, in [30, 37] for instance, to study the stability or effectiveness of certain time stepping methods. In our work, the aim is to show that the AC-SAV-BDF2 scheme produces reasonable simulations even with large time steps if appropriate stabilization and AC parameters are chosen.

Consider the flow in a 2.2×0.41 rectangular channel around a cylinder of radius 0.05 centered at $(0.2, 0.2)$. No-slip boundary conditions are imposed on the cylinder, also the top and bottom of the channel, while the inflow/outflow boundary conditions are prescribed as

$$u_1(0, y) = u_1(2.2, y) = \frac{6}{0.41^2} \sin(\pi t/8) y(0.41 - y),$$
$$u_2(0, y) = u_2(2.2, y) = 0.$$

The initial velocity and external force are set to zero. The viscosity is $\nu = 10^{-3}$. Based on the inflow profile and the cylinder diameter $L = 0.1$, the Reynolds number is $Re = 100$. For this value of Re, the problem features a laminar flow, with a Kármán vortex street developing behind the cylinder. In particular, the eddy becomes unstable from t = 2, it is then shed on alternate sides of the cylinder between t = 4 and t = 6.

The numerical solutions are computed with Taylor-Hood elements holding 63920 number of degrees of freedom for velocity and 16155 for pressure. The spatial resolution ranges from 0.0030 to 0.0147. Simulations

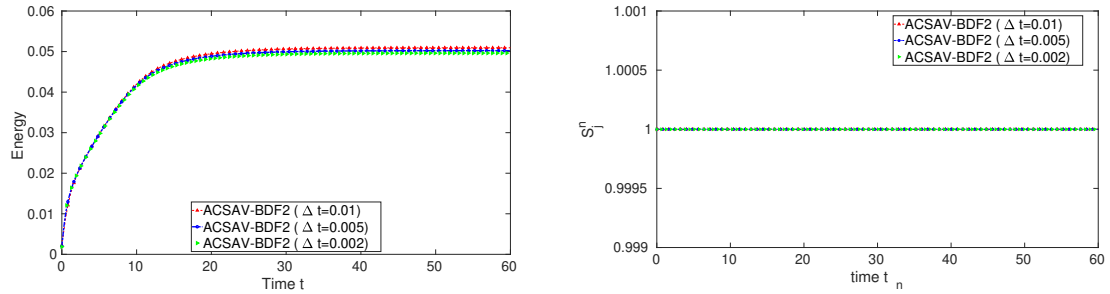


Fig. 5.2: Double driven cavity flow: time histories of energy and S_j^n computed by the AC-SAV-BDF2 scheme with $\nu = 10^{-3}, h = 1/64, T = 60, J = 1 = j$.

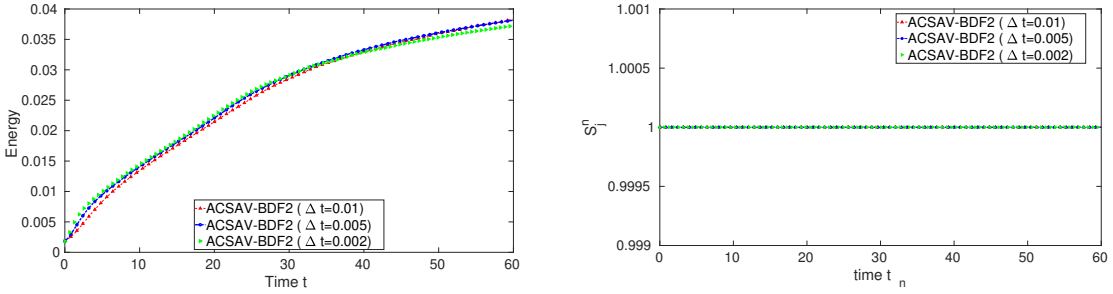


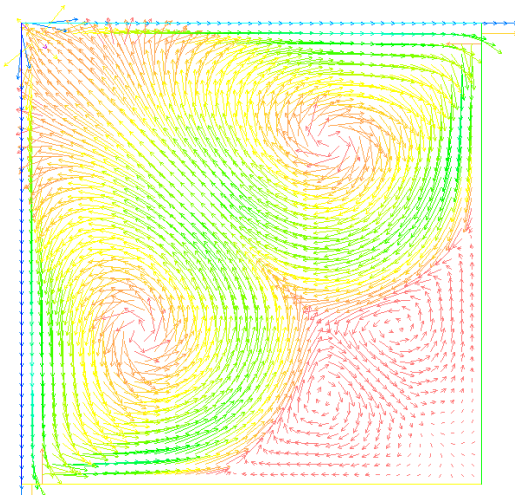
Fig. 5.3: Double driven cavity flow: time histories of energy and S_j^n computed by the AC-SAV-BDF2 scheme with $\nu = 10^{-4}, h = 1/64, T = 60, J = 1 = j$.

are performed with $\Delta t = 0.01, 0.005, 0.002$, while the stabilization and AC parameters are fixed as $\alpha = 4, \beta = 1$. The corresponding time histories of energy and S_j^n are plotted in Figure 5.5. Again, we observe that S_j^n does converge to one when the simulation is performed using relatively large time steps. For instance, a large $\Delta t = 0.01$ is good enough for stable and reliable simulations.

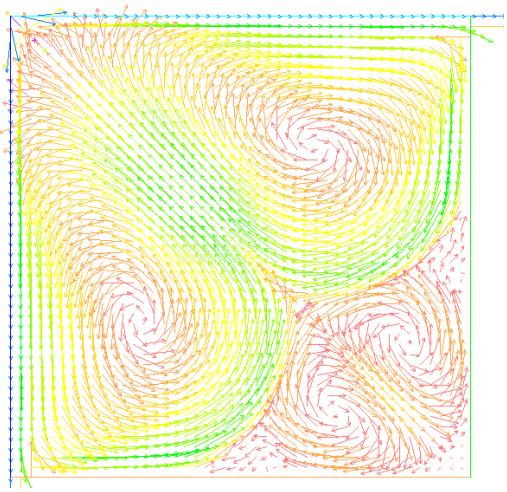
The velocity fields and velocity magnitudes of the flow at $t = 4, 5, 6, 7, 8$ computed by the AC-SAV-BDF2 method with $\Delta t = 0.01$ are plotted in Figure 5.6. Figure 5.7 is then for $\Delta t = 0.002$. In all plots, the simulation results are satisfactory: the AC-SAV-BDF2 method is stable and the flow patterns produced match with those in [30, 33].

6. Conclusions. We have presented two extremely fast ensemble simulation algorithms well-suited for UQ applications. The algorithms are designed based on recently developed ensemble timestepping method, the SAV approach, and the AC technique, resulting in linear systems that have the same constant coefficient matrix so that highly efficient block linear solvers can be used to significantly reduce the computational cost and simulation time. In particular the AC technique decouples the computation of the velocity and pressure reducing the size of the linear systems to be solved at each time step. Moreover the pressure can be updated directly without solving a Poisson equation avoiding the boundary layer in pressure errors due to artificial boundary condition. We proved both algorithm are long time stable without any timestep constraints. Extensive numerical experiments were performed to demonstrate that our ensemble algorithms are highly efficient and competitively accurate compared with traditional methods that run each realization independently.

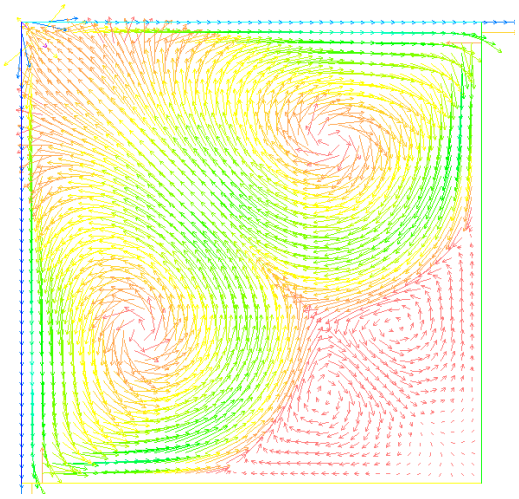
Funding. Nan Jiang was partially supported by the US National Science Foundation grants DMS-1720001 and DMS-2120413. Huanhuan Yang was supported in part by the National Natural Science Foundation of China under grant 11801348, the key research projects of general universities in Guangdong Province (grant 019KZDXM034), and the basic research and applied basic research projects in Guangdong Province (Projects of Guangdong, Hong Kong and Macao Center for Applied Mathematics, grant 2020B1515310018).



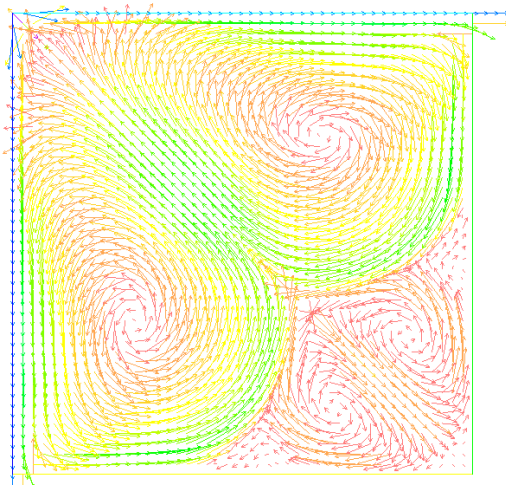
$\nu = 10^{-3},\ \Delta t = 0.01$



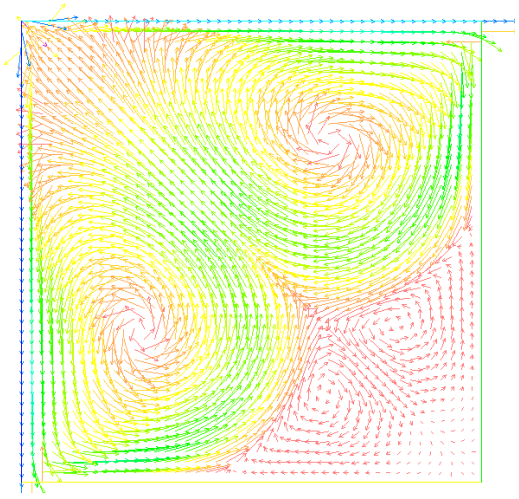
$\nu = 10^{-4},\ \Delta t = 0.01$



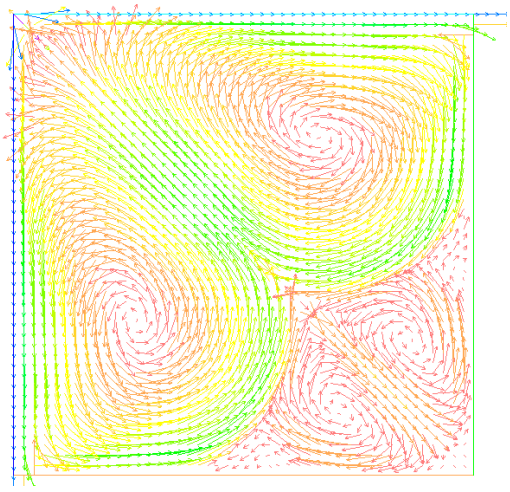
$\nu = 10^{-3},\ \Delta t = 0.005$



$\nu = 10^{-4},\ \Delta t = 0.005$



$\nu = 10^{-3},\ \Delta t = 0.002$



$\nu = 10^{-4},\ \Delta t = 0.002$

Fig. 5.4: Velocity fields of the double driven cavity flow at $T = 60$, computed by the AC-SAV-BDF2 scheme with $h = 1/64, J = 1$. Left: $\nu = 10^{-3}$; right: $\nu = 10^{-4}$. From top to bottom: $\Delta t = 0.01, 0.005, 0.002$.

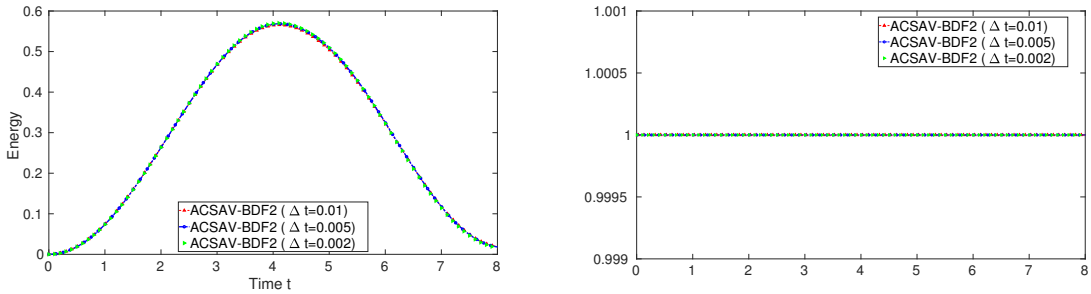


Fig. 5.5: Flow past cylinder: time histories of energy and S_j^n computed by the AC-SAV-BDF2 scheme with $\nu = 10^{-3}, h_{max} = 0.0147, T = 8, J = 1 = j$.

REFERENCES

[1] I. Babuška, F. Nobile and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007), 1005-1034.

[2] M. Ben-Artzi, J.-P. Croisille, and D. Fishelov, *Navier-stokes equations in planar domains*, Imperial College Press, 2013, London.

[3] 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.

[4] R.M. Chen, W. Layton and M. McLaughlin, *Analysis of variable-step/non-autonomous artificial compression methods*, Journal of Mathematical Fluid Mechanics, 21 (2019), 30.

[5] A.J. Chorin, *A numerical method for solving incompressible viscous flow problems*, Journal of Computational Physics, 2 (1967), pp. 12-26.

[6] J. Connors, *An ensemble-based conventional turbulence model for fluid-fluid interaction*, International Journal of Numerical Analysis and Modeling, 15 (2018), 492-519.

[7] V. DeCaria, T. Iliescu, W. Layton, M. McLaughlin and M. Schneier, *An artificial compression reduced order model*, SIAM Journal on Numerical Analysis, 58 (2020), pp. 565-589.

[8] V. DeCaria, W. Layton and M. McLaughlin, *A conservative, second order, unconditionally stable artificial compression method*, Computer Methods in Applied Mechanics and Engineering, 325 (2017), pp. 733-747.

[9] V. DeCaria, W. Layton and M. McLaughlin, *An analysis of the Robert–Asselin time filter for the correction of nonphysical acoustics in an artificial compression method*, Numerical Methods for Partial Differential Equations, 35 (2019), pp. 916-935.

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

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

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

[13] J. Fiordilino and M. McLaughlin, *An artificial compressibility ensemble timestepping algorithm for flow problems*, arXiv, 2017, https://arxiv.org/abs/1712.06271.

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

[15] J.-L. Guermond and L. Quartapelle, *On stability and convergence of projection methods based on pressure Poisson equation*, Int. J. Numer. Methods Fluids, 26 (1998), 1039-1053.

[16] 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.

[17] 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.

[18] 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.

[19] J. Guermond and P. Minev, *High-order time stepping for the incompressible Navier-Stokes equations*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2656-A2681.

[20] J. Guermond and P. Minev, *High-order time stepping for the Navier-Stokes equations with minimal computational complexity*, Journal of Computational and Applied Mathematics, 310 (2017), pp. 92-103.

[21] J. Guermond and P. Minev, *High-order adaptive time stepping for the incompressible Navier-Stokes equations*, SIAM Journal on Scientific Computing, 41 (2019), pp. A770-A788.

[22] M. Gunzburger, C. Webster and G. Zhang, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numerica, 23 (2014), 521-650.

[23] 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,

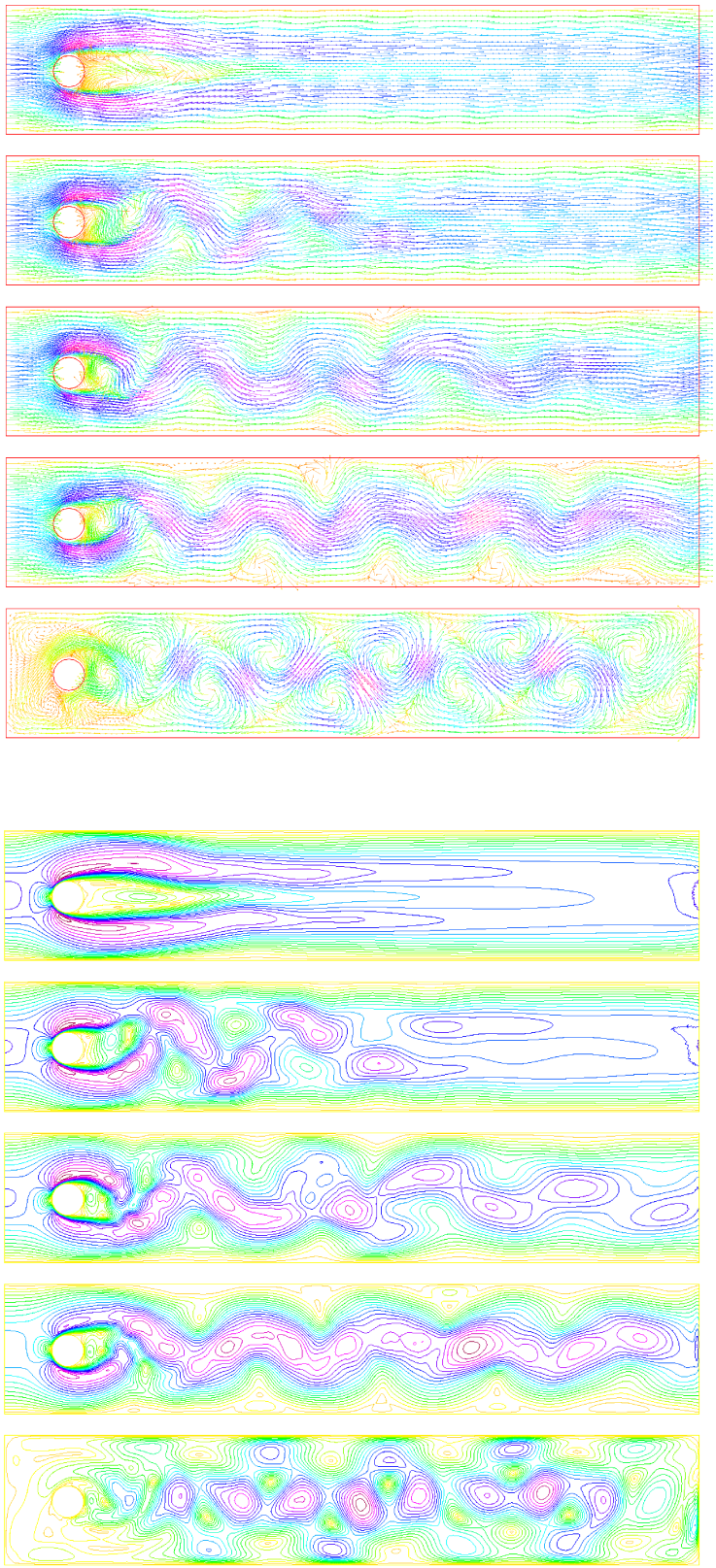


Fig. 5.6: Velocity fields and velocity magnitudes of the flow past cylinder at $t = 4, 5, 6, 7, 8$, computed by the AC-SAV-BDF2 scheme with $\Delta t = 0.01$.

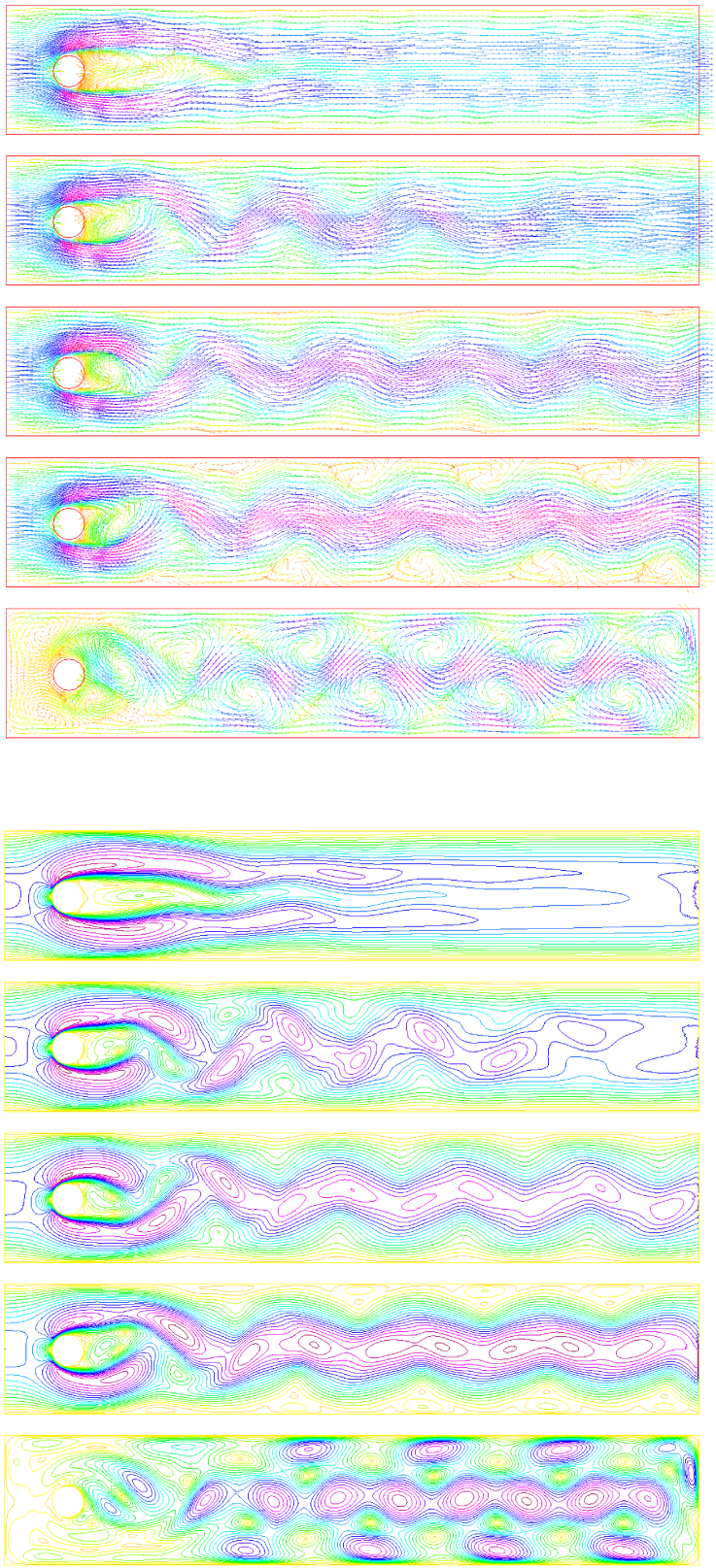


Fig. 5.7: Velocity fields and velocity magnitudes of the flow past cylinder at $t = 4, 5, 6, 7, 8$, computed by the AC-SAV-BDF2 scheme with $\Delta t = 0.002$.

121 (2020), 712-739.
[24] S. Hosder, R. Walters and R. Perez, *A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations, AIAA-Paper 2006-891*, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2006, CD-ROM.
[25] H. Ji and Y. Li, *A breakdown-free block conjugate gradient method*, BIT Numerical Mathematics, 57(2) (2017), 379-403.
[26] 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.
[27] N. Jiang and W. Layton, *An algorithm for fast calculation of flow ensembles*, International Journal for Uncertainty Quantification, 4 (2014), 273-301.
[28] 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.
[29] 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.
[30] N. Jiang and H. Tran, *Analysis of a stabilized CNLF method with fast slow wave splittings for flow problems*, Computational Methods in Applied Mathematics, 15 (2015), 307–330.
[31] N. Jiang and H. Yang, *Stabilized scalar auxiliary variable ensemble algorithms for parameterized flow problems*, SIAM Journal on Scientific Computing, 43 (2021), A2869-A2896.
[32] 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.
[33] V. John, *Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder*, Int. J. Numer. Meth. Fluids 44 (2004), 777–788.
[34] L. Ju, W. Leng, Z. Wang and S. Yuan, *Numerical investigation of ensemble methods with block iterative solvers for evolution problems*, Discrete and Continuous Dynamical Systems - Series B, 25 (2020), 4905-4923.
[35] B. Kuznetsov, N. Vladimirova and N. Yanenko, *Numerical Calculation of the Symmetrical Flow of Viscous Incompressible Liquid around a Plate (in Russian)*, Studies in Mathematics and its Applications, Moscow: Nauka, 1966.
[36] W. Layton and M. McLaughlin, *Doubly-adaptive artificial compression methods for incompressible flow*, Journal of Numerical Mathematics, 28 (2020), 175-192.
[37] W. Layton, A. Takhirov and M. Sussman, *Instability of Crank–Nicolson leap-frog for nonautonomous systems*, Int. J. Numer. Anal. Model. Ser. B 5 (2014), 289–298.
[38] N. Li, J. Fiordilino and X. Feng, *Ensemble time-stepping algorithm for the convection-diffusion equation with random diffusivity*, Journal of Scientific Computing, 79 (2019), 1271-1293.
[39] Y. Li, Y. Hou and Y. Rong, *A second-order artificial compression method for the evolutionary Stokes-Darcy system*, Numerical Algorithms, 84 (2020), 1019-1048.
[40] 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.
[41] 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.
[42] Y. Luo and Z. Wang, *An ensemble algorithm for numerical solutions to deterministic and random parabolic PDEs*, SIAM Journal on Numerical Analysis, 56 (2018), 859-876.
[43] Y. Luo and Z. Wang, *A multilevel Monte Carlo ensemble scheme for random parabolic PDEs*, SIAM Journal on Scientific Computing, 41 (2019), A622-A642.
[44] J.F. McCarthy, *Block-conjugate-gradient method*, Physical Review D, 40 (1989), 2149.
[45] M. Mohebujjaman and L. Rebholz, *An efficient algorithm for computation of MHD flow ensembles*, Computational Methods in Applied Mathematics, 17 (2017), 121-137.
[46] D.P. O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra and its Applications, 29 (1980), 293-322.
[47] A. Philippe and F. Pierre *Convergence results for the vector penalty-projection and two-step artificial compressibility methods*, Discrete & Continuous Dynamical Systems - Series B, 17 (2012), pp. 1383-1405.
[48] M. Reagan, H.N. Najm, R.G. Ghanem and O.M. Knio, *Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection*, Combustion and Flame, 132 (2003), 545-555.
[49] Y. Rong, W. Layton and H. Zhao, *Numerical analysis of an artificial compression method for Magnetohydrodynamic flows at low magnetic Reynolds numbers*, Journal of Scientific Computing, 76 (2018), pp. 1458-1483.
[50] M. Schäfer and S. Turek,*Benchmark computations of laminar flow around cylinder, in: Flow Simulation with High-Performance Computers II*, Notes Numer. Fluid Mech. 52, Vieweg, Wiesbaden (1996), 547–566.
[51] 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.
[52] J. Shen, J. Xu and J. Yang, *The scalar auxiliary variable (SAV) approach for gradient flows*, Journal of Computational Physics, 353 (2018), 407-416.
[53] A. Takhirov, M. Neda, and J. Waters, *Time relaxation algorithm for flow ensembles*, Numerical Methods for Partial Differential Equations, 32 (2016), 757-777.
[54] 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.
[55] R. Temam *Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I)*, Arch. Rational. Mech. Anal., 33 (1969), pp. 135-153.
[56] R. Temam *Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II)*, Arch. Rational. Mech. Anal., 33 (1969), pp. 377-385.
[57] D. Xiu and J.S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM Journal on Scientific Computing, 27 (2005), 1118-1139.

[58] D. Xiu and G.E. Karniadakis, *Modeling uncertainty in flow simulations via generalized polynomial chaos*, Journal of Computational Physics, 187 (2003), 137-167.
[59] D. Xiu and G.E. Karniadakis, *A new stochastic approach to transient heat conduction modeling with uncertainty*, International Journal of Heat and Mass Transfer, 46 (2003), 4681-4693.