A SECOND ORDER ENSEMBLE ALGORITHM FOR COMPUTING THE NAVIER-STOKES EQUATIONS

NAN JIANG* AND HUANHUAN YANG †

Abstract. We present an efficient, second order ensemble algorithm for computing the Navier-Stokes equations multiple times with different model parameters. The algorithm is based on the Crank-Nicolson Leap-frog (CNLF) scheme and the ensemble timestepping incorporating the artificial compressibility method and the recent scalar auxiliary variable (SAV) idea for developing unconditionally stable schemes for nonlinear flows. The computation of the velocity and pressure is decoupled in the algorithm resulting in smaller linear systems to be solved at each time step. All realizations of the flow corresponding to different model parameters share the same coefficient matrix so that efficient block solvers can be used to reduce the computational cost. The proposed algorithm is efficient in terms of both computer storage and CPU time. We prove the algorithm is long time stable without any timestep conditions. Ample numerical experiments are performed for various flow problems to validate the efficiency and effectiveness of the algorithm.

Key words. Navier-Stokes equations, Crank-Nicolson Leap-frog, artificial compressibility, ensemble algorithm, scalar auxiliary variable, stabilization

1. Introduction. Computing a flow equation multiple times with different input data is a common procedure used in many engineering applications for uncertainty quantification (UQ), sensitivity analysis and data assimilation [10, 42, 53]. For complex flow problems this procedure could be prohibitively expensive if a large ensemble of realizations are required to produce useful statistical data. To address this issue, an efficient ensemble timestepping approach was developed in [27] for fast computation of nonlinear flow ensembles. The main idea is to decompose the nonlinear term into two parts: the ensemble mean and the fluctuation, with the mean independent of the ensemble index j and the fluctuation being lagged to the previous time levels so that it does not contribute to the coefficient matrix. The resulting linear systems for all ensemble members have the same coefficient matrix for which there exist efficient block solvers, such as block CG [23,43,46], block GMRES [6,13], to compute all linear systems quickly at one pass, as opposed to solving them separately for each ensemble member.

The ensemble timestepping approach has been adapted to compute various nonlinear flow problems including the Navier-Stokes equations [15–19, 25, 28, 31, 33–35, 50, 51], MHD flows [4, 5, 30, 44, 45], Boussinesq equations [11, 12, 24], fluid-fluid interactions [8], and shown to be highly efficient and have comparable accuracy to that of traditional methods that compute each ensemble member individually. Although the fact that all realizations share a common coefficient matrix could lead to significant savings in the computational cost, especially for a large ensemble, it is worth mentioning that this common coefficient matrix changes from one time step to another and needs to be assembled at each time step. A recent development on the ensemble methods is to incorporate the scalar auxiliary variable (SAV) approach to further reduce the computational cost [33], in which the nonlinear term is treated fully explicitly and the corresponding linear systems after spatial discretization have a common constant coefficient matrix that is time independent. In this paper we follow the same idea and propose a highly efficient, second order ensemble algorithm incorporating the SAV approach, the Crank-Nicolson-Leap-Frog (CNLF) time stepping and the artificial compression method for fast computation of the Navier-Stokes flow ensembles.

The SAV approach was first studied for the gradient flows [48, 49] and then extended for solving the Navier-Stokes equations in [41]. The main idea is to introduce a new scalar variable and the associated differential equation to form a new governing system for the model problem, for which an unconditionally stable discrete scheme with fully explicit nonlinear terms can then be developed. For

*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 and 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).

the Navier-Stokes equations, making the nonlinear term fully explicit in standard discrete schemes will lead to a restrictive CFL condition to ensure stable simulations. The SAV approach puts forth a new way to redesign these standard schemes and develop a new family of methods with a fully explicit nonlinear term as well as provable unconditional stability. However, it has been noted that the low accuracy of the SAV schemes has compromised its claimed unconditional stability. For many flow problems, a very small time step is needed for accurate and stable simulations. In [33], a stabilization method was proposed and shown to be able to greatly increase the accuracy of the SAV schemes and therefore the stability. In this report we incorporate this stabilization with our SAV ensemble scheme to improve accuracy and stability.

The CNLF scheme is widely used in atmospheric and oceanic simulations for its high accuracy but was less studied and analyzed in the literature until recently [21, 26, 29, 32, 37–39]. We adopt the CNLF timestepping scheme combined with artificial compressibility(AC) studied in [9] and a special extrapolation from [22] to develop an efficient ensemble timestepping scheme with provable long time stability. The combined approach of the SAV method, ensemble timestepping and CNLFAC scheme is extremely efficient : (1) The coefficient matrix only needs to be assemble once for all realizations and all time steps; (2) Utilizing the GMRES iterative solver paired with an appropriate preconditioner, an significant amount of computational cost as well as CPU time can be saved, see Section 5.3.

The rest of the paper is outlined here. Section 2 presents the proposed AC-SAV-CNLF ensemble algorithm. We prove the algorithm is nonlinearly, long time stable under one parameter condition, without any timestep constraints in Section 3. An efficient, fully decoupled implementation algorithm as well as details about algebraic systems and linear solvers are presented in Section 4. We perform extensive numerical testing for the algorithm to demonstrate its efficiency and effectiveness for various flow problems including a 3D tess. Numerical results are summarized and discussed in Section 5.

2. The AC-SAV-CNLF Ensemble Algorithm. Consider the case that the solution of Navier-Stokes equations depends on a set of input parameters coming from uncertainties in the kinematic viscosity, body forces, initial conditions, and boundary conditions. To obtain solution data we assume J samples of the random parameters have been generated by an efficient UQ method, e.g., [20, 53], and next we need to find J solutions to the Navier-Stokes equations corresponding to the J different parameter sets.

Consider J Navier-Stokes equations on a bounded domain Ω:

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

where the initial conditions $u_j^0(x)$, boundary conditions $g_j(x,t)$, body forces $f_j(x,t)$ and viscosity $\nu_j(x)$, $j = 1, ..., J$, have mild difference among different samples. To construct the ensemble algorithms we need to define the ensemble mean $\bar{\nu}$ of the viscosities $\nu_j(x)$ and their ensemble fluctuations ν_j'. Assumed $\nu_j(x) \geq \nu_{j,min} > 0$, the minimum average $\bar{\nu}_{min}$ and maximum fluctuation of the kinematic viscosity ν_{max}' will be used in the proof of the long time stability.

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

Next we introduce a scalar auxiliary variable q_j for each realization and a differential equation for it, which will be added to the original Navier-Stokes equations to form a new governing system, following the SAV idea in [33, 40, 41]. The introduced scalar $q_j(t)$ is defined 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 and $\delta > 0$ is a user-defined parameter introduced to ensure positivity of $E(u_j)$. Taking derivative of $q_j(t)$ gives

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

Note that the last extral two terms in the above equation are equal to zero since $\nabla\cdot u_j = 0$. Combining this equation with the original Navier-Stokes equations we have a new governing system that is equivalent to (2.1):

$$\begin{aligned}
&\partial_t u_j + \frac{q_j(t)}{\sqrt{E(u_j)+\delta}}(u_j\cdot\nabla)u_j - \nabla\cdot(\nu_j\nabla u_j) + \nabla p_j = f_j(x,t),\\
&\nabla\cdot u_j = 0,\\
&\frac{dq_j}{dt} = \frac{1}{2q_j}\int_\Omega \frac{\partial u_j}{\partial t}\cdot u_j\,dx + \frac{1}{2\sqrt{E(u_j)+\delta}}\int_\Omega (u_j\cdot\nabla)u_j\cdot u_j\,dx - \frac{1}{2q_j}\int_{\partial\Omega}(\vec{n}\cdot g_j)\tfrac{1}{2}|g_j|^2\,d\sigma.
\end{aligned} \tag{2.4}$$

Herein we present a second order, artificial compression SAV ensemble algorithm based on the Crank-Nicolson Leap-frog timestepping (AC-SAV-CNLF).

To make the algorithm efficient, we desire to let all the realizations share a common coefficient matrix. The way to do this is to decompose the viscosity term into two parts: the mean $\bar{\nu}$ and the fluctuation ν_j', i.e. $\nu_j = \bar{\nu} + \nu_j'$. The mean viscosity does not depend on the ensemble index j but the fluctuation term does. Hence we need to make the fluctuation term explicit in time discretization so that it does not affect the common operator in the left hand side. We usually need to decompose the nonlinear terms as well if a standard semi-implicit method is used. But thanks to the SAV approach, the nonlinear term can be treated fully explicitly and thus goes directly to the right hand side not contributing to the coefficient matrix in full discretization.

For the AC-SAV-CNLF ensemble algorithm, we adopt a special second order linear extrapolation $\tilde{\tilde{u}}_j^n$ [22], defined below to approximate the fluctuation term.

Let $t_n = n\Delta t$, $n = 0, 1, 2, \cdots, N$, where $N = T/\Delta t$, denote a uniform partition of the interval $[0, T]$. Denote

$$\begin{aligned}
&\tilde{u}_j^n = \frac{u_j^{n+1}+u_j^{n-1}}{2}, \qquad \tilde{\tilde{u}}_j^n = 2\frac{u_j^n+u_j^{n-2}}{2} - \frac{u_j^{n-1}+u_j^{n-3}}{2} = 2\tilde{u}_j^{n-1} - \tilde{u}_j^{n-2},\\
&\tilde{p}_j^n = \frac{p_j^{n+1}+p_j^{n-1}}{2}, \qquad \tilde{q}_j^n = \frac{q_j^{n+1}+q_j^{n-1}}{2}.
\end{aligned}$$

Note that $\tilde{\tilde{u}}_j^n$ is a known quantity while $\tilde{u}_j^n$ contains the unknown function u_j^{n+1}. We now propose our second order, stabilized SAV ensemble algorithm based on the Crank-Nicolson Leapfrog timestepping as follows.

Algorithm 2.1 (AC-SAV-CNLF). *For $j = 1, 2, \cdots, J$, given $u_j^0, u_j^1, u_j^2, u_j^3, p_j^2, p_j^3, q_j^2$, for $n = 3$, $\cdots$, $N-1$, find $u_j^{n+1}, p_j^{n+1}, q_j^{n+1}$ satisfying*

$$\begin{aligned}
&\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} - \beta\Delta t^{-1}\nabla(\nabla\cdot(u_j^{n+1}-u_j^{n-1})) + \frac{\tilde{q}_j^n}{\sqrt{E(u_j^n)+\delta}}(u_j^n\cdot\nabla)u_j^n + \nabla p_j^n\\
&\qquad - \nabla\cdot(\bar{\nu}\nabla\tilde{u}_j^n) - \nabla\cdot(\nu_j'\nabla\tilde{\tilde{u}}_j^n) - \alpha h\Delta(u_j^{n+1}-u_j^{n-1}) = f_j^n,
\end{aligned} \tag{2.5}$$

$$\epsilon\Delta t(p_j^{n+1}-p_j^{n-1}) + \nabla\cdot u_j^n = 0, \tag{2.6}$$

$$\frac{q_j^{n+1}-q_j^{n-1}}{2\Delta t} = \frac{1}{2\tilde{q}_j^n}\left(\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t}, \tilde{u}_j^n\right) + \frac{1}{2\sqrt{E(u_j^n)+\delta}}\int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot\tilde{u}_j^n\,dx - \frac{b_j^n}{2\tilde{q}_j^n}, \tag{2.7}$$

where $b_j^n = \int_{\partial\Omega}(\vec{n}\cdot g_j^n)\frac{1}{2}|g_j^n|^2\,d\sigma, \alpha > 0, \beta > 0, \epsilon > 0$.

A stabilization term $-\alpha h\Delta(u_j^{n+1} - u_j^{n-1})$ for AC-SAV-CNLF, is added in the algorithm to increase the accuracy of the SAV scheme [33], in which h is the mesh size associated with the spatial discretization of choice. This algorithm is second order convergent. The efficiency lies in the fact that all realizations share the same constant coefficient matrix, which makes it possible to use a block GMRES solver for fast computation of all realizations simultaneously. We will present an efficient implementation algorithm in Section 4 and ample numerical tests in Section 5 to demonstrate its high efficiency and superior performance over traditional methods that compute each realization individually.

3. Stability of the AC-SAV-CNLF Ensemble Algorithm. In this section we prove the AC-SAV-CNLF ensemble algorithm is long time stable under a parameter fluctuation condition, without any timestep constraints.

THEOREM 3.1 (Stability of AC-SAV-CNLF). *Let* $\epsilon\beta \geq \frac{1}{4}$. *Assume the boundary condition is homogeneous,* q_j^n *is real, and the following parameter fluctuation condition holds*

$$\frac{\nu'_{max}}{\bar{\nu}_{min}} < \frac{1}{3}. \tag{3.1}$$

Algorithm 2.1 is long time stable in the sense that

$$\begin{aligned}
&|q_j^N|^2 + |q_j^N - 1|^2 + 3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-1}\|^2 + \nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-2}\|^2 + \alpha h\Delta t\|\nabla u_j^N\|^2 + \alpha h\Delta t\|\nabla u_j^{N-1}\|^2\\
&\leq |q_j^3|^2 + |q_j^2|^2 + 2\beta\|\nabla\cdot u_j^3\|^2 + 2\beta\|\nabla\cdot u_j^2\|^2 + 3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^2\|^2 + \nu'_{max}\Delta t\|\nabla\tilde{u}_j^1\|^2 + \alpha h\Delta t\|\nabla u_j^3\|^2\\
&+ \alpha h\Delta t\|\nabla u_j^2\|^2 + \frac{\Delta t}{(\bar{\nu}_{min} - 3\nu'_{max})}\sum_{n=3}^{N-1}\|f_j^n\|_{-1}^2 + 2\epsilon\Delta t^2\|p_j^3\|^2 + 2\epsilon\Delta t^2\|p_j^2\|^2.
\end{aligned} \tag{3.2}$$

Proof. Assuming homogeneous Dirichlet boundary condition, testing equation (2.5) with $\tilde{u}_j^n$ gives

$$\begin{aligned}
&\left(\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t}, \tilde{u}_j^n\right) + \frac{\beta}{2\Delta t}\left(\|\nabla\cdot u_j^{n+1}\|^2 - \|\nabla\cdot u_j^{n-1}\|^2\right) + \frac{\tilde{q}_j^n}{\sqrt{E(u_j^n)+\delta}}b(u_j^n, u_j^n, \tilde{u}_j^n) + \|\bar{\nu}^{\frac{1}{2}}\nabla\tilde{u}_j^n\|^2\\
&+ (\nu'_j\nabla\tilde{\tilde{u}}_j^n, \nabla\tilde{u}_j^n) - \frac{1}{2}(p_j^n, \nabla\cdot\left(u_j^{n+1} + u_j^{n-1}\right)) + \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right)\\
&= (f_j^n, \tilde{u}_j^n).
\end{aligned} \tag{3.3}$$

Testing equation (2.6) with $\tilde{p}_j^n$ gives

$$\frac{\epsilon\Delta t}{2}\left(\|p_j^{n+1}\|^2 - \|p_j^{n-1}\|^2\right) + \frac{1}{2}\left(p_j^{n+1} + p_j^{n-1}, \nabla\cdot u_j^n\right) = 0. \tag{3.4}$$

Multiplying (2.7) with $2\tilde{q}_j^n$ gives

$$\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) = \left(\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t}, \tilde{u}_j^n\right) + \frac{\tilde{q}_j^n}{\sqrt{E(u_j^n)+\delta}}b(u_j^n, u_j^n, \tilde{u}_j^n) - b_j^n. \tag{3.5}$$

Adding (3.3), (3.4) and (3.5) gives

$$\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) + \frac{\beta}{2\Delta t}\left(\|\nabla\cdot u_j^{n+1}\|^2 - \|\nabla\cdot u_j^{n-1}\|^2\right) + \|\bar{\nu}^{\frac{1}{2}}\nabla\tilde{u}_j^n\|^2\\
&+ \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right) + \frac{\epsilon\Delta t}{2}\left(\|p_j^{n+1}\|^2 - \|p_j^{n-1}\|^2\right)
\end{aligned}$$

$$+\frac{1}{2}\left(p_j^{n+1}+p_j^{n-1},\nabla\cdot u_j^n\right)-\frac{1}{2}(p_j^n,\nabla\cdot\left(u_j^{n+1}+u_j^{n-1}\right))=(f_j^n,\tilde{u}_j^n)-(\nu_j'\nabla\tilde{\tilde{u}}_j^n,\nabla\tilde{u}_j^n). \tag{3.6}$$

We then bound the right hand side of equation (3.6). By the Cauchy-Schwarz inequality, the Young's inequality, and $(2a-b)^2\le 6a^2+3b^2$, we derive, for any $\beta_1>0,\epsilon_1>0$,

$$\begin{aligned}
&(f_j^n,\tilde{u}_j^n)-(\nu_j'\nabla\tilde{\tilde{u}}_j^n,\nabla\tilde{u}_j^n)\\
&\le\|f_j^n\|_{-1}\|\nabla\tilde{u}_j^n\|+\nu'_{max}\|\nabla\tilde{\tilde{u}}_j^n\|\|\nabla\tilde{u}_j^n\| \qquad (3.7)\\
&\le\beta_1\bar{\nu}_{min}\|\nabla\tilde{u}_j^n\|^2+\frac{1}{4\beta_1\bar{\nu}_{min}}\|f_j^n\|_{-1}^2+\frac{\epsilon_1\nu'_{max}}{2}\|\nabla\tilde{u}_j^n\|^2+\frac{\nu'_{max}}{2\epsilon_1}\|\nabla\tilde{\tilde{u}}_j^n\|^2\\
&\le\beta_1\bar{\nu}_{min}\|\nabla\tilde{u}_j^n\|^2+\frac{1}{4\beta_1\bar{\nu}_{min}}\|f_j^n\|_{-1}^2+\frac{\epsilon_1\nu'_{max}}{2}\|\nabla\tilde{u}_j^n\|^2+\frac{\nu'_{max}}{2\epsilon_1}\|\nabla\left(2\tilde{u}_j^{n-1}-\tilde{u}_j^{n-2}\right)\|^2\\
&\le\beta_1\bar{\nu}_{min}\|\nabla\tilde{u}_j^n\|^2+\frac{1}{4\beta_1\bar{\nu}_{min}}\|f_j^n\|_{-1}^2+\frac{\epsilon_1\nu'_{max}}{2}\|\nabla\tilde{u}_j^n\|^2+\frac{3\nu'_{max}}{\epsilon_1}\|\nabla\tilde{u}_j^{n-1}\|^2+\frac{3\nu'_{max}}{2\epsilon_1}\|\nabla\tilde{u}_j^{n-2}\|^2.
\end{aligned}$$

Notice that the term $\|\bar{\nu}^{\frac{1}{2}}\nabla\tilde{u}_j^n\|^2$ in the left hand side of (3.6) is bounded from below by $\bar{\nu}_{min}\|\nabla\tilde{u}_j^n\|^2$, so we need all the last three terms in (3.7) to be bounded by $\bar{\nu}_{min}\|\nabla\tilde{u}_j^n\|^2$. To this end, we minimize $\frac{\epsilon_1}{2}+\frac{3}{\epsilon_1}+\frac{3}{2\epsilon_1}$ by taking $\epsilon_1=3$. Inequality (3.6) then reduces to

$$\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2-|q_j^{n-1}|^2\right)+\frac{\beta}{2\Delta t}\left(\|\nabla\cdot u_j^{n+1}\|^2-\|\nabla\cdot u_j^{n-1}\|^2\right)+\left((1-\beta_1)\bar{\nu}_{min}-3\nu'_{max}\right)\|\nabla\tilde{u}_j^n\|^2\\
&+\frac{3}{2}\nu'_{max}\left(\|\nabla\tilde{u}_j^n\|^2-\|\nabla\tilde{u}_j^{n-1}\|^2\right)+\frac{1}{2}\nu'_{max}\left(\|\nabla\tilde{u}_j^{n-1}\|^2-\|\nabla\tilde{u}_j^{n-2}\|^2\right)+\frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^{n-1}\|^2\right)\\
&+\frac{\epsilon\Delta t}{2}\left(\|p_j^{n+1}\|^2-\|p_j^{n-1}\|^2\right)+\frac{1}{2}\left(p_j^{n+1}+p_j^{n-1},\nabla\cdot u_j^n\right)-\frac{1}{2}(p_j^n,\nabla\cdot\left(u_j^{n+1}+u_j^{n-1}\right)) \qquad (3.8)\\
&\le\frac{1}{4\beta_1\bar{\nu}_{min}}\|f_j^n\|_{-1}^2.
\end{aligned}$$

Once the parameter fluctuation condition is satisfied, $\bar{\nu}_{min}-3\nu'_{max}>0$. Setting $\beta_1=\frac{1}{2}-\frac{3}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}}>0$, we then have

$$(1-\beta_1)\bar{\nu}_{min}-3\nu'_{max}=(\frac{1}{2}+\frac{3}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}})\bar{\nu}_{min}-3\nu'_{max}=\frac{1}{2}\left(\bar{\nu}_{min}-3\nu'_{max}\right)>0. \tag{3.9}$$

Inequality (3.8) is now reduced to

$$\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2-|q_j^{n-1}|^2\right)+\frac{\beta}{2\Delta t}\left(\|\nabla\cdot u_j^{n+1}\|^2-\|\nabla\cdot u_j^{n-1}\|^2\right)+\frac{3}{2}\nu'_{max}\left(\|\nabla\tilde{u}_j^n\|^2-\|\nabla\tilde{u}_j^{n-1}\|^2\right)\\
&+\frac{1}{2}\nu'_{max}\left(\|\nabla\tilde{u}_j^{n-1}\|^2-\|\nabla\tilde{u}_j^{n-2}\|^2\right)+\frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2-\|\nabla u_j^{n-1}\|^2\right)\\
&+\frac{\epsilon\Delta t}{2}\left(\|p_j^{n+1}\|^2-\|p_j^{n-1}\|^2\right)+\frac{1}{2}\left(p_j^{n+1}+p_j^{n-1},\nabla\cdot u_j^n\right)-\frac{1}{2}(p_j^n,\nabla\cdot\left(u_j^{n+1}+u_j^{n-1}\right))\\
&\le\frac{1}{2(\bar{\nu}_{min}-3\nu'_{max})}\|f_j^n\|_{-1}^2. \qquad (3.10)
\end{aligned}$$

We can rewritten the two pressure terms in (3.10) as

$$\begin{aligned}
&\left(p_j^{n+1}+p_j^{n-1},\nabla\cdot u_j^n\right)-(p_j^n,\nabla\cdot\left(u_j^{n+1}+u_j^{n-1}\right)) \qquad (3.11)\\
&=\left[\left(p_j^{n+1},\nabla\cdot u_j^n\right)-\left(p_j^n,\nabla\cdot u_j^{n+1}\right)\right]-\left[\left(p_j^n,\nabla\cdot u_j^{n-1}\right)-\left(p_j^{n-1},\nabla\cdot u_j^n\right)\right]
\end{aligned}$$

Summing up (3.10) from $n=3$ to $n=N-1$ and multiplying through by $2\Delta t$ gives

$$|q_j^N|^2+|q_j^N-1|^2+\beta\|\nabla\cdot u_j^N\|^2+\beta\|\nabla\cdot u_j^{N-1}\|^2+3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-1}\|^2 \tag{3.12}$$

$$
\begin{aligned}
&+\nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-2}\|^2+\alpha h\Delta t\|\nabla u_j^N\|^2+\alpha h\Delta t\|\nabla u_j^{N-1}\|^2+\epsilon\Delta t^2\|p_j^N\|^2\\
&+\epsilon\Delta t^2\|p_j^{N-1}\|^2+\Delta t[(p_j^N,\nabla\cdot u_j^{N-1})-(p_j^{N-1},\nabla\cdot u_j^N)]\\
\le\ &|q_j^3|^2+|q_j^2|^2+\beta\|\nabla\cdot u_j^3\|^2+\beta\|\nabla\cdot u_j^2\|^2+3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^2\|^2+\nu'_{max}\Delta t\|\nabla\tilde{u}_j^1\|^2\\
&+\alpha h\Delta t\|\nabla u_j^3\|^2+\alpha h\Delta t\|\nabla u_j^2\|^2+\frac{\Delta t}{(\bar{\nu}_{min}-3\nu'_{max})}\sum_{n=3}^{N-1}\|f_j^n\|_{-1}^2+\epsilon\Delta t^2\|p_j^3\|^2\\
&+\epsilon\Delta t^2\|p_j^2\|^2+\Delta t[(p_j^3,\nabla\cdot u_j^2)-(p_j^2,\nabla\cdot u_j^3)].
\end{aligned}
$$

Now we further bound the following term by Cauchy-Schwarz inequality and Young's inequality.

$$
\begin{aligned}
&\Delta t|(p_j^N,\nabla\cdot u_j^{N-1})-(p_j^{N-1},\nabla\cdot u_j^N)| \qquad (3.13)\\
&\le\Delta t\left[\|p_j^N\|\|\nabla\cdot u_j^{N-1}\|+\|p_j^{N-1}\|\|\nabla\cdot u_j^N\|\right]\\
&\le\Delta t\left[\epsilon\Delta t\|p_j^N\|^2+\frac{1}{4\epsilon\Delta t}\|\nabla\cdot u_j^{N-1}\|^2+\epsilon\Delta t\|p_j^{N-1}\|^2+\frac{1}{4\epsilon\Delta t}\|\nabla\cdot u_j^N\|^2\right]\\
&=\epsilon\Delta t^2(\|p_j^N\|^2+\|p_j^{N-1}\|^2)+\frac{1}{4\epsilon}(\|\nabla\cdot u_j^N\|^2+\|\nabla\cdot u_j^{N-1}\|^2)\,.
\end{aligned}
$$

If $\beta\ge\frac{1}{4\epsilon}$, then (3.12) reduces to

$$
\begin{aligned}
&|q_j^N|^2+|q_j^N-1|^2+3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-1}\|^2+\nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-2}\|^2\\
&\quad+\alpha h\Delta t\|\nabla u_j^N\|^2+\alpha h\Delta t\|\nabla u_j^{N-1}\|^2 \qquad (3.14)\\
&\le|q_j^3|^2+|q_j^2|^2+\beta\|\nabla\cdot u_j^3\|^2+\beta\|\nabla\cdot u_j^2\|^2+3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^2\|^2\\
&+\nu'_{max}\Delta t\|\nabla\tilde{u}_j^1\|^2+\alpha h\Delta t\|\nabla u_j^3\|^2+\alpha h\Delta t\|\nabla u_j^2\|^2+\frac{\Delta t}{(\bar{\nu}_{min}-3\nu'_{max})}\sum_{n=3}^{N-1}\|f_j^n\|_{-1}^2\\
&+\epsilon\Delta t^2\|p_j^3\|^2+\epsilon\Delta t^2\|p_j^2\|^2+\Delta t[(p_j^3,\nabla\cdot u_j^2)-(p_j^2,\nabla\cdot u_j^3)].
\end{aligned}
$$

Finally, by

$$
\Delta t[(p_j^3,\nabla\cdot u_j^2)-(p_j^2,\nabla\cdot u_j^3)]\le\epsilon\Delta t^2(\|p_j^3\|^2+\|p_j^2\|^2)+\beta(\|\nabla\cdot u_j^3\|^2+\|\nabla\cdot u_j^2\|^2)\,, \qquad (3.15)
$$

we have

$$
\begin{aligned}
&|q_j^N|^2+|q_j^N-1|^2+3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-1}\|^2+\nu'_{max}\Delta t\|\nabla\tilde{u}_j^{N-2}\|^2+\alpha h\Delta t\|\nabla u_j^N\|^2+\alpha h\Delta t\|\nabla u_j^{N-1}\|^2 \qquad (3.16)\\
&\le|q_j^3|^2+|q_j^2|^2+2\beta\|\nabla\cdot u_j^3\|^2+2\beta\|\nabla\cdot u_j^2\|^2+3\nu'_{max}\Delta t\|\nabla\tilde{u}_j^2\|^2+\nu'_{max}\Delta t\|\nabla\tilde{u}_j^1\|^2+\alpha h\Delta t\|\nabla u_j^3\|^2\\
&+\alpha h\Delta t\|\nabla u_j^2\|^2+\frac{\Delta t}{(\bar{\nu}_{min}-3\nu'_{max})}\sum_{n=3}^{N-1}\|f_j^n\|_{-1}^2+2\epsilon\Delta t^2\|p_j^3\|^2+2\epsilon\Delta t^2\|p_j^2\|^2.
\end{aligned}
$$

□

4. Implementation Algorithm. The equations (2.5) and (2.7) in the AC-SAV-CNLF scheme are coupled systems of u_j^{n+1} and q_j^{n+1}, hence we need a strategy to decouple u and q so that the implementation achieves desired efficiency. In the following we describe implementation skills for the AC-SAV-CNLF ensemble algorithm, following the decoupling strategy in [40, 41]. After that, we also state advantages of the AC-SAV-CNLF ensemble scheme from the aspect of numerical linear algebra.

A new scalar S_j^{n+1} will be introduced to split the numerical solution u_j^{n+1} into two parts yielding two subproblems. Based on this splitting, one can formulate a separate equation for S_j^{n+1} then.

Let

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

Substituting (4.1) into (2.5) and (2.7) and collecting the terms with S_j^{n+1}, we can derive an equation for $\hat{u}_j^{n+1}$; collecting those terms without S_j^{n+1} we then derive an equation for $\breve{u}_j^{n+1}$. Instead of solving (2.5) and (2.7), we then solve the resulting two subproblems to obtain $\hat{u}_j^{n+1}$ and $\breve{u}_j^{n+1}$ respectively:

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

(AC-SAV-CNLF subproblem 1)

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

(AC-SAV-CNLF subproblem 2)

To derive an equation for S_j^{n+1}, we proceed as follows.

$$S_j^{n+1} = \frac{q_j^{n+1}+q_j^{n-1}}{2\sqrt{E(u_j^n)+\delta}} \qquad \Longrightarrow \qquad q_j^{n+1} = 2\sqrt{E(u_j^n)+\delta}\, S_j^{n+1} - q_j^{n-1}. \tag{4.2}$$

Plugging this expression of q_j^{n+1} into (3.5) gives

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

The equation for solving S_j^{n+1} then writes

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

where

$$A_j^{n+1} = \frac{2}{\Delta t}(E(u_j^n)+\delta) - \left(\frac{\breve{u}_j^{n+1}}{2\Delta t}, \frac{\breve{u}_j^{n+1}}{2}\right) - \int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot\frac{\breve{u}_j^{n+1}}{2}\,dx,$$

$$B_j^{n+1} = -\frac{2}{\Delta t}\sqrt{E(u_j^n)+\delta}q_j^{n-1} - \left(\frac{\check{u}_j^{n+1}}{2\Delta t}, \frac{\hat{u}_j^{n+1}+u_j^{n-1}}{2}\right) - \left(\frac{\hat{u}_j^{n+1}-u_j^{n-1}}{2\Delta t}, \frac{\check{u}_j^{n+1}}{2}\right)$$
$$- \int_\Omega (u_j^n\cdot\nabla)u_j^n\cdot(\frac{\hat{u}_j^{n+1}+u_j^{n-1}}{2})\,dx,$$
$$C_j^{n+1} = -\left(\frac{\hat{u}_j^{n+1}-u_j^{n-1}}{2\Delta t}, \frac{\hat{u}_j^{n+1}+u_j^{n-1}}{2}\right) + b_j^n.$$

To compute S_j^{n+1}, we only need to pick the root of above scalar quadratic equation that is close to 1.

Subproblem 1 and 2 can be solved in a very efficient way, as their fully discretized systems have the same constant coefficient matrix shared by all realizations. Subproblem 3 can also be solved super fast since it is a scalar quadratic equation. After obtaining $\hat{u}_j^{n+1}, \check{u}_j^{n+1}$, and S_j^{n+1}, we then compute $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\check{u}_j^{n+1}$. The pressure can be updated directly through (2.6).

Remarks on Algebraic Linear Systems. For spatial discretization we resort to the finite element method and denote the basis functions for u and p by $\{\chi_j^u\}_{j=1}^{N_u}$ and $\{\chi_j^p\}_{j=1}^{N_p}$ respectively. In the fully discretized linear systems, the following matrices $\mathbf{M}_{uu}$, $\mathbf{D}_{uu}$, $\mathbf{D}_{uup}$, $\mathbf{S}(\nu)$, and $\mathbf{N}(u)$ will be used, and their (k,l) entries are

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

In the numerical experiment section, we will test efficiency of the AC-SAV-CNLF ensemble scheme by comparing it with the one not using AC technique, namely the SAV-CNLF ensemble scheme. To have a better understanding, below we write out then compare the matrices of algebraic linears system discretized from above schemes, and the standard CNLF nonensemble scheme, for the realization of sample $j = 1, \cdots, J$.

(1) AC-SAV-CNLF ensemble:

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

(2) SAV-CNLF ensemble:

$$\mathbf{A}_{\text{savens}} = \begin{pmatrix} \frac{1}{2\Delta t}\mathbf{M}_{uu} + \mathbf{S}(\frac{\bar{\nu}}{2}+\alpha h) & -\frac{1}{2}\mathbf{D}_{uup} \\ -\frac{1}{2}\mathbf{D}_{uup}^T & \mathbf{0} \end{pmatrix}.$$

(3) CNLF nonensemble:

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

Several advantages exist on the structure of the matrix $\mathbf{A}_{\text{acsavens}}$. When compared with $\mathbf{A}_{\text{savens}}$, the matrix $\mathbf{A}_{\text{acsavens}}$ is not only in reduced size, but also symmetric and positive definite (SPD), hence the conjugate gradient (CG) iterative linear solver is applicable. As compared with $\mathbf{A}_{\text{nonens}}^{(j,n)}$, the matrix $\mathbf{A}_{\text{acsavens}}$ is constant, independent of the sample index j or time index n, thus we can realize all J simulations concurrently by solving a single linear system with multiple right hand sides corresponding to different samples. In contrast, the matrix $\mathbf{A}_{\text{nonens}}^{(j,n)}$ from the nonensemble scheme varies with the sample index j and time index n, so we need to assemble a new matrix at each time step for each sample, which is much more time consuming.

Table 5.1: Numerical errors and convergence rates of the AC-SAV-CNLF ensemble algorithm ($J = 10$) with $T = 0.5, \Delta t = 0.1h$, $\alpha = 1$, $\beta = 0.1$, $\epsilon = 2.5$, $\nu_{min} = 0.1$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,10}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,10}$	Rate
1/8	2.504×10^{-3}	—	2.076×10^{-3}	—	2.303×10^{-3}	—	2.097×10^{-3}	—
1/16	6.321×10^{-4}	1.99	5.048×10^{-4}	2.04	5.797×10^{-4}	1.99	5.133×10^{-4}	2.03
1/32	1.577×10^{-4}	2.00	1.263×10^{-4}	2.00	1.447×10^{-4}	2.00	1.292×10^{-4}	1.99
1/64	3.934×10^{-5}	2.00	3.027×10^{-5}	2.06	3.612×10^{-5}	2.00	3.088×10^{-5}	2.06
1/128	9.835×10^{-6}	2.00	7.717×10^{-6}	1.97	9.029×10^{-6}	2.00	7.837×10^{-6}	1.98

The system associated with $\mathbf{A}_{\text{acsavens}}$ in the AC-SAV-CNLF ensemble scheme can be handeled by the block CG method [23, 43, 46] or CG method with vectorization operation, preconditioned by multigrid. In this way, we can achieve high computational efficiency by solving J solutions simutaneously, as we can remove redundance due to linear dependence of different realizations in advance. As for the SAV-CNLF ensemble scheme, the block GMRES solver [6, 13] with the least-squares commutator preconditioner was claimed having good performance [33]. So this linear solver is used for the SAV-CNLF ensemble scheme in the upcoming efficiency test.

5. Numerical Experiments.

5.1. Convergence test. In this section we use a Green-Taylor vortex solution [14] to validate the convergence rate of the proposed algorithm. The exact solution is defined on a square domain $\Omega = (0, 1)^2$:

$$u = (-\cos x \sin y, \sin x \cos y)^T g(t),$$
$$p = -\frac{1}{4}[\cos(2x) + \cos(2y)]g(t)^2.$$

Then the forcing term is given accordingly by

$$f(x, y, t) = [g'(t) + 2\nu g(t)](-\cos x \sin y, \sin x \cos y)^T,$$

with $g(t) = e^{\nu} \cos(2t)$. A Dirichlet boundary condition is used and set to match the exact solution.

In our test, the ensemble of flows are computed based on the following viscosities:

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

We compute $J = 10$ samples in this test with $\epsilon_j = 0.2(j - 1)/J$. The AC parameters are set as

$$\alpha = 1, \ \beta = 0.1, \ \epsilon = 0.25/\beta = 2.5.$$

Simulations are performed on successively refined meshs with $h = 1/8, \cdots, 1/128$. The final simulation time is set to be $T = 0.5$. Taking $\Delta t = ch$, the expected errors of numerical solutions on u in H^1 semi-norm and p in L^2 norm are both $O(h^2 + \Delta t^2) = O(\Delta t^2)$. In Table 5.1, 5.2, and 5.3, we report the errors and convergence rates for the first and tenth samples with $\nu_{min} = 0.1$, $\nu_{min} = 0.01$, and $\nu_{min} = 0.001$, respectively. Specifically, in these tables, $\lvert u_h - u\rvert_{H^1}^{E,j}$ denotes the numerical error for u_j at final time in H^1 semi-norm, and $\lvert p_h - p\rvert_{L^2}^{E,j}$ the numerical error for p_j in L^2 norm. The AC-SAV-CNLF ensemble scheme shows second-order convergence in time as expected.

5.2. Channel flow past a cylinder. We now consider a two-dimensional channel flow past a cylinder, a classical benchmark problem introduced in Schäfer and Turek [47] and then widely used [32, 39] to observe the stability or effectiveness of certain numerical schemes of Navier-Stokes

Table 5.2: Numerical errors and convergence rates of the AC-SAV-CNLF ensemble algorithm ($J = 10$) with $T = 0.5, \Delta t = 0.1h$, $\alpha = 1$, $\beta = 0.1$, $\epsilon = 2.5$, $\nu_{min} = 0.01$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,10}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,10}$	Rate
1/8	5.699×10^{-3}	—	1.874×10^{-3}	—	5.505×10^{-3}	—	1.877×10^{-3}	—
1/16	1.577×10^{-3}	1.85	4.627×10^{-4}	2.02	1.498×10^{-3}	1.88	4.643×10^{-4}	2.02
1/32	3.994×10^{-4}	1.98	1.218×10^{-4}	1.93	3.757×10^{-4}	2.00	1.224×10^{-4}	1.92
1/64	9.870×10^{-5}	2.02	2.746×10^{-5}	2.15	9.287×10^{-5}	2.02	2.750×10^{-5}	2.15
1/128	2.458×10^{-5}	2.01	6.845×10^{-6}	2.00	2.316×10^{-5}	2.00	6.845×10^{-6}	2.00

Table 5.3: Numerical errors and convergence rates of the AC-SAV-CNLF ensemble algorithm ($J = 10$) with $T = 0.5, \Delta t = 0.01h$, $\alpha = 1$, $\beta = 0.1$, $\epsilon = 2.5$, $\nu_{min} = 0.001$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,10}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,10}$	Rate
1/8	4.530×10^{-3}	—	3.212×10^{-4}	—	4.317×10^{-3}	—	3.213×10^{-4}	—
1/16	1.192×10^{-3}	1.93	7.235×10^{-5}	2.15	1.070×10^{-3}	2.01	7.245×10^{-5}	2.15
1/32	2.168×10^{-4}	2.46	1.947×10^{-5}	1.89	1.920×10^{-4}	2.48	1.947×10^{-5}	1.90
1/64	3.777×10^{-5}	2.52	4.716×10^{-6}	2.05	3.407×10^{-5}	2.49	4.714×10^{-6}	2.05
1/128	7.596×10^{-6}	2.31	1.220×10^{-6}	1.96	7.021×10^{-6}	2.28	1.216×10^{-6}	1.96

equations. Our purpose here is to illustrate that the AC-SAV-CNLF scheme produces reasonable simulations even with large time steps if appropriate stabilization and AC parameters are chosen.

The flow pasts through a 2.2×0.41 rectangular channel blocked by an internal cylinder of radius 0.05 centered at $(0.2, 0.2)$. Its inflow and outflow boundary conditions are set as

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

No-slip boundary conditions are imposed on the other parts, including the top and bottom of the channel, the boundary of cylinder. The initial condition and body force are set to be zero. When setting $\nu = 10^{-3}$, the problem features a laminar flow with Reynolds number $Re = 100$; a Kármán vortex street will be developed behind the cylinder.

Simulations are performed with $\Delta t = 0.01, 0.005, 0.001$, while the stabilization and AC parameters are set as $\alpha = 1, \beta = 0.1, \epsilon = 0.25/\beta = 2.5$. All simulations have a common spatial resolution with maximum 0.0147, and the Taylor-Hood elements hold 63920 (16155 resp.) degrees of freedom for the velocity (pressure resp.). The time histories of simulated energy and S_j^n are plotted in Figure 5.1. One can observe that S_j^n does converge to one when the simulation is performed using relatively large time steps. Specifically, a large $\Delta t = 0.01$ is good enough for stable and reliable simulations.

Figure 5.2 plots the velocity fields and magnitudes of the flow at $t = 4, 5, 7, 8$, simulated by the AC-SAV-CNLF method with $\Delta t = 0.01$. Figure 5.3 is produced with $\Delta t = 0.001$. In both figures, the simulation results are satisfactory: the AC-SAV-CNLF method is stable and the flow patterns match with those in [32, 36].

5.3. 2D and 3D efficiency tests.

2D performance. Here we resort to the analytic solution in Section 5.1 for the purpose of testing the AC efficiency and ensemble efficiency of the AC-SAV-CNLF ensemble scheme. The viscosities ν_j

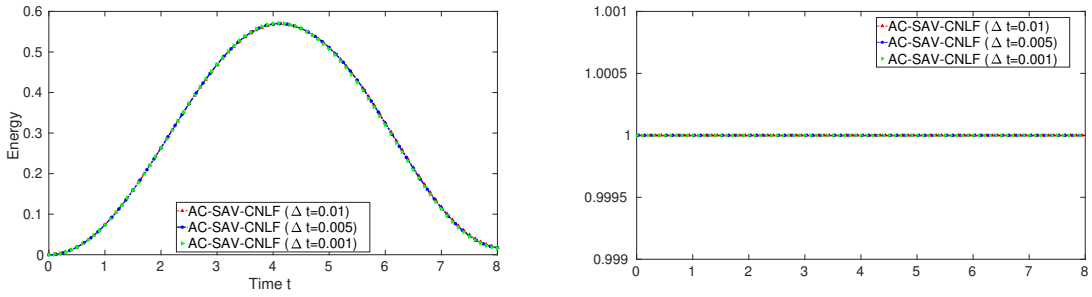

Fig. 5.1: Flow past cylinder: evolution of energy and S_j^n computed by the AC-SAV-CNLF scheme with $\nu = 10^{-3}, h_{max} = 0.0147, T = 8, J = 1 = j$.

are set as

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

the numerical resolutions are fixed as $h = \Delta t = 0.01$, and the final time is taken at $T = 0.5$.

To begin with, the performance of AC-SAV-CNLF and SAV-CNLF schemes on simulating a single flow, i.e. $J = 1$, is compared to observe the advantage of AC technique. Extensive testing is carried out with different ν_{min} values: $\nu_{min} = 0.1, 0.01, 0.001$. In this efficiency test, we use the block CG linear solver preconditioned by multi-grid for the AC-SAV-CNLF method. As for the SAV-CNLF scheme, the block GMRES linear solver with the least-squares commutator preconditioner is employed. The efficiency is studied in MATLAB with our finite-element code based on the data structure of iFEM package.

The execution times and numerical errors computed by the two schemes are reported in Table 5.4, where $|\mathbb{E}[u_h - u]|_{H^1}$ denotes the expection of error in H^1 semi-norm. We have set $\alpha = 1$ in both methods. To obtain comparable accuracy, we adjust the values of AC parameters β and ϵ in AC-SAV-CNLF. In particular, we take $\beta = 0.5, \epsilon = 0.25/\beta$ for the cases $\nu_{min} = 0.1$ and $\nu_{min} = 0.01$, whereas $\beta = 0.06, \epsilon = 0.25/\beta$ for the case $\nu_{min} = 0.001$. One can see from Table 5.4 that the AC-SAV-CNLF scheme outperforms SAV-CNLF since its execution consumes much less CPU time while providing similar accuracy. This is a consequence of using AC technique for splitting the velocity and pressure. We also mention the reason why the reported CPU time for SAV-CNLF is less as ν_{min} decreases: the least-squares commutator preconditioner is more efficient for smaller viscosity, as illustrated in Table 1 of [33].

Table 5.4: Numerical errors and CPU times evaluated with $T = 0.5, h = \Delta t = 0.01, J = 1$.

	AC-SAV-CNLF		SAV-CNLF	
ν_{min}	$\|\mathbb{E}[u_h - u]\|_{H^1}$	Exe time	$\|\mathbb{E}[u_h - u]\|_{H^1}$	Exe time
0.1	1.50×10^{-4}	157 s	1.18×10^{-4}	786 s
0.01	2.37×10^{-4}	222 s	2.29×10^{-4}	425 s
0.001	9.20×10^{-4}	147 s	9.20×10^{-4}	363 s

We then study the ensemble efficiency by simulating with different J values: $J = 1, 10, 100$. Figure 5.4 plots the CPU times of performing the AC-SAV-CNLF ensemble algorithm with respect to the sample size J. There, the red line of dashes serves as a reference to linear increase of execution time with respect to J. From this figure we can observe that the advantage of the AC-SAV-CNLF ensemble algorithm is apparent; when the sample size J is kind of large, the execution time is significantly

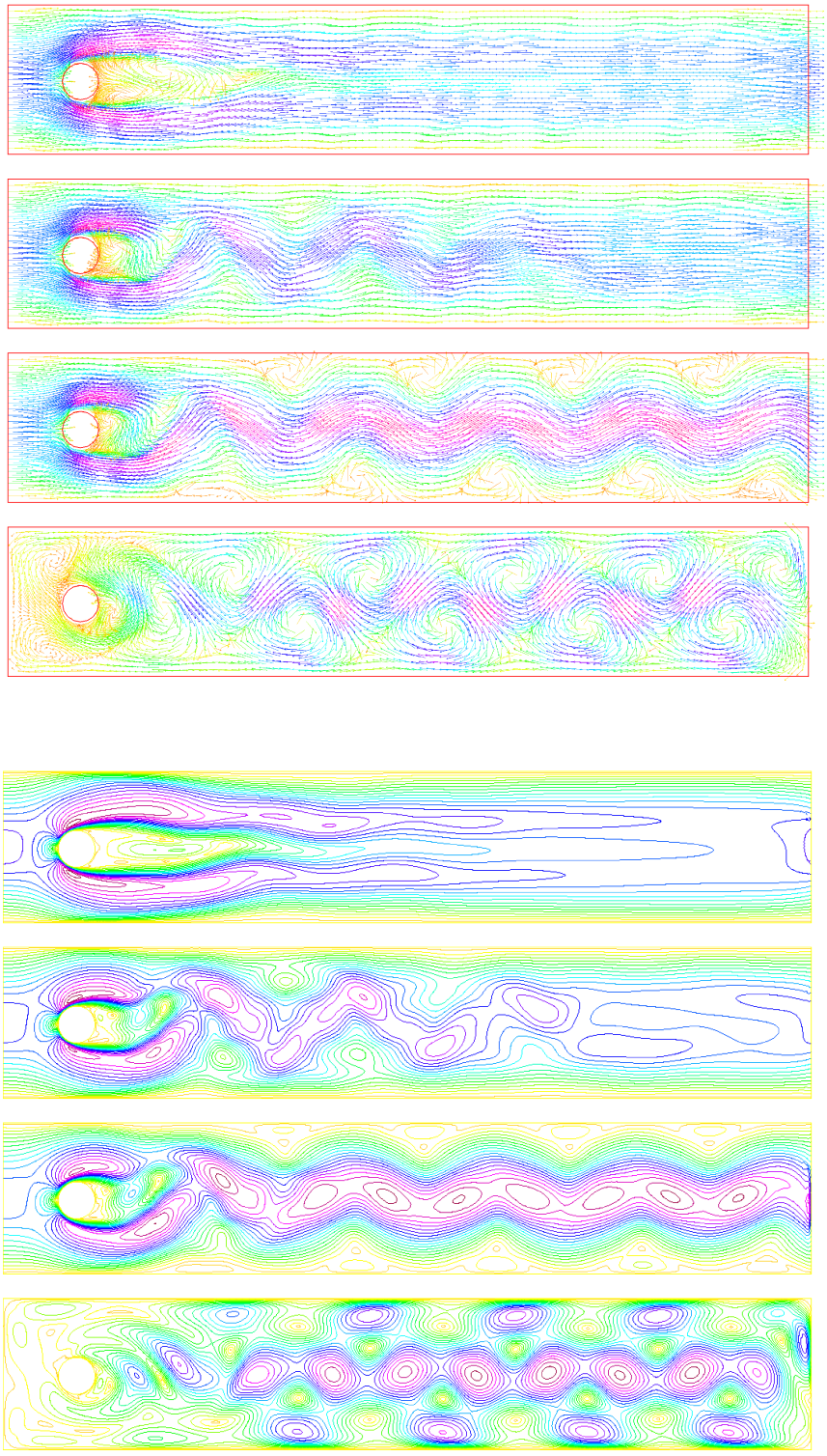

Fig. 5.2: Flow past cylinder: velocity fields and magnitudes at $t = 4, 5, 7, 8$, computed by the AC-SAV-CNLF scheme with $\Delta t = 0.01$.

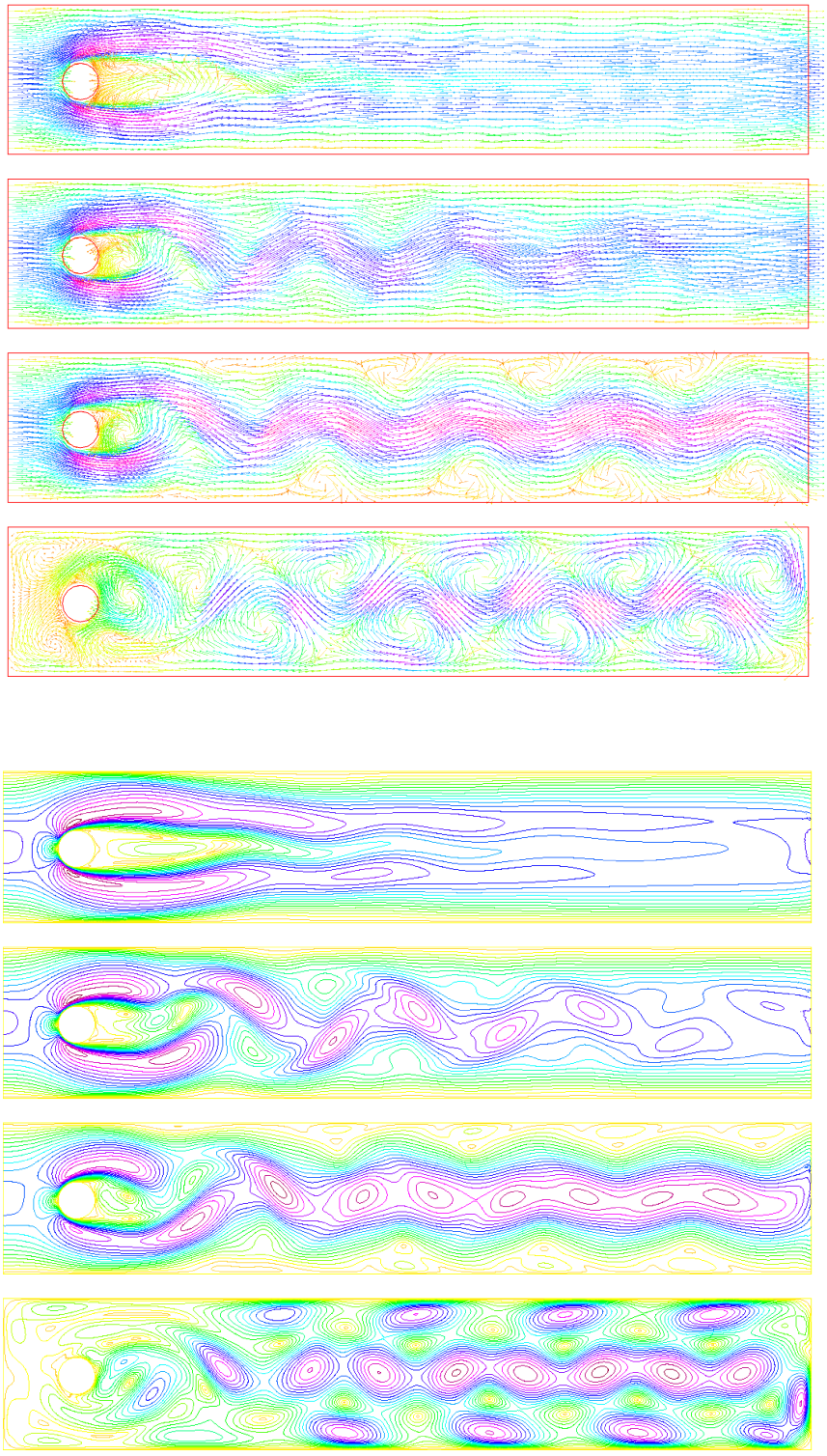

Fig. 5.3: Flow past cylinder: velocity fields and magnitudes at $t = 4, 5, 7, 8$, computed by the AC-SAV-CNLF scheme with $\Delta t = 0.001$.

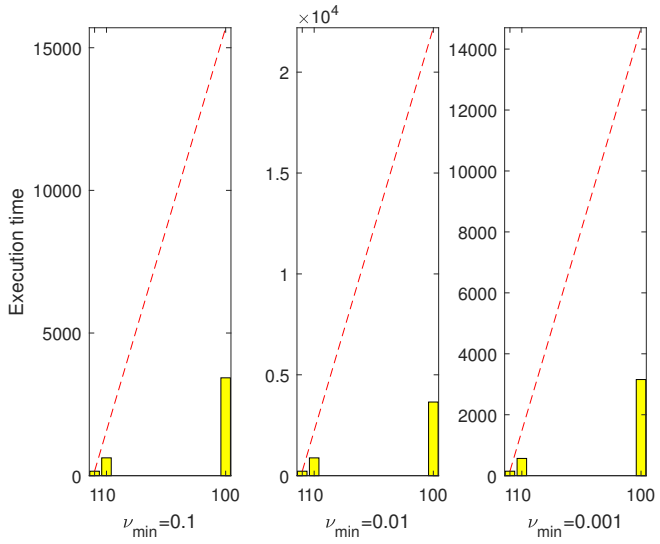

Fig. 5.4: 2D efficiency test: execution times of simulations with $J = 1, 10, 100$, and fixed $T = 0.5$, $h = \Delta t = 1/100$. The red line in dash refers to linear increase of CPU time with respect to the number J of samples.

reduced as compared to that of individual simulations for J flows. This thanks to the fact that all realizations in the ensemble method share a common constant operator in the left hand side, hence all the J simulations can be realized simutaneously.

3D performance. In this subsection, we observe the three-dimensional computational efficiency by simulating the Arnold-Beltrami-Childress flow introduced by Arnold [2] and Childress [7], which has been widely studied in the literature.

The Navier-Stokes flow is prescibed by an analytical solution

$$\begin{aligned} u_1 &= (\sin z + \cos y)e^{-\nu t}, \\ u_2 &= (\sin x + \cos z)e^{-\nu t}, \\ u_3 &= (\sin y + \cos x)e^{-\nu t}, \\ p &= -(\cos x \sin y + \sin x \cos z + \sin z \cos y)e^{-2\nu t}. \end{aligned}$$

For ensemble efficiency check, the number J of realizations will vary from 1 to 100. The viscosities ν_j are set as

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

with ϵ_j being a random variable uniformly distributed in $[0, 0.2]$. The numerical resolutions have fixed values $h = 1/16, \Delta t = 0.01$. The stabilization parameter α is set as $\alpha = 0$ for $\nu_{min} = 0.1$, and $\alpha = 0.1$ for smaller $\nu_{min} = 0.01$ or 0.001. To make a fair comparison with SAV-CNLF [34], the least-squares commutator preconditioner therein is solved by a multigrid V cycle.

Table 5.5 reports the execution times and expection of numerical errors $|\mathbb{E}[u_h - u]|_{H^1}$ at final time $T = 5$ in H^1 semi-norm computed by the AC-SAV-CNLF and SAV-CNLF schemes. A single realization, i.e. $J = 1$, is considered there. We have taken $\beta = 0.01, \epsilon = 0.25/\beta$ in AC-SAV-CNLF, ensuring accuracy comparable to those by SAV-CNLF. Table 5.5 shows that the AC-SAV-CNLF scheme outperforms SAV-CNLF since its execution consumes much less CPU time while maintaining comparable accuracy, which is similar to the observation in 2D efficiency test.

Table 5.5: 3D Arnold-Beltrami-Childress flow: numerical errors and execution times computed with $T = 5, h = 1/16, \Delta t = 0.01, J = 1$. Below $t_{\text{cpu,fixedmat}}$ denotes the execution time for assembling fixed matrices, $\bar{t}_{\text{cpu,sub1}}$ and $\bar{t}_{\text{cpu,sub2}}$ denote the averaged CPU times for solving subproblem 1 and 2 in one step respectively, $\bar{t}_{\text{cpu,1step}}$ denotes the average CPU time for one time step.

	ν_{min} (AC-SAV-CNLF scheme)			ν_{min} (SAV-CNLF scheme)		
	0.1	0.01	0.001	0.1	0.01	0.001
$t_{\text{cpu,fixedmat}}$	56.4s	60.9s	59.8s	20.2s	20.4s	20.3s
$\bar{t}_{\text{cpu,sub1}}$	2.1s	1.7s	1.5s	8.2s	5.3s	4.6s
$\bar{t}_{\text{cpu,sub2}}$	3.3s	4.4s	5.8s	15s	12s	12s
$\bar{t}_{\text{cpu,1step}}$	31.8s	32.9s	34s	51.3s	43s	42.7s
Total CPU time	16044s	16968s	17797s	26202s	21611s	21422s
$\lvert\mathbb{E}[u_h - u]\rvert_{H^1}$	2.29×10^{-4}	1.43×10^{-3}	6.88×10^{-3}	1.89×10^{-4}	1.66×10^{-3}	5.55×10^{-3}

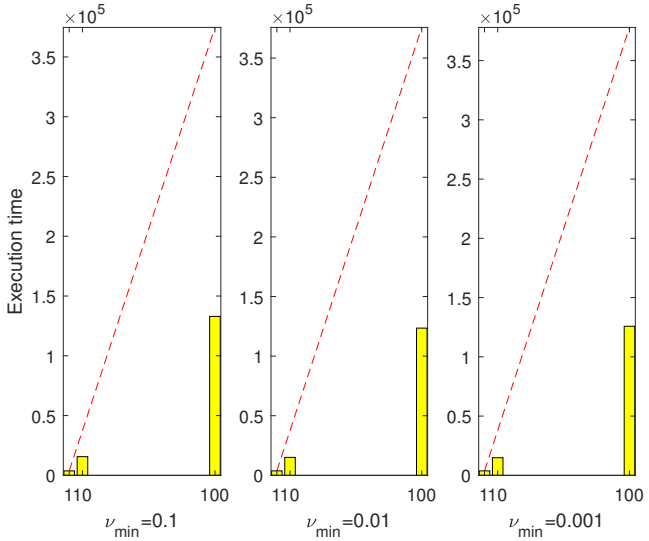

Fig. 5.5: 3D Arnold-Beltrami-Childress flow: execution times of simulations with $J = 1, 10, 100$, and fixed $T = 0.5$, $h = 1/16$, $\Delta t = 0.01$. The red line in dash refers to linear increase of execution time with respect to the number J of samples.

We then study the ensemble efficiency by simulating with different J values: $J = 1, 10, 100$. Similar to the 2D case, the execution times of performing the AC-SAV-CNLF ensemble algorithm with respect to the sample size J are plotted in Figure 5.5. From this figure, again we observe that the advantage of the AC-SAV-CNLF ensemble algorithm is apparent. Its better efficiency is more obvious when the sample size J is kind of large, since the execution time is significantly reduced as compared with CPU times of individual realizations predicted by the reference red line. This validates that the AC-SAV-CNLF ensemble scheme has an efficient scaling performance for ensemble computing.

5.4. Three-dimensional lid driven cavity flow. In this experiment, the three-dimensional lid driven cavity flow that has been widely studied in literatures [1, 52] will be simulated to illustrate the performance of the AC-SAV-CNLF scheme presented. Specifically, the flow in a cubic cavity $\Omega = (-0.5, 0.5)^3$ is driven by $(u_1, u_2, u_3) = (0, 1, 0)$ on the plane $x = -0.5$. No-slip boundary

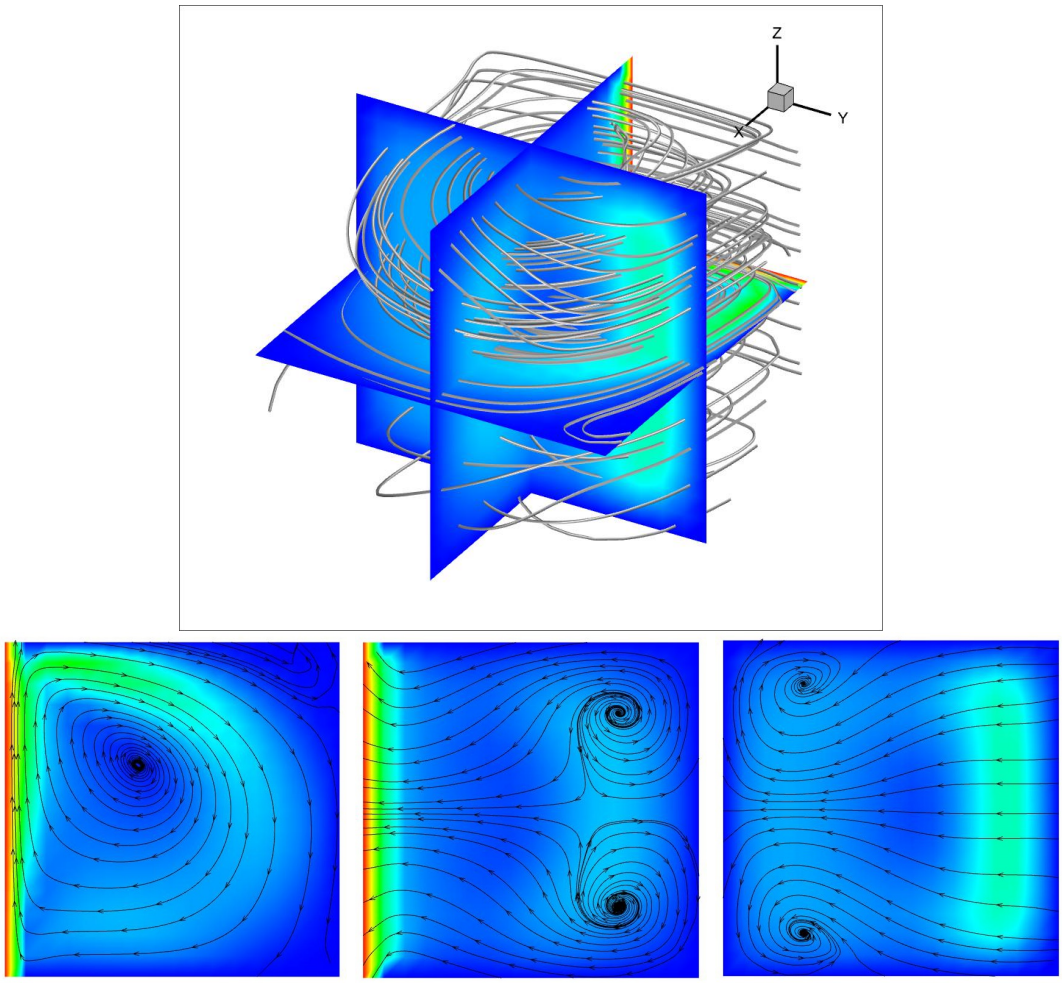

Fig. 5.6: 3D lid driven cavity flow with $Re = 400$ simulated by the AC-SAV-CNLF scheme. Top: streamlines and magnitude of velocity in 3D coordinates. Bottom: streamlines generated by velocities projected on the x-y plane, x-z plane, and y-z plane respectively.

conditions are imposed on the other parts of the cavity boundary. The initial condition and body force are set to be zero. Taking Reynolds number $Re = \frac{1}{\nu} = 400$ and $Re = 1000$ respecitively, we run simulations until $T = 40$, while the discretization reslutions are $h = 1/20$, $\Delta t = 0.02$.

In the top of Figure 5.6, the streamlines and magnitude of velocity in 3D coordinates are illustrated for the case $Re = 400$. Similarly, the top of Figure 5.7 plots simulations of the lid driven cavity flow with $Re = 1000$. All the flows are simulated by the AC-SAV-CNLF scheme, and almost idential to the solutions provided by the SAV-CNLF scheme (not illustrated here to keep the paper brief). In the bottom of these figures, streamlines are generated by velocities projected on the x-y plane, x-z plane, and y-z plane respectively.

6. Conclusions. We have presented an efficient, second order ensemble algorithm for numerically solving the Navier-Stokes equations in a UQ setting. The algorithm is based on the CNLF ensemble timestepping incorporating the AC technique and the recent SAV approach. It results in a constant operator on the left hand side of the system, shared by all realizations of the flow corresponding to different model parameters, so that efficient block linear solvers can be used to significantly improve

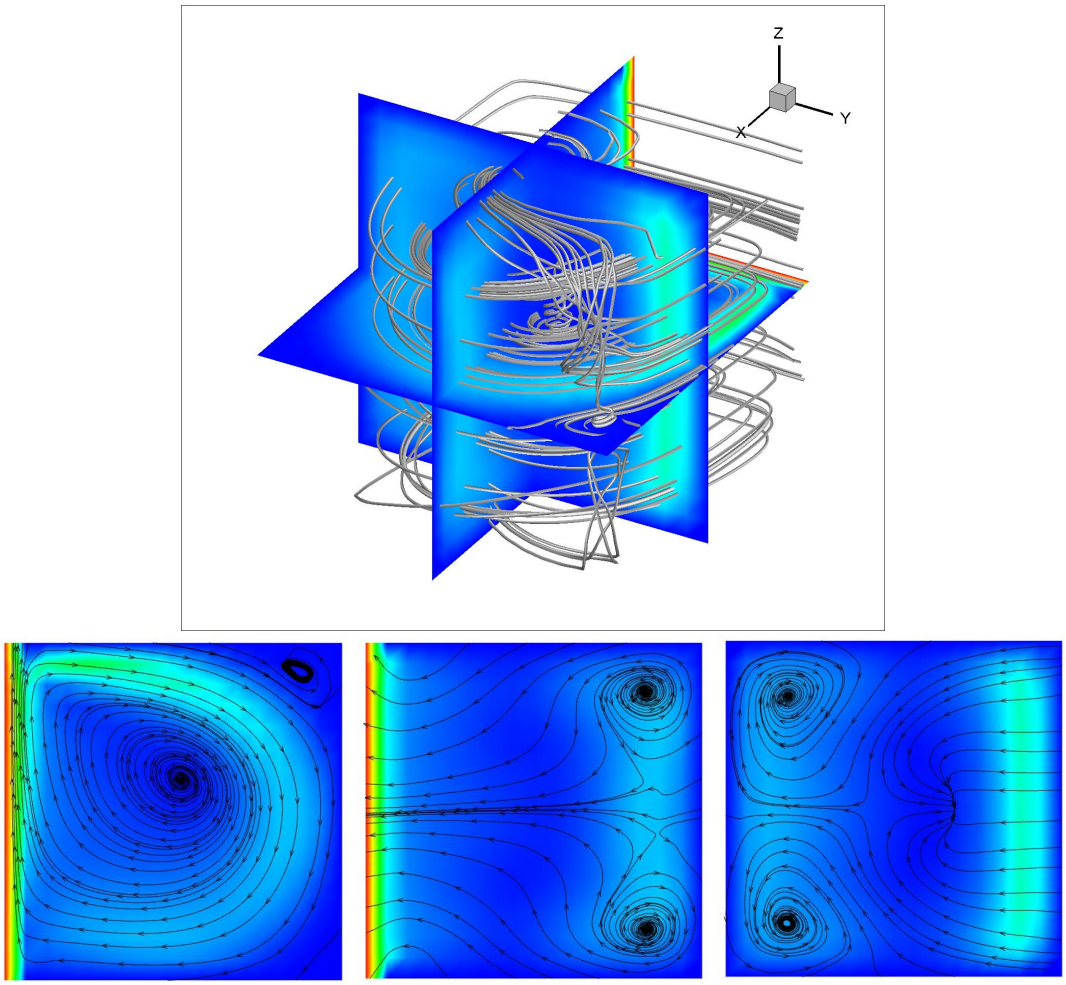

Fig. 5.7: 3D lid driven cavity flow with $Re = 1000$ simulated by the AC-SAV-CNLF scheme. Top: streamlines and magnitude of velocity in 3D coordinates. Bottom: streamlines generated by velocities projected on the x-y plane, x-z plane, and y-z plane respectively.

computational efficiency. In particular, the AC approach splits the computation of velocity and pressure resulting in smaller linear systems to be solved at each time step, and the pressure can be updated efficiently with no need to solve a Poisson equation. We have proved the proposed algorithm is long time stable without any timestep constraints. Ample numerical experiments were performed for various flow problems to illustrate that our ensemble algorithm is highly efficient and competitively accurate as compared with the pure SAV-CNLF scheme or individual realizations.

REFERENCES

[1] S. Albensoeder, H. Kuhlmann, *Accurate three-dimensional lid-driven cavity flow*, Journal of Computational Physics, 206 (2005) 536-558.

[2] V. Arnold, *Sur la topologic des ´ecoulements stationnaires des fluides parfaits*, Comptes Rendus Hebdomadaires des S´eances de l'Acad´emie des Sciences, 261 (1965), 17-20.

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

[4] J. Carter and N. Jiang, *Numerical analysis of a second order ensemble method for evolutionary magneto-*

hydrodynamics equations at small magnetic Reynolds number, Numerical Methods for Partial Differential Equations, 38 (2022), 1407-1436.

[5] 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.

[6] 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.

[7] S. Childress, *New solutions of the kinematic dynamo problem*, Journal of Mathematical Physics, 11 (1970), 3063-3076.

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

[9] 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.

[10] G. Evensen, *Data Assimilation - The Ensemble Kalman Filter*, Springer-Verlag, Berlin Heidelberg, 2009.

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

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

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

[14] 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.

[15] 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.

[16] 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.

[17] M. Gunzburger, N. Jiang and M. Schneier, *A higher-order ensemble/proper orthogonal decomposition method for the nonstationary Navier-Stokes equations*, International Journal of Numerical Analysis and Modeling, 15 (2018), 608-627.

[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] M. Gunzburger, N. Jiang and Z. Wang, *A second-order time-stepping scheme for simulating ensembles of parameterized flow problems*, Computational Methods in Applied Mathematics, 19 (2019), 681-701.

[20] 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.

[21] N. Hurl, W. Layton, Y. Li and C. Trenchea, *Stability analysis of the Crank-Nicolson-Leapfrog method with the Robert-Asselin-Williams time filter*, BIT Numerical Mathematics, 54 (2014), 1009-1021.

[22] R. Ingram, *A new linearly extrapolated Crank-Nicolson time-stepping scheme for the Navier-Stokes equations*, Mathematics of Computation, 82 (2013), 1953-1973.

[23] H. Ji and Y. Li, *A breakdown-free block conjugate gradient method*, BIT Numerical Mathematics, 57(2) (2017), 379-403.

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

[25] 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.

[26] N. Jiang, M. Kubacki, W. Layton, M. Moraiti and H. Tran, *A Crank-Nicolson Leapfrog stabilization: Unconditional stability and two applications*, Journal of Computational and Applied Mathematics, 281 (2015), 263-276.

[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 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.

[31] 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.

[32] N. Jiang and H. Tran, *Analysis of a Stabilized CNLF Method with Fast Slow Wave Splittings for Flow Problems*, Comput. Methods Appl. Math. 2015; 15 (3):307–330.

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

[34] 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.
[35] N. Jiang and H. Yang, *Artificial compressibility SAV ensemble algorithms for the incompressible Navier-Stokes equations*, Numerical Algorithms, in press, 2023.
[36] 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.
[37] M. Kubacki, *Uncoupling evolutionary groundwater-surface water flows using the Crank-Nicolson Leapfrog method*, Numerical Methods for Partial Differential Equations 29.4 (2013), 1192-1216.
[38] W. Layton and C. Trenchea, *Stability of two IMEX methods, CNLF and BDF2-AB2, for uncoupling systems of evolution equations*, Applied Numerical Mathematics, 62 (2012), 112-120.
[39] 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.
[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] W.J. Martin and M. Xue, *Initial condition sensitivity analysis of a mesoscale forecast using very-large ensembles*, Mon. Wea. Rev., 134 (2006), 192-207.
[43] J.F. McCarthy, *Block-conjugate-gradient method*, Physical Review D, 40 (1989), 2149.
[44] M. Mohebujjaman, *High order efficient algorithm for computation of MHD flow ensembles*, Advances in Applied Mathematics and Mechanics, 14 (2022), 1111-1137.
[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] M. Schäfer and S. Turek,*Benchmark computations of laminar flow around cylinder, in: Flow Simulation with HighPerformance Computers II*, Notes Numer. Fluid Mech. 52, Vieweg, Wiesbaden (1996), 547–566.
[48] 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.
[49] J. Shen, J. Xu and J. Yang, *The scalar auxiliary variable (SAV) approach for gradient flows*, Journal of Computational Physics, 353 (2018), 407-416.
[50] A. Takhirov, M. Neda, and J. Waters, *Time relaxation algorithm for flow ensembles*, Numerical Methods for Partial Differential Equations, 32 (2016), 757-777.
[51] 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.
[52] M. Tavelli and M. Dumbser, *A staggered space-time discontinuous Galerkin method for the three-dimensional incompressible Navier-Stokes equations on unstructured tetrahedral meshes*, Journal of Computational Physics, 319 (2016), pp.294-323.
[53] 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.