NUMERICAL INVESTIGATION OF TWO SECOND ORDER, STABILIZED SAV ENSEMBLE METHODS FOR THE NAVIER-STOKES EQUATIONS

NAN JIANG* AND HUANHUAN YANG †

Abstract. In this report we present a second order, stabilized SAV based, Crank–Nicolson leap-frog (CNLF) ensemble method, and perform a comprehensive numerical study of it as well as the Crank-Nicolson ensemble method with a linear extrapolation (CNLE) presented in [35]. Both methods are extremely efficient as only one linear system with multiple right hands needs to be solved at each time for a (potentially large) number of realizations of the flow problems. In particular the coefficient matrix of the fully discretized system is a constant matrix that does not change from one time step to another. We present extensive testing of these two methods and demonstrate the advantages of each. We also present long time stability analysis for both methods.

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

AMS subject classifications. 65C20, 65M12, 65M60, 76D05

1. Introduction.

Uncertainties are ubiquitous in the computations of fluid flow equations. A governing partial differential equation (PDE) system for a flow problem needs to be complemented with input data and parameters that specify physical characteristics of the simulated system. In many situations these input data, such as the initial conditions, body forces and model parameters, cannot be exactly specified due to limited experimental data available or the inherent variability of the system studied. It is thus important to study the impact of imprecise knowledge in the input data used to specify the system on the predicted response of the associated flow problem. This procedure is refereed to as uncertainty quantification (UQ) and has been extensively studied in the literature [3, 21, 22, 39, 48, 53] and applied for many practical problems. The main challenge for UQ is the high computational cost. For popular ensemble-based UQ methods, a potentially very large ensemble of samples of the uncertain parameter under consideration are usually needed to produce useful data and accurate predictions. But for complex flow problems, running the simulation once is already very expensive. Running the simulation a number of times for different parameter samples can be prohibitively expensive for many applications. To address this challenge, an efficient ensemble algorithm was developed in [27] to compute all realizations at one pass and lead to linear systems that can be computed efficiently using block solvers, e.g., block CG [46], block GMRES [12]. This fast algorithm has been extensively tested on different flow problems and shown to be fast and effective in predictive simulations, e.g., the Navier-Stokes equations [14–18, 23–25, 28, 38, 50, 51], MHD flows [33, 47], Boussinesq equations [9, 11, 25], heat equations [10, 44, 45], Stokes-Darcy equation [19, 29–32, 36], turbulent flows [7, 26].

One issue that arises in application of this algorithm for nonlinear flow problems is the fluctuation induced instability which leads to a time step condition that depends on the size of the fluctuation. A couple of fixes to this have been proposed in the literature such as numerical regularizations [23, 28] and time relaxation [50], which have relaxed the time step condition but did not fully eliminate the time step

[*] 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.

condition. A very recent paper [35] addressed this issue by incorporating a scalar auxiliary variable (SAV) approach with the ensemble timestepping idea, and devised ensemble schemes with provable long time stability without requiring any time step constraints. The proposed algorithms also have improved efficiency as the nonlinear term is made fully explicit and thus all discretized systems are sharing one constant coefficient matrix that does not change over time, as opposed to the original ensemble algorithms that have the coefficient matrix change from one time step to another. Two algorithms Stab-SAV-CN and Stab-SAV-BDF2 are proposed in [35], but analysis and the detailed decoupling strategy were only presented for Stab-SAV-BDF2 due to page limits. In this report, we provide detailed discussions of numerical implementation and analysis of the other algorithm Stab-SAV-CN. Moreover, we present a new second order SAV ensemble algorithm based on the Crank–Nicolson leap-frog timestepping, and compare the performance of both algorithms.

The paper is organized as follows. Two second order, stabilized SAV ensemble methods are introduced in Section 2. We prove in Section 3 that both methods are long time stable under two parameter conditions, without any time step conditions. The fully decoupled implementation algorithms for both methods are presented in Section 4. We will also provide detailed discussions on the algebraic systems after spatial discretization. Finally, the two methods are compared in Section 5 with various numerical tests. The second order accuracy in time is illustrated numerically with the Green-Taylor vortex solution. Efficiency tests as well as multiple flow problems are presented to demonstrate and compare the performance of both methods. Concluding remarks are reported in Section 6.

2. Second order SAV Ensemble Methods.

We consider the incompressible Navier–Stokes (NS) equations on a bounded domain $\Omega \subset \mathbb{R}^d$ with $d = 2$ or 3. Assume 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 $j = 1, ..., J$, have been generated with an ensemble-based UQ method and we need to solve the NS equations J times with different parameter samples to obtain flow data for UQ and predictions:

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

Here we assume $\nu_j(x) \in L^\infty(\Omega)$ and $\nu_j(x) \geq \nu_{j,min} > 0$.

The main difficulty in devising efficient and stable numerical methods for solving the Navier–Stokes equations lies in the nonlinear term that appears in the momentum equation. Implicit methods have good stability properties but are usually associated with high computational costs as iterative methods have to be used at each time step. Explicit methods are cheap but known for stability issues and to require a restrictive CFL condition. Semi-implicit methods appear as a compromise between the expensive fully implicit method and cheaper explicit methods that have stability issues, and have become increasingly popular in recent years. Nevertheless, the coefficient matrix resulting from a semi-implicit method is time-dependent and needs to be recomputed at each time step which prohibits significant improvement on efficiency. The recently developed SAV approach puts forth a new way to construct unconditionally stable methods that treat the nonlinear term fully explicitly and thus results in a coefficient matrix that is time-independent leading to significant savings in computational cost.

However the SAV approach was found to have low accuracy that compromises its unconditional stability in many test problems. [35] addressed this issue by introducing a stabilization term into the scheme. The idea is to add artificial viscosity to better condition the linear system and then add antidiffusion at previous time steps to avoid overdiffusion. It is shown in [35] that this stabilization method is able to substantially improve the accuracy of the SAV based numerical schemes.

We now present a new second order SAV ensemble method based on the Crank–Nicolson leap-frog (CNLF) timestepping, which is commonly used in atmospheric simulation codes. Following [35, 41, 43], we introduce the following scalar auxiliary variables $q_j(t)$, $j = 1, \cdots, J$:

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

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. These new variables and the associated differential equations will be added to the Navier–Stokes equations and form new governing systems for flow ensembles. The solutions to the new governing systems are equivalent to the solutions to the original Navier–Stokes equations, and the proposed ensemble methods are developed to solve the new governing systems. Taking derivative of $q_j(t)$ gives the following ordinary differential equation

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

The last two terms in the above equation are equal to zero since $\nabla\cdot u_j = 0$. The purpose of adding these two terms in the above equation is to cancel out the nonlinear term that comes from the momentum equation of the NS equations in the energy equation, and prove stability without assuming any time step conditions. Adding the above equation to the original NS equations, and putting the term $\frac{q_j(t)}{\sqrt{E(u_j)+\delta}}$, which is equal to 1, in front of the nonlinear term in the momentum equation yields a new governing system that is equivalent to (2.1):

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

The idea of ensemble timestepping is to split an uncertain parameter into two parts: mean + fluctuation, and lag the fluctuation term, which depends on the ensemble index j, to previous time steps so that it does not contribute to the common coefficient matrix. We thus define the ensemble mean $\bar{\nu}$ of the viscosity parameter and the ensemble fluctuations ν_j':

$$\bar{\nu}(x) := \frac{1}{J}\sum_{j=1}^{J}\nu_j(x), \qquad \nu_j'(x) := \nu_j(x) - \bar{\nu}(x).$$

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

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

We now propose our second order, stabilized SAV ensemble algorithm based on the CNLF timestepping as follows.

Algorithm 2.1 (Stab-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, q_j^3, for $n = 3, \cdots, N-1$, find $u_j^{n+1}, p_j^{n+1}, q_j^{n+1}$ satisfying

$$\begin{aligned}
(2.5) \quad & \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + \frac{\tilde{\tilde{q}}_j^n}{\sqrt{E(u_j^n) + \delta}} (u_j^n \cdot \nabla) u_j^n + \nabla \tilde{\tilde{p}}_j^n \\
& \quad - \nabla \cdot (\bar{\nu} \nabla \tilde{\tilde{u}}_j^n) - \nabla \cdot (\nu_j' \nabla (2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2})) - \frac{1}{2}\alpha h \Delta (u_j^{n+1} - u_j^{n-1}) = f_j^n, \\
(2.6) \quad & \nabla \cdot u_j^{n+1} = 0, \\
(2.7) \quad & \frac{q_j^{n+1} - q_j^{n-1}}{2\Delta t} = \frac{1}{2\tilde{\tilde{q}}_j^n} \int_\Omega \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} \cdot \tilde{\tilde{u}}_j^n \, dx \\
& \quad + \frac{1}{2\sqrt{E(u_j^n) + \delta}} \int_\Omega (u_j^n \cdot \nabla) u_j^n \cdot \tilde{\tilde{u}}_j^n \, dx - \frac{b_j^n}{2\tilde{\tilde{q}}_j^n},
\end{aligned}$$

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

This is a four-step method. One needs to know u_j^{n-3}, u_j^{n-2}, u_j^{n-1}, and u_j^n to compute u_j^{n+1}. We will show in an equivalent fully decoupled implementation scheme in Section 4 that all realizations are sharing the same coefficient matrix, and thanks to the SAV idea, the nonlinear term can be treated fully explicitly and thus the shared coefficient matrix is also constant. The stabilization $-\frac{1}{2}\alpha h \Delta(u_j^{n+1} - u_j^{n-1})$ is added in the algorithm following the argument in [35] to increase the accuracy and stability of the algorithm. We will prove this algorithm is long time stable without any time step constraints in Section 3.

In [35] the following Stab-SAV-CNLE ensemble algorithm was presented without giving any discussions on implementation or the corresponding stability analysis due to page limits. We will provide a detailed study of this algorithm in this report, and compare its performance with our proposed Stab-SAV-CNLF method. Denote

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

The second order, stabilized SAV ensemble algorithm based on the Crank–Nicolson timestepping with a linear extrapolation proposed in [35] is given by

Algorithm 2.2 (Stab-SAV-CNLE). For $j = 1, 2, \cdots, J$, given $u_j^0, u_j^1, u_j^2, p_j^2, q_j^2$, for $n = 2, 3, \cdots, N-1$, find $u_j^{n+1}, p_j^{n+1}, q_j^{n+1}$ satisfying

$$(2.8) \qquad \frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{q_j^{n+1/2}}{\sqrt{E(\tilde{u}_j^{n+1/2}) + \delta}} (\tilde{u}_j^{n+1/2} \cdot \nabla) \tilde{u}_j^{n+1/2} + \nabla p_j^{n+1/2}$$

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

$$\nabla \cdot u_j^{n+1} = 0, \tag{2.9}$$

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = \frac{1}{2 q_j^{n+1/2}} \int_\Omega \frac{u_j^{n+1} - u_j^n}{\Delta t} \cdot u_j^{n+1/2} \, dx \tag{2.10}$$
$$+ \frac{1}{2\sqrt{E(\tilde{u}_j^{n+1/2}) + \delta}} \int_\Omega (\tilde{u}_j^{n+1/2} \cdot \nabla) \tilde{u}_j^{n+1/2} \cdot u_j^{n+1/2} \, dx - \frac{b_j^{n+1/2}}{2 q_j^{n+1/2}},$$

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

This is a three-step method, and second order in time convergent. Similar to Stab-SAV-CNLF, as will be shown in Section 4, all realizations share a common constant coefficient matrix, and we will prove in Section 3 this algorithm is also long time stable without any time step constraints.

3. Long Time Stability of the SAV Ensemble Algorithms.

In this section we prove the long stability of both algorithms presented in Section 2, under the same parameter condition, without any time step conditions. The parameter condition limits the size of the fluctuation to be smaller than one third of the size of the uncertain parameter. This is usually easy to satisfy as in UQ applications the fluctuation is generally very small. We define the minimum average $\bar{\nu}_{min}$ and maximum fluctuation ν'_{max} of the kinematic viscosity:

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

which will be used in the following proofs.

We denote the $L^2(\Omega)$ norm by $\| \cdot \|$ and the usual L^2 inner product by $(\cdot, \cdot)$. Let X denote the velocity space $(d = 2, 3)$:

$$X : = H_0^1(\Omega)^d = \left\{ v \in L^2(\Omega)^d : \nabla v \in L^2(\Omega)^{d \times d} \text{ and } v = 0 \text{ on } \partial\Omega \right\}.$$

The norm on the dual space of X is defined by

$$\|f\|_{-1} = \sup_{0 \neq v \in X} \frac{(f, v)}{\|\nabla v\|} \ .$$

We first present the stability proof for Stab-SAV-CNLF. In the analysis we assume q_j^n is real and thus $|q_j^n|$ is positive. If in the simulation the numerical solution of q_j^n becomes complex, then the simulation fails in the sense that the linear solver will fail and the numerical solution will become inaccurate. Nevertheless, it is demonstrated both in [35] and Section 5 in this report that the stabilization we add can effectively prevent q_j^n from becoming complex and improve both accuracy and stability of the SAV based methods.

Theorem 3.1 (Long Time Stability of Stab-SAV-CNLF). *Assume q_j^n is real, for any $n = 0, \cdots, N$, $j = 1, \cdots, J$, and the following parameter fluctuation condition holds*

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

With homogeneous Dirichlet boundary condition, Algorithm 2.1 is nonlinearly, long time stable, and the following energy inequality holds

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

Proof. Taking the L^2 inner product of (2.5) with $\tilde{\tilde{u}}_j^n$ and using (2.6) gives

(3.3)
$$\begin{aligned} &\left(\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t}, \tilde{\tilde{u}}_j^n\right) + \frac{\tilde{\tilde{q}}_j^n}{\sqrt{E(u_j^n) + \delta}}\left((u_j^n \cdot \nabla)u_j^n, \tilde{\tilde{u}}_j^n\right) + \int_{\partial\Omega}(\vec{n}\cdot\tilde{\tilde{u}}_j^n)\tilde{\tilde{p}}_j^n d\sigma \\ &- \int_{\partial\Omega}(\vec{n}\cdot\bar{\nu}\nabla\tilde{\tilde{u}}_j^n)\cdot\tilde{\tilde{u}}_j^n d\sigma \\ &+ \|\bar{\nu}^{\frac{1}{2}}\nabla\tilde{\tilde{u}}_j^n\|^2 - \int_{\partial\Omega}(\vec{n}\cdot\nu'_j\nabla(2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2})\cdot\tilde{\tilde{u}}_j^n d\sigma + (\nu'_j\nabla(2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2}), \nabla\tilde{\tilde{u}}_j^n) \\ &- \frac{1}{2}\alpha h\int_{\partial\Omega}(\vec{n}\cdot(\nabla u_j^{n+1} - \nabla u_j^{n-1}))\cdot\tilde{\tilde{u}}_j^n d\sigma + \frac{1}{4}\alpha h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right) \\ &= (f_j^n, \tilde{\tilde{u}}_j^n). \end{aligned}$$

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

(3.4)
$$\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) = \left(\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t}, \tilde{\tilde{u}}_j^n\right) + \frac{\tilde{\tilde{q}}_j^n}{\sqrt{E(u_j^n) + \delta}}\left((u_j^n\cdot\nabla)u_j^n, \tilde{\tilde{u}}_j^n\right) - b_j^n.$$

Adding (3.3) and (3.4) gives

$$\begin{aligned} &\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) + \|\bar{\nu}^{\frac{1}{2}}\nabla\tilde{\tilde{u}}_j^n\|^2 + \frac{1}{4}\alpha h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right) \\ &= (f_j^n, \tilde{\tilde{u}}_j^n) - (\nu'_j\nabla(2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2}), \nabla\tilde{\tilde{u}}_j^n) - \int_{\partial\Omega}(\vec{n}\cdot\tilde{\tilde{u}}_j^n)\tilde{\tilde{p}}_j^n d\sigma \\ &\quad + \int_{\partial\Omega}(\vec{n}\cdot\bar{\nu}\nabla\tilde{\tilde{u}}_j^n)\cdot\tilde{\tilde{u}}_j^n d\sigma + \int_{\partial\Omega}(\vec{n}\cdot\nu'_j\nabla(2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2}))\cdot\tilde{\tilde{u}}_j^n d\sigma \\ &\quad + \frac{1}{2}\alpha h\int_{\partial\Omega}(\vec{n}\cdot(\nabla u_j^{n+1} - \nabla u_j^{n-1}))\cdot\tilde{\tilde{u}}_j^n d\sigma - b_j^n. \end{aligned} \tag{3.5}$$

In particular, with homogeneous Dirichlet condition the terms of integrals on the boundary in (3.5) are null. Note that $\bar{\nu}(x) \geq \bar{\nu}_{min} > 0$. Applying Cauchy-Schwarz and Young's inequalities to the right hand side and using $(2a-b)^2 \leq 6a^2 + 3b^2$ gives, for any $\beta > 0, \epsilon > 0$,

$$\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) + \|\bar{\nu}^{\frac{1}{2}}\nabla\tilde{\tilde{u}}_j^n\|^2 + \frac{1}{4}\alpha h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right)$$

(3.6)

$$\begin{aligned}
&\le \|f_j^n\|_{-1}\|\nabla\tilde{\tilde{u}}_j^n\| + \nu'_{max}\|\nabla(2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2})\|\|\nabla\tilde{\tilde{u}}_j^n\| \\
&\le \beta\bar{\nu}_{min}\|\nabla\tilde{\tilde{u}}_j^n\|^2 + \frac{1}{4\beta\bar{\nu}_{min}}\|f_j^n\|_{-1}^2 + \frac{\epsilon\nu'_{max}}{2}\|\nabla\tilde{\tilde{u}}_j^n\|^2 + \frac{\nu'_{max}}{2\epsilon}\|\nabla(2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2})\|^2 \\
&\le \beta\bar{\nu}_{min}\|\nabla\tilde{\tilde{u}}_j^n\|^2 + \frac{1}{4\beta\bar{\nu}_{min}}\|f_j^n\|_{-1}^2 + \frac{\epsilon\nu'_{max}}{2}\|\nabla\tilde{\tilde{u}}_j^n\|^2 \\
&\qquad + \frac{3\nu'_{max}}{\epsilon}\|\nabla\tilde{\tilde{u}}_j^{n-1}\|^2 + \frac{3\nu'_{max}}{2\epsilon}\|\nabla\tilde{\tilde{u}}_j^{n-2}\|^2.
\end{aligned}$$

As the last three terms all need to be bounded by $\bar{\nu}_{min}\|\nabla\tilde{\tilde{u}}_j^n\|^2$, we want to minimize $\frac{\epsilon}{2} + \frac{3}{\epsilon} + \frac{3}{2\epsilon}$ by taking $\epsilon = 3$. (3.6) then reduces to

$$\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) + \frac{3}{2}\nu'_{max}\left(\|\nabla\tilde{\tilde{u}}_j^n\|^2 - \|\nabla\tilde{\tilde{u}}_j^{n-1}\|^2\right) \\
&\quad + \frac{1}{2}\nu'_{max}\left(\|\nabla\tilde{\tilde{u}}_j^{n-1}\|^2 - \|\nabla\tilde{\tilde{u}}_j^{n-2}\|^2\right) + \frac{1}{4}\alpha h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right) \\
&\quad + \left((1-\beta)\bar{\nu}_{min} - 3\nu'_{max}\right)\|\nabla\tilde{\tilde{u}}_j^n\|^2 \le \frac{1}{4\beta\bar{\nu}_{min}}\|f_j^n\|_{-1}^2.
\end{aligned} \tag{3.7}$$

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

$$(1-\beta)\bar{\nu}_{min} - 3\nu'_{max} = (\frac{1}{2} + \frac{3}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}})\bar{\nu}_{min} - 3\nu'_{max} = \frac{1}{2}\left(\bar{\nu}_{min} - 3\nu'_{max}\right) > 0. \tag{3.8}$$

(3.7) can then be reduced to

$$\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2 - |q_j^{n-1}|^2\right) + \frac{3}{2}\nu'_{max}\left(\|\nabla\tilde{\tilde{u}}_j^n\|^2 - \|\nabla\tilde{\tilde{u}}_j^{n-1}\|^2\right) \\
&\quad + \frac{1}{2}\nu'_{max}\left(\|\nabla\tilde{\tilde{u}}_j^{n-1}\|^2 - \|\nabla\tilde{\tilde{u}}_j^{n-2}\|^2\right) \\
&\quad + \frac{1}{4}\alpha h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^{n-1}\|^2\right) \le \frac{1}{2(\bar{\nu}_{min} - 3\nu'_{max})}\|f_j^n\|_{-1}^2.
\end{aligned} \tag{3.9}$$

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

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

□

We next present the stability proof for Stab-SAV-CNLE under the same parameter condition.

Theorem 3.2 (Long Time Stability of Stab-SAV-CNLE). *Assume q_j^n is real, for any $n = 0, \cdots, N$, $j = 1, \cdots, J$, and the following parameter fluctuation condition holds*

$$\frac{\nu'_{max}}{\bar{\nu}_{min}} < \frac{1}{3}. \tag{3.10}$$

With homogeneous Dirichlet boundary condition, Algorithm 2.2 is nonlinearly, long time stable, and the following energy inequality holds

$$\begin{aligned}|q_j^N|^2 &+ \frac{3}{2}\nu'_{max}\Delta t\|\nabla u_j^{N-1/2}\|^2 + \frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^{N-3/2}\|^2 + \frac{\alpha}{2}h\Delta t\|\nabla u_j^N\|^2 \\ &\leq |q_j^2|^2 + \frac{3}{2}\nu'_{max}\Delta t\|\nabla u_j^{3/2}\|^2 + \frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^{1/2}\|^2 + \frac{\alpha}{2}h\Delta t\|\nabla u_j^2\|^2 \\ &\qquad + \frac{\Delta t}{2(\bar{\nu}_{min} - 3\nu'_{max})}\sum_{n=2}^{N-1}\|f_j^{n+1/2}\|_{-1}^2.\end{aligned} \tag{3.11}$$

Proof. Taking the L^2 inner product of (2.8) with $u_j^{n+1/2}$, multiplying (2.10) with $2q_j^{n+1/2}$, and adding the two equations gives

$$\begin{aligned}&\frac{1}{\Delta t}\left(|q_j^{n+1}|^2 - |q_j^n|^2\right) + \|\bar{\nu}^{\frac{1}{2}}\nabla u_j^{n+1/2}\|^2 + \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^n\|^2\right) \\ &= (f_j^{n+1/2}, u_j^{n+1/2}) - (\nu'_j\nabla\tilde{u}_j^{n+1/2}, \nabla u_j^{n+1/2}) - \int_{\partial\Omega}(\vec{n}\cdot u_j^{n+1/2})p_j^{n+1/2}d\sigma \\ &\quad + \int_{\partial\Omega}(\vec{n}\cdot\bar{\nu}\nabla u_j^{n+1/2})\cdot u_j^{n+1/2}d\sigma + \int_{\partial\Omega}(\vec{n}\cdot\nu'_j\nabla\tilde{u}_j^{n+1/2})\cdot u_j^{n+1/2}d\sigma \\ &\quad + \alpha h\int_{\partial\Omega}(\vec{n}\cdot(\nabla u_j^{n+1} - \nabla u_j^n))\cdot u_j^{n+1/2}d\sigma - b_j^{n+1/2}.\end{aligned} \tag{3.12}$$

Assuming homogeneous Dirichlet condition, following the proof of Theorem 3.1, we have for any $\beta > 0$,

$$\begin{aligned}&\frac{1}{\Delta t}\left(|q_j^{n+1}|^2 - |q_j^n|^2\right) + \frac{3}{2}\nu'_{max}\left(\|\nabla u_j^{n+1/2}\|^2 - \|\nabla u_j^{n-1/2}\|^2\right) \\ &+ \frac{1}{2}\nu'_{max}\left(\|\nabla u_j^{n-1/2}\|^2 - \|\nabla u_j^{n-3/2}\|^2\right) + \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^n\|^2\right) \\ &+ ((1-\beta)\bar{\nu}_{min} - 3\nu'_{max})\|\nabla u_j^{n+1/2}\|^2 \leq \frac{1}{4\beta\bar{\nu}_{min}}\|f_j^{n+1/2}\|_{-1}^2.\end{aligned} \tag{3.13}$$

If the parameter fluctuation condition is satisfied, (3.13) can be reduced to

$$\begin{aligned}&\frac{1}{\Delta t}\left(|q_j^{n+1}|^2 - |q_j^n|^2\right) + \frac{3}{2}\nu'_{max}\left(\|\nabla u_j^{n+1/2}\|^2 - \|\nabla u_j^{n-1/2}\|^2\right) \\ &+ \frac{1}{2}\nu'_{max}\left(\|\nabla u_j^{n-1/2}\|^2 - \|\nabla u_j^{n-3/2}\|^2\right) \\ &+ \frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^n\|^2\right) \leq \frac{1}{2(\bar{\nu}_{min} - 3\nu'_{max})}\|f_j^{n+1/2}\|_{-1}^2.\end{aligned} \tag{3.14}$$

□

Summing up from $n = 2$ to $n = N-1$ and multiplying through by Δt yields (3.11).

4. Numerical Implementation.

In this section we discuss the details of numerical implementation of the proposed two ensemble methods.

4.1. Fully Decoupled Implementation Algorithms.

The two algorithms Stab-SAV-CNLF and Stab-SAV-CNLE given by (2.5)-(2.7) and (2.8)-(2.10) are coupled systems of u, p, q, which needs further decoupling for fast computation. Following the decoupling strategy in [35] we will derive fully decoupled implementation algorithms for both algorithms.

We first drive the implementation algorithm for the Stab-SAV-CNLF method. Define a new scalar S_j^{n+1} by

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

Then we can decompose the numerical solution (u_j^{n+1}, p_j^{n+1}) into two parts: $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}$, $p_j^{n+1} = \hat{p}_j^{n+1} + S_j^{n+1}\breve{p}_j^{n+1}$. Substituting (4.1) into (2.5)-(2.7) and grouping the terms with S_j^{n+1} and those without S_j^{n+1}, we can derive two separate PDE systems for $(\hat{u}_j^{n+1}, \hat{p}_j^{n+1})$, $(\breve{u}_j^{n+1}, \breve{p}_j^{n+1})$, which are equivalent to (2.5)-(2.6).

(Stab-SAV-CNLF sub-problem 1)

$$\begin{cases} \frac{1}{2\Delta t}\hat{u}_j^{n+1} - \frac{1}{2}\nabla\cdot(\bar{\nu}\nabla\hat{u}_j^{n+1}) - \frac{1}{2}\alpha h\Delta\hat{u}_j^{n+1} + \frac{1}{2}\nabla\hat{p}_j^{n+1} \\ \quad = f_j^n + \frac{1}{2\Delta t}u_j^{n-1} + \frac{1}{2}\nabla\cdot(\bar{\nu}\nabla u_j^{n-1}) - \frac{1}{2}\alpha h\Delta u_j^{n-1} \\ \qquad\qquad + \nabla\cdot(\nu_j'\nabla(2\tilde{\tilde{u}}_j^{n-1} - \tilde{\tilde{u}}_j^{n-2})) - \frac{1}{2}\nabla p_j^{n-1}, \text{ in } \Omega \\ \nabla\cdot\hat{u}_j^{n+1} = 0, \text{ in } \Omega \\ \hat{u}_j^{n+1} = g_j^{n+1}, \text{ on } \partial\Omega. \end{cases}$$

(Stab-SAV-CNLF sub-problem 2)

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

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

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

Plugging this expression of q_j^{n+1} into (3.4) gives

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

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

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

where

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

For both sub-problem 1 and sub-problem 2, all realizations have the same constant coefficient matrix. The sub-problem 3 is a scalar equation for each realization, which can be solved quickly without incurring much increase in the computational cost. After obtaining $\hat{u}_j^{n+1}, \breve{u}_j^{n+1}$, and S_j^{n+1}, we recover our numerical solution (u_j^{n+1}, p_j^{n+1}) by the formula $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1},\ p_j^{n+1} = \hat{p}_j^{n+1} + S_j^{n+1}\breve{p}_j^{n+1}$.

Due to page limits, the implementation algorithm of Stab-SAV-CNLE was not presented in [35]. We will present it here.

Similar to the derivation of the fully decoupled implement algorithm for Stab-SAV-CNLF, we define a new scalar S_j^{n+1} by

$$
S_j^{n+1} = \frac{q_j^{n+1/2}}{\sqrt{E(\tilde{u}_j^{n+1/2})+\delta}}, \tag{4.3}
$$

and decompose the numerical solution (u_j^{n+1}, p_j^{n+1}) into two parts: $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1},\ p_j^{n+1} = \hat{p}_j^{n+1} + S_j^{n+1}\breve{p}_j^{n+1}$. Substituting (4.3) into (2.8)-(2.10) and grouping the terms with S_j^{n+1} and those without S_j^{n+1}, we can derive two separate PDE systems for $(\hat{u}_j^{n+1}, \hat{p}_j^{n+1})$, $(\breve{u}_j^{n+1}, \breve{p}_j^{n+1})$. The two subproblems that are equivalent to (2.8)-(2.9) are given by

(Stab-SAV-CNLE sub-problem 1)

$$
\begin{cases}
\frac{1}{\Delta t}\hat{u}_j^{n+1} - \frac{1}{2}\nabla\cdot(\bar{\nu}\nabla\hat{u}_j^{n+1}) - \alpha h\Delta\hat{u}_j^{n+1} + \frac{1}{2}\nabla\hat{p}_j^{n+1}\\
\quad = f_j^{n+1/2} + \frac{1}{\Delta t}u_j^n + \frac{1}{2}\nabla\cdot(\bar{\nu}\nabla u_j^n) - \alpha h\Delta u_j^n + \nabla\cdot(\nu_j'\nabla\tilde{u}_j^{n+1/2}) - \frac{1}{2}\nabla p_j^n, \text{ in } \Omega\\
\nabla\cdot\hat{u}_j^{n+1} = 0, \text{ in } \Omega\\
\hat{u}_j^{n+1} = g_j^{n+1}, \text{ on } \partial\Omega.
\end{cases}
$$

(Stab-SAV-CNLE sub-problem 2)

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

Plugging

$$(4.4)\quad S_j^{n+1}=\frac{q_j^{n+1}+q_j^n}{2\sqrt{E(\tilde{u}_j^{n+1/2})+\delta}}\qquad\Longrightarrow\qquad q_j^{n+1}=2\sqrt{E(\tilde{u}_j^{n+1/2})+\delta}\,S_j^{n+1}-q_j^n,$$

into (2.10)$\cdot 2q_j^{n+1/2}$ gives

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

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

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

where

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

4.2. Algebraic Systems after spatial discretization. Let $S_h^2(\Omega)^2$ and $S_h^1(\Omega)$ denote the spaces of Taylor-Hood elements (P2-P1) on Ω for velocity u and pressure p, their basis functions are $\{\chi_j^u\}_{j=1}^{N_u}$, $\{\chi_j^p\}_{j=1}^{N_p}$ respectively. Bold vectors will be used to denote the discrete finite element solutions. Superscript n and subscript j are applied to represent time step and sample index. Denote the velocity mass matrix and velocity stiffness matrix of the Poisson operator by $\mathbf{M}_{uu}$ and $\mathbf{S}_{uu}$ respectively. We also define matrices $\mathbf{D}_{uup}$, $\mathbf{S}(\nu)$ and $\mathbf{N}(u)$ whose entries are given as follows.

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

The proposed schemes in this study will be compared with some non-ensemble schemes:

Algorithm 4.1 (CNLF-nonensemble).

$$\begin{cases} \dfrac{u_j^{n+1}-u_j^{n-1}}{2\Delta t}+(u_j^n\cdot\nabla)\tilde{\tilde{u}}_j^n+\nabla\tilde{\tilde{p}}_j^n-\nabla\cdot(\nu_j\nabla\tilde{\tilde{u}}_j^n)=f_j^n,\\ \nabla\cdot u_j^{n+1}=0. \end{cases}$$

Algorithm 4.2 (CNLE-nonensemble).

$$\begin{cases} \dfrac{u_j^{n+1}-u_j^{n}}{\Delta t}+(\tilde{u}_j^{n+1/2}\cdot\nabla)u_j^{n+1/2}+\nabla p_j^{n+1/2}-\nabla\cdot(\nu_j\nabla u_j^{n+1/2})=f_j^{n+1/2},\\ \nabla\cdot u_j^{n+1}=0. \end{cases}$$

We then state the algebraic systems of different numerical algorithms, for sample $j=1,\cdots,J$, which will be considered in the ensemble efficiency testing.

1. Stab-SAV-CNLF ensemble:

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

with

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

$$\mathbf{b}_j^{n+1}=\mathbf{f}_j^n+\tfrac{1}{2\Delta t}\mathbf{M}_{uu}\mathbf{u}_j^{n-1}-\mathbf{S}(\tfrac{1}{2}\bar{\nu}-\tfrac{1}{2}\alpha h)\mathbf{u}_j^{n-1}-\mathbf{S}(\nu_j')(2\tilde{\tilde{\mathbf{u}}}_j^{n-1}-\tilde{\tilde{\mathbf{u}}}_j^{n-2})+\tfrac{1}{2}\mathbf{D}_{uup}\mathbf{p}_j^{n-1},$$

$$\mathbf{c}_j^{n+1}=-\mathbf{N}(u_j^n)\mathbf{u}_j^n.$$

2. Stab-SAV-CNLE ensemble:

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

with

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

$$\mathbf{b}_j^{n+1}=\mathbf{f}_j^{n+1/2}+\tfrac{1}{\Delta t}\mathbf{M}_{uu}\mathbf{u}_j^{n}-\mathbf{S}(\tfrac{1}{2}\bar{\nu}-\alpha h)\mathbf{u}_j^{n}-\mathbf{S}(\nu_j')\tilde{\mathbf{u}}_j^{n+1/2}+\tfrac{1}{2}\mathbf{D}_{uup}\mathbf{p}_j^{n},$$

$$\mathbf{c}_j^{n+1}=-\mathbf{N}(\tilde{u}_j^{n+1/2})\tilde{\mathbf{u}}_j^{n+1/2}.$$

3. CNLF nonensemble:

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

with

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

$$\mathbf{b}_j^{n+1} = \mathbf{f}_j^n + \tfrac{1}{2\Delta t}\mathbf{M}_{uu}\mathbf{u}_j^{n-1} - \tfrac{1}{2}\mathbf{S}(\nu_j)\mathbf{u}_j^{n-1} + \tfrac{1}{2}\mathbf{D}_{uup}\mathbf{p}_j^{n-1} - \tfrac{1}{2}\mathbf{N}(u_j^n)\mathbf{u}_j^{n-1}.$$

4. CNLE nonensemble:

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

with

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

$$\mathbf{b}_j^{n+1} = \mathbf{f}_j^{n+1/2} + \tfrac{1}{\Delta t}\mathbf{M}_{uu}\mathbf{u}_j^n - \tfrac{1}{2}\mathbf{S}(\nu_j)\mathbf{u}_j^n + \tfrac{1}{2}\mathbf{D}_{uup}\mathbf{p}_j^n - \tfrac{1}{2}\mathbf{N}(\tilde{u}_j^{n+1/2})\mathbf{u}_j^n.$$

It is apparent that the matrices $\mathbf{A}_{\text{savcnlf}}$ and $\mathbf{A}_{\text{savcnle}}$ in ensemble methods are common among different samples, thus one can simultaneously compute all realizations by solving a single linear system with multiple right hand sides (RHSs) corresponding to different samples; but $\mathbf{A}_{\text{cnlf}}^{(n,j)}$ and $\mathbf{A}_{\text{cnle}}^{(n,j)}$ in the classical nonensemble methods change over time index n and sample index j, so we need to achieve J realizations one by one.

In large-scale applications, the GMRES linear solver can be applied to handle the nonensemble schemes. As for the ensemble schemes, a block iterative solver should be considered to remove redundant information due to linear dependence of multiple residuals. In particular, we resort to the block GMRES algorithm with deflation [5, BFGMRESD(m)]. The least-square commutator precontioning [8] is a competitive choice to speeding up the convergence of GMRES. This has also been mentioned in [35]. In addition, for ensemble schemes, this preconditioner can be solved by the block CG or block GMRES algorithm with an ILU or multigrid preconditioner.

5. Numerical Experiments.

In this section, we perform numerical experiments to validate the stability, accuracy, and efficiency of the Stab-SAV-CNLE and Stab-SAV-CNLF ensemble algorithms.

5.1. Tests for convergence rate.

To validate the convergence rate of Stab-SAV-CNLF and Stab-SAV-CNLE, we consider a simple problem [13] with Green-Taylor vortex solution on a square domain $\Omega = (0,1)^2$. The analytical solution of the Navier–Stokes equations (NSE) is given by

$$u_{\text{true}} = (-\cos x \sin y, \sin x \cos y)^T m(t), \quad p_{\text{true}} = -\tfrac{1}{4}[\cos(2x) + \cos(2y)]m(t)^2,$$
$$f(x,y,t) = [m'(t) + 2\nu m(t)](-\cos x \sin y, \sin x \cos y)^T,$$

with $m(t) = e^{\nu}\cos(2t)$. The initial condition and Dirichlet boundary condition are then set to be consistent with the analytical solution. We will compute $J = 3$ realizations simutaneously by the ensemble schemes, in which the flows correspond to

$$\nu_j = \nu_{min}(1 + \epsilon_j),\ \epsilon_j = 0.1(j-1),\ j = 1, \cdots, J.$$

In this setup, we have three groups of different initial conditions, boundary conditions, and body forces.

Since stabilized SAV approaches are designed with the hope of good performance for relatively large Reynolds numbers, we consider $\nu_{min} = 0.0005$ in our first test

Table 1: Errors at $T = 5$ and convergence rates of the Stab-SAV-CNLE ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0.5$, $\nu_{min} = 0.0005$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,3}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,3}$	Rate
1/8	1.98×10^{-1}	—	1.34×10^{-2}	—	1.87×10^{-1}		1.33×10^{-2}	—
1/16	5.16×10^{-2}	1.94	3.25×10^{-3}	2.04	4.68×10^{-2}	2.00	3.17×10^{-3}	2.07
1/32	1.49×10^{-2}	1.79	6.45×10^{-4}	2.33	1.38×10^{-2}	1.76	6.30×10^{-4}	2.33
1/64	4.10×10^{-3}	1.86	1.41×10^{-4}	2.19	3.68×10^{-3}	1.91	1.41×10^{-4}	2.16
1/128	9.34×10^{-4}	2.14	3.31×10^{-5}	2.09	8.42×10^{-4}	2.13	3.32×10^{-5}	2.09

Table 2: Errors at $T = 5$ and convergence rates of the Stab-SAV-CNLF ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0.1$, $\nu_{min} = 0.0005$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,3}$	Rate	$\lvert p_h - p\rvert_{L^2}^{E,3}$	Rate
1/8	1.55×10^{-1}	—	7.39×10^{-3}	—	1.40×10^{-1}		7.56×10^{-3}	—
1/16	3.77×10^{-2}	2.03	6.87×10^{-4}	3.43	3.31×10^{-2}	2.09	6.96×10^{-4}	3.44
1/32	7.02×10^{-3}	2.43	1.90×10^{-4}	1.85	6.21×10^{-3}	2.41	1.89×10^{-4}	1.88
1/64	1.48×10^{-3}	2.25	5.85×10^{-5}	1.70	1.32×10^{-3}	2.23	5.74×10^{-5}	1.72
1/128	3.56×10^{-4}	2.05	1.53×10^{-5}	1.93	3.25×10^{-4}	2.03	1.49×10^{-5}	1.94

case. α is taken to be positive so that the numerical schemes converge, which is not guaranteed by pure SAV algorthms studied in the literature [41–43]. Taking $T = 5$, $h = \Delta t$, we compute numerical solutions by the Stab-SAV-CNLE and Stab-SAV-CNLF schemes with four successive timestep reductions. Table 1 lists the numerical errors at the final time computed by the Stab-SAV-CNLE scheme for the first and third samples. The stabilization parameter is taken as $\alpha = 0.5$ in particular. As a comparison, Table 2 lists the numerical errors computed by the Stab-SAV-CNLF scheme taking $\alpha = 0.1$.

As shown in the tables, both schemes have second order convergence as predicted. In the numerical experiments, slightly larger α values also work fine with acceptable accuracy. The optimal choice of the stabilization parameter, however, is generally application dependent. In comparing Stab-SAV-CNLF with Stab-SAV-CNLE, we observe that Stab-SAV-CNLF requires smaller value of α than Stab-SAV-CNLE for the case of large Reynolds number. Specifically, the small value $\alpha = 0.1$ works fine for Stab-SAV-CNLF, but does not ensure convergence of Stab-SAV-CNLE, which requires α to be at least around 0.5. Consequently, more additional errors are brought into the pure SAV-CNLE scheme by stabilization than into the pure SAV-CNLF scheme.

We also run a second test case by taking $\nu_{min} = 0.01$, so that the Reynolds number is relatively small and stabilization is possibely not required. If we set $\alpha = 0$, Stab-SAV-CNLE (equivalent to SAV-CNLE) achieves second-order convergence, but Stab-SAV-CNLF (equivalent to SAV-CNLF) does not converge. This is possibly because SAV-CNLF has a stricter timestep condition for convergence than SAV-CNLE. If a more appropriate value of α is taken, namely the timestep restriction for convergence is relaxed by stabilization, Stab-SAV-CNLF has comparable performance to Stab-SAV-CNLE.

Detailed experimental results for the case $\nu_{min} = 0.01$ are reported in Figure 1. Errors of velocity (resp. pressure) are plotted in pictures on the left (resp. right). The pictures on the top correspond to the first sample (i.e. $j = 1$), and the bottom pictures correspond to $j = J$. Figure 1 illustrates second-order convergence for the Stab-SAV-CNLE scheme with $\alpha = 0$ and $\alpha = 0.1$ and the Stab-SAV-CNLF scheme with $\alpha = 0.1$. When $\alpha = 0.1$ both schemes are stable with large time steps. The Stab-SAV-CNLF

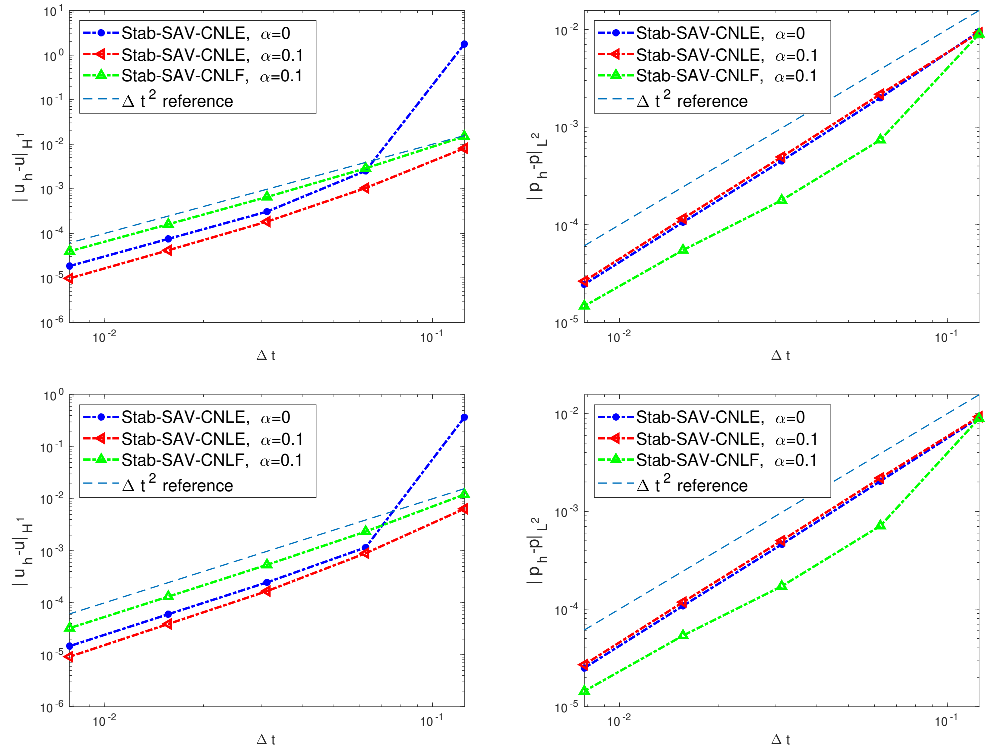


Fig. 1: Errors of velocity (on the left) and pressure (on the right) at $T = 5$ by the Stab-SAV-CNLE and Stab-SAV-CNLF ensemble algorithms ($J = 3$) with $\Delta t = h$, $\nu_{min} = 0.01$. Top: for the first sample; bottom: for the third sample.

method returns a slightly larger error on velocity, but a smaller error on pressure.

5.2. Flow past a cylinder. Our next test problem is the two-dimensional flow past a cylinder, a classical benchmark problem introduced in Schäfer and Turek [49]. This problem has been widely used, in [34, 40] for instance, to study the stability or effectiveness of certain time stepping methods. In our work, the aim is to show that Stab-SAV-CNLE and Stab-SAV-CNLF both produce reasonable simulations if appropriate stabilization parameters and relatively large time steps 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 condition is imposed on the cylinder, also the top and bottom of the channel, while the inflow/outflow boundary conditions are prescribed as

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

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 von Kármán vortex street developing behind the cylinder. In particular, the eddies become

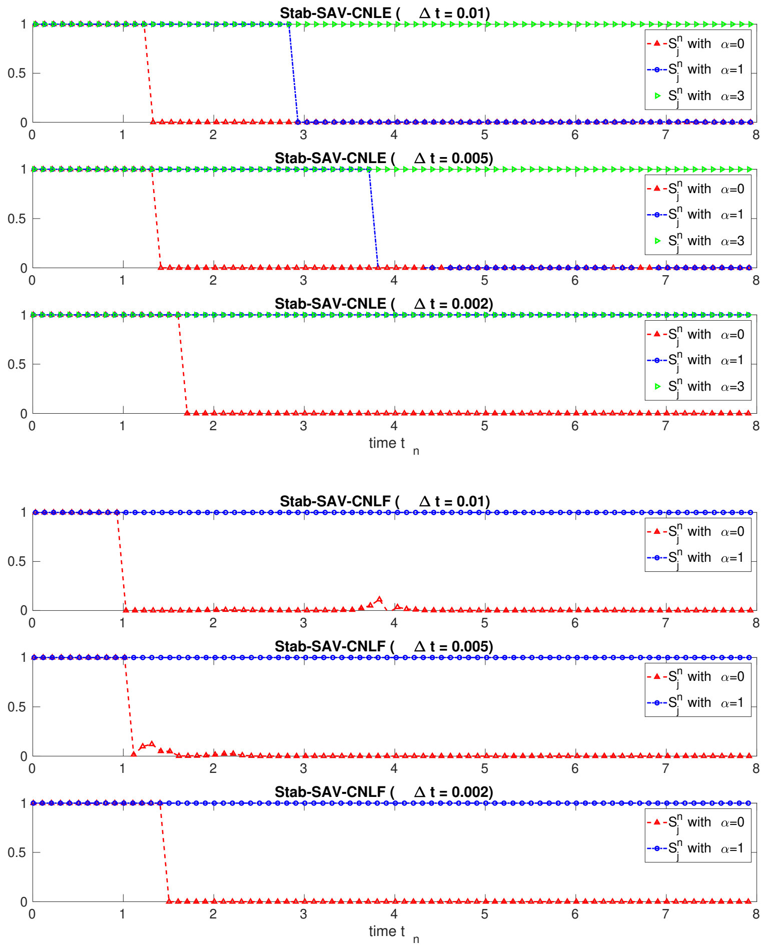


Fig. 2: Flow past cylinder: time histories of S_j^n computed by the Stab-SAV-CNLE and Stab-SAV-CNLF schemes with $\nu = 10^{-3}, h = 0.028, J = 1 = j$.

unstable from t = 2, and then are shed on alternate sides of the cylinder between t = 4 and t = 6.

The numerical solutions are computed with Taylor-Hood elements holding 31140 number of degrees of freedom for velocity and 3980 for pressure. The spatial resolution is 0.028. Simulations are performed with $\Delta t = 0.01, 0.005, 0.002$, and various stabilization parameters α. The corresponding time histories of S_j^n are plotted in Figure 2. In both schemes, the importance of adding stabilization into the SAV approach is obvious, since the convergence of S_j^n to one associates directly with the accuracy of numerical solution. According to the results, both methods work fine with $\Delta t = 0.01$ if an appropriate stabilization parameter α is provided. In addition, by comparing Stab-SAV-CNLF with Stab-SAV-CNLE we observe that the Stab-SAV-CNLF scheme is more flexible in choosing α. Specifically, while a large $\Delta t = 0.01$ with $\alpha = 1$ in Stab-SAV-CNLF is good enough for stable simulation, the Stab-SAV-CNLE method needs $\alpha = 3$.

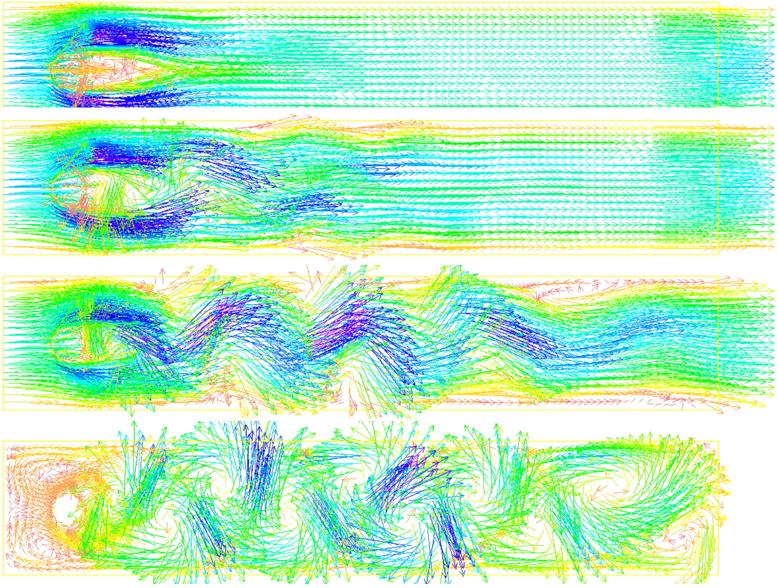


Fig. 3: Flow past cylinder: velocity fields at $t = 2, 5, 7, 8$. Computed by Stab-SAV-CNLE with $\Delta t = 0.01, \alpha = 3$.

The velocity fields of the flow at $t = 2, 5, 7, 8$ computed by the Stab-SAV-CNLE method with $\Delta t = 0.01$ and $\alpha = 3$ are plotted in Figure 3. Figure 4 is for the case of $\Delta t = 0.001$. As for the Stab-SAV-CNLF method, $\alpha = 1$ is used. Results with $\Delta t = 0.01$ and $\Delta t = 0.002$ are plotted in Figure 5 and 6, respectively. Overall, the simulation results of both schemes are satisfactory: the simulations are stable and the flow patterns produced match with those in [34, 37].

5.3. Flow between two offset cylinders. We also consider a two-dimensional flow between two offset cylinders to test the stability and efficiency of the two Stab-SAV algorithms. Consider a disk with a smaller off center obstacle inside. That is, the flow is in the domain

$$\Omega = \{(x, y) : x^2 + y^2 \leq 1 \text{ and } (x - 0.5)^2 + y^2 \geq 0.1^2\}.$$

No-slip boundary condition is enforced on both circles. Driven by a counterclockwise rotational body force

$$f(x, y) = \left(-4y(1 - x^2 - y^2), 4x(1 - x^2 - y^2)\right)^T,$$

the flow interacts with the inner circle generating complex flow structures. Specifically, a von Kármán vortex street is developed and then reinteracts with the inner circle. Extensive experiments on this flow have been performed in [23, 27, 35] for the study of certain first or second order ensemble methods. It is worth mentioning that the flow considered here is much more complicated than previous examples.

To generate perturbations of the initial conditions, we solve the steady Stokes problem with J perturbed body forces given by

$$f_j(x, y) = f(x, y) + \epsilon_j \left(\sin(3\pi x) \sin(3\pi y), \cos(3\pi x) \cos(3\pi y)\right)^T, \; j = 1, \cdots, J,$$

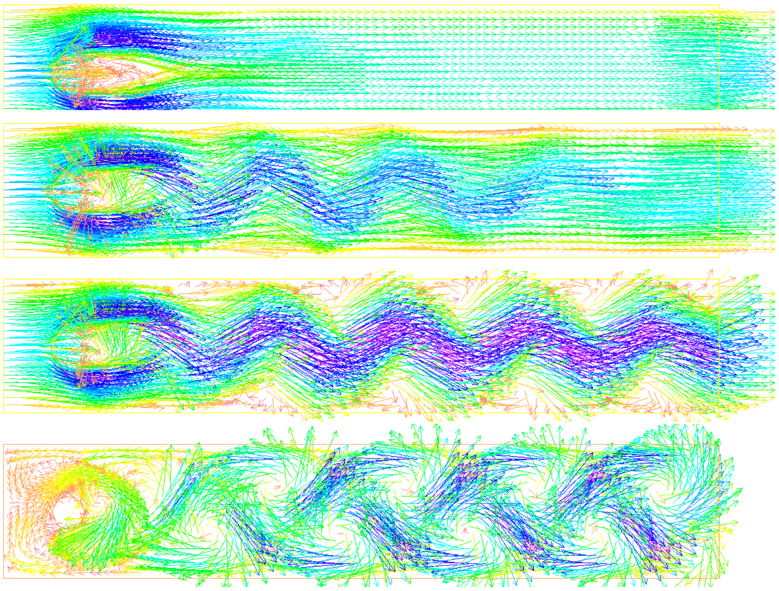


Fig. 4: Flow past cylinder: velocity fields at $t = 2, 5, 7, 8$. Computed by Stab-SAV-CNLE with $\Delta t = 0.002, \alpha = 3$.

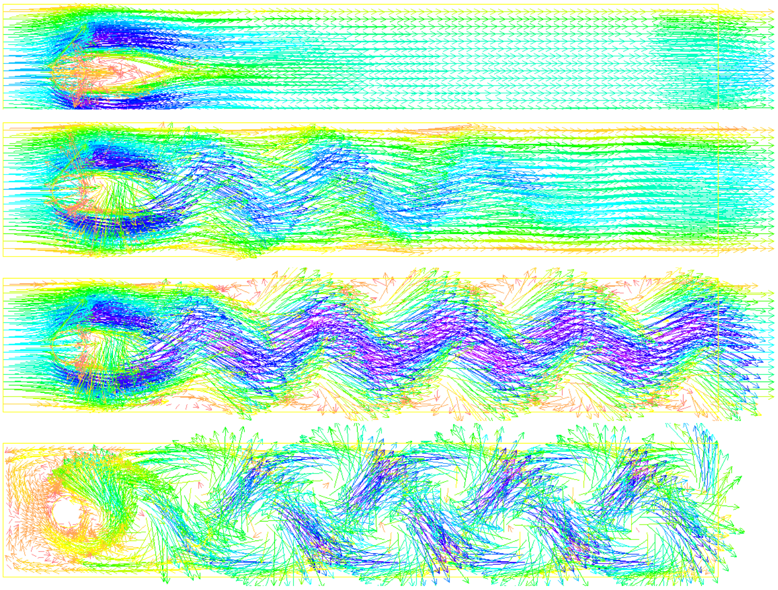


Fig. 5: Flow past cylinder: velocity fields at $t = 2, 5, 7, 8$. Computed by Stab-SAV-CNLF with $\Delta t = 0.01, \alpha = 1$.

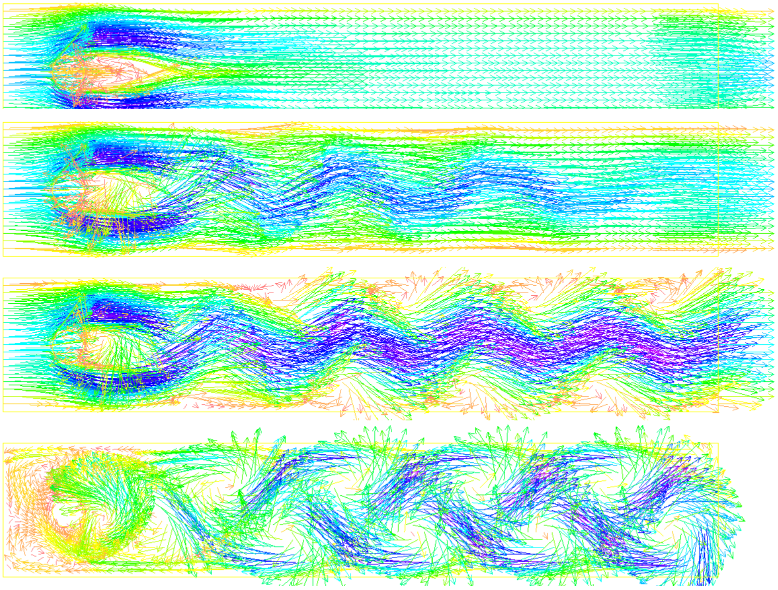


Fig. 6: Flow past cylinder: velocity fields at $t = 2, 5, 7, 8$. Computed by Stab-SAV-CNLF with $\Delta t = 0.002, \alpha = 1$.

where ϵ_j is uniformly distributed in $[10^{-3}, 10^{-2}]$. This results in $J = 10$ discretely divergence free initial conditions. The viscosities ν_j are uniformly distributed between 0.01 and 0.015 while satisfying the parameter fluctuation condition $\nu'_{max} < \frac{1}{3}\bar{\nu}_{min}$.

Numerical solutions are computed with Taylor-Hood elements holding 65292 number of degrees of freedom for velocity and 8244 for pressure. The spatial resolution is 0.038. Simulations are performed with $\Delta t = 0.01, 0.005, 0.001$, and various stabilization parameter α. The corresponding time histories of S_j^n are plotted in Figure 7. The plot corresponds to $\nu_{10} = 0.0104651$ in particular. As before, we observe that the Stab-SAV-CNLF method requires smaller stabilization values than Stab-SAV-CNLE to ensure stable simulations. For instance, when a large time step $\Delta t = 0.01$ is taken, Stab-SAV-CNLF needs $\alpha = 5$ whereas Stab-SAV-CNLE requires α being as large as 15.

The stability and accuracy of the two ensemble schemes are also illustrated in Figure 8. Specifically, Figure 8 plots the velocity fields of the flow at $T = 10$ computed by different schemes with $\Delta t = 0.01$ or $\Delta t = 0.001$. Consider (a) as a reference; (b) shows the flow computed by the Stab-SAV-CNLE ensemble scheme with $\alpha = 15$; (c) shows simulations obtained by the Stab-SAV-CNLE ensemble scheme with $\alpha = 5$.

We also report in Table 3 the CPU time for simulations with different schemes. As we expect, while using the same time step size, the Stab-SAV-CNLE and Stab-SAV-CNLF schemes take much less computational time than the CNLE nonensemble scheme.

5.4. Three-dimensional efficiency tests. In this subsection, ensemble efficiency of the two schemes will be reported by simulating the Arnold-Beltrami-Childress flow that was originally introduced by Arnold [2] and Childress [6] and

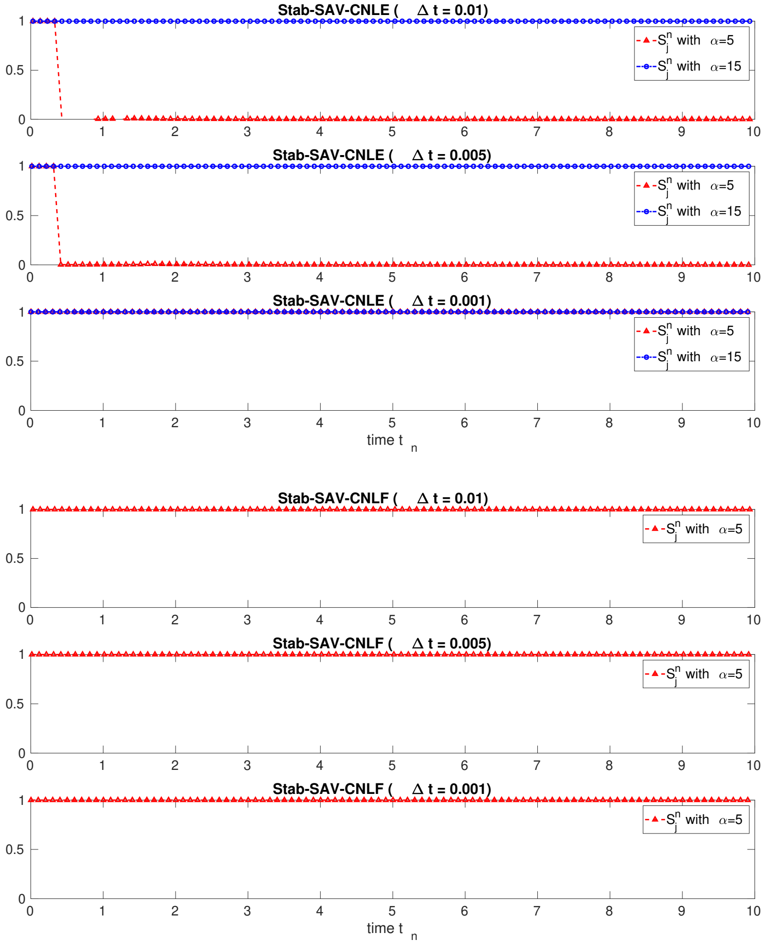


Fig. 7: Flow between two offset cylinders: time histories of S_j^n computed by the Stab-SAV-CNLE and Stab-SAV-CNLF schemes with $J = 10$. The plot corresponds to $\nu_{10} = 0.0104651$.

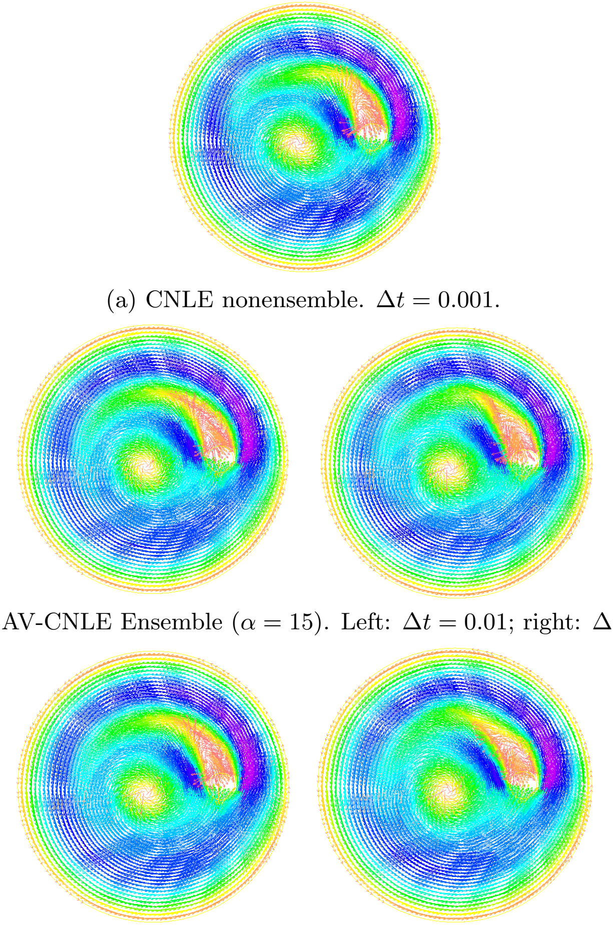
studied in several papers such as [52].

The analytical solution of the NSE is given by

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

In particular, the number J of realizations will vary from 1 to 100. The values of ν_j are set by taking

$$
\nu_j = \nu_{min}(1+\epsilon_j),\ \nu_{min} = 0.001,\ j = 1, \cdots, J,
$$



(a) CNLE nonensemble. $\Delta t = 0.001$.

(b) Stab-SAV-CNLE Ensemble ($\alpha = 15$). Left: $\Delta t = 0.01$; right: $\Delta t = 0.001$.

(c) Stab-SAV-CNLF Ensemble ($\alpha = 5$). Left: $\Delta t = 0.01$; right: $\Delta t = 0.001$.

Fig. 8: Velocity fields of the ensemble flow between two offset cylinders at $T = 10$, computed by different schemes with $J = 10$. Consider (a) as a reference, (b) shows the flow computed by the Stab-SAV-CNLE ensemble scheme with $\alpha = 15$, (c) shows simulations obtained by the Stab-SAV-CNLE ensemble scheme with $\alpha = 5$.

Table 3: CPU time for simulating the flow between two offset cylinder with $J = 10$, $T = 10$.

Scheme	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$
CNLE nonensemble	26588 s	55485 s	263416 s
Stab-SAV-CNLE ensemble	8127 s	16655 s	79466 s
Stab-SAV-CNLF ensemble	8439 s	17596 s	84209 s

Table 4: Three dimensional Arnold-Beltrami-Childress flow: execution time and errors of mean for different values of J at final time $T = 0.5$ with $h = 1/16, \Delta t = 0.02$.

	Stab-SAV-CNLE ensemble			CNLE nonensemble		
J	$\lvert\mathbb{E}[u_h - u]\rvert_{H^1}$	$\lvert\mathbb{E}[p_h - p]\rvert_{L^2}$	Exe time	$\lvert\mathbb{E}[u_h - u]\rvert_{H^1}$	$\lvert\mathbb{E}[p_h - p]\rvert_{L^2}$	Exe time
1	1.00×10^{-2}	7.57×10^{-3}	1035 s	1.14×10^{-2}	2.74×10^{-3}	1781 s
10	9.57×10^{-3}	3.20×10^{-3}	3116 s	1.01×10^{-2}	1.79×10^{-3}	17403 s
100	9.52×10^{-3}	3.48×10^{-3}	21324 s	1.07×10^{-2}	1.78×10^{-3}	147091 s

Table 5: Three-dimensional Arnold-Beltrami-Childress flow: execution time and errors of mean for different values of J at final time $T = 0.5$ with $h = 1/16, \Delta t = 0.02$.

	Stab-SAV-CNLF ensemble			CNLF nonensemble		
J	$\lvert\mathbb{E}[u_h - u]\rvert_{H^1}$	$\lvert\mathbb{E}[p_h - p]\rvert_{L^2}$	Exe time	$\lvert\mathbb{E}[u_h - u]\rvert_{H^1}$	$\lvert\mathbb{E}[p_h - p]\rvert_{L^2}$	Exe time
1	1.03×10^{-2}	2.32×10^{-3}	1653 s	1.14×10^{-2}	3.51×10^{-3}	1982 s
10	9.91×10^{-3}	1.57×10^{-3}	4329 s	1.08×10^{-2}	2.32×10^{-3}	19946 s
100	9.87×10^{-3}	1.57×10^{-3}	24697 s	1.07×10^{-2}	2.27×10^{-3}	164411 s

where ϵ_j is a random variable uniformly distributed in $[0, 0.2]$. The stabilization parameter is $\alpha = 0.5$. The space and time resolutions are fixed as $h = 1/16, \Delta t = 0.02$ and the final time is taken at $T = 0.5$. In this efficiency test, we use the block GMRES solver with the least-square commutator preconditioner to handle algebraic linear systems. Inside the block GMRES solver, this preconditioner is solved by a multigrid V cycle.

In Table 4 we report the numerical errors and execution time computed by the Stab-SAV-CNLE ensemble algorithm and CNLE nonensemble algorithm with different numbers of ensemble members. The performance of Stab-SAV-CNLF ensemble scheme is compared to CNLF nonensemble scheme in Table 5. One can observe that, in each comparison, the ensemble algorithm outperforms the nonensemble method since it takes much less CPU time while retaining similar accuracy. The advantage of the two Stab-SAV ensemble algorithms is more apparent as the ensemble size increases. This validates that both Stab-SAV-CNLE and Stab-SAV-CNLF ensemble algorithms have good scaling performance on ensemble computing. If Stab-SAV-CNLF is compared to Stab-SAV-CNLE directly, we still notice that the Stab-SAV-CNLF scheme returns slightly a larger error on velocity but a smaller error on pressure. The computational efficiency of these two schemes are comparable.

5.5. Three-dimensional lid driven cavity flow. In this last test case we simulate the three-dimensional lid driven cavity flow, which has been widely studied in literatures such as [1,52], to investigate the performance of the Stab-SAV-CNLF and Stab-SAV-CNLE schemes. To be specific, 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 conditions are imposed on the other parts of the boundary. The initial velocity and external body force are set to be zero. Taking Reynolds number $Re = \frac{1}{\nu} = 400$ and $Re = 1000$, we run simulations until $T = 40$ to study the stability of the proposed schemes.

Our numerical experiments are performed with $h = 1/20$ fixed, Δt various between 0.05 and 0.02. The time histories of S_j^n computed with $\alpha = 0$ and $\alpha = 0.5$ in the Stab-SAV-CNLE and Stab-SAV-CNLF schemes are plotted in Figure 9. In both schemes, the importance of adding stabilization into the SAV approach is apparent, since S_j^n does not converge to one when the simulation is performed using relatively

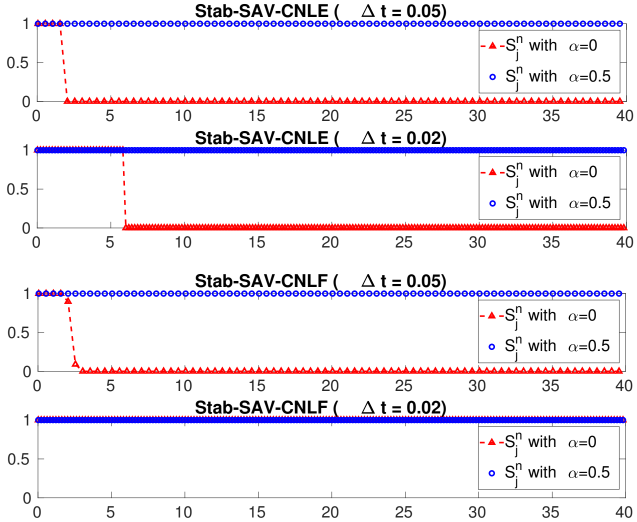


Fig. 9: 3D lid driven cavity flow: time histories of S_j^n computed by the Stab-SAV-CNLE and Stab-SAV-CNLF schemes with $Re = 1000, h = 1/20, J = 1 = j$.

large time steps and without stabilization. In other words, the two Stab-SAV schemes have relaxed time step constraints for convergence. For instance, a large $\Delta t = 0.05$ with $\alpha = 0.5$ is good enough for stable simulations. As before, we observe that the Stab-SAV-CNLF scheme is more flexible in choosing the stabilization parameter α.

In the top of Figure 10 and 11 we plot the streamlines and magnitude of velocity in 3D coordinates, for the lid driven cavity flow with $Re = 400$ and 1000 respectively. These flows are simulated by the Stab-SAV-CNLF scheme, and almost idential results can be provided by the Stab-SAV-CNLE scheme. 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. In this report we proposed a new second order, stabilized SAV ensemble method based on the CNLF timestepping: Stab-SAV-CNLF, for fast computation of nonlinear flow ensembles. We presented details of numerical implementation for Stab-SAV-CNLF as well as a similar method that was proposed in [35]: Stab-SAV-CNLE. Both methods are proved to be long time stable under a parameter condition that limits the size of the fluctuation around the uncertain parameter. We performed extensive numerical experiments to test the accuracy, stability, efficiency of the two methods. Both methods are demonstrated to be extremely efficient and competitively accurate. In addition, through several experiments we observe that the Stab-SAV-CNLF scheme is more flexible in choosing the stabilization parameter α.

Declarations.

Funding. N. Jiang was partially supported by the US National Science Foundation grants DMS-1720001, DMS-2120413, and DMS-2143331. H. 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 no. 2019KZDXM034), and the basic research and applied basic research projects

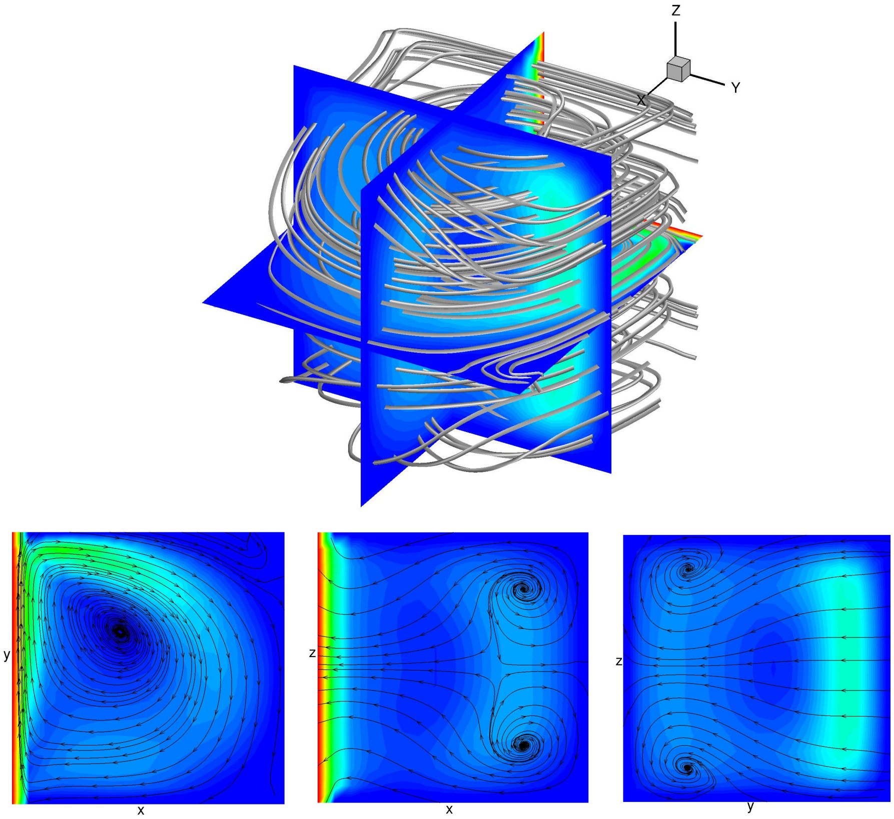


Fig. 10: 3D lid driven cavity flow with $Re = 400$ simulated by the Stab-SAV-CNLF scheme. Top: streamlines and magnitude of velocity in 3D coordinates. Bottom: streamlines generated by velocities projected on the x-y plane, x-z plane, and y-z plane respectively.

in Guangdong Province (Projects of Guangdong, Hong Kong and Macao Center for Applied Mathematics, grant no. 2020B1515310018).

Conflicts of interest. The authors declare that they have no conflict of interest.

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] 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.

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

[5] 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.

[6] S. Childress, *New solutions of the kinematic dynamo problem*, Journal of Mathematical

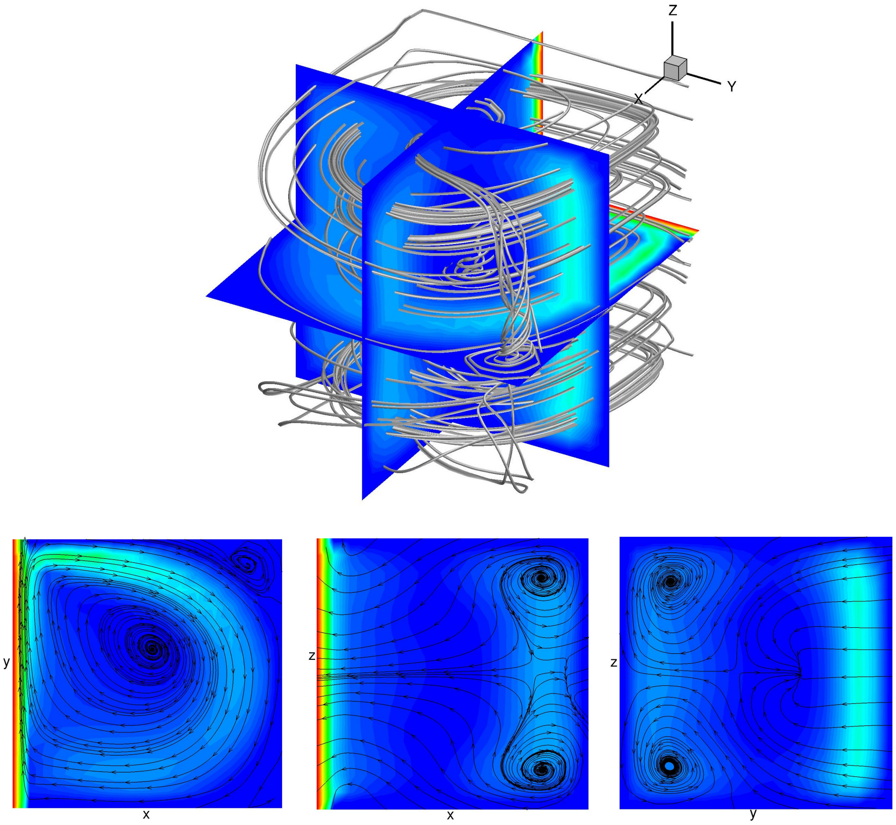


Fig. 11: 3D lid driven cavity flow with $Re = 1000$ simulated by the Stab-SAV-CNLF scheme. Top: streamlines and magnitude of velocity in 3D coordinates. Bottom: streamlines generated by velocities projected on the x-y plane, x-z plane, and y-z plane respectively.

Physics, 11 (1970), 3063-3076.

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

[8] H.C. Elman, D.J. Silvester, A.J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, New York, 2005.

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

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

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] F. Hecht, *New development in FreeFem++*, Journal of Numerical Mathematics, 20 (2012), 251-265.

[21] J.C. Helton and F.J. Davis, *Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems*, Reliability Engineering and System Safety, 81 (2003), 23-69.

[22] 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.

[23] N. Jiang, *A higher order ensemble simulation algorithm for fluid flows*, Journal of Scientific Computing, 64 (2015), 264-288.

[24] N. Jiang, *A second-order ensemble method based on a blended backward differentiation formula timestepping scheme for time-dependent Navier-Stokes equations*, Numerical Methods for Partial Differential Equations, 33 (2017), 34-61.

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

[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, Y. Li and H. Yang, *A second order ensemble method with different subdomain time steps for simulating coupled surface-groundwater flows*, accepted in Numerical Methods for Partial Differential Equations, in press, 2022. https://onlinelibrary.wiley.com/doi/10.1002/num.22846

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

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

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] F. Kuo, C. Schwab and I. Sloan, *Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients*, SIAM J. Numer. Anal., 50

(2012), 3351-3374.

[40] 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.

[41] 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.

[42] X. Li, J. Shen and Z. Liu, *New SAV-pressure correction methods for the Navier-Stokes equations: stability and error analysis*, Mathematics of Computation, 91 (2022), 141-167.

[43] 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.

[44] 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.

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

[46] J.F. McCarthy, *Block-conjugate-gradient method*, Physical Review D, 40 (1989), 2149.

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

[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] 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.

[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.