STABILIZED SAV ENSEMBLE ALGORITHMS FOR PARAMETERIZED FLOW PROBLEMS

NAN JIANG* AND HUANHUAN YANG †

Abstract. Computing a flow system a number of times with different samples of flow parameters is a common practice in many uncertainty quantification (UQ) applications, which can be prohibitively expensive for complex nonlinear flow problems. This report presents two second order, stabilized, scalar auxiliary variable (SAV) ensemble algorithms for fast computation of the Navier-Stokes flow ensembles: Stab-SAV-CN and Stab-SAV-BDF2. The proposed ensemble algorithms are based on the ensemble timestepping idea which makes use of a quantity called ensemble mean to construct a common coefficient matrix for all realizations at the same time step after spatial discretization, in which case efficient block solvers, e.g., block GMRES, can be used to significantly reduce both storage and computational time. The adoption of a recently developed SAV approach that treats the nonlinear term explicitly results in a *constant* shared coefficient matrix among all realizations at different time steps, which further cuts down the computational cost, yielding an extremely efficient ensemble algorithm for simulating nonlinear flow ensembles with provable long time stability without any timestep conditions. The SAV approach for the Navier-Stokes equations for a single realization was proved to be unconditionally stable in [39, 41]. However we found the SAV approach has very low accuracy that compromises its stability in our initial numerical investigations for several commonly tested benchmark flow problems. In this report, we propose to use the stabilization $-\alpha h\Delta(u^{n+1}-u^n)$ in Stab-SAV-CN and $-\alpha h\Delta(3u^{n+1}-4u^n+u^{n-1})$ in Stab-SAV-BDF2 to address this issue. We prove that both of our ensemble algorithms are long time stable under one parameter fluctuation condition, without any timestep constraints. For a single realization, both algorithms are unconditionally stable and have better accuracy than the SAV methods studied in [39, 41] for our test problems. Extensive numerical experiments are performed to show the efficiency of the proposed ensemble algorithms and the effectiveness of the stabilization for increasing accuracy and stability.

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

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

1. Introduction.

Parameterized flow problems arise in the uncertainty quantification (UQ) process for many engineering and geophysical applications, in which the model coefficients, boundary conditions, initial conditions and/or body forces depend on a set of input parameters, and usually repeated sampling of the parameters and computing the corresponding numerical solutions of the underlying partial differential equation (PDE) system are required. For this type of problems, numerical simulations have to be carefully designed and verified to produce useful and reliable statistics about the underlying physical systems. Much of the efforts have been devoted to developing efficient UQ methods with the goal to achieve high accuracy without incurring excessive computational cost, e.g. quasi-Monte Carlo sequences [36], Latin hypercube sampling [19], centroidal Voronoi tessellations [48], stochastic collocation methods [2, 53] and non-intrusive polynomial chaos methods [20, 47]. These methods mostly focus on the approximation of the stochastic spaces associated with the ran-

*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 and DMS-2120413.

†Department of Mathematics, Shantou University, Guangdong, China 515063, huan2yang@stu.edu.cn. This author 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 in Guangdong Province (Projects of Guangdong, Hong Kong and Macao Center for Applied Mathematics, grant no. 2020B1515310018).

dom parameters, independent of the selection of numerical approaches for discretizing the corresponding deterministic PDE systems. An accurate and efficient numerical discretization technique is certainly desirable to be used together with the aforementioned UQ methods, and numerous methods exist in the literature for this purpose. However, none of these numerical techniques are specifically designed for the computation of a large number of realizations of a PDE system with different parameter samples, until a recent development of ensemble algorithms for this setting, [23,26,27]. To be clear, existing numerical methods are mostly studied for solving one single PDE or PDE system, while the recently developed ensemble algorithms attempt to find a number of solutions of PDEs or PDEs systems with different parameters where the computations of these solutions are coupled through a quantity called ensemble mean. The ensemble algorithms have been extensively tested and demonstrated to be able to greatly reduce the computational cost for ensemble simulations in various UQ applications, e.g., [15, 16], and when combined with an efficient UQ method, they are well suited for simulating parameterized flow problems.

The ensemble algorithm was first studied by Jiang and Layton [27] in 2014 for computing the Navier-Stokes equations subject to uncertainties in the initial conditions and forcing terms. The idea is that if all the linear systems from different realizations share a common coefficient matrix at each time step, then one is actually solving a linear system with multiple right hand sides $A[x_1, x_2, \cdots, x_J] = [b_1, b_2, \cdots, b_J]$, instead of J separate linear systems $A_1x_1 = b_1, A_2x_2 = b_2, \cdots, A_Jx_J = b_J$ that a traditional method (individual simulations) would yield. For solving a linear system with multiple right hand sides, there is a large literature on the study of efficient block solvers, such as block CG [22, 44, 46], block GMRES [4, 10], which can be used to markedly reduce the computational cost compared with solving J different systems separately. This ensemble timestepping idea was subsequently extended to simulate different PDE systems with various uncertain model parameters, e.g. Navier-Stokes equations [12–16, 23, 24, 26, 28, 51, 52], Boussinesq equations [7, 9, 25], heat equation [8,42,43], MHD equations [31,45], coupled fluid-fluid model [5], Stokes-Darcy equations [17, 29, 30]. Some recent developments include combining artificial compressibility techniques to decouple the computation of velocity and pressure in ensemble simulations [17,29], using time relaxation regularization to penalize the deviation of the fluctuations from the ensemble mean for simulation of high Reynolds number flows [51], synthesizing the ensemble algorithm with the multilevel Monte Carlo method [43] and the pseudo-spectral stochastic collocation method [38], developing ensemble-based conventional turbulence models [5, 26]. Although shown to be highly efficient, the ensemble algorithms suffer from a restrictive timestep constraint for long time stability for nonlinear flow problems, [25, 27]. The timestep condition comes from bounding the fluctuation component of the nonlinear term which has been lagged to the previous time levels and thus does not contribute to the common coefficient matrix all realizations share. For the ensemble algorithms, this condition can not be avoided with standard semi-implicit timestepping methods unless additional regularizations are added [28, 51]. But a very recent scalar auxiliary variable (SAV) approach introduced in [39, 41] puts forth a possible way to get rid of this timestep condition and lead to unconditionally stable ensemble algorithms.

The SAV approach was introduced in [49, 50] to construct unconditionally energy stable schemes for approximating gradient flows. The outstanding feature is that the nonlinear term can be treated fully explicitly and one only needs to solve decoupled equations with constant coefficients. This approach was later extended to the Navier-Stokes equations in [41], where the authors devised and tested two uncon-

ditionally stable SAV schemes based on backward Euler and second order backward differentiation formula (BDF2) respectively. Following the same idea, [39] presented a SAV scheme based on the Crank-Nicolson timestepping and provided detailed error analysis for the scheme. In this report, we combine our ensemble algorithms with the SAV idea to construct extremely efficient and unconditionally stable ensemble algorithms. Specifically, the efficiency is achieved as follows. Consider computing the Navier-Stokes equations J times with J different input parameters. After spatial discretization, at the time level t_{n+1}, one needs to solve J linear systems $A_j^n x_j = b_j$, $j = 1, 2, \cdots, J$. The contributions of the ensemble timestepping and the SAV idea are respectively

$$\begin{aligned}
&\text{Ensemble timestepping:} \quad A_j^n \quad \longrightarrow \quad A^n \\
&\text{(all realizations share a common coefficient matrix),} \\
&\text{SAV:} \quad A^n \quad \longrightarrow \quad A \\
&\text{(the coefficient matrix is a constant matrix).}
\end{aligned}$$

The resulting SAV ensemble algorithms have improved efficiency and provable unconditional stability.

However, in our initial numerical investigations of the SAV approach [39, 41] on several benchmark flow problems, it shows very poor accuracy which apparently compromises its claimed unconditional stability. Some of these test results are presented in Table 7, Figure 2 for the Kovasznay flow, Figure 6 for the double driven cavity flow, Figure 7 for a flow between two offset cylinders, in comparison to the stabilized SAV ensemble methods we will present in this report.

Artificial viscosity is a common numerical technique used to stabilize high Reynolds number flows but may lead to overdiffusion. One possible fix is to add anti-diffusion at large scales [1] or previous time levels [37] to correct the solution and improve the accuracy. Similar ideas have been widely used for practical simulations within the computational fluid dynamics (CFD) community, e.g., for turbulence modeling, the variational multiscale (VMS) method has eddy viscosity acting only on small resolved scales [32,33]. In this report, we adopt a simple approach presented in [37] to stabilize our SAV based ensemble algorithms. In [37] the anti-diffusion is added at the previous time level for a Crank-Nicolson timestepping method to help reduce the effects of artificial viscosity but still retain the good property of better conditioning the linear system to be solved at each time step which gives better likelihood of success with iterative solvers and preconditioners. The incorporation of this stabilization approach with our SAV based ensemble algorithms is shown to be able to substantially improve the accuracy of the SAV based schemes (both ensemble and non-ensemble) for the several numerical tests in Section 6.

The rest of the paper is organized as follows. In Section 2, we present the proposed stabilized SAV ensemble algorithms: Stab-SAV-CN and Stab-SAV-BDF2. They are both proved to be long time stable under a fluctuation condition (without any timestep conditions) in Section 3. In particular, for a single simulation, both algorithms are unconditionally stable. Section 4 discusses their implementation algorithms. Section 5 is devoted to detailed explanations of the numerical implementation of the ensemble algorithms and the block GMRES solver for solving the corresponding linear systems. Numerical examples and computational results are presented and discussed in Section 6. We conclude with a brief discussion of the proposed ensemble algorithms in Section 7.

2. The SAV Based Ensemble Algorithms. We consider the setting that the Navier-Stokes equations are subject to uncertainties in the initial conditions, boundary conditions, body forces and the kinematic viscosity, and the solution depends on a set of input parameters that come from these uncertainties. To obtain solution data 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 with J slightly different initial conditions, Dirichlet boundary conditions, body forces and the kinematic viscosity, $u_j^0(x)$, $g_j(x,t)$, $f_j(x,t)$, $\nu_j(x)$, for $j = 1, ..., J$:

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

Here we assume $\nu_j(x) \in L^\infty(\Omega)$ and $\nu_j(x) \geq \nu_{j,min} > 0$. To construct the ensemble algorithms we need to define the ensemble mean $\bar{\nu}$ and the ensemble fluctuations ν_j'. The minimum average $\bar{\nu}_{min}$ and maximum fluctuation ν'_{max} of the kinematic viscosity, will be used in the proof of the long time stability. They are defined as

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

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

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 [39, 41]. Define the scalar auxiliary variable $q_j(t)$ by

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

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

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

Note that the last two terms in the above equation is 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):

$$u_{j,t} + \frac{q_j(t)}{\sqrt{E(u_j)+\delta}}(u_j\cdot\nabla)u_j - \nabla\cdot(\nu_j\nabla u_j) + \nabla p_j = f_j(x,t), \;\; \nabla\cdot u_j = 0, \tag{2.4}$$

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

The benefit from introducing a new variable q and the associated equation into the system is that we will have the same nonlinear term on different sides of the energy equations. With an appropriate time discretization method, by making the nonlinear terms fully explicitly in the same way, we can cancel out the nonlinear terms by adding the two equations and consequently have a provable unconditionally stable scheme for the Navier-Stokes equations. For the ensemble algorithms, one needs to deal with fluctuation induced instabilities and there will be a fluctuation condition to ensure stability. But usually in the context of UQ, the fluctuations of the parameter are small and thus the fluctuation condition can be easily satisfied. In other cases, one can always split a large ensemble into smaller ensembles to reduce the fluctuation size.

Herein we present two second order, stabilized SAV ensemble algorithms based on the Crank-Nicolson timestepping and the BDF2 timestepping respectively: Stab-SAV-CN and Stab-SAV-BDF2. To increase the accuracy of the SAV schemes, we add the following stabilization terms to the two proposed SAV ensemble algorithms.

$$-\alpha h\Delta(u_j^{n+1}-u_j^n) \text{ for Stab-SAV-CN,}$$
$$-\alpha h\Delta(3u_j^{n+1}-4u_j^n+u_j^{n-1}) \text{ for Stab-SAV-BDF2,}$$

where $\alpha>0$ is a tuning parameter that is application dependent, and h is the mesh size associated with the spatial discretization of choice.

The idea originates from the artifical viscosity stabilization which is to add an artificial viscosity term $-\alpha h\Delta u$ to the left hand side of the Navier-Stokes equations so that the cell Reynolds number becomes $O(1)$. For unsteady flows we consider adding anti-diffusion at previous time levels to help reduce the effects of overdiffusion, following the approach presented in [37]. The stabilized SAV ensemble algorithms have accuracy $O(\Delta t^2 + h\Delta t + \text{spatial error})$.

To construct an efficient ensemble algorithm we need to make all the realizations share a common coefficient matrix. The way to do this is to decompose the viscosity term into two parts: the mean and the fluctuation, i.e. $\nu_j = \bar{\nu} + \nu_j'$. The mean viscosity does not depend on the ensemble index j but the fluctuation term does. So we need to lag the fluctuation term to the previous time levels so that it goes to the right hand side of the equation and does not affect the common coefficient matrix. 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.

For the Stab-SAV-CN ensemble algorithm, we adopt a special linear extrapolation $\tilde{u}_j^{n+1/2}$ (defined below), see [21], to approximate the fluctuation term. This is a second order approximation and possibly the only choice that works with the Crank-Nicolson timestepping for a provable long time stable ensemble algorithm.

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

$$u_j^{n+1/2} = \frac{u_j^{n+1}+u_j^n}{2}, \qquad \tilde{u}_j^{n+1/2} = 2\frac{u_j^n+u_j^{n-1}}{2} - \frac{u_j^{n-1}+u_j^{n-2}}{2} = 2u_j^{n-1/2} - u_j^{n-3/2},$$

$$p_j^{n+1/2} = \frac{p_j^{n+1} + p_j^n}{2}, \qquad q_j^{n+1/2} = \frac{q_j^{n+1} + q_j^n}{2}.$$

Note that $\tilde{u}_j^{n+1/2}$ is a known quantity while $u_j^{n+1/2}$ contains the unknown function u_j^{n+1}. We now propose our second order, stabilized SAV ensemble algorithm based on the Crank-Nicolson timestepping as follows.

Algorithm 2.1 (Stab-SAV-CN). 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

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

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

The second order, stabilized SAV ensemble algorithm based on the BDF2 timestepping is given by

Algorithm 2.2 (Stab-SAV-BDF2). For $j = 1, 2, \cdots, J$, given $u_j^0, u_j^1, q_j^0, q_j^1$, for $n = 1, 2, \cdots, N-1$, find $u_j^{n+1}, p_j^{n+1}, q_j^{n+1}$ satisfying

$$\begin{aligned} (2.8)\quad & \frac{3u_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t} + \frac{q_j^{n+1}}{\sqrt{E(\tilde{\tilde{u}}_j^{n+1}) + \delta}}(\tilde{\tilde{u}}_j^{n+1} \cdot \nabla)\tilde{\tilde{u}}_j^{n+1} + \nabla p_j^{n+1} \\ & - \nabla \cdot (\bar{\nu} \nabla u_j^{n+1}) - \nabla \cdot (\nu_j' \nabla \tilde{\tilde{u}}_j^{n+1}) - \alpha h \Delta (3u_j^{n+1} - 4u_j^n + u_j^{n-1}) = f_j^{n+1}, \\ (2.9)\quad & \nabla \cdot u_j^{n+1} = 0, \\ (2.10)\quad & \frac{3q_j^{n+1} - 4q_j^n + q_j^{n-1}}{2\Delta t} = \frac{1}{2q_j^{n+1}} \left(\frac{3u_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t}, u_j^{n+1} \right) \\ & + \frac{1}{2\sqrt{E(\tilde{\tilde{u}}_j^{n+1}) + \delta}} \int_\Omega (\tilde{\tilde{u}}_j^{n+1} \cdot \nabla)\tilde{\tilde{u}}_j^{n+1} \cdot u_j^{n+1}\, dx - \frac{b_j^{n+1}}{2q_j^{n+1}}, \end{aligned}$$

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

It is clear that for both ensemble algorithms, the coefficients for u_j^{n+1} (1) do not vary from one step to another, and (2) are independent of the ensemble index j, which means after spatial discretization, the constant coefficient matrix is the same for all realizations, i.e. A does not depend on n or j. The linear system to be solved at each timestep is in the form of

$$A[x_1^{n+1}, x_2^{n+1}, \cdots, x_J^{n+1}] = [b_1^{n+1}, b_2^{n+1}, \cdots, b_J^{n+1}].$$

Solving this linear system with the block GMRES solver will result in significant savings in computational cost, compared with traditional methods that run the simulations individually, i.e.

$$A_1^n x_1^{n+1} = b_1^{n+1}, A_2^n x_2^{n+1} = b_2^{n+1}, \cdots, A_J^n x_J^{n+1} = b_J^{n+1}, \qquad \text{(non-SAV scheme)}$$
$$\text{or } A_1 x_1^{n+1} = b_1^{n+1}, A_2 x_2^{n+1} = b_2^{n+1}, \cdots, A_J x_J^{n+1} = b_J^{n+1}. \qquad \text{(SAV scheme)}$$

3. Stability of the SAV Ensemble Algorithms. In this section we prove both ensemble algorithms are long time stable under a parameter fluctuation condition, without any timestep conditions. In particular, for a single realization ($J = 1$), the fluctuation is zero and the parameter fluctuation condition is automatically satisfied. So the proposed stabilized algorithms are unconditionally stable for non-ensemble simulations. For simplicity of presentation, we will assume homogeneous Dirichlet boundary condition for the stability proofs. But the stability can also be obtained under the same parameter fluctuation condition for the case of inhomogeneous Dirichlet boundary conditions. Due to page limits, we will only present the proof of the stability for the Stab-SAV-BDF2 ensemble algorithm. In the following theorems we will assume q_j^n is real so that $|q_j^n|^2$ is nonnegative. In practical simulations, if q_j^n ever becomes complex at any time step, we claim the proposed stabilized SAV ensemble algorithms fail and the simulations should be stopped, as the appearance of complex q_j^n will lead to the linear solvers failing eventually which is observed in our numerical experiments for high Reynolds number flows. Nevertheless, it is also observed that the stabilization we propose here is able to effectively prevent q_j^n from becoming complex in many cases for low to moderate Reynolds number flows, improving both accuracy and stability of existing SAV algorithms in the literature.

THEOREM 3.1 (Long Time Stability of Stab-SAV-CN). *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

$$\begin{aligned} |q_j^N|^2 + \frac{3}{2}\nu'_{max}\Delta t\|\nabla u_j^{N-1/2}\|^2 + \frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^{N-3/2}\|^2 + \frac{\alpha}{2}h\Delta t\|\nabla u_j^N\|^2 \\ \le |q_j^2|^2 + \frac{3}{2}\nu'_{max}\Delta t\|\nabla u_j^{3/2}\|^2 + \frac{1}{2}\nu'_{max}\Delta t\|\nabla u_j^{1/2}\|^2 + \frac{\alpha}{2}h\Delta t\|\nabla u_j^2\|^2 \\ + \frac{\Delta t}{2(\bar{\nu}_{min} - 3\nu'_{max})}\sum_{n=2}^{N-1}\|f_j^{n+1/2}\|_{-1}^2. \end{aligned} \tag{3.2}$$

THEOREM 3.2 (Long Time Stability of Stab-SAV-BDF2). *Assume q_j^n is real, for any $n = 0, \cdots, N$, $j = 1, \cdots, J$, and the following parameter fluctuation condition holds*

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

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

$$
\begin{aligned}
&|q_j^N|^2+|2q_j^N-q_j^{N-1}|^2+\sum_{n=1}^{N-1}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2+\alpha h\Delta t\left(\|\nabla u_j^N\|^2+\|2\nabla u_j^N-\nabla u_j^{N-1}\|^2\right)\\
&+3\nu'_{max}\Delta t\|\nabla u_j^N\|^2+\alpha h\Delta t\sum_{n=1}^{N-1}\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2+\nu'_{max}\Delta t\|\nabla u_j^{N-1}\|^2\\
&\quad\le |q_j^1|^2+|2q_j^1-q_j^0|^2+\alpha h\Delta t\left(\|\nabla u_j^1\|^2+\|2\nabla u_j^1-\nabla u_j^0\|^2\right)\\
&\qquad+3\nu'_{max}\Delta t\|\nabla u_j^1\|^2 2+\nu'_{max}\Delta t\|\nabla u_j^0\|^2+\frac{\Delta t}{\bar\nu_{min}-3\nu'_{max}}\sum_{n=1}^{N-1}\|f_j^{n+1}\|_{-1}^2.
\end{aligned}
$$

Proof. Taking the L^2 inner product of (2.8) with u_j^{n+1} and using (2.9) gives

$$
\begin{aligned}
&\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},u_j^{n+1}\right)+\frac{q_j^{n+1}}{\sqrt{E(\tilde{\tilde u}_j^{n+1})+\delta}}b(\tilde{\tilde u}_j^{n+1},\tilde{\tilde u}_j^{n+1},u_j^{n+1})\\
&+\int_{\partial\Omega}(\vec n\cdot u_j^{n+1})p_j^{n+1}d\sigma-\int_{\partial\Omega}(\vec n\cdot\bar\nu\nabla u_j^{n+1})\cdot u_j^{n+1}d\sigma+\|\bar\nu^{\frac12}\nabla u_j^{n+1}\|^2\\
&-\int_{\partial\Omega}(\vec n\cdot\nu'_j\nabla\tilde{\tilde u}_j^{n+1})\cdot u_j^{n+1}d\sigma+(\nu'_j\nabla\tilde{\tilde u}_j^{n+1},\nabla u_j^{n+1})\\
&-\alpha h\int_{\partial\Omega}(\vec n\cdot(3\nabla u_j^{n+1}-4\nabla u_j^n+\nabla u_j^{n-1}))\cdot u_j^{n+1}d\sigma\\
&+\frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2+\|2\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)+\frac{\alpha}{2}h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2\\
&-\frac{\alpha}{2}h\left(\|\nabla u_j^n\|^2+\|2\nabla u_j^n-\nabla u_j^{n-1}\|^2\right)=(f_j^{n+1},u_j^{n+1}).
\end{aligned}
\tag{3.3}
$$

Multiplying (2.10) with $2q_j^{n+1}$ gives

$$
\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2+|2q_j^{n+1}-q_j^n|^2\right)-\frac{1}{2\Delta t}\left(|q_j^n|^2+|2q_j^n-q_j^{n-1}|^2\right)\\
&+\frac{1}{2\Delta t}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2=\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},u_j^{n+1}\right)\\
&\qquad+\frac{q_j^{n+1}}{\sqrt{E(\tilde{\tilde u}_j^{n+1})+\delta}}b(\tilde{\tilde u}_j^{n+1},\tilde{\tilde u}_j^{n+1},u_j^{n+1})-b_j^{n+1}.
\end{aligned}
\tag{3.4}
$$

Adding (3.3) and (3.4) gives

$$
\begin{aligned}
&\frac{1}{2\Delta t}\left(|q_j^{n+1}|^2+|2q_j^{n+1}-q_j^n|^2\right)-\frac{1}{2\Delta t}\left(|q_j^n|^2+|2q_j^n-q_j^{n-1}|^2\right)+\frac{1}{2\Delta t}|q_j^{n+1}-2q_j^n+q_j^{n-1}|^2\\
&+\|\bar\nu^{\frac12}\nabla u_j^{n+1}\|^2+\frac{\alpha}{2}h\left(\|\nabla u_j^{n+1}\|^2+\|2\nabla u_j^{n+1}-\nabla u_j^n\|^2\right)-\frac{\alpha}{2}h\left(\|\nabla u_j^n\|^2+\|2\nabla u_j^n-\nabla u_j^{n-1}\|^2\right)\\
&+\frac{\alpha}{2}h\|\nabla u_j^{n+1}-2\nabla u_j^n+\nabla u_j^{n-1}\|^2=(f_j^{n+1},u_j^{n+1})-(\nu'_j\nabla\tilde{\tilde u}_j^{n+1},\nabla u_j^{n+1})\\
&\quad+\alpha h\int_{\partial\Omega}(\vec n\cdot(3\nabla u_j^{n+1}-4\nabla u_j^n+\nabla u_j^{n-1}))\cdot u_j^{n+1}d\sigma\\
&\quad-\int_{\partial\Omega}(\vec n\cdot u_j^{n+1})p_j^{n+1}d\sigma+\int_{\partial\Omega}(\vec n\cdot\bar\nu\nabla u_j^{n+1})\cdot u_j^{n+1}d\sigma+\int_{\partial\Omega}(\vec n\cdot\nu'_j\nabla\tilde{\tilde u}_j^{n+1})\cdot u_j^{n+1}d\sigma-b_j^{n+1}.
\end{aligned}
$$

In particular, with homogeneous Dirichlet condition the terms of integrals on the boundary 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 and using $(2a-b)^2 \leq 6a^2+3b^2$ gives, for any $\beta > 0, \epsilon > 0$,

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

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

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

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

$$(1-\beta)\bar{\nu}_{min}-3\nu'_{max} = (\frac{1}{2}+\frac{3}{2}\frac{\nu'_{max}}{\bar{\nu}_{min}})\bar{\nu}_{min}-3\nu'_{max} = \frac{1}{2}\left(\bar{\nu}_{min}-3\nu'_{max}\right) > 0, \tag{3.5}$$

and

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

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

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

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

4. Implementation Algorithms. The Stab-SAV-CN given by (2.5)-(2.7) and Stab-SAV-BDF2 given by (2.8)-(2.10) are coupled systems of u, p, q, which requires appropriate decoupling strategies to achieve its claimed efficiency. In this section we describe the corresponding implementation algorithm for the Stab-SAV-BDF2 ensemble algorithm, following the decoupling strategy in [39, 41]. The corresponding implementation algorithm for the Stab-SAV-CN ensemble algorithm is similar, which is omitted here due to page limits.

We will introduce a new scalar S_j^{n+1} to decompose the numerical solution (u_j^{n+1}, p_j^{n+1}) into two parts yielding two sub-problems for the two components $(\hat u_j^{n+1}, \hat p_j^{n+1})$, $(\breve u_j^{n+1}, \breve p_j^{n+1})$ respectively, which do not contain S_j^{n+1}. A separate algebraic equation for S_j^{n+1} will be derived. Let

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

Then instead of solving (2.8)-(2.10), we solve the following two subproblems for $(\hat u_j^{n+1}, \hat p_j^{n+1}), (\breve u_j^{n+1}, \breve p_j^{n+1})$ respectively.

(Stab-SAV-BDF2 sub-problem 1)

$$\begin{cases}\dfrac{3}{2\Delta t}\hat u_j^{n+1}-\nabla\cdot(\bar\nu\nabla\hat u_j^{n+1})-3\alpha h\Delta\hat u_j^{n+1}+\nabla\hat p_j^{n+1}=f_j^{n+1}+\dfrac{2}{\Delta t}u_j^n\\ \qquad\qquad -\dfrac{1}{2\Delta t}u_j^{n-1}+\nabla\cdot(\nu'_j\nabla\tilde{\tilde u}_j^{n+1})-4\alpha h\Delta u_j^n+\alpha h\Delta u_j^{n-1},\ \text{in }\Omega\\ \nabla\cdot\hat u_j^{n+1}=0,\ \text{in }\Omega\\ \hat u_j^{n+1}=g_j^{n+1},\ \text{on }\partial\Omega.\end{cases}$$

(Stab-SAV-BDF2 sub-problem 2)

$$\begin{cases}\dfrac{3}{2\Delta t}\breve u_j^{n+1}-\nabla\cdot(\bar\nu\nabla\breve u_j^{n+1})-3\alpha h\Delta\breve u_j^{n+1}+\nabla\breve p_j^{n+1}=-(\tilde{\tilde u}_j^{n+1}\cdot\nabla)\tilde{\tilde u}_j^{n+1},\ \text{in }\Omega\\ \nabla\cdot\breve u_j^{n+1}=0,\ \text{in }\Omega\\ \breve u_j^{n+1}=0,\ \text{on }\partial\Omega.\end{cases}$$

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

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

Multiplying (2.10) with $2q_j^{n+1}$ and then plugging in (4.1) gives the equation for S_j^{n+1} as

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

where

$$A_j^{n+1} = \frac{3}{\Delta t}(E(\tilde{\tilde{u}}_j^{n+1}) + \delta) - \left(\frac{3\breve{u}_j^{n+1}}{2\Delta t}, \breve{u}_j^{n+1}\right) - \int_\Omega (\tilde{\tilde{u}}_j^{n+1} \cdot \nabla)\tilde{\tilde{u}}_j^{n+1} \cdot \breve{u}_j^{n+1}\, dx,$$

$$B_j^{n+1} = \frac{-4q_j^n + q_j^{n-1}}{\Delta t}\sqrt{E(\tilde{\tilde{u}}_j^{n+1}) + \delta} - \left(\frac{3\breve{u}_j^{n+1}}{2\Delta t}, \hat{u}_j^{n+1}\right) - \left(\frac{3\hat{u}_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t}, \breve{u}_j^{n+1}\right)$$
$$- \int_\Omega (\tilde{\tilde{u}}_j^{n+1} \cdot \nabla)\tilde{\tilde{u}}_j^{n+1} \cdot \hat{u}_j^{n+1}\, dx,$$

$$C_j^{n+1} = -\left(\frac{3\hat{u}_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t}, \hat{u}_j^{n+1}\right) + b_j^{n+1}.$$

In general, this is a scalar quadratic equations with two roots. By the definition of S_j^{n+1}, we should pick the root that is close to 1. If we ever get complex roots, the proposed stabilized SAV ensemble algorithms are considered to have failed and the simulation should be stopped as the linear solver will fail eventually in this case. We want to emphasize here that it is also observed in our numerical experiments that the stabilization we propose here is able to effectively prevent q_j^n from becoming complex in many cases for low to moderate Reynolds number flows, improving both accuracy and stability of existing SAV algorithms in the literature.

In solving sub-problem 1 and sub-problem 2, all realizations have the same constant coefficient matrix for all time steps and therefore can be solved very efficiently. In sub-problem 3, we need to solve each realization separately. But since it is a scalar quadratic equation, it can be solved quickly. After getting $\hat{u}_j^{n+1}, \breve{u}_j^{n+1}$ and S_j^{n+1}, we have $u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}$, and similarly for p_j^{n+1}.

5. Algebraic Systems and Block GMRES with Deflation. Let $S_h^2(\Omega)^2$ and $S_h^1(\Omega)$ denote the space of Taylor-Hood elements (P2-P1) on Ω. The basis functions of $S_h^2(\Omega)^2$ and $S_h^1(\Omega)$ are denoted by $\{\chi_j^u\}_{j=1}^{N_u}$, $\{\chi_j^p\}_{j=1}^{N_p}$ respectively. The approximations of the solutions will be represented by vectors of nodal values, denoted in bold. When a superscript n is applied to a bold vector, it represents the value at time $t_n = n\Delta t$, and a subscript j will be applied to represent the solution for the j-th sample. Let $\mathbf{M}_{uu}$ and $\mathbf{S}_{uu}$ denote the velocity mass matrix and velocity stiffness matrix respectively. We also define matrices $\mathbf{D}_{uup}$, $\mathbf{S}(\nu)$ and $\mathbf{N}(u)$ whose entries are given as follows.

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

Our proposed stabilized SAV ensemble schemes will be compared with some non-ensemble schemes:

Algorithm 5.1 (Stab-SAV-BDF2-nonensemble).

$$\begin{cases} \dfrac{3u_j^{n+1} - 4u_j^n + u_j^{n-1}}{2\Delta t} + \dfrac{q_j^{n+1}}{\sqrt{E(\tilde{\tilde{u}}_j^{n+1}) + \delta}}(\tilde{\tilde{u}}_j^{n+1} \cdot \nabla)\tilde{\tilde{u}}_j^{n+1} + \nabla p_j^{n+1} \\ \qquad - \nabla \cdot (\nu_j \nabla u_j^{n+1}) - \alpha h\Delta(3u_j^{n+1} - 4u_j^n + u_j^{n-1}) = f_j^{n+1}, \\ + \text{equations (2.9)} - \text{(2.10)}. \end{cases}$$

Algorithm 5.2 (BDF2-nonensemble).

$$\begin{cases} \dfrac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t}+(\tilde{\tilde{u}}_j^{n+1}\cdot\nabla)u_j^{n+1}+\nabla p_j^{n+1}-\nabla\cdot(\nu_j\nabla u_j^{n+1})=f_j^{n+1}, \\ \nabla\cdot u_j^{n+1}=0. \end{cases}$$

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

1. Stab-SAV-BDF2 Ensemble:

$$\mathbf{A}_{\text{savbdf2ens}}\begin{pmatrix}\hat{\mathbf{u}}_j^{n+1}\\ \hat{\mathbf{p}}_j^{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{b}_j^{n+1}\\ \mathbf{0}\end{pmatrix},\quad \mathbf{A}_{\text{savbdf2ens}}\begin{pmatrix}\breve{\mathbf{u}}_j^{n+1}\\ \breve{\mathbf{p}}_j^{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{c}_j^{n+1}\\ \mathbf{0}\end{pmatrix},$$

with

$$\mathbf{b}_j^{n+1}=\mathbf{f}_j^{n+1}+\mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n-\tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1})-\mathbf{S}(\nu_j')\tilde{\tilde{\mathbf{u}}}_j^{n+1}+\mathbf{S}_{uu}(4\alpha h\mathbf{u}_j^n-\alpha h\mathbf{u}_j^{n-1}),$$

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

$$\mathbf{A}_{\text{savbdf2ens}}=\begin{pmatrix}\frac{3}{2\Delta t}\mathbf{M}_{uu}+\mathbf{S}(\bar{\nu}+3\alpha h) & -\mathbf{D}_{uup}\\ -\mathbf{D}_{uup}^T & \mathbf{0}\end{pmatrix}.$$

2. Stab-SAV-BDF2 Non-ensemble:

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

with

$$\mathbf{b}_j^{n+1}=\mathbf{f}_j^{n+1}+\mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n-\tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1})+\mathbf{S}_{uu}(4\alpha h\mathbf{u}_j^n-\alpha h\mathbf{u}_j^{n-1}),$$

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

$$\mathbf{A}_{\text{savbdf2}}^{(j)}=\begin{pmatrix}\frac{3}{2\Delta t}\mathbf{M}_{uu}+\mathbf{S}(\nu_j+3\alpha h) & -\mathbf{D}_{uup}\\ -\mathbf{D}_{uup}^T & \mathbf{0}\end{pmatrix}.$$

3. BDF2 Non-ensemble:

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

with

$$\mathbf{b}_j^{n+1}=\mathbf{f}_j^{n+1}+\mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n-\tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1}),$$

$$\mathbf{A}_{\text{bdf2}}^{(j)}=\begin{pmatrix}\frac{3}{2\Delta t}\mathbf{M}_{uu}+\mathbf{N}(\tilde{\tilde{u}}_j^{n+1})+\mathbf{S}(\nu_j) & -\mathbf{D}_{uup}\\ -\mathbf{D}_{uup}^T & \mathbf{0}\end{pmatrix}.$$

One should notice that the matrix $\mathbf{A}_{\text{savbdf2ens}}$ in our ensemble method is fixed for different samples, so we can simultaneously compute all realizations by solving a

single linear system with multiple right hand sides (RHSs) corresponding to different samples; but $\mathbf{A}^{(j)}_{\text{savbdf2}}$ and $\mathbf{A}^{(j)}_{\text{bdf2}}$ in the non-ensemble methods change over j, and we need to simulate J samples one by one. In application, one can use the block GMRES algorithm with deflation [4, BFGMRESD(m)] to solve those algebraic linear systems, which is more practicable for large-scale simulations, especially for 3D problems. The algorithm is stated in Algorithm 5.1. In general, a block iterative solver is especially designed for huge sparse linear systems with multiple right-hand sides given at once. With deflation, redundant information due to linear dependence of multiple residuals are removed, so the block GMRES with deflation solver has good performance on ensemble scaling. In contrast, the non-ensemble schemes solve the linear systems for different samples one by one, so no simultaneous solving can be applied. The corresponding solutions are obtained by a standard GMRES solver.

Algorithm 5.1 Block-GMRES-Deflation [4]

Input: matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, right hand side $\mathbf{B} \in \mathbb{R}^{n\times p}$, initial guess $\mathbf{X}_0 \in \mathbb{R}^{n\times p}$, preconditioner $\mathbf{K} \in \mathbb{R}^{n\times n}$, convergence threshold tol, deflation threshold ϵ_d, restart number m, maximum iteration number it_{max}

Output: approximated solution of system $\mathbf{AX} = \mathbf{B}$

Define diagonal matrix $\mathbf{D} = \text{diag}(\|\mathbf{B}(:,1)\|_2, \cdots, \|\mathbf{B}(:,p)\|_2) \in \mathbb{R}^{p\times p}$
for $it = 1, \cdots, it_{max}$ **do**
  $\mathbf{R}_0 = \mathbf{B} - \mathbf{AX}_0$
  Do thin QR factorization of $\mathbf{R}_0\mathbf{D}^{-1}$: $\mathbf{R}_0\mathbf{D}^{-1} = \mathbf{QT}$ with $\mathbf{Q} \in \mathbb{R}^{n\times p}, \mathbf{T} \in \mathbb{R}^{p\times p}$
  Do SVD factorization of $\mathbf{T}$: $\mathbf{T} = \mathbf{U\Sigma W}^T$; choose the largest p_d such that $\mathbf{\Sigma}_{ii}/\mathbf{\Sigma}_{11} > \epsilon_d$ for all $i \le p_d$
  $\mathbf{V}_1 = \mathbf{QU}(:, 1:p_d)$
  for $j = 1, \cdots, m$ **do**
    $\mathbf{Z}_j = \mathbf{K}^{-1}\mathbf{V}_j$
    $\mathbf{S} = \mathbf{AZ}_j$
    for $i = 1, \cdots, j$ **do**
      $\mathbf{H}_{i,j} = \mathbf{V}_i^T\mathbf{S} \in \mathbb{R}^{p_d\times p_d}$
      $\mathbf{S} = \mathbf{S} - \mathbf{V}_i\mathbf{H}_{i,j}$
    end for
    Do thin QR factorization of $\mathbf{S}$: $\mathbf{S} = \mathbf{V}_{j+1}\mathbf{H}_{j+1,j}$ with $\mathbf{V}_{j+1} \in \mathbb{R}^{n\times p_d}, \mathbf{H}_{j+1,j} \in \mathbb{R}^{p_d\times p_d}$
    Record $\mathcal{Z}_j = [\mathbf{Z}_1, \cdots, \mathbf{Z}_j] \in \mathbb{R}^{n\times jp_d}$, $\bar{\mathcal{H}}_j = (\mathbf{H}_{k,l})_{1\le k\le j+1, 1\le l\le j} \in \mathbb{R}^{(j+1)p_d\times jp_d}$, where $\bar{\mathcal{H}}_j$ is a block Hessenberg matrix with $\mathbf{H}_{i,j} = 0_{p_d\times p_d}$ for $i > j+1$
    Solve $\mathbf{Y}_j = \text{argmin}_{\mathbf{Y}\in\mathbb{R}^{jp_d\times p_d}}\|\bar{\mathcal{H}}_j\mathbf{Y} - \mathcal{B}_j\|_F$, where $\mathcal{B}_j = \begin{bmatrix} I_{p_d} \\ 0_{jp_d\times p_d} \end{bmatrix}$
    $\mathcal{R}_j = (\bar{\mathcal{H}}_j\mathbf{Y}_j - \mathcal{B}_j)\mathbf{\Sigma}(1:p_d, 1:p_d)\mathbf{W}(1:p, 1:p_d)^T$
    if $\max_l \|\mathcal{R}_j(:,l)\|_2 < tol$ **then**
      $\mathbf{X}_j = \mathbf{X}_0 + \mathcal{Z}_j\mathbf{Y}_j\mathbf{\Sigma}(1:p_d, 1:p_d)\mathbf{W}(1:p, 1:p_d)^T\mathbf{D}$; stop
    end if
  end for
  $\mathbf{X}_0 \leftarrow \mathbf{X}_0 + \mathcal{Z}_m\mathbf{Y}_m\mathbf{\Sigma}(1:p_d, 1:p_d)\mathbf{W}(1:p, 1:p_d)^T\mathbf{D}$
end for

The least-square commutator precontioning [6] can be applied to speed up the

convergence of GMRES:

$$\text{System matrix: } \mathbf{A} = \begin{pmatrix} \mathbf{C} & -\mathbf{D}_{uup} \\ -\mathbf{D}_{uup}^T & \mathbf{0} \end{pmatrix}, \quad \text{Preconditioner: } \mathbf{K}_{LSC} = \begin{pmatrix} \widehat{\mathbf{C}} & -\mathbf{D}_{uup} \\ \mathbf{0} & -\widehat{\mathbf{R}}_{LSC} \end{pmatrix},$$

where $\widehat{\mathbf{C}}$ is a preconditioner for $\mathbf{C}$ given, for instance, by a multigrid V-cycle,

$$\widehat{\mathbf{R}}_{LSC} = (\mathbf{D}_{uup}^T \widehat{\mathbf{M}}_{uu}^{-1} \mathbf{D}_{uup})(\mathbf{D}_{uup}^T \widehat{\mathbf{M}}_{uu}^{-1} \mathbf{C} \widehat{\mathbf{M}}_{uu}^{-1} \mathbf{D}_{uup})^{-1} (\mathbf{D}_{uup}^T \widehat{\mathbf{M}}_{uu}^{-1} \mathbf{D}_{uup}),$$

with $\widehat{\mathbf{M}}_{uu} = \text{diag}(\mathbf{M}_{uu})$. Inside the block GMRES solver, in particular, this preconditioner can be solved by the block CG or block GMRES algorithm with an ILU or multigrid preconditioner.

Remark 5.3. The least-square commutator preconditioner is a competitive choice for our stabilized SAV ensemble algorithms. Although we have discrete Stokes equations to solve in the SAV sub-problems, the considered problem is still a convection dominated flow problem since the Reynolds number is generally not as small as the one in a standard Stokes model. Consequently, the MINRES or GMRES iterative solver with a block diagonal preconditioner

$$\mathbf{K}_{BDia} = \begin{pmatrix} \widehat{\mathbf{C}} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{R}}_{Mass} \end{pmatrix}$$

or a simpler block triangular preconditioner

$$\mathbf{K}_{BTri} = \begin{pmatrix} \widehat{\mathbf{C}} & -\mathbf{D}_{uup} \\ \mathbf{0} & -\widehat{\mathbf{R}}_{Mass} \end{pmatrix}$$

taking $\widehat{\mathbf{R}}_{Mass}$ as scaled pressure mass matrix $\frac{1}{\bar{\nu}+3\alpha h}\mathbf{M}_p$ would not have enough efficiency for the proposed stabilized SAV ensemble algorithms.

We report in Table 1 the performance of three iterative solvers on the test problem in Section 6.1: (1) $\mathbf{K}_{LSC}$ preconditioned GMRES, (2) $\mathbf{K}_{BTri}$ preconditioned GMRES, (3) $\mathbf{K}_{BDia}$ preconditioned MINRES. As expected, for large Reynolds number ($\bar{\nu} = 0.001$), GMRES with preconditioner $\mathbf{K}_{LSC}$ is highly competitive, whereas MINRES with preconditioner $\mathbf{K}_{BDia}$ is not efficient; for small Reynolds number ($\bar{\nu} = 0.1$), there is no significant difference in terms of CPU time.

6. Numerical Experiments. In this section, we perform numerical experiments to validate the stability and accuracy of the Stab-SAV-CN and Stab-SAV-BDF2 ensemble algorithms proposed. The choice of δ value, having been well-studied in the literature, can range from 10^{-8} to 10^6 and is not so significant, so we will not do benchmark tests on this. In this report, the following physical quantities will be used.

$$\text{Energy } = \frac{1}{2}\|u\|^2, \qquad \text{Enstrophy } = \frac{1}{2}\nu\|\nabla \times u\|^2.$$

The plan of this section is as follows. (1) In subsection 6.1, we validate the convergence rate of Stab-SAV-CN and Stab-SAV-BDF2 ensemble algorithms with a large Reynolds number, as compared to [39–41]. In addition, we show the ensemble efficiency of Stab-SAV-BDF2 by comparing with two non-ensemble schemes. (2) In

Table 1: Performance of different iterative solvers: (1) $\mathbf{K}_{LSC}$ preconditioned GMRES, (2) $\mathbf{K}_{BTri}$ preconditioned GMRES, (3) $\mathbf{K}_{BDia}$ preconditioned MINRES. The average number of iteration and average CPU time in each time step solving Stab-SAV-BDF2 sub-problem 1 are denoted by $\bar{n}_{itr,sub1}$ and $\bar{t}_{cpu,sub1}$ respectively; Similarly for Stab-SAV-BDF2 sub-problem 2.

$\bar{\nu}$	α	solver	$\bar{n}_{itr,sub1}$	$\bar{t}_{cpu,sub1}$	$\bar{n}_{itr,sub2}$	$\bar{t}_{cpu,sub2}$	total CPU time
0.001	0.05	(1)	8	3.2s	9	3.7s	552.0s
0.001	0.05	(2)	85	55s	177	180s	15112.0s
0.001	0.05	(3)	164	62s	339	140s	10493.8s
0.1	0	(1)	29	19s	31	21s	2408.5s
0.1	0	(2)	27	13s	38	20s	2202.7s
0.1	0	(3)	54	22s	75	32s	3603.1s

subsection 6.2, we simulate the Kovasznay flow and show the improvement on accuracy by using stabilization techniques we propose, as compared to [41]. (3) In subsection 6.3, we simulate the double driven cavity flow and validate the effectiveness of stabilization techniques we propose. (4) In subsection 6.4, we simulate the flow between two offset cylinders and show the efficiency of Stab-SAV-BDF2 ensemble scheme in simulating complex flows.

6.1. Tests for convergence rate and ensemble efficiency. In this subsection, we take a simple test problem [11] with Green-Taylor vortex solution on a square domain $\Omega = (0,1)^2$ to check the convergence rate of Stab-SAV-CN and Stab-SAV-BDF2 ensemble schemes, and also the ensemble efficiency. The analytical solution of the Navier-Stokes equations (NSE) is given by

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

with $g(t) = e^{\nu}\cos(2t)$. Inhomogeneous Dirichlet boundary condition is imposed on $\partial\Omega$. Initial and boundary conditions are selected to match the prescribed analytical solution. We will consider an ensemble of J members, which are the solutions to NSE corresponding to

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

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

For convergence rate check, we set $J = 3,\ \epsilon_1 = 0,\ \epsilon_2 = 0.1,\ \epsilon_3 = 0.2$. Taking $T = 1,\ h = \Delta t$, we compute approximation solutions to the test problem with both SAV-CN and SAV-BDF2 ensemble algorithms on four successive mesh refinements and corresponding timestep reductions. Specifically, we test two typical cases. In the first case, $\nu_{min} = 0.01$, so the Reynolds number is relatively small and no stabilization is needed in this case for this test problem. We set $\alpha = 0$ in this case, the corresponding results are reported in Table 2 and Table 3. As one can see, both algorithms have second order convergence as predicted.

In the second case, we take $\nu_{min} = 0.001$. The Reynolds number is relatively large in this case and stabilization will be added for the SAV algorithms to converge. SAV algorithms have been studied in several papers [39–41] for the Navier-Stokes equations

Table 2: Errors at $T = 1$ and convergence rates of the SAV-CN ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0$, $\nu_{min} = 0.01$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,2}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,3}$	Rate
1/8	3.205×10^{-2}	—	3.016×10^{-2}	—	2.847×10^{-2}	—
1/16	8.033×10^{-3}	2.00	7.567×10^{-3}	1.99	7.146×10^{-3}	1.99
1/32	2.021×10^{-3}	1.99	1.907×10^{-3}	1.99	1.803×10^{-3}	1.99
1/64	5.070×10^{-4}	1.99	4.784×10^{-4}	1.99	4.523×10^{-4}	1.99
1/128	1.268×10^{-4}	2.00	1.197×10^{-4}	2.00	1.131×10^{-4}	2.00

Table 3: Errors at $T = 1$ and convergence rates of the SAV-BDF2 ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0$, $\nu_{min} = 0.01$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,2}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,3}$	Rate
1/8	3.858×10^{-2}	—	3.631×10^{-2}	—	3.435×10^{-2}	—
1/16	9.350×10^{-3}	2.04	8.847×10^{-3}	2.04	8.407×10^{-3}	2.03
1/32	2.332×10^{-3}	2.00	2.210×10^{-3}	2.00	2.104×10^{-3}	2.00
1/64	5.844×10^{-4}	2.00	5.544×10^{-4}	2.00	5.282×10^{-4}	1.99
1/128	1.465×10^{-4}	2.00	1.390×10^{-4}	2.00	1.325×10^{-4}	2.00

Table 4: Errors at $T = 1$ and convergence rates of the Stab-SAV-CN ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0.2$, $\nu_{min} = 0.001$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,2}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,3}$	Rate
1/8	1.814×10^{-1}	—	1.756×10^{-1}	—	1.701×10^{-1}	—
1/16	4.174×10^{-2}	2.12	3.883×10^{-2}	2.18	3.621×10^{-2}	2.23
1/32	5.289×10^{-3}	2.98	4.908×10^{-3}	2.98	4.610×10^{-3}	2.97
1/64	1.103×10^{-3}	2.26	1.062×10^{-3}	2.21	1.028×10^{-3}	2.17
1/128	2.755×10^{-4}	2.00	2.666×10^{-4}	1.99	2.587×10^{-4}	1.99

Table 5: Errors at $T = 1$ and convergence rates of the Stab-SAV-BDF2 ensemble algorithm ($J = 3$) with $\Delta t = h$, $\alpha = 0.05$, $\nu_{min} = 0.001$.

Δt	$\lvert u_h - u\rvert_{H^1}^{E,1}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,2}$	Rate	$\lvert u_h - u\rvert_{H^1}^{E,3}$	Rate
1/8	1.710×10^{-1}	—	1.660×10^{-1}	—	1.611×10^{-1}	—
1/16	3.367×10^{-2}	2.35	3.099×10^{-2}	2.42	2.872×10^{-2}	2.49
1/32	6.713×10^{-3}	2.33	6.474×10^{-3}	2.26	6.274×10^{-3}	2.19
1/64	1.822×10^{-3}	1.88	1.759×10^{-3}	1.88	1.707×10^{-3}	1.88
1/128	4.750×10^{-4}	1.94	4.581×10^{-4}	1.94	4.432×10^{-4}	1.95

and unconditional stability has been proved and illustrated therein. However, the convergence rates were tested with relatively small Reynolds numbers. For instance, $\nu = 1$ in [39, 40] and $\nu = 0.01$ in [41]. In fact, those algorithms have restrictive timestep conditions for convergence, and the timestep conditions degrade quickly as the Reynolds number increases. We will show with numerical experiments that the timestep restriction for convergence can be relaxed with the stabilization we propose. The corresponding results are reported in Table 4 and Table 5. As we can see, both schemes have second order convergence as predicted.

Specifically, in this test we set $\alpha = 0.2$ for Stab-SAV-CN and $\alpha = 0.05$ for Stab-SAV-BDF2. Slightly larger α values also work fine with acceptable accuracy, but

may have a little impact on desired convergence rate. The choice of the stabilization parameter is application dependent and will need pre-computations to determine an appropriate value. Generally, when α is too small, SAV schemes may not converge for large time steps; when α is too large, it maintains the convergence of S_j^{n+1} but may bring additional errors to the original SAV schemes. In this test Stab-SAV-BDF2 requires smaller value of α than Stab-SAV-CN according to our experiments.

We then report the ensemble efficiency with the number J of samples varying from 1 to 100, $\nu_{min} = 0.001$, ϵ_j being random numbers uniformly distributed in $[0, 0.2]$. We take $h = \Delta t = 1/64$ and run the simulation until $T = 1$. In this particular test, we use the block GMRES algorithm with deflation [4] to solve those discretized linear systems, which is more practicable for large scale simulations, especially for 3D problems. The least-square commutator precontioning is applied to speed up the convergence of GMRES. For the convenience of implementation, this result is based on MATLAB computation while the results in other subsections are obtained by FreeFem++ [18] using direct linear solvers.

Table 6 shows the comparison of CPU time and errors computed by different algorithms with different numbers of ensemble members. It is observed that the Stab-SAV-BDF2 ensemble algorithm outperforms other methods as it takes less CPU times but keeps similar accuracy. Compared with the other non-ensemble schemes, the advantage of the ensemble algorithm becomes more obvious as the ensemble size increases. This is because all the realizations in the ensemble method share a common coefficient matrix, and the associated linear system with different right hand sides can be solved simultaneously.

Table 6: CPU time and errors of mean for different values of J at final time $T = 1$ with $h = \Delta t = 1/64$.

	Stab-SAV-BDF2 Ensemble		Stab-SAV-BDF2 Non-ensemble		BDF2 Non-ensemble	
J	$\|\mathbb{E}[u_h - u]\|_{H^1}$	CPU time	$\|\mathbb{E}[u_h - u]\|_{H^1}$	CPU time	$\|\mathbb{E}[u_h - u]\|_{H^1}$	CPU time
1	1.832×10^{-3}	642.8 s	1.832×10^{-3}	618.4 s	1.669×10^{-3}	733.8 s
10	1.678×10^{-3}	1967.4 s	1.742×10^{-3}	5266.3 s	1.615×10^{-3}	6229.3 s
100	1.669×10^{-3}	11317.4 s	1.735×10^{-3}	53536.7 s	1.610×10^{-3}	62634.2 s

6.2. Kovasznay flow. In this experiment we test the proposed algorithms using the Kovasznay flow [35,41], for which the analytic expression of the flow field at steady state is available. On the domain $\Omega = (-0.5, 1) \times (-0.5, 0.5)$, the Kovasznay flow solution to the NSE (with $f = 0$) is given by the following expressions for the velocity $u = (u_1, u_2)$ and pressure:

$$u_1 = 1 - e^{\lambda x}\cos(2\pi y), \quad u_2 = \frac{\lambda}{2\pi} e^{\lambda x}\sin(2\pi y), \quad p = \frac{1}{2}(1 - e^{2\lambda x}), \tag{6.1}$$

with $\lambda = \frac{1}{2\nu} - \sqrt{\frac{1}{4\nu^2} + 4\pi^2}$. To perform a direct comparision with [41], we also employ a non-dimensional viscosity $\nu = 0.025$ in this test, and consider $J = 1$. Dirichlet boundary conditions are imposed for the velocity according to the above expression, and the external body force is also set to $f = 0$. As was done in [41], we set a zero initial velocity, then perform simulations for a sufficiently long time until the steady state is reached. The flow pattern is shown in Figure 1. At last, we evaluate the errors of the numerical solution against the exact solution given in (6.1). While [41] claims

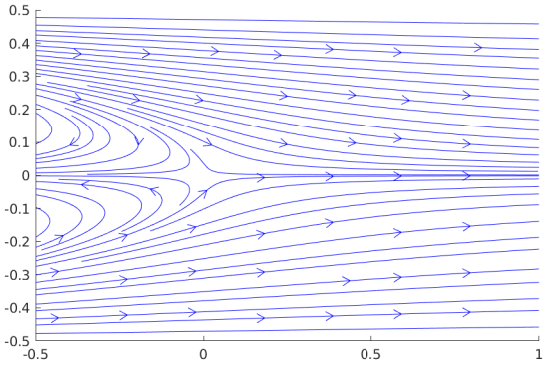


Fig. 1: Kovasznay flow.

that the SAV algorithms allow the use of fairly large time steps to maintain energy stability, here we stress and will use numerical results to illustrate that using pure SAV algorithms it is difficult to achieve desired accuracy. The stabilization techniques we propose play a fundamental role in improving accuracy.

Table 7: Kovasznay flow: L^2 errors of u_1 at $T = 10$ versus time-step size.

h	Δt	Stab-SAV-BDF2 with $\alpha = 0$	Stab-SAV-CN with $\alpha = 0$	Stab-SAV-BDF2 with $\alpha > 0$	Stab-SAV-CN with $\alpha > 0$
1/25	0.5	2.595×10^{-1}	2.878×10^{-1}	1.237×10^{-3}	2.060×10^{-3}
1/25	0.1	1.900×10^{-1}	1.963×10^{-1}	6.541×10^{-5}	6.453×10^{-5}
1/25	0.05	1.516×10^{-1}	1.358×10^{-1}	6.481×10^{-5}	6.443×10^{-5}
1/25	0.02	7.217×10^{-2}	8.254×10^{-2}	6.459×10^{-5}	6.440×10^{-5}
1/25	0.01	1.263×10^{-2}	5.440×10^{-2}	6.454×10^{-5}	6.439×10^{-5}
1/25	0.005	6.452×10^{-5}	6.439×10^{-5}	6.452×10^{-5}	6.438×10^{-5}
1/100	0.5	2.603×10^{-1}	2.874×10^{-1}	4.308×10^{-3}	1.952×10^{-3}
1/100	0.1	1.896×10^{-1}	1.868×10^{-1}	6.942×10^{-6}	2.898×10^{-6}
1/100	0.05	1.435×10^{-1}	1.368×10^{-1}	3.132×10^{-6}	1.447×10^{-6}
1/100	0.02	7.338×10^{-2}	8.044×10^{-2}	1.452×10^{-6}	1.029×10^{-6}
1/100	0.01	1.203×10^{-2}	2.212×10^{-2}	1.093×10^{-6}	9.726×10^{-7}
1/100	0.005	9.869×10^{-7}	3.919×10^{-6}	9.904×10^{-7}	9.588×10^{-7}

Similar to [41], in Table 7, we have listed the L^2 errors of u_1 at final time $T = 10$, computed with different time steps, ranging from $\Delta t = 0.5$ to $\Delta t = 0.005$. In the simulations we use both a coarse mesh ($h = 1/25$) and a fine mesh ($h = 1/100$). As one can see from the table, although the computation using pure SAV algorithms (i.e. Stab-SAV-BDF2 and Stab-SAV-CN with $\alpha = 0$) does not blow up when Δt is large, the computational accuracy is far from an acceptable level. There exists a gap on accuracy from $\Delta t = 0.01$ to $\Delta t = 0.005$. This gap is caused by the fact that the approximate ratio $\frac{q_j^n}{R_j^n}$, where $R_j^n = \sqrt{E(u_j^n) + \delta}$, evolves to one only with $\Delta t = 0.005$ but not with $\Delta t = 0.01$. This fact is illustrated in the left of Figure 2, in which we plot the time histories of the numerical solutions q_j^n, $R_j^n = \sqrt{E(u_j^n) + \delta}$, and S_j^n. With small Δt values, 0.005 for instance, the two quantities q_j^n and R_j^n are eventually close,

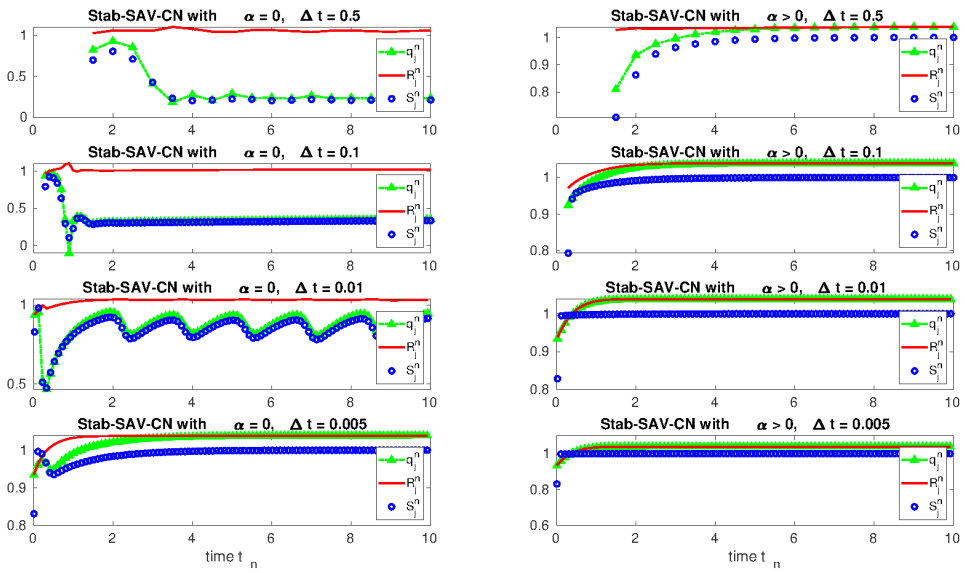
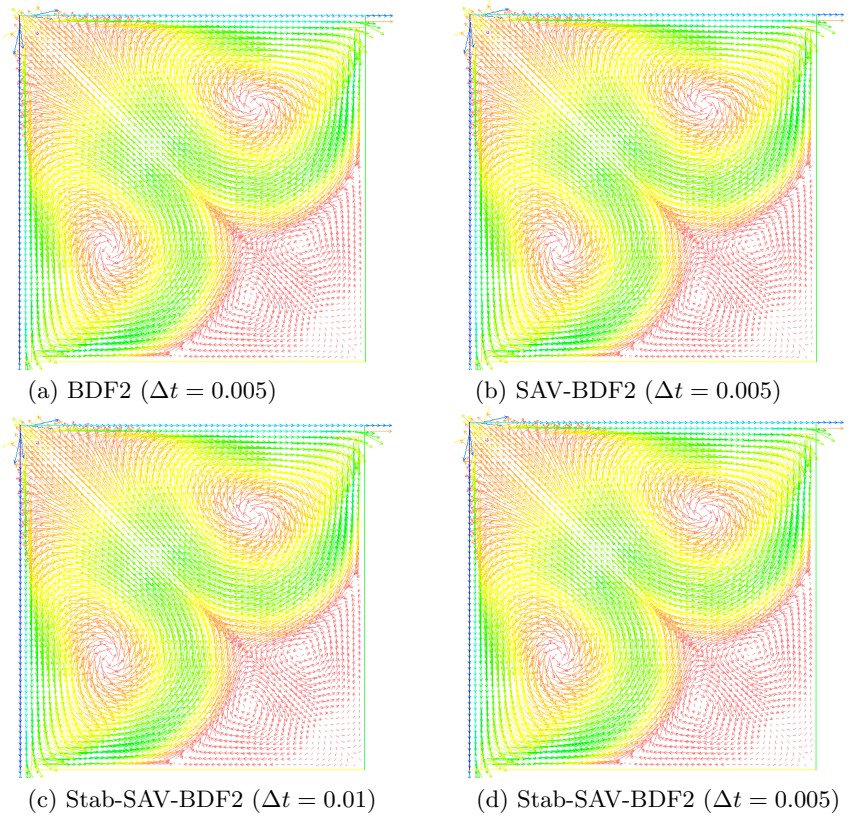


Fig. 2: Kovasznay flow: time histories of q_j^n, $R_j^n = \sqrt{E(u_j^n) + \delta}$, and S_j^n obtained using fixed $h = 1/100$ and various time step sizes: $\Delta t = 0.5,\ 0.1,\ 0.01,\ 0.005$. Here $J = 1 = j$. Left: pure SAV-CN scheme; right: stabilized SAV-CN scheme.

which is evident from the picture in bottom-left. When Δt is greater than 0.01, one can observe apparent difference between $q(t)$ and $R(t)$. The discrepancy grows larger with increasing Δt, see the first three pictures in the left of Figure 2. While $R(t)$ stays at an almost constant level, $q(t)$ and hence the auxiliary ratio $\frac{q(t)}{R(t)}$ in equation (2.5) appears to be driven towards zero with a large Δt. This automatic adjustment on the nonlinear term in the NSE seems to explain the stable feature of pure SAV algorithms. However, such mechanism does not guaratee the accuracy.

In contrast, when the stabilization techniques are added, the accuracy of the numerical solution obtained using a large Δt can be significantly improved. For instance, when $h = 1/25, \Delta t = 0.1, \delta = 0.01$, by setting $\alpha = 1$ in the Stab-SAV-BDF2 scheme, the error can be reduced to 6.541×10^{-5} as compared to 1.900×10^{-1} computed with $\alpha = 0$. Similarly, Stab-SAV-CN with $\alpha = 2$ is also enough to derive accurate simulations even when Δt is as large as 0.1. We also mention that the optimal choice of α is generally unknown and needs pre-computations to determine. The experience for Kovasznay flow is that one may use a slightly larger α for finer mesh to achieve desired accuracy with large Δt. For instance, when $h = 1/100$, we set $\alpha = 5$ for Stab-SAV-BDF2 and $\alpha = 10$ for Stab-SAV-CN. Slightly smaller α values also work fine, but lose a litte accuracy with large Δt mostly because of the discrepancy between $q(t)$ and $R(t)$. In the right of Figure 2, one can observe the effectiveness of the stabilization in forcing the convergence of $q(t)$ to $R(t)$. If α is too small, the time histories of $q(t)$ and $R(t)$ will be in a pattern between the left and the right of Figure 2.

6.3. Double driven cavity flow. In this example, we take a classical benchmark problem, the two-dimensional driven cavity problem [3], to test the Stab-SAV-BDF2 scheme proposed. The driven cavity problem consists of a viscous incompressible flow in the square $\Omega = (0, 1)^2$. Specifically, the double driven cavity flow is driven by two sides of the boundary: $(u_1, u_2) = (1, 0)$ on $y = 1$ and $(u_1, u_2) = (0, -1)$ on



(a) BDF2 ($\Delta t = 0.005$)

(b) SAV-BDF2 ($\Delta t = 0.005$)

(c) Stab-SAV-BDF2 ($\Delta t = 0.01$)

(d) Stab-SAV-BDF2 ($\Delta t = 0.005$)

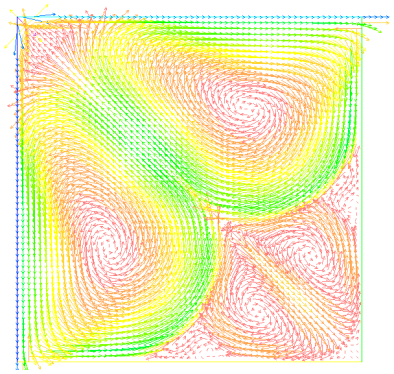
Fig. 3: Velocity fields of the double driven cavity flow at $T = 60$, computed with $\nu = 10^{-3}, h = 1/32$ and by different schemes. Consider (a) as a reference, while the pure SAV-BDF2 scheme in (b) gives accurate result, it does not work with $\Delta t = 0.01$ (simulation will blow up); whereas (c) shows the effectiveness of Stab-SAV-BDF2 with large $\Delta t = 0.01$.

$x = 0$. On the other parts of the boundary, the no-slip boundary condition is imposed, i.e. $u = 0$. The external body force is set to $f = 0$. We set a zero initial velocity, and then run simulations for a sufficiently long time to study the stability and accuracy of the Stab-SAV-BDF2 scheme.

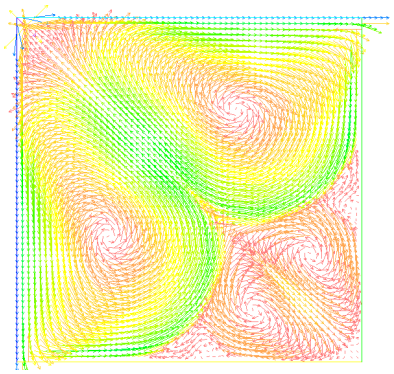
We will demonstrate that for this test (1) the timestep condition for convergence can be relaxed by the stabilization technique we propose; (2) one has to use a stabilization technique for large Reynolds numbers such as $Re = 10^4$.

For the case of $\nu = 10^{-3}$, we plot in Figure 3 the velocity fields of the double driven cavity flow at $T = 60$, computed with $h = 1/32$ and by different schemes. Consider the simulation by the BDF2 scheme with $\Delta t = 0.005$ as a reference. While the pure SAV-BDF2 scheme with $\Delta t = 0.005$ gives accurate results, it does not work with $\Delta t = 0.01$ (the simulation will blow up); whereas a simulation by Stab-SAV-BDF2 ($\alpha = 1$) with Δt as large as 0.01 is already accurate enough. This demonstrates the effectivenss of relaxing timestep condition for convergence by the stabilization.

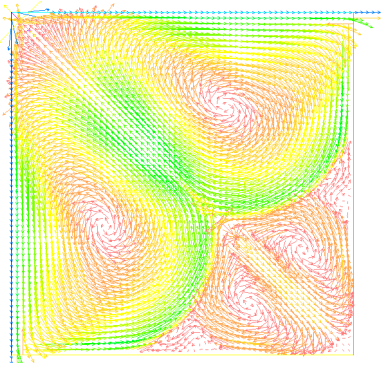
We then study the case of $\nu = 10^{-4}$. In Figure 4, we plot the velocity fields of the double driven cavity flow at $T = 60$, computed with $h = 1/64$ and by different schemes. Consider the simulation by BDF2 scheme with $\Delta t = 0.001$ as a reference. While the pure SAV-BDF2 scheme with $\Delta t = 0.001$ gives accurate results, it does not



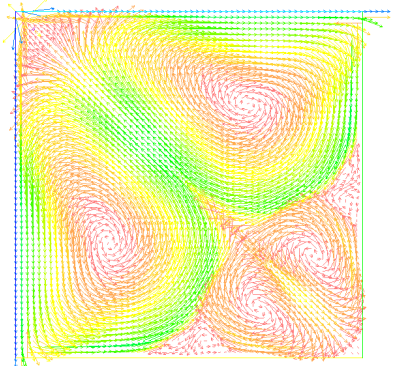
(a) BDF2 ($\Delta t = 0.001$)



(b) Stab-SAV-BDF2 ($\Delta t = 0.01$)



(c) Stab-SAV-BDF2 ($\Delta t = 0.005$)



(d) Stab-SAV-BDF2 ($\Delta t = 0.001$)

Fig. 4: Velocity fields of the double driven cavity flow at $T = 60$, computed with $\nu = 10^{-4}, h = 1/64$ and by different schemes. Consider (a) as a reference. The pure SAV-BDF2 scheme gives accurate results with $\Delta t = 0.001$ but not with $\Delta t = 0.005$ (simulation will blow up); (b) shows the effectiveness of Stab-SAV-BDF2 with large $\Delta t = 0.01$.

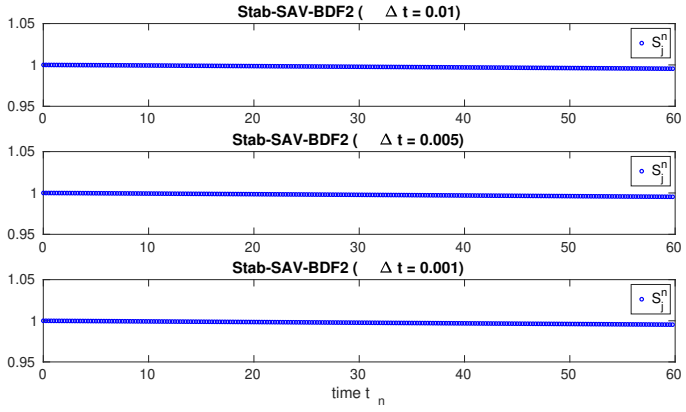


Fig. 5: Double driven cavity flow: time histories of S_j^n computed by the Stab-SAV-BDF2 scheme with $\nu = 10^{-4}, h = 1/64, J = 1 = j$.

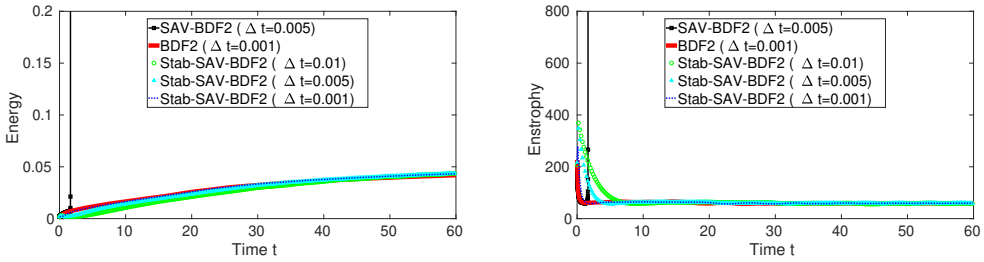


Fig. 6: Time histories of energy and enstrophy of the double driven cavity flow computed with $\nu = 10^{-4}, h = 1/64, T = 60$ and by different schemes.

work with $\Delta t = 0.005$ (the simulation will blow up); whereas a simulation by Stab-SAV-BDF2 ($\alpha = 1$) with Δt as large as 0.01 is already stable and accurate. We also plot the time history of S_j^n in Figure 5. The time histories of energy and enstrophy of the flow computed by different schemes are plotted in Figure 6.

This study shows the importance of adding stabilizations for simulating complex flows. When $\nu = 10^{-3}$, a timestep size 0.005 is enough for the convergence of pure SAV-BDF2; whereas Δt needs to be reduced to 0.001 when $\nu = 10^{-4}$, which corresponds to a more complex flow. This drawback can be overcome by our Stab-SAV-BDF2 method. For both $\nu = 10^{-3}$ and $\nu = 10^{-4}$, a large $\Delta t = 0.01$ is good enough for stable simulations with Stab-SAV-BDF2.

In the end, we report in Table 8 the CPU time for simulating the double driven cavity flow with $\nu = 10^{-4},\ h = 1/64,\ T = 60$. As expected, with the same timestep size the Stab-SAV-BDF2 scheme results in linear systems that are much cheaper to solve as compared with BDF2, since we treat the nonlinear term of the NSE explicitly. The BDF2 scheme mostly converges with large Δt in this example, but this is not always the case in other situations. We will see a typical example in Section 6.4 (Table 9) showing that BDF2 must be applied with very small Δt. In that case, Stab-SAV-BDF2 scheme will markedly save computational time since it has a relaxed timestep condition.

Table 8: CPU time for simulating the double driven cavity flow with $\nu = 10^{-4},\ h = 1/64,\ T = 60$.

Scheme	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$
BDF2	13140.4s	26531.3s	132718.0 s
Stab-SAV-BDF2	4636.1 s	9018.0 s	44177.4 s

6.4. Flow between two offset cylinders. In this example, we consider a 2D flow between two offset cylinders, and test the stability and efficiency of the proposed Stab-SAV-BDF2 ensemble algorithm. The domain is a disk with a smaller off center obstacle inside. It is given by

$$\Omega = \{(x_1, x_2) : x_1^2 + x_2^2 \le 1 \text{ and } (x_1 - 0.5)^2 + x_2^2 \ge 0.1^2\}.$$

We enforce no-slip boundary conditions on both circles (i.e. homogeneous Dirichlet boundary condition). The flow is driven by a counterclockwise rotational body force

$$f(x_1, x_2) = \left(-4x_2(1 - x_1^2 - x_2^2), 4x_1(1 - x_1^2 - x_2^2)\right)^T.$$

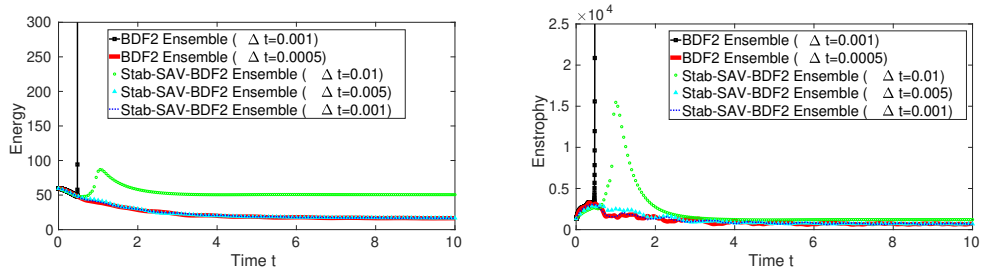


Fig. 7: Time histories of energy and enstrophy of the flow between two offset cylinders, computed by different ensemble schemes. We take $J = 2$, $\nu_1 = 0.008$, $\nu_2 = 0.01592$, and the plots correspond to ν_1.

It interacts with the inner circle generating complex flow structures. Specifically it forms a Von Kármán vortex street that reinteracts with the inner circle. Extensive experiments on this flow of a first order ensemble method have been studied in [27] and a second order scheme in [23]. To generate perturbations of the initial conditions, we solve the steady Stokes problem with J perturbed body forces given by

$$f_j(x_1, x_2) = f(x_1, x_2) + \epsilon_j \left(\sin(3\pi x)\sin(3\pi y), \cos(3\pi x)\cos(3\pi y)\right)^T, \ j = 1, \cdots, J,$$

where ϵ_j is uniformly distributed in $[10^{-3}, 10^{-2}]$. This gives us J discretely divergence free initial conditions. One should notice that the flow generated in this way is generally much more complicated than previous examples.

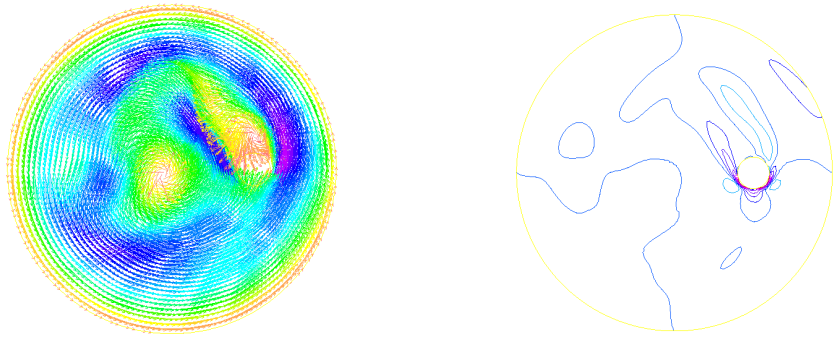
The Stab-SAV-BDF2 ensemble scheme will be compared to a standard BDF2 ensemble algorithm [23]. It has been studied in [23] that the BDF2 ensemble algorithm requires a CFL condition

$$\Delta t \|\nabla u_{j,h}'^n\|^2 \le C\nu_j h, \quad \text{where } u_{j,h}'^n = 2u_{j,h}^n - u_{j,h}^{n-1} - \frac{1}{J}\sum_{j=1}^{J}(2u_{j,h}^n - u_{j,h}^{n-1}),$$

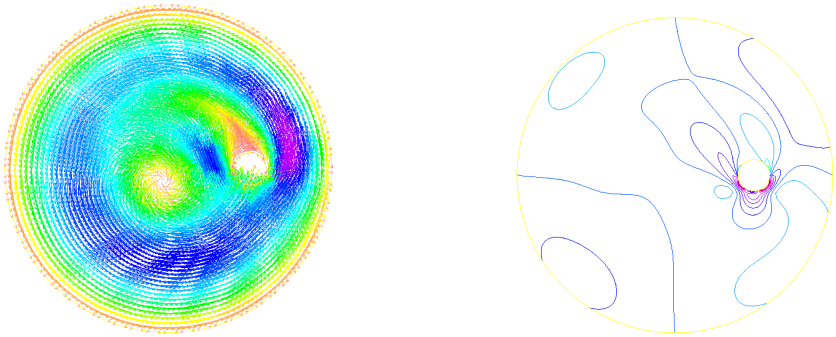
to guarantee stability. This is apparent in the simulation of the flow between two offset cylinders. When $J = 2$, $\nu_1 = 0.01$, $\nu_2 = 0.0199$, simulation by BDF2 ensemble with $\Delta t = 0.005$ will blow up. The situation is worse when ν_j is smaller. For instance, when $J = 2$, $\nu_1 = 0.008$, $\nu_2 = 0.01592$, the simulation by BDF2 ensemble with $\Delta t = 0.001$ will blow up. This has been illustrated in Figure 7, which plots the time histories of energy and enstrophy of the flow computed by different ensemble schemes with $h = 1/64$, $T = 10$, $\delta = 100$, $\alpha = 10$. The curves correspond to ν_1 in particular.

The stability and accuracy of Stab-SAV-BDF2 ensemble scheme with relatively large timesteps are also illustrated in Figure 8. Specifically, the left of Figure 8 plots the velocity fields of the flow computed by different ensemble schemes with $J = 2$, $\nu_1 = 0.01$, $\nu_2 = 0.0199$, $h = 1/64$, $T = 10$, $\delta = 100$, $\alpha = 5$. Consider the simulation by BDF2 ensemble scheme with $\Delta t = 0.001$ as a reference. While this scheme gives accurate results with $\Delta t = 0.001$, it does not work with $\Delta t = 0.005$ (the simulation will blow up); whereas a simulation by Stab-SAV-BDF2 ensemble with Δt as large as 0.01 is already stable. More accurate results can be achieved with Δt reducing to 0.005 or 0.001. The right of Figure 8 plots the corresponding vorticity contours of the flow. Moreover, we plot the time history of S_j^n in Figure 9.

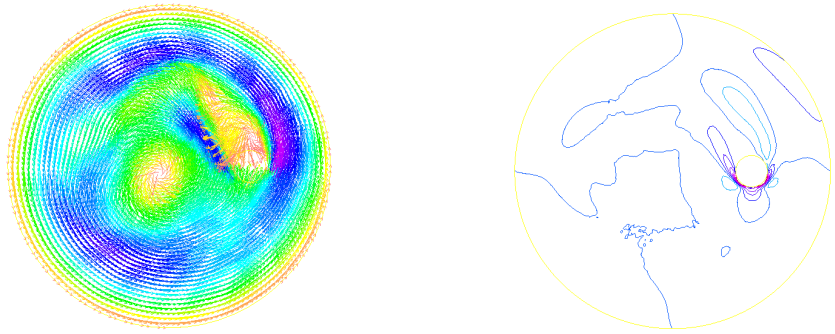
In the end, we report in Table 9 the CPU time for simulations with $J = 2$, $\nu_1 = 0.01$, $\nu_2 = 0.0199$, $h = 1/64$, $T = 10$. As we expect, with the same time-step



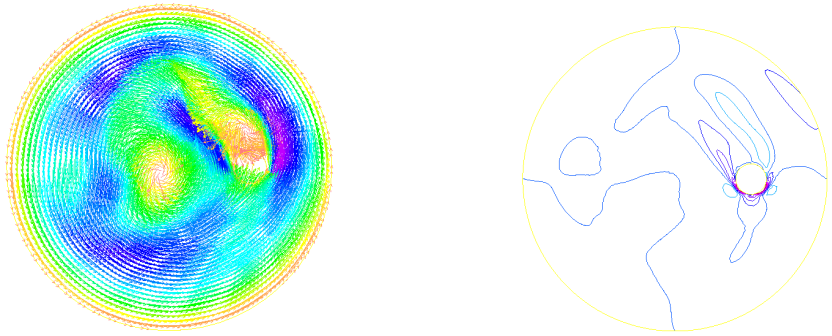
(a) BDF2 Ensemble ($\Delta t = 0.001$)



(b) Stab-SAV-BDF2 Ensemble ($\Delta t = 0.01$)



(c) Stab-SAV-BDF2 Ensemble ($\Delta t = 0.005$)



(d) Stab-SAV-BDF2 Ensemble ($\Delta t = 0.001$)

Fig. 8: Velocity fields (left column) and vorticity contours (right column) of the flow between two offset cylinders at $T = 10$, computed by different ensemble schemes with $J = 2,\ \nu_1 = 0.01, \nu_2 = 0.0199,\ h = 1/64$. Consider (a) as a reference. The BDF2 ensemble scheme gives accurate results with $\Delta t = 0.001$ but not with $\Delta t = 0.005$ (simulation will blow up), (b) shows the stability of Stab-SAV-BDF2 ensemble scheme with large $\Delta t = 0.01$, (c) and (d) shows more accurate simulations with reduced Δt.

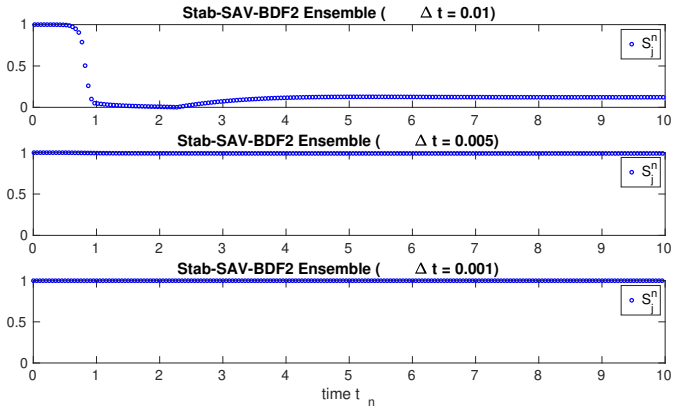


Fig. 9: Flow between two offset cylinders: time histories of S_j^n computed by the Stab-SAV-BDF2 ensemble scheme with $J = 2$, $\nu_1 = 0.01$, $\nu_2 = 0.0199$, $h = 1/64$, $T = 10$. The plot corresponds to $j = 1$.

size $\Delta t = 0.001$, the Stab-SAV-BDF2 scheme takes much less computational time than BDF2. While BDF2 ensemble scheme fails for the choices of $\Delta t = 0.01$ and $\Delta t = 0.005$, the Stab-SAV-BDF2 ensemble scheme markedly saves computational time since it works fine with larger time steps.

Table 9: CPU time for simulating the flow between two offset cylinder with $J = 2$, $\nu_1 = 0.01$, $\nu_2 = 0.0199$, $h = 1/64$, $T = 10$. The X symbol means a simulation failure.

Scheme	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$
BDF2 Ensemble	X	X	105448.0 s
Stab-SAV-BDF2 Ensemble	5243.4 s	10202.0 s	48144.5 s

7. Conclusions. We proposed two second order, stabilized, SAV-based, ensemble algorithms for fast computation of incompressible flow ensembles: Stab-SAV-CN and Stab-SAV-BDF2. The two ensemble algorithms are extremely efficient in that they result in a common constant coefficient matrix for all realizations of the flow equations so that efficient block GMRES solvers can be applied to significantly reduce the computational cost. The algorithms are based on a recently introduced SAV idea that makes it possible to construct unconditionally stable schemes while treating the nonlinear term fully explicitly. The incorporation of the SAV idea improves the efficiency of the ensemble algorithms by making the common coefficient matrix independent of the time index n and leads to provable long time stable ensemble schemes, without any timestep conditions. We also introduced a stabilization method to the SAV ensemble algorithms, which evidently improves the accuracy of SAV-based methods making them more suitable for simulations of practical flows. We proved that the two stabilized ensemble algorithms are both long time stable under a parameter fluctuation condition. Ample numerical experiments were performed and test results showed that the stabilization is able to substantially improve the accuracy of SAV

based schemes and the porposed stabilized SAV ensemble algorithms are extremely efficient, e.g., Table 6 showed the Stab-SAV-BDF2 ensemble algorithm can save 82% of the CPU time compared with a traditional BDF2 non-ensemble method, and 78% of the CPU time compared with a standard but stabilized SAV-BDF2 non-ensemble method, when computing with $J = 100$ realizations of the flow.

REFERENCES

[1] M. Anitescu, F. Pahlevani and W. Layton, *Implicit for local effects and explicit for nonlocal effects is unconditionally stable*, Electronic Transactions on Numerical Analysis, 18 (2004), 174-187.

[2] 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.

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

[4] 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.

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

[6] 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.

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

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

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

[22] H. Ji and Y. Li, *A breakdown-free block conjugate gradient method*, BIT Numerical Mathe-

matics, 57(2) (2017), 379-403.
[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 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.
[31] 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.
[32] V. John and S. Kaya, *Finite element error analysis of a variational multiscale method for the Navier-Stokes equations*, Advances in Computational Mathematics, 28 (2008), 43-61.
[33] V. John, S. Kaya and A. Kindl, *Finite element error analysis for a projection-based variational multiscale method with nonlinear eddy viscosity*, Journal of Mathematical Analysis and Applications, 344 (2008), 627-641.
[34] L. Ju, W. Leng, Z. Wang and S. Yuan, *Numerical investigation of ensemble methods with block iterative solvers for evolution problems*, Discrete and Continuous Dynamical Systems - Series B, 25 (2020), 4905-4923.
[35] L.I.G. Kovasznay, *Laminar flow behind a two-dimensional grid*, Proc. Camb. Philol. Soc. 44 (1948) 58.
[36] 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.
[37] A. Labovsky, W. Layton, C. Manica, M. Neda and L. Rebholz, *The stabilized extrapolated trapezoidal finite-element method for the Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), 958-974.
[38] N. Li, J. Fiordilino and X. Feng, *Ensemble time-stepping algorithm for the convection-diffusion equation with random diffusivity*, Journal of Scientific Computing, 79 (2019), 1271-1293.
[39] 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.
[40] X. Li, J. Shen and Z. Liu, *New SAV-pressure correction methods for the Navier-Stokes equations: stability and error analysis*, arXiv preprint arXiv:2002.09090, (2020).
[41] L. Lin, Z. Yang and S. Dong, *Numerical approximation of incompressible Naiver-Stokes equations based on an auxiliary energy variable*, Journal of Computational Physics, 388 (2019), 1-22.
[42] Y. Luo and Z. Wang, *An ensemble algorithm for numerical solutions to deterministic and random parabolic PDEs*, SIAM Journal on Numerical Analysis, 56 (2018), 859-876.
[43] Y. Luo and Z. Wang, *A multilevel Monte Carlo ensemble scheme for random parabolic PDEs*, SIAM Journal on Scientific Computing, 41 (2019), A622-A642.
[44] J.F. McCarthy, *Block-conjugate-gradient method*, Physical Review D, 40 (1989), 2149.
[45] M. Mohebujjaman and L. Rebholz, *An efficient algorithm for computation of MHD flow ensembles*, Computational Methods in Applied Mathematics, 17 (2017), 121-137.
[46] D.P. O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra and its Applications, 29 (1980), 293-322.
[47] 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.

[48] V. Romero, J. Burkardt, M. Gunzburger and J. Peterson, *Comparison of pure and "Latinized" centroidal Voronoi tessellation against various other statistical sampling methods*, Reliability Engineering and System Safety, 91 (2006), 1266-1280.

[49] 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.

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

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

[52] 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.

[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.