FAST AND ACCURATE ARTIFICIAL COMPRESSIBILITY ENSEMBLE ALGORITHMS FOR COMPUTING PARAMETERIZED STOKES-DARCY FLOW ENSEMBLES

NAN JIANG* AND HUANHUAN YANG †

Abstract. Accurate simulations of the Stokes-Darcy system face many difficulties including the coupling of flows in two different subdomains via interface conditions, the incompressibility constraint for the free flow and uncertainties in model parameters. In this report, we propose and study efficient, decoupled, artificial compressibility (AC) ensemble schemes based on a recently developed SAV approach for fast computation of Stokes-Darcy flow ensembles. The proposed algorithms (1) do not require any time step condition and (2) decouple the computation of the velocity and pressure in the free flow region, and (3) result in a common coefficient matrix for all realizations after spatial discretization for which efficient iterative linear solvers such as block CG or block GMRES can be used to greatly reduce the computational cost. We prove the long time stability under two parameter conditions, *without* any timestep constraints. In particular, for one single simulation, they are unconditionally stable schemes. Several numerical tests are presented to demonstrate the efficiency of the algorithms and illustrate their applications in realistic flow problems.

1. Introduction. The Stokes-Darcy model arises in many geophysical and biological applications, such as the coupling of surface and groundwater flows, oil filters and blood filtration through vessel walls. Its numerical approximation has been a subject of intensive study for recent decades, see [6, 8, 15, 16, 19, 25, 46, 60, 63, 64, 73, 80] and the references therein. Despite the fact that highly accurate numerical methods have become available, accurate simulations and predictions of the coupled flows are still infeasible in many engineering and industrial applications due to inherent uncertainties in the physical parameters and high demand for computer resources for uncertainty quantification (UQ). For many popular UQ methods, e.g., multilevel Monte Carlo method [2], quasi Monte Carlo method [58], centroidal Voronoi tessellations [77], Latin hypercube sampling [38], non-intrusive polynomial chaos methods [37, 76], stochastic collocation methods [1, 88, 74, 24], the computational burden lies in repeated simulations of the underlying physical model within a prescribed simulation time window. For instance, in numerical weather prediction, the simulation needs to be finished in a certain amount of time for the prediction to be useful. In such simulations, spatial resolution is often sacrificed to balance the need of computing a sufficient number of realizations and the limitation of available computer resources.

To undertake this challenge, efficient ensemble algorithms [47, 48, 43, 35, 36] have recently been proposed to facilitate fast and accurate simulations of flow ensembles. These are specially designed algorithms that compute all the realizations at one pass resulting in linear systems with a common coefficient matrix that can be efficiently solved with block solvers, such as block CG [22], block GMRES [23]. This ensemble timestepping idea has been extensively tested in different flow problems and shown to be highly efficient in terms of both storage and computational time, e.g., Navier-Stokes equations [33, 34, 32, 42, 43, 45, 54, 55, 83, 84], natural convection [20, 21, 44], fluid-fluid interaction [12], MHD flows [7, 53]. Moreover it is competitive in accuracy in comparison to traditional numerical methods. For Stokes-Darcy flows with random hydraulic conductivity, fast ensemble methods were developed in [39, 51, 52, 49, 50].

Timestep restrictions remain a significant limiting factor in practical simulations of fluid flows through highly heterogeneous porous media. Partitioned methods are growing popular due to their ability to reduce the size of the discrete problems and thus less computer resources for a prescribed accuracy. But they have to face extra timestep constraints from the decoupling step [63, 64], which degrade quickly as the porous media become increasingly heterogeneous. In the setting of ensemble simulations, this difficulty remains and may worsen as the fluctuation of the parameter appears in the timestep condition and a larger fluctuation may result in a stricter timestep condition [51, 39, 52]. Thus it is desirable to design a more robust decoupling strategy to avoid such situations and ensure the partitioned schemes achieve their expected efficiency. In [56], a new decoupling strategy was proposed based on the scalar auxiliary variable (SAV) idea, where a SAV was introduced to handle the lagged coupling terms in the Stokes-Darcy model. This leads to a modified partial differential equation (PDE) system which is equivalent to the original Stokes-Darcy model, and an efficient, fully decoupled discrete system can be derived and proved to be long time stable without any timestep constraints. It is worth noting in [49] an unconditionally stable numerical scheme was developed based on a combination of the Crank-Nicolson Leapfrog (CNLF) timestepping and an artificial compressibility

*Department of Mathematics, University of Florida, Gainesville, FL 32611, jiangn@ufl.edu.
†Department of Mathematics, Shantou University, Guangdong, China 515063, huan2yang@stu.edu.cn.

(AC) technique. Despite its good performance, it does not provide a general framework for developing unconditionally stable schemes based on other timestepping methods. Instead, the SAV decoupling strategy studied in [56] can be easily extended to other popular timestepping methods and produce a family of unconditionally stable numerical schemes. In this report we build on the idea of [56] and incorporate it with the artificial compressibility technique to develop more efficient ensemble algorithms for numerical simulations of the Stokes-Darcy flows subject to uncertainties.

Let D_f denote the surface fluid flow region and D_p the porous media flow region, where $D_f, D_p \subset \mathbb{R}^d (d = 2, 3)$ are both open, bounded domains. These two domains lie across an interface, I, from each other, and $D_f \cap D_p = \emptyset, \bar{D}_f \cap \bar{D}_p = I$, see Figure 1.1.

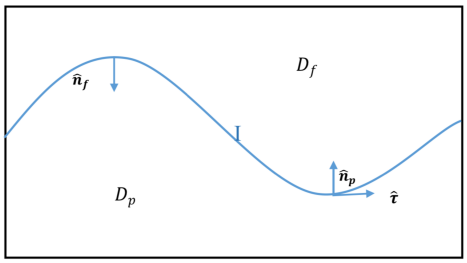


Fig. 1.1: A sketch of the porous median domain D_p, fluid domain D_f, and the interface I.

We consider computing an ensemble of J Stokes-Darcy systems [3, 15, 63], corresponding to J different parameter sets $(u_j^0, \phi_j^0, a_j, b_j, f_{f,j}, f_{p,j}, \mathcal{K}_j)$, $j = 1, ..., J$,

$$\begin{aligned} \partial_t u_j - \nu \Delta u_j + \nabla p_j = f_{f,j}(x,t),\ \nabla \cdot u_j = 0, \quad \text{in} \quad D_f, \\ S_0 \partial_t \phi_j - \nabla \cdot (\mathcal{K}_j(x) \nabla \phi_j) = f_{p,j}(x,t), \quad \text{in} \quad D_p, \\ \phi_j(x,0) = \phi_j^0(x), \ \text{in } D_p \text{ and } u_j(x,0) = u_j^0(x), \ \text{in } D_f, \\ \phi_j(x,t) = b_j(x,t), \ \text{in } \partial D_p \backslash I \text{ and } u_j(x,t) = a_j(x,t), \ \text{in } \partial D_f \backslash I. \end{aligned} \tag{1.1}$$

Here $u(x,t)$ is the fluid velocity , $p(x,t)$ the fluid pressure and $\phi(x,t)$ the hydraulic head. We assume there are uncertainties in initial conditions $u^0(x), \phi^0(x)$, Dirichlet boundary conditions $a(x,t), b(x,t)$, forcing terms $f_f(x,t), f_p(x,t)$ and the hydraulic conductivity tensor $\mathcal{K}(x)$, and $(u_j^0, \phi_j^0, a_j, b_j, f_{f,j}, f_{p,j}, \mathcal{K}_j)$ is one of the samples drawn from the respective probabilistic distributions, J is the number of total samples. For the simplicity of presentation we assume homogeneous Dirichlet boundary condition here, but inhomogeneous Drichlet boundary conditions will be implemented and tested in the numerical experiments shown in Section 5. Let $\widehat{n}_{f/p}$ denote the outward unit normal vector on I associated with $D_{f/p}$, where $\widehat{n}_f = -\widehat{n}_p$. The coupling conditions across I are conservation of mass, balance of forces and the Beavers-Joseph-Saffman condition on the tangential velocity:

$$\begin{aligned} u \cdot \widehat{n}_f - \mathcal{K} \nabla \phi \cdot \widehat{n}_p = 0 \text{ and } p - \nu\, \widehat{n}_f \cdot \nabla u \cdot \widehat{n}_f = g\phi \text{ on } I, \\ -\nu \widehat{\tau}_i \cdot \nabla u \cdot \widehat{n}_f = \frac{\alpha_{\text{BJS}}}{\sqrt{\widehat{\tau}_i \cdot \mathcal{K} \widehat{\tau}_i}} u \cdot \widehat{\tau}_i \ \text{ on } I, \text{ for any tangential vector } \widehat{\tau}_i \text{ on } I. \end{aligned}$$

see [4], [79], [40]. Here, g, $\mathcal{K}$, ν and S_0 are the gravitational acceleration constant, hydraulic conductivity tensor, kinematic viscosity and specific mass storativity coefficient, respectively, which are all positive. The conductivity tensor $\mathcal{K}$ is assumed to be symmetric positive definite (SPD).

We first introduce the scalar auxiliary variables $r_j(t)$ that will be added to the governing PDE system and will play an essential role in the decoupling scheme. Following the SAV ideas in [56, 68], we define

$$r_j(t) = exp(-\frac{t}{T}), \qquad j = 1, \cdots, J. \tag{1.2}$$

The true solutions of these scalar variables r_j are the same but the approximate solutions from the decoupling ensemble schemes will be different. They satisfy the following differential equations.

$$\frac{dr_j}{dt} = -\frac{1}{T} r_j + \frac{1}{exp(-\frac{t}{T})} \left(c_I(u_j, \phi_j) - c_I(u_j, \phi_j) \right), \tag{1.3}$$

where $c_I(u, \phi)$ is the coupling term from the Stokes-Darcy system, defined by

$$c_I(u, \phi) = g \int_I \phi u \cdot \widehat{n}_f \ ds.$$

To decouple the original Stokes-Darcy system into two subphysics problems, one usually needs to lag the coupling term $c_I(u, \phi)$ to the previous time steps and this inevitably results in a timestep condition to ensure long time stability [63, 64] if no other stabilization/decoupling techniques are considered. The introduction of the zero term $c_I(u_j, \phi_j) - c_I(u_j, \phi_j)$ in (1.3) makes it possible to cancel out the same term in the partitioned scheme with lagged coupling terms.

One other difficulty in numerical solution of the Stokes-Darcy equations is the incompressibility constraint in the free flow region. The projection-type methods are the most frequently employed techniques to overcome this difficulty, which feature lagging the pressure term to the previous time steps and then projecting the intermediate velocity into the divergence-free space in a later step, [29]. They were first introduced by Chorin [10] and Temam [86] in the late sixties and have been intensively studied since then [17, 26, 30, 31, 81, 82, 87] and widely used in the practice. Unlike the projection-type methods, the artificial compressibility methods (AC) which were also first studied in the sixties by Chorin [11], Temam [85, 86], Kuznetsov, Vladimirova and Yanenko [59], are less studied in the literature and have only recently received increasing attention. The AC methods relax the incompressibility constraint by adding a perturbation, e.g., ϵp_t, to the mass conservation equation which facilities decoupling the computation of velocity and pressure in time marching schemes, see [13, 9, 62, 27, 28, 14, 39, 67, 78] for recent developments. We will incorporate the AC technique with our SAV decoupling strategy to develop highly efficient ensemble simulation schemes. It is worth noting the outstanding feature of the AC schemes is that the pressure can be updated directly without solving a Poisson's equation which avoids the spurious oscillations in the boundary layer of pressure due to artificial boundary conditions.

Define the function spaces:

$$\begin{aligned} \text{Velocity:} \quad & X_f := \{ v \in \left(H^1(D_f) \right)^d : v = 0 \text{ on } \ \partial D_f \backslash I \}, \\ \text{Pressure:} \quad & Q_f := L^2(D_f) \\ \text{Hydraulic Head:} \quad & X_p := \{ \psi \in H^1(D_p) : \psi = 0 \text{ on } \ \partial D_p \backslash I \}. \end{aligned}$$

We next present two SAV decoupled artificial compressibility ensemble algorithms that based on the Backward Euler (BE) and the second order backward differentiation formula (BDF2) respectively, for fast computation of the Stokes-Darcy flow ensembles.

ALGORITHM 1.1 (SAV-BEAC-En). *Find* $(u_j^{n+1}, p_j^{n+1}, \phi_j^{n+1}) \in X_f \times Q_f \times X_p$ *and* r_j^{n+1} *satisfying for any* $(v, \psi) \in X_f \times X_p$,

$$\begin{aligned} &\left(\frac{u_j^{n+1} - u_j^n}{\Delta t}, v \right)_f + \nu (\nabla u_j^{n+1}, \nabla v)_f + \sum_i \int_I \bar{\eta}_i (u_j^{n+1} \cdot \widehat{\tau}_i)(v \cdot \widehat{\tau}_i) \ ds - \left(p_j^{n+1}, \nabla \cdot v \right)_f \\ &\qquad + \sum_i \int_I (\eta_{i,j} - \bar{\eta}_i)(u_j^n \cdot \widehat{\tau}_i)(v \cdot \widehat{\tau}_i) \ ds + \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})} c_I(v, \phi_j^n) = (f_{f,j}^{n+1}, v)_f, \end{aligned} \tag{1.4}$$

$$\alpha (p_j^{n+1} - p_j^n) + \nabla \cdot u_j^{n+1} = 0, \ in \ D_f, \tag{1.5}$$

$$\begin{aligned} &g S_0 \left(\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t}, \psi \right)_p + g(\bar{\mathcal{K}} \nabla \phi_j^{n+1}, \nabla \psi)_p + g((\mathcal{K}_j - \bar{\mathcal{K}}) \nabla \phi_j^n, \nabla \psi)_p \\ &\qquad - \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})} c_I(u_j^n, \psi) = g(f_{p,j}^{n+1}, \psi)_p, \end{aligned} \tag{1.6}$$

$$\frac{r_j^{n+1}-r_j^n}{\Delta t}=-\frac{1}{T}r_j^{n+1}+\frac{1}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1},\phi_j^n)-c_I(u_j^n,\phi_j^{n+1})\right), \tag{1.7}$$

$$\textit{where}\qquad \bar{\mathcal{K}}=\frac{1}{J}\sum_{j=1}^{J}\mathcal{K}_j,\quad \eta_{i,j}=\frac{\alpha_{\mathrm{BJS}}}{\sqrt{\widehat{\tau}_i\cdot\mathcal{K}_j\widehat{\tau}_i}},\quad \textit{and}\quad \bar{\eta}_i=\frac{1}{J}\sum_{j=1}^{J}\eta_{i,j}.$$

ALGORITHM 1.2 (SAV-BDF2AC-En). *Find* $(u_j^{n+1},p_j^{n+1},\phi_j^{n+1})\in X_f\times Q_f\times X_p$ *and* r_j^{n+1} *satisfying for any* $(v,\psi)\in X_f\times X_p$,

$$\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},v\right)_f+\nu(\nabla u_j^{n+1},\nabla v)_f+\sum_i\int_I\bar{\eta}_i(u_j^{n+1}\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\,ds-\left(p_j^{n+1},\nabla\cdot v\right)_f \tag{1.8}$$
$$+\sum_i\int_I(\eta_{i,j}-\bar{\eta}_i)((2u_j^n-u_j^{n-1})\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\,ds+\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(v,2\phi_j^n-\phi_j^{n-1})=(f_{f,j}^{n+1},v)_f,$$
$$\alpha\Delta t(3p_j^{n+1}-4p_j^n+p_j^{n-1})+\nabla\cdot u_j^{n+1}=0,\ in\ D_f, \tag{1.9}$$
$$gS_0\left(\frac{3\phi_j^{n+1}-4\phi_j^n+\phi_j^{n-1}}{2\Delta t},\psi\right)_p+g(\bar{\mathcal{K}}\nabla\phi_j^{n+1},\nabla\psi)_p+g((\mathcal{K}_j-\bar{\mathcal{K}})\nabla(2\phi_j^n-\phi_j^{n-1}),\nabla\psi)_p \tag{1.10}$$
$$-\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(2u_j^n-u_j^{n-1},\psi)=g(f_{p,j}^{n+1},\psi)_p,$$
$$\frac{3r_j^{n+1}-4r_j^n+r_j^{n-1}}{2\Delta t}=-\frac{1}{T}r_j^{n+1}+\frac{1}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1},2\phi_j^n-\phi_j^{n-1})-c_I(2u_j^n-u_j^{n-1},\phi_j^{n+1})\right). \tag{1.11}$$

For Algorithm 1.2, one needs to use another numerical method to compute (u_j^1,p_j^1,ϕ_j^1) to start the algorithm. This can be done with any non-ensemble or ensemble one step method, such as the standard BE or Crank-Nicolson method, or the SAV-BEAC-En method presented in Algorithm 1.1.

The rest of this paper is structured as follows. In Section 2 we prove the long time stability of both algorithms without any timestep conditions. The fully decoupled, algorithmic implementation of the schemes and related linear algebra are discussed in Section 3. In Section 4 we use manufactured analytic solutions to demonstrate the convergence rates of the proposed schemes, and use several realistic flow problems to test the performance and demonstrate efficiency of the presented ensemble algorithms. Section 5 concludes the discussions.

2. Stability Analysis. We denote the $L^2(D_{f/p})$ norms by $\|\cdot\|_{f/p}$ and the corresponding inner products are denoted by $(\cdot,\cdot)_{f/p}$. Let $|\cdot|_2$ denote the 2-norm of either vectors or matrices. Let $k_{j,min}(x)$, $\bar{k}_{min}(x)$ be the minimum eigenvalue of the hydraulic conductivity tensor $\mathcal{K}_j(x)$, $\bar{\mathcal{K}}(x)$ respectively, and $\rho_j'(x)$ be the spectral radius of the fluctuation of hydraulic conductivity tensor $\mathcal{K}_j(x)-\bar{\mathcal{K}}(x)$. Since both $\mathcal{K}_j(x)$ and $\bar{\mathcal{K}}(x)$ are symmetric, $|\mathcal{K}_j(x)-\bar{\mathcal{K}}(x)|_2=\rho_j'(x)$. We then define the following quantities that will be used in our proof.

$$\eta_i'^{max}=\max_j\max_{x\in I}|\eta_{i,j}(x)-\bar{\eta}_i(x)|,\quad \bar{\eta}_i^{min}=\min_{x\in I}\bar{\eta}_i(x),\quad \bar{k}_{min}=\min_{x\in D_p}\bar{k}_{min}(x),\quad \rho'_{max}=\max_j\max_{x\in D_p}\rho_j'(x).$$

2.1. Stability of SAV-BEAC. We prove long time stability of Algorithm 1.1 under two parameter conditions, *without* any timestep conditions.

THEOREM 2.1 (Long time stability of Algorithm 1.1). *If the following two parameter conditions hold,*

$$\eta_i'^{max}\le\bar{\eta}_i^{min},\qquad \rho'_{max}<\bar{k}_{min}, \tag{2.1}$$

then Algorithm 1.1 is long time stable: for any $N\ge1$,

$$\frac{1}{2}\|u_j^N\|_f^2+\frac{\alpha}{2}\Delta t\|p_j^N\|^2+\Delta t\sum_{n=0}^{N-1}\frac{\alpha}{2}\|p_j^{n+1}-p_j^n\|^2+\frac{gS_0}{2}\|\phi_j^N\|_p^2+\Delta t\sum_i\frac{\bar{\eta}_i^{min}}{2}\int_I(u_j^N\cdot\widehat{\tau}_i)^2\,ds \tag{2.2}$$

$$
\begin{aligned}
&+\Delta t\frac{g\rho'_{max}}{2}\|\nabla\phi_j^N\|_p^2+\frac{1}{2}|r_j^N|^2+\frac{1}{2}\sum_{n=0}^{N-1}|r_j^{n+1}-r_j^n|^2+\frac{\Delta t}{T}\sum_{n=0}^{N-1}|r_j^{n+1}|^2\\
&\le\frac{1}{2}\|u_j^0\|_f^2+\frac{\alpha}{2}\Delta t\|p_j^0\|^2+\frac{gS_0}{2}\|\phi_j^0\|_p^2+\Delta t\sum_i\frac{\bar\eta_i^{min}}{2}\int_I(u_j^0\cdot\widehat{\tau}_i)^2\,ds+\Delta t\frac{g\rho'_{max}}{2}\|\nabla\phi_j^0\|_p^2\\
&\quad+\frac{1}{2}|r_j^0|^2+\Delta t\sum_{n=0}^{N-1}\frac{C_{P,f}^2}{2\nu}\|f_{f,j}^{n+1}\|_f^2+\Delta t\sum_{n=0}^{N-1}\frac{gC_{P,p}^2}{2(\bar k_{min}-\rho'_{max})}\|f_{p,j}^{n+1}\|_p^2.
\end{aligned}
$$

Proof. Setting $v=u_j^{n+1}$, $\psi=\phi_j^{n+1}$ in Algorithm 1.1, testing (1.5) by $q=p_j^{n+1}$, multiplying (1.7) by r_j^{n+1}, and adding all four equations yields

$$
\begin{aligned}
&\frac{1}{2\Delta t}\|u_j^{n+1}\|_f^2-\frac{1}{2\Delta t}\|u_j^n\|_f^2+\frac{1}{2\Delta t}\|u_j^{n+1}-u_j^n\|_f^2+\nu\|\nabla u_j^{n+1}\|_f^2+\sum_i\int_I\bar\eta_i(u_j^{n+1}\cdot\widehat{\tau}_i)(u_j^{n+1}\cdot\widehat{\tau}_i)\,ds \qquad (2.3)\\
&+\frac{\alpha}{2}\|p_j^{n+1}\|^2-\frac{\alpha}{2}\|p_j^n\|^2+\frac{\alpha}{2}\|p_j^{n+1}-p_j^n\|^2+\frac{gS_0}{2\Delta t}\|\phi_j^{n+1}\|_p^2-\frac{gS_0}{2\Delta t}\|\phi_j^n\|_p^2+\frac{gS_0}{2\Delta t}\|\phi_j^{n+1}-\phi_j^n\|_p^2\\
&+g(\bar{\mathcal{K}}\nabla\phi_j^{n+1},\nabla\phi_j^{n+1})_p+\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(u_j^{n+1},\phi_j^n)-\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(u_j^n,\phi_j^{n+1})\\
&+\frac{1}{2\Delta t}|r_j^{n+1}|^2-\frac{1}{2\Delta t}|r_j^n|^2+\frac{1}{2\Delta t}|r_j^{n+1}-r_j^n|^2+\frac{1}{T}|r_j^{n+1}|^2-\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1},\phi_j^n)-c_I(u_j^n,\phi_j^{n+1})\right)\\
&=(f_{f,j}^{n+1},u_j^{n+1})_f+g(f_{p,j}^{n+1},\phi_j^{n+1})_p-\sum_i\int_I(\eta_{i,j}-\bar\eta_i)(u_j^n\cdot\widehat{\tau}_i)(u_j^{n+1}\cdot\widehat{\tau}_i)\,ds-g((\mathcal{K}_j-\bar{\mathcal{K}})\nabla\phi_j^n,\nabla\phi_j^{n+1})_p.
\end{aligned}
$$

Applying Cauchy-Schwarz and Young's inequalities to the source terms, for any $\beta>0$ we have

$$
\begin{aligned}
&(f_{f,j}^{n+1},u_j^{n+1})_f+g(f_{p,j}^{n+1},\phi_j^{n+1})_p \qquad (2.4)\\
&\qquad\le\frac{C_{P,f}^2}{2\nu}\|f_{f,j}^{n+1}\|_f^2+\frac{1}{2}\nu\|\nabla u_j^{n+1}\|_f^2+\frac{gC_{P,p}^2}{4\beta\bar k_{min}}\|f_{p,j}^{n+1}\|_p^2+\beta g\bar k_{min}\|\nabla\phi_j^{n+1}\|_p^2.
\end{aligned}
$$

For the other terms on the right hand side of (2.3) we have the following bounds.

$$
\begin{aligned}
&-\sum_i\int_I(\eta_{i,j}-\bar\eta_i)(u_j^n\cdot\widehat{\tau}_i)(u_j^{n+1}\cdot\widehat{\tau}_i)\,ds\le\sum_i\left[\frac{\eta_i'^{max}}{2}\int_I(u_j^n\cdot\widehat{\tau}_i)^2\,ds+\frac{\eta_i'^{max}}{2}\int_I(u_j^{n+1}\cdot\widehat{\tau}_i)^2\,ds\right],\\
&-g\left((\mathcal{K}_j-\bar{\mathcal{K}})\nabla\phi_j^n,\nabla\phi_j^{n+1}\right)_p\le\frac{g\rho'_{max}}{2}\|\nabla\phi_j^n\|_p^2+\frac{g\rho'_{max}}{2}\|\nabla\phi_j^{n+1}\|_p^2.
\end{aligned}
$$

Using above estimates, equation (2.3) becomes

$$
\begin{aligned}
&\frac{1}{2\Delta t}\|u_j^{n+1}\|_f^2-\frac{1}{2\Delta t}\|u_j^n\|_f^2+\frac{1}{2}\nu\|\nabla u_j^{n+1}\|_f^2+\sum_i\left[\frac{\bar\eta_i^{min}}{2}-\frac{\eta_i'^{max}}{2}\right]\int_I(u_j^{n+1}\cdot\widehat{\tau}_i)^2\,ds \qquad (2.5)\\
&+\sum_i\frac{\bar\eta_i^{min}}{2}\left[\int_I(u_j^{n+1}\cdot\widehat{\tau}_i)^2\,ds-\int_I(u_j^n\cdot\widehat{\tau}_i)^2\,ds\right]+\sum_i\left[\frac{\bar\eta_i^{min}}{2}-\frac{\eta_i'^{max}}{2}\right]\int_I(u_j^n\cdot\widehat{\tau}_i)^2\,ds\\
&+\frac{\alpha}{2}\|p_j^{n+1}\|^2-\frac{\alpha}{2}\|p_j^n\|^2+\frac{\alpha}{2}\|p_j^{n+1}-p_j^n\|^2+\frac{gS_0}{2\Delta t}\|\phi_j^{n+1}\|_p^2-\frac{gS_0}{2\Delta t}\|\phi_j^n\|_p^2\\
&+(1-\beta-\frac{\rho'_{max}}{\bar k_{min}})g\bar k_{min}\|\nabla\phi_j^{n+1}\|_p^2+\frac{g\rho'_{max}}{2}\left(\|\nabla\phi_j^{n+1}\|_p^2-\|\nabla\phi_j^n\|_p^2\right)\\
&+\frac{1}{2\Delta t}|r_j^{n+1}|^2-\frac{1}{2\Delta t}|r_j^n|^2+\frac{1}{2\Delta t}|r_j^{n+1}-r_j^n|^2+\frac{1}{T}|r_j^{n+1}|^2\le\frac{C_{P,f}^2}{2\nu}\|f_{f,j}^{n+1}\|_f^2+\frac{gC_{P,p}^2}{4\beta\bar k_{min}}\|f_{p,j}^{n+1}\|_p^2.
\end{aligned}
$$

Assuming both of the two parameter conditions in (2.1) are satisfied, and taking $\beta = \frac{1}{2}(1 - \frac{\rho'_{max}}{\bar{k}_{min}})$, inequality (2.5) reduces to

$$\frac{1}{2\Delta t}\|u_j^{n+1}\|_f^2 - \frac{1}{2\Delta t}\|u_j^n\|_f^2 + \sum_i \frac{\bar{\eta}_i^{min}}{2}\left[\int_I (u_j^{n+1}\cdot\widehat{\tau}_i)^2\,ds - \int_I (u_j^n\cdot\widehat{\tau}_i)^2\,ds\right] + \frac{\alpha}{2}\|p_j^{n+1}\|^2 - \frac{\alpha}{2}\|p_j^n\|^2 \tag{2.6}$$
$$+ \frac{\alpha}{2}\|p_j^{n+1}-p_j^n\|^2 + \frac{gS_0}{2\Delta t}\|\phi_j^{n+1}\|_p^2 - \frac{gS_0}{2\Delta t}\|\phi_j^n\|_p^2 + \frac{g\rho'_{max}}{2}\left(\|\nabla\phi_j^{n+1}\|_p^2 - \|\nabla\phi_j^n\|_p^2\right)$$
$$+ \frac{1}{2\Delta t}|r_j^{n+1}|^2 - \frac{1}{2\Delta t}|r_j^n|^2 + \frac{1}{2\Delta t}|r_j^{n+1}-r_j^n|^2 + \frac{1}{T}|r_j^{n+1}|^2 \le \frac{C_{P,f}^2}{2\nu}\|f_{f,j}^{n+1}\|_f^2 + \frac{gC_{P,p}^2}{2(\bar{k}_{min}-\rho'_{max})}\|f_{p,j}^{n+1}\|_p^2.$$

Summing (2.6) from $n = 0$ to $N-1$ and multiplying through by Δt we get (2.2).
□

2.2. Stability of SAV-BDF2. We prove long time stability of Algorithm 1.2 under two parameter conditions, *without* any timestep conditions.

$$\eta_i'^{max} \le \frac{\bar{\eta}_i^{min}}{3}, \qquad \rho'_{max} < \frac{\bar{k}_{min}}{3}. \tag{2.7}$$

THEOREM 2.2 (Long time stability of Algorithm 1.2). *If the two parameter conditions in* (2.7) *hold, then the Algorithm 1.2 is long time stable: for any* $N \ge 2$,

$$\|u_j^N\|_f^2 + \|2u_j^N - u_j^{N-1}\|_f^2 + \sum_{n=1}^{N-1}\|u_j^{n+1} - 2u_j^n + u_j^{n-1}\|_f^2 + \Delta t\sum_{n=1}^{N-1} 2\nu\|\nabla u_j^{n+1}\|_f^2 \tag{2.8}$$
$$+ \Delta t\sum_i 2\bar{\eta}_i^{min}\int_I (u_j^N\cdot\widehat{\tau}_i)^2\,ds + \Delta t\sum_i \frac{2\bar{\eta}_i^{min}}{3}\int_I (u_j^{N-1}\cdot\widehat{\tau}_i)^2\,ds + 2\alpha\Delta t^2\|p_j^N\|_f^2 + 2\alpha\Delta t^2\|2p_j^N - p_j^{N-1}\|_f^2$$
$$+ 2\alpha\Delta t^2\sum_{n=1}^{N-1}\|p_j^{n+1} - 2p_j^n + p_j^{n-1}\|_f^2 + gS_0\|\phi_j^N\|_p^2 + gS_0\|2\phi_j^N - \phi_j^{N-1}\|_p^2$$
$$+ gS_0\sum_{n=1}^{N-1}\|\phi_j^{n+1} - 2\phi_j^n + \phi_j^{n-1}\|_p^2 + 6g\rho'_{max}\Delta t\|\nabla\phi_j^N\|_p^2 + 2g\rho'_{max}\Delta t\|\nabla\phi_j^{N-1}\|_p^2 + |r_j^N|^2$$
$$+ |2r_j^N - r_j^{N-1}|^2 + \sum_{n=1}^{N-1}|r_j^{n+1} - 2r_j^n + r_j^{n-1}|^2 + \frac{4\Delta t}{T}\sum_{n=1}^{N-1}|r_j^{n+1}|^2$$
$$\le \|u_j^1\|_f^2 + \|2u_j^1 - u_j^0\|_f^2 + \Delta t\sum_i 2\bar{\eta}_i^{min}\int_I (u_j^1\cdot\widehat{\tau}_i)^2\,ds + \Delta t\sum_i \frac{2\bar{\eta}_i^{min}}{3}\int_I (u_j^0\cdot\widehat{\tau}_i)^2\,ds$$
$$+ 2\alpha\Delta t^2\|p_j^1\|_f^2 + 2\alpha\Delta t^2\|2p_j^1 - p_j^0\|_f^2 + gS_0\|\phi_j^1\|_p^2 + gS_0\|2\phi_j^1 - \phi_j^0\|_p^2 + 6g\rho'_{max}\Delta t\|\nabla\phi_j^1\|_p^2$$
$$+ 2g\rho'_{max}\Delta t\|\nabla\phi_j^0\|_p^2 + |r_j^1|^2 + |2r_j^1 - r_j^0|^2 + \Delta t\sum_{n=1}^{N-1}\frac{2C_{P,f}^2}{\nu}\|f_{f,j}^{n+1}\|_f^2 + \Delta t\sum_{n=1}^{N-1}\frac{2gC_{P,p}^2}{\bar{k}_{min} - 3\rho'_{max}}\|f_{p,j}^{n+1}\|_p^2.$$

Proof. Setting $v = u_j^{n+1}$, $\psi = \phi_j^{n+1}$ in Algorithm 1.2, testing (1.9) by $q = p_j^{n+1}$, multiplying (1.11) by r_j^{n+1}, adding all four equations and using the estimate (2.4) yields

$$\frac{1}{4\Delta t}\|u_j^{n+1}\|_f^2 + \frac{1}{4\Delta t}\|2u_j^{n+1} - u_j^n\|_f^2 - \frac{1}{4\Delta t}\|u_j^n\|_f^2 - \frac{1}{4\Delta t}\|2u_j^n - u_j^{n-1}\|_f^2 \tag{2.9}$$
$$+ \frac{1}{4\Delta t}\|u_j^{n+1} - 2u_j^n + u_j^{n-1}\|_f^2 + \nu\|\nabla u_j^{n+1}\|_f^2 + \sum_i\int_I \bar{\eta}_i(u_j^{n+1}\cdot\widehat{\tau}_i)(u_j^{n+1}\cdot\widehat{\tau}_i)\,ds + \frac{\alpha\Delta t}{2}\|p_j^{n+1}\|_f^2$$
$$+ \frac{\alpha\Delta t}{2}\|2p_j^{n+1} - p_j^n\|_f^2 - \frac{\alpha\Delta t}{2}\|p_j^n\|_f^2 - \frac{\alpha\Delta t}{2}\|2p_j^n - p_j^{n-1}\|_f^2 + \frac{\alpha\Delta t}{2}\|p_j^{n+1} - 2p_j^n + p_j^{n-1}\|_f^2$$

$$
\begin{aligned}
&+\frac{gS_0}{4\Delta t}\|\phi_j^{n+1}\|_p^2+\frac{gS_0}{4\Delta t}\|2\phi_j^{n+1}-\phi_j^n\|_p^2-\frac{gS_0}{4\Delta t}\|\phi_j^n\|_p^2-\frac{gS_0}{4\Delta t}\|2\phi_j^n-\phi_j^{n-1}\|_p^2\\
&+\frac{gS_0}{4\Delta t}\|\phi_j^{n+1}-2\phi_j^n+\phi_j^{n-1}\|_p^2+g(\bar{\mathcal{K}}\nabla\phi_j^{n+1},\nabla\phi_j^{n+1})_p+\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(u_j^{n+1},2\phi_j^n-\phi_j^{n-1})\\
&-\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(2u_j^n-u_j^{n-1},\phi_j^{n+1})+\frac{1}{4\Delta t}|r_j^{n+1}|^2+\frac{1}{4\Delta t}|2r_j^{n+1}-r_j^n|^2\\
&-\frac{1}{4\Delta t}|r_j^n|^2-\frac{1}{4\Delta t}|2r_j^n-r_j^{n-1}|^2+\frac{1}{4\Delta t}|r_j^{n+1}-2r_j^n+r_j^{n-1}|^2+\frac{1}{T}|r_j^{n+1}|^2\\
&-\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1},2\phi_j^n-\phi_j^{n-1})-c_I(2u_j^n-u_j^{n-1},\phi_j^{n+1})\right)\\
&\le\frac{C_{P,f}^2}{2\nu}\|f_{f,j}^{n+1}\|_f^2+\frac{1}{2}\nu\|\nabla u_j^{n+1}\|_f^2+\frac{gC_{P,p}^2}{4\beta\bar{k}_{min}}\|f_{p,j}^{n+1}\|_p^2+\beta g\bar{k}_{min}\|\nabla\phi_j^{n+1}\|_p^2\\
&\quad-\sum_i\int_I(\eta_{i,j}-\bar{\eta}_i)((2u_j^n-u_j^{n-1})\cdot\widehat{\tau}_i)(u_j^{n+1}\cdot\widehat{\tau}_i)\,ds-g((\mathcal{K}_j-\bar{\mathcal{K}})\nabla(2\phi_j^n-\phi_j^{n-1}),\nabla\phi_j^{n+1})_p.
\end{aligned}
$$

Using the inequality $(2a-b)^2\le 6a^2+3b^2$, we have

$$
\begin{aligned}
&-\sum_i\int_I(\eta_{i,j}-\bar{\eta}_i)((2u_j^n-u_j^{n-1})\cdot\widehat{\tau}_i)(u_j^{n+1}\cdot\widehat{\tau}_i)\,ds \qquad (2.10)\\
&\qquad\le\sum_i\left[\eta_i'^{max}\int_I(u_j^n\cdot\widehat{\tau}_i)^2\,ds+\frac{\eta_i'^{max}}{2}\int_I(u_j^{n-1}\cdot\widehat{\tau}_i)^2\,ds+\frac{3\eta_i'^{max}}{2}\int_I(u_j^{n+1}\cdot\widehat{\tau}_i)^2\,ds\right]
\end{aligned}
$$

and

$$
-g\left((\mathcal{K}_j-\bar{\mathcal{K}})\nabla(2\phi_j^n-\phi_j^{n-1}),\nabla\phi_j^{n+1}\right)_p\le g\rho'_{max}\|\nabla\phi_j^n\|_p^2+\frac{g\rho'_{max}}{2}\|\nabla\phi_j^{n-1}\|_p^2+\frac{3g\rho'_{max}}{2}\|\nabla\phi_j^{n+1}\|_p^2. \quad (2.11)
$$

Using above estimates, equation (2.9) becomes

$$
\begin{aligned}
&\frac{1}{4\Delta t}\|u_j^{n+1}\|_f^2+\frac{1}{4\Delta t}\|2u_j^{n+1}-u_j^n\|_f^2-\frac{1}{4\Delta t}\|u_j^n\|_f^2-\frac{1}{4\Delta t}\|2u_j^n-u_j^{n-1}\|_f^2 \qquad (2.12)\\
&+\frac{1}{4\Delta t}\|u_j^{n+1}-2u_j^n+u_j^{n-1}\|_f^2+\frac{1}{2}\nu\|\nabla u_j^{n+1}\|_f^2+\sum_i\big[\frac{\bar{\eta}_i^{min}}{2}-\frac{3\eta_i'^{max}}{2}\big]\int_I(u_j^{n+1}\cdot\widehat{\tau}_i)^2\,ds\\
&+\sum_i\frac{\bar{\eta}_i^{min}}{2}\left[\int_I(u_j^{n+1}\cdot\widehat{\tau}_i)^2\,ds-\int_I(u_j^n\cdot\widehat{\tau}_i)^2\,ds\right]+\sum_i\big[\frac{\bar{\eta}_i^{min}}{3}-\eta_i'^{max}\big]\int_I(u_j^n\cdot\widehat{\tau}_i)^2\,ds\\
&+\sum_i\frac{\bar{\eta}_i^{min}}{6}\left[\int_I(u_j^n\cdot\widehat{\tau}_i)^2\,ds-\int_I(u_j^{n-1}\cdot\widehat{\tau}_i)^2\,ds\right]+\sum_i\big[\frac{\bar{\eta}_i^{min}}{6}-\frac{\eta_i'^{max}}{2}\big]\int_I(u_j^{n-1}\cdot\widehat{\tau}_i)^2\,ds\\
&+\frac{\alpha\Delta t}{2}\|p_j^{n+1}\|_f^2+\frac{\alpha\Delta t}{2}\|2p_j^{n+1}-p_j^n\|_f^2-\frac{\alpha\Delta t}{2}\|p_j^n\|_f^2-\frac{\alpha\Delta t}{2}\|2p_j^n-p_j^{n-1}\|_f^2+\frac{\alpha\Delta t}{2}\|p_j^{n+1}-2p_j^n+p_j^{n-1}\|_f^2\\
&+\frac{gS_0}{4\Delta t}\|\phi_j^{n+1}\|_p^2+\frac{gS_0}{4\Delta t}\|2\phi_j^{n+1}-\phi_j^n\|_p^2-\frac{gS_0}{4\Delta t}\|\phi_j^n\|_p^2-\frac{gS_0}{4\Delta t}\|2\phi_j^n-\phi_j^{n-1}\|_p^2\\
&+\frac{gS_0}{4\Delta t}\|\phi_j^{n+1}-2\phi_j^n+\phi_j^{n-1}\|_p^2+\big(1-\beta-\frac{3\rho'_{max}}{\bar{k}_{min}}\big)g\bar{k}_{min}\|\nabla\phi_j^{n+1}\|_p^2+\frac{3g\rho'_{max}}{2}\big(\|\nabla\phi_j^{n+1}\|_p^2-\|\nabla\phi_j^n\|_p^2\big)\\
&+\frac{g\rho'_{max}}{2}\big(\|\nabla\phi_j^n\|_p^2-\|\nabla\phi_j^{n-1}\|_p^2\big)+\frac{1}{4\Delta t}|r_j^{n+1}|^2+\frac{1}{4\Delta t}|2r_j^{n+1}-r_j^n|^2\\
&-\frac{1}{4\Delta t}|r_j^n|^2-\frac{1}{4\Delta t}|2r_j^n-r_j^{n-1}|^2+\frac{1}{4\Delta t}|r_j^{n+1}-2r_j^n+r_j^{n-1}|^2+\frac{1}{T}|r_j^{n+1}|^2\\
&\le\frac{C_{P,f}^2}{2\nu}\|f_{f,j}^{n+1}\|_f^2+\frac{gC_{P,p}^2}{4\beta\bar{k}_{min}}\|f_{p,j}^{n+1}\|_p^2.
\end{aligned}
$$

Now if the two parameter conditions in (2.7) both hold, and taking $\beta = \frac{1}{2}(1 - \frac{3\rho'_{max}}{\bar{k}_{min}})$, inequality (2.12) reduces to

$$
\begin{aligned}
&\frac{1}{4\Delta t}\|u_j^{n+1}\|_f^2 + \frac{1}{4\Delta t}\|2u_j^{n+1} - u_j^n\|_f^2 - \frac{1}{4\Delta t}\|u_j^n\|_f^2 - \frac{1}{4\Delta t}\|2u_j^n - u_j^{n-1}\|_f^2 \qquad (2.13)\\
&+ \frac{1}{4\Delta t}\|u_j^{n+1} - 2u_j^n + u_j^{n-1}\|_f^2 + \frac{1}{2}\nu\|\nabla u_j^{n+1}\|_f^2 + \sum_i \frac{\bar{\eta}_i^{min}}{2}\left[\int_I (u_j^{n+1}\cdot\widehat{\tau}_i)^2\, ds - \int_I (u_j^n\cdot\widehat{\tau}_i)^2\, ds\right]\\
&+ \sum_i \frac{\bar{\eta}_i^{min}}{6}\left[\int_I (u_j^n\cdot\widehat{\tau}_i)^2\, ds - \int_I (u_j^{n-1}\cdot\widehat{\tau}_i)^2\, ds\right] + \frac{\alpha\Delta t}{2}\|p_j^{n+1}\|_f^2 + \frac{\alpha\Delta t}{2}\|2p_j^{n+1} - p_j^n\|_f^2 - \frac{\alpha\Delta t}{2}\|p_j^n\|_f^2\\
&- \frac{\alpha\Delta t}{2}\|2p_j^n - p_j^{n-1}\|_f^2 + \frac{\alpha\Delta t}{2}\|p_j^{n+1} - 2p_j^n + p_j^{n-1}\|_f^2 + \frac{gS_0}{4\Delta t}\|\phi_j^{n+1}\|_p^2 + \frac{gS_0}{4\Delta t}\|2\phi_j^{n+1} - \phi_j^n\|_p^2\\
&- \frac{gS_0}{4\Delta t}\|\phi_j^n\|_p^2 - \frac{gS_0}{4\Delta t}\|2\phi_j^n - \phi_j^{n-1}\|_p^2 + \frac{gS_0}{4\Delta t}\|\phi_j^{n+1} - 2\phi_j^n + \phi_j^{n-1}\|_p^2\\
&+ \frac{3g\rho'_{max}}{2}\left(\|\nabla\phi_j^{n+1}\|_p^2 - \|\nabla\phi_j^n\|_p^2\right) + \frac{g\rho'_{max}}{2}\left(\|\nabla\phi_j^n\|_p^2 - \|\nabla\phi_j^{n-1}\|_p^2\right) + \frac{1}{4\Delta t}|r_j^{n+1}|^2\\
&+ \frac{1}{4\Delta t}|2r_j^{n+1} - r_j^n|^2 - \frac{1}{4\Delta t}|r_j^n|^2 - \frac{1}{4\Delta t}|2r_j^n - r_j^{n-1}|^2 + \frac{1}{4\Delta t}|r_j^{n+1} - 2r_j^n + r_j^{n-1}|^2 + \frac{1}{T}|r_j^{n+1}|^2\\
&\le \frac{C_{P,f}^2}{2\nu}\|f_{f,j}^{n+1}\|_f^2 + \frac{gC_{P,p}^2}{2(\bar{k}_{min} - 3\rho'_{max})}\|f_{p,j}^{n+1}\|_p^2.
\end{aligned}
$$

Summing (2.13) from $n = 1$ to $N - 1$ and multiplying through by $4\Delta t$ yields (2.8).
□

REMARK 2.3. *For the first order method (SAV-BEAC-En), the parameter conditions in* (2.1) *indicate that the fluctuation needs to be smaller than the mean. Usually they are easy to fulfill in UQ applications in which the magnitude of the fluctuation is generally much smaller than the magnitude of the uncertain parameter. The parameter conditions in* (2.7) *for the second order method (SAV-BDF2AC-En) are a bit stricter than the ones for the first order method, but can still be easily satisfied in many practical UQ problems. If they are not satisfied for a large ensemble, one can split the large ensemble into smaller ones to make these conditions satisfied, and apply the ensemble algorithm to each smaller ensemble.*

3. Solution Algorithms and Numerical Implementation. The proposed algorithms SAV-BEAC-En and SAV-BDF2-En have the scalar variables r_j coupled with the primitive variables, which needs to be decoupled for fast computation of the intended systems. In this section, we present corresponding fully decoupled solution algorithms that are equivalent to Algorithm 1.1 and Algorithm 1.2 respectively. Detailed numerical implementation and related linear algebra will also be discussed.

3.1. Solution Algorithms. Let

$$
S_j^{n+1} = \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}, \qquad u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}, \qquad \phi_j^{n+1} = \hat{\phi}_j^{n+1} + S_j^{n+1}\breve{\phi}_j^{n+1} \qquad (3.1)
$$

3.1.1. SAV-BEAC-En. Instead of solving (1.4)-(1.7), we solve the following four subproblems for $\hat{u}_j^{n+1}$, $\hat{\phi}_j^{n+1}$, $\breve{u}_j^{n+1}$,$\breve{\phi}_j^{n+1}$ respectively.

(BE sub-problem 1): Find $\hat{u}_j^{n+1} \in X_f$ satisfying $\forall\, v \in X_f$,

$$
\begin{cases}
\frac{1}{\Delta t}\left(\hat{u}_j^{n+1}, v\right)_f + \nu(\nabla\hat{u}_j^{n+1}, \nabla v)_f + \sum_i \int_I \bar{\eta}_i(\hat{u}_j^{n+1}\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\, ds + \frac{1}{\alpha}\left(\nabla\cdot\hat{u}_j^{n+1}, \nabla\cdot v\right)_f\\
\qquad = (f_{f,j}^{n+1}, v)_f + \frac{1}{\Delta t}\left(u_j^n, v\right)_f - \sum_i \int_I (\eta_{i,j} - \bar{\eta}_i)(u_j^n\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\, ds + \left(p_j^n, \nabla\cdot v\right)_f, \quad \text{in } D_f,\\
\hat{u}_j^{n+1} = a_j^{n+1}, \quad \text{on } \partial D_f\backslash I.
\end{cases}
$$

(BE sub-problem 2): Find $\hat{\phi}_j^{n+1} \in X_p$ satisfying $\forall\ \psi \in X_p$,

$$\begin{cases} \dfrac{gS_0}{\Delta t}\left(\hat{\phi}_j^{n+1}, \psi\right)_p + g(\bar{\mathcal{K}}\nabla\hat{\phi}_j^{n+1}, \nabla\psi)_p \\ \qquad = g(f_{p,j}^{n+1}, \psi)_p + \dfrac{gS_0}{\Delta t}\left(\phi_j^n, \psi\right)_p - g((\mathcal{K}_j - \bar{\mathcal{K}})\nabla\phi_j^n, \nabla\psi)_p, \quad \text{in } D_p, \\ \hat{\phi}_j^{n+1} = b_j^{n+1}, \quad \text{on } \partial D_p \backslash I. \end{cases}$$

(BE sub-problem 3): Find $\breve{u}_j^{n+1} \in X_f$ satisfying $\forall\, v \in X_f$,

$$\begin{cases} \dfrac{1}{\Delta t}\left(\breve{u}_j^{n+1}, v\right)_f + \nu(\nabla\breve{u}_j^{n+1}, \nabla v)_f + \displaystyle\sum_i \int_I \bar{\eta}_i(\breve{u}_j^{n+1}\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\, ds + \frac{1}{\alpha}\left(\nabla\cdot\breve{u}_j^{n+1}, \nabla\cdot v\right)_f \\ \qquad = -c_I(v, \phi_j^n), \quad \text{in } D_f, \\ \breve{u}_j^{n+1} = 0, \quad \text{on } \partial D_f \backslash I. \end{cases}$$

(BE sub-problem 4): Find $\breve{\phi}_j^{n+1} \in X_p$ satisfying $\forall\ \psi \in X_p$,

$$\begin{cases} \dfrac{gS_0}{\Delta t}\left(\breve{\phi}_j^{n+1}, \psi\right)_p + g(\bar{\mathcal{K}}\nabla\breve{\phi}_j^{n+1}, \nabla\psi)_p = c_I(u_j^n, \psi), \quad \text{in } D_p, \\ \breve{\phi}_j^{n+1} = 0, \quad \text{on } \partial D_p \backslash I. \end{cases}$$

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

$$S_j^{n+1} = \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})} \qquad \Longrightarrow \qquad r_j^{n+1} = exp(-\frac{t^{n+1}}{T})\, S_j^{n+1}. \tag{3.2}$$

Multiplying (1.7) by r_j^{n+1} gives

$$\frac{r_j^{n+1} - r_j^n}{\Delta t}\cdot r_j^{n+1} + \frac{1}{T}|r_j^{n+1}|^2 - \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1}, \phi_j^n) - c_I(u_j^n, \phi_j^{n+1})\right) = 0. \tag{3.3}$$

Plugging (3.2) into (3.3) gives

$$\begin{aligned} &(\frac{1}{\Delta t} + \frac{1}{T})(r_j^{n+1})^2 - \frac{1}{\Delta t} r_j^n r_j^{n+1} - S_j^{n+1}\left(c_I(u_j^{n+1}, \phi_j^n) - c_I(u_j^n, \phi_j^{n+1})\right) = 0 \\ \Longrightarrow\ &(\frac{1}{\Delta t} + \frac{1}{T})exp(-\frac{2t^{n+1}}{T})\,(S_j^{n+1})^2 - \frac{1}{\Delta t} r_j^n exp(-\frac{t^{n+1}}{T})\, S_j^{n+1} \\ &\qquad - S_j^{n+1}\left(c_I(\hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}, \phi_j^n) - c_I(u_j^n, \hat{\phi}_j^{n+1} + S_j^{n+1}\breve{\phi}_j^{n+1})\right) = 0 \end{aligned}$$

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

$$S_j^{n+1}\left(A_j^{n+1} S_j^{n+1} + B_j^{n+1}\right) = 0 \qquad \Longrightarrow \qquad S_j^{n+1} = -\frac{B_j^{n+1}}{A_j^{n+1}}. \tag{3.4}$$

where

$$\begin{aligned} A_j^{n+1} &= (\frac{1}{\Delta t} + \frac{1}{T})exp(-\frac{2t^{n+1}}{T}) - c_I(\breve{u}_j^{n+1}, \phi_j^n) + c_I(u_j^n, \breve{\phi}_j^{n+1}) \\ B_j^{n+1} &= -\frac{1}{\Delta t} r_j^n exp(-\frac{t^{n+1}}{T}) - c_I(\hat{u}_j^{n+1}, \phi_j^n) + c_I(u_j^n, \hat{\phi}_j^{n+1}). \end{aligned}$$

After getting $\hat{u}_j^{n+1}, \breve{u}_j^{n+1}, \hat{\phi}_j^{n+1}, \breve{\phi}_j^{n+1}, S_j^{n+1}$ can be computed directly using formula (3.4), and then we have

$$u_j^{n+1} = \hat{u}_j^{n+1} + S_j^{n+1}\breve{u}_j^{n+1}, \qquad \phi_j^{n+1} = \hat{\phi}_j^{n+1} + S_j^{n+1}\breve{\phi}_j^{n+1}, \qquad p_j^{n+1} = p_j^n - \frac{1}{\alpha}\nabla\cdot u_j^{n+1}. \tag{3.5}$$

3.1.2. SAV-BDF2AC-En. Instead of solving (1.8)-(1.11), we solve the following four subproblems for $\hat{u}_j^{n+1}$, $\hat{\phi}_j^{n+1}$, $\breve{u}_j^{n+1}$, $\breve{\phi}_j^{n+1}$ respectively.

(BDF2 sub-problem 1): Find $\hat{u}_j^{n+1} \in X_f$ satisfying $\forall\, v \in X_f$,

$$\begin{cases} \dfrac{3}{2\Delta t}\left(\hat{u}_j^{n+1}, v\right)_f + \nu(\nabla \hat{u}_j^{n+1}, \nabla v)_f + \displaystyle\sum_i \int_I \bar{\eta}_i (\hat{u}_j^{n+1}\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\, ds + \frac{1}{3\alpha\Delta t}\left(\nabla\cdot\hat{u}_j^{n+1}, \nabla\cdot v\right)_f \\ \qquad = (f_{f,j}^{n+1}, v)_f + \dfrac{2}{\Delta t}\left(u_j^n, v\right)_f - \dfrac{1}{2\Delta t}\left(u_j^{n-1}, v\right)_f \\ \qquad\quad - \displaystyle\sum_i \int_I (\eta_{i,j} - \bar{\eta}_i)((2u_j^n - u_j^{n-1})\cdot\widehat{\tau}_i)(v_h\cdot\widehat{\tau}_i)\, ds + \left(\frac{4}{3}p_j^n - \frac{1}{3}p_j^{n-1}, \nabla\cdot v\right)_f, \quad \text{in } D_f, \\ \hat{u}_j^{n+1} = a_j^{n+1}, \quad \text{on } \partial D_f \backslash I. \end{cases}$$

(BDF2 sub-problem 2): Find $\hat{\phi}_j^{n+1} \in X_p$ satisfying $\forall\ \psi \in X_p$,

$$\begin{cases} \dfrac{3gS_0}{2\Delta t}\left(\hat{\phi}_j^{n+1}, \psi\right)_p + g(\bar{\mathcal{K}}\nabla\hat{\phi}_j^{n+1}, \nabla\psi)_p \\ \qquad = g(f_{p,j}^{n+1}, \psi)_p + \dfrac{2gS_0}{\Delta t}\left(\phi_j^n, \psi\right)_p - \dfrac{gS_0}{2\Delta t}\left(\phi_j^{n-1}, \psi\right)_p - g((\mathcal{K}_j - \bar{\mathcal{K}})\nabla(2\phi_j^n - \phi_j^{n-1}, \nabla\psi)_p, \quad \text{in } D_p, \\ \hat{\phi}_j^{n+1} = b_j^{n+1}, \quad \text{on } \partial D_p \backslash I. \end{cases}$$

(BDF2 sub-problem 3): Find $\breve{u}_j^{n+1} \in X_f$ satisfying $\forall\, v \in X_f$,

$$\begin{cases} \dfrac{3}{2\Delta t}\left(\breve{u}_j^{n+1}, v\right)_f + \nu(\nabla \breve{u}_j^{n+1}, \nabla v)_f + \displaystyle\sum_i \int_I \bar{\eta}_i (\breve{u}_j^{n+1}\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\, ds + \frac{1}{3\alpha\Delta t}\left(\nabla\cdot\breve{u}_j^{n+1}, \nabla\cdot v\right)_f \\ \qquad = -c_I(v, 2\phi_j^n - \phi_j^{n-1}), \quad \text{in } D_f, \\ \breve{u}_j^{n+1} = 0, \quad \text{on } \partial D_f \backslash I. \end{cases}$$

(BDF2 sub-problem 4): Find $\breve{\phi}_j^{n+1} \in X_p$ satisfying $\forall\ \psi \in X_p$,

$$\begin{cases} \dfrac{3gS_0}{2\Delta t}\left(\breve{\phi}_j^{n+1}, \psi\right)_p + g(\bar{\mathcal{K}}\nabla\breve{\phi}_j^{n+1}, \nabla\psi)_p = c_I(2u_j^n - u_j^{n-1}, \psi), \quad \text{in } D_p, \\ \breve{\phi}_j^{n+1} = 0, \quad \text{on } \partial D_p \backslash I. \end{cases}$$

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

$$S_j^{n+1} = \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})} \qquad \Longrightarrow \qquad r_j^{n+1} = exp(-\frac{t^{n+1}}{T})\, S_j^{n+1}. \tag{3.6}$$

Multiplying (1.11) by r_j^{n+1} gives

$$\begin{aligned} &\frac{3r_j^{n+1} - 4r_j^n + r_j^{n-1}}{2\Delta t}\cdot r_j^{n+1} + \frac{1}{T}|r_j^{n+1}|^2 \\ &\qquad - \frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1}, 2\phi_j^n - \phi_j^{n-1}) - c_I(2u_j^n - u_j^{n-1}, \phi_j^{n+1})\right) = 0. \end{aligned} \tag{3.7}$$

Plugging (3.6) into (3.7) gives

$$\begin{aligned} &(\frac{3}{2\Delta t} + \frac{1}{T})(r_j^{n+1})^2 - \frac{2}{\Delta t} r_j^n r_j^{n+1} + \frac{1}{2\Delta t} r_j^{n-1} r_j^{n+1} \\ &\qquad - S_j^{n+1}\left(c_I(u_j^{n+1}, 2\phi_j^n - \phi_j^{n-1}) - c_I(2u_j^n - u_j^{n-1}, \phi_j^{n+1})\right) = 0 \\ \Longrightarrow\ &(\frac{3}{2\Delta t} + \frac{1}{T})exp(-\frac{2t^{n+1}}{T})\,(S_j^{n+1})^2 - \frac{2}{\Delta t} r_j^n exp(-\frac{t^{n+1}}{T})\, S_j^{n+1} + \frac{1}{2\Delta t} r_j^{n-1} exp(-\frac{t^{n+1}}{T})\, S_j^{n+1} \end{aligned}$$

$$- S_j^{n+1}\left(c_I(\hat{u}_j^{n+1}+S_j^{n+1}\breve{u}_j^{n+1},2\phi_j^n-\phi_j^{n-1})-c_I(2u_j^n-u_j^{n-1},\hat{\phi}_j^{n+1}+S_j^{n+1}\breve{\phi}_j^{n+1})\right)=0.$$

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

$$S_j^{n+1}\left(A_j^{n+1}S_j^{n+1}+B_j^{n+1}\right)=0 \qquad \Longrightarrow \qquad S_j^{n+1}=-\frac{B_j^{n+1}}{A_j^{n+1}}. \tag{3.8}$$

where

$$A_j^{n+1}=(\frac{3}{2\Delta t}+\frac{1}{T})exp(-\frac{2t^{n+1}}{T})-c_I(\breve{u}_j^{n+1},2\phi_j^n-\phi_j^{n-1})+c_I(2u_j^n-u_j^{n-1},\breve{\phi}_j^{n+1})$$
$$B_j^{n+1}=-\frac{2}{\Delta t}r_j^n exp(-\frac{t^{n+1}}{T})+\frac{1}{2\Delta t}r_j^{n-1}exp(-\frac{t^{n+1}}{T})-c_I(\hat{u}_j^{n+1},2\phi_j^n-\phi_j^{n-1})+c_I(2u_j^n-u_j^{n-1},\hat{\phi}_j^{n+1}).$$

After getting $\hat{u}_j^{n+1},\breve{u}_j^{n+1}$, $\hat{\phi}_j^{n+1},\breve{\phi}_j^{n+1}$, S_j^{n+1} can be computed directly using formula (3.8), and then we have

$$u_j^{n+1}=\hat{u}_j^{n+1}+S_j^{n+1}\breve{u}_j^{n+1},\quad \phi_j^{n+1}=\hat{\phi}_j^{n+1}+S_j^{n+1}\breve{\phi}_j^{n+1},\quad p_j^{n+1}=\frac{4}{3}p_j^n-\frac{1}{3}p_j^{n-1}-\frac{1}{3\alpha\Delta t}\nabla\cdot u_j^{n+1}. \tag{3.9}$$

3.2. Algebraic Systems. The proposed SAV-BDF2AC-En scheme will be compared to other schemes such as SAV-BDF2AC-NonEn and SAV-BDF2-En stated below for computational efficiency check. Specifically, the SAV-BDF2AC-NonEn method is using AC decoupling but without ensemble technique; the SAV-BDF2-En method is an ensemble scheme without using AC decoupling.

ALGORITHM 3.1 (SAV-BDF2AC-NonEn). *Find* $(u_j^{n+1},p_j^{n+1},\phi_j^{n+1})\in X_f\times Q_f\times X_p$ *and* r_j^{n+1} *satisfying* $\forall\ (v,\psi)\in X_f\times X_p$,

$$\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},v\right)_f+\nu(\nabla u_j^{n+1},\nabla v)_f+\sum_i\int_I\eta_{i,j}(u_j^{n+1}\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\ ds-\left(p_j^{n+1},\nabla\cdot v\right)_f$$
$$+\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(v,2\phi_j^n-\phi_j^{n-1})=(f_{f,j}^{n+1},v)_f,$$
$$\alpha\Delta t(3p_j^{n+1}-4p_j^n+p_j^{n-1})+\nabla\cdot u_j^{n+1}=0,$$
$$gS_0\left(\frac{3\phi_j^{n+1}-4\phi_j^n+\phi_j^{n-1}}{2\Delta t},\psi\right)_p+g(\mathcal{K}_j\nabla\phi_j^{n+1},\nabla\psi)_p-\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(2u_j^n-u_j^{n-1},\psi)=g(f_{p,j}^{n+1},\psi)_p,$$
$$\frac{3r_j^{n+1}-4r_j^n+r_j^{n-1}}{2\Delta t}=-\frac{1}{T}r_j^{n+1}+\frac{1}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1},2\phi_j^n-\phi_j^{n-1})-c_I(2u_j^n-u_j^{n-1},\phi_j^{n+1})\right).$$

ALGORITHM 3.2 (SAV-BDF2-En). *Find* $(u_j^{n+1},p_j^{n+1},\phi_j^{n+1})\in X_f\times Q_f\times X_p$ *and* r_j^{n+1} *satisfying* $\forall\ (v,q,\psi)\in X_f\times Q_f\times X_p$,

$$\left(\frac{3u_j^{n+1}-4u_j^n+u_j^{n-1}}{2\Delta t},v\right)_f+\nu(\nabla u_j^{n+1},\nabla v)_f+\sum_i\int_I\bar{\eta}_i(u_j^{n+1}\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\ ds-\left(p_j^{n+1},\nabla\cdot v\right)_f$$
$$+\sum_i\int_I(\eta_{i,j}-\bar{\eta}_i)((2u_j^n-u_j^{n-1})\cdot\widehat{\tau}_i)(v\cdot\widehat{\tau}_i)\ ds+\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(v,2\phi_j^n-\phi_j^{n-1})=(f_{f,j}^{n+1},v)_f,$$
$$(q,\nabla\cdot u_j^{n+1})_f=0,$$
$$gS_0\left(\frac{3\phi_j^{n+1}-4\phi_j^n+\phi_j^{n-1}}{2\Delta t},\psi\right)_p+g(\bar{\mathcal{K}}\nabla\phi_j^{n+1},\nabla\psi)_p+g((\mathcal{K}_j-\bar{\mathcal{K}})\nabla(2\phi_j^n-\phi_j^{n-1}),\nabla\psi)_p$$
$$-\frac{r_j^{n+1}}{exp(-\frac{t^{n+1}}{T})}c_I(2u_j^n-u_j^{n-1},\psi)=g(f_{p,j}^{n+1},\psi)_p,$$

$$\frac{3r_j^{n+1}-4r_j^n+r_j^{n-1}}{2\Delta t}=-\frac{1}{T}r_j^{n+1}+\frac{1}{exp(-\frac{t^{n+1}}{T})}\left(c_I(u_j^{n+1},2\phi_j^n-\phi_j^{n-1})-c_I(2u_j^n-u_j^{n-1},\phi_j^{n+1})\right).$$

We then use the finite element method for spatial discretization. Let $S_h^2(\Omega)$ and $S_h^1(\Omega)$ denote the space of P2 and P1 finite element functions on domain Ω respectively. The basis functions of $S_h^2(D_f)^2$, $S_h^1(D_f)$, and $S_h^2(D_p)$ are $\{\chi_j^u\}_{j=1}^{N_u}$, $\{\chi_j^p\}_{j=1}^{N_p}$, and $\{\chi_j^\phi\}_{j=1}^{N_\phi}$ respectively. The finite element solutions will be represented by vectors of nodal values, denoted in bold. A superscript n will be applied to mark the $n-$th time step, and a subscript j will denote the j-th sample. Let $\mathbf{M}_{uu}$, $\mathbf{M}_p$, and $\mathbf{M}_\phi$ denote the mass matrices for velocity u, pressure p, hydraulic head ϕ, respectively; the matrices $\mathbf{S}_{uu}$ and $\mathbf{S}_\phi$ are the stiffness matrices of the Poisson operator corresponding to $S_h^2(D_f)^2$ and $S_h^1(D_p)$, respectively. We also define matrices $\mathbf{D}_{uu}$, $\mathbf{D}_{uup}$, $\mathbf{C}_{uu\phi}$, $\mathbf{B}_{juu}$, and $\mathbf{S}_{j\phi}$ whose entries are given as follows

$$[\mathbf{D}_{uu}]_{kl}=(\nabla\cdot\chi_l^u,\nabla\cdot\chi_k^u)_f,\quad [\mathbf{D}_{uup}]_{kl}=(\chi_l^p,\nabla\cdot\chi_k^u)_f,\quad [\mathbf{C}_{uu\phi}]_{kl}=\int_I\chi_l^\phi(\chi_k^u\cdot\widehat{n}_f)$$
$$[\mathbf{B}_{juu}]_{kl}=\int_I\eta_j(\chi_l^u\cdot\widehat{\tau})(\chi_k^u\cdot\widehat{\tau}),\quad [\mathbf{S}_{j\phi}]_{kl}=(\mathcal{K}_j\nabla\chi_l^u,\nabla\chi_k^u)_p.$$

The mean matrices $\overline{\mathbf{B}}_{uu}=\frac{1}{J}\sum_{j=1}^J\mathbf{B}_{juu}$, $\overline{\mathbf{S}}_\phi=\frac{1}{J}\sum_{j=1}^J\mathbf{S}_{j\phi}$ will also be used in ensemble algorithms.

The algebraic systems of different schemes that will be considered in efficiency test are listed below.

1. SAV-BDF2AC-En:

$$\mathbf{A}_{\text{uAC}}\hat{\mathbf{u}}_j^{n+1}=\hat{\mathbf{b}}_j^{n+1},\ \mathbf{A}_{\text{uAC}}\breve{\mathbf{u}}_j^{n+1}=\breve{\mathbf{b}}_j^{n+1},\quad \mathbf{A}_\phi\hat{\boldsymbol{\phi}}_j^{n+1}=\hat{\mathbf{c}}_j^{n+1},\ \mathbf{A}_\phi\breve{\boldsymbol{\phi}}_j^{n+1}=\breve{\mathbf{c}}_j^{n+1},\ \mathbf{M}_p\mathbf{p}_j^{n+1}=\mathbf{d}_j^{n+1}$$

with

$$\mathbf{A}_{\text{uAC}}=\frac{3}{2\Delta t}\mathbf{M}_{uu}+\nu\mathbf{S}_{uu}+\overline{\mathbf{B}}_{uu}+\frac{1}{3\alpha\Delta t}\mathbf{D}_{uu},\qquad \mathbf{A}_\phi=\frac{3S_0}{2\Delta t}\mathbf{M}_\phi+\overline{\mathbf{S}}_\phi,$$
$$\hat{\mathbf{b}}_j^{n+1}=\mathbf{f}_{f,j}^{n+1}+\mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n-\tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1})-(\mathbf{B}_{juu}-\overline{\mathbf{B}}_{uu})(2\mathbf{u}_j^n-\mathbf{u}_j^{n-1})+\mathbf{D}_{uup}(\frac{4}{3}\mathbf{p}_j^n-\frac{1}{3}\mathbf{p}_j^{n-1}),$$
$$\breve{\mathbf{b}}_j^{n+1}=-g\mathbf{C}_{uu\phi}(2\boldsymbol{\phi}_j^n-\boldsymbol{\phi}_j^{n-1}),$$
$$\hat{\mathbf{c}}_j^{n+1}=\mathbf{f}_{p,j}^{n+1}+\mathbf{M}_\phi(\tfrac{2S_0}{\Delta t}\boldsymbol{\phi}_j^n-\tfrac{S_0}{2\Delta t}\boldsymbol{\phi}_j^{n-1})-(\mathbf{S}_{j\phi}-\overline{\mathbf{S}}_\phi)(2\boldsymbol{\phi}_j^n-\boldsymbol{\phi}_j^{n-1}),$$
$$\breve{\mathbf{c}}_j^{n+1}=\mathbf{C}_{uu\phi}^T(2\mathbf{u}_j^n-\mathbf{u}_j^{n-1}),$$
$$\mathbf{d}_j^{n+1}=\mathbf{M}_p(\frac{4}{3}\mathbf{p}_j^n-\frac{1}{3}\mathbf{p}_j^{n-1})-\frac{1}{3\alpha\Delta t}\mathbf{D}_{uup}^T\mathbf{u}_j^{n+1}.$$

2. SAV-BDF2AC-NonEn:

$$\mathbf{A}_{\text{uAC}}^{(j)}\hat{\mathbf{u}}_j^{n+1}=\hat{\mathbf{e}}_j^{n+1},\ \mathbf{A}_{\text{uAC}}^{(j)}\breve{\mathbf{u}}_j^{n+1}=\breve{\mathbf{b}}_j^{n+1},\ \mathbf{A}_\phi^{(j)}\hat{\boldsymbol{\phi}}_j^{n+1}=\hat{\mathbf{h}}_j^{n+1},\ \mathbf{A}_\phi^{(j)}\breve{\boldsymbol{\phi}}_j^{n+1}=\breve{\mathbf{c}}_j^{n+1},\ \mathbf{M}_p\mathbf{p}_j^{n+1}=\mathbf{d}_j^{n+1}$$

with

$$\mathbf{A}_{\text{uAC}}^{(j)}=\frac{3}{2\Delta t}\mathbf{M}_{uu}+\nu\mathbf{S}_{uu}+\mathbf{B}_{juu}+\frac{1}{3\alpha\Delta t}\mathbf{D}_{uu},\qquad \mathbf{A}_\phi^{(j)}=\frac{3S_0}{2\Delta t}\mathbf{M}_\phi+\mathbf{S}_{j\phi},$$
$$\hat{\mathbf{e}}_j^{n+1}=\mathbf{f}_{f,j}^{n+1}+\mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n-\tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1})+\mathbf{D}_{uup}(\frac{4}{3}\mathbf{p}_j^n-\frac{1}{3}\mathbf{p}_j^{n-1}),$$
$$\hat{\mathbf{h}}_j^{n+1}=\mathbf{f}_{p,j}^{n+1}+\mathbf{M}_\phi(\tfrac{2S_0}{\Delta t}\boldsymbol{\phi}_j^n-\tfrac{S_0}{2\Delta t}\boldsymbol{\phi}_j^{n-1}).$$

3. SAV-BDF2-En:

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

with

$$\mathbf{A}_{\text{uNoAC}} = \begin{pmatrix} \frac{3}{2\Delta t}\mathbf{M}_{uu} + \nu\mathbf{S}_{uu} + \overline{\mathbf{B}}_{uu} & -\mathbf{D}_{uup} \\ -\mathbf{D}_{uup}^T & \mathbf{0} \end{pmatrix},$$

$$\hat{\mathbf{m}}_j^{n+1} = \mathbf{f}_{f,j}^{n+1} + \mathbf{M}_{uu}(\tfrac{2}{\Delta t}\mathbf{u}_j^n - \tfrac{1}{2\Delta t}\mathbf{u}_j^{n-1}) - (\mathbf{B}_{juu} - \overline{\mathbf{B}}_{uu})(2\mathbf{u}_j^n - \mathbf{u}_j^{n-1}).$$

As one can see, the ensemble scheme results in a common coefficient matrix for all realizations. The J simulations can be simultaneously handled by a direct linear solver such as LU factorization when the problem is of small scale and by block iterative solvers when it is in large-scale. In contrast, the non-ensemble scheme needs to handle J systems for different realizations one by one.

In next section, we will use iterative solvers to observe the computational efficiency of the three schemes stated above. Specifically, for the SPD matrix $\mathbf{A}_{\text{uAC}}^{(j)}$, the classical CG solver with multigrid preconditioner is enough. For the common matrix $\mathbf{A}_{\text{uAC}}$, however, the block conjugate gradient (Block CG [75, 71, 41]) solver will be applied to reduce redundant information from different samples. We perform similarly for $\mathbf{A}_\phi$ and $\mathbf{A}_\phi^{(j)}$. As for the coefficient matrix $\mathbf{A}_{\text{uNoAC}}$, CG can not be applied due to matrix indefinity, two feasible choices are minimum residual method (MINRES) with block diagonal preconditioner

$$\begin{pmatrix} \frac{3}{2\Delta t}\mathbf{M}_{uu} + \nu\mathbf{S}_{uu} + \overline{\mathbf{B}}_{uu} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_p \end{pmatrix}$$

and generalized minimum residual method (GMRES) with block triangular preconditioner

$$\begin{pmatrix} \frac{3}{2\Delta t}\mathbf{M}_{uu} + \nu\mathbf{S}_{uu} + \overline{\mathbf{B}}_{uu} & -\mathbf{D}_{uup} \\ \mathbf{0} & -\mathbf{M}_p \end{pmatrix}.$$

In our work, the later performs better in terms of both the number of iterations and the CPU time. For more details on preconditioner, one can refer to [18]. While ensembling technique is used in addition, we take the block GMRES algorithm with deflation [5, BFGMRESD(m)] to achieve J realizations simultaneously. To sum up, the algebraic solvers for above schemes are listed below in the table:

Coeffcient matrix	iterative solver	preconditioner	solver of preconditioner
$\mathbf{A}_{\text{uAC}}$	Block CG	Multigrid	Multigrid
$\mathbf{A}_{\text{uAC}}^{(j)}$	CG	Multigrid	Multigrid
$\mathbf{A}_{\text{uNoAC}}$	Block GMRES	Block Triangular	Multigrid preconditioned CG
$\mathbf{A}_\phi$	Block CG	Multigrid	Multigrid
$\mathbf{A}_\phi^{(j)}$	CG	Multigrid	Multigrid

REMARK 3.3. *Although ill-conditioning of $\mathbf{A}_{\text{uAC}}$ is a disadvantage, this ill-conditioning is not severe when measured by the condition number at the solution, also called effective condition number. This has been studied recently by Layton and Xu in [65]. The matrix $\mathbf{A}_{\text{uAC}}$ has condition number*

$$\kappa = \mathcal{O}(c\Delta t^{-1} + c(\nu,\bar{\eta})h^{-2} + c(\alpha)\Delta t^{-1}h^{-2}).$$

The effective condition number for a general system $Ac = b$ is defined as $\kappa^*(c) = \|A^{-1}\|\frac{\|Ac\|}{\|c\|}$. Note that the relative error in solving $Ac = b$ is bounded by $\kappa^*(c)$ multiplying the relative residual. For the linear system $\mathbf{A}_{\text{uAC}}\hat{\mathbf{u}}_j^{n+1} = \hat{\mathbf{b}}_j^{n+1}$, one can prove that

$$\kappa^*(\hat{\mathbf{u}}_j^{n+1}) \le c\Delta t^{-1} + c(\nu,\bar{\eta})h^{-2} + c(\alpha)\Delta t^{-1}h^{-1}\frac{\|\nabla\cdot\hat{\mathbf{u}}_j^{n+1}\|}{\|\hat{\mathbf{u}}_j^{n+1}\|}$$

by following the steps in [65]. Since the solution is approximately incompressible, $\frac{\|\nabla\cdot\hat{\mathbf{u}}_j^{n+1}\|}{\|\hat{\mathbf{u}}_j^{n+1}\|}$ can be small. Therefore, the effective condition number $\kappa^*(\hat{\mathbf{u}}_j^{n+1})$ can be much smaller than the standard condition number κ, as observed in [65].

4. Numerical Tests. In this section, we perform numerical experiments to validate the convergence rate of the proposed SAV-BEAC-En and SAV-BDF2AC-En algorithms, then study the computational efficiency of SAV-BDF2AC-En with stochastic hydraulic conductivity, and finally make a realistic application. For spatial discretization, Taylor-Hood elements (P2-P1) for the Stokes problem and piecewise quadratic finite elements (P2) for the Darcy problem are used. In all simulations, the AC parameter α is set to be one. Our MATLAB implementation is based on the data structure of iFEM package.

Table 4.1: Fluid velocity: errors at $T = 5$ and convergence rates of the SAV-BEAC-En algorithm ($J = 3$) with $\Delta t = h$.

Δt	$\|u_h - u\|_{H^1}^{E,1}$	Rate	$\|u_h - u\|_{H^1}^{E,2}$	Rate	$\|u_h - u\|_{H^1}^{E,3}$	Rate
1/8	8.518×10^{-2}	-	8.594×10^{-2}	-	8.749×10^{-2}	-
1/16	4.226×10^{-2}	1.01	4.239×10^{-2}	1.02	4.282×10^{-2}	1.03
1/32	2.090×10^{-2}	1.02	2.092×10^{-2}	1.02	2.108×10^{-2}	1.02
1/64	1.038×10^{-2}	1.01	1.038×10^{-2}	1.01	1.046×10^{-2}	1.01
1/128	5.168×10^{-3}	1.01	5.170×10^{-3}	1.01	5.207×10^{-3}	1.01

Table 4.2: Fluid pressure: errors at $T = 5$ and convergence rates of the SAV-BEAC-En algorithm ($J = 3$) with $\Delta t = h$.

Δt	$\|p_h - p\|_{L^2}^{E,1}$	Rate	$\|p_h - p\|_{L^2}^{E,2}$	Rate	$\|p_h - p\|_{L^2}^{E,3}$	Rate
1/8	9.186×10^{-2}	-	8.986×10^{-2}	-	9.022×10^{-2}	-
1/16	4.657×10^{-2}	0.98	4.514×10^{-2}	0.98	4.576×10^{-2}	0.98
1/32	2.335×10^{-2}	1.00	2.272×10^{-2}	0.98	2.333×10^{-2}	0.97
1/64	1.168×10^{-2}	1.00	1.142×10^{-2}	0.98	1.181×10^{-2}	0.98
1/128	5.841×10^{-3}	1.00	5.729×10^{-3}	1.00	5.943×10^{-3}	0.99

Table 4.3: Hydraulic head: errors at $T = 5$ and convergence rates of the SAV-BEAC-En algorithm ($J = 3$) with $\Delta t = h$.

Δt	$\|\phi_h - \phi\|_{H^1}^{E,1}$	Rate	$\|\phi_h - \phi\|_{H^1}^{E,2}$	Rate	$\|\phi_h - \phi\|_{H^1}^{E,3}$	Rate
1/8	7.551×10^{-2}	-	4.424×10^{-2}	-	4.563×10^{-2}	-
1/16	3.761×10^{-2}	1.01	2.174×10^{-2}	1.03	2.271×10^{-2}	1.01
1/32	1.865×10^{-2}	1.01	1.067×10^{-2}	1.03	1.131×10^{-2}	1.01
1/64	9.263×10^{-3}	1.01	5.269×10^{-3}	1.02	5.643×10^{-3}	1.00
1/128	4.614×10^{-3}	1.01	2.617×10^{-3}	1.01	2.817×10^{-3}	1.00

4.1. Convergence test. A model problem on $D_f = (0,1)\times(1,2)$ and $D_p = (0,1)\times(0,1)$ with interface $I = [0,1]\times\{1\}$ is considered, in which the physical parameters, g, ν, S_0, α_{BJS}, are set to be one. The hydraulic conductivity tensor is set as a diagonal matrix $\mathrm{diag}(k_{11}, k_{22})$ with k_{11} and k_{22} being constants. The initial conditions, boundary conditions, and the forcing terms are constructed from the exact solution given by

$$
\begin{aligned}
&u(x,y,t) = (u_1(x,y,t),\ u_2(x,y,t)),\\
&u_1(x,y,t) = \big(x^2(y-1)^2 + \exp(y/\sqrt{k_{11}})\big)\cos(t),\\
&u_2(x,y,t) = \big(\frac{2}{3}x(1-y)^3 + k_{22}(2 - \pi\sin(\pi x)\big)\cos(t),
\end{aligned}
$$

Table 4.4: Fluid velocity: errors at $T = 5$ and convergence rates of the SAV-BDF2AC-En algorithm ($J = 3$) with $\Delta t = h$.

Δt	$\|u_h - u\|_{H^1}^{E,1}$	Rate	$\|u_h - u\|_{H^1}^{E,2}$	Rate	$\|u_h - u\|_{H^1}^{E,3}$	Rate
1/8	2.039×10^{-2}	-	2.039×10^{-2}	-	2.039×10^{-2}	-
1/16	5.033×10^{-3}	2.02	5.036×10^{-3}	2.02	5.041×10^{-3}	2.02
1/32	1.254×10^{-3}	2.01	1.255×10^{-3}	2.01	1.256×10^{-3}	2.00
1/64	3.131×10^{-4}	2.00	3.133×10^{-4}	2.00	1.137×10^{-4}	2.00
1/128	7.825×10^{-5}	2.00	7.831×10^{-5}	2.00	7.841×10^{-5}	2.00

Table 4.5: Fluid pressure: errors at $T = 5$ and convergence rates of the SAV-BDF2AC-En algorithm ($J = 3$) with $\Delta t = h$.

Δt	$\|p_h - p\|_{L^2}^{E,1}$	Rate	$\|p_h - p\|_{L^2}^{E,2}$	Rate	$\|p_h - p\|_{L^2}^{E,3}$	Rate
1/8	2.219×10^{-2}	-	2.215×10^{-2}	-	2.216×10^{-2}	-
1/16	5.438×10^{-3}	2.03	5.494×10^{-3}	2.01	5.565×10^{-3}	1.99
1/32	1.352×10^{-3}	2.01	1.371×10^{-3}	2.00	1.395×10^{-3}	2.00
1/64	3.374×10^{-4}	2.00	3.428×10^{-4}	2.00	3.494×10^{-4}	2.00
1/128	8.433×10^{-5}	2.00	8.572×10^{-5}	2.00	8.744×10^{-5}	2.00

Table 4.6: Hydraulic head: errors at $T = 5$ and convergence rates of the SAV-BDF2AC-En algorithm ($J = 3$) with $\Delta t = h$.

Δt	$\|\phi_h - \phi\|_{H^1}^{E,1}$	Rate	$\|\phi_h - \phi\|_{H^1}^{E,2}$	Rate	$\|\phi_h - \phi\|_{H^1}^{E,3}$	Rate
1/8	4.928×10^{-3}	-	4.960×10^{-3}	-	5.131×10^{-3}	-
1/16	9.405×10^{-4}	2.39	8.889×10^{-4}	2.48	8.944×10^{-4}	2.52
1/32	2.085×10^{-4}	2.17	1.821×10^{-4}	2.29	1.758×10^{-4}	2.35
1/64	4.999×10^{-5}	2.06	4.133×10^{-5}	2.14	3.859×10^{-5}	2.19
1/128	1.233×10^{-5}	2.02	9.895×10^{-6}	2.06	9.049×10^{-6}	2.09

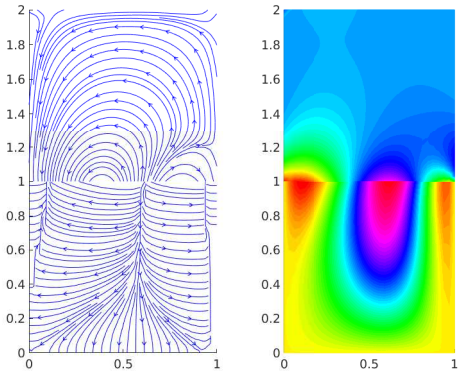
$$p(x, y, t) = (2 - \pi \sin(\pi x)) \sin(0.5\pi y) \cos(t),$$
$$\phi(x, y, t) = (2 - \pi \sin(\pi x))(1 - y - \cos(\pi y)) \cos(t).$$

We perform $J = 3$ realizations simultaneously in this convergence test. Each realization is chosen by setting a different hydraulic conductivity tensor from others, i.e. the j-th samples of k_{11} and k_{22} are

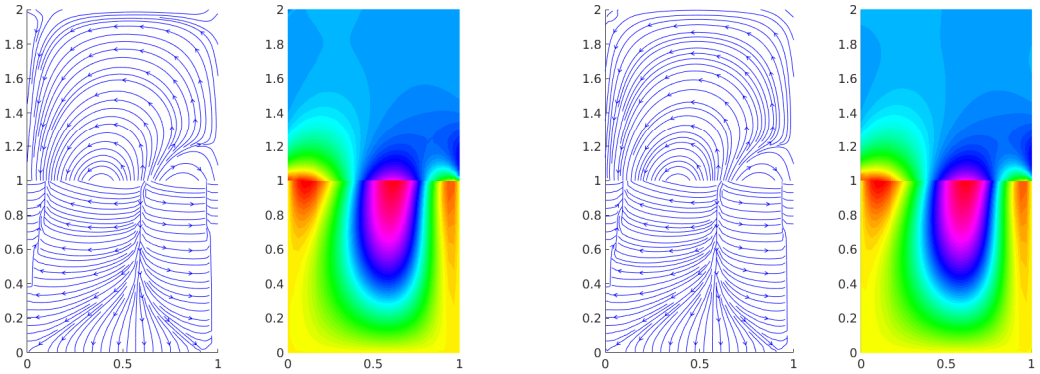
$$k_{11}^j = 1 - 0.1(j - 1), \quad k_{22}^j = 1 + 0.1(j - 1), \quad j = 1, 2, 3.$$

To check the temporal convergence rate, we consecutively refine the time step size Δt, from initial time step size $\Delta t = 1/8$ to final size $\Delta t = 1/128$, and alway choose $h = \Delta t$. In this setup, the expected error is $O(h^2 + \Delta t) = O(\Delta t)$ for SAV-BEAC-En and $O(h^2 + \Delta t^2) = O(\Delta t^2)$ for SAV-BDF2AC-En. The approximation errors at final time $T = 5$ by the SAV-BE-En scheme are listed in Table 4.1, 4.2, and 4.3, for the fluid velocity u, fluid pressure p, and hydraulic head ϕ respectively, illustrating that the SAV-BE-En algorithm is first order in time convergence. We also list the results by the SAV-BDF2AC-En scheme in Table 4.4, 4.5, and 4.6, from which we see the expected second order convergence is apparent.

4.2. Stochastic example. Next, we apply the SAV-BDF2AC-En algorithm for solving the stochastic Stokes-Darcy model with a random hydraulic conductivity tensor $\mathcal{K}(x, y, \omega)$ that depends on spatial variables. Here $\omega \in \Omega$, where $(\Omega, \mathcal{F}, \mathcal{P})$ is a complete probability space. The hydraulic conductivity $\mathcal{K}(x, y, \omega)$ is set to



(a) SAV-BDF2AC-En



(b) SAV-BDF2AC-NonEn (c) SAV-BDF2-En

Fig. 4.1: Streamlines and plots of $\mathbb{E}[u]$, $\mathbb{E}[-\mathcal{K}\nabla\phi]$, $\mathbb{E}[p]$, $\mathbb{E}[\phi]$ at $T = 5.0$ by different schemes using sparse-grid method with $J = 241$ collocation points, $h = 1/50, \Delta t = 1/100$.

be a diagonal stochastic tensor $\mathrm{diag}(k_{11}(x, y, \omega), k_{22}(x, y, \omega))$ that has continuous and bounded correlation function. To be specific, the entries $k_{11}(x, y, \omega)$ and $k_{22}(x, y, \omega)$ are given by the Karhunen-Loève expansion

$$k_{11}(x, y, \omega) = k_{22}(x, y, \omega) = a_0 + \sigma\sqrt{\lambda_0}Y_0(\omega) + \sum_{i=1}^{n_f} \sigma\sqrt{\lambda_i}[Y_i(\omega)\cos(i\pi x) + Y_{n_f+i}(\omega)\sin(i\pi x))], \tag{4.1}$$

where $\lambda_0 = \frac{1}{2}\sqrt{\pi L_c}$, $\lambda_i = \sqrt{\pi}L_c\exp(-\frac{1}{4}(i\pi L_c)^2)$ for $i = 1, \cdots, n_f$, and $Y_0, \cdots, Y_{2n_f}$ are indepdendent and identically uniformly distributed in the interval $[-\sqrt{3}, \sqrt{3}]$, so they have zero mean and unit variance. In this expriment, we take $n_f = 2$, so there are totally 5 random variables $Y_0, Y_1, \cdots, Y_4$. The other quantities are set as $L_c = 0.25, a_0 = 1, \sigma = 0.15$.

Other physical parameters in the model and the computational domain are set as in Subsection 4.1. As for the initial condition and Dirichlet boundary condition, they are given by

$$\begin{aligned}
&u(x, y, t, \omega) = (u_1(x, y, t, \omega),\ u_2(x, y, t, \omega)),\\
&u_1(x, y, t, \omega) = Y_0(\omega)\big(y^2 - 2y + 1\big)\cos(t),\\
&u_2(x, y, t, \omega) = Y_1(\omega)\big(x^2 - x\big)\cos(t),\\
&\phi(x, y, t, \omega) = Y_2(\omega)y\cos(t).
\end{aligned}$$

The model associates with forcing terms $f_f = (Y_3(\omega)xy, Y_3(\omega)xy)$, $f_p = Y_4(\omega)xy$.

The stochastic Stokes-Darcy problem is then solved by a sparse-grid collocation method, since it is non-intrusive and superior to the Monte Carlo method in terms of the curse of dimensionality. In numerical

Table 4.7: Computational performance using sparse-grid method with $J = 241$ collocation points, $h = 1/50, \Delta t = 1/100, T = 5.0$. Below n_{itr} denotes the number of iterations in using iterative linear solver and t_{cpu} denotes the CPU time in seconds.

	SAV-BDF2AC-En	SAV-BDF2AC-NonEn	SAV-BDF2-En
n_{itr} for u & p at $t = 5$	28 & 5	28×241 & 5×241	9
t_{cpu} for u & p at $t = 5$	14 s & 0.9 s	0.57×241 s & 0.01×241 s	45 s
n_{itr} for ϕ at $t = 5$	7	8×241	7
t_{cpu} for ϕ at $t = 5$	7.1 s	0.01×241 s	7.0 s
Average t_{cpu} per time step	27.6 s	0.58×241 s	61.8 s
Total CPU time	13923.1 s	62152.1 s	31008.9 s

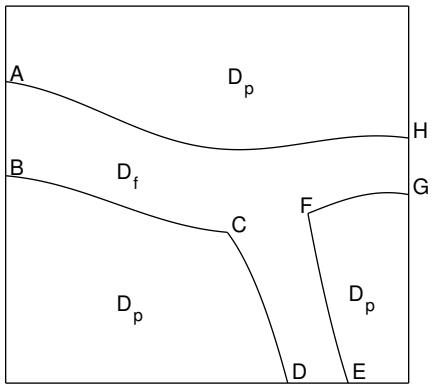


Fig. 4.2: Domains with curvy interface for simulating the subsurface flow in a karst aquifer.

implementation, the $J = 241$ collocation points are computed by the Smolyak formula utilizing Guassian quadrature. It ensures computational efficiency of integrating multidimensional functions by referring to a univariate Guassian quadrature. Taking $h = 1/50$ and $\Delta t = 1/100$, the numerical solutions at $T = 5$ by three different schemes, SAV-BDF2AC-En, SAV-BDF2AC-NonEn, and SAV-BDF2-En, are illustrated in Figure 4.1. In each subfigure, the streamlines of the expectations of fluid flow velocity u and porous media flow velocity $v = -\mathcal{K}\nabla\phi$ are plotted in the left, and the expectations of fluid flow pressure p and hydraulic head ϕ are plotted in the right. Figure 4.1 shows that all the schemes achieve almost identical simulations.

A particular test on computational efficiency of using these three algorithms is also reported in Table 4.7. It is apparent that the SAV-BDF2AC-En method takes much less CPU time than the SAV-BDF2AC-NonEn algorithm. This thanks to the design that the algebraic systems in the ensemble scheme share a constant matrix by all realizations, so the 241 linear systems in each time step can be simultaneously solved by block iterative solvers. As compared to SAV-BDF2-En, the proposed SAV-BDF2AC-En scheme still outperforms the coupled method in terms of CPU time, thanks to the splitting of velocity and pressure.

4.3. Realistic application. We then apply the proposed SAV-BDF2AC-En scheme to a more realistic simulation of the subsurface flow in a karst aquifer. As shown in Figure 4.2, the free flow domain D_f with curvy boundary $\overline{ABCDEFGH}$ is a T-shape conduit associating a curvy interface with the porous media flow D_p. The two domains form a unit square, and $A = (0, 0.8)$, $B = (0, 0.55)$, $C = (0.55, 0.4)$, $D = (0.7, 0)$, $E = (0.85, 0)$, $F = (0.75, 0.45)$, $G = (1, 0.5)$, and $H = (1, 0.7)$. The model parameters g, ν, and S_0 are equal to one, and $\alpha_{\text{BJS}} = 0.1$. In the simulation, we set the source terms to be zero and $\phi = 0$ on $\partial D_p \backslash I$. The hydraulic conductivity tensor is set to be $m\mathbf{I}$ with various magnitude m in our experiments. The inflow and

outflow boundary conditions for u are given as

$$u = \begin{cases} (s_1, 0) & on \quad \overline{AB} \\ (0, s_2) & on \quad \overline{DE} \\ (s_3, 0) & on \quad \overline{GH} \end{cases},$$

where s_1, s_2 and s_3 are constants.

With different hydraulic conductivity values, the numerical solutions of the Stokes-Darcy problem at $T = 1.0$ solved by SAV-BDF2AC-En ($h = 0.022$ and $\Delta t = 0.01$) are illustrated in Figure 4.3. There the computed expectations of u, $v = -\mathcal{K}\nabla\phi$, p, and ϕ for different scenarioes are presented. In all cases, the inflow conditions are fixed by setting $s_2 = 1$ and $s_3 = -1$, and the outflow condition is given by $s_1 = -1.5$. To test the effect of hydraulic conductivity on the solution, we set the magnitude m to be $1, 10^{-2}$, and 10^{-4}. The corresponding simulations are plotted from top to bottom. It is obvious that when m decreases, the flow speed in porous media is significantly reduced. The same phenomenon is also observed in Figure 4.4, which corresponds to the case $s_1 = -0.2$. Moreover, the larger hydraulic conductivity in Figure 4.3(a) causes the water easier to be pushed out of the conduit; the smaller conductivities in Figure 4.3(c) and (d) cause water to flow into the conduit near the outflow boundary.

Compared with the case $s_1 = -1.5$ in Figure 4.3, one can observe from Figure 4.4 that the less outflow rate causes more water to be pushed out of the conduit into the porous media, especially through the $\overline{AH}$ boundary. This is generally what happens during a rain season. In other words, comparing Figure 4.3(c) and 4.4(c) concludes that the more outflow rate causes more water to flow into the conduit from the porous media, which often happens during a dry season.

5. Conclusions. We have presented two fast and accurate algorithms that can be used to compute an ensemble of the Stokes-Darcy flows at one pass. The proposed algorithms are developed based on a SAV idea and the AC technique to fully decouple the primitive variables, leading to smaller algebraic linear systems and thus great savings in computer storage and CPU time. They also exploit the efficiency of solving linear systems with a common coefficient matrix and multiple right hands. For this purpose, very efficient block solvers such as block CG and block GMRES methods can be used to significantly reduce the CPU time of computing an ensemble of Stokes-Darcy flows. We proved the proposed algorithms are long time stable without any timestep conditions. Unconditional stability is generally difficult to achieve for partitioned numerical schemes of coupled flow problems. But the recently developed SAV idea puts forth a way to design such schemes. After introducing auxiliary scalar variables into the Stokes-Darcy equations, we also present efficient implementation methods to decouple the auxiliary scalar variables from computation of other primitive variables. Numerical tests show that our proposed algorithms are very fast in computing an ensemble of Stokes-Darcy equations compared to a non-ensemble numerical method or an ensemble scheme without AC technique, while maintaining comparable accuracy.

Declarations.

Funding. Nan Jiang was partially supported by the US National Science Foundation grants DMS-1720001 and DMS-2120413. Huanhuan Yang was supported in part by the National Natural Science Foundation of China under grant 11801348, the key research projects of general universities in Guangdong Province (grant 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).

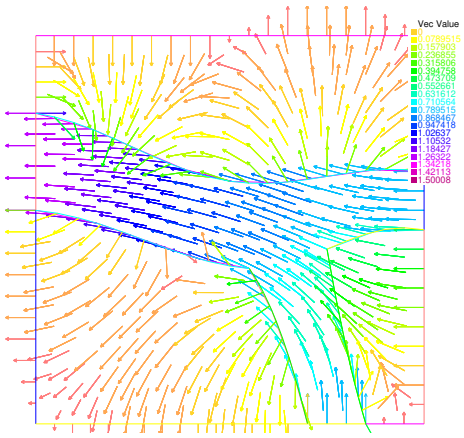
Conflicts of interest. The authors declare that they have no conflict of interest.

Availability of data and material. Parts of the data and materials are available upon reasonable request.

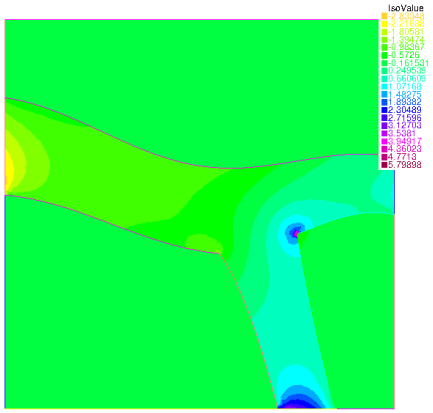
Code availability. Parts of the code are available upon reasonable request.

REFERENCES

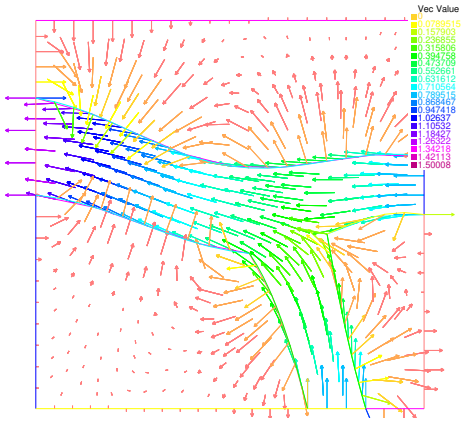
[1] I. Babuška, F. Nobile and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007), 1005-1034.



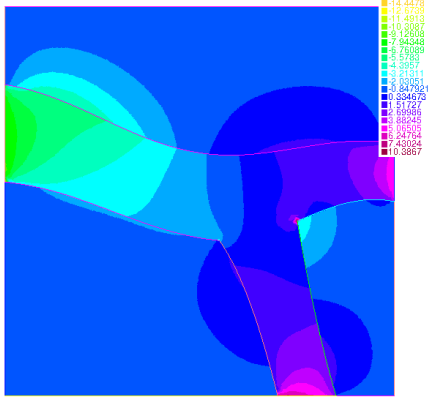
(a) $\mathbb{E}[u]$ and $\mathbb{E}[v]$ with $s_1 = -1.5,\ m = 1$



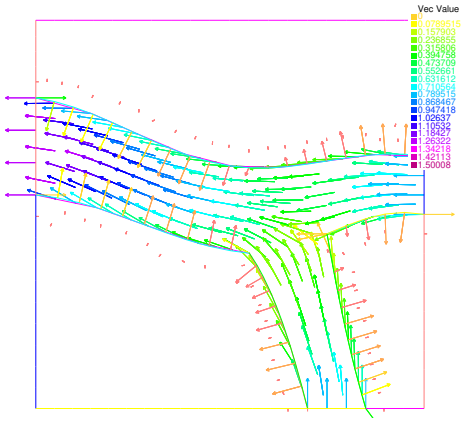
(b) $\mathbb{E}[p]$ and $\mathbb{E}[\phi]$ with $s_1 = -1.5,\ m = 1$



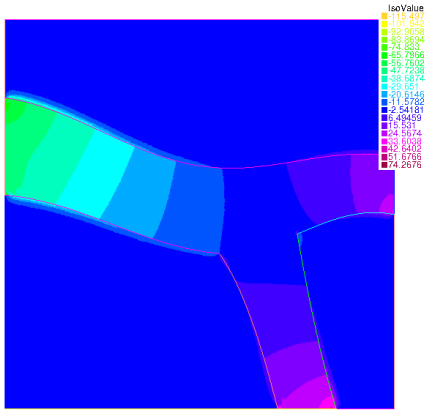
(c) $\mathbb{E}[u]$ and $\mathbb{E}[v]$ with $s_1 = -1.5,\ m = 10^{-2}$



(d) $\mathbb{E}[p]$ and $\mathbb{E}[\phi]$ with $s_1 = -1.5,\ m = 10^{-2}$



(e) $\mathbb{E}[u]$ and $\mathbb{E}[v]$ with $s_1 = -1.5,\ m = 10^{-4}$



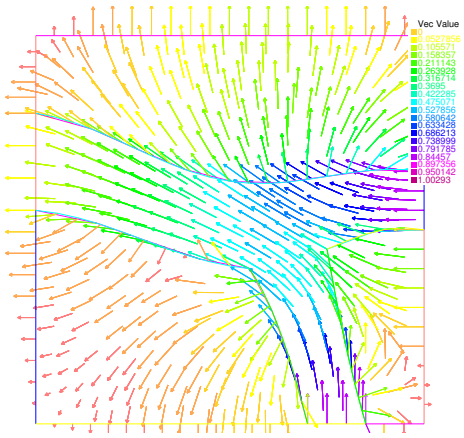
(f) $\mathbb{E}[p]$ and $\mathbb{E}[\phi]$ with $s_1 = -1.5,\ m = 10^{-4}$

Fig. 4.3: Simulations with outflow condition $s_1 = -1.5$. Left: expectations of fluid flow velocity u and porous media flow velocity $v = -\mathcal{K}\nabla\phi$; right: expectations of fluid flow pressure p and hydraulic head ϕ. From top to bottom: conductivity magnitude $m = 1,\ 10^{-2},\ 10^{-4}$.

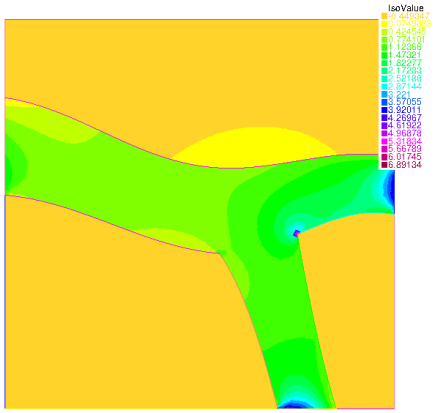
[2] A. Barth and A. Lang, *Multilevel Monte Carlo method with applications to stochastic partial differential equations*, International Journal of Computer Mathematics, 89 (2012), 2479-2498.

[3] J. Bear, *Hydraulics of Groundwater*, McGraw-Hill, New York, 1979.

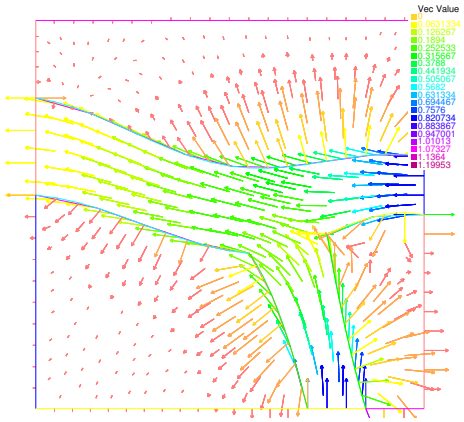
[4] G. Beavers and D. Joseph, *Boundary Conditions at a Naturally Impermeable Wall*, J. Fluid Mech., 30 (1967), 197-207.



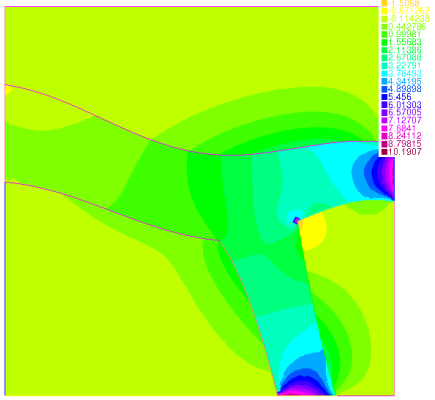
(a) $\mathbb{E}[u]$ and $\mathbb{E}[v]$ with $s_1 = -0.2,\ m = 1$



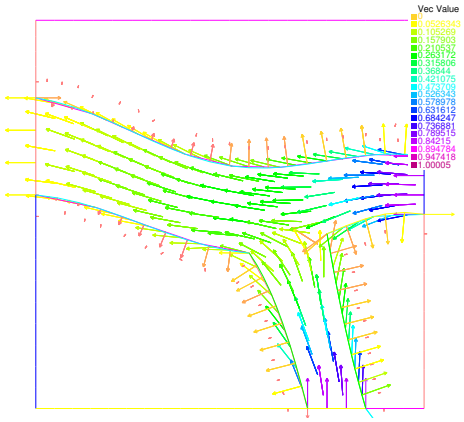
(b) $\mathbb{E}[p]$ and $\mathbb{E}[\phi]$ with $s_1 = -0.2,\ m = 1$



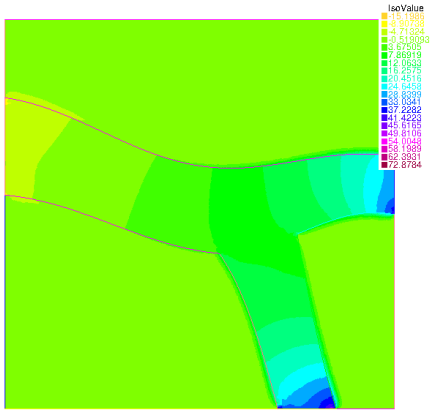
(c) $\mathbb{E}[u]$ and $\mathbb{E}[v]$ with $s_1 = -0.2,\ m = 10^{-2}$



(d) $\mathbb{E}[p]$ and $\mathbb{E}[\phi]$ with $s_1 = -0.2,\ m = 10^{-2}$



(e) $\mathbb{E}[u]$ and $\mathbb{E}[v]$ with $s_1 = -0.2,\ m = 10^{-4}$



(f) $\mathbb{E}[p]$ and $\mathbb{E}[\phi]$ with $s_1 = -0.2,\ m = 10^{-4}$

Fig. 4.4: Simulations with outflow condition $s_1 = -0.2$. Left: expectations of fluid flow velocity u and porous media flow velocity $v = -\mathcal{K}\nabla\phi$; right: expectations of fluid flow pressure p and hydraulic head ϕ. From top to bottom: conductivity magnitude $m = 1,\ 10^{-2},\ 10^{-4}$.

[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] Y. Cao, M. Gunzburger, X. He and X. Wang, *Parallel, non-iterative, multi-physics domain decomposition methods for time-dependent Stokes-Darcy systems*, Mathematics of Computation, 83 (2014), 1617-1644.

[7] J. Carter and N. Jiang, *Numerical Analysis of A Second Order Ensemble Method for Evolutionary Magnetohydrodynamics Equations at Small Magnetic Reynolds Number*, Numerical Methods for Partial Differential Equations, 38 (2022), 1407-1436.
[8] W. Chen and M. Gunzburger, D. Sun and X. Wang, *Efficient and long-time accurate second-order methods for the Stokes-Darcy system*, SIAM Journal on Numerical Analysis, 51 (2013), 2563-2584.
[9] R.M. Chen, W. Layton and M. McLaughlin, *Analysis of variable-step/non-autonomous artificial compression methods*, Journal of Mathematical Fluid Mechanics, 21 (2019), 30.
[10] A.J. Chorin, *Numerical solution of the Navier-Stokes equations*, Mathematics of computation, 22 (1968), 745-762.
[11] A.J. Chorin, *A numerical method for solving incompressible viscous flow problems*, Journal of Computational Physics, 2 (1967), pp. 12-26.
[12] J. Connors, *An ensemble-based conventional turbulence model for fluid-fluid interactions*, Int. J. Numer. Anal. Model., 15 (2018), 492-519.
[13] V. DeCaria, W. Layton and M. McLaughlin, *A conservative, second order, unconditionally stable artificial compression method*, Computer Methods in Applied Mechanics and Engineering, 325 (2017), pp. 733-747.
[14] V. DeCaria, T. Iliescu, W. Layton, M. McLaughlin and M. Schneier, *An artificial compression reduced order model*, SIAM Journal on Numerical Analysis, 58 (2020), pp. 565-589.
[15] M. Discacciati, E. Miglio and A. Quarteroni, *Mathematical and numerical models for coupling surface and groundwater flows*, Appl. Numer. Math., 43 (2002), 57-74.
[16] M. Discacciati and A. Quarteroni, *Convergence analysis of a subdomain iterative method for the finite element approximation of the coupling of Stokes and Darcy equations*, Computing and Visualization in Science, 6 (2004), 93-103.
[17] W. E and J.-G. Liu, *Projection method I: Convergence and numerical boundary layers*, SIAM Journal on Numerical Analysis, 32 (1995), 1017-1057.
[18] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, 2nd edition, Oxford University Press, New York, 2014.
[19] V. Ervin, E. Jenkins and S. Sun, *Coupling nonlinear Stokes and Darcy flow using mortar finite elements*, Appl. Numer. Math., 61 (2011), 1198-1222.
[20] J. Fiordilino, *A second order ensemble timestepping algorithm for natural convection*, SIAM Journal on Numerical Analysis, 56 (2018), 816-837.
[21] J. Fiordilino and S. Khankan, *Ensemble timestepping algorithms for natural convection*, International Journal of Numerical Analysis and Modeling, 15 (2018), 524-551.
[22] Y. T. Feng, D. R. J. Owen and D. Peric, *A block Conjugate Gradient method applied to linear systems with multiple right hand sides*, Comp. Meth. Appl. Mech., 127 (1995), 1-4.
[23] E. Gallopulos and V. Simoncini, *Convergence of BLOCK GMRES and matrix polynomials*, Lin. Alg. Appl., 247 (1996), 97-119.
[24] B. Ganis, H. Klie, M. Wheeler, T. Wildey, I. Yotov and D. Zhang, *Stochastic collocation and mixed finite elements for flow in porous media*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), 3547-3559.
[25] V. Girault, D. Vassilev and I. Yotov, *Mortar multiscale finite element methods for Stokes-Darcy flows*, Numerische Mathematik, 127 (2014), 93-165.
[26] K. Goda, *A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows*, Journal of Computational Physics, 30 (1979), 76-95.
[27] J. Guermond and P. Minev, *High-order time stepping for the incompressible Navier-Stokes equations*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2656-A2681.
[28] J. Guermond and P. Minev, *High-order adaptive time stepping for the incompressible Navier-Stokes equations*, SIAM Journal on Scientific Computing, 41 (2019), pp. A770-A788.
[29] J.L. Guermond, P. Minev and J. Shen, *An overview of projection methods for incompressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), 6011-6045.
[30] J.L. Guermond and J. Shen, *Velocity-correction projection methods for incompressible flows*, SIAM Journal on Numerical Analysis, 41 (2003), 112-134.
[31] J.L. Guermond and J. Shen, *On the error estimates for the rotational pressure-correction projection methods*, Mathematics of Computation, 73 (2004), 1719-1737.
[32] 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.
[33] 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.
[34] 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.
[35] 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.
[36] 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.
[37] 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.
[38] 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.
[39] 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.
[40] W. Jager and A. Mikelic, *On the Boundary Condition at the Interface Between a Porous Medium and a Free Fluid*, SIAM J. Appl. Math., 60 (2000), 1111-1127.
[41] H. Ji and Y. Li, *A breakdown-free block conjugate gradient method*, BIT Numerical Mathematics, 57(2) (2017), 379-403.
[42] N. Jiang, *A higher order ensemble simulation algorithm for fluid flows*, Journal of Scientific Computing, 64 (2015), 264-288.
[43] 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.
[44] N. Jiang, *A pressure-correction ensemble scheme for computing evolutionary Boussinesq equations*, Journal of Scientific Computing, 80 (2019), 315-350.
[45] 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.
[46] N. Jiang, M. Kubacki, W. Layton, M. Moraiti and H. Tran, *A Crank-Nicolson Leapfrog stabilization: unconditional stability and two Applications*, Journal of Computational and Applied Mathematics, 281 (2015), 263-276.
[47] N. Jiang and W. Layton, *An algorithm for fast calculation of flow ensembles*, International Journal for Uncertainty Quantification, 4 (2014), 273-301.
[48] 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.
[49] 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.
[50] 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.
[51] 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.
[52] 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.
[53] 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.
[54] N. Jiang, A. Takhirov and J. Waters, *Robust SAV-ensemble algorithms for parametrized flow problems with energy stable open boundary conditions*, Computer Methods in Applied Mechanics and Engineering, 392 (2022), 114709.
[55] N. Jiang and H. Yang, *Stabilized scalar auxiliary variable ensemble algorithms for parameterized flow problems*, SIAM Journal on Scientific Computing, 43(4) (2021), A2869-A2896.
[56] 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.
[57] 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.
[58] 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.
[59] B. Kuznetsov, N. Vladimirova and N. Yanenko, *Numerical Calculation of the Symmetrical Flow of Viscous Incompressible Liquid around a Plate (in Russian)*, Studies in Mathematics and its Applications, Moscow: Nauka, 1966.
[60] M. Kubacki and M. Moraiti, *Analysis of a second-order, unconditionally stable, partitioned method for the evolutionary Stokes-Darcy model*, Int. J. Numer. Anal. Model., 12 (2015), 704-730.
[61] W. Layton, *Introduction to the Numerical Analysis of Incompressible Viscous Flows*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2008.
[62] W. Layton and M. McLaughlin, *Doubly-adaptive artificial compression methods for incompressible flow*, Journal of Numerical Mathematics, 28 (2020), 175-192.
[63] W. Layton and H. Tran and C. Trenchea, *Analysis of long time stability and errors of two partitioned methods for uncoupling evolutionary groundwater-surface water flows*, SIAM J. Numer. Anal., 51 (2013), 248-272.
[64] W. Layton and H. Tran and X. Xiong, *Long time stability of four methods for splitting the evolutionary Stokes-Darcy problem into Stokes and Darcy subproblems*, Journal of Computational and Applied Mathematics, 236 (2012), 3198-3217.
[65] W. Layton and S. Xu, *Conditioning of linear systems arising from penalty methods*, Arxiv, 2022, https://arxiv.org/abs/2206.06971
[66] 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.
[67] Y. Li, Y. Hou and Y. Rong, *A second-order artificial compression method for the evolutionary Stokes-Darcy system*, Numerical Algorithms, 84 (2020), 1019-1048.
[68] 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.
[69] 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.
[70] Y. Luo and Z. Wang, *A multilevel Monte Carlo ensemble scheme for random parabolic PDEs*, SIAM Journal on Scientific Computing, 41 (2019), A622-A642.
[71] JF McCarthy, *Block-conjugate-gradient method*, Physical Review D, 40 (1989), 2149.
[72] M. Mohebujjaman and L. Rebholz, *An efficient algorithm for computation of MHD flow ensembles*, Computational Methods in Applied Mathematics, 17 (2017), 121-137.
[73] M. Mu and X. Zhu, *Decoupled Schemes for a Non-Stationary Mixed Stokes-Darcy Model*, Math. Comp., 79 (2010), 707-731.

[74] F. Nobile, R. Tempone and C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), 2309-2345.
[75] DP O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra and its Applications, 29 (1980), pp. 293-322.
[76] 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.
[77] 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.
[78] Y. Rong, W. Layton and H. Zhao, *Numerical analysis of an artificial compression method for Magnetohydrodynamic flows at low magnetic Reynolds numbers*, Journal of Scientific Computing, 76 (2018), pp. 1458-1483.
[79] P. Saffman, *On the boundary condition at the interface of a porous medium*, Stud. Appl. Math., 1 (1971), 93-101.
[80] L. Shan, H. Zheng and W. Layton, *A decoupling method with different subdomain time steps for the nonstationary Stokes–Darcy model*, Numer. Methods for Partial Differential Eq., 29 (2013), 549-583.
[81] J. Shen, *On error estimates of projection methods for Navier-Stokes equations: first-order schemes*, SIAM Journal on Numerical Analysis, 29 (1992), 57-77.
[82] J. Shen, *On error estimates of projection methods for Navier-Stokes equations: second-order schemes*, Mathematics of Computation, 65 (1996), 1039-1065.
[83] A. Takhirov, M. Neda, and J. Waters, *Time relaxation algorithm for flow ensembles*, Numerical Methods for Partial Differential Equations, 32 (2016), 757-777.
[84] 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.
[85] R. Temam *Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I)*, Arch. Rational. Mech. Anal., 33 (1969), pp. 135-153.
[86] R. Temam, *Sur l'approximation de la solution des equations de Navier-Stokes par la méthode des fractionnarires II*, Archive for Rational Mechanics and Analysis, 33 (1969), 377-385.
[87] J. van Kan, *A second-order accurate pressure-correction scheme for viscous incompressible flow*, SIAM Journal on Scientific and Statistical Computing, 7 (1987), 870-891.
[88] 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.