An Ensemble-Based Projection Method and Its Numerical Investigation

Shuai Yuan

Follow this and additional works at: https://scholarcommons.sc.edu/etd

Part of the Mathematics Commons

Recommended Citation

This Open Access Dissertation is brought to you by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact dillarda@mailbox.sc.edu.
AN ENSEMBLE-BASED PROJECTION METHOD AND ITS NUMERICAL INVESTIGATION

by

Shuai Yuan

Bachelor of Science
Hubei University 2012

Master of Science
Georgia Southern University 2014

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in
Mathematics
College of Arts and Sciences
University of South Carolina
2020

Accepted by:
Zhu Wang, Major Professor
Lili Ju, Co-Major Professor
Xinfeng Liu, Committee Member
Yi Sun, Committee Member
Yi Wang, Committee Member
Cheryl L. Addy, Vice Provost and Dean of the Graduate School
ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Zhu Wang, for your guidance and foresight. You have taught me many things about mathematics research, sharing exciting problems as well as equipping me with useful coding skills. This dissertation would not have been possible without you.

I am also deeply grateful to my co-advisor Dr. Lili Ju, thank you for many ways in which you have advised me for my future career and many valuable discussions.

Thank you, Dr. Xinfeng Liu, Dr. Yi Sun, Dr. Yi Wang for taking time to serve on my committee. To the many other faculty, staff, and graduate students in the mathematics department, thank you for your help and collaborations.

Finally, I would like to thank my parents for everything you have taught me. You have always supported me no matter what.
Abstract

In many cases, partial differential equation (PDE) models involve a set of parameters whose values may vary over a wide range in application problems, such as optimization, control and uncertainty quantification. Performing multiple numerical simulations in large-scale settings often leads to tremendous demands on computational resources. Thus, the ensemble method has been developed for accelerating a sequence of numerical simulations. In this work we first consider numerical solutions of Navier-Stokes equations under different conditions and introduce the ensemble-based projection method to reduce the computational cost. In particular, we incorporate a sparse grad-div stabilization into the method as a nonzero penalty term in discretization that does not strongly enforce mass conservation, and derive the long time stability and the error estimate. Numerical tests are presented to illustrate the theoretical results.

A simple way to solve the linear system generated in the ensemble method is to use a direct solver. Compared with individual simulations of the same problems, the ensemble method is more efficient because there is only one linear system needs to solve for the ensemble. However, for large-scale problems, iterative linear solvers have to be used. Therefore, in the second part of this work we investigate numerical performance of the ensemble method with block iterative solvers for two typical evolution problems: the heat equation and the Navier-Stokes equations. Numerical results are provided to demonstrate the effectiveness and efficiency of the ensemble method when working together with the block iterative solvers.
TABLE OF CONTENTS

Acknowledgments ................................................................. iii

Abstract ........................................................................ iv

List of Tables .................................................................... vii

List of Figures ..................................................................... viii

Chapter 1 Introduction .......................................................... 1
  1.1 Notations and preliminaries .............................................. 2
  1.2 The ensemble method ..................................................... 3
  1.3 Outline of dissertation .................................................... 20

Chapter 2 Ensemble-based projection method with sparse grad-
  div stabilization ................................................................... 22
  2.1 The projection method .................................................... 23
  2.2 Notations and preliminaries .............................................. 26
  2.3 Ensemble-based projection method and sparse grad-div stabilization . 27
  2.4 Numerical analysis ........................................................ 29
  2.5 Numerical experiments .................................................... 37
  2.6 Summary ........................................................................ 44
CHAPTER 3 NUMERICAL INVESTIGATION OF ENSEMBLE METHODS WITH BLOCK ITERATIVE SOLVERS FOR EVOLUTION PROBLEMS . . . 46

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

3.2 Ensemble methods for evolution problems . . . . . . . . . . . . . . 50

3.3 Block-based iterative solvers . . . . . . . . . . . . . . . . . . . . . . 53

3.4 Numerical experiments . . . . . . . . . . . . . . . . . . . . . . . . . . 60

3.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67

CHAPTER 4 FUTURE WORK . . . . . . . . . . . . . . . . . . . . . . . . . 71

BIBLIOGRAPHY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
# List of Tables

Table 2.1 The $L^2$ errors at final time ................................................................. 40
Table 2.2 Comparison of the efficiency (2d). ......................................................... 42
Table 2.3 Comparison of the efficiency (3d). ......................................................... 42
Table 2.4 Comparison of the efficiency (2d, $h = 1/8$). ........................................ 42
Table 2.5 Comparison of the divergence errors. .................................................... 44

Table 3.1 The $L^2$ errors at final time: first-order ensemble, $Q_1$ elements. .... 62
Table 3.2 The $L^2$ errors at final time: second-order ensemble, $Q_2$ elements. . 62
Table 3.3 CPU time comparison. ............................................................................. 62
Table 3.4 The $L^2$ errors at final time: first-order ensemble, $Q_2/Q_1$ elements.  64
Table 3.5 The $L^2$ errors at final time: second-order ensemble, $Q_2/Q_1$ elements. 65
Table 3.6 CPU time comparison. ............................................................................. 65
Table 3.7 CPU time comparison. ............................................................................. 67
List of Figures

Figure 2.1 (2d) Error plots of projection method with no stabilization, projection method with sparse grad-div stabilization and ensemble based projection method with sparse grad-div stabilization . . . . 40

Figure 2.2 Error plots of projection method with no stabilization, projection method with sparse grad-div stabilization and ensemble-based projection method with sparse grad-div stabilization . . . . 40

Figure 2.3 Velocity streamlines . . . . . . . . . . . . . . . . . . . . . . . . 45

Figure 3.1 Time evolution of the rank of searching space in BFBCG . . . . . 63

Figure 3.2 Time evolution of the rank of RHS matrix using the BGMRES-D solver . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66

Figure 3.3 Speed field at $t = 60$ for $j = 1, 20, \text{ and } 40$ in ensemble simulations. 68

Figure 3.4 Speed field at $t = 60$ for $j = 1, 20, \text{ and } 40$ in individual simulations. 68

Figure 3.5 Time evolution of velocity magnitude for $j = 1, 20, \text{ and } 40$ in ensemble simulations. . . . . . . . . . . . . . . . . . . . . 69

Figure 3.6 Time evolution of velocity magnitude for $j = 1, 20, \text{ and } 40$ in individual simulations. . . . . . . . . . . . . . . . . . . . . 69

Figure 3.7 Evolution of the rank of RHS matrix at the beginning of all time steps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

Figure 4.1 Error $\|u - u_h\|_{L^2}$ with different $\gamma$ and time step . . . . . . . . . . 72

Figure 4.2 Error $\|u - u_h\|_{H^1}$ with different $\gamma$ and time step . . . . . . . . . . 73
CHAPTER 1

INTRODUCTION

Dynamical systems play a significant role in modeling enormous phenomena such as heat transfer, fluid motions, chemically reacting flows or electromagnets. In many interdisciplinary application problems, the systems could become very complex due to the integration of different models, and numerical simulations have to be performed in order to quantify the behaviours of the dynamical systems. Because of the complexity of the systems, such simulations are usually of large scales. Furthermore, the applications of optimization, control and uncertainty quantification generally need repeated model evaluations over a certain range of parameter values. Performing many numerical simulations in large-scale settings leads to tremendous demands on computational resources.

For instance, in transonic airfoil shape design [6], sophisticated models were used in experiments, analyses, and computations of optimal designs. Quantative information was obtained by doing experiments and solving equations for a set of configurations and then comparing results. Even though the improvements of parallel computing as well as intensive research in ensemble-based data assimilation alleviate the burden of computation and storage, the current available computing power is still insufficient to perform high accuracy ensemble computations for applications with large spatial scales such as weather prediction. Thus developing a fast algorithm for computing PDE ensembles at a sufficiently fine spatial resolution is urgent and significant.
Therefore, the ensemble method has recently been introduced to alleviate the computational burden for solving a group of problems in [7, 16, 41, 14, 19, 23, 68]. It has been extended to an increasing number of fields, including turbulence [15], magnetohydrodynamic [67], natural convection [25, 26] and heat equation with uncertain conductivity [24]. We will review the method in this chapter.

The organization of the rest of this chapter is as follows: Section 1.1 provides some notations and preliminaries. Section 1.2 presents the ensemble method. The final section posts an outline of this dissertation.

1.1 Notations and preliminaries

Let $\Omega$ be an open, regular domain in $\mathbb{R}^d$, $d = 2$ or $3$ having boundary denoted by $\partial \Omega$. The $L^2(\Omega)$ norm and inner product are denoted by $\| \cdot \|$ and $(\cdot, \cdot)$, respectively. The Sobolev space $W^k_2(\Omega)$ is simply denoted by $H^k(\Omega)$ and equipped with the norm denoted by $\| \cdot \|_k$. For function $v(x, t)$ defined on $(0, T)$, we have, for $1 \leq m < \infty$,

$$
\|v\|_{\infty,k} := \text{EssSup}_{0 \leq t \leq T} \|v(\cdot, t)\|_k \quad \text{and} \quad \|v\|_{m,k} := \left( \int_0^T \|v(\cdot, t)\|_{m,k}^m dt \right)^{1/m}.
$$

Given a time step $\Delta t$, let $N = \frac{T}{\Delta t}$, associated discrete norms are defined as

$$
\|v\|_{\infty,k} := \max_{0 \leq n \leq N} \|v^n\| \quad \text{and} \quad \|v\|_{m,k} := \left( \sum_{n=0}^N \|v^n\|_{m,k}^m \Delta t \right)^{1/m},
$$

where $v^n = v(t_n)$ and $t_n = n\Delta t$. Denoted by $H^{-1}(\Omega)$ the dual space of bounded linear functions on $H^1_0(\Omega) = \{v \in H^1 : v = 0 \text{ on } \partial \Omega\}$; a norm on $H^{-1}(\Omega)$ is given by

$$
\|f\|_{-1} := \sup_{0 \neq v \in H^1_0(\Omega)} \frac{(f, v)}{\|\nabla v\|}.
$$

We also denote the temporal approximate and finite element (FE) approximate solutions at $t = t^n$ as $u^n_j$ and $u^n_{j,h}$, respectively.
1.2 The ensemble method

Here we take Navier-Stokes equation (NSE) as an example, which models incompressible fluid flows. In fluid mechanics, incompressible flow refers to a flow in which the material density is constant within a fluid parcel (an infinitesimal volume that moves with the flow velocity). An equivalent mathematical statement that implies incompressibility is that the divergence of the flow velocity is zero.

Considering the following $J$ NSE problems. For each $j = 1, \cdots, J$

\[
\begin{align*}
\frac{\partial u_j}{\partial t} + (u_j \cdot \nabla)u_j - \nu \Delta u_j + \nabla p_j &= f_j, \\
\nabla \cdot u_j &= 0, \\
u_j(x,0) = u_j^0(x) &\text{ in } \Omega \quad \text{and} \quad u_j(x,t) = 0 \text{ on } \partial \Omega.
\end{align*}
\]

(1.1)

Different initial conditions and body forces might be used in these problems. Using an implicit-explicit time discretization and suppressing the spacial discretization leads to the following scheme:

\[
\begin{align*}
\frac{u_j^{n+1} - u_j^n}{\Delta t} + u_j^n \cdot \nabla u_j^{n+1} - \nu \Delta u_j^{n+1} + \nabla p_j^{n+1} &= f_j^{n+1}, \\
\nabla \cdot u_j^{n+1} &= 0.
\end{align*}
\]

(1.2)

It can be proved that the scheme is first order accurate in time.

Rewriting (1.2) in a matrix form yields

\[
\begin{bmatrix}
\frac{1}{\Delta t} + u_j^n \cdot \nabla - \nu \Delta \\
\nabla \cdot
\end{bmatrix}
\begin{bmatrix}
u_j^{n+1} \\
p_j^{n+1}
\end{bmatrix} =
\begin{bmatrix}
f_j^{n+1} + \frac{1}{\Delta t} u_j^n
\end{bmatrix}.
\]

(1.3)

We observe that the coefficient matrices of the linear systems in (1.3) depend on $j$, which means for $J$ problems, we need to solve $J$ linear systems. When a high resolution mesh is used for spatial discretization, solving $J$ such systems is time consuming, especially when $J$ is large.
Therefore, Jiang and Layton proposed the following ensemble algorithms in [16, 41] to speed up a flow ensemble simulation. The essential idea is to make all ensemble members share the same coefficient matrix at each time step.

### 1.2.1 The first and second order time-stepping schemes for simulating ensembles of flow problems

In [16], the j-independent coefficient matrix of an ensemble member is acquired by adopting the following time discretization:

\[
\begin{align*}
\frac{u_{j}^{n+1} - u_{j}^{n}}{\Delta t} + \bar{u}^{n} \cdot \nabla u_{j}^{n+1} + u_{j}^{n} \cdot \nabla u_{j}^{n} - \nu \Delta u_{j}^{n+1} + \nabla p_{j}^{n+1} &= f_{j}^{n+1}, \\
\nabla \cdot u_{j}^{n+1} &= 0,
\end{align*}
\]

(1.4)

where the ensemble mean \( \bar{u}^{n} \) is the average velocity at time instance \( t_{n} \)

\[
\bar{u}^{n} = \frac{1}{J} \sum_{j=1}^{J} u_{j}^{n}
\]

(1.5)

and the fluctuation \( u_{j}^{n} \) at time instance \( t_{n} \) is defined by

\[
u_{j}^{n} = u_{j}^{n} - \bar{u}^{n} .
\]

(1.6)

It is seen that the nonlinear term is split into two by introducing the average velocity. One is independent of the index of ensemble members, the other contains the fluctuation that characterizes each realization. The one with the fluctuation is lagged to previous time level, and goes to the right-hand side of the linear systems.

In a matrix form, we have

\[
\begin{bmatrix}
\frac{1}{\Delta t} + \bar{u}^{n} \cdot \nabla - \nu \Delta \\
\nabla . \\
0
\end{bmatrix}
\begin{bmatrix}
\begin{array}{c}
\nabla u_{j}^{n+1} \\
p_{j}^{n+1}
\end{array}
\end{bmatrix}
= \begin{bmatrix}
f_{j}^{n+1} + \frac{1}{\Delta t} u_{j}^{n} - u_{j}^{n} \cdot \nabla u_{j}^{n} \\
0
\end{bmatrix} .
\]

(1.7)

The above scheme involves an explicit discretization of a stretching term, which results in a time step restriction in order to guarantee the long time, nonlinear stability. If a finite element discretization use a mesh of size \( h \), the scheme is stable
if
\[ C \frac{\Delta t}{\nu h} \| \nabla u^n_{j,h} \|^2 \leq 1 \quad \text{for} \quad j = 1, \cdots, J. \] (1.8)

Then we have the following stability and convergence results.

**Theorem 1.** *(Stability)* Consider the method (1.4), suppose the time step condition (1.8) holds. Then for any \( N > 1 \)
\[
\frac{1}{2} \| u^N_{j,h} \|^2 + \frac{1}{4} \sum_{n=0}^{N-1} \| u^{n+1}_{j,h} - u^n_{j,h} \|^2 + \frac{\nu \Delta t}{4} \| \nabla u^N_{j,h} \|^2 + \frac{\nu \Delta t}{4} \sum_{n=0}^{N-1} \| \nabla u^{n+1}_{j,h} \|^2 \\
\leq \frac{\Delta t}{2\nu} \| f^{n+1}_{j} \|^{2} + \frac{1}{2} \| u^0_{j,h} \|^2 + \frac{\nu \Delta t}{4} \| \nabla u^0_{j,h} \|^2. \] (1.9)

**Theorem 2.** *(Convergence)* Consider the time stepping (1.4), suppose the time step condition (1.8) holds and the \( P2 - P1 \) Taylor-Hood FE pair is used for the spatial discretization. Assume that \( \| u^0_j - u^0_{j,h} \|, \| \nabla (u^0_j - u^0_{j,h}) \| \) are both \( O(h) \) accurate or better, then for any \( N > 1 \), there is a positive constant \( C \) independent of the mesh size and time step such that
\[
\| u^N_j - u^N_{j,h} \|^2 + C \nu \Delta t \| \nabla (u^N_j - u^N_{j,h}) \|^2 \sim O(h^2 + \Delta t^2 + h \Delta t). \] (1.10)

The extension of the ensemble method to a higher order is nontrivial. A first step was taken in [41] where a second order ensemble method was developed by making use of a special combination of a second order backward difference formula and an explicit second order Adams-Bashforth treatment of the advection term. The scheme uses the following time discretization:
\[
\begin{aligned}
\frac{3u^n_{j+1} - 4u^n_j + u^n_{j-1}}{2\Delta t} + \bar{u}^n \cdot \nabla u^n_{j+1} + u^n_j \cdot \nabla (2u^n_j - u^n_{j-1}) - \nu \Delta u^{n+1}_j + \nabla p^{n+1}_j = f^{n+1}_j, \\
\nabla \cdot u^{n+1}_j = 0,
\end{aligned} \] (1.11)

where the ensemble mean \( \bar{u}^n \) at time instance \( t_n \) is defined by
\[
\bar{u}^n = \frac{1}{J} \sum_{j=1}^{J} (2u^n_j - u^n_{j-1}) \] (1.12)
and the fluctuation $u^{n}_{j}$ at time instance $t_{n}$ is defined by

$$u^{n}_{j} = 2u^{n}_{j} - u^{n-1}_{j} - \bar{u}^{n}.$$  \tag{1.13}$$

Putting it in matrix form, we have

$$\begin{bmatrix} \frac{4}{2\Delta x} + \bar{u} \cdot \nabla - \nu \Delta \nabla \\ \nabla \end{bmatrix} \begin{bmatrix} u^{n+1}_{j} \\ p^{n+1}_{j} \end{bmatrix} = \begin{bmatrix} f^{n+1}_{j} + \frac{3}{32\Delta t} (4u^{n}_{j} - u^{n-1}_{j}) - u^{n}_{j} \cdot \nabla (2u^{n}_{j} - u^{n-1}_{j}) \end{bmatrix}.  \tag{1.14}$$

The corresponding stability and convergence results are as follows.

**Theorem 3. (Stability)** Consider the method (1.11) and suppose the time step condition (1.8) holds. Then for any $N > 1$,

$$\begin{align*}
\frac{1}{4} \left\| u^{N}_{j,h} \right\|^{2} + \frac{1}{4} \left\| 2u^{N}_{j,h} - u^{N-1}_{j,h} \right\|^{2} + \frac{1}{8} \sum_{n=1}^{N-1} \left\| u^{n+1}_{j,h} - 2u^{n}_{j,h} + u^{n-1}_{j,h} \right\|^{2} + \frac{\nu \Delta t}{4} \sum_{n=1}^{N-1} \left\| \nabla u_{j,h}^{n+1} \right\|^{2} \\
\leq \sum_{n=1}^{N-1} \frac{\Delta t}{\nu} \left\| f^{n+1}_{j,h} \right\|^{2} - 1 + \frac{1}{4} \left\| u^{1}_{j,h} \right\|^{2} + \frac{1}{4} \left\| 2u^{1}_{j,h} - u^{0}_{j,h} \right\|^{2}.
\end{align*}$$

\[ \tag{1.15} \]

**Theorem 4. (Convergence)** Consider the second order ensemble time-stepping (1.11). Suppose the time step condition (1.8) holds and the $P2 - P1$ Taylor-Hood FE pair is used for the spatial discretization. Assume that $\left\| u^{0}_{j} - u^{0}_{j,h} \right\|$, $\left\| \nabla (u^{0}_{j} - u^{0}_{j,h}) \right\|$, $\left\| u^{1}_{j} - u^{1}_{j,h} \right\|$, $\left\| \nabla (u^{1}_{j} - u^{1}_{j,h}) \right\|$ are all $O(h^2)$ accurate or better, then for any $N > 2$, there is a positive constant $C$ independent of the mesh size and time step such that

$$\left\| u^{N}_{j} - u^{N}_{j,h} \right\|^{2} + C\nu \Delta t \left\| \nabla (u^{N}_{j} - u^{N}_{j,h}) \right\|^{2} \sim O(h^4 + \Delta t^4 + h \Delta t^3).$$

\[ \tag{1.16} \]

Another second order ensemble method with improved accuracy was presented in [14], which adopts a blended three-step backward differentiation formula time-stepping:

$$\begin{align*}
\frac{10u^{n+1}_{j} - 15u^{n}_{j} + 6u^{n-1}_{j} - u^{n-2}_{j}}{6\Delta t} + \bar{u} \cdot \nabla u^{n+1}_{j} + u^{n}_{j} \cdot \nabla (3u^{n}_{j} - 3u^{n-1}_{j} + u^{n-2}_{j}) \\
- \nu \Delta u^{n+1}_{j} + \nabla p^{n+1}_{j} = f^{n+1}_{j},
\end{align*}$$

$$\nabla \cdot u^{n+1}_{j} = 0,$$

\[ \tag{1.17} \]
where the ensemble mean $\bar{u}^n$ at time instance $t_n$ is defined by

$$\bar{u}^n = \frac{1}{J} \sum_{j=1}^{J} (3u_j^n - 3u_j^{n-1} + u_j^{n-2})$$  \hspace{1cm} (1.18)

and the fluctuation $u_j^n$ at time instance $t_n$ is defined by

$$u_j^n = 3u_j^n - 3u_j^{n-1} + u_j^{n-2} - \bar{u}^n.$$  \hspace{1cm} (1.19)

It can be rewritten as

$$f_j^{n+1} + \frac{1}{6\Delta t} \left( 15u_j^n - 6u_j^{n-1} + u_j^{n-2} \right) - u_j^n \cdot \nabla (3u_j^n - 3u_j^{n-1} + u_j^{n-2}) \right] \hspace{1cm} (1.20)$$

This time-stepping scheme is long time, nonlinearly stable under a CFL-like time step condition (1.8). It has the following stability and convergence results.

**Theorem 5.** *(Stability)* Consider the method (1.17) and suppose the time step condition (1.8) holds. Then for any $N > 2$

$$\frac{1}{12} \left\| u_{j,h}^N \right\|^2 + \frac{1}{12} \left\| 3u_{j,h}^N - u_{j,h}^{N-1} \right\|^2 + \frac{1}{12} \left\| 3u_{j,h}^N - 3u_{j,h}^{N-1} + u_{j,h}^{N-2} \right\|^2$$

$$+ \frac{1}{24} \sum_{n=2}^{N-1} \left\| u_{j,h}^{n+1} - 3u_{j,h}^{n-1} + 3u_{j,h}^{n-2} - u_{j,h}^{n-1} \right\|^2 + \frac{\nu \Delta t}{4} \sum_{n=2}^{N-1} \left\| \nabla u_{j,h}^{n+1} \right\|^2$$

$$\leq \sum_{n=2}^{N-1} \frac{\Delta t}{\nu} \left( f_{j,n}^{n+1} \right|_{-1}^2 + \frac{1}{12} \left\| u_{j,h}^2 \right\|^2 + \frac{1}{12} \left\| 3u_{j,h}^2 - u_{j,h}^1 \right\|^2 + \frac{1}{12} \left\| 3u_{j,h}^2 - 3u_{j,h}^1 + u_{j,h}^0 \right\|^2.$$

\hspace{1cm} (1.21)

**Theorem 6.** *(Convergence)* Consider the second order ensemble time-stepping (1.17).

Suppose the time step condition (1.8) holds and the $P2 - P1$ Taylor-Hood FE pair is used for the spatial discretization. Assume that $\left\| 3u_j^2 - u_j^2 \right\|_h - 3(u_j^1 - u_j^1_h) + (u_j^0 - u_j^0_h)$, $\left\| \nabla (u_j^0 - u_j^0_h) \right\|_h$, $\left\| 3(u_j^2 - u_j^2_h) - (u_j^1 - u_j^1_h) \right\|_h$, $\left\| \nabla (u_j^1 - u_j^1_h) \right\|_h$, $\left\| u_j^2 - u_j^2_h \right\|_h$, $\left\| \nabla (u_j^2 - u_j^2_h) \right\|_h$ are all $O(h^2)$ accurate or better, then for any $N > 2$, there is a positive constant $C$ independent of the mesh size and time step such that

$$\left\| u_j^N - u_j^{N,h} \right\|^2 + C \nu \Delta t \left\| \nabla (u_j^N - u_j^{N,h}) \right\|^2 \sim O(h^4 + \Delta t^4).$$  \hspace{1cm} (1.22)
1.2.2 Ensemble methods for high Reynolds number flows

The ensemble methods presented in [16, 41, 14] require a CFL-like time step condition

$$C \frac{\Delta t}{\nu h} ||\nabla u_{j,h}^m||^2 \leq 1 \quad \text{for } j = 1, \cdots, J$$

to ensure the stability and convergence. This time step condition begins mild but degrades quickly as Renoylds number increases and fluctuations grow. To relax this restriction, two ensemble eddy viscosity numerical regularization methods are proposed in [17]. They stabilize the system by adding extra numerical dissipation parameterized by mixing length and kinetic energy in fluctuations.

The eddy viscosity models contain an additional additive eddy viscosity (EV) term $-\nabla \cdot (\nu_T \nabla u_j)$ that uses the Kolmogorov-Prandtl relation:

$$\nu_T = \nu_T(l, \kappa') = \mu l \sqrt{\kappa'},$$

where $\mu$ is a constant, $l$ is the characteristic length scale of fluctuations and $\kappa'$ is the kinetic energy of the fluctuations or any dimensionally consistent relation involving the same variables.

For the ensemble algorithm, the fluctuations about the mean can be directly computed rather than modelled. Accordingly, we take

$$\kappa' = \sum_{j=1}^{J} \frac{1}{2} |u'_j|^2 := \frac{1}{2} |u'|^2,$$

where the fluctuation $u'_j$ is

$$u'_j = u_j - \frac{1}{J} \sum_{j=1}^{J} u_j = u_j - \bar{u}, \quad j = 1, \cdots, J.$$

For the characteristic length scale there are two natural and dimensionally correct choices:

$$l_1 = \Delta x, \quad \text{after a spacial discretization,}$$

$$l_2 = |u'| \Delta t, \quad \text{for the considered time discretization.}$$
The second relation, \( l_2 = |u'| \Delta t \), expresses the characteristic length of turbulent fluctuations, which is the distance they travel in one time step. We shall thus consider the parametrizations induced by these two length scales:

\[
EEV1 : \nu_T = \mu_1 \Delta x |u'|, \\
EEV2 : \nu_T = \mu_2 |u'|^2 \Delta t.
\]

The following time discretization is adopted:

\[
\begin{aligned}
&u_j^{n+1} - u_j^n \Delta t + \bar{u} \cdot \nabla u_j^{n+1} + u_j^n \cdot \nabla u_j^n - (\nu + \nu_T^n) \Delta u_j^{n+1} + \nabla p_j^{n+1} = f_j^{n+1}, \\
\nabla \cdot u_j^{n+1} = 0.
\end{aligned}
\]

(1.23)

Then we have the following stability results.

**Theorem 7.** (Stability) Consider the method (1.23) with EEV1, \( \nu_T = \mu_1 \Delta x |u'| \). A sufficient condition for stability is that if for some \( \theta \), \( 0 \leq \theta \leq 1 \), the two time step conditions hold,

\[
(1 - \theta) \nu - \frac{C}{2} \Delta t \left\| \nabla \cdot u_j^n \right\|_{L^4}^2 \geq 0, \quad \frac{\Delta t |u'(x, t_n)|}{\Delta x} \leq \frac{1}{2} \mu_1 + \frac{1}{2} \sqrt{\mu_1^2 + \frac{\theta \nu \Delta t}{\Delta x^2}}.
\]

This is implied by the two special cases

\[
(1 - \theta) \nu - \frac{C}{2} \Delta t \left\| \nabla \cdot u_j^n \right\|_{L^4}^2 \geq 0, \quad \frac{\Delta t |u'(x, t_n)|}{\Delta x} \leq \frac{1}{2} \mu_1,
\]

or \( (1 - \theta) \nu - \frac{C}{2} \Delta t \left\| \nabla \cdot u_j^n \right\|_{L^4}^2 \geq 0, \quad \frac{\Delta t |u'(x, t_n)|}{\theta \nu} \leq \frac{1}{4} \).

**Theorem 8.** (Stability) The method (1.23) with EEV2, \( \nu_T = \mu_2 \Delta t |u'|^2 \), is nonlinearly, long time stable if, for some \( \theta \) and \( \alpha \), \( 0 \leq \theta \leq 1 \), \( 0 < \alpha < 1 \), the two time step conditions hold,

\[
\theta \nu + 2 \Delta t (\nu_2 - \frac{1}{2\alpha}) |u_j^n| \geq 0, \quad (1 - \theta) \nu - \frac{C}{4(1 - \alpha)} \Delta t \left\| \nabla \cdot u_j^n \right\|_{L^4}^2 \geq 0.
\]

In particular, stability follows if

\[
\nabla \cdot u_j^n = 0, \quad \mu_2 > \frac{1}{2}.
\]
The new condition depends on the size of $\nabla \cdot u_{j,h}'$ which emerges from the nonlinear term. Note that both EEV terms are nonnegative, thus stability holds under the condition (1.8) for the laminar case where $\nu_T = 0$. Numerical tests indicate that EEV1 and EEV2 models are more stable with appropriate choose of $\mu_1$ and $\mu_2$ respectively comparing to the standard NSE model. They also show that EEV2’s result in a test case is more stable and less likely to over-diffuse the flow comparing to EEV1.

A time relaxation model obtained by adding a linear time regularization term was studied in [18]. That time relaxation regularization penalizes the deviation of the fluctuations from the ensemble average:

$$\begin{cases}
\frac{u_{j}^{n+1} - u_{j}^{n}}{\Delta t} + \bar{u} \cdot \nabla u_{j}^{n+1} + u_{j}^{n} \cdot \nabla u_{j}^{n} - \nu \Delta u_{j}^{n+1} + \chi(u_{j}^{n+1} - \bar{u}^{n}) + \nabla p_{j}^{n+1} = f_{j}^{n+1}, \\
\nabla \cdot u_{j}^{n+1} = 0.
\end{cases}$$

(1.24)

A natural choice of $\chi$ in (1.24) is $O\left(\frac{1}{\Delta t}\right)$. This proposed time relaxation model has dissipative effect on the fluctuations of realizations and is easily implementable in existing computer platforms.

Remark 1. The continuous ensemble system based on the algorithm (1.24) is:

$$\begin{cases}
\frac{\partial u_{j}}{\partial t} + u_{j} \cdot \nabla u_{j} - \nu \Delta u_{j} + \chi(u_{j} - \bar{u}) + \nabla p_{j} = f_{j}, \\
\nabla \cdot u_{j} = 0.
\end{cases}$$

(1.25)

Taking the average of (1.25), the term $u_{j} - \bar{u}$ vanishes and gives

$$\begin{cases}
\frac{1}{J} \sum_{j=1}^{J} \frac{\partial u_{j}}{\partial t} + \frac{1}{J} \sum_{j=1}^{J} u_{j} \cdot \nabla u_{j} - \frac{1}{J} \sum_{j=1}^{J} \nu \Delta u_{j} + \frac{1}{J} \sum_{j=1}^{J} \nabla p_{j} = \frac{1}{J} \sum_{j=1}^{J} f_{j}, \\
\frac{1}{J} \sum_{j=1}^{J} \nabla \cdot u_{j} = 0,
\end{cases}$$

(1.26)

which is exactly the ensemble averaged NSE. Thus, even though (1.26) contains no time relaxation term, the mean velocity $\bar{u}$ regularized implicitly via (1.25), which in turn regularizes each ensemble member.
Let $\chi = \frac{C_\chi}{\Delta t}$ for some $C_\chi = \mathcal{O}(1) \geq 0$, assume $(u_{j,h}^{n+1}, p_{j,h}^{n+1}) \in (X_h, Q_h)$ for $n = 0, 1, \cdots, N-1$, the authors established stability and convergence results under a time step condition:

$$
\frac{4\Delta t}{\nu} \left( \| u_j^{n+1} \|_\infty^2 + \| \nabla \cdot u_j^{n+1} \|_\infty^2 \frac{\text{diam}(\Omega)^2}{d^2} \right) \leq 1, \quad j = 1, \cdots, J, \quad (1.27)
$$

where $d = 2, 3$, is the space dimension.

**Theorem 9. (Stability)** Consider the method (1.24), and suppose the time step condition (1.27) holds. Then for any $N > 1$

$$
\| u_h^N \|^2 + C_\chi \langle u_h^N \rangle^2 + \frac{\nu \Delta t}{2} \sum_{n=0}^{N-1} \| \nabla u_h^{n+1} \|^2 + C_\chi \sum_{n=0}^{N-1} (2 \| u_h^{n+1} \|^2 + \| \langle u_h^{n+1} \rangle - \langle u_h^n \rangle \|^2 )
\leq \frac{\Delta t}{\nu} \sum_{n=0}^{N-1} \| f_{n+1} \|_n^2 + \| u_h^0 \|^2 + C_\chi \| \langle u_h^0 \rangle \|^2,
$$

(1.28)

where $\| u_h^i \|^2 = \frac{1}{J} \sum_{j=1}^J \| u_{j,h}^i \|^2$ and $\| \nabla u_h^i \|^2 = \frac{1}{J} \sum_{j=1}^J \| \nabla u_{j,h}^i \|^2$ for any superscript index $i$.

**Theorem 10. (Convergence)** Consider the method (1.24) and suppose the time step condition (1.27) holds. Let $(X_h, Q_h) = (P_{k+1}, P_k), k \geq 1$ be a Taylor-Hood pair. Then for any $N > 1$

$$
\| e^N \|^2 + \nu \Delta t \sum_{n=1}^{N-1} \| \nabla e^{n+1} \|^2 + C_\chi \sum_{n=0}^{N-1} \| e^{n+1} \|^2 + C_\chi \| \langle e^N \rangle \|^2
\leq C(\nu, T) \left( h^{2k} + \Delta t^2 + C_\chi \sum_{n=0}^{N-1} \| u^{n+1} \|^2 \right),
$$

(1.29)

where $\| e^i \|^2 = \frac{1}{J} \sum_{j=1}^J \| e_{j,h}^i \|^2$ for any superscript index $i$.

Numerical experiments in [18] indicate that grad-div stabilization can further weaken the time step restriction significantly.
1.2.3 The first and second order time-stepping schemes for simulating ensembles of parameterized flow problems

In the previous work, ensemble algorithms for flow with fixed parameters are considered. [19, 23] extend these ensemble algorithms to parameterized flows. First and second order ensemble algorithms were developed.

Considering the following $J$ equations. For each $j = 1, \cdots, J$

\[
\begin{cases}
    u_{j,t} + (u_j \cdot \nabla)u_j - \nu_j \Delta u_j + \nabla p_j = f_j, \\
    \nabla \cdot u_j = 0, \\
    u_j(x,0) = u_0^j(x) \text{ in } \Omega \text{ and } u_j(x,t) = 0 \text{ on } \partial \Omega.
\end{cases}
\]

(1.30)

The first order ensemble-based scheme is as follows.

\[
\begin{cases}
    \frac{u_j^{n+1} - u_j^n}{\Delta t} + \bar{u}^n \cdot \nabla u_j^{n+1} + u_j^n \cdot \nabla u_j^n - \bar{\nu} \Delta u_j^{n+1} - \nu'_j \Delta u_j^n + \nabla p_j^{n+1} = f_j^{n+1}, \\
    \nabla \cdot u_j^{n+1} = 0,
\end{cases}
\]

(1.31)

where the ensemble mean $\bar{u}^n$ and the fluctuation $u_j^n$ are defined as (1.5) and (1.6), and $\bar{\nu}$ and $\nu'$ are defined as:

\[
\bar{\nu} = \frac{1}{J} \sum_{j=1}^{J} \nu_j,
\]

(1.32)

\[
\nu'_j = \nu_j - \bar{\nu}.
\]

(1.33)

The authors established the following stability and convergence results:

Theorem 11. (Stability) For all $j = 1, \cdots, J$, if for some $\mu$, $0 \leq \mu < 1$, and some $\epsilon$, $0 < \epsilon \leq 2 - 2\sqrt{\mu}$, the following time step condition and parameter deviation condition hold:

\[
C \frac{\Delta t}{\bar{\nu} h} ||\nabla u_{j,h}||^2 \leq \frac{(2 - 2\sqrt{\mu} - \epsilon)\sqrt{\mu}}{2(\sqrt{\mu} + \epsilon)},
\]

(1.34)

\[
\frac{|\nu_j - \bar{\nu}|}{\bar{\nu}} \leq \sqrt{\mu}.
\]

(1.35)
Then the scheme (1.30) is nonlinearly, long time stable. In particular, for $j = 1, \ldots, J$ and for any $N > 1$, we have

$$
\frac{1}{2} \| u_{j,h}^N \|^2 + \frac{1}{4} \sum_{n=0}^{N-1} \left\| u_{j,h}^{n+1} - u_{j,h}^n \right\|^2 + \bar{\nu} \Delta t \left( \frac{\sqrt{\mu} + \epsilon}{2 \sqrt{\mu} + \epsilon} - \frac{\left| \nu_j - \bar{\nu} \right|^{2 \bar{\nu}}} \right) \left\| \nabla u_{j,h}^N \right\|^2
\leq \sum_{n=0}^{N-1} \frac{\Delta t}{2 \bar{\nu}} \left\| f_{j}^{n+1} \right\|^2 - \frac{1}{2} \left\| u_{j,h}^0 \right\|^2 + \bar{\nu} \Delta t \left( \frac{\sqrt{\mu} + \epsilon}{2 \sqrt{\mu} + \epsilon} - \frac{\left| \nu_j - \bar{\nu} \right|^{2 \bar{\nu}}} \right) \left\| \nabla u_{j,h}^0 \right\|^2.
$$

(1.36)

**Theorem 12.** (Convergence) Consider the ensemble time stepping (1.31). Suppose the time step condition (1.34) and parameter deviation condition (1.35) hold for some $\mu, 0 \leq \mu < 1$, some $\epsilon, 0 < \epsilon \leq 2 - 2\sqrt{\mu}$ and all $j$, for $j = 1, \ldots, J$. Assume that $\| u_j^0 - u_{j,h}^0 \|, \| \nabla (u_j^0 - u_{j,h}^0) \|$ are both $O(h)$ accurate or better, then if $(X_h, Q_h)$ is chosen as the $(P2, P1)$ Taylor-Hood FE pairs, there is a positive constant $C$ independent of the mesh size and time step such that for any $N > 1$,

$$
\| u_j^N - u_{j,h}^N \|^2 + C \bar{\nu} \Delta t \left\| \nabla (u_j^N - u_{j,h}^N) \right\|^2 \sim O(h^2 + \Delta t^2 + h \Delta t).
$$

(1.37)

The satisfaction of a parameter deviation condition is a prerequisite to obtain the long time stability of the algorithm above. By replacing the maximum value of the viscosity coefficients $\nu_{\max}$ rather than the average result in a superior stability condition, this numerical discretization is stable without the requirement of (1.35), however (1.34) is still needed.

The second order time stepping scheme for simulating ensembles of parameterized flow problems is the following:

$$
\begin{cases}
\frac{3u_{j}^{n+1} - 4u_{j}^{n} + u_{j}^{n-1}}{2\Delta t} + \bar{w}^n \cdot \nabla u_{j}^{n+1} + u_j^n \cdot \nabla (2u_{j}^{n} - u_{j}^{n-1}) - \bar{\nu} \Delta u_{j}^{n+1} = f_j^{n+1}, \\
\nabla \cdot u_{j}^{n+1} = 0,
\end{cases}
$$

(1.38)

where the ensemble mean $\bar{w}^n$ and the fluctuation $u_j^n$ are defined as (1.12) and (1.13), and $\bar{\nu}$ and $u_j^n$ are defined as (1.33) and (1.32).
The authors established the following stability and convergence results:

**Theorem 13.** (Stability) For all \( j = 1, \cdots, J \), if for some \( \mu \), \( 0 \leq \mu < 1 \), and some \( \epsilon \), \( 0 < \epsilon \leq 2 - 2\sqrt{\mu} \), the following time step condition and parameter deviation condition hold:

\[
C \frac{\Delta t}{\bar{\nu} h} \| \nabla u_{j,h}^n \|^2 \leq \frac{(2 - 2\sqrt{\mu} - \epsilon)\sqrt{\mu}}{2(\sqrt{\mu} + \epsilon)},
\]

(1.39)

\[
\left| \nu_j - \bar{\nu} \right| \leq \frac{\sqrt{\mu}}{3},
\]

(1.40)

where \( C \) denotes a generic constant depending on the domain and the minimum angle of the mesh. Then the scheme (1.38) is nonlinearly, long time stable. In particular, for \( j = 1, \cdots, J \) and for any \( N > 1 \), we have

\[
\frac{1}{4} \left( \| u_{j,h}^N \|^2 + \| 2u_{j,h}^N - u_{j,h}^{N-1} \|^2 \right) + \frac{1}{8} \sum_{n=1}^{N-1} \left( \| u_{j,h}^{n+1} - 2u_{j,h}^n + u_{j,h}^{n-1} \|^2 \right)
\]

\[
+ \bar{\nu} \frac{\Delta t}{\sqrt{\mu} + \epsilon} \left( \frac{\sqrt{\mu}}{2} \frac{2 + \epsilon}{\sqrt{\mu} + \epsilon} - \frac{3|\nu_j - \bar{\nu}|}{2\bar{\nu}} \right) \| \nabla u_{j,h}^N \|^2
\]

\[
\leq \sum_{n=1}^{N-1} \frac{\sqrt{\mu} + \epsilon}{2(2 - \sqrt{\mu})} \frac{\Delta t}{\bar{\nu}} \left( \| f_{j,h}^{n+1} \|_{-1}^2 + \frac{1}{4} (\| u_{j,h}^1 \|^2 + \| 2u_{j,h} - u_{j,h}^0 \|^2) \right)
\]

\[
+ \bar{\nu} \frac{\Delta t}{\sqrt{\mu} + \epsilon} \left( \frac{\sqrt{\mu}}{2} \frac{2 + \epsilon}{\sqrt{\mu} + \epsilon} - \frac{3|\nu_j - \bar{\nu}|}{2\bar{\nu}} \right) \| \nabla u_{j,h}^1 \|^2.
\]

(1.41)

**Theorem 14.** (Convergence) Consider the ensemble time stepping (1.31). Suppose the time step condition (1.39) and parameter deviation condition (1.40) hold for some \( \mu \), \( 0 \leq \mu < 1 \), some \( \epsilon \), \( 0 < \epsilon \leq 2 - 2\sqrt{\mu} \) and all \( j, j = 1, \cdots, J \). Assume that \( \| u_j^0 - u_{j,h}^0 \|, \| \nabla (u_j^0 - u_{j,h}^0) \|, \| u_j^1 - u_{j,h}^1 \|, \| \nabla (u_j^1 - u_{j,h}^1) \| \) are all \( \mathcal{O}(h^2) \) accurate or better, then if \((X_h, Q_h)\) is chosen as the \((P_2, P_1)\) Taylor-Hood FE pairs, there is a positive constant \( C \) independent of the mesh size and time step such that for any \( N > 1 \),

\[
\| u_j^N - u_{j,h}^N \|^2 + C\bar{\nu} \Delta t \| \nabla (u_j^N - u_{j,h}^N) \|^2 \sim \mathcal{O}(h^4 + \Delta t^4 + h\Delta t^2).
\]

(1.42)
1.2.4 Ensemble methods for solving PDE with random coefficients

In [68], the ensemble algorithm was extended to equations with random coefficients. The authors considered a group of numerical solutions to second order parabolic PDE, they first developed an ensemble algorithm for deterministic problems, in which the physical parameter, i.e. diffusion coefficient, is a function in space and time; and then extended it to stochastic problems. This is the first time that an ensemble scheme is derived for problems with non-constant parameters and is further applied to PDE with random coefficients.

Considering the following $J$ equations. For each $j = 1, \ldots, J$

\[
\begin{cases}
    u_{j,t} - \nabla \cdot [a_j(x,t)\nabla u_j] = f_j(x,t) & \text{in } D \times [0, T], \\
    u_j(x,t) = g_j(x,t) & \text{on } \partial D \times [0, T], \\
    u_j(x,0) = u_j^0(x) & \text{on } x \in D.
\end{cases}
\] (1.43)

The ensemble-based time stepping scheme reads, for $j = 1, \ldots, J$,

\[
\frac{u_{j}^{n+1} - u_{j}^{n}}{\Delta t} - \nabla \cdot (\bar{a}^{n+1}\nabla u_{j}^{n+1}) = f_{j}^{n+1} + \nabla \cdot [(a_{j}^{n+1} - \bar{a}^{n+1})\nabla u_{j}^{n}]
\] (1.44)

where the ensemble mean of the diffusion coefficient functions at time $t_n$ is

\[
\bar{a}^{n} := \frac{1}{J} \sum_{j=1}^{J} a_j(x, t_n).
\] (1.45)

It is obvious that the coefficient matrix of the resulting linear system will be independent with $j$ after spatial discretization. For more general parabolic equations such as those with random coefficients, the ensemble method is still applicable by combining with ensemble-based sampling approaches.

The following equation is to be considered:

\[
\begin{cases}
    u_t(\omega,x,t) - \nabla \cdot [a(\omega,x)\nabla u(\omega,x,t)] = f(\omega,x,t) & \text{in } \Omega \times D \times [0, T], \\
    u(\omega,x,t) = 0 & \text{on } \Omega \times \partial D \times [0, T], \\
    u(\omega,x,0) = u_0 & \text{on } \Omega \times \partial D.
\end{cases}
\] (1.46)
The goal is to find a random function \( u : \Omega \times \bar{D} \times [0,T] \to \mathbb{R}^d \) satisfying (1.46) almost surely, where \( D \) is a bounded Lipschitz domain in \( \mathbb{R}^d \) and \((\Omega,\mathcal{F}, P)\) is a probability space with sample space \( \Omega \), \( \sigma \)-algebra \( \mathcal{F} \), and probability measure \( P \), diffusion coefficient \( a : \Omega \times D \to \mathbb{R} \) and source term \( f : \Omega \times D \times [0,T] \to \mathbb{R} \) are random fields with continuous and bounded covariance functions. Suppose the diffusion coefficient \( a(\omega, x) \) in (1.46) is a small random perturbation of some deterministic diffusion coefficient function \( a_0(x) \) such that

\[
a(\omega, \cdot) := a_0(\cdot) + \epsilon \eta(\omega, \cdot),
\]

(1.47)

where \( a_0 \in W^{1,\infty}(D) \) is the diffusion of some deterministic background medium and \( \epsilon \eta \) represents a random fluctuation with the magnitude \( \epsilon \). It is required that \( a(\omega, x) \) is uniformly coercive. That is, there exists a positive constant \( \theta \) such that

\[
P\{ \omega \in \Omega : \min_{x \in D} a(\omega, x) > \theta \} = 1.
\]

The authors used the Monte Carlo method for random sampling because it is nonintrusive, easy to implement and its convergence is independent of the dimension of the uncertain model parameters. When applying the Monte Carlo method, we randomly select a large number of samples first, then implement a group of independent simulations in order to quantify the underlying stochastic information of the problem. This process leads to high computational cost. The authors proposed an ensemble-based Monte Carlo (EMC) method for the uncertainty quantification purpose to improve its computational efficiency. The method consists of the following steps:

- Choose a random sample of size \( J \) for the random medium coefficient and source term: \( a_j \equiv a(\omega_j, \cdot) = a_0(\cdot) + \epsilon_j \eta(\omega_j, \cdot), f_j \equiv f(\omega_j, \cdot, \cdot) \) for \( j = 1, \ldots, J \). Note that the corresponding solutions \( u(\omega_j, x, t) \) are independent, identically distributed because random variables \( \omega_j \) are independent, identically distributed;
Choose a finite element space $V_h$ and a uniform time partition with the step size $\Delta t = T/N$. Denote $u_{j,h}^n = u_h(\omega_j, x, t_n)$. For the $j$-th ensemble member and for $n = 0, \cdots, N - 1$, find an approximation solution $u_{j,h}^{n+1}$ such that

$$
\left( \frac{u_{j,h}^{n+1} - u_{j,h}^n}{\Delta t}, v_h \right) + (a_0 \nabla u_{j,h}^{n+1}, \nabla v_h) = (\epsilon_j \eta(\omega_j, \cdot) \nabla u_j^n, \nabla v_h), \quad \forall v_h \in V_h;
$$

(1.48)

Given a quantity of interest $g(u)$, one analyzes the ensemble simulation outputs $g(u_h(\omega_1, \cdot, \cdot)), \cdots, g(u_h(\omega_J, \cdot, \cdot))$ to exact its stochastic information. For instance, its expected value $E(g(u))$ is approximated by the sample average

$$
\frac{1}{J} \sum_{j=1}^J g(u_h(\omega_j, \cdot, \cdot)).
$$

Comparing with the ensemble scheme developed in (1.44) for the deterministic case, $a_0$ can be regarded as the ensemble average in (1.45). In general, we can choose $a_0 = E(a)$. The following are stability and convergence results.

Theorem 15. (Stability) Suppose $f_j \in \tilde{L}^2(H^{-1}(D); 0, T)$ and $u_{j,h}^0 \in V_h$. Then for any $\Delta t > 0$, the solution to (1.48) satisfies

$$
E \left[ \|u_{j,h}^N\|^2 \right] + \theta_- \Delta t E \left[ \|\nabla u_{j,h}^N\|^2 \right] + (\theta - \theta_+) \Delta t \sum_{n=1}^N E \left[ \|\nabla u_{j,h}^n\|^2 \right] \leq C \Delta t \sum_{n=1}^N E \left[ \|f_j^n\|^2 \right] + C \Delta t E \left[ \|\nabla u_{j,h}^0\|^2 \right] + E \left[ \|u_{j,h}^0\|^2 \right]
$$

(1.49)

provided that

$$
P\{\omega \in \Omega : 0 < \theta_- \leq \|\epsilon_j \eta(\omega_j, \cdot)\|_{L^\infty(D)} \leq \theta_+ \} = 1 \quad \text{and} \quad \theta > \theta_+,
$$

where $C$ is a generic constant independent of $J, h$ and $\Delta t$.

Theorem 16. (Convergence) Let $u_j^n$ be the solution to equation (1.46) when $\omega = \omega_j$ and $t = t_n$, and $u_{j,h}^n$ be the solution to (1.48). Then for $u_0(\omega_j, x) \in \tilde{L}^2(H_0^1(D) \cap H^{l+1}(D)), f_j(\omega_j, x, t) \in \tilde{L}^2(H^{-1}(D); 0, T)$, there exists a generic constant $C$ indepen-
dent of $J$, $h$ and $\Delta t$ such that

$$E \left[ \| u_j^N - u_{j,h}^N \|^2 \right] + \left( \theta - \frac{1 + \delta}{1 - \delta} \theta_+ \right) \Delta t \sum_{n=1}^{N} E \left[ \| \nabla (u_j^n - u_{j,h}^n) \|^2 \right]$$

$$+ \Delta t \theta \sum_{n=1}^{N} E \left[ \| \nabla (u_j^n - u_{j,h}^n) \|^2 \right] \leq C (\Delta t^2 + h^2)$$

providing that

$$\theta - \frac{1 + \delta}{1 - \delta} \theta_+ > 0,$$

where $0 < \delta < 1$.

A new multilevel Monte Carlo ensemble method was developed for solving random parabolic partial differential equations in the same direction in [4]. The ensemble-based multilevel Monte Carlo (EMLMC) applied to (1.46) solves the following group of simulations at the $\ell$-th level: for $j = 0, \ldots, J_\ell$ ($J_\ell$ is the number of selected samples), given $u_{n-1,j,\ell}$ and $u_{n,j,\ell}$, to find $u_{n+1,j,\ell} \in V_\ell^g$ such that,

$$\left( \frac{3u_{n+1,j,\ell} - 4u_{n,j,\ell} + u_{n-1,j,\ell}}{2\Delta t_\ell}, v_\ell \right) + (\bar{a}_\ell \nabla u_{n+1,j,\ell}, \nabla v_\ell) = - \left( (a_j - \bar{a}_\ell) \nabla (2u_{n,j,\ell} - u_{n-1,j,\ell}), \nabla v_\ell \right)$$

$$+ (f_{n+1,j}, v_\ell), \quad \forall v_\ell \in V_\ell^0,$$

for $n = 0, \ldots, N_\ell - 1$ where $u_{n,j,\ell} = u_\ell(\omega_j, x, t_n)$ is the finite element approximation of $u(\omega_j, x, t_n)$ at the $\ell$-th level, $V_\ell^0$ is the test function space at the $\ell$-th level and the ensemble mean of the diffusion coefficient function $\bar{a}_\ell = \frac{1}{J_\ell} \sum_{j=1}^{J_\ell} a(\omega_j, x)$.

Once the numerical solutions at all the $L$ levels are found, the EMLMC approximates the stochastic PDE solutions at the time instance $t_n$:

$$E[u_L(\omega, x, t_n)] = E[\sum_{\ell=1}^{L} (u_\ell(\omega, x, t_n) - u_{\ell-1}(\omega, x, t_n)) + u_0(\omega, x, t_n)]$$

$$= \sum_{\ell=1}^{L} E[(u_\ell(\omega, x, t_n) - u_{\ell-1}(\omega, x, t_n))] + E[u_0(\omega, x, t_n)].$$

Numerically, the expected value of $E[u_\ell(\omega, x, t_n)]$ is approximated by the sampling average $\Psi_{J_\ell}[u_\ell(\omega, x, t_n)] = \frac{1}{J_\ell} \sum_{j=1}^{J_\ell} u_\ell(\omega, x, t_n)$. Meanwhile, given a quantity of interest $Q(u)$, one can analyze the outputs from the ensemble simulations, $Q(u_h(\omega_1, \cdot, \cdot)), \ldots, Q(u_h(\omega_J, \cdot, \cdot))$, for extracting the underlying stochastic information of the system.
Assume the exact solution of (1.46) is smooth enough, in particular,

\[ u_j \in \tilde{L}^2(H^1_0(D) \cap H^{m+1}(D); 0, T) \cap \tilde{H}^1(H^{m+1}(D); 0, T) \cap \tilde{H}^2(L^2(D); 0, T) \]

and suppose

\[ f_j \in \tilde{L}^2(H^{-1}(D); 0, T). \]

Assume the following two conditions hold:

(i) There exists a positive constant \( \theta \) such that

\[ P\{ \omega \in \Omega : \min_{x \in D} a(\omega, x) > \theta \} = 1. \]

(ii) There exists a positive constant \( \theta_+ \), for \( \ell = 0, \cdots, L \), such that

\[ P\{ \omega_j \in \Omega : |a(\omega_j, x) - \bar{a}_\ell|_\infty \leq \theta_+ \} = 1. \]

The condition (i) guarantees the uniform coercivity a.s. and condition (ii) gives an upper bound of the distance from coefficient \( a(\omega_j, x) \) to the ensemble average \( \bar{a} \) a.s.

**Theorem 17. (Stability)** Under conditions (i) and (ii), the scheme (1.51) is stable provided that

\[ \theta > 3\theta_. \]  \hspace{1cm} (1.53)

Furthermore, the numerical solution to (1.51) satisfies

\[
\begin{align*}
\frac{1}{4} E \left[ \|u_{j,\ell}^N\|^2 \right] + \frac{1}{4} E \left[ \|2u_{j,\ell}^N - u_{j,\ell}^{N-1}\|^2 \right] &+ \frac{\theta}{2} \Delta t_\ell E \left[ \|\nabla u_{j,\ell}^N\|^2 \right] \\
+ (\frac{\theta}{3} - \theta_+) \Delta t_\ell \sum_{n=1}^{N-1} E \left[ \|\nabla u_{j,\ell}^n\|^2 \right] &\leq \frac{\Delta t_\ell}{2(\theta - 3\theta_+)} \sum_{n=1}^{N-1} E \left[ \|f_j^n\|_1^2 \right] + \frac{1}{4} E \left[ \|u_{j,\ell}^1\|^2 \right] + \frac{1}{4} E \left[ \|u_{j,\ell}^0\|^2 \right] + \frac{1}{4} E \left[ \|2u_{j,\ell}^1 - u_{j,\ell}^0\|^2 \right].
\end{align*}
\]  \hspace{1cm} (1.54)
Theorem 18. (Convergence) Suppose conditions (i) and (ii) and the stability condition (1.53) hold, then the EMLMC approximation error satisfies

\[
\frac{1}{4}E \left[ \|E[u(t_{N_L})] - \Psi[u_{L}(t_{N_L})]\|^2 \right] + \frac{1}{4}E \left[ \|E[u^{NL}] - \Psi[u_{L}(t_{N_L})] - (E[u^{NL-1}] - \Psi[u_{L}(t_{N_L-1})])\|^2 \right] \\
- \Psi[u_{L}(t_{N_L-1})]\|^2 \right] + \left( \frac{\theta}{3} - \theta_+ \right) \Delta t_L \sum_{n=1}^{N_L} E \left[ \|\nabla E[u(t_n)] - \nabla \Psi[u_{L}(t_n)]\|^2 \right]
\leq C(h_L^2m + \Delta t_L^4 + \sum_{\ell=1}^{L} \frac{1}{J_\ell}(h_\ell^{2m} + \Delta t_\ell^4)) + \frac{C}{J_0}(\Delta t_0 \sum_{n=1}^{N_0} E \left[ \|f^n_{j,-1}\|^2 \right])
+ \Delta t_0 E \left[ \|\nabla u^1_{j,0}\|^2 + \|\nabla u^0_{j,0}\|^2 \right] + E \left[ \|u^1_{j,0}\|^2 + \|2u^1_{j,0} - u^0_{j,0}\|^2 \right],
\]  

(1.55)

where \(C > 0\) is a constant independent of \(J_\ell, \Delta t_\ell\) and \(h_\ell\).

1.2.5 Other applications

Ensemble algorithms, which could efficiently reduce computing cost, have been widely adopted in an increasing number of fields. It is applied in the proper orthogonal decomposition [9, 10, 13, 12, 11, 8] setting in [20, 21, 22]. In [15], the authors combined the approach of ensemble time stepping and ensemble eddy viscosity modeling that gives an unconditionally stable algorithm. In [67], MHD systems were decoupled into Oseen problems by using an Elsasser variable formulation and ensemble algorithm. In [25, 26], first order and second order ensemble time stepping algorithms for laminar natural convection problems were provided. Ensemble algorithms for heat equation with uncertain conductivity were considered in [24]. In [5], the authors proposed and analyzed an efficient ensemble algorithm for fast computation of multiple realizations of the stochastic Stokes-Darcy model with a random hydraulic conductivity tensor.

1.3 Outline of dissertation

In this dissertation, we develop an ensemble-based projection method to efficiently find numerical solutions to a group of the Navier-Stokes equations. Particularly, members in the group are subject to different viscosity coefficients, initial conditions,
and body forces. The ensemble method solves a single linear system with multiple right-hand sides for the entire group, and thus is computationally more efficient than individual simulations. It is well known that projection methods could lead to numerical boundary layer in pressure and intermediate velocity because of the artificial Neumann boundary condition. Thus a sparse grad-div stabilization is considered in our scheme. The rest of this dissertation is organized as follows. In Chapter 2, we propose an efficient and accurate scheme to solve a group of NSE based on the ensemble algorithm and the sparse grad-div stabilization. In Chapter 3, we study the numerical behavior of ensemble methods combining with block iterative algorithms on a set of evolution problems including heat conduction and incompressible fluid dynamics applications. Some future work is presented in last chapter.
When conducting numerical simulations of incompressible flows, one of the main difficulties is that the velocity and pressure are coupled by the incompressibility constraint. The projection methods, originally developed by Chorin and Temam [33, 42], are designed to overcome this difficulty in time-dependent viscous incompressible flows. These methods lead to more attractive linear systems that solve a sequence of decoupled elliptic equations for the velocity and pressure at each time step. Although the projection methods have been widely used because of their efficiency and simplicity [39, 40, 35, 43, 44, 45], some drawbacks exist. In these methods, some artificial Neumann boundary conditions are enforced on pressure that induce numerical boundary layers and limit the accuracy of the schemes. Thus, compared to coupled methods, projection methods have irreducible splitting error.

Recently, a grad-div stabilization was developed for solving coupled Stokes and Navier-Stokes equations (NSE) [34, 3, 37, 38, 36, 28, 32, 27]. It is considered as a constant term added to finite element discretizations that decreases divergence error of the velocity and reduces the effect of the pressure error on the velocity. As we know, the grad-div operator produces fully coupled matrices for velocity, which often makes solving the resulting linear algebraic systems more difficult. In [30], a new sparse grad-div stabilization was proposed that has a similar positive effect on error
as the grad-div stabilization, but has a more efficient implementation, because its corresponding matrix has a sparser block structure.

In order to accelerate the solutions of a group of time-dependent incompressible flows, we propose to combine the projection method with ensemble method. Furthermore, we add a sparse grad-div stabilization term in step 1 of the ensemble-based projection method to overcome the inaccuracy caused by the artificial Neumann boundary conditions.

The rest of this chapter is organized as follows. In Section 2.1, we briefly introduce the projection method. In Section 2.2, we present some notations and mathematical preliminaries. In Section 2.3, we introduce ensemble-based projection method and sparse grad-div stabilization. In Section 2.4, we analyze the proposed algorithm and prove its stability and convergence. Numerical experiments are presented in Section 2.5, which illustrate the effectiveness of the proposed scheme. Finally, this chapter is summarized in the last section.

2.1 The projection method

The projection method was originally introduced by Alexandre Chorin in 1967 as an effective way of numerically solving the incompressible NSE. The key feature of the projection method is that the computations of velocity and pressure are decoupled at each time step, which makes the method very efficient especially for large-scale numerical simulations. Over the years the projection method has played a dominant role in the computation of viscous incompressible flow. It can be viewed as fractional/splitting step method, but the usual methodology developed for fractional step method does not apply directly. We review several time discretization schemes in this section. Since the nonlinear term in NSE does not affect the convergence rate of the splitting error, we only consider the time-dependent Stokes equation here:
\[
\begin{aligned}
\frac{\partial u}{\partial t} - \nu \Delta u + \nabla p &= f \quad \text{in} \quad \Omega \times [0, T], \\
\nabla \cdot u &= 0 \quad \text{in} \quad \Omega \times [0, T], \\
u(x, 0) = u_0(x) \quad \text{in} \quad \Omega \quad \text{and} \quad u(x, t) = 0 \quad \text{on} \quad \partial \Omega.
\end{aligned}
\] (2.1)

All the results stated in this section are applicable to the NSE if sufficient regularity of the solution holds.

2.1.1 The pressure-correction scheme and the velocity-correction scheme

The simplest pressure-correction scheme has originally been proposed by Chorin and Temam [33, 42]. Using the implicit Euler time stepping, the algorithm is as follows:

Set \( u^0 = u_0 \), then for \( k \geq 0 \) compute \((\bar{u}^{k+1}, u^{k+1}, p^{k+1})\) by solving

\[
\frac{1}{\Delta t}(\bar{u}^{k+1} - u^k) - \nu \nabla^2 \bar{u}^{k+1} = f(t^{k+1}), \quad \bar{u}^{k+1}|_\Gamma = 0, \quad (2.2)
\]

\[
\begin{aligned}
\frac{1}{\Delta t}(u^{k+1} - \bar{u}^{k+1}) + \nabla p^{k+1} &= 0, \\
\nabla \cdot u^{k+1} = 0, \quad u^{k+1} \cdot n|_\Gamma &= 0. \quad (2.3)
\end{aligned}
\]

It has the following error estimate.

**Theorem 19.** Let \((u, p)\) be a smooth solution of (2.1), let \((\bar{u}_{\Delta t}, p_{\Delta t})\) and \((u_{\Delta t}, p_{\Delta t})\) be the numerical solutions for the semi-discrete projection method (2.2) and (2.3) respectively. Then we have

\[
\|u - u_{\Delta t}\|_{L^\infty([L^2(\Omega)]^d)} + \|u - \bar{u}_{\Delta t}\|_{L^\infty([L^2(\Omega)]^d)} \leq c(u, p, T)\Delta t, \\
\|p - p_{\Delta t}\|_{L^\infty([L^2(\Omega)])} + \|u - \bar{u}_{\Delta t}\|_{L^\infty([H^1(\Omega)]^d)} \leq c(u, p, T)\Delta t^{1/2}
\]

where

\[
\|v_{\Delta t}\|_{L^\infty(E)} := \max_{0 \leq n \leq N} (\|v^n\|_E) \quad \text{for} \quad 0 \leq n \leq N = T/\Delta t.
\]
From (2.3), we observe that \( \nabla p^{k+1} \cdot n|_\Gamma = 0 \) is enforced on the pressure. This artificial Neumann boundary condition induces a numerical boundary layer and consequently prevents the scheme to be fully temporal first order on the velocity in \( H^1 \) norm and on the pressure in \( L^2 \) norm.

The main idea of the velocity-correction scheme is to switch the role of the velocity and pressure in the pressure-correction scheme, in other words, the viscous term is treated explicitly or ignored in the first step and the velocity is corrected in the second step.

Set \( \hat{u}^0 = u_0 \), for \( k \geq 0 \) compute \((\tilde{u}^{k+1}, u^{k+1}, p^{k+1})\) by solving

\[
\begin{cases}
\frac{1}{\Delta t} (u^{k+1} - \tilde{u}^k) + \nabla p^{k+1} = f(t^{k+1}), \\
\nabla \cdot u^{k+1} = 0, \quad u^{k+1} \cdot n|_{\Gamma} = 0.
\end{cases}
\tag{2.4}
\]

\[
\frac{1}{\Delta t} (\tilde{u}^{k+1} - u^{k+1}) - \nu \nabla^2 \tilde{u}^{k+1} = 0, \quad \tilde{u}^{k+1}|_{\Gamma} = 0.
\tag{2.5}
\]

This algorithm also suffers from the artificial Neumann boundary condition \( \nabla p^{k+1} \cdot n|_{\Gamma} = f(t^{k+1}) \cdot n \) and \( \nabla^2 \tilde{u}^{k+1} \cdot n|_{\Gamma} = 0 \), whereas the non-incremental pressure-correction scheme enforces \( \nabla p^{k+1} \cdot n|_{\Gamma} = 0 \) and \( \nu \nabla^2 \tilde{u}^{k+1} \cdot n|_{\Gamma} = f(t^{k+1}) \cdot n \). This scheme has equivalent results to the non-incremental pressure-correction scheme in terms of accuracy.

**Theorem 20.** Let \((u,p)\) be a smooth solution of (2.1), let \((\tilde{u}_{\Delta t}, p_{\Delta t})\) and \((u_{\Delta t}, p_{\Delta t})\) be the numerical solutions for the semi-discrete projection method (2.5) and (2.4) respectively. Then we have

\[
\|u - u_{\Delta t}\|_{L^\infty[L^2(\Omega)]} + \|u - \tilde{u}_{\Delta t}\|_{L^\infty[L^2(\Omega)]} \leq c(u,p,T)\Delta t,
\]

\[
\|p - p_{\Delta t}\|_{L^\infty[L^2(\Omega)]} + \|u - \tilde{u}_{\Delta t}\|_{H^1(\Omega)} \leq c(u,p,T)\Delta t^{1/2}.
\]

To overcome the difficulty caused by the artificial Neumann boundary condition, some temporal higher order projection methods were proposed. In the standard incremental pressure-correction scheme and the standard incremental velocity-correction
scheme, old value of the pressure gradient is added into the first step, and then accordingly the velocity in the second step increases the accuracy. In the rotational incremental pressure-correction scheme and the rotational incremental velocity-correction scheme, $-\nabla^2 u$ is replaced by $\nabla \times \nabla \times u$ to correct the artificial boundary condition, then a better pressure approximation can be obtained.

Another class of schemes, which is called consistent splitting method, are also referred to in the literature as projection method. It computes the velocity and the pressure in two consecutive steps: First, compute the velocity by treating the pressure explicitly, then update the pressure using $\int_\Omega \nabla p \cdot \nabla q = \int_\Omega (f + \nu \nabla^2 u) \cdot \nabla q, \forall q \in H^1(\Omega)$. Strictly speaking, the consistent splitting schemes are not projection scheme since the velocity approximation is not divergence free.

2.2 Notations and preliminaries

The natural function spaces for our method are defined by

$$X := H_0^1(\Omega) = \{ v \in H^1(\Omega), v |_{\partial \Omega} = 0 \},$$

$$Y := \{ v \in L^2(\Omega), v \cdot n |_{\partial \Omega} = 0 \},$$

$$Q := L_0^2(\Omega) = \{ q \in L^2(\Omega), \int_\Omega q = 0 \}.$$  

We assume the mesh and finite element spaces satisfy the standard inverse inequality

$$h \| \nabla v_h \| \leq C \| v_h \| \quad \forall v_h \in X_h,$$  

that is known to hold for standard finite element spaces with locally quasi-uniform mesh [29]. We also define the standard explicitly skew-symmetric trilinear form

$$b(u, v, w) := \frac{1}{2} (u \cdot \nabla v, w) - \frac{1}{2} (u \cdot \nabla w, v),$$
that satisfies the bounds [31]:

\begin{align}
  b(u, v, w) &\leq C(\|\nabla u\|\|u\|)^{1/2}\|\nabla v\|\|\nabla w\| \quad \forall \ u, v, w \in X, \\
  b(u, v, w) &\leq C(\|\nabla u\|\|\nabla v\|\|\nabla w\|\|w\|)^{1/2} \quad \forall \ u, v, w \in X, \\
  b(u, v, w) &\leq C\|u\||v|_2\|\nabla w\| \quad \forall \ u, w \in X, v \in X \cap (H^2(\Omega))^d, \\
  b(u, v, w) &\leq C\|\nabla u\|\|\nabla v\||\nabla w\| \quad \forall \ u, v, w \in X.
\end{align}

2.3 Ensemble-based projection method and sparse grad-div stabilization

In this section, we develop a non-incremental pressure-correction projection method for ensemble-based simulations of the NSE in which not only the initial data and body force function, but also the viscosity coefficient, may vary from one ensemble member to another. Specifically, we consider a set of \( J \) NSE problems defined as (1.1).

Given an initial guess \( \bar{u}_j^0 = u_0^n \), a time step \( 0 < \Delta t \leq T \), for \( n = 0, 1, \ldots, N - 1 (= T/\Delta t - 1) \),

**Step 1:** Find \( u_j^{n+1} \) such that

\begin{equation}
\frac{u_j^{n+1} - \bar{u}_j^n}{\Delta t} + \bar{u}^n \cdot \nabla u_j^{n+1} + (u_j^n - \bar{u}_j^n) \cdot \nabla u_j^n - \bar{\nu} \Delta u_j^{n+1} - (\nu_j - \bar{\nu}) \Delta u_j^n = f_j^{n+1},
\end{equation}

\begin{equation}
u_j^{n+1} \mid_{\partial \Omega} = 0.
\end{equation}

**Step 2:** Find \((\bar{u}_j^{n+1}, p_j^{n+1})\) such that

\begin{equation}
\frac{\bar{u}_j^{n+1} - u_j^n}{\Delta t} + \nabla p_j^{n+1} = 0,
\end{equation}

\begin{equation}\nabla \cdot \bar{u}_j^{n+1} = 0,
\end{equation}

\begin{equation}\bar{u}_j^{n+1} \cdot n \mid_{\partial \Omega} = 0,
\end{equation}

where \( \bar{u}^n \) and \( \bar{\nu} \) are the ensemble mean of velocity and viscosity coefficient respectively, defined as (1.5) and (1.32). The first step now leads to solve one linear system with
multiple right-hand sides, thus is computationally more efficient than solving for all simulations individually.

It is well known that the projection methods tend to be less accurate due to the splitting error, stabilizations of grad-div type can alleviate this problem by reducing the divergence error of the velocity and furthermore reduce the scaling of the velocity error and pressure error [27, 28]. It adds \(-\gamma \nabla (\nabla \cdot u)\) to the momentum equation of the NSE that appears as a nonzero penalty term in discretizations to enforce mass conservation. We apply the stabilization to Step 1 of the ensemble-based projection algorithm. To improve the efficiency, the sparse grad-div stabilization operator proposed in [29] will be applied.

**Definition 2.1.** The sparse grad-div stabilization (i.e., divergence penalization) operator \(g\) is defined by

\[
g_{2d}(u) = -\nabla (\nabla \cdot u) + \begin{pmatrix} - (u_2)_{xy} \\ (u_1)_{xy} \end{pmatrix}, \quad (2.17)
\]

\[
g_{3d}(u) = -\nabla (\nabla \cdot u) + \begin{pmatrix} - (u_2)_{xy} - (u_3)_{xz} \\ (u_1)_{xy} \\ (u_1)_{xz} \end{pmatrix}. \quad (2.18)
\]

**Lemma 2.2.** The operator \(g\) has the following properties: for \(u, v \in H_0^1(\Omega)^d\)

1. The inner product of \(g(u)\) with \(v\) can be written as

\[
(g_{2d}(u), v) = (\nabla \cdot u, \nabla \cdot v) - \left((u_{1x}, v_{2y}) - (u_{2y}, v_{1x})\right),
\]

\[
(g_{3d}(u), v) = (\nabla \cdot u, \nabla \cdot v) - \left((u_{1x}, v_{2y}) - (u_{2y}, v_{1x}) + (u_{1x}, v_{3z}) - (u_{3z}, v_{1x})\right). \quad (2.19, 2.20)
\]

2. \(g\) satisfies in 2d and 3d

\[
(g(u), u) = \|\nabla \cdot u\|^2. \quad (2.21)
\]
3. if $\nabla \cdot u = 0$, then in 2d and 3d

\[
\left( g(u), v \right) = -(u_{1x}, \nabla \cdot v).
\]  

Taking the sparse grad-div stabilization into consideration, we have the following temporal discretization. Given an initial guess $\tilde{u}_j^0 = u_j^0$, a time step $0 < \Delta t \leq T$, for $n = 0, 1, \ldots, N - 1 (= T / \Delta t - 1)$,

Step 1: Find $u_j^{n+1}$ such that

\[
\frac{u_j^{n+1} - \tilde{u}_j^n}{\Delta t} + \tilde{u}_j^n \cdot \nabla u_j^{n+1} + (u_j^n - \tilde{u}_j^n) \cdot \nabla u_j^n - \tilde{\nu} \Delta u_j^{n+1} - (\nu_j - \tilde{\nu}) \Delta u_j^n + \gamma g(u_j^{n+1}) = f_j^{n+1},
\]

\[
u_j^{n+1} |_{\partial \Omega} = 0.
\]  

Step 2: Find $(\tilde{u}_j^{n+1}, p_j^{n+1})$ which satisfies (2.14) - (2.16).

Next, we derive some numerical analysis for the proposed method.

2.4 Numerical analysis

We first discuss the stability of the above ensemble algorithm.

**Theorem 21.** Suppose $f_j \in L^2(H^{-1}(\Omega); 0, T)$ under the following conditions

\[
|\nu_j - \tilde{\nu}| \leq \tilde{\nu},
\]

\[
C \Delta t \left( \frac{1}{2} \right) \| \nabla (u_j^n - \tilde{u}^n) \|^2 \leq (\tilde{\nu} - |\nu_j - \tilde{\nu}|)^{\frac{3}{2}},
\]

the schemes (2.23)-(2.24) and (2.14) - (2.16) are stable. Furthermore, the numerical solution to the schemes satisfies

\[
\| u_j^N \|^2 + \tilde{\nu} \Delta t \| \nabla u_j^N \|^2 + \frac{\tilde{\nu} - |\nu_j - \tilde{\nu}|}{4} \Delta t \sum_{n=0}^{N-1} \| \nabla u_j^{n+1} \|^2 + 2 \gamma \Delta t \sum_{n=0}^{N-1} \| \nabla \cdot u_j^{n+1} \|^2 \leq \sum_{n=0}^{N-1} \frac{4 \Delta t}{\tilde{\nu} - |\nu_j - \tilde{\nu}|} \| f_j^{n+1} \|_1^2 + \| u_j^0 \|^2 + \tilde{\nu} \Delta t \| \nabla u_j^0 \|^2,
\]

where $C > 0$ is a constant independent of $\Delta t$. 

Proof. Multiplying $u_j^{n+1}$ and integrating over the domain in (2.23) gives

$$\frac{u_j^{n+1} - \tilde{u}_j^n}{\Delta t} + b((u_j^n - \tilde{u}_j^n), u_j^n, u_j^{n+1}) + (\nu_j - \tilde{\nu})(\nabla u_j^n, \nabla u_j^{n+1}) + \gamma(g(u_j^{n+1}), u_j^{n+1}) = (f_j^{n+1}, u_j^{n+1}).$$

(2.28)

Multiplying both sides by $\Delta t$, and using the polarization identity, we get

$$\frac{1}{2} \|u_j^{n+1}\|^2 - \frac{1}{2} \|\tilde{u}_j^n\|^2 + \frac{1}{2} \|u_j^{n+1} - \tilde{u}_j^n\|^2 + \Delta t \tilde{\nu}\|\nabla u_j^{n+1}\|^2 + \Delta t \gamma\|\nabla \cdot u_j^{n+1}\|^2
= -\Delta t b((u_j^n - \tilde{u}_j^n), u_j^n, u_j^{n+1}) - \Delta t (\nu_j - \tilde{\nu})(\nabla u_j^n, \nabla u_j^{n+1}) + \Delta t (f_j^{n+1}, u_j^{n+1}).$$

(2.29)

Next, multiplying $\tilde{u}_j^n$ and integrating over the domain in (2.14) - (2.16) gives

$$\frac{1}{2} \|\tilde{u}_j^n\|^2 - \frac{1}{2} \|u_j^n\|^2 + \frac{1}{2} \|\tilde{u}_j^n - u_j^n\|^2 = 0.$$

(2.30)

Adding above two equations, we obtain

$$\frac{1}{2} \|u_j^{n+1}\|^2 - \frac{1}{2} \|u_j^n\|^2 + \frac{1}{2} \|u_j^{n+1} - \tilde{u}_j^n\|^2 + \frac{1}{2} \|\tilde{u}_j^n - u_j^n\|^2 + \Delta t \tilde{\nu}\|\nabla u_j^{n+1}\|^2 + \Delta t \gamma\|\nabla \cdot u_j^{n+1}\|^2
= -\Delta t b((u_j^n - \tilde{u}_j^n), u_j^n, u_j^{n+1}) - \Delta t (\nu_j - \tilde{\nu})(\nabla u_j^n, \nabla u_j^{n+1}) + \Delta t (f_j^{n+1}, u_j^{n+1}).$$

(2.31)

We bound the trilinear term using the inequality and (2.9), obtaining

$$| - \Delta t b((u_j^n - \tilde{u}_j^n), u_j^n, u_j^{n+1})|
= | - \Delta t b((u_j^n - \tilde{u}_j^n), u_j^n, u_j^{n+1} - u_j^n)|
\leq C \Delta t \|\nabla (u_j^n - \tilde{u}_j^n)\| \|\nabla u_j^{n+1}\| \left(\|\nabla (u_j^{n+1} - u_j^n)\| \|u_j^{n+1} - u_j^n\|\right)^{1/2}
\leq C \Delta t^2 \beta_1^{-1/2} \|\nabla (u_j^n - \tilde{u}_j^n)\|^2 \|\nabla u_j^n\|^2 + (\beta_1)^{1/2} \|\nabla (u_j^{n+1} - u_j^n)\| \|u_j^{n+1} - u_j^n\|
\leq C \Delta t^2 \beta_1^{-1/2} \|\nabla (u_j^n - \tilde{u}_j^n)\|^2 \|\nabla u_j^n\|^2 + \frac{1}{4} \|u_j^{n+1} - u_j^n\|^2 + \beta_1 \|\nabla u_j^{n+1}\|^2 + \beta_1 \|\nabla u_j^n\|^2.$$

(2.32)
Using Young’s inequality gives

\[
| - \Delta t(\nu_j - \tilde{\nu})(\nabla u^n_j, \nabla u_j^{n+1}) | \leq \frac{\Delta t|\nu_j - \tilde{\nu}|}{2} \| \nabla u^n_j \|^2 + \frac{\Delta t|\nu_j - \tilde{\nu}|}{2} \| \nabla u_j^{n+1} \|^2. \tag{2.33}
\]

\[
|\Delta t(f_j^{n+1}, u_j^{n+1})| \leq \frac{\Delta t}{4\beta_2} \| f_j^{n+1} \|_{-1}^2 + \beta_2 \Delta t \| \nabla u_j^{n+1} \|^2. \tag{2.34}
\]

Substituting (2.32)-(2.34) into (2.29), we have

\[
\frac{1}{2} \| u_j^{n+1} \|^2 - \frac{1}{2} \| u_j^n \|^2 + \frac{1}{2} \| u_j^{n+1} - \bar{u}_j^n \|^2 + \frac{1}{2} \| \tilde{u}_j^n - u_j^n \|^2 - \frac{1}{4} \| u_j^{n+1} - u_j^n \|^2 + \Delta t\tilde{\nu} \| u_j^{n+1} \|^2 + \Delta t \| \nabla \cdot u_j^{n+1} \|^2
\]

\[
\leq \left( C\Delta t^2 \beta_1 \frac{1-\nu}{2} \| \nabla (u^n_j - \bar{u}^n) \|^2 + \frac{\Delta t|\nu_j - \tilde{\nu}|}{2} + \beta_1 \| \nabla u^n_j \|^2 \right)
\]

\[
+ \left( \frac{\Delta t|\nu_j - \tilde{\nu}|}{2} + \beta_2 \Delta t + \beta_1 \| \nabla u_j^{n+1} \|^2 + \frac{\Delta t}{4\beta_2} \| f_j^{n+1} \|_{-1}^2 \right).
\tag{2.35}
\]

Now we split the term \( \Delta t\tilde{\nu} \| \nabla u_j^{n+1} \|^2 \) into the following parts:

\[
\Delta t\tilde{\nu} \| \nabla u_j^{n+1} \|^2 = \frac{1}{2} \Delta t\tilde{\nu} \| \nabla u_j^{n+1} \|^2 + \frac{1}{2} \Delta t\tilde{\nu} (\| \nabla u_j^{n+1} \|^2 - \| \nabla u_j^n \|^2) + \frac{1}{2} \Delta t\tilde{\nu} \| \nabla u_j^n \|^2.
\tag{2.36}
\]

Using above equality in (2.35), we get

\[
\frac{1}{2} \| u_j^{n+1} \|^2 - \frac{1}{2} \| u_j^n \|^2 + \frac{1}{2} \| u_j^{n+1} - \bar{u}_j^n \|^2 + \frac{1}{2} \| \tilde{u}_j^n - u_j^n \|^2 - \frac{1}{4} \| u_j^{n+1} - u_j^n \|^2 + \Delta t\tilde{\nu} \| u_j^{n+1} \|^2 + \Delta t \| \nabla \cdot u_j^{n+1} \|^2
\]

\[
\leq \frac{\Delta t}{4\beta_2} \| f_j^{n+1} \|_{-1}^2.
\tag{2.37}
\]

Note that \( \frac{1}{2} \| u_j^{n+1} - \bar{u}_j^n \|^2 + \frac{1}{2} \| \tilde{u}_j^n - u_j^n \|^2 - \frac{1}{4} \| u_j^{n+1} - u_j^n \|^2 \geq 0 \). Dropping this positive term on the left hand side and multiplying both sides by 2 provides the estimate

\[
\left( \| u_j^{n+1} \|^2 - \| u_j^n \|^2 \right) + \Delta t\tilde{\nu} (\| \nabla u_j^{n+1} \|^2 - \| \nabla u_j^n \|^2) + \Delta t \left( \tilde{\nu} - |\nu_j - \tilde{\nu}| - 2\beta_2 - \frac{2\beta_1}{\Delta t} \right) \| \nabla u_j^{n+1} \|^2
\]

\[
+ 2\Delta t \| \nabla \cdot u_j^{n+1} \|^2 \leq \frac{\Delta t}{2\beta_2} \| f_j^{n+1} \|_{-1}^2.
\tag{2.38}
\]
Taking $\beta_1 = \frac{\Delta t (\bar{v} - |\nu_j - \bar{v}|)}{4}$ and $\beta_2 = \frac{\bar{v} - |\nu_j - \bar{v}|}{8}$, under the assumptions (2.25) and (2.26), we have

$$\bar{v} - |\nu_j - \bar{v}| - 2\beta_2 - \frac{2\beta_1}{\Delta t} = \frac{\bar{v} - |\nu_j - \bar{v}|}{4} > 0,$$

$$\bar{v} - C \Delta t \beta_1^{1/2} \|\nabla (u_j^n - \bar{u}^n)\|^2 - |\nu_j - \bar{v}| - 2\beta_1 \Delta t \geq 0.$$  

Therefore, assuming that (2.25) and (2.26) hold, (2.38) reduces to

$$\left(\|u_j^{n+1}\|^2 - \|u_j^n\|^2\right) + \Delta t \bar{v} \left(\|\nabla u_j^{n+1}\|^2 - \|\nabla u_j^n\|^2\right) + \frac{\bar{v} - |\nu_j - \bar{v}|}{4} \Delta t \|\nabla u_j^{n+1}\|^2
+ 2\Delta t \gamma \|\nabla \cdot u_j^{n+1}\|^2 \leq \frac{4 \Delta t}{\bar{v} - |\nu_j - \bar{v}|} \|f_j^{n+1}\|_1^2. \quad (2.39)$$

Summing (2.39) from $n = 0$ to $n = N - 1$, produces

$$\|u_j^N\|^2 + \bar{v} \Delta t \|\nabla u_j^N\|^2 + \frac{\bar{v} - |\nu_j - \bar{v}|}{4} \Delta t \sum_{n=0}^{N-1} \|\nabla u_j^{n+1}\|^2 + 2\gamma \Delta t \sum_{n=0}^{N-1} \|\nabla \cdot u_j^{n+1}\|^2
\leq \sum_{n=0}^{N-1} \frac{4 \Delta t}{\bar{v} - |\nu_j - \bar{v}|} \|f_j^{n+1}\|_1^2 + \|u_j^0\|^2 + \bar{v} \Delta t \|\nabla u_j^0\|^2. \quad (2.40)$$

This concludes the proof.

Next, we analyze the temporal error in the ensemble-based projection method with sparse grad-div stabilization.

**Theorem 22.** For $j = 1, 2, \ldots, J$, let $(u_j, p_j)$ be the exact solution of the NSE on $\Omega \times (0, T)$ for a given initial $u_j^0 \in X \cap V$ and forcing $f_j \in L^\infty(H^{-1}(\Omega); 0, T)$, with smoothness

$$u_j \in L^\infty(X \cap L^\infty(\Omega) \cap H^2(\Omega); 0, T), \quad u_{j,t} \in L^\infty(L^2(\Omega); 0, T),$$

$$u_{j,tt} \in L^\infty(L^2(\Omega); 0, T), \quad p_j \in L^\infty(H^1(\Omega); 0, T),$$

under the parameter deviation (2.25) and time-step conditions (2.26), the error in the ensemble-based projection method satisfies, $\forall \Delta t > 0,$

$$\|u_j(t^N) - u_j^N\|^2 + 2\gamma \Delta t \sum_{n=0}^{N-1} \|\nabla \cdot (u_j(t^{n+1}) - u_j^{n+1})\|^2
\leq e^{\frac{C}{\alpha_j}} \left(\left(\frac{|\nu_j - \bar{v}|}{\alpha_j}\Delta t^2 + \frac{C}{\alpha_j} \Delta t^{3/2}\right)\|\nabla u_{j,t}\|^2_{2,0} + \frac{C}{\alpha_j} \Delta t^2 \|u_{j,tt}\|^2_{2,0}
+ C \Delta t \|\gamma u_{j,1x} - p_j\|^2_{\infty,1}\right), \quad (2.41)$$
where $C > 0$ is a constant independent of $\Delta t$, $\alpha_j = \frac{\tilde{\nu} - |\nu_j - \tilde{\nu}|}{16}$.

**Proof.** Evaluating the equation (1.1) at $t = t_{n+1}$ and subtracting (2.23) gives

$$
\frac{e_j^{n+1} - \tilde{e}_j^n}{\Delta t} + u_j(t^{n+1}) \cdot \nabla u_j(t^{n+1}) - \tilde{u}^n \cdot \nabla u_j^{n+1} - (u_j^n - \tilde{u}^n) \cdot \nabla u_j^n + \gamma g(e_j^{n+1})
$$

$$
-\tilde{\nu} \Delta e_j^{n+1} - (\nu_j - \tilde{\nu}) \Delta e_j^n - (\nu_j - \tilde{\nu}) \Delta (u_j(t^{n+1}) - u_j(t^n)) + \nabla p_j(t^{n+1})
$$

$$
= \gamma g(u_j(t^{n+1})) + R_j(t^{n+1})
$$

(2.42)

where $e_j^n := u_j(t^n) - u_j^n$, $\tilde{e}_j^n := u_j(t^n) - \tilde{u}_j^n$, and $R_j(t^{n+1}) = \frac{u_j(t^{n+1}) - u_j(t^n)}{\Delta t} - u_{j,t}(t^{n+1})$.

Multiplying by $e_j^{n+1}$ and integrating over the domain gives

$$
\frac{1}{2\Delta t} (\|e_j^{n+1}\|^2 - \|e_j^n\|^2 + \|e_j^{n+1} - \tilde{e}_j^n\|^2) + \tilde{\nu}\|\nabla e_j^{n+1}\|^2 + \gamma \|\nabla \cdot e_j^{n+1}\|^2
$$

$$
= -b(u_j(t^{n+1}), u_j(t^{n+1}), e_j^{n+1}) + b(\tilde{u}^n, u_j^{n+1}, e_j^{n+1}) + b(u_j^n - \tilde{u}^n, u_j^n, e_j^{n+1})
$$

$$
-(\nu_j - \tilde{\nu})(\nabla e_j^n, \nabla e_j^{n+1}) - (\nu_j - \tilde{\nu})(\nabla u_j(t^{n+1}) - \nabla u_j(t^n), \nabla e_j^{n+1})
$$

$$
-(\nabla p_j(t^{n+1}), e_j^{n+1}) + \gamma (g(u_j(t^{n+1})), e_j^{n+1}) + (R_j(t^{n+1}), e_j^{n+1}).
$$

(2.43)

The equation (2.14) is equivalent to

$$
\frac{\tilde{e}_j^n - e_j^n}{\Delta t} = \nabla p_j^n.
$$

(2.44)

Multiplying by $\tilde{e}_j^n$ gives

$$
\|\tilde{e}_j^n\|^2 - \|e_j^n\|^2 + \|\tilde{e}_j^n - e_j^n\|^2 = 0.
$$

(2.45)

Adding (2.43) and (2.45) gives

$$
\frac{1}{2\Delta t} (\|e_j^{n+1}\|^2 - \|e_j^n\|^2 + \|e_j^{n+1} - \tilde{e}_j^n\|^2 + \|\tilde{e}_j^n - e_j^n\|^2)
$$

$$
+\tilde{\nu}\|\nabla e_j^{n+1}\|^2 + \gamma \|\nabla \cdot e_j^{n+1}\|^2
$$

$$
= -b(u_j(t^{n+1}), u_j(t^{n+1}), e_j^{n+1}) + b(\tilde{u}^n, u_j^{n+1}, e_j^{n+1}) + b(u_j^n - \tilde{u}^n, u_j^n, e_j^{n+1})
$$

$$
-(\nu_j - \tilde{\nu})(\nabla e_j^n, \nabla e_j^{n+1}) - (\nu_j - \tilde{\nu})(\nabla u_j(t^{n+1}) - \nabla u_j(t^n), \nabla e_j^{n+1})
$$

$$
-(\nabla p_j(t^{n+1}), e_j^{n+1}) + \gamma (g(u_j(t^{n+1})), e_j^{n+1}) + (R_j(t^{n+1}), e_j^{n+1}).
$$

(2.46)
We rearrange the nonlinear terms as follows.

\[-b(u_j(t^{n+1}), u_j(t^{n+1}), e_j^{n+1}) + b(\bar{u}^n, u_j^{n+1}, e_j^{n+1}) + b(u_j^n - \bar{u}^n, u_j^n, e_j^{n+1})\]
\[= -b(u_j(t^{n+1}), u_j(t^{n+1}), e_j^{n+1}) + b(u_j(t^n), u_j(t^{n+1}), e_j^{n+1})\]
\[-b(u_j(t^n), u_j(t^{n+1}), e_j^{n+1}) + b(u_j^n, u_j(t^{n+1}), e_j^{n+1}) - b(u_j^n, u_j(t^{n+1}), e_j^{n+1})\]
\[+ b(u_j^n, u_j^{n+1}, e_j^{n+1}) - b(u_j^n - \bar{u}^n, u_j^{n+1}, e_j^{n+1}) + b(u_j^n - \bar{u}^n, u_j^n, e_j^{n+1})\]
\[= -b(u_j(t^{n+1}) - u_j(t^n), u_j(t^{n+1}), e_j^{n+1}) - b(e_j^n, u_j(t^{n+1}), e_j^{n+1})\]
\[-b(u_j^n, e_j^{n+1}, e_j^{n+1}) - b(u_j^n - \bar{u}^n, u_j^{n+1} - u_j^n, e_j^{n+1})\]
\[= -b(u_j(t^{n+1}) - u_j(t^n), u_j(t^{n+1}), e_j^{n+1}) - b(e_j^n, u_j(t^{n+1}), e_j^{n+1})\]
\[+ b(u_j^n - \bar{u}^n, e_j^n, e_j^{n+1}) + b(u_j^n - \bar{u}^n, u_j(t^{n+1}) - u_j(t^n), e_j^{n+1}).\]

(2.47)

Using the inequality (2.8), Young’s inequality, and \(u_j \in L^\infty(0, T; H^1(\Omega))\), we get

\[-b(e_j^n, u_j(t^{n+1}), e_j^{n+1})\]
\[\leq C \|\nabla e_j^n\|^{1/2} \|e_j^n\|^{1/2} \|\nabla u_j(t^{n+1})\| \|\nabla e_j^{n+1}\|\]
\[\leq C \|\nabla e_j^n\|^{1/2} \|e_j^n\|^{1/2} \|\nabla e_j^{n+1}\|\]
\[\leq \frac{C}{\alpha_j} \|\nabla e_j^n\| \|e_j^n\| + \alpha_j \|\nabla e_j^{n+1}\|^2\]
\[\leq \frac{C}{\alpha_j} \|e_j^n\|^2 + \alpha_j \|\nabla e_j^n\|^2 + \alpha_j \|\nabla e_j^{n+1}\|^2\]

and

\[-b(u_j(t^{n+1}) - u_j(t^n), u_j(t^{n+1}), e_j^{n+1})\]
\[\leq C \|\nabla u_j(t^{n+1}) - \nabla u_j(t^n)\| \|\nabla u_j(t^{n+1})\| \|\nabla e_j^{n+1}\|\]
\[\leq \frac{C}{\alpha_j} \|\nabla u_j(t^{n+1}) - \nabla u_j(t^n)\|^2 + \alpha_j \|\nabla e_j^{n+1}\|^2\]
\[= \frac{C}{\alpha_j} \int_{t_n}^{t_{n+1}} \left\|\frac{d}{dt} u_j(t^n)\right\|^2 dt + \alpha_j \|\nabla e_j^{n+1}\|^2\]
\[\leq \frac{C}{\alpha_j} \Delta t \int_{t_n}^{t_{n+1}} \|\nabla u_{j,t}\|^2 dt + \alpha_j \|\nabla e_j^{n+1}\|^2;\]
We bound the third nonlinear term on the right hand side of (2.47) as follows.

\[ b(u^n_j - \bar{u}^n, e^n_j, e^{n+1}_j) \]
\[ = b((u^n_j - \bar{u}^n), e^n_j, e^{n+1}_j - e^n_j) \]
\[ \leq C\|\nabla(u^n_j - \bar{u}^n)\|\|\nabla e^n_j\| \left( \|\nabla(e^{n+1}_j - e^n_j)\|\|e^{n+1}_j - e^n_j\| \right)^{1/2} \]
\[ \leq \frac{C}{\alpha_j^{1/2}} \Delta t^{1/2}\|\nabla(u^n_j - \bar{u}^n)\|^2\|\nabla e^n_j\|^2 + \frac{\alpha_j}{8\Delta t}^1\|\nabla(e^{n+1}_j - e^n_j)\|\|e^{n+1}_j - e^n_j\|^2 \]
\[ \leq \frac{C}{\alpha_j^{1/2}} \Delta t^{1/2}\|\nabla(u^n_j - \bar{u}^n)\|^2\|\nabla e^n_j\|^2 + \frac{1}{4\Delta t}\|\nabla e^{n+1}_j - e^n_j\|^2 + \frac{\alpha_j}{2}\|\nabla(e^{n+1}_j - e^n_j)\|^2 \]
\[ \leq \frac{C}{\alpha_j^{1/2}} \Delta t^{1/2}\|\nabla(u^n_j - \bar{u}^n)\|^2\|\nabla e^n_j\|^2 + \frac{1}{4\Delta t}\|\nabla e^{n+1}_j - e^n_j\|^2 \]
\[ + \alpha_j\|\nabla e^{n+1}_j\|^2 + \alpha_j\|\nabla e^n_j\|^2. \]

For the last term of (2.47), we have

\[ b(u^n_j - \bar{u}^n, u_j(t^{n+1}) - u_j(t^n), e^{n+1}_j) \]
\[ \leq C\|\nabla(u^n_j - \bar{u}^n)\|\|\nabla(u_j(t^{n+1}) - u_j(t^n))\|\|\nabla e^{n+1}_j\| \]
\[ \leq \frac{C}{\alpha_j}\|\nabla(u_j(t^{n+1}) - u_j(t^n))\|^2\|\nabla e^{n+1}_j\|^2 \]
\[ \leq \frac{C}{\alpha_j}\Delta t \int_{t^n}^{t^{n+1}} \|\nabla u_{j,t}\|^2dt\|\nabla(u^n_j - \bar{u}^n)\|^2 + \alpha_j\|\nabla e^{n+1}_j\|^2. \]

Next we analyze the viscous terms on the right hand side of (2.43).

\[-(\nu_j - \bar{\nu})(\nabla u_j(t^{n+1}) - \nabla u_j(t^n), \nabla e^{n+1}_j) \]
\[ \leq \frac{|\nu_j - \bar{\nu}|^2}{\alpha_j}\|\nabla u_j(t^{n+1}) - \nabla u_j(t^n)\|^2 + \alpha_j\|\nabla e^{n+1}_j\|^2 \]
\[ \leq \frac{|\nu_j - \bar{\nu}|^2}{\alpha_j}\Delta t \int_{t^n}^{t^{n+1}} \|\nabla u_{j,t}\|^2dt + \alpha_j\|\nabla e^{n+1}_j\|^2. \]

and

\[-(\nu_j - \bar{\nu})(\nabla e^n_j, \nabla e^{n+1}_j) \leq \frac{|\nu_j - \bar{\nu}|}{2}\|\nabla e^n_j\|^2 + \frac{|\nu_j - \bar{\nu}|}{2}\|\nabla e^{n+1}_j\|^2. \]
Since $\nabla \cdot u_j = 0$, the second and third terms from last on the right hand side of (2.43) combine to form
\[
\gamma (g(u_j(t^{n+1})), e_j^{n+1}) - (\nabla p_j(t^{n+1}), e_j^{n+1})
\]
\[
= (\nabla (\gamma u_{j,1x}(t^{n+1}) - p_j(t^{n+1})), e_j^{n+1})
\]
\[
= (\nabla (\gamma u_{j,1x}(t^{n+1}) - p_j(t^{n+1})), e_j^{n+1} - \bar{e}_j^n)
\]
\[
\leq \Delta t \| \nabla (\gamma u_{j,1x}(t^{n+1}) - p_j(t^{n+1})) \|^2 + \frac{1}{4\Delta t} \| e_j^{n+1} - \bar{e}_j^n \|^2.
\]
The last term on the right hand side of (2.43) is bounded as
\[
(R_j(t^{n+1}), e_j^{n+1})
\]
\[
\leq C \| R_j(t^{n+1}) \| \| \nabla e_j^{n+1} \|
\]
\[
\leq \frac{C}{\alpha_j} \| R_j(t^{n+1}) \|^2 + \alpha_j \| \nabla e_j^{n+1} \|^2
\]
\[
\leq \frac{C}{\alpha_j} \Delta t \int_{t^n}^{t^{n+1}} \| u_{j,tt} \|^2 dt + \alpha_j \| \nabla e_j^{n+1} \|^2.
\]
Combining above estimates, we have
\[
\frac{1}{2\Delta t} (\| e_j^{n+1} \|^2 - \| e_j^n \|^2) + \frac{1}{4\Delta t} (\| e_j^{n+1} - \bar{e}_j^n \|^2 + \| \bar{e}_j^n - e_j^n \|^2 - \| e_j^{n+1} - e_j^n \|^2)
\]
\[
+ \frac{1}{4\Delta t} \| e_j^n - e_j^n \|^2 + \frac{5\tilde{\nu} - |\nu_j - \tilde{\nu}|}{8} (\| \nabla e_j^{n+1} \|^2 - \| \nabla e_j^n \|^2)
\]
\[
+ \left( \frac{\tilde{\nu} - |\nu_j - \tilde{\nu}|}{2} - \frac{C}{\alpha_j^{1/2}} \Delta t^{1/2} \| \nabla (u_j^n - \bar{u}_n) \|^2 \| \nabla e_j^n \|^2 + \gamma \| \nabla \cdot e_j^{n+1} \|^2 \right)
\]
\[
\leq \frac{C}{\alpha_j} \| e_j^n \|^2 + \frac{C}{\alpha_j} \Delta t \int_{t^n}^{t^{n+1}} \| \nabla u_{j,t} \|^2 dt + \frac{C}{\alpha_j} \Delta t \int_{t^n}^{t^{n+1}} \| \nabla u_{j,t} \|^2 dt \| \nabla (u_j^n - \bar{u}_n) \|^2
\]
\[
+ \frac{|\nu_j - \tilde{\nu}|}{\alpha_j} \Delta t \int_{t^n}^{t^{n+1}} \| \nabla u_{j,t} \|^2 dt + \Delta t \| \nabla (\gamma u_{j,1x}(t^{n+1}) - p_j(t^{n+1})) \|^2
\]
\[
+ \frac{C}{\alpha_j} \Delta t \int_{t^n}^{t^{n+1}} \| u_{j,tt} \|^2 dt.
\] (2.48)
Since
\[
\| e_j^{n+1} - \bar{e}_j^n \|^2 + \| \bar{e}_j^n - e_j^n \|^2 - \| e_j^{n+1} - e_j^n \|^2 \geq 0,
\]
\[
\frac{C}{\alpha_j^{3/2}} \Delta t^{1/2} \| \nabla (u_j^n - \bar{u}_n) \|^2 \leq 1,
\]

36
we have

\[
\frac{1}{2\Delta t}(\|e_{j}^{n+1}\|^2 - \|e_{j}^{n}\|^2) + \frac{1}{4\Delta t}\|\bar{e}_{j}^{n} - e_{j}^{n}\|^2 \\
+ \frac{5\tilde{\nu} - |\nu_{j} - \tilde{\nu}|}{8}\|(\nabla e_{j}^{n+1})\|^2 - \|(\nabla e_{j}^{n})\|^2) + \gamma \|\nabla \cdot e_{j}^{n+1}\|^2 \\
\leq \frac{C}{\alpha_j}\|e_{j}^{n}\|^2 + \frac{C}{\alpha_j} \int_{t_n}^{t_{n+1}} \|\nabla u_{j,t}\|^2dt + \frac{C}{\alpha_j} \Delta t \int_{t_n}^{t_{n+1}} \|\nabla u_{j,t}\|^2dt\|\nabla(u_{j}^{n} - \bar{u}_{j}^{n})\|^2 \\
+ \frac{|\nu_{j} - \tilde{\nu}|^2}{\alpha_j} \Delta t \int_{t_n}^{t_{n+1}} \|\nabla u_{j,t}\|^2dt + \Delta t \|\nabla(\gamma u_{j,1x}(t^{n+1}) - p_{j}(t^{n+1}))\|^2 \\
+ \frac{C}{\alpha_j} \Delta t \int_{t_n}^{t_{n+1}} \|u_{j,tt}\|^2dt. 
\]

(2.49)

Summing from \(n = 0\) to \(N - 1\) and multiplying both side by \(\Delta t\) gives

\[
\|e_{j}^{N}\|^2 + \frac{5\tilde{\nu} - |\nu_{j} - \tilde{\nu}|}{4} \Delta t\|\nabla e_{j}^{N}\|^2 + \frac{1}{2} \sum_{n=0}^{N-1} \|\bar{e}_{j}^{n} - e_{j}^{n}\|^2 + 2\gamma \Delta t \sum_{n=0}^{N-1} \|\nabla \cdot e_{j}^{n+1}\|^2 \\
\leq \frac{C}{\alpha_j^3} \Delta t \sum_{n=0}^{N-1} \|e_{j}^{n}\|^2 + (\frac{|\nu_{j} - \tilde{\nu}|^2}{\alpha_j} \Delta t^2 + \frac{C}{\alpha_j} \Delta t^{3/2})\|\nabla u_{j,t}\|^2_{2,0} + \frac{C}{\alpha_j} \Delta t^2 \|u_{j,tt}\|^2_{2,0} \\
+ C\Delta t \|\gamma u_{j,1x} - p_{j}\|^2_{\infty,1}. 
\]

(2.50)

Applying the discrete Gronwall inequality gives

\[
\|e_{j}^{N}\|^2 + 2\gamma \Delta t \sum_{n=0}^{N-1} \|\nabla \cdot e_{j}^{n+1}\|^2 \\
\leq \frac{C}{\alpha_j^3} \left((\frac{|\nu_{j} - \tilde{\nu}|^2}{\alpha_j} \Delta t^2 + \frac{C}{\alpha_j} \Delta t^{3/2})\|\nabla u_{j,t}\|^2_{2,0} + \frac{C}{\alpha_j} \Delta t^2 \|u_{j,tt}\|^2_{2,0} \\
+ C\Delta t \|\gamma u_{j,1x} - p_{j}\|^2_{\infty,1}\right). 
\]

(2.51)

This completes the proof.

\[\square\]

2.5 Numerical experiments

In this section three numerical tests are presented that confirm the accuracy and efficiency of the ensemble based projection method with sparse grad-div stabilization. The incompressible Navier-Stokes equation in a square and a cube are considered in Sections 2.5.1 and 2.5.2. In Section 2.5.1, velocity errors in \(L^2\) and \(H^1\) norm are
displayed to show the validity and accuracy of our algorithm for both problems. In Section 2.5.2, the efficiency of the ensemble based projection method with sparse grad-div stabilization is studied. We then test the proposed algorithm on a 2d channel flow over a step in Section 2.5.3. The accuracy and efficiency are shown by comparing to the independent projection method without stabilization.

2.5.1 Accuracy test

In the first numerical experiment, we compare the accuracy of projection method with no stabilization, projection method with sparse grad-div stabilization and ensemble based projection method with sparse grad-div stabilization using test problems with analytical solutions. The problems we consider are the Navier-Stokes equation with Taylor-Green vortex solution on the domain $\Omega = (0,1)^2$, (2.52)-(2.54); and the Ethier-Steinman solution on the domain $\Omega = (0,1)^3$, (2.55)-(2.58).

\[
\begin{align*}
  u_j &= (1 + \epsilon_j) \sin(\pi x) \cos(\pi y) e^{-2\nu_j \pi^2 t}, \quad (2.52) \\
  v_j &= -(1 + \epsilon_j) \cos(\pi x) \sin(\pi y) e^{-2\nu_j \pi^2 t}, \quad (2.53) \\
  p_j &= (1 + \epsilon_j)^2 \frac{1}{4} \left( \cos(2\pi x) + \cos(2\pi y) \right) e^{-4\nu_j \pi^2 t}, \quad (2.54)
\end{align*}
\]

where parameter $\epsilon_j$ is perturbation and $\nu_j$ is viscosity, $j = 1, 2, \cdots J$. We choose $a = d = 1$ in the following tests.
For the first problem, we use a standard finite element discretization on a uniform triangular mesh in Step 1 + Step 2. In particular, the \((P_2)^2, P_1\) Taylor-Hood velocity-pressure elements are used that consist of 162 velocity degrees of freedom and 25 pressure degrees of freedom. For the second problem, \((P_2)^3, P_1\) Taylor-Hood velocity-pressure elements are used that contains 2,187 velocity degrees of freedom and 125 pressure degrees of freedom.

In our experiment, we test a group of 3 NSE problems \((J=3)\) for 2d and 3d cases with different parameters: \(\epsilon_1 = 0, \nu_1 = 0.01\); \(\epsilon_2 = -0.002, \nu_2 = 0.005\) and \(\epsilon_3 = 0.002, \nu_3 = 0.015\). We choose the time step \(\Delta t = 0.001\) and end time \(T = 1\).

Time evolution of the step 1 velocity error in \(L^2\) and \(H^1\) norms are plotted in Figure 2.1 (2d) and Figure 2.2 (3d). It is observed that the individual and ensemble solution curves overlap with each other in both test problems, which reflects the accuracy of ensemble algorithm. The solution of the ensemble based projection method with sparse grad-div stabilization is significantly more accurate than the solution of projection method with no grad-div stabilization. In the 2d test case, the velocity error in solution of the ensemble based method with sparse grad-div stabilization is reduced by over 50% compared to the projection method with no grad-div stabilization. In the 3d NSE test case, the velocity error in the solution of ensemble based projection method with sparse grad-div stabilization is over 70% smaller than the velocity error in the solution of the projection method without stabilization. Especially, the ensemble based projection method with sparse grad-div stabilization provides a remarkable improvement on \(H^1\) norm of velocity error compared to the projection method without stabilization.

We test the temporal convergence rate of the ensemble based projection method with sparse grad-div stabilization. We use the same finite element but increase the number of degrees of freedom for velocity to 8450 and the number of degrees of freedom for pressure to 1089. We test a group of 3 NSE problems \((J=3)\) with different
Figure 2.1: (2d) Error plots of projection method with no stabilization, projection method with sparse grad-div stabilization and ensemble based projection method with sparse grad-div stabilization.

Figure 2.2: Error plots of projection method with no stabilization, projection method with sparse grad-div stabilization and ensemble-based projection method with sparse grad-div stabilization.

Table 2.1: The $L^2$ errors at final time

<table>
<thead>
<tr>
<th>dt</th>
<th>$| \mathbf{E}<em>1 |</em>{L^2}$</th>
<th>rate</th>
<th>$| \mathbf{E}<em>2 |</em>{L^2}$</th>
<th>rate</th>
<th>$| \mathbf{E}<em>3 |</em>{L^2}$</th>
<th>rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.025</td>
<td>0.057626</td>
<td>-</td>
<td>0.063519</td>
<td>-</td>
<td>0.055099</td>
<td>-</td>
</tr>
<tr>
<td>0.0125</td>
<td>0.031945</td>
<td>0.85</td>
<td>0.035331</td>
<td>0.85</td>
<td>0.030501</td>
<td>0.85</td>
</tr>
<tr>
<td>0.00625</td>
<td>0.016936</td>
<td>0.92</td>
<td>0.018775</td>
<td>0.91</td>
<td>0.016156</td>
<td>0.92</td>
</tr>
<tr>
<td>0.003125</td>
<td>0.008740</td>
<td>0.95</td>
<td>0.009701</td>
<td>0.95</td>
<td>0.008332</td>
<td>0.96</td>
</tr>
<tr>
<td>0.0015625</td>
<td>0.004442</td>
<td>0.98</td>
<td>0.004935</td>
<td>0.98</td>
<td>0.004234</td>
<td>0.98</td>
</tr>
<tr>
<td>0.00078125</td>
<td>0.002240</td>
<td>0.99</td>
<td>0.002489</td>
<td>0.99</td>
<td>0.002134</td>
<td>0.99</td>
</tr>
</tbody>
</table>

parameters: $\epsilon_1 = 0, \nu_1 = 0.01$; $\epsilon_2 = -0.001, \nu_2 = 0.008$ and $\epsilon_3 = 0.002, \nu_3 = 0.011$ and choose $\gamma = 0.02$. The $L^2$ norm of velocity errors at final time $T = 1$ are shown in Table 2.1. We observe that the convergence rate is close to 1, which is better than the error analysis result we proved.
2.5.2 Efficiency comparison

In this numerical experiment, we compare the efficiency of projection method with sparse grad-div stabilization, ensemble based projection method with standard grad-div stabilization and ensemble based projection method with sparse grad-div stabilization using the same 2d and 3d NSE test problems as the previous experiment.

We use Matlab’s backslash to solve linear systems in all the experiment. We display the simulation times of the 2d problem with different mesh sizes for the three methods in Table 2.2. We observe that the ensemble based projection method with sparse grad-div stabilization is two times faster than the projection method with sparse grad-div stabilization when the mesh is fine enough, in other words, the size of matrix of the linear system is not too small. The ensemble based projection method with sparse grad-div stabilization is over 10% faster than the ensemble based projection method with standard grad-div stabilization when mesh size $h$ is equal to or below $\frac{1}{16}$ under the setting in this problem. Table 2.3 displays the elapsed time for the three methods in the 3d problem with different mesh sizes. We observe that the ensemble based projection method with sparse grad-div stabilization is better than the ensemble based projection method with standard grad-div stabilization method from efficiency point of view.

We also test the speedup of the two ensemble based methods using the Navier-Stokes equation with Taylor-Green vortex solution on the domain $\Omega = (0,1)^2$ (2.52)-(2.54) with $h = 1/8$, $\Delta t = 0.001$, $T = 1$, $\nu_j = 0.01 + 0.001 \cdot j/(J-1)$ and $\epsilon_i = 0.001 \cdot j/(J-1)$, $j = 0, 1, \cdots, J-1$. Table 2.4 displays the elapsed time and the corresponding speedup of the two ensemble based methods for different values of $J$. We observe that the speedup increases as $J$ increases for both ensemble based methods. Ensemble methods are computationally more efficient because the same coefficient matrix is used in all ensemble members at each time step. When direct solver is applied, a single LU factorization can be used in solving all the problems.
Table 2.2: Comparison of the efficiency (2d).

<table>
<thead>
<tr>
<th>1/h</th>
<th>INPJ with SPGD</th>
<th>ENPJ with STGD</th>
<th>ENPJ with SPGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7 + 7 + 7 = 21</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>15 + 15 + 15 = 45</td>
<td>30</td>
<td>29</td>
</tr>
<tr>
<td>16</td>
<td>53 + 53 + 53 =159</td>
<td>108</td>
<td>83</td>
</tr>
<tr>
<td>32</td>
<td>306 + 323 + 313=942</td>
<td>472</td>
<td>422</td>
</tr>
</tbody>
</table>

Table 2.3: Comparison of the efficiency (3d).

<table>
<thead>
<tr>
<th>1/h</th>
<th>INPJ with SPGD</th>
<th>ENPJ with STGD</th>
<th>ENPJ with SPGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>17 + 17 + 18 = 52</td>
<td>48</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>88 + 87 + 88 = 263</td>
<td>200</td>
<td>176</td>
</tr>
<tr>
<td>8</td>
<td>1490+1491+1471 = 4452</td>
<td>3086</td>
<td>2191</td>
</tr>
</tbody>
</table>

Table 2.4: Comparison of the efficiency (2d, h = 1/8).

<table>
<thead>
<tr>
<th>J</th>
<th>ENPJ with STGD</th>
<th>speedup</th>
<th>ENPJ with SPGD</th>
<th>speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>30</td>
<td>1.50</td>
<td>29</td>
<td>1.55</td>
</tr>
<tr>
<td>10</td>
<td>66</td>
<td>2.27</td>
<td>66</td>
<td>2.27</td>
</tr>
<tr>
<td>30</td>
<td>167</td>
<td>2.69</td>
<td>163</td>
<td>2.76</td>
</tr>
<tr>
<td>100</td>
<td>516</td>
<td>2.90</td>
<td>505</td>
<td>2.97</td>
</tr>
</tbody>
</table>

Furthermore, by taking advantage of the typical sparsity of the matrices arising in Step 1 of the projection method, ensemble based projection method with sparse grad-div stabilization reduces the computational cost without loss of accuracy compared to the ensemble based method with standard grad-div stabilization, thus it highly improves the efficiency compared to the other two projection methods.

We also remark that since the right hand side of the linear system generated from the ensemble based methods is a matrix instead of a single column vector, it requires more memory volume in the simulations. This is one drawback of the ensemble method compared with other types of time stepping.
In this numerical experiment, we compare the projection method and ensemble based projection method with sparse grad-div stabilization using a 2d channel flow over a step problem. The domain is a $40 \times 10$ rectangle with a unit $1 \times 1$ step into the channel between 5 units and 6 units on the bottom. It is associated with a parabolic inflow and outflow, which is given by $\mathbf{u}_i = (1 + \epsilon_i)(u_1, u_2)^T$, with $u_1 = y(10 - y)/25, u_2 = 0$. No-slip boundary condition is prescribed on the top and bottom boundary as well as on the step. We use three perturbations of the boundary conditions corresponding to $\epsilon_1 = 0, \nu_1 = 1/600, \epsilon_2 = -0.002, \nu_2 = 1/600 - 1/4000$ and $\epsilon_3 = 0.002, \nu_3 = 1/600 + 1/4000$. The most distinctive feature of this flow is a recirculating vortex behind the step that detaches in the range $500 \leq Re \leq 700$.

For the spatial discretization, we apply $((P_2)^2, P_1)$ Taylor-Hood elements for velocity and pressure on a mesh with 9108 velocity degrees of freedom and 1178 pressure degrees of freedom. We choose the stabilization parameter $\gamma = 0.05$, time step $\Delta t = 0.1$ and $T = 40$. Streamline plots of the velocity at the final time $T$ are shown in Figure 2.3 for the projection method and ensemble based projection method with sparse grad-div stabilization. The velocity streamlines in three problems corresponding to the ensemble based projection method with sparse grad div stabilization are shown on the left column, and those corresponding to the independent projection method without stabilization are shown on the right column. It is observed that the eddies formed and detached behind the step in the ensemble simulation with stabilization, however, the eddies fails to detach behind the step in the individual simulation without stabilization. Divergence errors are shown in Table 2.5. It is seen that the approximation of the ensemble based projection method with sparse grad-div stabilization has better mass conservation, which is more physical.

Direct solver (Matlab’s backslash) is also used in this test case, it takes 358s to finish the individual simulations of the three problems, but only takes 188s to
Table 2.5: Comparison of the divergence errors.

<table>
<thead>
<tr>
<th>Member</th>
<th>INPJ without stabilization</th>
<th>ENPJ with sparse GD</th>
</tr>
</thead>
<tbody>
<tr>
<td>member 1</td>
<td>0.7597</td>
<td>0.2309</td>
</tr>
<tr>
<td>member 2</td>
<td>0.8821</td>
<td>0.2462</td>
</tr>
<tr>
<td>member 3</td>
<td>0.6791</td>
<td>0.2193</td>
</tr>
</tbody>
</table>

complete the ensemble simulations. Thus the ensemble simulation is two times faster than the individual ones.

2.6 Summary

In order to develop an efficient and accurate algorithm to simulate a group of incompressible flow problems, we combine the projection method with ensemble method: the former decouples the velocity and pressure; the later provides a fast solver of flow ensembles. In particular, an ensemble-based projection with sparse grad-div stabilization is proposed. Numerical results are presented to illustrate the theoretical analysis and computational efficiency of the method. So far, the direct solver is used in all the numerical tests. Next, we will investigate the performance of ensemble methods working with block iterative algorithms.
Figure 2.3: Velocity streamlines
CHAPTER 3

NUMERICAL INVESTIGATION OF ENSEMBLE METHODS
WITH BLOCK ITERATIVE SOLVERS FOR EVOLUTION
PROBLEMS

The ensemble time-stepping has been developed to accelerate a group of numerical simulations. Its main idea is to manipulate the numerical scheme so that all the group members could share a common coefficient matrix, then, instead of solving a sequence of linear systems with one right-hand-side vector, the method needs to solve one linear system with multiple right-hand-sides. Such a system could be solved more efficiently when it is of a small dimension as the same LU factorization works within the group of problems. If its dimension is high, it is possible to find solutions by using block iterative solvers. In this chapter, we investigate the numerical performance of the ensemble time-stepping algorithms on two application problems including heat equation and incompressible Navier-Stokes equations. The numerical results demonstrate the efficiency of the ensemble time-stepping together with block iterative algorithms.

3.1 Introduction

It is common to run a sequence of numerical simulations in scientific research and engineering application problems, such as numerical weather prediction using the ensemble-based data assimilation, proper orthogonal decomposition reduced order modeling that requires offline data generation, and uncertainty quantifications by
random sampling approaches, etc. The numerical simulations first need a full discretization of the problems in both space and time, after that, the task turns to solve a sequence of linear systems,

$$A_j x_j = b_j,$$

where $J$ is the total amount of simulations, named ensemble size; $A_j$, $x_j$ and $b_j$ are respectively the coefficient matrix, state variable vector and right-hand-side (RHS) vector in the $j$th discrete problem. The choice of linear solvers is determined by the size and structure of $A_j$. It is well known that dense direct methods depend only on the size of the problem, work well for size up to a few thousand; and sparse direct methods depend on both size and sparsity pattern, requires good ordering, can work for size up to $10^5$ or more. For problems of larger size such as $10^6$ or more, iterative methods have to be used, which depend on size, sparsity of the coefficient matrix and usually require preconditioning for accelerating convergence [46].

When all simulations in the sequence possess a common coefficient matrix $A$ (that is, $A_j = A$ for all $j$), direct methods can easily share information among RHSs: one factorization of $A$ could be used in solving all the problems. If the problem size is too big for direct solvers, block iterative methods have to be used, which are also able to share system information among all RHSs by using the same Krylov subspace, and solve all the linear systems simultaneously. For sequences sharing a common coefficient matrix, block iterative algorithms, such as block CG and variants for symmetric positive definite (SPD) system [47, 48, 49, 50], block GMRES and variants for nonsymmetric systems [51, 52, 53, 54, 55, 56, 57, 58] and hybrid block algorithms [59], have been developed to solve the system with many RHSs. The block algorithms have been used to accelerate the solution even for linear systems with only one RHS in [48, 60]. The advantage of a block solver over individual ones lies in the following facts: (i) the product of matrix and $J$ vectors is more efficient than $J$ times matrix-vector products [61, 62]; (ii) the searching space generated from
$J$ RHS vectors is usually larger than that from one single vector, thus the Krylov subspace method could potentially converge in less iterations [63]; (iii) when some of the RHSs are dependent, the searching space could be compressed and problems can be solved more efficiently; and (iv) it only accesses the coefficient matrix once a time for $J$ problems; when accessing $A$ represents a main computational bottleneck of a linear solver, or $A$ needs to re-generate at each time step, this leads to a significant computational advantage [64].

The appearance of a common coefficient matrix for a group of problems is appealing, however, more generally, the coefficient matrix would vary from one problem to another, then it becomes unapparent on how to share the information among RHSs. Hence, both direct solvers and block iterative algorithms cannot be directly applied. Approaches such as the seed/recycled Krylov subspace methods have been developed, which solve each RHS independently, while storing some information from the solve and using it in subsequent solves [65]. The accumulated information enlarges the searching space, thus would potentially accelerate the iterations. For slowly-changing linear systems, subspace recycling techniques such as GCRO with deflated restarting (GCRO-DR) have been introduced for accelerating the solutions in [66]. Recently, the block version of GCRO-DR was introduced in [63], and its high-performance implementation is available in the Belos package of the Trilinos project developed at Sandia National Laboratories. Note that the underlying assumption of such approaches is that all the systems are closely related, which certainly holds in some applications, but is lack of rigorous mathematical foundation.

As seen from the afore-discussion, the research for accelerating a sequence of numerical simulations mainly starts from the numerical linear algebra’s point of view and the goal is to solve (3.1) more efficiently when $A_j$ varies from one case to another. Recently, the ensemble method has been introduced to tackle this problem at the numerical algorithm level. The idea is to ensure all the linear systems share a common
coefficient matrix by manipulating numerical discretization schemes. Then, either
direct or block iterative solvers could be naturally applied. Such a approach would
lead to the following system:

\[ AX = B, \quad (3.2) \]

where \( A \) is the common coefficient matrix, \( X = [x_1, \ldots, x_J] \) and \( B = [b_1, \ldots, b_J] \)
consists of state variable vectors and RHS vectors from \( J \) discrete problems, respectively.

To the best of our knowledge, the research investigations on ensemble methods
primarily focus on the numerical analysis, including stability analysis and error es-
timates so far. The resultant linear systems are solved using directive solvers, but
not with block iterative ones. For instance, the ensemble-based Monte-Carlo and
multi-level Monte-Carlo methods have been developed in [68, 4]. With the LU fac-
torization, it has been shown that the use of ensemble methods leads to significant
computational savings over the individual simulations. However, large-scale appli-
cations need to be considered in practice. Hence, in this chapter, we take a couple
of widely used PDE models in incompressible fluid flows and study the numerical
behavior of the ensemble methods for numerical discretization together with block
iterative algorithms for linear solvers.

The rest of this chapter is organized as follows. In Section 3.2, we briefly review
the ensemble-based time-stepping algorithms for heat equation and Navier-Stokes
equations; Block iterative solvers are discussed in Section 3.3; Several numerical ex-
periments are presented in Section 3.4 for demonstrating the effectiveness of the
ensemble methods with block iterative solvers; We draw a few concluding remarks in
last section.
3.2 Ensemble methods for evolution problems

We consider two popular mathematical models governing heat transfer and incompressible fluid flows: heat equation and Navier-Stokes equations. Assume that, for either case, one needs to complete \( J \) numerical simulations under different computational settings including distinct body source functions, boundary and initial conditions, and physical parameters. We first present the ensemble time-stepping schemes for the aforementioned mathematical models and discuss the associated stability conditions and error estimates.

For simplicity of presentation, we assume the diffusion parameter \( \nu_j \), in the \( j \)-th problem, to be a constant function in this paper. We use the finite element method for a spatial discretization, but other types of numerical methods could be used as well. A uniform mesh \( \mathcal{T}_h \) with size \( h \), and a uniform time partition with time step size \( \Delta t \) are taken throughout our discussion. The following notations are to be used.

\[
\begin{align*}
t_n &= n\Delta t, \quad u^n_j = u_j(t_n), \quad f^n_j = f_j(t_n) \\
\bar{u}^n &= \frac{1}{J} \sum_{j=1}^{J} \bar{u}^n_j, \quad \nu^n = \frac{1}{J} \sum_{j=1}^{J} \nu_j, \quad \text{and} \quad \nu' = \nu_j - \nu.
\end{align*}
\]

(3.3)

- [first order] \( \bar{u}^n_j = u^n_j \), \( D_t u_j^{n+1} = \frac{u^{n+1}_j - u^n_j}{\Delta t} \); (3.4)
- [second order] \( \bar{u}^n_j = 2u^n_j - u^{n-1}_j \), \( D_t u_j^{n+1} = \frac{3u^{n+1}_j - 4u^n_j + u^{n-1}_j}{2\Delta t} \). (3.5)

3.2.1 Heat equation

Consider the heat equation that describes the distribution of temperature in a given region over time: to find \( u_j(x,t) \), for \( j = 1, \ldots, J \), satisfying

\[
\begin{align*}
\frac{\partial u_j}{\partial t} - \nabla \cdot (\nu_j \nabla u_j) &= f_j(x,t) \quad \text{in} \quad \Omega \times [0,T], \\
u_j &= g_j^D(x,t) \quad \text{on} \quad \Gamma_D \times [0,T], \\
\frac{\partial u_j}{\partial n} &= g_j^N(x,t) \quad \text{on} \quad \Gamma_N \times [0,T], \\
u_j(x,0) &= u_j^0(x) \quad \text{in} \quad \Omega.
\end{align*}
\]

(3.6)
In the ensemble-based time stepping, we introduce the ensemble average of diffusion coefficient and have the following semi-discrete equation as (3.3), and retain the same boundary and initial conditions. The semi-discrete system reads:

\[ D_t u^{n+1}_j - \nabla \cdot (\nu \nabla u^{n+1}_j) = f^{n+1}_j + \nabla \cdot (\nu' \nabla \tilde{u}^n_j). \] (3.7)

Define the trail and test spaces to be 
\[ V_{gD}^h := \{ u \in H^1(\Omega) | u = g_D \text{ on } \Gamma_D \} \]
and
\[ V_0^h := \{ u \in H^1(\Omega) | u = 0 \text{ on } \Gamma_D \}. \]
Denoted by \( V_{gD}^h \) the space of piecewise continuous functions on \( \Omega \) that reduce to polynomials of degree \( \leq m \) on each element of \( T^h \). Assume that \( V_{gD}^h \subset V_{gD} \) and apply the standard conforming finite element method, the fully discrete system reads: to find \( u_{n+1}^{j,h} \in V_{gD}^h \) such that 

\[ (D_t u_{j,h}^{n+1}, v) + (\nu \nabla u_{j,h}^{n+1}, \nabla v) = (f_{j}^{n+1}, v) - (\nu' \nabla \tilde{u}^n_j, \nabla v) + \langle \nu_j g_N, v \rangle_{\Gamma_N}, \quad \forall v \in V_0^h. \] (3.8)

The choices of \( D_t \) and \( \tilde{u}^n_j \), (3.4) or (3.5), lead to the ensemble-based time-stepping algorithms of accuracy order 1 or 2, respectively. The scheme needs an initial condition \( u_{j,h}^0 \) to start with, which can be the projection of \( u_j^0 \) onto the finite element space. The second-order scheme is a two-step method that requires one more initial condition \( u_{j,h}^1 \), which could be obtained from the first-order scheme.

As for the stability condition and error estimate, we have the following results ([68, 4]):

**Theorem 23.** Suppose \( f_j \in L^2(H^{-1}(D); 0, T) \), the ensemble scheme (3.8) is stable provided that

\[ \frac{|\nu_j - \bar{\nu}|_{\infty}}{\bar{\nu}} < 1 \quad \text{for the first-order scheme;} \] (3.9)

and

\[ \frac{|\nu_j - \bar{\nu}|_{\infty}}{\bar{\nu}} < \frac{1}{3} \quad \text{for the second-order scheme.} \] (3.10)

**Theorem 24.** Let \( u^n_j \) and \( u^{n}_{j,h} \) be the solutions of equations (3.6) and (3.8) at time \( t_n \), respectively. Assume \( f_j \in L^2(H^{-1}(\Omega); 0, T) \) and the stability condition, (3.9) or
(3.10), holds. Then there exists a generic constant $C > 0$ independent of $J$, $h$ and $\Delta t$ such that
\[
\|u_j^N - u_{j,h}^N\|^2 + \left(\nu - |\nu_j - \nu|_\infty\right)\Delta t \sum_{n=1}^N \|\nabla(u_j^n - u_{j,h}^n)\|^2 \leq C(\Delta t^2 + h^{2m}) \tag{3.11}
\]
for the first order scheme; and
\[
\frac{1}{4}\|u_j^N - u_{j,h}^N\|^2 + \left(\frac{\nu}{3} - |\nu_j - \nu|_\infty\right)\Delta t \sum_{n=1}^N \|\nabla(u_j^n - u_{j,h}^n)\|^2 \leq C(\Delta t^4 + h^{2m}) \tag{3.12}
\]
for the second order scheme.

3.2.2 Navier-Stokes equations

We next consider the Navier-Stokes equations that describe the motion of incompressible Newtonian fluid flows: to find the velocity vector $u_j(x,t)$, for $j = 1, \ldots, J$, satisfying
\[
\begin{cases}
\frac{\partial u_j}{\partial t} + u_j \cdot \nabla u_j - \nabla \cdot (\nu_j \nabla u_j) + \nabla p_j = f_j(x,t) & \text{in } \Omega \times [0,T], \\
\nabla \cdot u_j = 0 & \text{in } \Omega \times [0,T], \\
u_j = g_j^D(x,t) & \text{on } \Gamma_D \times [0,T], \\
u_j(x,0) = u_j^0(x) & \text{in } \Omega.
\end{cases} \tag{3.13}
\]

After introducing ensemble averages of diffusion coefficient and velocity field and using the ensemble-based time stepping, we achieve the following semi-discrete system associated with the same boundary and initial conditions of (3.13):
\[
\begin{cases}
D_t u_j^{n+1} + \bar{u}^n \cdot \nabla u_j^{n+1} - \nabla \cdot (\bar{\nu} \nabla u_j^{n+1}) + \nabla p_j^{n+1} = f_j^{n+1} \\
-(\bar{u}_j^n - \bar{u}^n) \cdot \nabla \bar{u}_j^n + \nabla \cdot ((\nu_j - \bar{\nu}) \nabla \bar{u}_j^n) \\
\nabla \cdot u_j^{n+1} = 0.
\end{cases} \tag{3.14}
\]

Let $Q^h$ be the space of piecewise continuous functions on $\Omega$ that reduce to polynomials of degree $\leq s$ on each element of $T^h$. Assume that the pair of spaces $(V_0^h, Q^h)$ satisfies the discrete inf-sup (or $LBB^h$) condition, which is required to guarantee the
stability of FE approximations. One example for which the $LBB^h$ stability condition holds is the family of Taylor-Hood $P^{s+1}-P^s$ element pairs (i.e., $m = s + 1$ in the definition of $V^h_0$), for $s \geq 1$ [69]. The fully discrete system reads: to find $u_{j,h}^{n+1} \in V^h_{gD}$ and $p_{j,h}^{n+1} \in Q^h$ satisfying

$$
\left\{
\begin{aligned}
&\left(D_t u_{j,h}^{n+1}, v_h\right) + b^* (\mathbf{u}_h, u_{j,h}^{n+1}, v_h) + (\nabla u_{j,h}^{n+1}, \nabla v_h) - (p_{j,h}^{n+1}, \nabla \cdot v_h) \\
= & \left(f_{j}^{n+1}, v_h\right) - b^* (\mathbf{u}_h, u_{j,h}^{n}, \mathbf{u}_j^{n}, v_h) + \left((\nu_j - \nu) \nabla \mathbf{u}_j^{n}, \nabla v_h\right), \quad \forall v_h \in V^h_0, \\
&\left(\nabla \cdot u_{j,h}^{n+1}, q_h\right) = 0, \quad \forall q_h \in Q^h.
\end{aligned}
\right.
$$

(3.15)

The choices of $D_t$ and $\mathbf{u}_j^n$, (3.4) or (3.5), lead to the ensemble-based time stepping of first order accuracy or second order, respectively. The initial condition $u_{j,h}^0$ can be obtained by projecting $u_j^0$ onto the FE space, while the second initial condition required for starting the second-order scheme can be sought by the first-order scheme.

The stability conditions of the above scheme (3.15) are provided in Theorem 11 and Theorem 13, and the error estimates are provided in Theorem 12 and Theorem 14.

### 3.3 Block-based iterative solvers

Since ensemble-based time-stepping schemes lead to a single system (3.2) for all the problems, direct methods can be used to efficiently solve it if the size of problems is small. But for large-scale problems, block-based iterative solvers have to be used in order to accelerate the solutions.

Assume there are $n_u$ local degrees of freedom and define the local functions to be $\{\psi_1, \psi_2, \ldots, \psi_{n_u}\}$. Let the approximate solution be $u_{j,h} = \sum_{i=1}^{n_u} u_i^{(j)} \psi_i$. The mass and stiffness matrices are denoted by $\mathbf{M} = [m_{ik}]$ and $\mathbf{S} = [s_{ik}]$ with elements $m_{ik} = \int_{\Omega} \psi_k \psi_i \, dx$, and $s_{ik} = \int_{\Omega} \nabla \psi_k \cdot \nabla \psi_i \, dx$. The vector related to the source and boundary terms is defined to be $\mathbf{h}_j^n = [h_i^{(j)}]$ with entries $h_i^{(j)} = \int_{\Omega} f_j^n \psi_i \, dx + \langle \nu g_N, \psi_i \rangle$. Denote the approximate solution vector at $t_n$ by $\mathbf{h}_j^n$. 

53
3.3.1 Solution to discrete heat equation

The full discretization of the equation (3.8) yields the following linear system for the ensemble:

\[ \mathbf{A} \mathbf{U}^{n+1} = \mathbf{B}^{n+1}, \]  

(3.16)

where \( \mathbf{A} \) is a sparse, symmetric and positive defined (SPD), \( n \times n \) matrix and \( \mathbf{B}^{n+1} \) is a \( n \times J \) matrix. Furthermore,

\[
\mathbf{A} = \frac{1}{\Delta t} \mathbf{M} + \nu \mathbf{S}, \\
\mathbf{B}^{n+1} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_J] \text{ with } \mathbf{b}_j = \mathbf{h}_j^{n+1} - \nu' \mathbf{S} \mathbf{u}_j^n + \frac{1}{\Delta t} \mathbf{M} \mathbf{u}_j^n
\]

for first-order scheme and

\[
\mathbf{A} = \frac{3}{2\Delta t} \mathbf{M} + \nu \mathbf{S}, \\
\mathbf{B}^{n+1} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_J] \text{ with } \mathbf{b}_j = \mathbf{h}_j^{n+1} - \nu' \mathbf{S} \tilde{\mathbf{u}}_j^n + \frac{4}{2\Delta t} \mathbf{M} \mathbf{u}_j^n - \frac{1}{2\Delta t} \mathbf{M} \mathbf{u}_j^{n-1}
\]

for second-order scheme.

If one has access to the semi-implicit numerical solvers (the coefficient matrix \( \mathbf{A}^{(j)} \) and vector \( \mathbf{b}^{(j)} \)) for individual simulations, then it is straightforward to obtain the linear system for the aforementioned ensemble simulations. Take the first-order scheme for example, the coefficient matrix \( \mathbf{A}^{(j)} = \frac{1}{\Delta t} \mathbf{M} + \nu_j \mathbf{S} \) and vector \( \mathbf{b}^{(j)} = \mathbf{h}_j^{n+1} + \frac{1}{\Delta t} \mathbf{M} \mathbf{u}_j^n \) can be extracted from individual simulation codes for \( j = 1, \ldots, J \), then the matrices in ensemble simulations are: \( \mathbf{A} = \frac{1}{J} \sum_{j=1}^{J} \mathbf{A}^{(j)} \) and \( \mathbf{B}^{n+1} = [\mathbf{b}_1, \ldots, \mathbf{b}_J] \) with \( \mathbf{b}_j = \mathbf{b}^{(j)} + (\mathbf{A} - \mathbf{A}^{(j)}) \mathbf{u}_j^n \). What’s more, \( \mathbf{A} \) inherits the same sparse, SPD structure of \( \mathbf{A}^{(j)} \).

When the size of \( \mathbf{A} \) is large, the iterative methods need to be used for accelerating the solution process. Among the iterative algorithms in the general family of Krylov subspace methods, the conjugate gradient method (CG) developed by Hestenes and Stiefel is the most well known for solving a real, SPD system [46]. For a SPD system with multiple RHSs, a block conjugate gradient method (BCG) has been developed as
a generalization of CG in this context in [47] and further in [48, 49]. Comparing with CG, BCG has the advantage of potentially faster convergence because the searching space is augmented when multiple problems are considered instead of a single problem. It becomes more attractive when accessing $A$ represents the main computational bottleneck of a linear solver (e.g. when $A$ is stored outside of the system memory, or the elements of $A$ have to be regenerated at each use) as it can explore multiple search directions in a single pass over $A$. However, in practice, BCG could fail due to the rank deficiency problems in which the block searching direction vectors become linearly dependent. A simple solution was developed in [50], which extracts a set of basis for the searching space at each iteration and uses it as the new search direction vectors, thus could avoid inverting a potentially non-full rank matrix. It is presented in Algorithm 1.

**Algorithm 1:** The breakdown-free block CG [50]

<table>
<thead>
<tr>
<th>Input:</th>
<th>matrix $A \in \mathbb{R}^{n \times n}$, matrix $B \in \mathbb{R}^{n \times J}$, initial guess $X_0 \in \mathbb{R}^{n \times J}$, preconditioner $M \in \mathbb{R}^{n \times n}$, tolerance $tol \in \mathcal{R}$ and maximum number of iteration $\text{maxit} \in \mathcal{R}$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>approximate solution $X_s \in \mathbb{R}^{n \times J}$</td>
</tr>
<tr>
<td></td>
<td>$R_0 = B - AX_0$;</td>
</tr>
<tr>
<td></td>
<td>solve $MZ_0 = R_0$;</td>
</tr>
<tr>
<td></td>
<td>$P_0 = \text{orth}(Z_0)$;</td>
</tr>
<tr>
<td></td>
<td>for $i=0, \ldots, \text{maxit}$ do</td>
</tr>
<tr>
<td></td>
<td>$Q_i = AP_i$;</td>
</tr>
<tr>
<td></td>
<td>$T_i = P_i^T Q_i$;</td>
</tr>
<tr>
<td></td>
<td>$\Theta_i = T_i^{-1} (P_i^T R_i)$;</td>
</tr>
<tr>
<td></td>
<td>$X_{i+1} = X_i + P_i \Theta_i$;</td>
</tr>
<tr>
<td></td>
<td>$R_{i+1} = R_i - Q_i \Theta_i$;</td>
</tr>
<tr>
<td></td>
<td>if converged then exit;</td>
</tr>
<tr>
<td></td>
<td>$Z_{i+1} = M^{-1} R_{i+1}$;</td>
</tr>
<tr>
<td></td>
<td>$\Lambda_i = -T_i^{-1} (Q_i^T Z_{i+1})$;</td>
</tr>
<tr>
<td></td>
<td>$P_{i+1} = \text{orth}(Z_{i+1} + P_i \Lambda_i)$;</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
<tr>
<td></td>
<td>$X_s = X_{i+1}$.</td>
</tr>
</tbody>
</table>

Different from the original BCG, the breakdown-free BCG introduces an orthogonalization process, $\text{orth}$, which extracts an orthogonal basis $P_i$ from the searching space (denoted by $P_i$). This is useful to overcome situations in which two or more
vector components in the residual matrix $R_i$ are dependent because the lack of full rank in $R_i$ would lead to rank deficiency in $Z_i$ and $P_i$, and further fails the BCG method. Assume the rank of $P_i$ is $r_i$, the resulting orthogonal matrix $P_i \in \mathbb{R}^{n_u \times r_i}$.

In Algorithm 1, the choice of $\Theta_i$ and $\Lambda_i$ guarantees the column spaces of $R_{i+1}$ is orthogonal to the search space $P_i$, and $P_{i+1}$ is conjugate to all previous searching spaces.

At each iteration, the algorithm involves three matrix-matrix products: $A_{n_u \times n_u}$, $(P_i)_{n_u \times r_i}$, $(Q_i)_{n_u \times J}$, $(Z_i)_{n_u \times J}$; three matrix updates that include three $n_u \times r_i$ matrix and $r_i \times J$ matrix products $P_i \Theta_i$, $Q_i \Theta_i$ and $P_i \Lambda_i$; two solutions of linear systems with coefficient matrix $(T_i)_{r_i \times r_i}$; one solution of a linear system with coefficient matrix $M_{n_u \times n_u}$; and an orthogonalization. Suppose $n_u$ is much bigger than $r_i$, and $M$ is easy to invert as a preconditioner, the above mentioned matrix-matrix products and system solvers require $O(n_u J \max(\ell, r))$, where $\ell$ is the number of nonzero entries in each row of $A$. Special attention should be paid to the orthogonalization process $\text{orth}$. It could be implemented by reduced (economy) SVD. However, to reduce the computational complexity, we choose the method of snapshots and drop the singular values less than $10^{-12}$. To determine left singular vectors associated with the retained singular values, it only requires $O(n_u J^2)$, thus is comparable to the other operations in the algorithm.

The performance of the algorithm will be investigated in Section 3.4, in which we choose the incomplete LU factorization for preconditioning, but other types of preconditioners such as multigrid and domain decomposition can be certainly used.

3.3.2 Solution to discrete Navier-Stokes equations

Assume there are $n_u$ local degrees of freedom for velocity and define the local functions by $\{\phi_1(x), \phi_2(x), \ldots, \phi_{n_u}(x)\}$, and assume $n_p$ local degrees of freedom for pressure with the local functions defined by $\{\psi_1(x), \psi_2(x), \ldots, \psi_{n_p}(x)\}$. Let the approximate
solution be $u_{j,h}(t, x) = \sum_{i=1}^{n_u} u_i^{(j)}(t) \phi_i(x)$ and $p_h = \sum_{i=1}^{n_p} p_i \psi_i$. The mass and stiffness matrices for velocity are defined to be $M = [m_{ik}]$ and $S = [s_{ik}]$ with elements $m_{ik} = \int_\Omega \phi_k \cdot \phi_i \, dx$, and $s_{ik} = \int_\Omega \nabla \phi_k \cdot \nabla \phi_i \, dx$. The source term is $h^n_j = [h^n_{j(i)}]$ with entries $h^n_{j(i)} = \int_\Omega f^n_j \cdot \phi_i \, dx$. We also define the matrix $P = [d_{ik}]$ with entries $d_{ik} = -\int_\Omega \psi_k \nabla \cdot \phi_i \, dx$, and the matrices from the discretization of the convection term $\mathbb{N}^n_j = [n_{ik}]$ and $\mathbb{R}^n_j = [r_{ik}]$ with entries $n_{ik} = \int_\Omega (u^n_h \cdot \nabla \phi_k) \cdot \phi_i \, dx$ and $r_{ik} = \int_\Omega (u^n_{j,h} \cdot \nabla \phi_k) \cdot \phi_i \, dx$.

The discrete linear system of (3.15) reads

$$
\begin{bmatrix}
C & D^T \\
D & 0 \\
\end{bmatrix}
\begin{bmatrix}
U^{n+1} \\
P^{n+1} \\
\end{bmatrix}
= 
\begin{bmatrix}
B^{n+1} \\
0 \\
\end{bmatrix}.
$$

(3.17)

where

$$
C = \frac{1}{\Delta t} M + \mathbb{N}^n + \nu S,
$$

$$
B^{n+1} = [b_1, b_2, \ldots, b_J] \text{ with } b_j = h^{n+1}_j - R^n_j \tilde{u}^n_j - \nu' \mathbb{S} \tilde{u}^n_j + \frac{1}{\Delta t} M u^n_j
$$

for first-order scheme and

$$
C = \frac{3}{2\Delta t} M + \mathbb{N}^n + \nu S,
$$

$$
B^{n+1} = [b_1, b_2, \ldots, b_J] \text{ with } b_j = h^{n+1}_j - R^n_j \tilde{u}^n_j - \nu' \mathbb{S} \tilde{u}^n_j + \frac{4}{2\Delta t} M u^n_j - \frac{1}{2\Delta t} M u^{n-1}_j
$$

for second-order scheme.

Similar to the aforementioned heat equation case, if one has access to the semi-implicit numerical solvers (the coefficient matrix $A^{(j)}$ and vector $b^{(j)}$), then the linear system for the ensemble simulations could be assembled in a straightforward manner. Take the first-order scheme for example, we have $A^{(j)} = \begin{bmatrix}
\frac{1}{\Delta t} M + N^n_j + \nu_j S & D^T \\
D & 0 \\
\end{bmatrix}$ and $b^{(j)} = \begin{bmatrix}
h^{n+1}_j + \frac{1}{\Delta t} M u^n_j \\
0 \\
\end{bmatrix}$ for $j = 1, \ldots, J$, then the linear system in ensemble-based time stepping could be generated by using $A = \frac{1}{J} \sum_{j=1}^{J} A^{(j)}$ and $b_j = b^{(j)} + (A - A^{(j)})[\tilde{u}^n_j; 0]^T$. Note that $A$ has the same sparse structure of $A^{(j)}$ and is non-symmetric as well. When the size of $A$ is large, the generalized minimum residual
method (GMRES) method could be used to solve the linear system of a single RHS vector. To accelerate the iteration, we follow [70] and use the least-square commutator preconditioner $K$, which has the following form:

$$K = \begin{bmatrix} C & D^T \\ 0 & -K_S \end{bmatrix} \text{ and } K_S = (D\hat{M}^{-1}D^T)(D\hat{M}^{-1}C\hat{M}^{-1}D^T)^{-1}(D\hat{Q}^{-1}D^T),$$

where $\hat{M} = \text{diag}(M)$ consists of the diagonal entries of the velocity mass matrix. This preconditioning is applicable when the mixed approximation is uniformly stable with respect to the inf-sup condition, which is also fully automated, without requiring the construction of any auxiliary operators.

The GMRES, developed by Saad and Schultz [71], is to find an approximate solution from the Krylov subspace that minimizes the residual. The algorithm has been extended to block versions (see [72, 54] for an introduction and [51] for analysis), which uses block Arnoldi algorithm in generating Krylov subspace vectors. Algorithm 2 shows the BGMRES that uses Algorithm 3 for block Arnoldi process.

**Algorithm 2**: Block GMRES algorithm

**Input**: matrix $A \in \mathbb{R}^{n \times n}$, matrix $B \in \mathbb{R}^{n \times J}$, initial guess $X_0 \in \mathbb{R}^{n \times J}$, preconditioner $K \in \mathbb{R}^{n \times n}$, tolerance $\text{tol} \in \mathbb{R}$ and maximum number of iteration $\text{maxit} \in \mathbb{R}$.

**Output**: approximate solution $X_s \in \mathbb{R}^{n \times J}$.

1. $R_0 = B - AX_0$;
2. find $V_1$ via a reduced QR factorization of $R_0 = V_1Z$;
3. for $m = 1, 2, \ldots, \text{maxit}$ do
   1. compute $V_{m+1}$ by performing the block Arnoldi algorithm on $\{V_i\}_{i=1}^m$ and form the block upper Hessenberg $H_m$;
   2. solve $V_m = \arg\min_{Y} \| H_m Y - E_1 Z \|_F$ by Householder reflections, where $E_1 \in \mathbb{R}^{(m+1) \times J}$ is the matrix containing the first $J$ columns of the identity;
   3. if converged then exit;
4. end
5. form the solutions: $X_m = X_0 + [V_1, V_2, \ldots, V_m]Y_m$.

Since the RHS vectors during the ensemble simulations could become linearly dependent, deflation can be executed to remove reductant information. Such algorithms have been designed in [56, 57] that perform deflations either at the beginning
Algorithm 3: One step of block Arnoldi algorithm

**Input:** matrix $A \in \mathbb{R}^{n \times n}$, matrices $V_1, \ldots, V_m \in \mathbb{R}^{n \times J}$ and $H_{m-1} \in \mathbb{R}^{mJ \times mJ}$.

**Output:** $V_{m+1} \in \mathbb{R}^{n \times J}$, $H_m \in \mathbb{R}^{(m+1)J \times (m+1)J}$.

$$V_{m+1} = A V_m;$$

for $i=1, 2, \ldots, m$ do

- $H_{i,m} = V_i^T V_{m+1};$
- $V_{m+1} = V_{m+1} - V_i H_{i,m};$

end

find $V_{m+1}, H_{m+1,m}$ such that $V_{m+1} = V_{m+1} H_{m+1,m}$ via reduced QR factorization;

set $H_m = [H_{1,m}, \ldots, H_{m,m}]^T$.

of iterations or during the whole iterations. As for time dependent problems, the solution at previous time step provides a good initial guess of solutions, the number of iterations could be small with the further help of a well-designed preconditioner.

Therefore, in our numerical experiments in Section 3.4, we choose to perform the deflation once a time step at the beginning of the iterative solve. The Algorithm is shown in Algorithm 4.

Algorithm 4: Block GMRES algorithm with deflations [56]

**Input:** matrix $A \in \mathbb{R}^{n \times n}$, matrix $B \in \mathbb{R}^{n \times J}$, initial guess $X_0 \in \mathbb{R}^{n \times J}$, preconditioner $K \in \mathbb{R}^{n \times n}$, scaling matrix $D \in \mathbb{R}^{J \times J}$, tolerance $tol \in \mathbb{R}$ and maximum number of iteration $maxit \in \mathbb{R}$.

**Output:** Solution of linear system $AX = B$.

$R_0 = B - AX_0$;
compute reduced QR factorization of $R_0 D^{-1}$ as $R_0 D^{-1} = QT$;
compute SVD of $T = U \Sigma W^T$ and determine the first diagonal entry of $\Sigma$, $\sigma_{P_d}$, such that $\sigma_{P_d} < \epsilon_d$;
find $V_1 \in \mathbb{R}^{n \times P_d}$ with $V_1 = QU(:,1:P_d)$;
for $m=1, 2, \ldots, maxit$ do

- compute $V_{m+1}$ by performing the block Arnoldi iteration and form the block upper Hessenberg $H_m$;
- let $B_m = \begin{bmatrix} 0_{P_d \times P_d}; 0_{(m-1)P_d \times P_d} \end{bmatrix}$;
- solve $Y_m = \text{argmin} \| H_m Y - B_m \|_F$ by Householder reflections;
- compute $\mathbb{R}_m = (B_m - H_m Y_m) \Sigma (1 : P_d, 1 : P_d) W(:,1 : P_d)^T$;
- if $\| \mathbb{R}_m \|_F \leq tol$ then
  - compute $X_j = X_0 + Z_j Y_j \Sigma (1 : P_d, 1 : P_d) W(:,1 : P_d)^T D$, and stop
end
form the solutions: $X_m = X_0 + Z_m Y_m \Sigma (1 : P_d, 1 : P_d) W(:,1 : P_d)^T D;$
Comparing with the block GMRES in Algorithm 2, Algorithm 4 has an extra execution of SVD of the matrix \( T \in \mathcal{R}^{J \times J} \) at the beginning of the iterative solution, which only cost \( \mathcal{O}(J^3) \) that is relatively small. Due to the truncation of singular values, the dimension of \( \mathbf{V}_1 \) usually is less than, and never greater than, \( J \), which reduce the computational efforts in performing the block Arnoldi iterations.

### 3.4 Numerical experiments

The goal of this section is to test the performance of ensemble simulations that use the aforementioned block iterative algorithms to solve the resulting linear systems. The performance will be compared with the corresponding individual simulations. To this end, we first validate our simulation codes by checking the convergence rates of ensemble approximations, then use the CPU time as a criterion for measuring the computational efficiency. In particular, for measuring approximation errors, we define

\[ E_j^N := \| u_j(t_N) - u_{j,h}^N \| \],

the \( L^2 \) error of ensemble simulations at the final time \( t_N \); and

\[ E_j^N \] to be the \( L^2 \) error of individual simulations at the final time. All simulations will be implemented on Matlab and performed on a PC, equipped with an Intel Core i7 processor running at 2.9 GHz.

**Problem 1** Consider a group of 100 \((J = 100)\) heat diffusion problems on a rectangular domain \([0, 1] \times [0, 2]\) over time interval \([0, 1]\). Dirichlet boundary conditions are imposed on the left and right edges. Neumann boundary conditions are imposed on the top and bottom. In the \( j \)th problem, initial and boundary conditions as well as body source in (3.6) are selected to match the prescribed analytic solution

\[ u_j(x, y, t) = (1 + w_j)[\sin(2\pi x) \cos(2\pi y) + \sin(4\pi t)], \]

where \( w_j \) is a random perturbation on \([-0.2, 0.2]\). The diffusion coefficient \( \nu_j = 0.01(1 + \epsilon_j) \) with \( \epsilon_j \) a random perturbation on \([-0.2, 0.2]\).
We first check the rate of convergence in $\mathcal{E}^N_j$ by considering two test cases: (i) uses the first-order ensemble method together with bilinear elements; and (ii) the second one uses the second-order ensemble method together with bi-quadratic elements. Uniform rectangular meshes with size $h$ and uniform time discretization with step size $\Delta t$ are used for partitioning spatial domain and temporal interval, respectively. Denote by $N_x$, $N_y$ and $K$ the number of partitions in horizontal, vertical and temporal directions. Based on Theorem 24, the ensemble simulation is expected to converge linearly in the first case and converge quadratically in the second case when $h \sim \mathcal{O}(\Delta t)$ and a uniform mesh refinement is taken.

Since the associated discrete systems are not large-scale, one could use a direct solver such as LU or sparse LU in solving the systems. However, we would like to check the performance of ensemble methods working with a block iterative algorithm. Hence, we use the breakdown-free preconditioned BCG as the linear solver for the ensemble simulations as discussed in subsection 3.3.1. An incomplete LU factorization is applied for preconditioning. In the iterative algorithm, the maximum number of iterations is set to be $\text{maxit} = 20$ and convergence criteria is relative residual less than $\text{tol} = 1 \times 10^{-8}$. The ensemble size is $J = 100$. But due to the limit of space, we only list the results of three problems with $j = 1, 50$ and 100 in Tables 3.1 - 3.2 for these two test cases, respectively. Wherein,

$$
\nu_1 = 1.1901 \times 10^{-2}, \quad \nu_{50} = 8.4951 \times 10^{-3}, \quad \nu_{100} = 1.0154 \times 10^{-2}; \\
\nu_{1} = 9.6995 \times 10^{-2}, \quad \nu_{50} = -9.4653 \times 10^{-2}, \quad \nu_{100} = -3.3367 \times 10^{-2}.
$$

It is observed that, in either case, the expected rate of convergence has been achieved.

Next, we compare the performance of ensemble simulations with individual simulations on the same group of problems. For this purpose, we take the first-order ensemble method in time and bilinear finite elements in space with the following parameters: $N_x = 128$, $N_y = 256$ and $K = 400$. The total number of degrees of free-
Table 3.1: The $L^2$ errors at final time: first-order ensemble, $Q_1$ elements.

<table>
<thead>
<tr>
<th>$(N_x, N_y, K)$</th>
<th>$\mathcal{E}^N_1$</th>
<th>Rate</th>
<th>$\mathcal{E}^N_{50}$</th>
<th>Rate</th>
<th>$\mathcal{E}^N_{100}$</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>(16, 32, 50)</td>
<td>$5.8050 \times 10^{-2}$</td>
<td>–</td>
<td>$4.3544 \times 10^{-2}$</td>
<td>–</td>
<td>$4.8908 \times 10^{-2}$</td>
<td>–</td>
</tr>
<tr>
<td>(32, 64, 100)</td>
<td>$2.9140 \times 10^{-2}$</td>
<td>0.99</td>
<td>$1.0610 \times 10^{-2}$</td>
<td>0.99</td>
<td>$1.2371 \times 10^{-2}$</td>
<td>0.99</td>
</tr>
<tr>
<td>(64, 128, 200)</td>
<td>$7.3326 \times 10^{-3}$</td>
<td>1.00</td>
<td>$5.5529 \times 10^{-3}$</td>
<td>1.00</td>
<td>$6.2053 \times 10^{-3}$</td>
<td>1.00</td>
</tr>
<tr>
<td>(128, 256, 400)</td>
<td>$1.4629 \times 10^{-3}$</td>
<td>1.00</td>
<td>$1.1061 \times 10^{-3}$</td>
<td>1.00</td>
<td>$1.2371 \times 10^{-3}$</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 3.2: The $L^2$ errors at final time: second-order ensemble, $Q_2$ elements.

<table>
<thead>
<tr>
<th>$(N_x, N_y, K)$</th>
<th>$\mathcal{E}^N_1$</th>
<th>Rate</th>
<th>$\mathcal{E}^N_{50}$</th>
<th>Rate</th>
<th>$\mathcal{E}^N_{100}$</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8, 16, 50)</td>
<td>$3.1827 \times 10^{-3}$</td>
<td>–</td>
<td>$2.4799 \times 10^{-3}$</td>
<td>–</td>
<td>$2.7259 \times 10^{-3}$</td>
<td>–</td>
</tr>
<tr>
<td>(16, 32, 100)</td>
<td>$7.6003 \times 10^{-4}$</td>
<td>2.07</td>
<td>$5.8014 \times 10^{-4}$</td>
<td>2.10</td>
<td>$6.4617 \times 10^{-4}$</td>
<td>2.08</td>
</tr>
<tr>
<td>(32, 64, 200)</td>
<td>$1.9288 \times 10^{-4}$</td>
<td>1.98</td>
<td>$1.4695 \times 10^{-4}$</td>
<td>1.98</td>
<td>$1.6366 \times 10^{-4}$</td>
<td>1.98</td>
</tr>
<tr>
<td>(64, 128, 400)</td>
<td>$4.9629 \times 10^{-5}$</td>
<td>1.96</td>
<td>$3.7682 \times 10^{-5}$</td>
<td>1.96</td>
<td>$4.2046 \times 10^{-5}$</td>
<td>1.96</td>
</tr>
</tbody>
</table>

The individual simulations are based on a semi-implicit time stepping together with the standard PCG algorithm, but each problem in the group is solved separately. We choose the convergence criteria of PCG to be same as the BFBCG: $\text{maxit} = 20$ and $\text{tol} = 1 \times 10^{-8}$. Three individual simulations of problems with $j = 1, 50$ and $100$ have the following approximation errors:

$$E_1^N = 7.3268 \times 10^{-3}, \quad E_{50}^N = 5.5584 \times 10^{-3}, \quad E_{100}^N = 6.2051 \times 10^{-3}.$$ 

Comparing them with the corresponding ensemble simulation errors (listed in the last row of Table 3.1), we observe the accuracy of both approaches are very close. However, as listed in Table 3.3, the CPU time for time integrations in the ensemble with BFBCG solver is $5.658 \times 10^2$ seconds, while that of individual simulations with PCG solvers is $1.4640 \times 10^3$ seconds, which leads to a speedup factor of nearly 2.60.

Table 3.3: CPU time comparison.

<table>
<thead>
<tr>
<th>Iteration &amp; CPU time</th>
<th>BFBCG</th>
<th>100 CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>average iteration number per time step</td>
<td>4</td>
<td>4 \times 100</td>
</tr>
<tr>
<td>average execution time per step (seconds)</td>
<td>$5.6362 \times 10^{-1}$</td>
<td>$(1.1482 \times 10^{-2}) \times 100$</td>
</tr>
<tr>
<td>total CPU time for integration (seconds)</td>
<td>$5.658 \times 10^2$</td>
<td>$1.464 \times 10^3$</td>
</tr>
</tbody>
</table>

In this case, the number of iterations in BFBCG per time step is the same as each individual PCG solve per time step. The average execution time per time step
in BFBCG (for 100 problems) is $5.6362 \times 10^{-1}$ seconds, and that in each individual PCG solve is $1.1482 \times 10^{-2}$ seconds. For a fair comparison, we multiply the execution time of a single PCG solve by 100, which costs about 2 times larger than one BFBCG solve. We observe that the main computational saving in BFBCG comes from the reduction of searching directions. We plot the change in the rank of $P_i$ with respect to iterations every 10 time steps in Figure 3.1. It is seen that the searching dimension increases with iterations, but the largest rank of $P_i$ during the simulation is 9, which is much less than $J$.

Figure 3.1: Time evolution of the rank of searching space in BFBCG

Problem 2 Consider a group of 40 ($J = 40$) Navier-Stokes equations with Taylor-Green vortex solutions on a square domain $[-1, 1] \times [-1, 1]$ over time interval $[0, 1]$. Dirichlet boundary conditions are imposed on the edges. Initial and boundary conditions and body force are selected to match the prescribed analytic solution. In the $j$th problem,

\[
\begin{align*}
  u_j(x, y, t) &= \sin(\pi x) \cos(\pi y)e^{-2\nu_j \pi^2 t}, \\
  v_j(x, y, t) &= -\cos(\pi x) \sin(\pi y)e^{-2\nu_j \pi^2 t}, \\
  p_j(x, y, t) &= \frac{1}{4}(\cos 2\pi x + \cos 2\pi y)e^{-4\nu_j \pi^2 t},
\end{align*}
\]

and the diffusion coefficient $\nu_j = 0.01(1 + \epsilon_j)$ with $\epsilon_j$ a random perturbation of magnitude 0.2.
We first check the rate of convergence in velocity approximations for the first-order and second-order ensemble methods, respectively. The $Q_2/Q_1$ Taylor-Hood elements are used in both cases for spatial discretizations. Uniform rectangular meshes with size $h$ and uniform time discretization with step size $\Delta t$ are selected respectively for partitioning the spatial domain and temporal interval. The same notation $h$, $\Delta t$, $N_x$, $N_y$ and $K$ as Problem 1 are used. In both tests, we fix big enough $N_x$ and $N_y$ so that the temporal error would dominate the approximation errors, and vary the time step size to check the rate of convergence. Based on Theorem 12 and 14, the ensemble simulation is expected to converge linearly in the first case and converge quadratically in the second case when a uniform time refinement is taken.

As discussed in subsection 3.3.2, the block GMRES algorithm together with the least-square commutator preconditioner is used for solving the discrete systems resulted from ensemble methods. The maximum number of iterations is $\maxit = 50$ and stopping criterion is fulfilled if the relative residual is no greater than $\text{tol} = 1 \times 10^{-8}$. The ensemble size is $J = 40$. However due to the limit of space, we only list the results of three problems for $j = 1, 20$ and 40 in Tables 3.4 - 3.5. Wherein,

$$\nu_1 = 8.0619 \times 10^{-3}, \quad \nu_{20} = 1.1681 \times 10^{-2}, \quad \nu_{40} = 1.0804 \times 10^{-2},$$

and $\mathcal{E}_j^N$ denotes the velocity approximation errors in $L^2$ norm at the final time. It is seen, in both cases, that the expected rates of convergence have been obtained.

<table>
<thead>
<tr>
<th>$(N_x, N_y, K)$</th>
<th>$\mathcal{E}_1^N$</th>
<th>Rate</th>
<th>$\mathcal{E}_{20}^N$</th>
<th>Rate</th>
<th>$\mathcal{E}_{40}^N$</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>(128, 128, 5   )</td>
<td>3.7899$\times 10^{-3}$</td>
<td>-</td>
<td>3.4660$\times 10^{-3}$</td>
<td>-</td>
<td>3.6324$\times 10^{-3}$</td>
<td>-</td>
</tr>
<tr>
<td>(128, 128, 10  )</td>
<td>1.9331$\times 10^{-3}$</td>
<td>0.97</td>
<td>1.7626$\times 10^{-3}$</td>
<td>0.98</td>
<td>1.8478$\times 10^{-3}$</td>
<td>0.98</td>
</tr>
<tr>
<td>(128, 128, 20  )</td>
<td>9.7905$\times 10^{-4}$</td>
<td>0.98</td>
<td>8.9096$\times 10^{-4}$</td>
<td>0.98</td>
<td>9.3418$\times 10^{-4}$</td>
<td>0.98</td>
</tr>
<tr>
<td>(128, 128, 40  )</td>
<td>4.8947$\times 10^{-4}$</td>
<td>1.00</td>
<td>4.4999$\times 10^{-4}$</td>
<td>1.00</td>
<td>4.6666$\times 10^{-4}$</td>
<td>1.00</td>
</tr>
</tbody>
</table>

To illustrate the efficiency of ensemble algorithms, we take the second test case when $N_x = N_y = 256$ and $K = 40$ for example, and compare the execution time of ensemble simulations with 40 individual simulations using the same mesh and time
Table 3.5: The $L^2$ errors at final time: second-order ensemble, $Q_2/Q_1$ elements.

<table>
<thead>
<tr>
<th>$(N_x, N_y, K)$</th>
<th>$\mathcal{E}_1^N$</th>
<th>Rate</th>
<th>$\mathcal{E}_{20}^N$</th>
<th>Rate</th>
<th>$\mathcal{E}_{40}^N$</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(256, 256, 5)$</td>
<td>$1.0597 \times 10^{-3}$</td>
<td>–</td>
<td>$9.5139 \times 10^{-4}$</td>
<td>–</td>
<td>$9.9694 \times 10^{-4}$</td>
<td>–</td>
</tr>
<tr>
<td>$(256, 256, 10)$</td>
<td>$2.5992 \times 10^{-4}$</td>
<td>2.03</td>
<td>$2.3215 \times 10^{-4}$</td>
<td>2.03</td>
<td>$2.4328 \times 10^{-4}$</td>
<td>2.03</td>
</tr>
<tr>
<td>$(256, 256, 20)$</td>
<td>$6.3997 \times 10^{-5}$</td>
<td>2.02</td>
<td>$5.7010 \times 10^{-5}$</td>
<td>2.03</td>
<td>$5.9748 \times 10^{-5}$</td>
<td>2.02</td>
</tr>
<tr>
<td>$(256, 256, 40)$</td>
<td>$1.5905 \times 10^{-5}$</td>
<td>2.00</td>
<td>$1.4112 \times 10^{-5}$</td>
<td>2.01</td>
<td>$1.4796 \times 10^{-5}$</td>
<td>2.01</td>
</tr>
</tbody>
</table>

step sizes. In this case, the total number of degrees of freedom is $n_u = 132,098$ and $n_p = 16,641$. In individual simulations, GMRES algorithm is applied to solve discrete linear systems with the same stopping criterion as the aforementioned block GMRES, but each problem in the group is solved separately. We choose the convergence criteria of GMRES to be same as the BGMRES-D: $\text{maxit} = 50$ and $\text{tol} = 1 \times 10^{-8}$. Three individual simulations of problems with $j = 1, 20$ and $40$ have the following approximation errors:

$$
\mathcal{E}_1^N = 1.0377 \times 10^{-5}, \quad \mathcal{E}_{20}^N = 1.9140 \times 10^{-5}, \quad \mathcal{E}_{40}^N = 1.7075 \times 10^{-5}.
$$

Comparing with Table 3.5, we find that the accuracy of ensemble simulations is close to individual simulations. The corresponding simulation times of both approaches are listed in Table 3.6. It is shown that the time taken of ensemble simulations is above 10 times faster than the individual simulations.

Table 3.6: CPU time comparison.

<table>
<thead>
<tr>
<th>Iteration &amp; CPU time</th>
<th>BGMRES-D</th>
<th>40 GMRES</th>
</tr>
</thead>
<tbody>
<tr>
<td>average iteration number per time step</td>
<td>5</td>
<td>$5 \times 40$</td>
</tr>
<tr>
<td>average execution time per step (seconds)</td>
<td>19.658</td>
<td>$12.032 \times 40$</td>
</tr>
<tr>
<td>total CPU time for integration (seconds)</td>
<td>$1.965 \times 10^3$</td>
<td>$2.103 \times 10^4$</td>
</tr>
</tbody>
</table>

We observe that the iteration number in the block GMRES solver is the same as that of a single GMRES solver in individual simulations. But as the deflation is performed at each time step, the dimension of the RHS matrices should be no greater than $J$. The time evolution of its rank is plotted in Figure 3.2, which shows the maximum of the rank in this case is 8. This implies less matrix vector products are
evaluated in ensemble simulations than individual ones. On the other hand, the least-
square commutator preconditioning involves two discrete Poisson solves and matrix-
vector products. Since for the group of problems, the ensemble simulations only
apply one preconditioner to the coefficient matrix per iteration, while each individual
simulation would require one preconditioner to the coefficient matrix per iteration,
which leads to the significant computational savings.

Problem 3  Next, we consider a group of two-dimensional flow past a square cylin-
der problems, which are governed by Navier-Stokes equations in the domain \( \Omega = \Omega_0/\Omega_s \) over time interval \([0, 60]\) with \( \Omega_0 = [0, 8] \times [-1, 1] \) and interior obstacle
\( \Omega_s = [1.75, 2.25] \times [-0.25, 0.25] \). Dirichlet condition \( u = 1 - y^2, v = 0 \) is imposed at
the inflow boundary \((x = 0 \text{ and } -1 \leq y \leq 1)\), zero Dirichlet condition on the top
and bottom of the channel \((0 \leq x \leq 8 \text{ and } y = \pm 1)\), and do-nothing boundary on
the outflow boundary \((x = 8 \text{ and } -1 < y < 1)\). The flow is at rest at the initial time.
Viscosity coefficients vary among the problems, in particular, \( \nu_j = \frac{1}{300}(1 + \epsilon_j) \) with \( \epsilon_j \)
a random perturbation of magnitude 0.2 in the \( j \)-th problem.

For time integration, we use second order schemes with a uniform time step size
\( \Delta t = 0.01 \). As there is no analytic solutions, we only show the ensemble simulation
results together with individual ones, and compare the CPU time for both. Due
to the limit of space, we only show the simulation results at the final time for three
problems associated with Reynolds numbers: $Re_1 = 126.7001$, 176.9705 and 150.6167 in Figures 3.3-3.4. The time evolution of the speed field at a point $(6, 2)$ is plotted in Figures 3.5-3.6. The simulation results are so close that no obvious difference can be observed.

The time evolution of the dimension of RHSs in the ensemble simulation is shown in Figure 3.7. It is seen that the RHSs matrix size does shrink after deflations. The corresponding simulation times of both approaches are listed in Table 3.7. It is shown that, comparing with the individual simulations, the ensemble simulations saves 85% CPU time.

Table 3.7: CPU time comparison.

<table>
<thead>
<tr>
<th>Iteration &amp; CPU time</th>
<th>BGMRES-D</th>
<th>40 GMRES</th>
</tr>
</thead>
<tbody>
<tr>
<td>average iteration number per time step</td>
<td>7</td>
<td>8 × 40</td>
</tr>
<tr>
<td>average execution time per step (seconds)</td>
<td>4.7894</td>
<td>1.7313 × 40</td>
</tr>
<tr>
<td>total CPU time for integration (seconds)</td>
<td>7.106×10^4</td>
<td>4.836×10^5</td>
</tr>
</tbody>
</table>

3.5 Summary

The ensemble method has recently been developed for efficiently solving a group of evolution problems. It, using an ensemble-based time stepping, leads to a single linear system of multiple right-hand-side vectors for the entire group. Thus, all the problems can be solved simultaneously at each time step, which naturally share information among right hand sides. In this chapter, we demonstrate, for the first time, the efficacy of the ensemble method when it works together with block iterative solvers.
Figure 3.3: Speed field at $t = 60$ for $j = 1, 20,$ and $40$ in ensemble simulations.

Figure 3.4: Speed field at $t = 60$ for $j = 1, 20,$ and $40$ in individual simulations.
Figure 3.5: Time evolution of velocity magnitude for $j = 1, 20, \text{ and } 40$ in ensemble simulations.

Figure 3.6: Time evolution of velocity magnitude for $j = 1, 20, \text{ and } 40$ in individual simulations.
Figure 3.7: Evolution of the rank of RHS matrix at the beginning of all time steps
Grad-div stabilization results from adding $-\gamma \nabla (\nabla \cdot \mathbf{u})$ to the continuous NSE, which yields the term $\gamma (\nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)$ in a finite element formulation. Since $\nabla \cdot \mathbf{u}_h$ does not necessarily equal zero in finite element discretizations, the additional term acts as a penalty of lackness of mass conservation. Sparse grad-div stabilization results from adding $g(\mathbf{u})$ to the continuous NSE, which yields the term $\gamma (\nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h) - \gamma (\mathbf{u}_{1xh}, \nabla \cdot \mathbf{v}_h)$ in the finite element formulation. Thus the additional term can be interpreted as grad-div stabilization together with a pressure alteration $P = p + \gamma \mathbf{u}_{1x}$. Comparing to the grad-div stabilization, sparse grad-div stabilization creates a sparser linear algebraic system which could be solved more efficiently, and at the same time retains the penalty of lackness of mass conservation.

Since sparse grad-div stabilization is both a useful and a relatively new method, there is much interest in developing and understanding of it to maximize its effectiveness, in particular in the choice of the associated stabilization parameter $\gamma$. Theoretical analysis and numerical simulations originally proposed in [3] and later in [36] indicate that $\gamma = O(1)$ is often a good choice for standard grad-div stabilization. However, in [2], it is shown that the optimal $\gamma$ could be much larger than $O(1)$ in certain situations for standard grad-div stabilization, which depends on the size of the pressure relative to the size of velocity.

In this work, we developed the ensemble-based projection method with sparse grad-div stabilization. It is natural to find the optimal value of $\gamma$ to improve its numerical performance. Recall that in our numerical experiments, we solved the
problems associated to (2.52)-(2.54). We use a standard \((P_2, P_1)\) Taylor-Hood velocity-pressure elements for the spatial discretization in Step 1 + Step 2. A uniform triangle mesh containing 33282 velocity degrees of freedom and 4225 pressure degrees of freedom is used. Three problems are considered with perturbed initial and boundary conditions: \(\epsilon_1 = 0, \nu_1 = 0.01, \epsilon_2 = -0.002, \nu_2 = 0.005\) and \(\epsilon_3 = 0.002, \nu_3 = 0.015\); and the final time \(T = 0.1\).

For different time steps \(\Delta t = 0.005, 0.0025, 0.00125, 0.000625\), we plot the \(L^2\) and \(H^1\) velocity errors in Step 1 in Figure 4.1 and Figure 4.2 respectively as parameter \(\gamma\) varies from 0 to 0.2. It is observed that as the parameter \(\gamma\) increases, velocity errors first decrease then increase. Taking both \(L^2\) and \(H^1\) errors into consideration, \(\gamma = 0.05\) is an overall good choice. But we have not done any numerical analysis yet.
In the future, we are going to investigate the optimal choice of $\gamma$ theoretically and numerically for the ensemble-based projection method. We also plan to study the ensemble-based projection method working with different incremental pressure corrections later on in order to achieve better accuracy.

Figure 4.2: Error $||u - u_h||_{H^1}$ with different $\gamma$ and time step
BIBLIOGRAPHY


[43] J. Guermond, Un résultat de convergence d’ordre deux en temps pour l’approximation des équations de Navier-Stokes par une technique de projec-


