12-1-2021

Numerical Studies of Regularized Navier-Stokes Equations and an Application of a Run-to-Run Control model for Membrane Filtration at a Large Urban Water Treatment Facility

Jeffrey Belding

NUMERICAL STUDIES OF REGULARIZED NAVIER-STOKES EQUATIONS AND AN

APPLICATION OF A RUN-TO-RUN CONTROL MODEL FOR MEMBRANE

FILTRATION AT A LARGE URBAN WATER TREATMENT FACILITY

By

Jeffrey Belding

Bachelor of Science - Pure Mathematics and Physics
University of Nevada, Reno
2015

A dissertation submitted in partial fulfillment
of the requirements for the

Doctor of Philosophy - Mathematical Sciences

Department of Mathematical Sciences
College of Sciences
The Graduate College

University of Nevada, Las Vegas
December 2021

UNLV | GRADUATE COLLEGE

Dissertation Approval

The Graduate College
The University of Nevada, Las Vegas

November 9, 2021

This dissertation prepared by

Jeffrey Belding

entitled

Numerical Studies of Regularized Navier-Stokes Equations and an Application of a Run-to-Run Control model for Membrane Filtration at a Large Urban Water Treatment Facility

is approved in partial fulfillment of the requirements for the degree of

Doctor of Philosophy – Mathematical Sciences
Department of Mathematical Sciences

Monika Neda, Ph.D.
Examination Committee Chair

Jichun Li, Ph.D.
Examination Committee Member

Hongtao Yang, Ph.D.
Examination Committee Member

Pengtao Sun, Ph.D.
Examination Committee Member

Pushkin Kachroo, Ph.D.
Graduate College Faculty Representative

Kathryn Hausbeck Korgan, Ph.D.
*Vice Provost for Graduate Education &
Dean of the Graduate College*

ii

ABSTRACT

NUMERICAL STUDIES OF REGULARIZED NAVIER-STOKES EQUATIONS AND AN APPLICATION OF A RUN-TO-RUN CONTROL MODEL FOR MEMBRANE FILTRATION AT A LARGE URBAN WATER TREATMENT FACILITY

by

Jeffrey Belding

Dr. Monika Neda, Examination Committee Chair
Professor of Mathematics
University of Nevada, Las Vegas, USA

This dissertation consists of two parts. The first part consists of research on accurate and efficient turbulent fluid flow modeling via a family of regularizations of the Navier-Stokes equation which are known as Time Relaxation models. In the second part, we look into the modeling application for the filtration/backwash process at the River Mountains Water Treatment Facility in Henderson, NV.

In the first two chapters, we introduce the Time Relaxation models and their associated differential filter equations. In addition, we develop the regularization method which employs the Nth van Cittert deconvolution operator, which gives rise to the family of models. We also justify theoretically and computationally the use of an effective averaging length scale δ in the time relaxation model when using the van Cittert operator for higher orders of deconvolution N, by presenting experimental results from our use of this model in the Shear Layer Roll Up benchmarking problem. In addition, we will perform a sensitivity analysis with respect to the time relaxation coefficient χ which appears as a scaling factor for the regularization term in the model, and show how sensitivities with respect to χ are improved when utilizing the effective averaging length scale δ.

In the third chapter, we develop the time relaxation model with the newly proposed energy-

momentum-angular momentum conservation (EMAC) discretization of the non-linear term. We will present energy, momentum and angular momentum balances for the continuous formulation of the TRM with EMAC as well as the full discretized scheme using TRM with EMAC, and we will show that the fully discrete balances for TRM with EMAC reduce to the fully discrete analogues of the conservation of energy, momentum and angular momentum for the continuous Navier-Stokes equations under the assumption of no viscosity, no regularization, and no body force. In addition, we will present the stability and error estimate of the TRM with EMAC, and we will compare these results with the stability and error estimate for the TRM with the well known skew symmetric formulation for the non-linear term. We show that the error estimate for the EMAC scheme under high Reynolds number is much improved over the skew-symmetric scheme. In particular, we will show that the error for the EMAC scheme is $O(e^{\nu^{-1}})$, while under the same conditions, the skew-symmetric scheme error is $O(e^{\nu^{-3}})$, which is a significant improvement for high Reynolds number, i.e. low values of kinematic viscosity ν. We will then present numerical experiments on the Taylor Green vortex problem to verify the convergence rates for our error estimates, and experiments on the 3D Ethier-Steinman problem and the 2D Lattice Vortex problem to show that the numerical errors produced by EMAC are much smaller than the skew-symmetric scheme.

In the fourth chapter, We begin the work of part two by introducing the general run-to-run control model, which is used in a wide array of applications in addition to water treatment. Then we will introduce the specific run-to-run control model which is formulated specifically for a general filtration/backwash system, and we will modify it to fit the parameters, specifications, and measured data that is available from the River Mountains facility. In particular, we will discuss the implementation of a least squares problem to fit parameters for a filtration cycle ODE model. We will also discuss the backwashing component of the proposed model, and the difficulties in implementing such a model with real time plant data. From there, we formulate a cost of power

objective function for the whole filtration/backwashing cycle in terms of the setpoints of filtration/backwashing operation, namely the length of both filtration and backwashing cycles, and the setpoints for the fluid flux during each of these cycles. We show the process of minimizing this objective function with respect to the setpoints of plant operation, including our various implementations of this model using standard function optimization with constraints, genetic algorithms, and MCMC methods.

In the final chapter, we will draw some conclusions and summarize the findings of all the work contained in the dissertation.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PART I

THE TIME RELAXATION MODEL

CHAPTER 1

INTRODUCTION AND PRELIMINARIES

1.1 Time Relaxation Modeling and Literature Review

It is well known that fluid flows with high Reynolds number exhibit wildly chaotic behavior at many different scales. In turbulent flows, many interlaced eddies of various sizes develop overtime. In 1941, Kolmogorov gave a fairly simple, yet profound description of the eddies in turbulent flow [72]. In this description, he proposes that the largest eddies are deterministic in nature, while the smaller eddies are much more random in their evolution. The smaller eddies, ones below a certain length threshold, end up being dominated by viscous forces and die off very quickly as a result. This threshold, known as the Kolmogorov microscale, is $\eta = O(Re^{-3/4})$ for 3D simulations, and $\eta = O(Re^{-1/2})$ for 2D simulations. Accordingly, this requires the simulation to have a very fine mesh to resolve the smaller eddies which lie barely above this length scale. For instance, a direct numerical simulation in 3D would need

$$\triangle x = \triangle y = \triangle z = O(Re^{-3/4}),$$

which would require $O(Re^{9/4})$ mesh points in space per time step to fully resolve the structure of the eddies in any given turbulent flow. Obviously for simulations with very large Reynolds number, the associated computational cost is unacceptable, hence a lot of research has been done towards developing more useful numerical regularizations of the Navier Stokes equations (NSE). One such family of regularizations, the so called Time Relaxation model (TRM), has been proposed by Adams, Stolz, and Kleiser [9, 71], and we consider it within. The TRM equations are given in a domain $\Omega \subset \mathbb{R}^d$, d=2 or 3, and for $t > 0$ by

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \chi \boldsymbol{u}^* + \nabla p = \boldsymbol{f}, \text{ in } \Omega, \tag{1.1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \text{ in } \Omega, \tag{1.2}$$

where $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t)$ denotes the fluid velocity and $p = p(\boldsymbol{x}, t)$ is the fluid pressure, χ is the time relaxation coefficient, $\mathbf{f}$ is the body force, ν is the kinematic viscosity, and $\boldsymbol{u}^*$ is a regularization term defined more precisely later in Section 2 of this chapter. The aim of $\boldsymbol{u}^*$ is to drive the unresolved fluctuations of the fluid velocity to 0.

The original formulation of the TRM regularization term in Adams, Stolz, and Kleiser was inspired by regularizations of Chapman-Enskog expansions [27, 74], and it was constructed to introduce added energy dissipation to the approximate deconvolution models, which are a class of large eddy simulation models [44]. As a general regularization term, it has been useful for the Euler equations for shock-entropy wave interaction [7, 71, 9] and also for aerodynamic noise prediction and control [52]. Layton and Neda [77] applied these techniques to the NSE. The authors introduced the TRM given in (1.1)-(1.2), and studied the weak solutions. They also showed that solutions to (1.1)-(1.2) possess an energy cascade which terminates at the mesh scale δ with an appropriate choice of the relaxation coefficient χ. Continuous finite element studies have been performed in [55, 37, 54]. An analysis of discontinuous TRM methods for the NSE can be found in [17], and TRM with a non-linear term $\boldsymbol{u}^*$ is investigated in [56]. TRM has also been applied to flow ensembles in [76, 30]. In [68, 69, 32], sensitivity analysis was performed on the TRM to test for the sensitivity of the model with respect to the time relaxation parameter χ. A more comprehensive review of numerical studies on TRM can be found in [5].

1.2   Notation and Preliminaries

The $L^2(\Omega)$ norm and inner product will be denoted by $|| \cdot ||$ and $(\cdot, \cdot)$. Likewise, the $L^p(\Omega)$ norms

and the Sobolev $W_p^k(\Omega)$ norms are denoted by $||\cdot||_{L^p}$ and $||\cdot||_{W_p^k}$, respectively. For the semi-norm in $W_p^k(\Omega)$ we use $|\cdot|_{W_p^k}$. H^k is used to represent the Sobolev space W_2^k, and $||\cdot||_k$ denotes the norm in H^k. For functions $\boldsymbol{v}(\boldsymbol{x}, t)$ defined on the entire time interval $(0, T)$, we have

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

The velocity and pressure finite element spaces (X^h, Q^h) are defined respectively,

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

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

and the space of weakly divergence free velocity is

$$V^h = \{\boldsymbol{v} \in X^h : (q, \nabla \cdot \boldsymbol{v}) = 0, \forall q \in Q^h\}.$$

The dual space of X is denoted as X', with norm $||\cdot||_{-1}$.

We assume that the spaces X^h, Q^h satisfy the discrete inf-sup condition, i.e. there exists a $\gamma \in \mathbb{R}, \gamma > 0$, for which

$$\gamma \leq \inf_{q_h \in Q_h} \sup_{\boldsymbol{v}_h \in X_h} \frac{\int_\Omega q_h \nabla \cdot \boldsymbol{v}_h dA}{||q_h||_Q ||\boldsymbol{v}_h||_X}. \tag{1.3}$$

For notational clarity, let $v(t^{n+\frac{1}{2}}) = v(\frac{t^{n+1}+t^n}{2})$ for continuous variables, and then let $v^{n+\frac{1}{2}} = \frac{v^{n+1}+v^n}{2}$ for both continuous and discrete variables.

Let $\triangle t$ be the time step size so that $t_n = n\triangle t, n = 0, 1, 2, ..., N_T$, with final time $T := N_T \triangle t$. We then define the following discrete norms

$$|||v|||_{\infty,k} := \max_{0 \leq n \leq N_T} ||v^n||_k, \qquad\qquad |||v_{\frac{1}{2}}||| := \max_{1 \leq n \leq N_T} ||v^{n-\frac{1}{2}}||_k,$$

$$|||v|||_{m,k} = \left(\sum_{n=0}^{N_T} ||v^n||_k^m \triangle t \right)^{\frac{1}{m}}, \qquad\qquad |||v_{\frac{1}{2}}|||_{m,k} = \left(\sum_{n=1}^{N_T} ||v^{n-\frac{1}{2}}||_k^m \triangle t \right)^{\frac{1}{m}}.$$

We also will make use of the following approximation properties from [73],

$$\inf_{\boldsymbol{v} \in X_h} ||\boldsymbol{u} - \boldsymbol{v}|| \leq Ch^{k+1}||\boldsymbol{u}||_{k+1}, \boldsymbol{u} \in H^{k+1}(\Omega)^d,$$

$$\inf_{\boldsymbol{v} \in X_h} ||\boldsymbol{u} - \boldsymbol{v}||_1 \leq Ch^k||\boldsymbol{u}||_{k+1}, \boldsymbol{u} \in H^{k+1}(\Omega)^d, \tag{1.4}$$

$$\inf_{q \in Q_h} ||p - q|| \leq Ch^{s+1}||p||_{s+1}, p \in H^{s+1}(\Omega).$$

Moreover, the following consistency error formulas will be useful, all of which are standard and can be found in the appendix of [55].

Lemma 1.2.1. Assume $\boldsymbol{u} \in C^0(t_n, t_{n+1}; L^2(\Omega))$ and $\boldsymbol{u}_{tt} \in L^2((t_n, t_{n+1}) \times \Omega)$, then

$$||\boldsymbol{u}_{n+\frac{1}{2}} - \boldsymbol{u}(t_{n+\frac{1}{2}})||^2 \leq \frac{(\triangle t)^3}{48} \int_{t_n}^{t_{n+1}} ||\boldsymbol{u}_{tt}||^2 dt. \tag{1.5}$$

If $\nabla \boldsymbol{u} \in C^0(t_n, t_{n+1}; L^2(\Omega))$ and $\nabla \boldsymbol{u}_{tt} \in L^2((t_n, t_{n+1}) \times \Omega)$, then

$$||\nabla(\boldsymbol{u}_{n+\frac{1}{2}} - \boldsymbol{u}(t_{n+\frac{1}{2}}))||^2 \leq \frac{(\triangle t)^3}{48} \int_{t_n}^{t_{n+1}} ||\nabla \boldsymbol{u}_{tt}||^2 dt. \tag{1.6}$$

If $\boldsymbol{u}_t \in C^0(t_n, t_{n+1}; L^2(\Omega))$ and $\boldsymbol{u}_{ttt} \in L^2((t_n, t_{n+1}) \times \Omega)$, then

$$||\frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\triangle t} - \boldsymbol{u}_t(t_{n+\frac{1}{2}})||^2 \leq \frac{(\triangle t)^3}{1280} \int_{t_n}^{t_{n+1}} ||\boldsymbol{u}_{ttt}||^2 dt. \tag{1.7}$$

Next, we define the following three trilinear forms $b(\cdot, \cdot, \cdot) : X \times X \times X \to \mathbb{R}$, $b^*(\cdot, \cdot, \cdot) : X \times X \times X \to \mathbb{R}$ and $c(\cdot, \cdot, \cdot) : X \times X \times X \to \mathbb{R}$,

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{u} \cdot \nabla \boldsymbol{v}, \boldsymbol{w}), \tag{1.8}$$

$$b^*(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \frac{1}{2}b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) - \frac{1}{2}b(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}), \tag{1.9}$$

$$c(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = 2(\boldsymbol{D}(\boldsymbol{u})\boldsymbol{v}, \boldsymbol{w}) + ((\nabla \cdot \boldsymbol{u})\boldsymbol{v}, \boldsymbol{w}), \tag{1.10}$$

5

where $\boldsymbol{D}(\boldsymbol{u})$ is defined as the symmetric part of $\nabla\boldsymbol{u}$ by $\boldsymbol{D}(\boldsymbol{u}) = \frac{\nabla\boldsymbol{u}+(\nabla\boldsymbol{u})^T}{2}$. These trilinear forms will be used to represent the non-linear term in the TRM model for both the SKEW and EMAC schemes.

Lemma 1.2.2. For $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in X$, and also $\boldsymbol{v} \in L^{\infty}(\Omega)$ for the first estimate, we have the following bounds for the triple product $b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{u} \cdot \nabla\boldsymbol{v}, \boldsymbol{w})$,

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \leq \frac{1}{2} \left(||\boldsymbol{u}||\ ||\nabla\boldsymbol{v}||_{\infty}\ ||\boldsymbol{w}|| + ||\boldsymbol{u}||\ ||\boldsymbol{v}||_{\infty}\ ||\nabla\boldsymbol{w}|| \right), \tag{1.11}$$

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \leq C(\Omega)||\boldsymbol{u}||^{\frac{1}{2}}||\nabla\boldsymbol{u}||^{\frac{1}{2}}||\nabla\boldsymbol{v}||\ ||\nabla\boldsymbol{w}||, \tag{1.12}$$

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \leq C(\Omega)||\nabla\boldsymbol{u}||\ ||\nabla\boldsymbol{v}||\ ||\nabla\boldsymbol{w}||. \tag{1.13}$$

By definition of our trilinear form b^*, it is clear that b^* also satisfies the above bounds in place of b.

For the trilinear forms b, we also have the following algebraic properties which are derived in [58].

Lemma 1.2.3. The following properties hold for any $\boldsymbol{u} \in H_0^1(\Omega)$, $\boldsymbol{v}, \boldsymbol{w} \in H^1(\Omega)$,

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = -b(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}) - ((\nabla \cdot \boldsymbol{u})\boldsymbol{v}, \boldsymbol{w}), \tag{1.14}$$

$$b(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{w}) = -\frac{1}{2}((\nabla \cdot \boldsymbol{u})\boldsymbol{w}, \boldsymbol{w}), \tag{1.15}$$

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = ((\nabla\boldsymbol{v})\boldsymbol{w}, \boldsymbol{u}) = ((\nabla\boldsymbol{v})^T\boldsymbol{u}, \boldsymbol{w}), \tag{1.16}$$

$$(\boldsymbol{D}(\boldsymbol{u})\boldsymbol{u}, \boldsymbol{u}) = ((\nabla\boldsymbol{u})\boldsymbol{u}, \boldsymbol{u}) = b(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{u}), \tag{1.17}$$

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = 2\boldsymbol{D}(\boldsymbol{u})\boldsymbol{u} - \nabla \left(\frac{1}{2}|\boldsymbol{u}|^2 \right). \tag{1.18}$$

In the error analysis, we will use a discrete Gronwall inequality, the proof of which can be found in [29].

Lemma 1.2.4. (Discrete Gronwall Lemma) Let $\triangle t$, H, and a_n, b_n, c_n, d_n (for integers $n \geq 0$) be finite non-negative numbers such that

$$a_l + \triangle t \sum_{n=0}^{l} b_n \leq \triangle t \sum_{n=0}^{l} d_n a_n + \triangle t \sum_{n=0}^{l} c_n + H \text{ for } l \geq 1. \tag{1.19}$$

Suppose that $d_n \triangle t < 1$ for all n, and set $\sigma_n = (1 - d_n \triangle t)^{-1}$. Then,

$$a_l + \triangle t \sum_{n=0}^{l} b_n \leq \exp\left(\triangle t \sum_{n=0}^{l} \sigma_n d_n\right)\left(\triangle t \sum_{n=0}^{l} c_n + H\right) \text{ for } l \geq 1. \tag{1.20}$$

Finally, we will also make use of the L^2 projection and the Stokes projection onto X_h in the error analysis, so we define them next.

Definition 1.2.5. The L^2 projection, $P_{L^2} : X \to X_h$, $P_{L^2}(\boldsymbol{v}) = \tilde{\boldsymbol{v}}$ satisfies the following

$$(\boldsymbol{v} - \tilde{\boldsymbol{v}}, \boldsymbol{w}) = 0, \forall \boldsymbol{w} \in X_h. \tag{1.21}$$

Definition 1.2.6. The Stokes projection, $P_{St} : (X, Q) \to (X_h, Q_h)$, $P_{St}(\boldsymbol{v}, p) = (\tilde{\boldsymbol{v}}, \tilde{p})$ satisfies the following

$$(\nabla(\boldsymbol{v} - \tilde{\boldsymbol{v}}), \boldsymbol{w}) - (p - \tilde{p}, \nabla \cdot \boldsymbol{w}) = 0, \forall \boldsymbol{w} \in X_h, \tag{1.22}$$

$$(\nabla \cdot (\boldsymbol{v} - \tilde{\boldsymbol{v}}), q_h) = 0, \forall q_h \in Q_h. \tag{1.23}$$

For the Stokes projection, we will need two results, a stability lemma and an error estimate. The stability lemma was proved in [45], while the error estimate was shown in [75] in conjunction with the Aubin-Nitsche lift technique [57].

Lemma 1.2.7. Given $\boldsymbol{w} \in X$, the following inequality holds,

$$\|\nabla P_{St} \boldsymbol{w}\|_{L^r} \leq C \|\nabla \boldsymbol{w}\|_{L^r}, r \in [2, \infty). \tag{1.24}$$

7

Lemma 1.2.8. Given $(\boldsymbol{w}, p)$ and it's Stokes projection $(\tilde{\boldsymbol{w}}, \tilde{p})$, the following error estimate holds,

$$||\boldsymbol{w} - \tilde{\boldsymbol{w}}|| \leq Ch \left(\inf_{\boldsymbol{v}_h \in X_h} ||\nabla(\boldsymbol{w} - \boldsymbol{v}_h)|| + \inf_{q_h \in Q_h} ||p - q_h|| \right), \tag{1.25}$$

with $C = C(\nu, \Omega)$.

1.2.1 Filter and Deconvolution

In this section, we introduce the deconvolution methods and give a precise definition for $\boldsymbol{u}^*$ in equation (1.1). The idea relies on selecting a resolution scale δ which is feasible for the application, and using a differential filter [16]. Thus we can write the family of TRM as

$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \triangle \boldsymbol{u} + \chi(\boldsymbol{u} - G_N \overline{\boldsymbol{u}}) + \nabla p &= \mathbf{f}, \text{ in } \Omega, \\ \nabla \cdot \boldsymbol{u} &= 0, \text{ in } \Omega, \\ \boldsymbol{u}(x, 0) &= \boldsymbol{u}_0, \text{ in } \Omega. \end{aligned} \tag{1.26}$$

In the above, $\boldsymbol{u}$ is the true velocity solution to our TRM model, while $\overline{\boldsymbol{u}}$ represents a spatially averaged representation of $\boldsymbol{u}$, given by the following differential equation

$$\begin{aligned} -\delta^2 \Delta \overline{\boldsymbol{u}} + \overline{\boldsymbol{u}} &= \boldsymbol{u}, \text{ in } \Omega, \\ \overline{\boldsymbol{u}} &= \mathbf{0}, \text{ on } \partial\Omega. \end{aligned} \tag{1.27}$$

In the above filtering equation, δ is a constant corresponding to the filter width, or averaging length scale. Thus for large δ values, $\overline{\boldsymbol{u}}$ is smooth and for small δ values, $\overline{\boldsymbol{u}}$ is close to $\boldsymbol{u}$. Setting $\overline{\boldsymbol{u}} = G(\boldsymbol{u})$ and solving for G, we obtain the operator $G := (-\delta^2 \triangle + I)^{-1}$. We rewrite the filtering equation $\overline{\boldsymbol{u}} = G(\boldsymbol{u})$ as the following fixed point problem: Given $\overline{\boldsymbol{u}}$, solve $\boldsymbol{u} = \boldsymbol{u} + (\overline{\boldsymbol{u}} - G\boldsymbol{u})$, which gives the following deconvolution approximation

$$\boldsymbol{u}_0 = \overline{\boldsymbol{u}},$$

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + (\overline{\boldsymbol{u}} - G\boldsymbol{u}_n), \quad n = 1, 2, 3, ..., N-1.$$

This deconvolution algorithm is precisely the first order Richardson iteration for the operator $G : (L^2(\Omega))^d \longrightarrow (L^2(\Omega))^d$, which is possibly non-invertible. The deconvolution problem is ill-

posed, so we cannot expect such an algorithm to converge as $N \to \infty$ [39]. Van Cittert also studied

this deconvolution problem in 1931 and gave a more compact formula for it defined next.

Definition 1.2.9. The Nth van Cittert approximate deconvolution operator $G_N : (L^2(\Omega))^d \longrightarrow$

$(L^2(\Omega))^d$ is

$$G_N \overline{u} := \sum_{n=0}^{N} (I - G)^n \overline{u}, \quad N = 0, 1, 2, ...$$ (1.28)

For convenience, we list the first few values of the G_N operator here.

$$G_0 \overline{u} = \overline{u},$$

$$G_1 \overline{u} = 2\overline{u} - \overline{\overline{u}}, \qquad (\text{ where } \overline{\overline{u}} = G\overline{u}),$$

$$G_2 \overline{u} = 3\overline{u} - 3\overline{\overline{u}} + \overline{\overline{\overline{u}}}, \qquad (\text{ where } \overline{\overline{\overline{u}}} = G\overline{\overline{u}}).$$

The operator G_N has been shown to be an $O(\delta^{2N+2})$ inverse approximate to the filter operator

G in [44], hence for the smaller length scales δ, the higher order operators give a more accurate

approximation for u.

1.2.2   Discrete Deconvolution Filter

Following the work of Manica and Merdan [15], we define the discrete differential filter as follows:

For $u \in L^2(\Omega)^d$ and a given filter width $\delta > 0$, we let $G_h : L^2(\Omega)^d \to X_h$ be defined by $G_h(u) := \overline{u}^h$,

where $\overline{u}^h \in X_h$ is the unique solution of the following variational problem

$$\delta^2 (\nabla \overline{u}^h, \nabla v_h) + (\overline{u}^h, v_h) = (u, v_h), \forall v_h \in X_h.$$ (1.29)

One useful alternative to the above definition is to define the differential filter by a discrete

Stokes problem, which will weakly preserve the fluid incompressibility in the case of internal flows

under no slip boundary conditions [22, 14]. In this case, for $u \in L^2(\Omega)^d, \delta > 0$ given, we let

$G_h : L^2(\Omega)^d \to X_h$ be defined by $G_h(\boldsymbol{u}) := \overline{\boldsymbol{u}}^h$, where $(\overline{\boldsymbol{u}}^h, \rho) \in X_h \times Q_h$ is the unique solution of the variational problem,

$$\delta^2 (\nabla \overline{\boldsymbol{u}}^h, \nabla \boldsymbol{v}_h) + (\overline{\boldsymbol{u}}^h, \boldsymbol{v}_h) - (\rho, \nabla \cdot \boldsymbol{v}_h) = (\boldsymbol{u}, \boldsymbol{v}_h), \forall v_h \in X_h,$$

$$(\nabla \cdot \overline{\boldsymbol{u}}^h, q_h) = 0, \forall q_h \in Q_h.$$

Once the above is defined, we can then proceed with a similar procedure to the continuous case and define the Nth van Cittert discrete deconvolution operator as follows,

$$G_N^h(\boldsymbol{\phi}) := \sum_{n=0}^{N} (I - G_h)^n \boldsymbol{\phi}. \tag{1.30}$$

The operator G_N^h is algebraically similar to the operator G_N, and it can be shown it also is an $O(\delta^{2N+2})$ inverse approximate to G_h by a nearly identical proof to the one for G_N shown in [44]. We have the following lemma from [37], which will be used in the error analysis in Chapter 3.

Lemma 1.2.10. For smooth $\boldsymbol{\phi}$, the Nth order discrete deconvolution operator satisfies

$$||\boldsymbol{\phi} - G_N^h G_h \boldsymbol{\phi}|| \le C_1 \delta^{2N+2} ||\boldsymbol{\phi}||_{H^{2N+2}} + C_2 \left(\delta h^k + h^{k+1} \right) ||\boldsymbol{\phi}||_{k+1}. \tag{1.31}$$

We will assume throughout this dissertation that the discrete filter and deconvolution operators satisfy the following assumptions which are proved by Stanculescu [26].

Lemma 1.2.11. For the operators G_N^h and G_h, the following assumptions hold.

Assumption 1: G_N^h and G_h are symmetric, positive definite operators and preserve discrete incompressibility.

Assumption 2: $||G_N^h G_h||_{\mathcal{L}:L^2 \to L^2} \le 1$ and $||I - G_N^h G_h||_{\mathcal{L}:L^2 \to L^2} \le 1$.

Assumption 3: $(I - G_N^h G_h)$ and $G_N^h G_h$ are symmetric, positive definite operators.

By Assumption 3 above, we can define the following norm based on the existence of a square root operator for $I - G_N^h G_h$,

$$||\phi^*|| = \left((I - G_N^h G_h)\phi, \phi \right)^{\frac{1}{2}}. \tag{1.32}$$

CHAPTER 2

SENSITIVITY ANALYSIS OF THE TIME RELAXATION MODEL

In this chapter, we will investigate the sensitivity of the TRM to the time relaxation parameter χ. The time relaxation parameter is considered to be a positive quantity, i.e. $\chi > 0$, and has units of $1/time$. The term $\chi(\mathbf{u} - G_N \bar{\mathbf{u}})$ aims to drive the unresolved scales to zero exponentially [55]. In working with TRM, the parameter χ must be specified and scaled appropriately in relation to other parameters in the problem [77]. In particular, we will show in this chapter the advantage of using an effective averaging length scale which modifies the chosen resolution scale δ for a particular problem as you increase the order of deconvolution N.

2.1   The Effective Averaging Length Scale

In this section, based on Olson's work in [46], we identify the effective averaging length scale in the TRM model as $\delta_N = \delta_0 \sqrt{N + 1}$, where δ_0 is a fixed length scale suited for the $N = 0$ version of our model.

We consider the effects of the deconvolution filter for the time relaxation term $\chi(u - G_N \bar{u})$ on regular 2π-periodic functions with a spatial average of 0 in Fourier Space. Hence, we can write the following forms for u and $\bar{u}$,

$$u(x,t) = \sum_{k \in \mathbf{Z}^2 \backslash 0} u_k(t) e^{ik \cdot x} \quad \text{and} \quad \bar{u}(x,t) = \sum_{k \in \mathbf{Z}^2 \backslash 0} \bar{u}_k(t) e^{ik \cdot x}.$$

It follows that the differential filter (1.27) may be written as

$$\bar{u}_k = G_{N,k}(1 - \delta^2 |k|^2)^{-1} u_k,$$

where

$$G_{N,k} = \sum_{n=0}^{N} \left(1 - \frac{1}{1+\delta^2|k|^2} \right)^n = \sum_{n=0}^{N} \left(\frac{\delta^2|k|^2}{1+\delta^2|k|^2} \right)^n.$$

Noting that the above is a geometric series, we sum up the series to obtain

$$\frac{G_{N,k}}{1+\delta^2|k|^2} = 1 - \left(\frac{\delta^2|k|^2}{1+\delta^2|k|^2} \right)^{N+1}.$$

For the LHS of above, we note the following limits.

$$\frac{G_{N,k}}{1+\delta^2|k|^2} \to 1 \ \text{ as } \ N \to \infty \ \text{ for fixed } \ \delta, \ \text{ or as } \ \delta \to 0 \ \text{ for fixed N.}$$

Also,

$$\frac{G_{N,k}}{1+\delta^2|k|^2} \to 0 \ \text{ as } \ \delta \to \infty \ \text{ for fixed } \ N.$$

These limits suggest that there is a relationship between δ and N which can be used to identify an effective averaging length scale that leaves the small scale attenuation of the smoothing filter unchanged as we increase the order of convolution N. Now we do asymptotic analysis on the term $G_{N,k}/(1+\delta^2|k|^2)$ as $k \to \infty$ to determine this relationship for the averaging length scale. We have that

$$\frac{G_{N,k}}{1+\delta^2|k|^2} = 1 - \left(\frac{\delta^2|k|^2}{1+\delta^2|k|^2} \right)^{N+1} \sim \frac{N+1}{\delta^2|k|^2} \ \text{ for large } k.$$

From this behavior, it is clear that if we scale our δ for higher N values as $\delta_N = \delta_0 \sqrt{N+1}$ for some chosen constant δ_0 for $N = 0$, we obtain an asymptotic decay for the above expression that is independent of N as $k \to \infty$. In this chapter, we will see how scaling our resolution parameter δ in this manner is crucial for resolving the large scale structures when running TRM with higher orders of deconvolution. Clearly this scaling is not present when the order of deconvolution is 0,

but in experiments with even deconvolution orders of 1 or 2, we will see that this scaling allows the large scale structures in fluid flows to be resolved very clearly, while the same simulations without scaling will display the same structures with a lot of noise and interference.

2.2 Variational Formulation and the Computational Algorithm

The variational formulation of TRM based on the set of equations (1.1)-(1.2) and (1.27) using a suitable choice of test functions from X and Q are respectively given as: Find $(\mathbf{u}, p) \in (X, Q)$ such that

$$(\mathbf{u}_t, \mathbf{v}) + \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) + b^*(\mathbf{u}, \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) + \chi(\mathbf{u} - G_N \bar{\mathbf{u}}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in X, \qquad (2.1)$$

$$(\nabla \cdot \mathbf{u}, q) = 0, \quad \forall q \in Q, \qquad (2.2)$$

$$\delta^2(\nabla \bar{\mathbf{u}}, \nabla \mathbf{v}) + (\bar{\mathbf{u}}, \mathbf{v}) = (\mathbf{u}, \mathbf{v}), \quad \forall \mathbf{v} \in X. \qquad (2.3)$$

The method of Crank-Nicolson is used for time discretization, and we also use the skew-symmetric formulation for discretizing the non-linear term. Our notations in the time discretization equations of TRM read as following: $v(t^{n+1/2}) = v((t^{n+1} + t^n)/2)$ for the continuous variable and $v^{n+1/2} = (v^{n+1} + v^n)/2$ for both, continuous and discrete variables. Thus, the fully discretized finite element variational formulation of TRM is written as follows.

Given (X^h, Q^h), the time interval $[0, T]$, the time step chosen as $\triangle t < T = M \triangle t$, find the approximated TRM solution $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in (X^h, Q^h)$, for $n = 0, 1, 2....M - 1$ satisfying

$$\frac{1}{\triangle t}(\mathbf{u}_h^{n+1} - \mathbf{u}_h^n, \mathbf{v}_h) + \nu(\nabla \mathbf{u}_h^{n+1/2}, \nabla \mathbf{v}_h) + b^*(\mathbf{u}_h^{n+1/2}, \mathbf{u}_h^{n+1/2}, \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h)$$

$$+ \chi(\mathbf{u}_h^{n+1/2} - G_N^h \overline{\mathbf{u}_h}^{n+1/2h}, \mathbf{v}_h) = (\mathbf{f}^{n+1/2}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in X^h, \qquad (2.4)$$

$$(\nabla \cdot \mathbf{u}_h^{n+1}, q_h) = 0, \quad \forall q_h \in Q^h, \qquad (2.5)$$

$$\delta^2(\nabla \bar{\mathbf{u}}_h^{n+1}, \nabla \mathbf{v}_h) + (\bar{\mathbf{u}}_h^{n+1}, \mathbf{v}_h) = (\mathbf{u}_h^{n+1}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in X^h. \qquad (2.6)$$

In the space V^h, equations (2.4)-(2.6) are equivalently rewritten as follows. Find $\mathbf{u}_h^{n+1} \in V^h$, for $n = 0, 1, 2....M - 1$ satisfying

$$\frac{1}{\triangle t}(\mathbf{u}_h^{n+1} - \mathbf{u}_h^n, \mathbf{v}_h) + \nu(\nabla \mathbf{u}_h^{n+1/2}, \nabla \mathbf{v}_h) + b^*(\mathbf{u}_h^{n+1/2}, \mathbf{u}_h^{n+1/2}, \mathbf{v}_h)$$

$$+\chi(\mathbf{u}_h^{n+1/2} - G_N^h \overline{\mathbf{u}_h^{n+1/2}}^h, \mathbf{v}_h) = (\mathbf{f}^{n+1/2}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in V^h, \tag{2.7}$$

$$\delta^2(\nabla \bar{\mathbf{u}}_h^{n+1}, \nabla \mathbf{v}_h) + (\bar{\mathbf{u}}_h^{n+1}, \mathbf{v}_h) = (\mathbf{u}_h^{n+1}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in V^h. \tag{2.8}$$

The existence, stability, and covergence analysis of the finite element solution of the above schemes are investigated in [55]. In our computations, the non-linear term $b^*(\mathbf{u}_h^{n+1/2}, \mathbf{u}_h^{n+1/2}, \mathbf{v}_h)$ in equation (2.7) is approximated by the method of fixed point iteration. We implemented the Stokes filter for all our computations.

2.3 Computational Results

This section contains computations for approximated velocity using TRM with deconvolution orders of $N =$0, 1, and 2 for the so called shear layer roll-up test problem. The sensitivity computations with respect to the variations of time relaxation parameter χ for all the tested deconvolution orders are performed as well. All simulations in this section were performed using the Freefem++ open source PDE solver package. Details on this package can be found here [49].

2.3.1 Shear Layer Roll-up Problem

This is a benchmark problem involving a doubly periodic pair of shear layers studied in [28, 6, 62]. The initial flow field consists of a horizontal shear layer of finite thickness, perturbed by a small amplitude vertical velocity. Each of the shear layers forming the boundaries of the initial jet will eventually evolve into a periodic array of large vortices. Meanwhile, the shear layers between the rolls are thinned out by the large straining field, which then causes these layers to wrap around the large rolls. An exact solution is not known for this problem, thus a reference solution is generated

for the problem using a very fine mesh and will be used to test for the accuracy of our simulations.

The domain is $\Omega = [0,1]^2$, and initial conditions are given by

$$u = \begin{cases} \tanh(\rho(y - 0.25)) & \text{if } y \leq 0.5, \\ \tanh(\rho(0.75 - y)) & \text{if } y > 0.5, \end{cases}$$

$$v = \delta \sin(2\pi x).$$
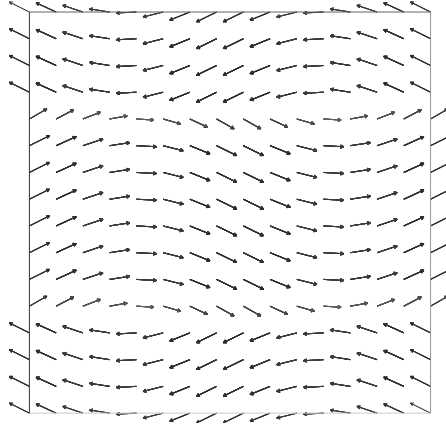
Figure 2.1: Initial Velocity field for the Shear Layer problem, with $\rho = 80$, $\delta = 0.5$.

Doubly-periodic boundary conditions are applied on our domain Ω. We let our time step be $\triangle t = 0.004$ and Reynolds number $Re = 10,000$ (i.e. $\nu = 1/10,000$). For all simulations, we used Taylor-Hood finite elements and a uniform triangular mesh. The time relaxation parameters used are $\chi = 0.1$ and $\chi = 1$, the shear layer width parameter is $\rho = 80$, while the filter radius is given by the scaled values $\delta_N = \delta_0\sqrt{N+1}$ with $\delta_0 = \frac{5}{m}$, where m denotes the number of subintervals on each side of the square domain Ω. Thus, the filter width for TRM with order of deconvoltuion $N = 0$ is denoted by δ_0, for TRM with $N = 1$ the scaled filter width is given by $\delta_1 = \delta_0\sqrt{2}$, and for TRM with $N = 2$ we have the scaled filter width $\delta_2 = \delta_0\sqrt{3}$. The initial condition vector field is plotted in Figure 2.1.

We present the true unfiltered vorticity solution (i.e. Navier-Stokes solution) obtained on a fine mesh with $m = 250$ in Figure 2.2. Then, the simulations from Figures 2.3 - 2.6 are on a coarse

mesh with $m = 100$. The vorticity plots from Figures 2.3 and 2.5 show the advantage of applying the time relaxation modelling in comparison with no filtering and deconvolution, i.e. Navier-Stokes solution that has more of spurious vortices on the coarse mesh, especially as the vorticity develops in time. In Figures 2.4 and 2.6 we can see the advantage of using scaled values for the filter width over the constant value δ_0 as we increase the order of deconvolution N. The produced vorticity plots for scaled values of the filter width had less noise and increased accuracy comparing to the true solution.

Figure 2.2: Vorticity for the NSE on a fine mesh (m=250)

(a) $T = 0.2$

(b) $T = 0.4$

(c) $T = 0.6$

(d) $T = 0.8$

(e) $T = 1.0$

(f) $T = 1.2$
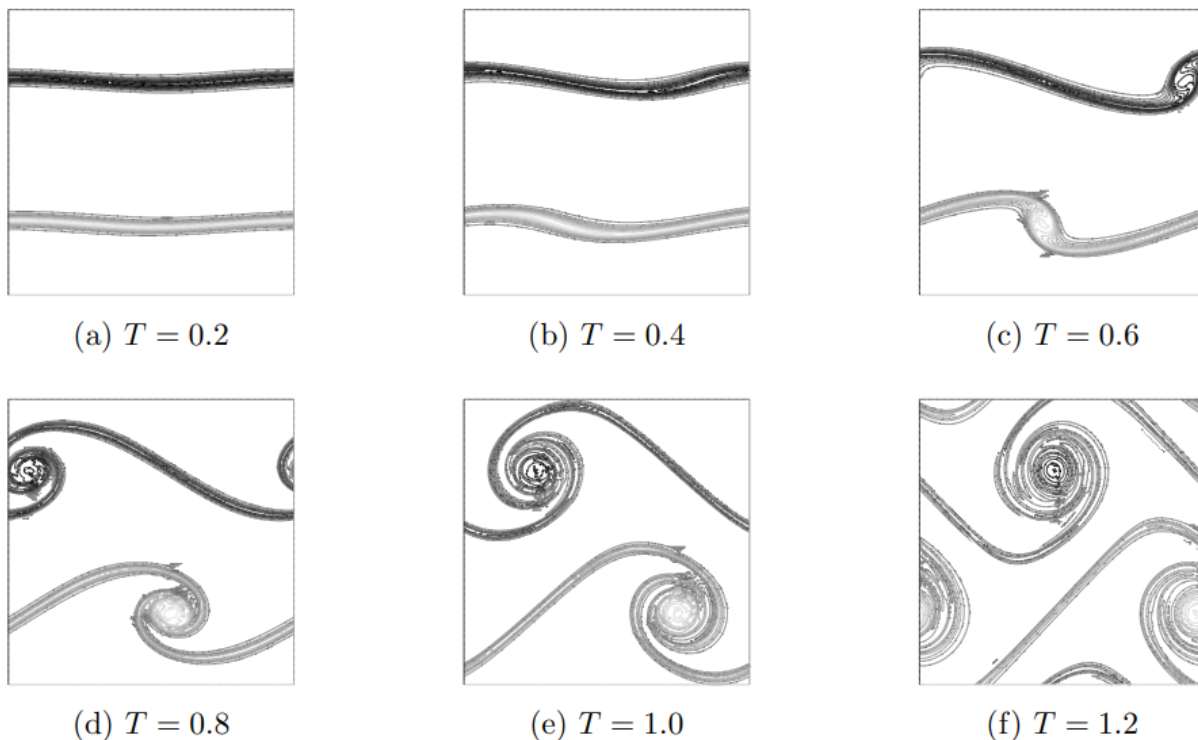
2.3.2 Parameter Sensitivity Computations

In this section, we provide a computational assessment of the parameter sensitivity of TRM model with respect to the variation of χ values for the van Cittert deconvolution operator of orders $N = 0, 1$, and 2. A detailed analysis of TRM sensitivity with respect to the time relaxation

Figure 2.3: Vorticity for NSE Vs. TRM for various N, $\chi = 0.1$, scaled $\delta_N = \delta_0\sqrt{N+1}$

(a) NSE, $T = 0.8$ (b) $N = 0, T = 0.8$ (c) $N = 1, T = 0.8$ (d) $N = 2, T = 0.8$

(e) NSE, $T = 1.0$ (f) $N = 0, T = 1.0$ (g) $N = 1, T = 1.0$ (h) $N = 2, T = 1.0$

(i) NSE, $T = 1.2$ (j) $N = 0, T = 1.2$ (k) $N = 1, T = 1.2$ (l) $N = 2, T = 1.2$
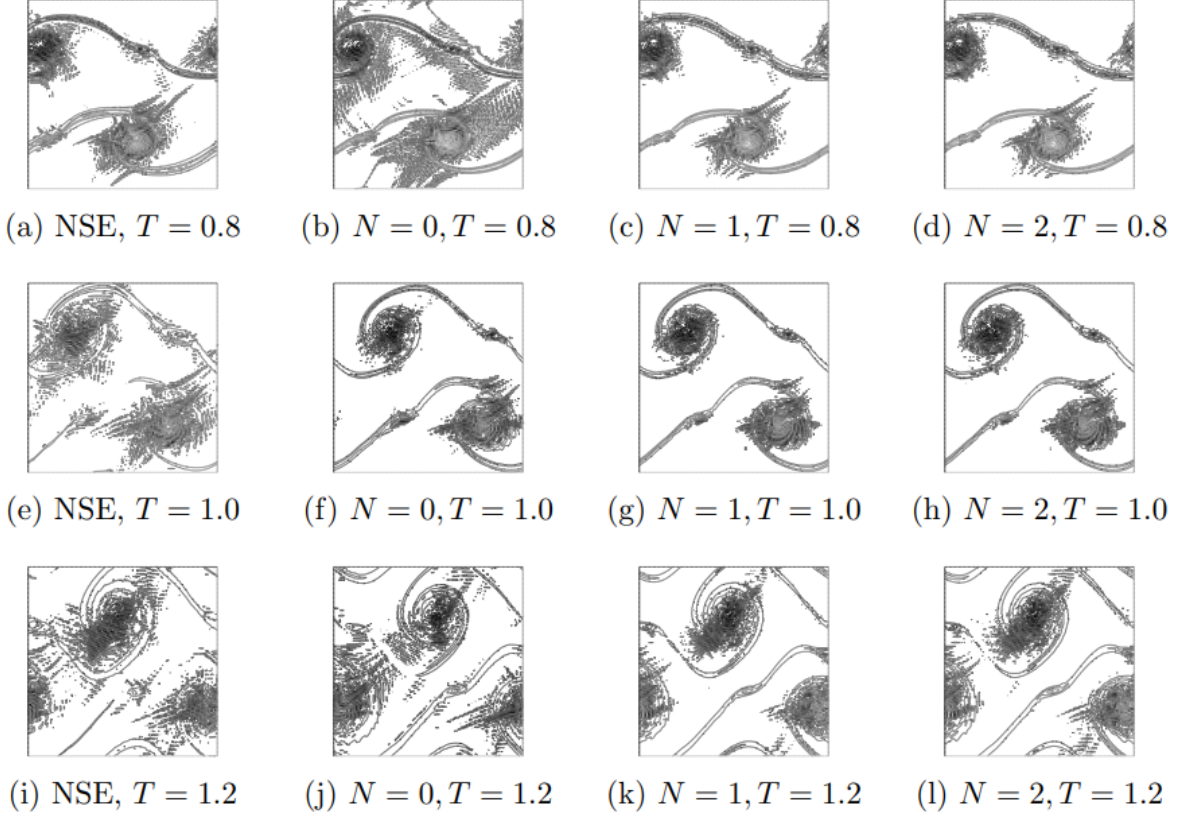
parameter χ using the Continuous Sensitivity Equation Method, CSEM, for $N = 0$ can be found in [68]. We derive the sensitivity equations by differentiating the TRM equations (1.26) with respect to χ. Thus given $(\mathbf{u}, p)$, we find $(\mathbf{s}, r)$ such that

$$\mathbf{s}_t + \mathbf{u} \cdot \nabla\mathbf{s} + \mathbf{s} \cdot \nabla\mathbf{u} + \nabla r - \nu\triangle\mathbf{s} + (\mathbf{u} - G_N\bar{\mathbf{u}}) + \chi(\mathbf{s} - G_N\mathbf{w}) = \boldsymbol{0}, \text{ in} \times [0, T],$$

$$\nabla \cdot \mathbf{s} = 0, \text{ in } \Omega \times [0, T],$$

$$\mathbf{s} = 0, \text{ on } \partial\Omega \times [0, T]. \qquad (2.9)$$

where $\mathbf{s} = \dfrac{\partial\mathbf{u}}{\partial\chi}$, $r = \dfrac{\partial p}{\partial\chi}$ and $\mathbf{w} = \dfrac{\partial\bar{\mathbf{u}}}{\partial\chi}$. Here, $\mathbf{w}$ satisfies the following sensitivity filtering equation,

Figure 2.4: Vorticity for $N = 1, 2$, $\chi = 0.1$, unscaled δ_0 and scaled δ_N, at times $T = 0.8$ (first row), $T = 1.0$ (second row), and $T = 1.2$ (third row)

(a) $N = 1, \delta_0$ (b) $N = 1, \delta_1$ (c) $N = 2, \delta_0$ (d) $N = 2, \delta_2$

(e) $N = 1, \delta_0$ (f) $N = 1, \delta_1$ (g) $N = 2, \delta_0$ (h) $N = 2, \delta_2$

(i) $N = 1, \delta_0$ (j) $N = 1, \delta_1$ (k) $N = 2, \delta_0$ (l) $N = 2, \delta_2$
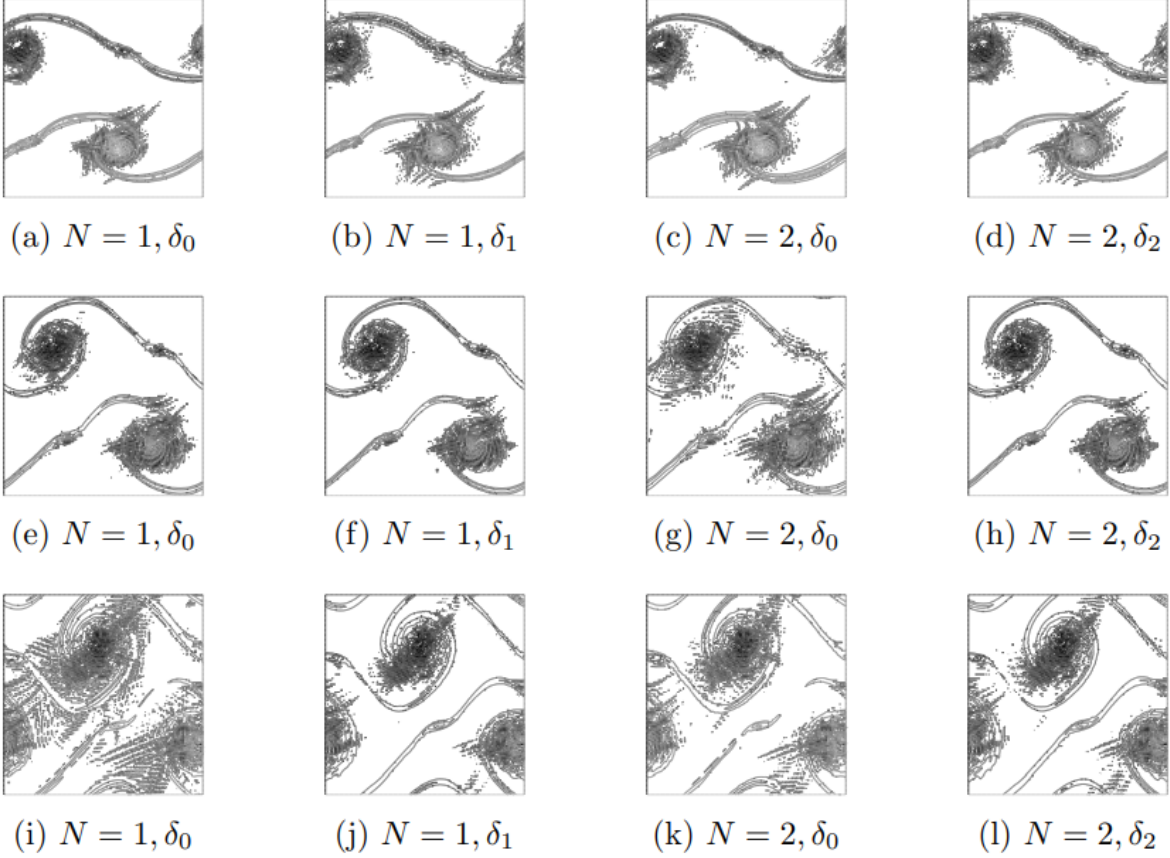
$$-\delta^2 \triangle \mathbf{w} + \mathbf{w} = \mathbf{s}, \text{in } \Omega,$$

$$\mathbf{w} = 0, \text{on } \partial\Omega. \tag{2.10}$$

In order to obtain the solution for (2.9) we need to couple (1.26) with (2.9), as $\mathbf{u}$ appears in the sensitivity equation. The fully discretized variational formulation for the sensitivity using Crank-Nicolson is to find $\mathbf{s}_h^{n+1} \in V^h$ satisfying:

$$\frac{1}{\triangle t}(\mathbf{s}_h^{n+1} - \mathbf{s}_h^n, \mathbf{v}_h) + \nu a(\mathbf{s}_h^{n+1/2}, \mathbf{v}_h) + b^*(\mathbf{s}_h^{n+1/2}, \mathbf{u}_h^{n+1/2}, \mathbf{v}_h) + b^*(\mathbf{u}_h^{n+1/2}, \mathbf{s}_h^{n+1/2}, \mathbf{v}_h)$$

$$+(\mathbf{u}_h^{n+1/2} - \overline{\mathbf{u}}_h^{h^{n+1/2}}, \mathbf{v}_h) + \chi(\mathbf{s}_h^{n+1/2} - \mathbf{w}_h^{n+1/2}, \mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h \in V^h \tag{2.11}$$

$$\delta^2(\nabla \mathbf{w}_h^{n+1}, \nabla v_h) + (\mathbf{w}_h^{n+1}, \mathbf{v}_h) = (\mathbf{s}_h^{n+1}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in V^h \tag{2.12}$$

19

Figure 2.5: Vorticity for NSE Vs. TRM for various N, $\chi = 1$, scaled $\delta_N = \delta_0\sqrt{N+1}$

(a) NSE, $T = 0.8$ (b) $N = 0, T = 0.8$ (c) $N = 1, T = 0.8$ (d) $N = 2, T = 0.8$

(e) NSE, $T = 1.0$ (f) $N = 0, T = 1.0$ (g) $N = 1, T = 1.0$ (h) $N = 2, T = 1.0$

(i) NSE, $T = 1.2$ (j) $N = 0, T = 1.2$ (k) $N = 1, T = 1.2$ (l) $N = 2, T = 1.2$
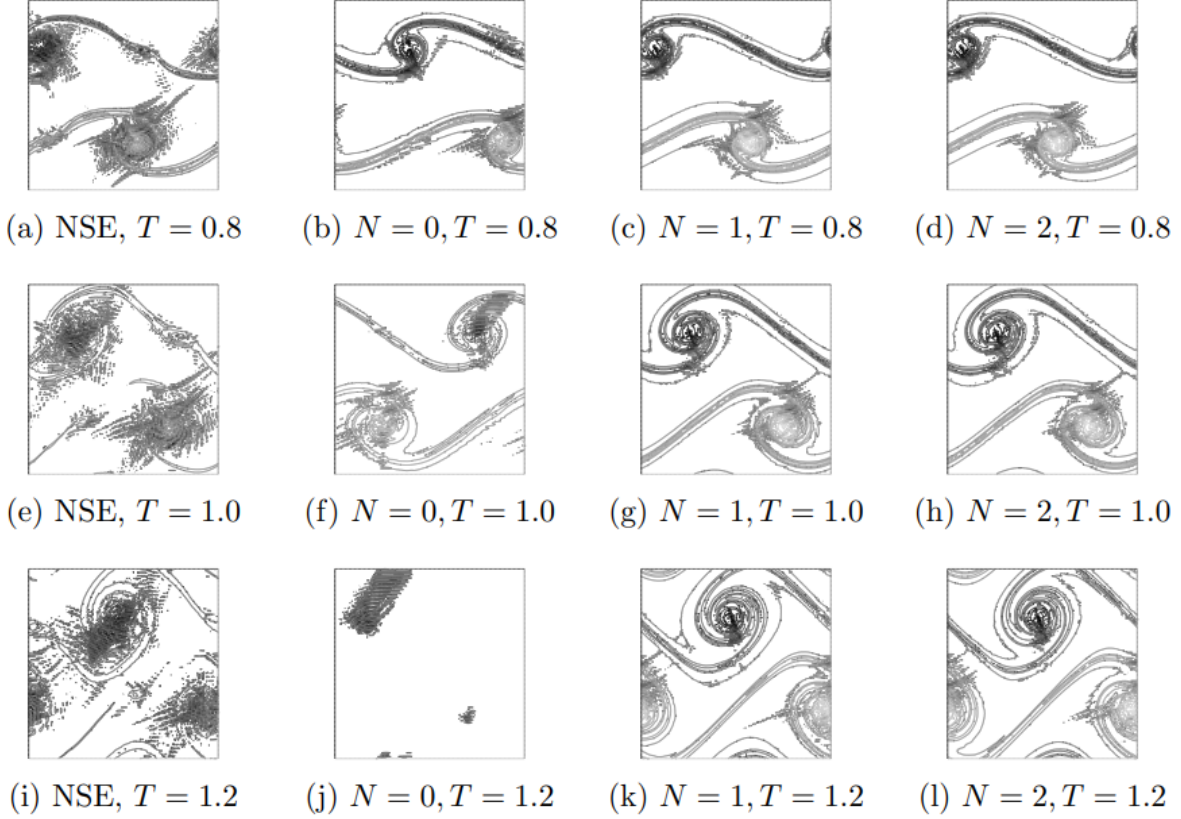
The existence, stability, and convergence analysis of the finite element solution of the above scheme is investigated in [68]. In the following numerical computations, we consider using sensitivity as an accuracy assessment for the approximated velocity solution with different values of parameter χ via computing $\chi\|\mathbf{s}\|_{l^2(0,1.2;L^2(\Omega))}$. This is a simple result based on the following difference quotient for the sensitivity,

$$\mathbf{s} = \frac{\partial \mathbf{u}}{\partial \chi} \approx \frac{\mathbf{u}(\chi) - \mathbf{u}(0)}{\chi}$$

where $\mathbf{u}$ is considered an implicit function of χ. Thus, $\mathbf{u}(0)$ indicates the true solution of Navier–Stokes equations while $\mathbf{u}(\chi)$ for $\chi > 0$ denotes the corresponding TRM approximation of the velocity.

Here we present the results of running the sensitivity variational formulation in equations (2.11)-

20

Figure 2.6: Vorticity for $N = 1, 2$, $\chi = 1$, unscaled δ_0 and scaled δ_N, at times $T = 0.8$ (first row), $T = 1.0$ (second row), and $T = 1.2$ (third row)

(a) $N = 1, \delta_0$ (b) $N = 1, \delta_1$ (c) $N = 2, \delta_0$ (d) $N = 2, \delta_2$

(e) $N = 1, \delta_0$ (f) $N = 1, \delta_1$ (g) $N = 2, \delta_0$ (h) $N = 2, \delta_2$

(i) $N = 1, \delta_0$ (j) $N = 1, \delta_1$ (k) $N = 2, \delta_0$ (l) $N = 2, \delta_2$
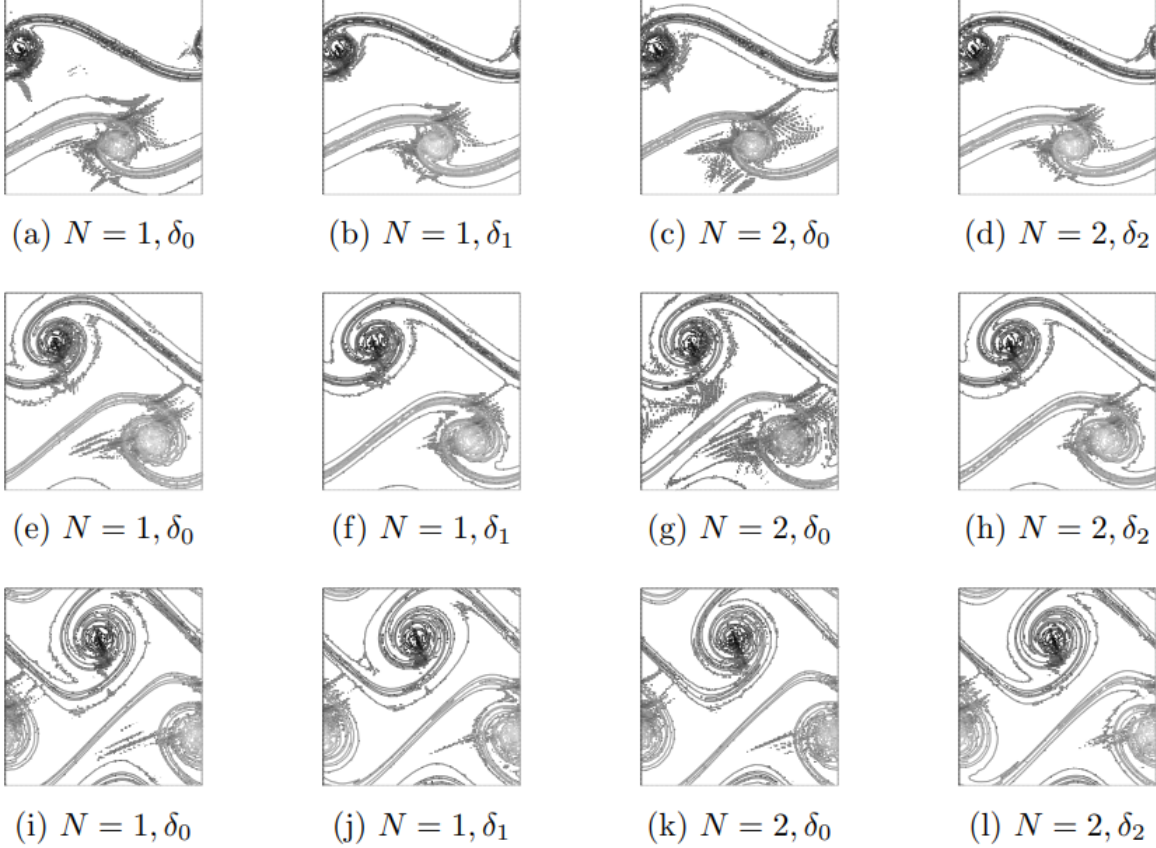
(2.12) for the Shear Layer Roll-up problem. Tables 2.1 and 2.2 contain the sensitivity values of $\chi \|\mathbf{s}\|_{l^2(0,1.2;L^2(\Omega))}$ for Reynolds number $Re = 10,000$ and the time relaxation parameter $\chi = 0.01, 0.1, 1$, and 10 with two choices for a fixed filter length of $\delta_0 = 5/m = 0.05$ and $\delta_N = \delta_0 \sqrt{N+1}$ (i.e. $\delta_1 = \delta_0 \sqrt{2}$ and $\delta_2 = \delta_0 \sqrt{3}$) as well as different values of the van Cittert operator, i.e. $N = 0, 1$, and 2. The selected spatial mesh size consists of $m = 100$ subintervals in both x and y directions. All the computations are carried out with a uniform time-step $\Delta t = 0.004$ using the Taylor–Hood finite elements on the time interval $[0, 1.2]$. We note that the fixed point iteration implemented for the nonlinear term failed to converge (after 50 performed iterations) for $N = 0$ and $\chi = 10$.

Across all the N values, the least values of $\chi \|\mathbf{s}\|_{l^2(0,1.2;L^2(\Omega))}$ are associated with $\chi \leq 0.1$. Thus, we identify the interval of $[0, 0.1]$ as the interval of reliability for this parameter value. All the tested

Table 2.1: $\chi\|\mathbf{s}\|_{l^2(0,1.2;L^2(\Omega))}$ values for unscaled δ_0

χ	$N=0$	$N=1$	$N=2$
0.01	3.035e-2	3.44e-3	2.655e-3
0.1	2.5865e-1	2.356e-2	1.8893e-2
1	2.08791	9.313e-2	7.0404e-2
10	-	1.8300e-1	2.13989e-1

Table 2.2: $\chi\|\mathbf{s}\|_{l^2(0,1.2;L^2(\Omega))}$ values for scaled δ_N

χ	$N=0$	$N=1$	$N=2$
0.01	3.035e-2	4.64592e-3	4.46365e-3
0.1	2.5865e-1	3.17663e-2	3.04986e-2
1	2.08791	1.29948e-1	1.22917e-1
10	-	1.79795e-1	1.40921e-1

χ values in both tables show an improved sensitivity value for higher order van Cittert operator, i.e. for larger N values, indicating a better accuracy for the approximated velocity $\mathbf{u}$.

With this same experiment, the progression of sensitivity values across different times, $t = 0.2$, 0.4, 0.6, 0.8, 1.0, and 1.2, for $\delta_0 = 5/m = 0.05$ was tested. The plots in Figure 2.7 present the results of these computations for $\chi = 0.01, 0.1, 1, 10$ and $N = 0, 1$, and 2. For any χ values, the sensitivity values at all times are smaller for larger values of N. For all N and χ values, $\chi\|\mathbf{s}\|_{l^2(0,T;L^2(\Omega))}$ increases in value as T progresses to 1.2.

Figure 2.8 shows the norm of sensitivity of TRM with $\chi = 0.1$ at times $t = 1.0$ and $t = 1.2$ for $Re=10,000$, with unscaled $\delta_0 = 5/m$, and for order of deconvolution $N = 1$. We can see that the sensitivity, i.e. velocity deviations with respect to χ, are concentrated in areas of vorticity actions.

Figure 2.7: Sensitivity $\chi\|\mathbf{s}\|_{l^2(0,T;L^2(\Omega))}$ values in time for unscaled $\delta_0 = \frac{5}{m}$
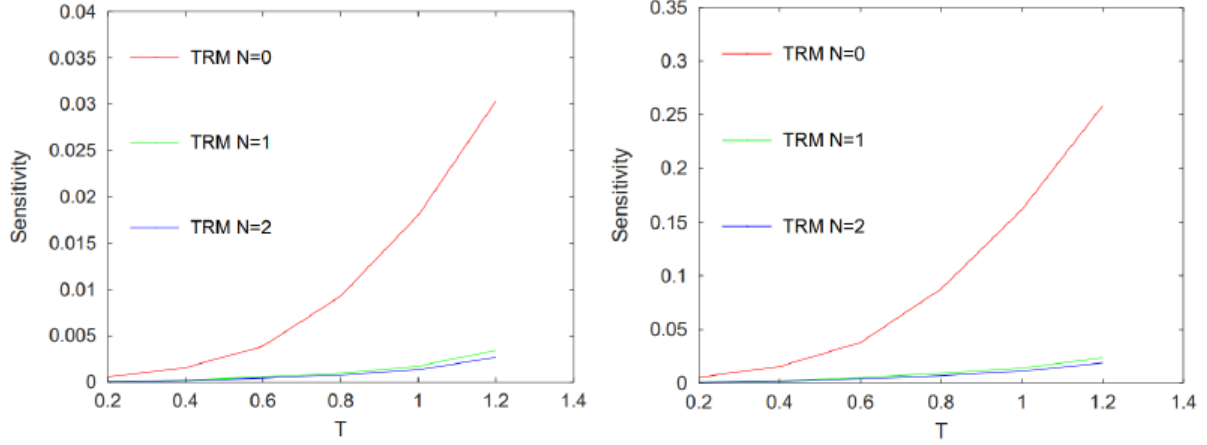
(a) $\chi = 0.01$

(b) $\chi = 0.1$
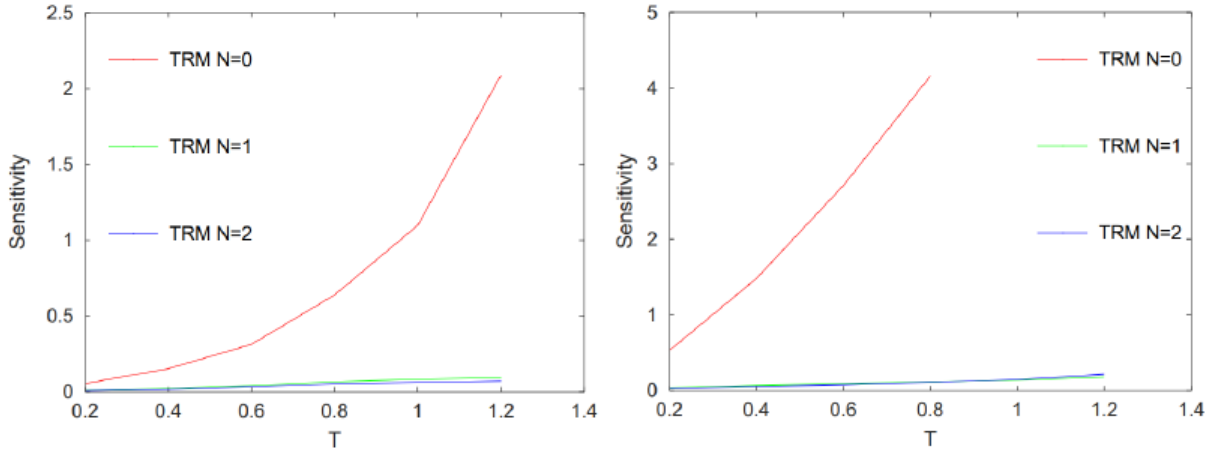
(c) $\chi = 1$

(d) $\chi = 10$

Figure 2.8: Sensitivity norm of TRM for $\chi = 0.1$, $\delta = \frac{5}{m}$, $N = 1$ at $t = 1.0, 1.2$
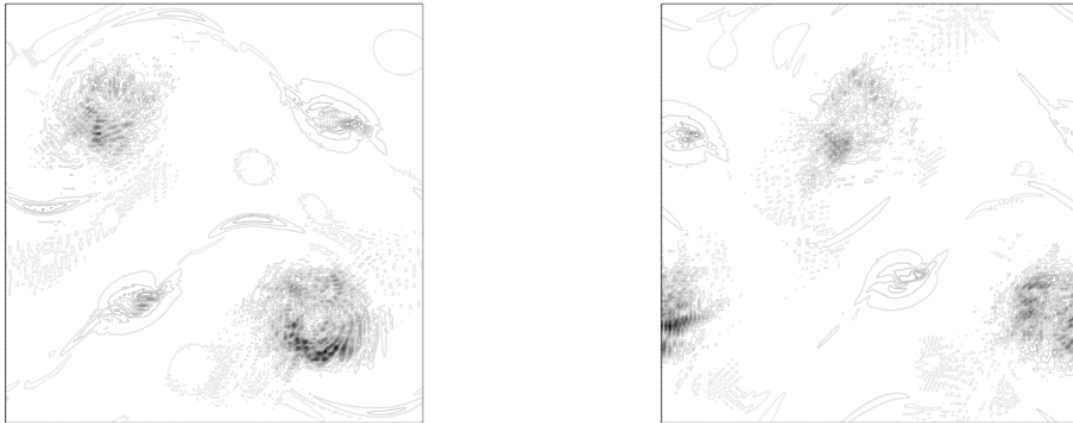
(a) $t = 1.0$

(b) $t = 1.2$

CHAPTER 3

TIME RELAXATION MODEL WITH THE EMAC FORMULATION

It has been known that energy conservation breaks down for Galerkin discretizations of the NSE, and the standard method for fixing this problem is to use either the skew-symmetric or rotational forms of the non-linearity [48]. Many successful attempts have been made to discretize the NSE in a way that retains conservation properties of various physical quantities of interest, such as energy, momentum, enstrophy, helicity, etc. [13, 4, 31, 23, 3, 10, 34, 18, 59]. Throughout all of this research, the common conclusion is that as more physics is built into the discretization, one can expect more accurate and stable discrete solutions, especially over longer time intervals. Recently, Charyni et. al. [58] developed a new scheme they called the EMAC scheme, which is short for energy, momentum, and angular momentum conservation. This scheme gives a new form of the non-linear term in the NSE, and it was compared to 4 different forms of the non-linear term: the convective, the skew symmetric, the rotational, and the conservative form; across several types of temporal discretizations. Since its development, the EMAC formulation has been used for several applications, including vortex-induced vibration [41], turbulent flow simulation [2], cardiovascular simulations and hemodynamics [24, 42], and noise radiated by an open cavity [50]. The conservation properties were also explored for different types of linearizations of semi-discrete schemes of the EMAC in [20], and very recently, Olshanskii and Rebholz have shown that semi-discrete formulations of the NSE using EMAC have better longer time estimates than those based on the skew-symmetric formulation [43]. To that end, and with the aim to handle higher Reynolds number on coarser physical meshes, we study this EMAC numerical scheme applied to TRM. We also provide a comparison with the skew-symmetric numerical scheme for TRM.

In Section 1 of this chapter, we introduce two fully discretized finite element schemes for the TRM. One will be the standard skew-symmetric scheme (SKEW), and the other is the newly proposed EMAC scheme. We will present balances for the energy, momentum, and angular momentum for the continuous and fully discrete TRM, and we will show that only the balances for EMAC will reduce to the conservation properties of the continuous Navier Stokes equations in the case of no external forces, viscosity, and time relaxation term. In Section 2 of this chapter, we will prove stability and error estimates for both schemes, and comparisons will be drawn between them. In particular, we will confirm that the standard estimate for the SKEW scheme results in an error which is of the order $O(e^{\nu^{-3}})$, and for the EMAC scheme, we will show that the error is of the order $O(e^{\nu^{-1}})$, which is a significant improvement for high Reynolds number problems. In Section 3, we test our schemes with several numerical experiments on benchmark problems in 2D and 3D, and convergence rates for these schemes will be verified as well.

3.1   Numerical Scheme with EMAC

Based on model equations (1.26) and integration by parts, we obtain the following variational form for the Time Relaxation model: Find $(\boldsymbol{u}, p) \in X \times Q$, satisfying the following $\forall (\boldsymbol{v}, q) \in X \times Q$.

$$(\boldsymbol{u}_t, \boldsymbol{v}) + (\boldsymbol{u} \cdot \nabla \boldsymbol{u}, \boldsymbol{v}) + \nu(\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) - (p, \nabla \cdot \boldsymbol{v}) + (\chi(\boldsymbol{u} - G_N(\overline{\boldsymbol{u}})), \boldsymbol{v}) = \mathbf{f}, \ \text{in } \Omega, \qquad (3.1)$$

$$(q, \nabla \cdot \boldsymbol{u}) = 0, \ \text{in } \Omega. \qquad (3.2)$$

We use finite element for space discretization and Crank-Nicolson for the time derivative. By defining $\boldsymbol{v}_h^{n+\frac{1}{2}} := \frac{1}{2}(v_h^{n+1} + v_h^n)$, we then obtain the following fully discrete numerical schemes for SKEW and EMAC.

SKEW scheme: For $n = 1, 2, ..., l-1$, where $l = \frac{T}{\triangle t}$, $\boldsymbol{u}_h^0 := P_{L^2}(\boldsymbol{u}_0)$, find $(\boldsymbol{u}_h^{n+1}, p_h^{n+1}) \in \boldsymbol{X}_h \times Q_h$

satisfying

$$\frac{1}{\triangle t}(\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n, \boldsymbol{v}_h) + b^*(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{v}_h) + \nu(\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}, \nabla\boldsymbol{v}_h) - (p_h^{n+\frac{1}{2}}, \nabla\cdot\boldsymbol{v}_h)$$

$$+\chi\left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{v}_h\right) = (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{v}_h), \text{ in } \Omega, \qquad (3.3)$$

$$(q_h, \nabla\cdot\boldsymbol{u}_h^{n+\frac{1}{2}}) = 0, \text{ in } \Omega, \qquad (3.4)$$

for all $\boldsymbol{v}_h \in \boldsymbol{X}_h, q_h \in Q_h$.

EMAC scheme: For $n = 1, 2, ..., l-1$, where $l = \frac{T}{\triangle t}$, $\boldsymbol{u}_h^0 := P_{St}(\boldsymbol{u}_0)$, find $(\boldsymbol{u}_h^{n+1}, P_h^{n+1}) \in \boldsymbol{X}_h \times Q_h$

satisfying

$$\frac{1}{\triangle t}(\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n, \boldsymbol{v}_h) + c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{v}_h) + \nu(\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}, \nabla\boldsymbol{v}_h) - (P_h^{n+\frac{1}{2}}, \nabla\cdot\boldsymbol{v}_h)$$

$$+\chi\left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{v}_h\right) = (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{v}_h), \text{ in } \Omega, \qquad (3.5)$$

$$(q_h, \nabla\cdot\boldsymbol{u}_h^{n+\frac{1}{2}}) = 0, \text{ in } \Omega, \qquad (3.6)$$

for all $\boldsymbol{v}_h \in \boldsymbol{X}_h, q_h \in Q_h$.

Note that for each scheme, we select $\boldsymbol{u}_h^0$ with different projection operators. For the SKEW scheme, we choose the L^2 projection operator, and for the EMAC scheme we choose the Stokes projection operator. The Stokes projection operator will be necessary to use for EMAC to obtain a better error estimate than what can be obtained using the L^2 projection.

3.1.1 Balances for continuous formulation

Following the work for the Navier Stokes equations in [38], we recall the energy balance for TRM in the continuous case, which can be derived from equation (1.26) using the techniques in [77],

$$\frac{1}{2}||\boldsymbol{u}(t)||^2 + \int_0^t \int_\Omega \left(\nu|\nabla\boldsymbol{u}|^2 + \chi(I - G_N G)\boldsymbol{u}\cdot\boldsymbol{u}\right) dxd\tau = \frac{1}{2}||\boldsymbol{u}(0)||^2 + \int_0^t (\mathbf{f}, \boldsymbol{u})d\tau. \qquad (3.7)$$

Note that the time relaxation term enhances the energy dissipation. In a similar way, we can derive momentum and angular momentum balances for TRM as well.

Proposition 3.1.1. The solutions of TRM given in (1.26) satisfy the following momentum and angular momentum balances, where $\boldsymbol{e}_i$ for $i = 1, 2, 3$ represents each coordinate direction, and $\psi_i = \boldsymbol{x} \times \boldsymbol{e}_i$ for $i = 1, 2, 3$

$$(\boldsymbol{u}(t), \boldsymbol{e}_i) + \chi \int_0^t ((I - G_N G)\boldsymbol{u}, \boldsymbol{e}_i)\, d\tau = (\boldsymbol{u}(0), \boldsymbol{e}_i) + \int_0^t (\mathbf{f}, \boldsymbol{e}_i) d\tau, \tag{3.8}$$

$$(\boldsymbol{u}(t), \psi_i) + \chi \int_0^t ((I - G_N G)\boldsymbol{u}, \psi_i)\, d\tau = (\boldsymbol{u}(0), \psi_i) + \int_0^t (\mathbf{f}, \psi_i) d\tau. \tag{3.9}$$

Proof. For the momentum balance, first we take equation (1.26), multiply both sides by $\boldsymbol{e}_i$ and integrate over Ω. Note that $\nabla \cdot \boldsymbol{e}_i = \nabla \boldsymbol{e}_i = 0$, which causes the viscous term and pressure term to vanish,

$$(\boldsymbol{u}_t, \boldsymbol{e}_i) + (\boldsymbol{u} \cdot \nabla \boldsymbol{u}, \boldsymbol{e}_i) + \chi\,((I - G_N G)\boldsymbol{u}, \boldsymbol{e}_i) = (\mathbf{f}, \boldsymbol{e}_i). \tag{3.10}$$

Using integration by parts, we see that the non-linear term also vanishes since $\boldsymbol{u} = 0$ on $\partial\Omega$, $\nabla \boldsymbol{e}_i = 0$, and $\boldsymbol{u}$ is weakly divergence free, i.e.

$$\int_\Omega \boldsymbol{u} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{e}_i dx = \int_{\partial\Omega} (\boldsymbol{u} \cdot \boldsymbol{e}_i)\boldsymbol{u}\, dA - \int_\Omega (\nabla \cdot \boldsymbol{u})(\boldsymbol{u} \cdot \boldsymbol{e}_i) dx - \int_\Omega \boldsymbol{u} \cdot \nabla \boldsymbol{e}_i \cdot \boldsymbol{u}\, dx = 0.$$

From here, integrating with respect to time in (3.10) will give us the desired balance. For the angular momentum proof, we multiply equation (1.26) by ψ_i and this time we have $\nabla \cdot \psi_i = 0$ and $\triangle \psi_i = 0$, so the pressure term drops out as well as the viscous term after applying integration by parts, leaving us with

$$(\boldsymbol{u}_t, \psi_i) + (\boldsymbol{u} \cdot \nabla \boldsymbol{u}, \psi_i) + \chi\,((I - G_N G)\boldsymbol{u}, \psi_i) = (\mathbf{f}, \psi_i) \tag{3.11}$$

27

Using integration by parts on the non-linear term, we obtain the following, noting that two of the integrals vanish due to $\boldsymbol{u}$ being both weakly divergence free and equal to 0 on $\partial\Omega$,

$$\int_\Omega \boldsymbol{u} \cdot \nabla \boldsymbol{u} \cdot \psi_i dx = \int_{\partial\Omega} (\boldsymbol{u} \cdot \psi_i)\boldsymbol{u}dA - \int_\Omega (\nabla \cdot \boldsymbol{u})(\boldsymbol{u} \cdot \psi_i)dx - \int_\Omega \boldsymbol{u} \cdot \nabla\psi_i \cdot \boldsymbol{u}dx$$

$$= -\int_\Omega \boldsymbol{u} \cdot \nabla\psi_i \cdot \boldsymbol{u}dx = 0.$$

The last equality above follows by expanding the components of $\boldsymbol{u} \cdot \nabla\psi_i \cdot \boldsymbol{u}$ and finding that they vanish, hence the non-linear term drops out of (3.11). Finally, integrating both sides of (3.11) with respect to time will give us our final result. $\qquad\square$

3.1.2   Balances for fully discrete formulation

Now we present the energy, momentum, and angular momentum balances for the fully discretized TRM, and we will see that both SKEW and EMAC have an energy balance with no contributions from the non-linear term, but only the EMAC scheme will have balances for momentum and angular momentum with no contributions from non-linearity. Thus only the fully discretized TRM with EMAC has analogous balances to the continuous TRM balances given in (3.7)-(3.9).

Proposition 3.1.2. Solutions to TRM with SKEW and EMAC both satisfy the following energy balance for $l = 1, 2, 3, ...,$

$$\frac{1}{2}||\boldsymbol{u}_h^l||^2 + \nu\triangle t \sum_{n=0}^{l-1}||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||^2 + \chi\triangle t \sum_{n=0}^{l-1}||\boldsymbol{u}_h^{n+\frac{1}{2}*}||^2 = \frac{1}{2}||\boldsymbol{u}_h^0||^2 + \sum_{n=0}^{l-1}(\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}).$$

Proof. Taking $\boldsymbol{v}_h = \boldsymbol{u}_h^{n+\frac{1}{2}}$ in (3.5)-(3.6), we obtain the following (noting that the pressure term will vanish since $\boldsymbol{u}_h^{n+\frac{1}{2}}$ is weakly divergence free)

$$\frac{1}{2\triangle t}\left(||\boldsymbol{u}_h^{n+1}||^2 - ||\boldsymbol{u}_h^n||^2\right) + c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}) + \nu||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||^2 + \chi||\boldsymbol{u}_h^{n+\frac{1}{2}*}||^2 = (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}).$$

Then by our properties of the non-linear form c (and also for b^* in the case of SKEW), we have that $c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}) = 0$, so by summing everything from $n = 0$ to $n = l - 1$ and multiplying both sides by $\triangle t$, we obtain our energy balance for both schemes.

$\square$

Proposition 3.1.3. Solutions to the TRM with EMAC satisfy the following momentum balance for $l = 1, 2, 3, ...$,

$$(\boldsymbol{u}_h^l, \boldsymbol{e}_i) + \chi \triangle t \sum_{n=0}^{l-1} \left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right) = (\boldsymbol{u}_h^0, \boldsymbol{e}_i) + \sum_{n=0}^{l-1} (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{e}_i).$$

Proof. First, we note that for each $\boldsymbol{e}_i$, we have that $\nabla \boldsymbol{e}_i = \nabla \cdot \boldsymbol{e}_i = 0$. To get this momentum balance from (3.5)-(3.6), we want to take $\boldsymbol{v}_h = \boldsymbol{e}_i$, but $\boldsymbol{e}_i$ is not in X_h. However, if we assume that $\boldsymbol{u} \neq 0$ only in a strictly interior subdomain $\tilde{\Omega} \subset \Omega$, then we can define the restriction $\rho(\boldsymbol{e}_i)$ by letting $\rho(\boldsymbol{e}_i) = \boldsymbol{e}_i$ in $\tilde{\Omega}$ and $\rho(\boldsymbol{e}_i)$ be arbitrarily defined on $\Omega/\tilde{\Omega}$ to satisfy the zero boundary conditions. Thus in our scheme, we let $\boldsymbol{v}_h = \rho(\boldsymbol{e}_i)$ and $q_h = 0$ to obtain

$$\frac{1}{\triangle t}(\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n, \boldsymbol{e}_i) + c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) + \chi((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) = (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{e}_i).$$

Next, we sum everything from $n = 0$ to $n = l - 1$ and then multiplying both sides by $\triangle t$ to obtain

$$(\boldsymbol{u}_h^l, \boldsymbol{e}_i) + \triangle t \sum_{n=0}^{l-1} c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) + \chi \triangle t \sum_{n=0}^{l-1} \left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right) = (\boldsymbol{u}_h^0, \boldsymbol{e}_i) + \sum_{n=0}^{l-1} (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{e}_i).$$

Thus momentum balance for the EMAC scheme will hold as long as $c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) = 0$ for $n = 0$ to $l - 1$. This is readily seen to be the case below, using equations (1.14), (1.16) and the fact that $\nabla \cdot \boldsymbol{e}_i = 0$ implies that $b(\boldsymbol{e}_i, \boldsymbol{v}, \boldsymbol{v}) = b^*(\boldsymbol{e}_i, \boldsymbol{v}, \boldsymbol{v}) = 0$ for any $\boldsymbol{v} \in X_h$.

$$c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) = 2 \left(D(\boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right) + \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right)$$

$$= \left((\nabla \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right) + \left((\nabla \boldsymbol{u}_h^{n+\frac{1}{2}})^T \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right) + \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right)$$

$$= b(\boldsymbol{e}_i, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}) + b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) + \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right)$$

$$= b(\boldsymbol{e}_i, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}) - b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i, \boldsymbol{u}_h^{n+\frac{1}{2}})$$

$$= 0.$$

Note that the argument above does not work for the SKEW scheme, since if we replace $c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i)$ with $b^*(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i)$, then the proof for SKEW reduces to showing that $b^*(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) = 0$ for $n = 0$ to $l - 1$. However this is not true in general, because we have the following based on equation (1.14)

$$b^*(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) = \frac{1}{2}b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i) - \frac{1}{2}b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i, \boldsymbol{u}_h^{n+\frac{1}{2}})$$

$$= -b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i, \boldsymbol{u}_h^{n+\frac{1}{2}}) - \frac{1}{2}\left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right)$$

$$= -\frac{1}{2}\left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{e}_i \right) \neq 0.$$

$\square$

Proposition 3.1.4. Solutions to the TRM with EMAC satisfy the following angular momentum balance for $l = 1, 2, 3, ...,$

$$(\boldsymbol{u}_h^l, \psi_i) + \chi \triangle t \sum_{n=0}^{l-1} \left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i \right) = (\boldsymbol{u}_h^0, \psi_i) + \sum_{n=0}^{l-1} (\mathbf{f}^{n+\frac{1}{2}}, \psi_i).$$

Proof. Similar to the momentum proof, we cannot select $\boldsymbol{v}_h = \psi_i$ in (3.5)-(3.6) since again ψ_i is not in X_h, so we define the restriction $\rho(\psi_i)$ similarly to what was done in the momentum proof,

and let $\boldsymbol{v}_h = \rho(\psi_i)$ and $q_h = 0$. This time, we get the following, observing that the pressure term and the viscous term drop out since for each ψ_i, we have $\nabla \cdot \psi_i = 0$ and $\triangle \psi_i = 0$,

$$\frac{1}{\triangle t}(\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n, \psi_i) + c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) + \chi((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) = (\mathbf{f}^{n+\frac{1}{2}}, \psi_i).$$

Summing the above from $n = 0$ to $l - 1$, multiplying both sides by $\triangle t$, we obtain

$$(\boldsymbol{u}_h^l, \psi_i) + \triangle t \sum_{n=0}^{l-1} c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) + \chi \triangle t \sum_{n=0}^{l-1}\left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) = (\boldsymbol{u}_h^0, \psi_i) + \sum_{n=0}^{l-1}(\mathbf{f}^{n+\frac{1}{2}}, \psi_i).$$

Thus the angular momentum balance for EMAC will hold as long as $c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) = 0$ for any $n = 0$ to $l - 1$. This is readily shown again using equations (1.14) and (1.16).

$$
\begin{aligned}
c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) &= 2\left(D(\boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) + \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) \\
&= \left((\nabla \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) + \left((\nabla \boldsymbol{u}_h^{n+\frac{1}{2}})^T \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) + \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) \\
&= b(\psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}) + b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) + \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) \\
&= \left((\nabla \cdot \psi_i)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}\right) - b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}}) \\
&= -b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}}).
\end{aligned}
$$

From here, one can expand the components of $b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}})$ to find that they vanish, so that $c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) = 0$. This gives us the desired angular momentum balance.

For the SKEW scheme, the above proof does not work out. If one replaces c with b^*, then everything plays out the same, and the result would hold if one could show that $b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}}) = 0$. However this is not the case, as shown below, using equations (1.14), (1.16) and the fact that $b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}}) = 0$.

$$b^*(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) = \frac{1}{2}b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i) - \frac{1}{2}b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}})$$

$$= -b(\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i, \boldsymbol{u}_h^{n+\frac{1}{2}}) - \frac{1}{2}\left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right)$$

$$= -\frac{1}{2}\left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \psi_i\right) \neq 0.$$

$\square$

We remark that the above balances for the EMAC scheme with $\nu = 0$, $\chi = 0$, and $\mathbf{f} = 0$ reduce to the fully discrete analogues of the conservation of energy, momentum, and angular momentum for the continuous Navier Stokes equations.

3.2 Numerical Analysis of the SKEW and EMAC schemes

Here we provide stability results and error estimates for both the SKEW and EMAC formulations of the Time Relaxation model. It is useful to compare the results for these two schemes since considerable work has been done in the literature to confirm SKEW as an effective scheme for computations, while very little has been done for EMAC so far. Moreover, it is well known that the error estimate for SKEW under typical regularity assumptions gives an error which is bounded by a term on the order of $O(e^{\nu^{-3}})$, which obviously for large-eddy simulations with high Reynolds number is an issue for convergence. It will be shown that EMAC brings this dependence down to the order of $O(e^{\nu^{-1}})$ under the same regularity assumptions, which implies that EMAC is a theoretically stronger converging scheme than SKEW, especially so in the high turbulence domain.

First we begin with the stability estimates, which are exactly the same for both the SKEW and EMAC schemes.

Proposition 3.2.1. The solutions to the schemes given by (3.3)-(3.4) and (3.5)-(3.6) exist, are unique, and satisfy the following stability condition

$$||\boldsymbol{u}_h^M||^2 + \nu\triangle t \sum_{n=0}^{M-1} ||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||^2 + 2\triangle t\chi \sum_{n=0}^{M-1} ||\boldsymbol{u}_h^{n+\frac{1}{2}*}||^2 \le C(\nu, \mathbf{f}, \boldsymbol{u}_0, T). \qquad (3.12)$$

Proof. Choose $\boldsymbol{v}_h = \boldsymbol{u}_h^{n+\frac{1}{2}}$ in (3.3)-(3.4) and (3.5)-(3.6) and note that the non-linear term and the pressure term vanish, and thus we obtain the following for both schemes

$$\frac{1}{\triangle t}(\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n, \boldsymbol{u}_h^{n+\frac{1}{2}}) + \nu(\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}, \nabla\boldsymbol{u}_h^{n+\frac{1}{2}}) + \chi\left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}\right) = (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}),$$

which, after using the the dual norm of $\mathbf{f}$, Young's inequality and (1.32), reduces to

$$\frac{1}{2\triangle t}(||\boldsymbol{u}_h^{n+1}||^2 - ||\boldsymbol{u}_h^n||^2) + \nu||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||^2 + \chi||\boldsymbol{u}_h^{n+\frac{1}{2}*}||^2 \le \frac{\nu}{2}||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||^2 + \frac{1}{2\nu}||\mathbf{f}^{n+\frac{1}{2}}||_{-1}^2.$$

Next we multiply each side by $2\triangle t$, hide the $||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||^2$ term on the left hand side while moving the initial solution to the right hand side, and sum from $n = 0$ to $M - 1$ to obtain

$$||\boldsymbol{u}_h^M||^2 + \nu\triangle t \sum_{n=0}^{M-1} ||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||^2 + 2\chi\triangle t \sum_{n=0}^{M-1} ||\boldsymbol{u}_h^{n+\frac{1}{2}*}||^2 \le ||\boldsymbol{u}_h^0||^2 + \frac{\triangle t}{\nu} \sum_{n=0}^{M-1} ||\mathbf{f}^{n+\frac{1}{2}}||_{-1}^2.$$

Using the definition of $||\boldsymbol{u}_h^0||^2$, our result is achieved. Once this bound is established, the existence and uniqueness of the solution follows from the arguments in [55, 53]. □

3.2.1 SKEW Error Estimate

Now we will proceed to derive an error estimate for both the SKEW and EMAC schemes. We will first perform the SKEW error estimate, and then we will see how the EMAC error estimate is modified from the SKEW error estimate when taking into account the different form of the non-linear term.

33

Theorem 3.2.2. Let $(\boldsymbol{w}(t), p(t))$ be a smooth enough solution of the NSE satisfying no-slip boundary conditions, $\mathbf{f} \in L^\infty(0, T; H^{-1}(\Omega))$ and $\boldsymbol{w}_0 \in H^1(\Omega)$. Let (u_h^n, p_h^n), for $n = 0, 1, 2, ..., l$ be the solution to the SKEW scheme given by (3.3)-(3.4) using (P_k, P_{k-1}) elements for some $k \geq 2$. Then the error in the discrete solution satisfies

$$|||\boldsymbol{w} - \boldsymbol{u}_h|||_{\infty,0}^2 + \frac{\nu \triangle t}{2} \sum_{n=0}^{l-1} ||\nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{u}_h^{n+\frac{1}{2}})||^2 + 2\chi \triangle t \sum_{n=0}^{l-1} ||(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{u}_h^{n+\frac{1}{2}})^*||^2$$

$$\leq C(e^{\nu^{-3}}) \left\{ h^{2k} + h^{2k+1} + \chi^2 \delta^{4N+4} + \chi^2 (\delta h^k + h^{k+1})^2 + (\triangle t)^4 \right\}. \tag{3.13}$$

Proof. An equivalent scheme to (3.3)-(3.4) is given as follows: Find $\boldsymbol{u}_h^n \in V_h$ such that $\forall \boldsymbol{v}_h \in V_h$, we have

$$\frac{1}{\triangle t}(\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n, \boldsymbol{v}_h) + b^*(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{v}_h) + \nu(\nabla \boldsymbol{u}_h^{n+\frac{1}{2}}, \nabla \boldsymbol{v}_h)$$

$$+\chi \left((I - G_N^h G_h)\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{v}_h \right) = (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{v}_h). \tag{3.14}$$

At time $t_{n+\frac{1}{2}}$, the true solution $\boldsymbol{w}$ satisfies the following for all $\boldsymbol{v}_h \in V_h$,

$$\frac{1}{\triangle t}\left(\boldsymbol{w}(t^{n+1}) - \boldsymbol{w}(t^n), v_h\right) + b^*\left(\boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{v}_h\right) + \nu \left(\nabla \boldsymbol{w}^{n+\frac{1}{2}}, \nabla \boldsymbol{v}_h\right)$$

$$+\chi \left(\left(I - G_N^h G_h\right) \boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{v}_h \right) - (p(t^{n+\frac{1}{2}}), \nabla \cdot \boldsymbol{v}_h) = (\mathbf{f}^{n+\frac{1}{2}}, \boldsymbol{v}_h) + \mathrm{Intp}(\boldsymbol{w}(t^n), \boldsymbol{v}_h), \tag{3.15}$$

where $\mathrm{Intp}(\boldsymbol{w}(t^n), \boldsymbol{v}_h)$, representing the interpolating error, is given as

$$\mathrm{Intp}(\boldsymbol{w}(t^n), \boldsymbol{v}_h) = \left(\frac{\boldsymbol{w}(t^{n+1}) - \boldsymbol{w}(t^n)}{\triangle t} - \boldsymbol{w}_t(t^{n+\frac{1}{2}}), \boldsymbol{v}_h\right) + b^*\left(\boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{v}_h\right)$$

$$- b^*(\boldsymbol{w}(t^{n+\frac{1}{2}}), \boldsymbol{w}(t^{n+\frac{1}{2}}), \boldsymbol{v}_h) + \nu \left(\nabla \boldsymbol{w}^{n+\frac{1}{2}} - \nabla \boldsymbol{w}(t^{n+\frac{1}{2}}), \nabla \boldsymbol{v}_h\right)$$

$$+ \chi \left(\left(I - G_N^h G_h\right) \boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{v}_h \right). \tag{3.16}$$

Subtracting (3.14) from (3.15) and letting $\boldsymbol{e}^n = \boldsymbol{w}(t^n) - \boldsymbol{u}_h^n$, we obtain

34

$$\frac{1}{\triangle t}(e^{n+1} - e^n, v_h) + b^*\left(w^{n+\frac{1}{2}}, w^{n+\frac{1}{2}}, v_h\right) - b^*(u_h^{n+\frac{1}{2}}, u_h^{n+\frac{1}{2}}, v_h) + \nu(\nabla e^{n+\frac{1}{2}}, \nabla v_h)$$

$$+ \chi\left((I - G_N^h G_h)e^{n+\frac{1}{2}}, v_h\right) - (p(t^{n+\frac{1}{2}}), \nabla \cdot v_h) = \mathrm{Intp}(w(t^n), v_h), \tag{3.17}$$

for all $v_h \in V_h$.

Let $e^n = w(t^n) - u_h^n = (w(t^n) - U^n) + (U^n - u_h^n) := \eta^n + \phi_h^n$, where $U^n = P_{L^2}(w(t^n))$ is the L^2 projection of the true solution at time t^n into V_h, and thus $\phi_h^n \in V_h$. We set $u_h^0 = U(0)$. Then we let $v_h = \phi_h^{n+\frac{1}{2}}$, and using the fact that $(q_h, \nabla \cdot \phi_h^{n+\frac{1}{2}}) = 0$ for all $q_h \in Q_h$ and the definition of the norm in (1.32), equation (3.17) simplifies to

$$\frac{1}{2\triangle t}(||\phi_h^{n+1}||^2 - ||\phi_h^n||^2) + \nu||\nabla\phi_h^{n+\frac{1}{2}}||^2 + \chi||\phi_h^{n+\frac{1}{2}*}||^2 = b^*(u_h^{n+\frac{1}{2}}, u_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}})$$

$$- b^*\left(w^{n+\frac{1}{2}}, w^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) - \frac{1}{\triangle t}(\eta^{n+1} - \eta^n, \phi_h^{n+\frac{1}{2}}) - \nu(\nabla\eta^{n+\frac{1}{2}}, \nabla\phi_h^{n+\frac{1}{2}})$$

$$- \chi\left((I - G_N^h G_h)\eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) + (p(t^{n+\frac{1}{2}}) - q_h, \nabla \cdot \phi_h^{n+\frac{1}{2}}) + Intp(w(t^n), \phi_h^{n+\frac{1}{2}}). \tag{3.18}$$

Now we begin estimating the right hand side of (3.18). We first begin with the skew symmetric terms, since these terms will provide the key difference between this proof and the proof for the EMAC scheme.

First, we expand the non-linear terms as follows, noting that one term will drop out since $b^*(u, v, v) = 0$ for all $u, v \in X_h$,

$$b^*(u_h^{n+\frac{1}{2}}, u_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) - b^*\left(w^{n+\frac{1}{2}}, w^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right)$$

$$= b^*\left(u_h^{n+\frac{1}{2}}, u_h^{n+\frac{1}{2}} - w^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) + b^*\left(u_h^{n+\frac{1}{2}} - w^{n+\frac{1}{2}}, w^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right)$$

$$= -b^*(u_h^{n+\frac{1}{2}}, \eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) - b^*\left(\eta^{n+\frac{1}{2}}, w^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) - b^*\left(\phi_h^{n+\frac{1}{2}}, w^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right). \tag{3.19}$$

We estimate the three terms on the RHS of (3.19) using equation (1.12) and Young's inequality. The 3rd term is where the $e^{\nu^{-3}}$ dependence comes from, which we will hide using the Gronwall's inequality at the end.

$$b^*(\boldsymbol{u}_h^{n+\frac{1}{2}}, \eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) \leq C\|\boldsymbol{u}_h^{n+\frac{1}{2}}\|^{\frac{1}{2}} \|\nabla \boldsymbol{u}_h^{n+\frac{1}{2}}\|^{\frac{1}{2}} \|\nabla \eta^{n+\frac{1}{2}}\| \|\nabla \phi_h^{n+\frac{1}{2}}\|$$

$$\leq \frac{\nu}{12}\|\nabla \phi_h^{n+\frac{1}{2}}\|^2 + C\nu^{-1}\|\boldsymbol{u}_h^{n+\frac{1}{2}}\| \|\nabla \boldsymbol{u}_h^{n+\frac{1}{2}}\| \|\nabla \eta^{n+\frac{1}{2}}\|^2, \qquad (3.20)$$

$$b^*\left(\eta^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) \leq C\|\eta^{n+\frac{1}{2}}\|^{\frac{1}{2}} \|\nabla \eta^{n+\frac{1}{2}}\|^{\frac{1}{2}} \|\nabla \boldsymbol{w}^{n+\frac{1}{2}}\| \|\nabla \phi_h^{n+\frac{1}{2}}\|$$

$$\leq \frac{\nu}{12}\|\nabla \phi_h^{n+\frac{1}{2}}\|^2 + C\nu^{-1}\|\eta^{n+\frac{1}{2}}\| \|\nabla \eta^{n+\frac{1}{2}}\| \|\nabla \boldsymbol{w}^{n+\frac{1}{2}}\|^2, \qquad (3.21)$$

$$b^*\left(\phi_h^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) \leq C\|\phi_h^{n+\frac{1}{2}}\|^{\frac{1}{2}} \|\nabla \phi_h^{n+\frac{1}{2}}\|^{\frac{3}{2}} \|\nabla \boldsymbol{w}^{n+\frac{1}{2}}\|$$

$$\leq \frac{\nu}{12}\|\nabla \phi_h^{n+\frac{1}{2}}\|^2 + C\nu^{-3}\|\phi_h^{n+\frac{1}{2}}\|^2 \|\nabla \boldsymbol{w}^{n+\frac{1}{2}}\|^4. \qquad (3.22)$$

For the remaining 4 terms on the RHS, we note that since U^n is the L^2 projection of $\boldsymbol{w}$ into V_h, then we must have $(\eta^{n+1} - \eta^n, \phi_h^{n+\frac{1}{2}}) = 0$, so this term will also drop out of the analysis. Next, we do similar estimates on the 3 remaining terms other than the interpolation error, using both Poincare and Young's inequality, and assumption 2 from Lemma 1.2.11,

$$\nu(\nabla \eta^{n+\frac{1}{2}}, \nabla \phi_h^{n+\frac{1}{2}}) \leq \frac{\nu}{12}\|\nabla \phi_h^{n+\frac{1}{2}}\|^2 + C\nu\|\nabla \eta^{n+\frac{1}{2}}\|^2, \qquad (3.23)$$

$$\chi\left((I - G_N^h G_h)\eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) \leq \chi\|\eta^{n+\frac{1}{2}}\| \|\phi_h^{n+\frac{1}{2}}\|$$

$$\leq \frac{\nu}{12}\|\nabla \phi_h^{n+\frac{1}{2}}\|^2 + C\frac{\chi^2}{\nu}\|\eta^{n+\frac{1}{2}}\|^2, \qquad (3.24)$$

36

$$\left(p(t^{n+\frac{1}{2}}) - q_h, \nabla \cdot \phi_h^{n+\frac{1}{2}}\right) \leq \frac{\nu}{12}||\nabla \phi_h^{n+\frac{1}{2}}||^2 + C\nu^{-1} \inf_{q_h \in Q_h} ||p - q_h||^2. \tag{3.25}$$

Combining all the estimates from equations (3.20) - (3.25) and substituting them into equation (3.18), summing the inequality from $n = 0$ to $l - 1$ (using that $||\phi^0|| = 0$), and multiplying each side by $2\triangle t$, we obtain,

$$||\phi_h^l||^2 + \nu\triangle t \sum_{n=0}^{l-1} ||\nabla \phi_h^{n+\frac{1}{2}}||^2 + 2\chi\triangle t \sum_{n=0}^{l-1} ||\phi_h^{n+\frac{1}{2}*}||^2$$

$$\leq C\nu^{-3}\triangle t \sum_{n=0}^{l-1} ||\phi_h^{n+\frac{1}{2}}||^2 \, ||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||^4 + C\nu\triangle t \sum_{n=0}^{l-1} ||\nabla \eta^{n+\frac{1}{2}}||^2$$

$$+ C\nu^{-1}\chi^2\triangle t \sum_{n=0}^{l-1} ||\eta^{n+\frac{1}{2}}||^2 + C\nu^{-1}\triangle t \sum_{n=0}^{l-1} \inf_{q_h \in Q_h} ||p - q_h||^2$$

$$+ C\nu^{-1}\triangle t \sum_{n=0}^{l-1} \left(||\boldsymbol{u}_h^{n+\frac{1}{2}}|| \, ||\nabla \boldsymbol{u}_h^{n+\frac{1}{2}}|| \, ||\nabla \eta^{n+\frac{1}{2}}||^2 + ||\eta^{n+\frac{1}{2}}|| \, ||\nabla \eta^{n+\frac{1}{2}}|| \, ||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||^2 \right)$$

$$+ 2\triangle t \sum_{n=0}^{l-1} |\mathrm{Intp}(\boldsymbol{w}(t^n), \phi_h^{n+\frac{1}{2}})|. \tag{3.26}$$

From here, we further bound terms on the RHS of (3.26) using the discrete norms,

$$C\nu\triangle t \sum_{n=0}^{l-1} ||\nabla \eta^{n+\frac{1}{2}}||^2 \leq C\nu\triangle t \sum_{n=0}^{l} ||\nabla \eta^n||^2 \leq C\nu h^{2k}\triangle t \sum_{n=0}^{l} ||\boldsymbol{w}(t^n)||_{k+1}^2$$

$$\leq C\nu h^{2k}|||\boldsymbol{w}|||_{2,k+1}^2, \tag{3.27}$$

$$C\nu^{-1}\chi^2\triangle t \sum_{n=0}^{l-1} ||\eta^{n+\frac{1}{2}}||^2 \leq C\nu^{-1}\chi^2\triangle t \sum_{n=0}^{l} ||\eta^n||^2 \leq C\nu^{-1}\chi^2 h^{2k+2}\triangle t \sum_{n=0}^{l-1} ||\boldsymbol{w}(t^n)||_{k+1}^2$$

$$\leq C\nu^{-1}\chi^2 h^{2k+2}|||\boldsymbol{w}|||_{2,k+1}^2, \tag{3.28}$$

$$C\nu^{-1}\triangle t \sum_{n=0}^{l-1} \inf_{q_h \in Q_h} ||p - q_h||^2 \leq C\nu^{-1}\triangle t \sum_{n=0}^{l} h^{2k}||p||_k^2 \leq C\nu^{-1} h^{2k}|||p|||_{2,k}^2. \tag{3.29}$$

Next, we bound the larger summation in (3.26 with several applications of Young's inequality to consolidate all of the terms into the discrete norms. For the 2nd term, we also use proposition 3.2.1,

$$C\nu^{-1}\triangle t\sum_{n=0}^{l-1}||\eta^{n+\frac{1}{2}}||\,||\nabla\eta^{n+\frac{1}{2}}||\,||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}\triangle t\sum_{n=0}^{l-1}\Big(||\eta^{n+1}||\,||\nabla\eta^{n+1}||+||\eta^{n+1}||\,||\nabla\eta^{n}||+||\eta^{n}||\,||\nabla\eta^{n+1}||+||\eta^{n}||\,||\nabla\eta^{n}||\Big)||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}h^{2k+1}\triangle t\sum_{n=0}^{l-1}\Big(||\boldsymbol{w}(t^{n+1})||_{k+1}^2+||\boldsymbol{w}(t^{n+1})||_{k+1}\,||\boldsymbol{w}(t^{n})||_{k+1}+||\boldsymbol{w}(t^{n})||_{k+1}^2\Big)||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}h^{2k+1}\Big(\triangle t\sum_{n=0}^{l}||\boldsymbol{w}(t^n)||_{k+1}^4+\triangle t\sum_{n=0}^{l}||\nabla\boldsymbol{w}(t^n)||^4\Big)$$

$$= C\nu^{-1}h^{2k+1}\left(|||\boldsymbol{w}|||_{4,k+1}^4+|||\nabla\boldsymbol{w}|||_{4,0}^4\right), \tag{3.30}$$

$$C\nu^{-1}\triangle t\sum_{n=0}^{l-1}||\boldsymbol{u}_h^{n+\frac{1}{2}}||\,||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||\,||\nabla\eta^{n+\frac{1}{2}}||^2 \leq C\nu^{-1}\triangle t\sum_{n=0}^{l-1}||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||\,||\nabla\eta^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}\triangle t\sum_{n=0}^{l-1}||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||\Big(||\nabla\eta^{n+1}||^2+||\nabla\eta^{n}||^2\Big)$$

$$\leq C\nu^{-1}h^{2k}\triangle t\sum_{n=0}^{l-1}||\nabla\boldsymbol{u}_h^{n+\frac{1}{2}}||\Big(||\boldsymbol{w}(t^{n+1})||_{k+1}^2+||\boldsymbol{w}(t^{n})||_{k+1}^2\Big)$$

$$\leq C\nu^{-1}h^{2k}\Big(\triangle t\sum_{n=0}^{l}||\nabla\boldsymbol{u}_h^{n}||^2+\triangle t\sum_{n=0}^{l}||\boldsymbol{w}(t^n)||_{k+1}^4\Big)$$

$$\leq C\nu^{-1}h^{2k}\Big(|||\boldsymbol{w}|||_{4,k+1}^4+\nu^{-1}\left(||\boldsymbol{w}_h^0||^2+\nu^{-1}|||\mathbf{f}|||_{2,-1}^2\right)\Big). \tag{3.31}$$

Now we begin the process of bounding the terms in our interpolation error. The first few are all consequences of our consistency error formulas (1.6)-(1.7) and Lemma 1.2.10,

$$\left(\frac{\boldsymbol{w}(t^{n+1})-\boldsymbol{w}(t^n)}{\triangle t}-\boldsymbol{w}_t(t^{n+\frac{1}{2}}),\phi_h^{n+\frac{1}{2}}\right)\leq\frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2+C\nu^{-1}||\frac{\boldsymbol{w}(t^{n+1})-\boldsymbol{w}(t^n)}{\triangle t}-\boldsymbol{w}_t(t^{n+\frac{1}{2}})||^2$$

$$\leq\frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2+C\nu^{-1}(\triangle t)^3\int_{t^n}^{t^{n+1}}||\boldsymbol{w}_{ttt}||^2dt, \tag{3.32}$$

$$\chi\left(\left(I - G_N^h G_h\right) \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) \le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu^{-1}\chi^2||\left(I - G_N^h G_h\right)\boldsymbol{w}^{n+\frac{1}{2}}||^2$$

$$\le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu^{-1}\chi^2\left(\delta^{4N+4}||\boldsymbol{w}||_{2N+2}^2 + (\delta h^k + h^{k+1})^2||\boldsymbol{w}||_{k+1}^2\right), \quad (3.33)$$

$$\nu\left(\nabla\boldsymbol{w}^{n+\frac{1}{2}} - \nabla\boldsymbol{w}(t^{n+\frac{1}{2}}), \nabla\phi_h^{n+\frac{1}{2}}\right) \le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu||\nabla\boldsymbol{w}^{n+\frac{1}{2}} - \nabla\boldsymbol{w}(t^{n+\frac{1}{2}})||^2$$

$$\le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu(\triangle t)^3 \int_{t^n}^{t^{n+1}} ||\nabla\boldsymbol{w}_{tt}||^2 dt. \quad (3.34)$$

Next we will bound the non-linear terms in the interpolation error using equation (1.13) and (1.6),

$$b^*\left(\boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) - b^*(\boldsymbol{w}(t^{n+\frac{1}{2}}), \boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}})$$

$$= b^*\left(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}), \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) + b^*\left(\boldsymbol{w}(t^{n+\frac{1}{2}}), \boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}}\right)$$

$$\le C||\nabla\left(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})\right)||\,||\nabla\phi_h^{n+\frac{1}{2}}||\left(||\nabla\boldsymbol{w}^{n+\frac{1}{2}}|| + ||\nabla\boldsymbol{w}(t^{n+\frac{1}{2}})||\right)$$

$$\le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu^{-1}\left(||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2 + ||\nabla\boldsymbol{w}(t^{n+\frac{1}{2}})||^2\right)||\nabla\left(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})\right)||^2$$

$$\le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu^{-1}(\triangle t)^3\left(||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2 + ||\nabla\boldsymbol{w}(t^{n+\frac{1}{2}})||^2\right)\int_{t_n}^{t_{n+1}} ||\nabla\boldsymbol{w}_{tt}||^2 dt$$

$$\le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu^{-1}(\triangle t)^3\left(\int_{t_n}^{t_{n+1}} ||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^4 + ||\nabla\boldsymbol{w}(t^{n+\frac{1}{2}})||^4 + ||\nabla\boldsymbol{w}_{tt}||^4 dt\right)$$

$$\le \frac{\nu}{8}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + C\nu^{-1}(\triangle t)^4\left(||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^4 + ||\nabla\boldsymbol{w}(t^{n+\frac{1}{2}})||^4\right)$$

$$+ C\nu^{-1}(\triangle t)^3\int_{t_n}^{t_{n+1}} ||\nabla\boldsymbol{w}_{tt}||^4 dt. \quad (3.35)$$

Finally, we insert all of our bounds from (3.27)-(3.35) into (3.26) and after rearranging terms we have

$$||\phi_h^l||^2 + \frac{\nu\triangle t}{2}\sum_{n=0}^{l-1}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + 2\chi\triangle t\sum_{n=0}^{l-1}||\phi_h^{n+\frac{1}{2}*}||^2$$

$$\leq C\nu^{-3}\triangle t\sum_{n=0}^{l-1}||\phi_h^{n+\frac{1}{2}}||^2\,||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^4 + C\nu^{-1}h^{2k+1}\Big(|||\boldsymbol{w}|||_{4,k+1}^4 + |||\nabla\boldsymbol{w}|||_{4,0}^4\Big) + C\nu h^{2k}|||\boldsymbol{w}|||_{2,k+1}^2$$

$$+C\nu^{-1}h^{2k}\Big(|||\boldsymbol{w}|||_{4,k+1}^4 + \nu^{-1}(||\boldsymbol{w}_h^0||^2 + \nu^{-1}|||\mathbf{f}|||_{2,-1}^2)\Big) + C\nu^{-1}\chi^2 h^{2k+2}|||\boldsymbol{w}|||_{2,k+1}^2$$

$$+C\nu^{-1}h^{2k}|||p|||_{2,k}^2 + C\nu^{-1}\chi^2\Big(\delta^{4N+4}|||\boldsymbol{w}|||_{2,2N+2}^2 + (\delta h^k + h^{k+1})^2|||\boldsymbol{w}|||_{2,k+1}^2\Big)$$

$$+C\nu^{-1}(\triangle t)^4||\boldsymbol{w}_{ttt}||_{2,0}^2 + C\nu^{-1}(\triangle t)^4\Big(|||\nabla\boldsymbol{w}|||_{4,0}^4 + |||\nabla\boldsymbol{w}_{1/2}|||_{4,0}^4\Big)$$

$$+C\nu^{-1}(\triangle t)^4||\nabla\boldsymbol{w}_{tt}||_{4,0}^4 + C\nu(\triangle t)^4||\nabla\boldsymbol{w}_{tt}||_{2,0}^2. \tag{3.36}$$

We apply the Gronwall's lemma to (3.36) to hide the first term on the RHS, and this gives us

$$||\phi_h^l||^2 + \frac{\nu\triangle t}{2}\sum_{n=0}^{l-1}||\nabla\phi_h^{n+\frac{1}{2}}||^2 + 2\chi\triangle t\sum_{n=0}^{l-1}||\phi_h^{n+\frac{1}{2}*}||^2$$

$$\leq C(e^{\nu^{-3}})\bigg\{h^{2k+1}\left[\nu^{-1}(|||\boldsymbol{w}|||_{4,k+1}^4 + |||\nabla\boldsymbol{w}|||_{4,0}^4)\right] + \chi^2 h^{2k+2}\left[\nu^{-1}|||\boldsymbol{w}|||_{2,k+1}^2\right]$$

$$+h^{2k}\bigg[\nu|||\boldsymbol{w}|||_{2,k+1}^2 + \nu^{-1}\left(|||p|||_{2,k}^2 + |||\boldsymbol{w}|||_{4,k+1}^4 + \nu^{-1}(||\boldsymbol{w}_h^0||^2 + \nu^{-1}|||\mathbf{f}|||_{2,-1}^2)\right)\bigg]$$

$$+\chi^2\delta^{4N+4}\left(\nu^{-1}|||\boldsymbol{w}|||_{2,2N+2}^2\right) + \chi^2(\delta h^k + h^{k+1})^2\left(\nu^{-1}|||\boldsymbol{w}|||_{2,k+1}^2\right)$$

$$+(\triangle t)^4\bigg[\nu^{-1}\left(|||\boldsymbol{w}_{ttt}|||_{2,0}^2 + ||\nabla\boldsymbol{w}_{tt}||_{4,0}^4 + |||\nabla\boldsymbol{w}|||_{4,0}^4 + |||\nabla\boldsymbol{w}_{1/2}|||_{4,0}^4\right) + \nu||\nabla\boldsymbol{w}_{tt}||_{2,0}^2\bigg]\bigg\}. \tag{3.37}$$

The final statement of the proof is then obtained from (3.37) using the triangle inequality.

$\square$

3.2.2 EMAC Error Estimate

Note that the error estimate in the previous section is of the order $O(e^{\nu^{-3}})$. We will now see that this dependence is improved to $O(e^{\nu^{-1}})$ for the EMAC scheme. We present an abridged proof of the error estimate for the EMAC scheme, showing only the parts which differ from the previous proof

as both proofs are mostly identical except for our choice of $\boldsymbol{U}$ and the treatment of the non-linear term.

Theorem 3.2.3. Let $(\boldsymbol{w}(t), p(t))$ be a smooth enough solution of the NSE satisfying no-slip boundary conditions, $\mathbf{f} \in L^\infty(0, T; H^{-1}(\Omega))$ and $\boldsymbol{w}_0 \in H^1(\Omega)$. Let (u_h^n, p_h^n), for $n = 0, 1, 2, ..., l$ be the solution to the EMAC scheme given by (3.5)-(3.6), using (P_k, P_{k-1}) elements for some $k \geq 2$. Then the error in the discrete solution satisfies

$$|||\boldsymbol{w} - \boldsymbol{u}_h|||_{\infty,0}^2 + \frac{\nu \triangle t}{2} \sum_{n=0}^{l-1} ||\nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{u}_h^{n+\frac{1}{2}})||^2 + 2\chi\triangle t \sum_{n=0}^{l-1} ||(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{u}_h^{n+\frac{1}{2}})^*||^2$$
$$\leq C(e^{\nu^{-1}})\left\{ h^{2k} + h^{2k+1} + \chi^2 \delta^{4N+4} + \chi^2(\delta h^k + h^{k+1})^2 + (\triangle t)^4 \right\}. \tag{3.38}$$

Proof. The proof follows the SKEW estimate very closely, one just needs to replace the nonlinear terms b^* with c for most of the proof. However, two major parts of the proof do change, and we highlight the changes here.

We let $\boldsymbol{U} = P_{St}(\boldsymbol{w})$ be the Stokes projection instead of the L^2 projection which was used for SKEW. Hence in equation (3.18), the terms $\nu(\nabla\eta^{n+\frac{1}{2}}, \nabla\phi_h^{n+\frac{1}{2}}) + (p(t^{n+\frac{1}{2}}) - q_h, \nabla \cdot \phi_h^{n+\frac{1}{2}}) = 0$, while the term $\frac{1}{\triangle t}(\eta^{n+1} - \eta^n, \phi_h^{n+\frac{1}{2}}) \neq 0$, and thus we will need the following estimate for this new term which follows from Lemma 1.2.8 and the fact that if $\boldsymbol{U}^n$ is the Stokes projection of $\boldsymbol{w}^n$ for all n, then $\dfrac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\triangle t}$ is the Stokes projection of $\dfrac{\boldsymbol{w}^{n+1} - \boldsymbol{w}^n}{\triangle t}$,

$$\frac{1}{\triangle t}(\eta^{n+1} - \eta^n, \phi_h^{n+\frac{1}{2}}) = \left(\frac{\boldsymbol{w}^{n+1} - \boldsymbol{w}^n}{\triangle t} - \frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\triangle t}, \phi_h^{n+\frac{1}{2}}\right)$$

$$\leq C \left\|\frac{\boldsymbol{w}^{n+1} - \boldsymbol{w}^n}{\triangle t} - \frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\triangle t}\right\| \|\nabla\phi_h^{n+\frac{1}{2}}\|$$

$$\leq C\nu^{-1} \left\|\frac{\boldsymbol{w}^{n+1} - \boldsymbol{w}^n}{\triangle t} - \frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\triangle t}\right\|^2 + \frac{\nu}{12}\|\nabla\phi_h^{n+\frac{1}{2}}\|^2$$

$$\leq C\nu^{-1}h^2 \left(\inf_{\boldsymbol{v}_h \in X_h}\left\|\nabla(\frac{\boldsymbol{w}^{n+1} - \boldsymbol{w}^n}{\triangle t} - \boldsymbol{v}_h)\right\|^2 + \inf_{q_h \in Q_h}\left\|\frac{p^{n+1} - p^n}{\triangle t} - q_h\right\|^2\right) + \frac{\nu}{12}\|\nabla\phi_h^{n+\frac{1}{2}}\|^2$$

$$\leq C\nu^{-1}h^{2k+2} \left(\left|\frac{\boldsymbol{w}^{n+1} - \boldsymbol{w}^n}{\triangle t}\right|_{k+1}^2 + \left|\frac{p^{n+1} - p^n}{\triangle t}\right|_k^2\right) + \frac{\nu}{12}\|\nabla\phi_h^{n+\frac{1}{2}}\|^2$$

$$\leq C\nu^{-1}h^{2k+2} \left(|\boldsymbol{w}_t^n|_{k+1}^2 + |p_t^n|_k^2\right) + \frac{\nu}{12}\|\nabla\phi_h^{n+\frac{1}{2}}\|^2. \tag{3.39}$$

Next we look at the bounds for the non-linear terms. First we break down the term outside the interpolation error as follows,

$$c(\boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) - c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}})$$

$$=2\left[\left(D(\boldsymbol{w}^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) - \left(D(\boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right)\right]$$

$$+\left[\left((\nabla \cdot \boldsymbol{w}^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right) - \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}\right)\right]. \tag{3.40}$$

We show the breakdown for the left bracketed term in (3.40), which is performed using the definitions of $\eta^{n+\frac{1}{2}}$ and $\phi_h^{n+\frac{1}{2}}$,

$$\left(D(\boldsymbol{w}^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left(D(\boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$= \left(D(\eta^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left(D(\boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$= \left(D(\eta^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\boldsymbol{U}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$- \left(D(\boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$= \left(D(\eta^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$+ \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left(D(\boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$= \left(D(\eta^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$+ \left(D(\phi_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$= \left(D(\eta^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$+ \left(D(\phi_h^{n+\frac{1}{2}})\boldsymbol{U}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left(D(\phi_h^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right). \tag{3.41}$$

Similarly, for the right bracketed term in (3.40), we obtain

$$\left((\nabla \cdot \boldsymbol{w}^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left((\nabla \cdot \boldsymbol{u}_h^{n+\frac{1}{2}})\boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$= \left((\nabla \cdot \eta^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left((\nabla \cdot \boldsymbol{U}^{n+\frac{1}{2}})\eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) + \left((\nabla \cdot \boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right)$$

$$+ \left((\nabla \cdot \phi_h^{n+\frac{1}{2}})\boldsymbol{U}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left((\nabla \cdot \phi_h^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right). \tag{3.42}$$

We also further break down the 4th term from the last line of equation (3.41) using the trilinear form properties (1.14)-(1.16) to get the following,

$$2\left(D(\phi_h^{n+\frac{1}{2}})\boldsymbol{U}^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right) = 2\left(\boldsymbol{U}^{n+\frac{1}{2}}\cdot D(\phi_h^{n+\frac{1}{2}}),\phi_h^{n+\frac{1}{2}}\right)$$

$$=\left(\boldsymbol{U}^{n+\frac{1}{2}}\cdot\nabla\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)+\left(\boldsymbol{U}^{n+\frac{1}{2}}\cdot(\nabla\phi_h^{n+\frac{1}{2}})^T,\phi_h^{n+\frac{1}{2}}\right)$$

$$=\left(\boldsymbol{U}^{n+\frac{1}{2}}\cdot\nabla\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)+\left((\nabla\phi_h^{n+\frac{1}{2}})^T\phi_h^{n+\frac{1}{2}},\boldsymbol{U}^{n+\frac{1}{2}}\right)$$

$$=\left(\boldsymbol{U}^{n+\frac{1}{2}}\cdot\nabla\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)+\left(\phi_h^{n+\frac{1}{2}}\cdot\nabla\phi_h^{n+\frac{1}{2}},\boldsymbol{U}^{n+\frac{1}{2}}\right)$$

$$=-\frac{1}{2}\left((\nabla\cdot\boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)-\left(\phi_h^{n+\frac{1}{2}}\cdot\nabla\boldsymbol{U}^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)-\left((\nabla\cdot\phi_h^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}},\boldsymbol{U}^{n+\frac{1}{2}}\right)$$

$$=-\frac{1}{2}\left((\nabla\cdot\boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)-\left(\phi_h^{n+\frac{1}{2}}\cdot D(\boldsymbol{U}^{n+\frac{1}{2}}),\phi_h^{n+\frac{1}{2}}\right)-\left((\nabla\cdot\phi_h^{n+\frac{1}{2}})\boldsymbol{U}^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)$$

$$=-\frac{1}{2}\left((\nabla\cdot\boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)-\left(D(\boldsymbol{U}^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)-\left((\nabla\cdot\phi_h^{n+\frac{1}{2}})\boldsymbol{U}^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right). \quad (3.43)$$

Combining the breakdowns from equations (3.41) - (3.43), and noting that the terms

$$2\left(D(\phi_h^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)+\left((\nabla\cdot\phi_h^{n+\frac{1}{2}})\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)=c(\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}})=0, \text{ we have}$$

$$c(\boldsymbol{w}^{n+\frac{1}{2}},\boldsymbol{w}^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}})-c(\boldsymbol{u}_h^{n+\frac{1}{2}},\boldsymbol{u}_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}})=\left([2D(\eta^{n+\frac{1}{2}})+\nabla\cdot\eta^{n+\frac{1}{2}}]\boldsymbol{w}^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)$$

$$+\left([2D(\boldsymbol{U}^{n+\frac{1}{2}})+\nabla\cdot\boldsymbol{U}^{n+\frac{1}{2}}]\eta^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right)$$

$$+\left([D(\boldsymbol{U}^{n+\frac{1}{2}})+\frac{1}{2}\nabla\cdot\boldsymbol{U}^{n+\frac{1}{2}}]\phi_h^{n+\frac{1}{2}},\phi_h^{n+\frac{1}{2}}\right). \quad (3.44)$$

The first two terms on the right hand side of (3.44) can be bounded using equations (1.12), (1.14), (1.16) and Lemma 1.2.7 with $r = 2$,

$$\left| \left([2D(\eta^{n+\frac{1}{2}}) + \nabla \cdot \eta^{n+\frac{1}{2}}] \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) \right| \leq \left| \left(\eta^{n+\frac{1}{2}} \cdot \nabla \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) \right| + \left| \left(\eta^{n+\frac{1}{2}} \cdot \nabla \phi_h^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}} \right) \right|$$

$$+ 2 \left| \left(\boldsymbol{w}^{n+\frac{1}{2}} \cdot D(\eta^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$\leq C \| \nabla \phi_h^{n+\frac{1}{2}} \| \left(\| \eta^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \eta^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \| \right.$$

$$\left. + \| \boldsymbol{w}^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \eta^{n+\frac{1}{2}} \| \right)$$

$$\leq C \nu^{-1} \left(\| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \|^2 \| \eta^{n+\frac{1}{2}} \| \ \| \nabla \eta^{n+\frac{1}{2}} \| \right.$$

$$\left. + \| \nabla \eta^{n+\frac{1}{2}} \|^2 \| \boldsymbol{w}^{n+\frac{1}{2}} \| \ \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \| \right) + \frac{\nu}{12} \| \nabla \phi_h^{n+\frac{1}{2}} \|^2, \qquad (3.45)$$

$$\left| \left([2D(\boldsymbol{U}^{n+\frac{1}{2}}) + \nabla \cdot \boldsymbol{U}^{n+\frac{1}{2}}] \eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) \right| \leq \left| \left(\boldsymbol{U}^{n+\frac{1}{2}} \cdot \nabla \eta^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) \right| + \left| \left(\boldsymbol{U}^{n+\frac{1}{2}} \cdot \nabla \phi_h^{n+\frac{1}{2}}, \eta^{n+\frac{1}{2}} \right) \right|$$

$$+ 2 \left| \left(\eta^{n+\frac{1}{2}} \cdot D(\boldsymbol{U}^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$\leq C \| \nabla \phi_h^{n+\frac{1}{2}} \| \left(\| \eta^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \eta^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \| \right.$$

$$\left. + \| \boldsymbol{w}^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \|^{\frac{1}{2}} \| \nabla \eta^{n+\frac{1}{2}} \| \right)$$

$$\leq C \nu^{-1} \left(\| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \|^2 \| \eta^{n+\frac{1}{2}} \| \ \| \nabla \eta^{n+\frac{1}{2}} \| \right.$$

$$\left. + \| \nabla \eta^{n+\frac{1}{2}} \|^2 \| \boldsymbol{w}^{n+\frac{1}{2}} \| \ \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \| \right) + \frac{\nu}{12} \| \nabla \phi_h^{n+\frac{1}{2}} \|^2. \qquad (3.46)$$

The next estimate for the third term is very similar, only now we use Lemma 1.2.7 with $r = 3$. Note that this estimate is where we get the dependence of our error on $e^{\nu^{-1}}$, as we will hide the ν^{-1} term on the RHS with Gronwall's inequality at the end of the proof.

$$\left| \left([D(\boldsymbol{U}^{n+\frac{1}{2}}) + \frac{1}{2} \nabla \cdot \boldsymbol{U}^{n+\frac{1}{2}}] \phi_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) \right| \leq \frac{3}{2} \| D(\boldsymbol{U}^{n+\frac{1}{2}}) \|_{L^3} \| \phi_h^{n+\frac{1}{2}} \|_{L^2} \| \phi_h^{n+\frac{1}{2}} \|_{L^6}$$

$$\leq C \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \|_{L^3} \| \phi_h^{n+\frac{1}{2}} \|_{L^2} \| \nabla \phi_h^{n+\frac{1}{2}} \| \leq C \nu^{-1} \| \nabla \boldsymbol{w}^{n+\frac{1}{2}} \|_{L^3}^2 \| \phi_h^{n+\frac{1}{2}} \|_{L^2}^2 + \frac{\nu}{12} \| \nabla \phi_h^{n+\frac{1}{2}} \|^2. \qquad (3.47)$$

Combining these estimates (3.45)-(3.47), we obtain the following bound for the EMAC nonlinear terms

$$\left| c(\boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) - c(\boldsymbol{u}_h^{n+\frac{1}{2}}, \boldsymbol{u}_h^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) \right| \leq C\nu^{-1} \Bigg[||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||^2 ||\eta^{n+\frac{1}{2}}|| \; ||\nabla \eta^{n+\frac{1}{2}}||$$

$$+ ||\nabla \eta^{n+\frac{1}{2}}||^2 ||\boldsymbol{w}^{n+\frac{1}{2}}|| \; ||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||$$

$$+ ||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||^2_{L^3} ||\phi_h^{n+\frac{1}{2}}||^2 \Bigg] + \frac{\nu}{4} ||\nabla \phi_h^{n+\frac{1}{2}}||^2. \quad (3.48)$$

Next we deal with the non-linear term in the interpolation error. We have

$$\left| c(\boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) - c(\boldsymbol{w}(t^{n+\frac{1}{2}}), \boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}}) \right|$$

$$\leq 2 \left| \left(D(\boldsymbol{w}^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left(D(\boldsymbol{w}(t^{n+\frac{1}{2}}))\boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$+ \left| \left((\nabla \cdot \boldsymbol{w}^{n+\frac{1}{2}})\boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}} \right) - \left((\nabla \cdot \boldsymbol{w}(t^{n+\frac{1}{2}}))\boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$\leq 2 \left| \left(D(\boldsymbol{w}^{n+\frac{1}{2}})(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})), \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))\boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$+ \left| \left((\nabla \cdot \boldsymbol{w}^{n+\frac{1}{2}})(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})), \phi_h^{n+\frac{1}{2}} \right) + \left(\nabla \cdot (\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))\boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|. \quad (3.49)$$

Each term on the RHS of (3.49) can be bound using (1.13) and (1.16),

$$\left| \left(D(\boldsymbol{w}^{n+\frac{1}{2}})(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})), \phi_h^{n+\frac{1}{2}} \right) + \left(D(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))\boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$\leq \left| \left((\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})) \cdot D(\boldsymbol{w}^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right| + \left| \left(\boldsymbol{w}(t^{n+\frac{1}{2}}) \cdot D(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$\leq C ||\nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))|| \; ||\nabla \phi_h^{n+\frac{1}{2}}|| \left(||\nabla \boldsymbol{w}^{n+\frac{1}{2}}|| + ||\nabla \boldsymbol{w}(t^{n+\frac{1}{2}})|| \right), \quad (3.50)$$

$$\left| \left((\nabla \cdot \boldsymbol{w}^{n+\frac{1}{2}})(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}})), \phi_h^{n+\frac{1}{2}} \right) + \left(\nabla \cdot (\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))\boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$\leq \left| \left(\boldsymbol{w}^{n+\frac{1}{2}} \cdot \nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))^T, \phi_h^{n+\frac{1}{2}} \right) \right| + \left| \left(\boldsymbol{w}(t^{n+\frac{1}{2}}) \cdot \nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))^T, \phi_h^{n+\frac{1}{2}} \right) \right|$$

$$\leq C ||\nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))|| \; ||\nabla \phi_h^{n+\frac{1}{2}}|| \left(||\nabla \boldsymbol{w}^{n+\frac{1}{2}}|| + ||\nabla \boldsymbol{w}(t^{n+\frac{1}{2}})|| \right). \quad (3.51)$$

46

Plugging the estimates from (3.50) and (3.51) into (3.49) and utilizing equation (1.6), we get

$$\left| c(\boldsymbol{w}^{n+\frac{1}{2}}, \boldsymbol{w}^{n+\frac{1}{2}}, \phi_h^{n+\frac{1}{2}}) - c(\boldsymbol{w}(t^{n+\frac{1}{2}}), \boldsymbol{w}(t^{n+\frac{1}{2}}), \phi_h^{n+\frac{1}{2}}) \right|$$

$$\leq C ||\nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))|| \, ||\nabla \phi_h^{n+\frac{1}{2}}|| \left(||\nabla \boldsymbol{w}^{n+\frac{1}{2}}|| + ||\nabla \boldsymbol{w}(t^{n+\frac{1}{2}})|| \right)$$

$$\leq C\nu^{-1} \left[||\nabla(\boldsymbol{w}^{n+\frac{1}{2}} - \boldsymbol{w}(t^{n+\frac{1}{2}}))||^2 \left(||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||^2 + ||\nabla \boldsymbol{w}(t^{n+\frac{1}{2}})||^2 \right) \right] + \frac{\nu}{8} ||\nabla \phi_h^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}(\triangle t)^3 \int_{t_n}^{t_{n+1}} ||\nabla \boldsymbol{w}_{tt}||^2 dt \left[||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||^2 + ||\nabla \boldsymbol{w}(t^{n+\frac{1}{2}})||^2 \right] + \frac{\nu}{8} ||\nabla \phi_h^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}(\triangle t)^4 \left(||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||^4 + ||\nabla \boldsymbol{w}(t^{n+\frac{1}{2}})||^4 \right)$$

$$+ C\nu^{-1}(\triangle t)^3 \int_{t_n}^{t_{n+1}} ||\nabla \boldsymbol{w}_{tt}||^4 dt + \frac{\nu}{8} ||\nabla \phi_h^{n+\frac{1}{2}}||^2. \tag{3.52}$$

Now we insert equations (3.32)-(3.34) and (3.52) into an analogue of equation (3.16) for the EMAC. After summing from $n = 0$ to $l - 1$ and multiplying by $2\triangle t$, the total contribution from our interpolation error is given by the following

$$2\triangle t \sum_{n=0}^{l-1} |\text{Intp}(\boldsymbol{w}(t^n), \phi_h^{n+\frac{1}{2}})| \leq \frac{\nu \triangle t}{2} \sum_{n=0}^{l-1} ||\nabla \phi_h^{n+\frac{1}{2}}||^2$$

$$+ C\nu^{-1}\chi^2 \left[\delta^{4N+4} |||\boldsymbol{w}|||_{2,2N+2}^2 + (\delta h^k + h^{k+1})^2 |||\boldsymbol{w}|||_{2,k+1}^2 \right]$$

$$+ C\nu^{-1}(\triangle t)^4 \left[||\boldsymbol{w}_{ttt}||_{2,0}^2 + ||\nabla \boldsymbol{w}_{tt}||_{4,0}^4 + ||\nabla \boldsymbol{w}_{tt}||_{2,0}^2 + |||\nabla \boldsymbol{w}|||_{4,0}^4 + |||\nabla \boldsymbol{w}_{1/2}|||_{4,0}^4 \right]. \tag{3.53}$$

Now we insert the estimates from equations (3.24), (3.39) and (3.48) into equation (3.18) with c instead of b^* to create the analogue of equation (3.26) for the EMAC proof,

$$||\phi_h^l||^2 + \frac{\nu\triangle t}{2}\sum_{n=0}^{l-1}||\phi_h^{n+\frac{1}{2}}||^2 + 2\chi\triangle t\sum_{n=0}^{l-1}||\phi_h^{n+\frac{1}{2}}{}^*||^2$$

$$\leq C\nu^{-1}\triangle t\sum_{n=0}^{l-1}\left[||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2||\eta^{n+\frac{1}{2}}||\,||\nabla\eta^{n+\frac{1}{2}}|| + ||\nabla\eta^{n+\frac{1}{2}}||^2||\boldsymbol{w}^{n+\frac{1}{2}}||\,||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||\right.$$

$$+||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||_{L^3}^2\,||\phi_h^{n+\frac{1}{2}}||^2 + \chi^2||\eta^{n+\frac{1}{2}}||^2\bigg] + 2\triangle t\sum_{n=0}^{l-1}|\mathrm{Intp}(\boldsymbol{w}(t^n),\phi_h^{n+\frac{1}{2}})|$$

$$+C\nu^{-1}h^2\triangle t\sum_{n=0}^{l-1}\left(|\boldsymbol{w}_t|_{k+1}^2 + |p_t|_k^2\right). \tag{3.54}$$

Next, we bound one more term inside the bracketed sum in equation (3.54) that was not previously bounded in the SKEW proof. The bound is as follows, making use of standard inequalities,

$$C\nu^{-1}\triangle t\sum_{n=0}^{l-1}||\nabla\eta^{n+\frac{1}{2}}||^2||\boldsymbol{w}^{n+\frac{1}{2}}||\,||\nabla\boldsymbol{w}^{n+\frac{1}{2}}|| \leq C\nu^{-1}\triangle t\sum_{n=0}^{l-1}||\nabla\eta^{n+\frac{1}{2}}||^2||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}\triangle t\sum_{n=0}^{l-1}\left(||\nabla\eta^{n+1}||^2 + ||\nabla\eta^n||^2\right)||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}h^{2k}\triangle t\sum_{n=0}^{l-1}\left(||\nabla\boldsymbol{w}(t^{n+1})||_{k+1}^2 + ||\nabla\boldsymbol{w}(t^n)||_{k+1}^2\right)||\nabla\boldsymbol{w}^{n+\frac{1}{2}}||^2$$

$$\leq C\nu^{-1}h^{2k}\left(\triangle t\sum_{n=0}^{l}||\nabla\boldsymbol{w}(t^n)||_{k+1}^4 + \triangle t\sum_{n=0}^{l}||\nabla\boldsymbol{w}(t^n)||^4\right)$$

$$\leq C\nu^{-1}h^{2k}\left(|||\nabla\boldsymbol{w}|||_{4,k+1}^4 + |||\nabla\boldsymbol{w}|||_{4,0}^4\right). \tag{3.55}$$

Now, inserting the bounds from equations (3.28), (3.30), (3.53), and (3.55) into (3.54), we obtain

$$
||\phi_h^l||^2 + \frac{\nu \triangle t}{2} \sum_{n=0}^{l-1} ||\phi_h^{n+\frac{1}{2}}||^2 + 2\chi \triangle t \sum_{n=0}^{l-1} ||\phi_h^{n+\frac{1}{2}}{}^*||^2
$$

$$
\leq C\nu^{-1} \triangle t \sum_{n=0}^{l-1} ||\nabla \boldsymbol{w}^{n+\frac{1}{2}}||_{L^3}^2 ||\phi_h^{n+\frac{1}{2}}||^2
$$

$$
+ C\nu^{-1} h^{2k+1} \left(|||\boldsymbol{w}|||_{4,k+1}^4 + |||\nabla \boldsymbol{w}|||_{4,0}^4 + |||\boldsymbol{w}_t|||_{2,k+1}^2 + |||p_t|||_{2,k}^2 \right)
$$

$$
+ C\nu^{-1} h^{2k} \left(|||\nabla \boldsymbol{w}|||_{4,k+1}^4 + |||\nabla \boldsymbol{w}|||_{4,0}^4 \right) + C\nu^{-1} \chi^2 h^{2k+2} |||\boldsymbol{w}|||_{2,k+1}^2
$$

$$
+ C\nu^{-1} \chi^2 \left[\delta^{4N+4} |||\boldsymbol{w}|||_{2,2N+2}^2 + (\delta h^k + h^{k+1})^2 |||\boldsymbol{w}|||_{2,k+1}^2 \right]
$$

$$
+ C\nu^{-1} (\triangle t)^4 \left[||\boldsymbol{w}_{ttt}||_{2,0}^2 + ||\nabla \boldsymbol{w}_{tt}||_{4,0}^4 + ||\nabla \boldsymbol{w}_{tt}||_{2,0}^2 + |||\nabla \boldsymbol{w}|||_{4,0}^4 + |||\nabla \boldsymbol{w}_{1/2}|||_{4,0}^4 \right]. \tag{3.56}
$$

Finally, our last step is to use Gronwall's inequality on (3.56) to again hide the first term on the RHS. Rearranging terms, we end up with the following

$$
||\phi_h^l||^2 + \frac{\nu \triangle t}{2} \sum_{n=0}^{l-1} ||\phi_h^{n+\frac{1}{2}}||^2 + 2\chi \triangle t \sum_{n=0}^{l-1} ||\phi_h^{n+\frac{1}{2}}{}^*||^2
$$

$$
\leq C(e^{\nu^{-1}}) \Bigg\{ h^{2k} \left[\nu^{-1} (|||\nabla \boldsymbol{w}|||_{4,k+1}^4 + |||\nabla \boldsymbol{w}|||_{4,0}^4) \right] + \chi^2 h^{2k+2} \left[\nu^{-1} |||\boldsymbol{w}|||_{2,k+1}^2 \right]
$$

$$
+ h^{2k+1} \left[\nu^{-1} (|||\boldsymbol{w}|||_{4,k+1}^4 + |||\nabla \boldsymbol{w}|||_{4,0}^4 + |||\boldsymbol{w}_t|||_{2,k+1}^2 + |||p_t|||_{2,k}^2) \right]
$$

$$
+ \chi^2 \delta^{4N+4} \left[\nu^{-1} |||\boldsymbol{w}|||_{2,2N+2}^2 \right] + \chi^2 (\delta h^k + h^{k+1})^2 \left[\nu^{-1} |||\boldsymbol{w}|||_{2,k+1}^2 \right]
$$

$$
+ (\triangle t)^4 \left[\nu^{-1} (||\boldsymbol{w}_{ttt}||_{2,0}^2 + ||\nabla \boldsymbol{w}_{tt}||_{4,0}^4 + ||\nabla \boldsymbol{w}_{tt}||_{2,0}^2 + |||\nabla \boldsymbol{w}|||_{4,0}^4 + |||\nabla \boldsymbol{w}_{1/2}|||_{4,0}^4) \right] \Bigg\}. \tag{3.57}
$$

The proof is then completed by applying the triangle inequality.

$\square$

3.3   Numerical Experiments

In this section, we present some numerical experiments that test the convergence rates and conservation properties of both the SKEW and EMAC schemes for TRM. All simulations in this section were performed using the Freefem++ open source PDE solver package. Details on this package

can be found here [49]. All the experiments were conducted with a 3 step Newton linearization implemented for the non-linear term. The Newton loop was coded to break early if the error is within a tolerance of 10^{-9}.

3.3.1 Taylor Green Vortex Problem

In this section we will consider the well known Taylor Green Vortex problem, which was first studied in [60]. This 2D problem is very common for testing convergence rates of schemes since the solution is known. In the domain $\Omega = (0,1)^2$, its solution for the NSE (with $\mathbf{f} = 0$) is given by

$$u_1(x,y,t) = -\cos(\omega\pi x)\sin(\omega\pi y)e^{-2\nu\omega^2\pi^2 t},$$

$$u_2(x,y,t) = \sin(\omega\pi x)\cos(\omega\pi y)e^{-2\nu\omega^2\pi^2 t},$$

$$p(x,y,t) = -\frac{1}{4}(\cos(2\omega\pi x) + \cos(2\omega\pi y))e^{-2\nu\omega^2\pi^2 t}.$$

This solution consists of an $\omega \times \omega$ array of vortices which decay as $t \to \infty$. We use a uniform triangular mesh with m subdivisions along each edge, and we vary the value of m from 16 to 96. We chose model parameter $\omega = 1$, final time $T = 0.1$, time step $\triangle t = 0.0005$, time relaxation coefficient $\chi = 0.1$, viscosity $\nu = 10^{-2}$ and filter width $\delta = \frac{1}{m}$. The initial condition of this problem is $\boldsymbol{u}_0 = \boldsymbol{u}(0)$. Dirichlet boundary conditions are implemented based on the true solution.

The results for the convergence rates are presented in tables 3.1 through 3.4 for NSE and TRM with SKEW and EMAC schemes and different orders of deconvolution. We used Taylor Hood P2/P1 finite elements. From these results, we can see that the rates obtained for NSE and TRM are in agreement with the error estimates proved in theorems 3.2.2 and 3.2.3. For TRM with order of deconvolution N=0, we can see the influence of the δ^{2N+2} term, which gives us a lower rate as can be seen in Table 3.2.

Table 3.1: Convergence rates for NSE

	SKEW				EMAC			
m	L2 Inf	rate	Grad L2	rate	L2 Inf	rate	Grad L2	rate
16	2.6664e-04	-	9.8512e-03	-	5.0993e-04	-	1.7781e-02	-
32	1.9057e-05	3.81	1.5274e-03	2.69	3.4320e-05	3.89	2.6320e-03	2.76
48	4.4301e-06	3.60	5.5093e-04	2.51	7.1769e-06	3.86	8.5053e-04	2.79
64	1.6512e-06	3.43	2.7950e-04	2.36	2.4315e-06	3.76	3.9214e-04	2.69
80	7.8844e-07	3.31	1.6900e-04	2.25	1.0750e-06	3.66	2.2024e-04	2.59
96	4.5250e-07	3.05	1.1344e-04	2.19	6.1041e-07	3.10	1.3990e-04	2.49

Table 3.2: Convergence rates for TRM, N=0

	SKEW				EMAC			
m	L2 Inf	rate	Grad L2	rate	L2 Inf	rate	Grad L2	rate
16	2.7949e-04	-	9.8464e-03	-	5.1301e-04	-	1.7771e-02	-
32	3.7730e-05	2.89	1.5302e-03	2.69	4.6889e-05	3.45	2.6333e-03	2.75
48	1.6561e-05	2.03	5.5349e-04	2.51	1.7472e-05	2.43	8.5212e-04	2.78
64	9.5181e-06	1.93	2.8148e-04	2.35	9.6873e-06	2.05	3.9352e-04	2.69
80	6.2065e-06	1.92	1.6900e-04	2.25	6.2531e-06	1.96	2.2140e-04	2.58
96	4.3779e-06	1.91	1.1463e-04	2.18	4.3945e-06	1.93	1.4086e-04	2.48

Table 3.3: Convergence rates for TRM, N=1

	SKEW				EMAC			
m	L2 Inf	rate	Grad L2	rate	L2 Inf	rate	Grad L2	rate
16	2.6729e-04	-	9.8453e-03	-	5.1016e-04	-	1.7771e-02	-
32	1.9770e-05	3.76	1.5274e-03	2.69	3.4344e-05	3.45	2.6317e-03	2.75
48	4.7656e-06	3.51	5.5102e-04	2.51	7.2649e-06	2.43	8.5055e-04	2.78
64	1.8197e-06	3.35	2.7955e-04	2.36	2.5146e-06	2.05	3.9216e-04	2.69
80	8.8892e-07	3.21	1.6903e-04	2.25	1.1389e-06	1.96	2.2027e-04	2.58
96	5.0922e-07	3.05	1.1345e-04	2.18	6.1463e-07	1.93	1.3991e-04	2.48

Table 3.4: Convergence rates for TRM, N=2

	SKEW				EMAC			
m	L2 Inf	rate	Grad L2	rate	L2 Inf	rate	Grad L2	rate
16	2.6675e-04	-	9.8455e-03	-	5.0980e-04	-	1.7771e-02	-
32	1.9064e-05	3.80	1.5272e-03	2.69	3.4321e-05	3.89	2.6317e-03	2.76
48	4.4311e-06	3.60	5.5093e-04	2.51	7.1774e-06	3.86	8.5050e-04	2.79
64	1.6514e-06	3.43	2.7950e-04	2.35	2.4316e-06	3.76	3.9213e-04	2.69
80	7.8859e-07	3.31	1.6901e-04	2.25	1.0750e-06	3.66	2.2024e-04	2.58
96	4.5250e-07	3.05	1.1344e-04	2.18	6.1041e-07	3.10	1.3990e-04	2.49

3.3.2   3D Ethier-Steinman Problem

Here, we consider the Ethier-Steinman exact Navier-Stokes solution on the unit cube $[-1, 1]^3$,

which was developed in [25]. The exact solution in terms of the viscosity ν and given constants a and d is given by

$$
\begin{aligned}
u_1 &= -a(e^{ax}\sin(ay+dz) + e^{az}cos(ax+dy))e^{-\nu d^2 t}, \\[2ex]
u_2 &= -a(e^{ay}\sin(az+dx) + e^{ax}cos(ay+dz))e^{-\nu d^2 t}, \\[2ex]
u_3 &= -a(e^{az}\sin(ax+dy) + e^{ay}cos(az+dx))e^{-\nu d^2 t}, \\[2ex]
p &= -\frac{a^2}{2}(e^{2ax} + e^{2ay} + e^{2az} + 2\sin(ax+dy)cos(az+dx)e^{a(y+z)} \\[2ex]
& \quad +2\sin(ay+dz)cos(ax+dy)e^{a(z+x)} \\[2ex]
& \quad +2\sin(az+dx)cos(ay+dz)e^{a(x+y)})e^{-\nu d^2 t}.
\end{aligned}
$$

In the following tests, we used $a = 1.25$, $d = 1$, viscosity $\nu = 10^{-4}$, $\chi = 1$, time step $\triangle t = 0.01$ and final time $T = 1$. We also ran experiments using a scaled filter width $\delta_N = 0.1\sqrt{N+1}$ versus an unscaled filter width $\delta = 0.1$. The initial condition of this problem is $\boldsymbol{u}_0 = \boldsymbol{u}(0)$, and Dirichlet boundary conditions are implemented based on the true solution. The standard P2/P1 elements were used on a uniform tetrahedral mesh over the unit cube with 6,000 tetrahedral elements. Note that over the entire time interval, the solution will not change very much from the initial condition due to the small size of ν.

We present the results of the L^2 and H^1 errors for both the EMAC and SKEW simulations in Figures 3.1-3.3. Based on Figure 3.1, it is clear that the EMAC scheme reduces the L^2 and H^1 solution error by a significant amount versus its SKEW counterpart, and that the error for the TRM schemes for $N = 0$ through $N = 2$ are much better than if one simply implements a traditional NSE scheme. The scaling of the filter width does not produce any significant changes in the overall errors of the schemes versus unscaled, however we observe that for unscaled filter width,

the error slightly increases as you increase N (see Figure 3.3), while for the scaled filter width, the

error stays the same over time as you increase N (see Figure 3.2).

Figure 3.1: Error plots for NSE and TRM with scaled filter width δ_N.

(a) EMAC　　　　　　　　　　　　　　　　(b) SKEW
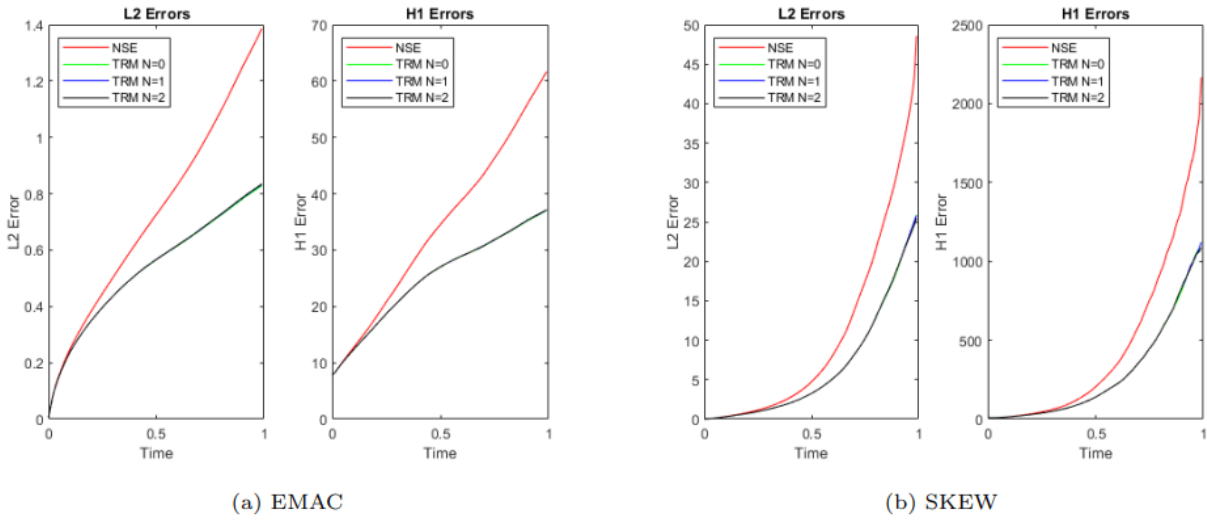
Figure 3.2: Error plots for TRM with scaled filter width δ_N.

(a) EMAC　　　　　　　　　　　　　　　　(b) SKEW
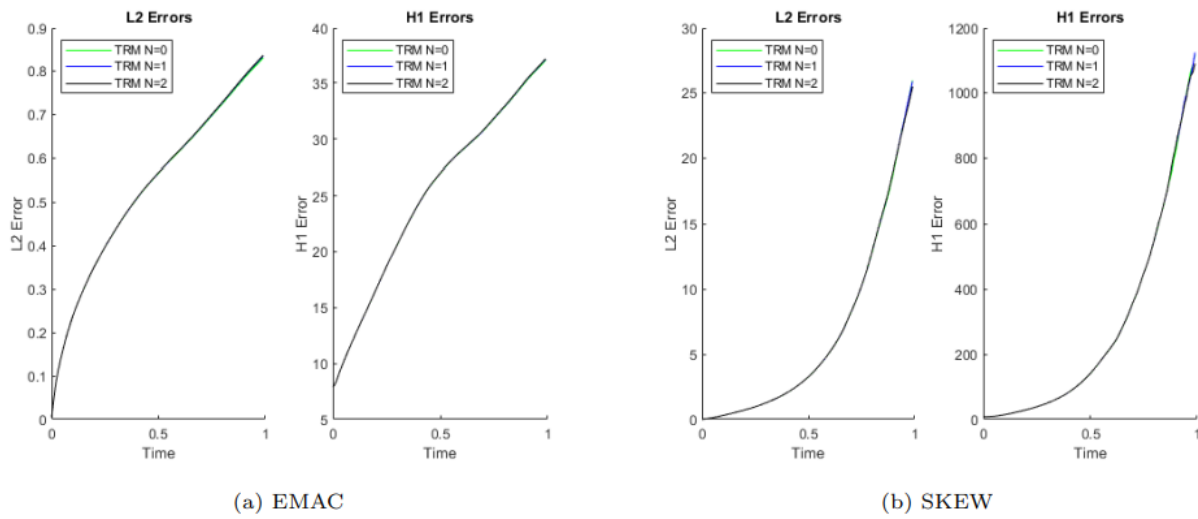
3.3.3　Lattice-Vortex Problem

In this section, we consider the Lattice Vortex problem studied in [80, 63] with small viscosity ν.

The true solution of this problem is given by

Figure 3.3: Error plots for TRM with unscaled filter width δ.

(a) EMAC

(b) SKEW
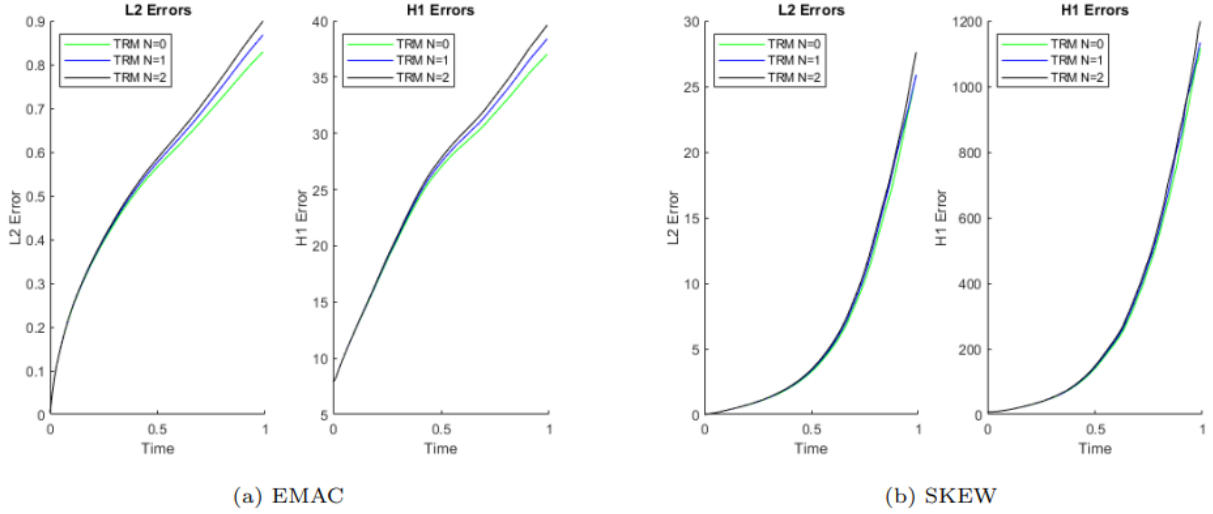
$$\boldsymbol{u} = \boldsymbol{v}e^{-8\nu\pi^2 t}, \ p = qe^{-16\pi^2 t},$$

where

$$\boldsymbol{v} = [\sin(2\pi x)\sin(2\pi y), \cos(2\pi x)\cos(2\pi y)], \ \text{and} \ q = -\frac{1}{2}(\sin^2(2\pi x) + \cos^2(2\pi y)).$$

The domain of this test problem is $\Omega = (0,1)^2$. We set the initial condition to $\boldsymbol{u}_0 = \boldsymbol{u}(0)$, viscosity $\nu = 10^{-7}$, time step $\triangle t = 0.01$ and final time $T = 10$. Note that over the simulation time, because of the small viscosity, the velocity solution will decay slightly. However, it is difficult to resolve the periodic vortices, as this problem is very susceptible to deviations from the true solution of the continuous problem, and hence many traditional schemes to resolve it end up with exponential growth of error over time [20].

A triangular mesh was used with 2,416 elements. We used Taylor Hood P2/P1 finite elements, and we strongly enforce Dirichlet boundary conditions for all schemes to be the true velocity solution along the boundaries.

Here we present the results of these numerical experiments for both the SKEW (see Figure

3.4) and EMAC (see Figure 3.5) schemes using the value $\chi = 0.5$ and a scaled filter width $\delta_N = 0.1\sqrt{N+1}$. Note that based on figure 3.4 the SKEW schemes have L^2 and H^1 errors which spike up to very high numbers over time, and the time relaxation regularization does very little to smooth out these spikes. Based on figure 3.5, we see that the TRM with EMAC greatly improves the L^2 and H^1 error over time. Moreover, we also see that energy, momentum and angular momentum do not blow up in time from $T = 0$ to $T = 10$. This is a major improvement over the SKEW scheme, which shows periodic large spikes from $T = 0$ to $T = 10$. Based on figure 3.5 we can see that the TRM with EMAC gives better results than NSE with EMAC, which validates the need for regularizations of the Navier Stokes equations.

In addition, we study how ν and N affect the numerical L^2 and H^1 errors. Fixing the mesh size $h = 0.01$ and time step-size $\Delta t = 1$e-3, so that their contribution to the numerical errors is negligible. The TRM parameters are chosen as: $\chi = 0.1$, and $\delta = 0.01$. The computed errors for various ν and N are shown in Tables 3.5-3.7.

Overall, we can see that EMAC scheme can obtain smaller L^2 and H^1 errors than SKEW scheme. In particular, L^2 error is about 1/4 of SKEW L^2 error and H^1 error is about 1/5 when $\nu = 1$e-2/2048. Moreover, the value of ν has stronger negative impact on H^1 error for SKEW scheme than EMAC scheme, meaning H^1 error increases about 5 times for SKEW scheme and only about twice for EMAC scheme on our tested ν values. In addition, we notice that L^2 and H^1 errors didn't decrease by much when we use $N = 1, 2$ comparing to $N = 0$.

Figure 3.4: Lattice Vortex Problem: SKEW 3 Step Error and Conserved Quantity plots, scaled delta, $Re = 1.0e7$, $\chi = 0.5$.
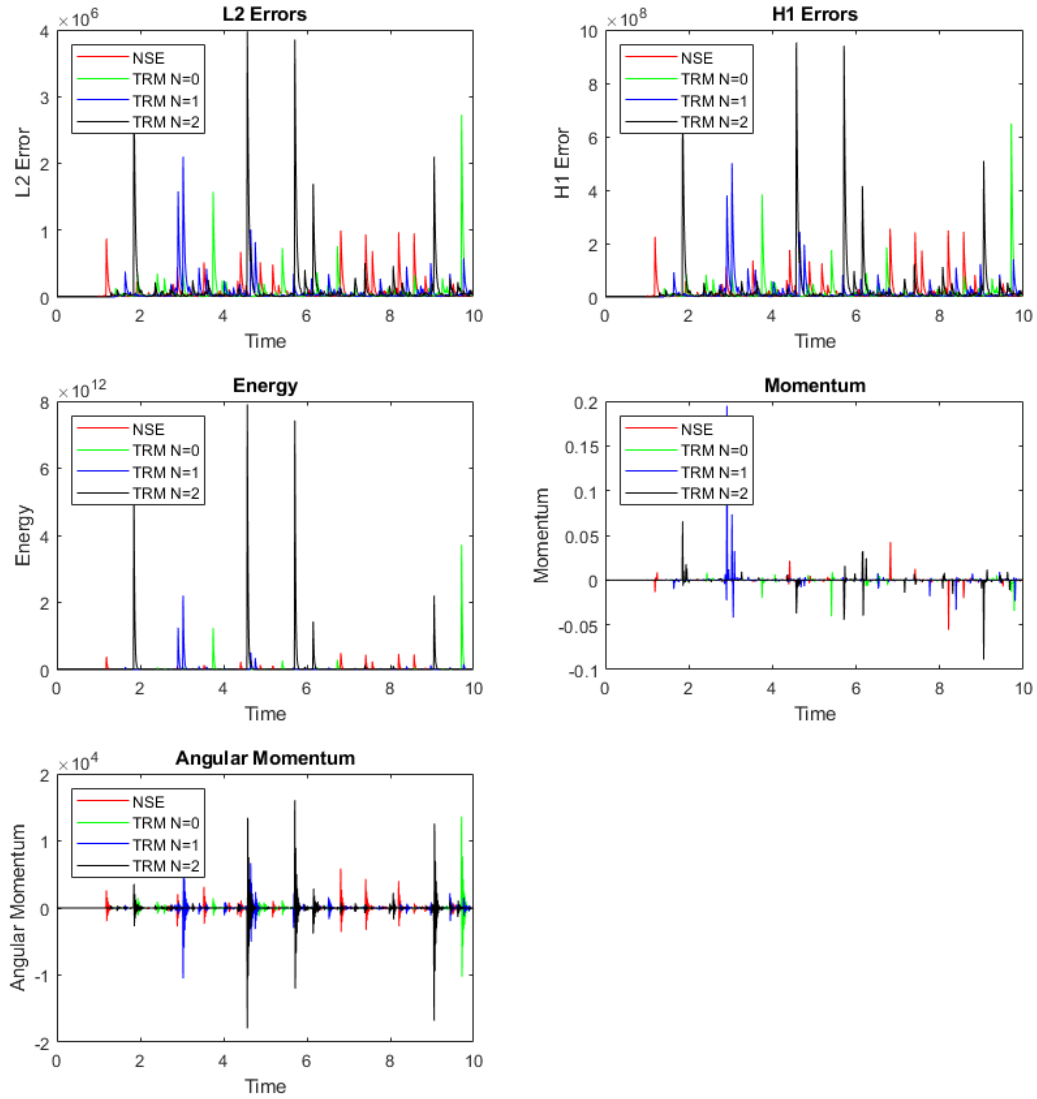
Figure 3.5: Lattice Vortex Problem: EMAC 3 Step Error and Conserved Quantity plots, scaled delta, $Re = 1.0e7$, $\chi = 0.5$.
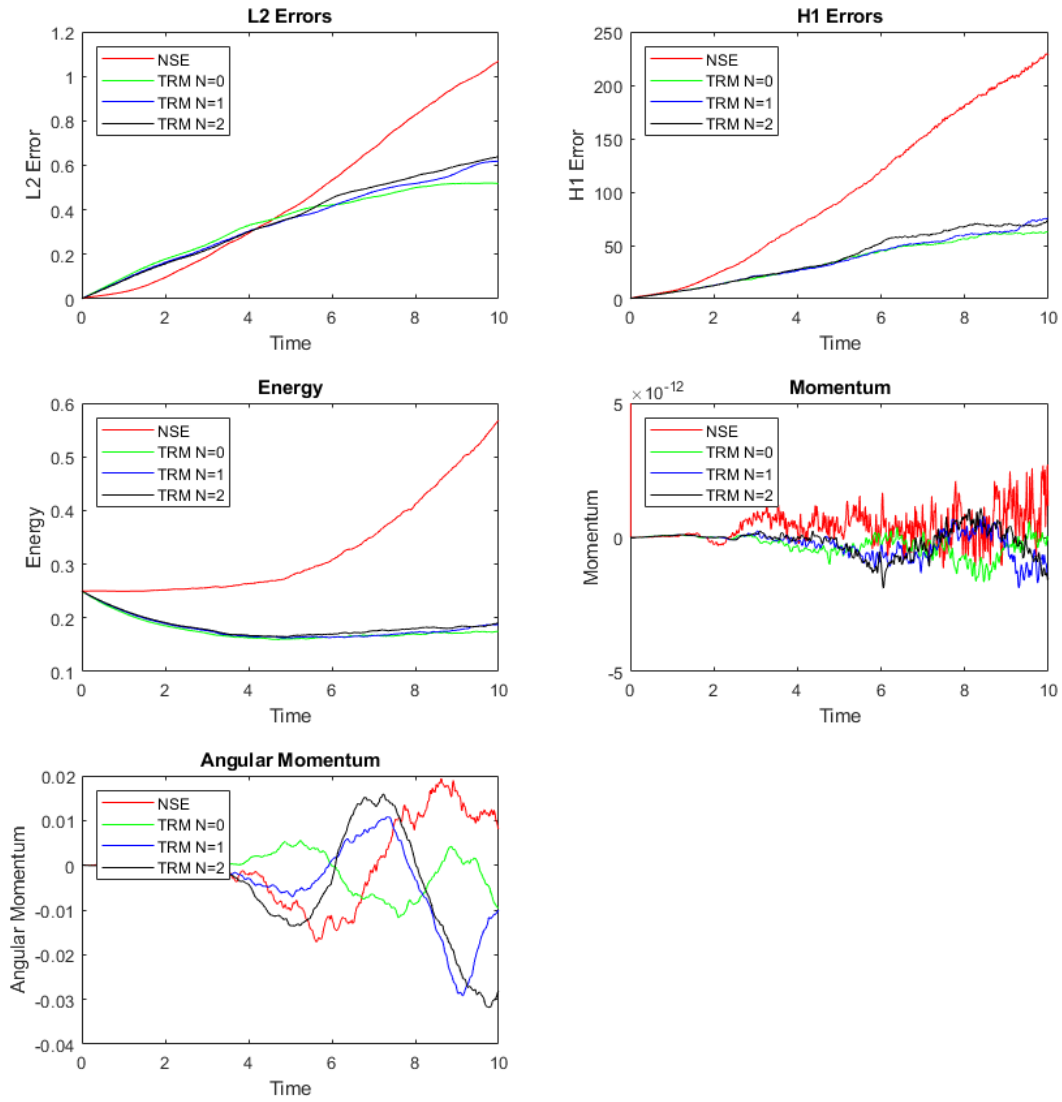
ν	SKEW		EMAC	
	L^2 error	H^1 error	L^2 error	H^1 error
1e-2/128	1.011e-03	0.5360	4.242e-04	0.2251
1e-2/256	1.654e-03	0.9903	5.419e-04	0.3080
1e-2/512	2.373e-03	1.5761	6.551e-04	0.4035
1e-2/1024	3.006e-03	2.1341	7.516e-04	0.4945
1e-2/2048	3.456e-03	2.5449	8.212e-04	0.5634

Table 3.5: Different νs case with $N = 0$, $\delta = 0.01$, and $\chi = 0.1$.

ν	SKEW		EMAC	
	L^2 error	H^1 error	L^2 error	H^1 error
1e-2/128	1.011e-03	0.5361	4.220e-04	0.2251
1e-2/256	1.653e-03	0.9906	5.402e-04	0.3081
1e-2/512	2.373e-03	1.5765	6.537e-04	0.4035
1e-2/1024	3.007e-03	2.1346	7.505e-04	0.4946
1e-2/2048	3.457e-03	2.5455	8.202e-04	0.5635

Table 3.6: Different νs case with $N = 1$, $\delta = 0.01$, and $\chi = 0.1$.

ν	SKEW		EMAC	
	L^2 error	H^1 error	L^2 error	H^1 error
1e-2/128	1.011e-03	0.5362	4.221e-04	0.2252
1e-2/256	1.654e-03	0.9907	5.403e-04	0.3081
1e-2/512	2.374e-03	1.5768	6.539e-04	0.4036
1e-2/1024	3.007e-03	2.1350	7.506e-04	0.4947
1e-2/2048	3.457e-03	2.5459	8.203e-04	0.5635

Table 3.7: Different νs case with $N = 2$, $\delta = 0.01$, and $\chi = 0.1$.

PART II

RUN-TO-RUN CONTROL OF MEMBRANE FILTRATION AT RIVER MOUNTAINS WATER TREATMENT FACILITY

CHAPTER 4

ANALYTICAL RUN TO RUN MODEL

Filtration and membrane filtration technologies are useful for the separation of suspended particles and macro-molecules from a given feed fluid. In this type of filtration, a transmembrane pressure (TMP) difference drives the feed fluid with its suspended particles through the membrane pores. Typical membranes in the micro- and ultrafiltration applications will admit particle sizes from several hundred microns large down to the nanometer scale, depending on the given application [64].

The main problem with membrane filtration is maintenance and wear and tear over time. Membrane fouling, or buildup of permeate on the membrane over time, leads to a decrease in filter efficiency. Membrane fouling has many different causes, such as pore blocking, biofilm formation, concentration polarization, and cake layering [61]. There are two main ways to limit membrane fouling. One is to provide crossflow along the membrane surface, which can reduce several different fouling effects. The other is to periodically reverse the flow direction through the membrane for the purpose of forcing out any built up permeate in the membrane pores. This process is known as backwashing or backpulsing [66]. Typically, membrane filtration is thus performed in two phases: a filtration phase, in which the membrane filters the feed fluid up to a point where filter efficiency is too diminished, at which point a backwashing phase reverses the fluid flow and cleans out the membrane pores, increasing the efficiency for the next filtration phase.

Most process control models for filtration processes use fixed values for their controls, which are determined based on experience and trial and error over time [64]. Typically, these are adjusted only to meet the required net flux for a particular cycle, which is given by

$$J_{net} = \frac{J_f \triangle t_f - J_b \triangle t_b}{\triangle t_f + \triangle t_b}.$$

The independent variables for this process are the permeate and backwashing fluid fluxes J_f and J_b and the filtration and backwashing durations $\triangle t_f$ and $\triangle t_b$. Here the subscripts f and b indicate which process these variables are associated with, either filtration or backwashing, respectively. Another variable one can consider is the intensity of the cross flow u_c applied across the membrane.

Control strategies for these filtration processes tend to be simple due to the complexity of the flows during these processes. The constant switching between filtration and backwashing makes it so the fluxes do not typically reach a steady state, especially during cross-flow implementation. Not only that, usually very limited measurement data is available in any industrial application [64].

In [65, 66], a model-based control approach was proposed to operate filtration processes at an optimal level throughout each filtration/backwashing cycle. The validity of this model was tested against a pilot-scale membrane bioreactor for wastewater treatment [64]. This model employs many different concepts such as dynamic real-time optimization, nonlinear model predictive control, and run-to-run control. Run-to-run control is the highlighting feature of this model, since it allows us to exploit the cyclic nature of the filtration/backwash process.

To further test the validity of this model, we applied this model to the filtration systems used in the River Mountains Water Treatment Facility in Henderson, NV. In Section 1, we give a brief introduction to the general concept of run-to-run control processes. In Section 2, we introduce the run-to-run control model proposed in [65, 66] and make modifications to fit the situation for the River Mountains filtration process. In Section 3, we present the implementation of our control strategy based on the run-to-run control model, and we will present the results of our tests on this control strategy based on measured plant data.

4.1   General Run-to-Run Control

Run-to-run process control was initially considered for semi-conductor manufacturing processes in which silicon wafers are created in batches [47]. Controls for this process during batch creation are the standard continuous PID-type controllers, which are control systems that continuously compute an error between a measured process variable and its desired setpoint, and then use proportional, integral, and derivative terms to correct the error in real time. The run-to-run controller is not a PID-controller, since it is active only in between batches. The purpose of the run-to-run controller is to calculate setpoints in between the batches. The PID controllers will then realize these setpoints during the next batch. The calculation of these setpoints is based on a very simple process model that is updated using data gathered from the previous batch. Run-to-run process control has been proposed for a wide variety of applications, ranging from the control of batch chromatography [40], yeast fermentation [11], and batch polymerization [1]. In a sense, run-to-run controllers update the model itself to adapt to any changes that occur in the system over countless cycles. This is especially useful for water and wastewater treatment, since over time the running of filtration/backwashing cycles affects the permeability of the filtration membrane, which in turn will affect the underlying physical parameters that govern the modeling of the resistance of these membranes.

The idea of a general run-to-run process control is illustrated in Figure 4.1. Here, u represents the control setpoints for the independent variables that are computed by the run-to-run controller, v represents the outputs from the base controller, y represents the measurements taken from the process, p represents the parameters and initial states of the process, and ϕ is the objective function for the control problem. A thorough literature review on run-to-run control processes is given in [67].

A critical component of the process model proposed within is the dual control problem [8]. For the parameter estimation, it is not guaranteed that any one cycle of measurements will be sufficient
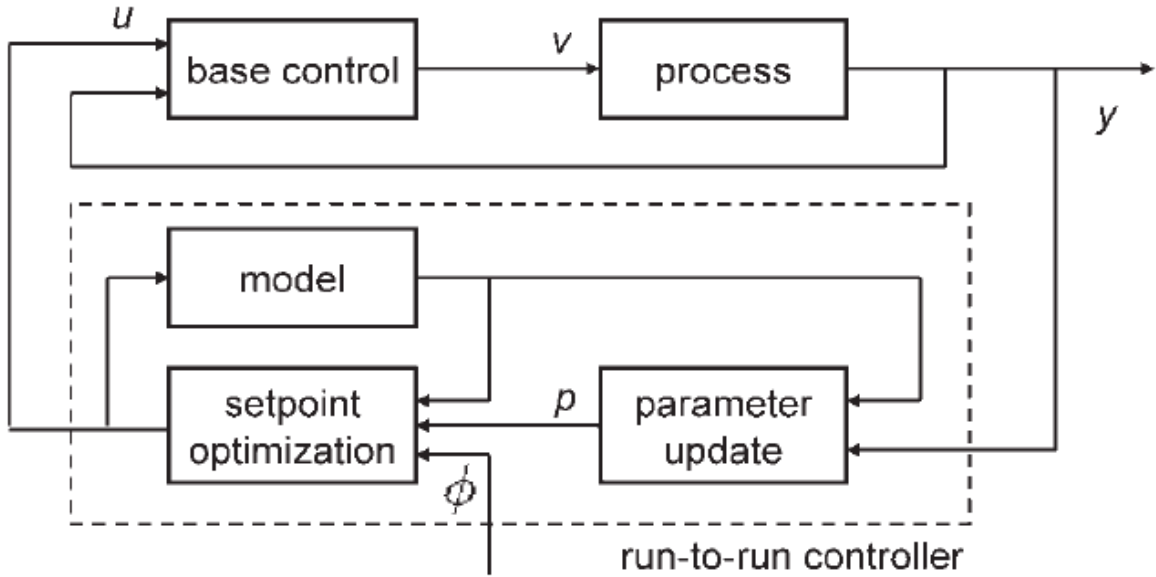
Figure 4.1: General Run-to-Run Process Control [66]

enough to accurately estimate all of the unknown parameters and initial values that the run-to-run

controller will need. An illustration of this problem is a simple model given by

$$y(t) = \alpha u(t)^{\beta}, t_1 \leq t \leq t_2.$$

Suppose $u(t)$ is constant over the entire estimation period, possibly several cycles long. Then

$y(t)$ is also constant, and hence α and β are not simultaneously identifiable. For example, if $y(t) = 2$

and $u(t) = 3$, then there are infinitely many solutions for α and β.

To address this problem, the controller proposed will exploit relationships from stochastics and

parameter estimation theory [51]. This involves splitting the list of estimation parameters into two

groups: one set $p_s = (p_s^1, p_s^2, ..., p_s^{n_s})$, which will be estimated over the data of one filtration cycle,

and another set $p_l = (p_l^1, p_l^2, ..., p_l^{n_l})$, which are not simultaneously identifiable over one cycle. The

p_l parameters will instead be estimated on appropriately chosen longer time horizons spanning N

filtration cycles, where N is appropriately chosen to give enough data to sufficiently identify all the

p_l parameters. In the case of this model, we will have two p_s parameters and only one p_l parameter to estimate.

The one parameter estimated over long periods will also be evaluated for its quality. The Hessian $\boldsymbol{H} = \dfrac{\partial^2 \phi}{\partial p^2}(p_l)$ at the solution describes a confidence interval for this parameter given by

$$\boldsymbol{H} * \delta p^2 \leq 2\epsilon,$$

where ϵ is a parameter that defines the size of the indifference region, or the largest deviation which is considered insignificant for this application, and δp is the deviation of our estimate from the optimal parameter value $\hat{p}$. Solving for δp gives us

$$\delta p = \sqrt{\frac{2\epsilon}{\boldsymbol{H}}}.$$

If the estimate for δp is below a specified tolerance, then the parameter estimate is accepted as the new value. If not, the old parameter value is adopted for the next period of cycles.

4.2   Control Model

This model is based on the one proposed in [65, 66]; however, several modifications were made to properly fit this model to the filtration systems at River Mountains. One primary modification is that the original model accounts for the aeration crossflow intensity during a filtration cycle that may be present during operation. However, the River Mountains facility only uses crossflow during backwashing and not during filtration. Thus, the crossflow intensity variable has been removed from the membrane resistance model for filtration, and will now only be used in cost optimization to determine the power consumed by the air scour blowers during a backwash cycle.

The original model also solved for resistance during backwashing, however there is not sufficient data available from measurements to fully realize this model. For this reason, the resistance during

backwashing is approximated by the setpoint for the main backwash controller valve during high wash, which is the dominant portion of the backwashing cycle where the brunt of the effluent backwash flow is run through the membrane at very high flow rates, typically around 55 million gallons per day (MGD). This is a reasonable choice since the backwashing phase is negligibly short compared to the filtration phase, hence its contribution to the overall power cost is negligible compared to filtration. Moreover, the high wash phase of backwashing is the primary contributor to the net backwash flux, and thus the average flux is reasonably close to its high wash setpoint.

4.2.1   Filtration ODE and Least Squares Estimation

In this section, we shall outline the model process. First, we present the ODE for the filtration membrane resistance. This ODE is based on [65, 66], but is modified to account for the lack of crossflow aeration in the River Mountains filtration process. The ODE is given by

$$\frac{dR_f}{dt} = mJ_f^\alpha, \quad t \in [t_{f,0}, t_{f,e}], \tag{4.1}$$

$$R_f(t_{f,0}) = R_f^0, \quad t_{f,e} = t_{f,0} + \triangle t_f. \tag{4.2}$$

Here, R_f is the filtration membrane resistance as a function of time, J_f is the filtration fluid flux, and m and α are modeling parameters to be estimated. This ODE describes the cake layer formation resistance, which is the largest contributing factor to the overall resistance on the timescale of a typical filtration cycle. This resistance depends primarily on the fluid flux and any aeration crossflow that is present. The original differential equation from [66] has an extra term that accounts for this contribution from the crossflow u_c,

$$\frac{dR_f}{dt} = mJ_f^\alpha u_c^\beta. \tag{4.3}$$

Since no crossflow is realized during filtration, the variable u_c and by extension the modeling parameter β have been removed from the filtration ODE.

Based on historical plant data, we correctly assume that the filtration flux J_f is relatively constant during operation, and we will see a linear increase in membrane resistance over time. Moreover, we can see that the parameter $\alpha > 0$, since it is observed that an increase in fluid flux during a filtration cycle leads to a corresponding increase in membrane resistance.

The resistance model above forms the basis for our estimation of the membrane resistance using the TMP data from the previous filtration phase. This estimation is achieved by the following least squares optimization problem

$$\min_{R_f^0, m} \sum_{l=1}^{n_{f,l}} \frac{1}{2} (\triangle p_{\tilde{f},l} - \triangle p_{f,l})^2 \tag{4.4}$$

$$\text{s.t. } \triangle p_{f,l} = \overline{J_f} \eta R_{f,l}, \tag{4.5}$$

$$R_{f,l} = R_f^0 + m \overline{J_f}^\alpha t_l, \tag{4.6}$$

$$t_l \in [t_{f,0}, t_{f,e}], l \in \{1, 2, ..., n_{f,l}\}. \tag{4.7}$$

Here, $\overline{J_f}$ represents the mean filtration flux from the previous cycle. $\triangle p_{\tilde{f},l}$ represent the discrete TMP measurements during the previous filtration cycle, while $\triangle p_{f,l}$ represents the approximation of the TMP using the analytical solution to equation (4.3) and Darcy's Law, which is an equation for fluid flow through a porous media that relates the fluid pressure on the media (TMP), the fluid flux, and the porous membrane resistance by means of equation (4.5). We note that in the above optimization problem, there are three parameters that need to be estimated, namely R_f^0, m, and α. However, we cannot estimate all three parameters on the same cycle due to the dual-control problem. We employ a run-to-run control model in accordance with [65] to overcome such difficulties. The parameters R_f^0 and m are estimated through the least squares problem above for

each cycle, while α will be estimated over multiple cycles.

A similar least squares estimation and ODE problem was proposed for the backwashing resistance in [65] as well, however it was realized that plant data is often insufficient in fully realizing such a model due to high measurement noise and effects from the ramp up time of the backwashing pumps. The River Mountains facility is no exception to this rule, and hence the backwashing fluid flux is modeled by the constant setpoint for the high wash phase of the backwashing, and the mean TMP for backwashing is also estimated based on available plant data for previous cycles. Using these, we can then calculate the approximate backwash resistance via Darcy's Law (4.5). This estimation is reasonable due to the relatively small run time of the backwashing phase compared to the much longer filtration phase, hence the discrepancies of our backwashing model to what is actually observed is negligible overall.

4.2.2 Cost Optimization Problem

Once filtration and backwashing have been fully estimated for one cycle, a cost optimization is performed to determine setpoints that minimize the overall cost of operation. The operating cost can be broken down into three separate sources:

- The cost of energy provided to the TMP,

- The cost of running the air scours for cross-flow during backwashing, and

- The cost of membrane replacement over time.

The electrical energy needed to supply the TMP during the interval $[t_{f,0}, t]$ is given by the following differential equation,

$$\frac{dE_p}{dt} = \frac{|\triangle p_j J_j A|}{\eta_p (\triangle t_f + \triangle t_b)}, \quad E_p(t_{f,0}) = 0, \quad t \in [t_{f,0}, t_{f,e}] \cup [t_{b,0}, t_{b,e}].$$

In the above, the index $j = f$ when $t \in [t_{f,0}, t_{f,e}]$, and $j = b$ when $t \in [t_{b,0}, t_{b,e}]$. We can solve the differential equation above analytically by splitting the ODE into two separate ODEs for the filtration period and backwashing period. For the filtration period, we use Darcy's Law and the analytical solution of the filtration ODE to substitute in for $\triangle p_j$, while for the backwashing period, we simply replace $\triangle p_b$ and J_b by their mean values during the backwashing. This approximation must be made due to the lack of reliable TMP data available during backwashing cycles. This gives us the following piecewise ODE,

$$\frac{dE_p}{dt} = \begin{cases} \dfrac{J_f^2 \eta (R_f^0 + m J_f^\alpha (t - t_{f,0})) A}{\eta_p (\triangle t_f + \triangle t_b)}, \text{if } t \in [t_{f,0}, t_{f,e}] \\[4mm] -\dfrac{\overline{\triangle p_b}\, \overline{J_b} A}{\eta_p (\triangle t_f + \triangle t_b)}, \text{if } t \in [t_{b,0}, t_{b,e}] \end{cases} .$$

Solving these ODEs together with the additional continuity condition that $E_p(t_{f,e}) = E_p(t_{b,0})$, we can solve for $E_p(t)$. Then plugging in $t = t_{b,e}$ will give us the total energy required to supply the TMP for the entire filtration/backwash cycle,

$$E_p(t_{b,e}) = \frac{J_f^2 \eta A \left(R_f^0 \triangle t_f + \frac{1}{2} m J_f^\alpha \triangle t_f^2 + \overline{\triangle p_b}\, \overline{J_b} \triangle t_b \right)}{\eta_p (\triangle t_f + \triangle t_b)}. \tag{4.8}$$

Next, the the power required to run the air scours during the backwashing cycle is determined by the following formula that is based on the polytropic compression of air,

$$E_c = \frac{QRT \triangle t_b}{v_a \eta_A (\gamma - 1)} \left[(1 + p_a)^{\frac{\gamma - 1}{\gamma}} - 1 \right]. \tag{4.9}$$

In the above, $Q = 16,500 \frac{m^3}{hr}$ is the air flow rate for the air scours, T is the ambient room temperature, R is the ideal gas law constant, v_a is the molar volume of air, $\gamma_a = 1.4$ is the polytropic coefficient, and p_a is the difference in pressure across the compressor, in bars. This equation is slightly different from preceding papers, hence we will derive it below to show that this

formula is mathematically sound, and to offer a correction to previous formulations of this power cost found in [66, 65].

Theorem 4.2.1. The energy supplied to the air scours during one backwash cycle is given by equation (4.9).

Proof. First, we derive the work done by the air compressor in the air scours. Since the compression process is assume to be polytropic, the pressure P and volume V of the air being compressed satisfies $PV^{\gamma} = C$, where C is some constant value. Moreover, we assume the gas is ideal, and thus we have $PV = nRT$, where n is the number of moles of gas, T is the temperature of the gas, and R is the ideal gas constant. Now we compute the work done to compress a volume of gas in atmosphere with pressure, temperature, and volume P_1, T_1, V_1 to a state with pressure, temperature, and volume P_2, T_2, V_2,

$$W = -\int_{V_1}^{V_2} P dV = -P_1 V_1^{\gamma} \int_{V_1}^{V_2} V^{-\gamma} dV = P_1 V_1^{\gamma} \left(\frac{V_2^{1-\gamma} - V_1^{1-\gamma}}{\gamma - 1} \right) = \frac{P_2 V_2 - P_1 V_1}{\gamma - 1}.$$

Now we factor out $P_1 V_1$, and substitute temperature in using the ideal gas law to obtain

$$W = \frac{P_1 V_1}{\gamma - 1} \left(\frac{P_2 V_2}{P_1 V_1} - 1 \right) = \frac{P_1 V_1}{\gamma - 1} \left(\frac{T_2}{T_1} - 1 \right). \tag{4.10}$$

Combining the polytropic assumption and the ideal gas assumption, we can rewrite $\frac{T_2}{T_1}$ in terms of P_2 and P_1 as follows,

$$\frac{T_2}{T_1} = \left(\frac{P_2}{P_1} \right)^{\frac{\gamma - 1}{\gamma}}. \tag{4.11}$$

Using equation (4.11), we can rewrite (4.10) as

$$W = \frac{P_1 V_1}{\gamma - 1} \left[\left(\frac{P_1}{P_2} \right)^{\frac{\gamma}{\gamma - 1}} - 1 \right] = \frac{P_1 V_1}{\gamma - 1} \left[\left(1 + \frac{P_2 - P_1}{P_1} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right] = \frac{P_1 V_1}{\gamma - 1} \left((1 + p_a)^{\frac{\gamma - 1}{\gamma}} - 1 \right)$$

$$(4.12)$$

where $p_a = \dfrac{P_2 - P_1}{P_1}$ is the pressure difference across the compressor, normalized to the pressure P_1, which is standard atmosphere pressure, or 1 bar. From here, by ideal gas law, we have $P_1 V_1 = \dfrac{mRT}{v_a}$, where m is the mass of the volume of gas being compressed, and v_a is the molar volume of that gas. Substituting this into equation (4.12), then taking the time derivative of both sides and letting $Q = \dfrac{dm}{dt}$, we obtain the following for the power supplied by the air compressor:

$$\frac{dW}{dt} = \frac{QRT}{v_a(\gamma - 1)} \left[(1 + p_a)^{\frac{\gamma - 1}{\gamma}} - 1 \right]$$

Integrating this over the backwashing time interval, and adding the efficiency factor η_p on the denominator of the expression gives us our desired result. □

The last contribution to cost of operation considered is the cost of membrane replacement. This cost is not as straightforward as the other two to calculate, as there is no rigorous model describing this cost in terms of our given independent variables. Different models of membrane replacement cost must be used for different situations. The following cost model was proposed in [66] for microfiltration membranes in wastewater applications. They observed that longer filtration cycles tend to produce more irreversible resistance due to excessive biofilm formation during the cycle. Since an increase in TMP is generally associated with an increase in resistance, the following cost model penalizes TMP increases, and the penalty rises exponentially with time, so that

$$E_r = \xi_1 \left(\triangle p(t_{f,e}) - \triangle p(t_{f,0}) \right) e^{\triangle t_f / \xi_2}.$$

Here, ξ_1 and ξ_2 are tuning parameters. ξ_1 will linearly scale the cost of membrane fouling, and

ξ_2 is a time constant.

From here, we can summarize our total cost function as the sum of the previous three costs,

$$
\begin{aligned}
\phi(t_{b,e}) &= E_p(t_{b,e}) + E_c + E_r \\
&= \frac{J_f^2 \eta A \left(R_f^0 \triangle t_f + \frac{1}{2} m J_f^\alpha \triangle t_f^2 + \overline{\triangle p_b}\, \overline{J_b} \triangle t_b \right)}{\eta_p(\triangle t_f + \triangle t_b)} + \frac{QRT \triangle t_b}{v_a \eta_A (\gamma - 1)} \left[(1 + p_a)^{\frac{\gamma - 1}{\gamma}} - 1 \right] \\
&\quad + \xi_1 \left(\triangle p(t_{f,e}) - \triangle p(t_{f,0}) \right) e^{\triangle t_f / \xi_2}.
\end{aligned}
$$

Using this cost function, we run the following cost optimization problem at the end of every filtration/backwashing cycle,

$$
\min_{J_f, \triangle t_f} \phi \tag{4.13}
$$

$$
\text{s.t. } \triangle p_f = J_f \eta R_f \tag{4.14}
$$

$$
R_f = R_f^0 + m J_f^\alpha t \tag{4.15}
$$

$$
J_{net} = \frac{J_f \triangle t_f - J_b \triangle t_b}{\triangle t_f + \triangle t_b} \tag{4.16}
$$

$$
J_f \leq J_b \tag{4.17}
$$

$$
\triangle p_{min} \leq \triangle p \leq \triangle p_{max} \tag{4.18}
$$

$$
0 \leq J_f \leq J_f^{max}, 0 \leq \triangle t_f \leq \triangle t_f^{max}. \tag{4.19}
$$

Equation (4.16) is a constraint that captures the net flux for an overall filtration/backwash cycle, while equation (4.17) forces the backwash flux to be stronger than the filtration flux.

4.3   Implementation of the Run-to-Run Model

To compute solutions of this optimization control problem, we will utilize all data collected by the Las Vegas Valley Water District SCADA database. This database collects real-time data for

every working part of the water treatment process at River Mountains. For the filtration systems at River Mountains, SCADA provides access to many different real time measurements for filter operation, such as particle counts, headloss (related to TMP), turbidity, filter run time, flow rates for filtration and backwash flows, durations of previous filtration and backwashing cycles, air flow rates for air scours, and much more.

We created a script using MATLAB to download real-time plant data from the SCADA database. The script was designed to test how well the model could compute setpoints for a series of N filtration and backwash cycles. The model reads in data for N consecutive filtration/backwashing cycles. It then loops through each cycle and does the following:

1. The unknown parameters R_0^f and m are computed for the filtration cycle using the least squares optimization problem given in equations (4.4)-(4.7), which is implemented using `lsqnonlin` in MATLAB.

2. Using these parameters, the script computes the solution to the filtration resistance ODE in equation (4.3) via the function `ode45` in MATLAB.

3. The optimal setpoints are computed for the next cycle via the cost optimization problem given by equations (4.13)-(4.19). The algorithms used to implement the cost function will be discussed later in this section.

Run-to-run control is implemented in this script as well. Since our modified model drops the β paramter in the filtration ODE given in equation (4.3) due to the fact that no crossflow aeration is present during filtration, it should be verified that the so called dual control problem is still present with only three parameters that need to be simultaneously identified for each cycle. In order to compute the filtration resistance from equation (4.3), we need to approximate all three parameters that appear in this equation R_0^f, m, and α.

We have two approaches that can be taken, either estimating only two of these on small horizons while estimating the third over longer time horizons, or estimating all three at once. Both approaches were tested and the results can be seen in Figure 4.2. Each estimation is run four times with four different initial guesses for the parameters. The left column presents the results of the two parameter estimation runs, and the right column presents results for the three parameter estimation runs. It is clear that in the two parameter estimations, all the initial guesses still end up at the same optimal values for R_f^0 and m and result in the same linear fit to the given TMP data for that cycle. However, for the three parameter estimations, the initial guess for our parameters heavily influences what optimal values we get for R_f^0, m and α, as the solver will often not have to look far from the initial value to find a local minimum. This shows that the dual control problem is very much present under the three parameter estimation, and hence the run-to-run control algorithm is a necessity when estimating these parameters for calibration of the full model.

To implement run-to-run control, we can select a period of a discrete number of N cycles to use for the long time α parameter estimation. After the cost optimization has been performed for all the cycles in that period, the script will solve the least squares optimization problem given by (4.4)-(4.7), but instead of optimizing for R_f^0 and m over one cycle, we optimize for α alone over all the filtration cycles in the given period, using the R_f^0 and m values that were computed for each cycle respectively. Since α is estimated over long time horizons, it is not known for the first period of filtration/backwashing cycles, and hence an initial estimation or guess for α must be supplied for these cycles to prime the algorithm.

The quantities that are readily available from the SCADA in real time are the filtration effluent volume total, effluent flow setpoint, current effluent flow, headloss, turbidity, and particle counts. One downfall of the run-to-run model for this particular application is the incompatibility of the model to turbidity and particle count data. The run-to-run model is primarily reliant on the

Figure 4.2: Comparison of 2 parameter estimation vs. 3 parameter estimation
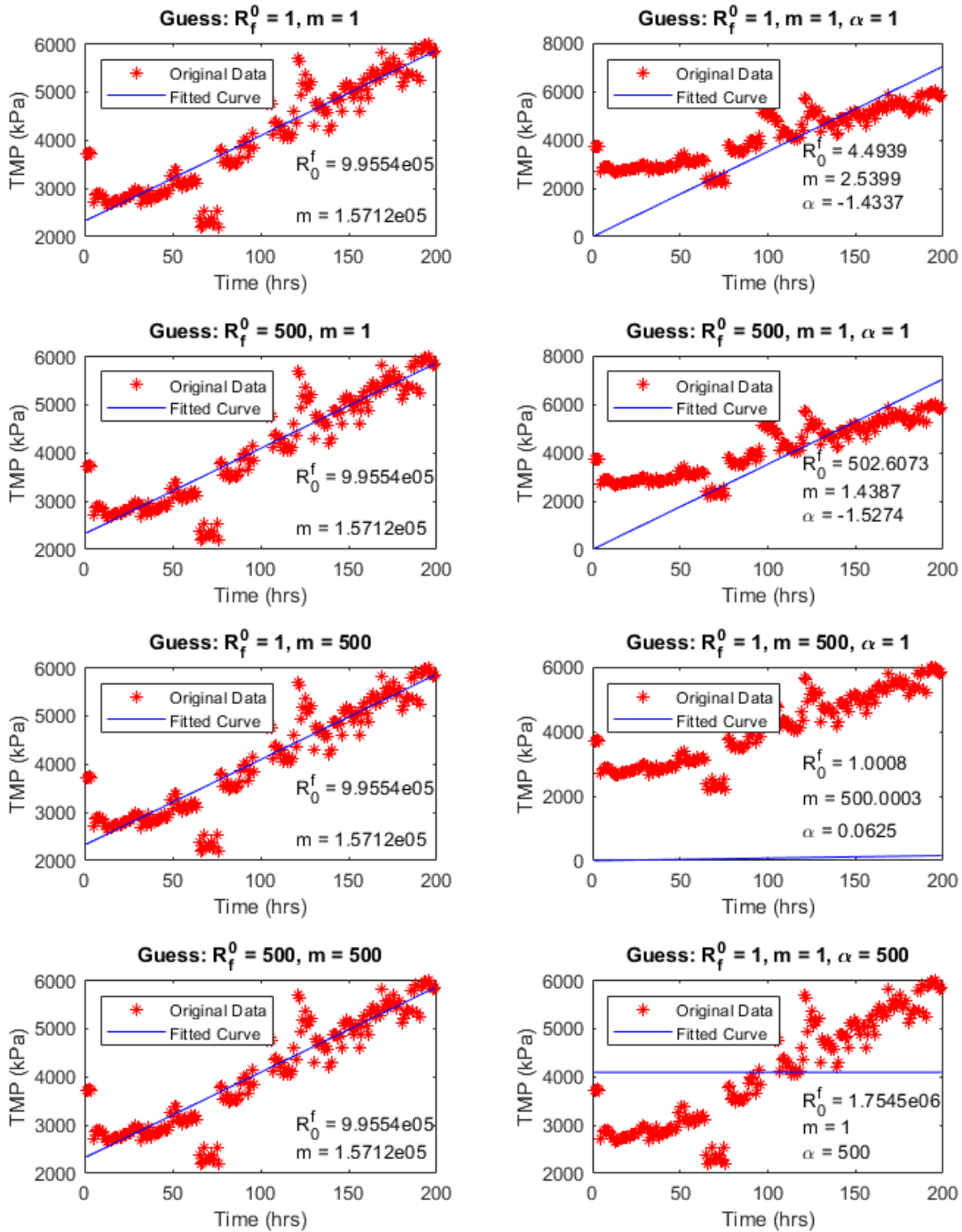
effluent flows and the TMP (which is easily related to headloss), but also on viscosity, which is not a quantity that is tracked by SCADA in real time. It would be beneficial if one could relate the viscosity of the water to the turbidity and particle counts, however as far as I am aware, no such relation between these concepts has been established throughout the body of research on this subject. Many different papers such as [78] have done analysis to relate viscosity, turbidty, and particle counts to other properties of a given fluid, such as concentrations of suspended solutions. In all of these works however, no such relation has been established between the viscosity of a given fluid and its turbidity and particle counts. In fact, such a relationship must necessarily be specific to a given type of fluid, as one can find many examples of fluids with no turbidity but vastly different viscosites, as well as examples of fluids with similar viscosities and vastly different turbidities. It would be interesting to see if such a relationship could be developed and used to test the contribution of viscosity to the overall model, but for now, we set the viscosity to the normal viscosity of water in room temperature.

4.3.1 Initial Low Viscosity Simulations

To implement this model, a selection of data from 6 total filtration/backwash cycles spanning 3 months of plant operation were taken from SCADA as input for the cost optimization script. For the cost optimization problem given by equations (4.13)-(4.19), we first attempted to optimize the cost using `fmincon`. One thing was noted right away after initial runnings, the power costs computed from the setpoints were far too large to represent actual costs of running a single filtration cycle. Upon further investigation, we discovered that the power for running the air scours was the reason. The power cost associated with this was several orders of magnitude larger than the other two cost sources. For this reason, we decided to remove the air scour cost from the cost formulation itself, which is a reasonable decision for this application considering that the air scours for backwashing have historically always been used for the same amount of time during each backwashing cycle,

hence the cost of running the air scours for backwashing is virtually constant for each cycle.

After removing the air scour term from the cost function, we implemented `fmincon` again, and this time costs were brought down to a more reasonable level. For each cycle, the setpoints for the filtration fluid flux was reasonably close to observed data, as effluent flows during each cycle ranged anywhere from 2×10^{-3} to $3 \times 10^{-3} \frac{m^3/s}{m^2}$, however the setpoints for filtration time for each cycle have severely undershot what is observed. Typical filtration times average around 200 hours, yet we predict filtration times that are no more than 10 hours.

Since these setpoints were off, we wondered if the solver was unintentionally finding some local minimum for the setpoints instead of the global minimum, so we decided to try global optimization tools as well to solve the cost problem. The two algorithms we tried were `patternsearch` and `ga`.

`patternsearch` works by finding a sequence of points $x_0, x_1, x_2, ...$ that converge to the optimal solution. The algorithm starts with an initial guess x_0 and a mesh size m (default size is one). It then generates, or polls, a mesh of points, each of distance m away from x_0 in the direction of each component for a total of $2N$ mesh points, N being the number of components in x_0. The algorithm plugs in each of these mesh points to the objective function. If none of these points give a lower value for the objective function than x_0, this is an unsuccessful poll and the algorithm picks $x_1 = x_0$, and the mesh size is multiplied by the mesh expansion factor (default value of two) for the next set of mesh points. If any of the points give a lower value for the objective function, this is a successful poll and the algorithm picks the one that gives the lowest value and calls that x_1, and the mesh size is multiplied by the mesh contraction factor (default value of 0.5) for the next set of mesh points. Repeating this process, our sequence of points is generated until one of the following happens,

1. The mesh size goes smaller than a specified mesh tolerance,

2. The number of iterations passes a specified maximum iteration number,

3. The total number of objective function evaluations passes a specified maximum,

4. The time the algorithm runs exceeds a specified maximum value,

5. After a successful poll, the distance between the last two points in the sequence is below a specified step tolerance,

6. After a successful poll, the change in the objective function between the last two points in the sequence is below a specified function tolerance and the mesh size is also less than the step tolerance.

For more on the specifics of `patternsearch`, the reader is referred to [35].

`ga` works by generating a random population of points from the parameter space. It then begins the process of creating a sequence of new populations. At each step, the points in the current generation are used to create the next population. To do so, the algorithm performs the following steps,

1. The algorithm scores each member of the current population by computing its fitness value (objective function value), which is called the raw fitness score.

2. The raw fitness scores are scaled and converted to a more usable range of values which are called expectation values.

3. The algorithm selects members, called parents, based on their expectation.

4. Some individuals in the current population that have lower fitness are chosen as elite. These elite points are passed to the next population.

5. The algorithm produces children points from the parent points. Children are produced either by making random changes to a single parent (mutation), or by combing the vector entries of a pair of parents (crossover).

77

6. The algorithm repalces the current population with the children to form the next generation.

The algorithm will terminate under any of the following conditions,

1. After the number of generations exceeds a specific maximum.

2. After running for a specified maximum amount of computing time.

3. When the value of the fitness function for the best point in the current population is less than or equal to a specified fitness limit.

4. When the average relative change in the fitness function over a specified number of "stalled" generations is less than a specified function tolerance.

5. When there is no improvement in the objective function during an interval of time in seconds equal to a specified "stall" time.

6. Until the average relative change in the fitness function value over a number of "stalled" generations is less than the function tolerance.

For more detailed information on the `ga` algorithm, the reader is referred to [36].

A breakdown of the results for each setpoint and the power cost for each cycle across all three algorithms are given in Tables 4.1-4.3. The results from `patternsearch` are nearly identical to `fmincon`, while the results from `ga` were slightly smaller than the other two for both setpoints. As a result, `ga` also predict smaller costs for each cycle compared to the other two algorithms. But for all three algorithms, the filtration times were still way too small compared to what is utilized at River Mountains. One possible reason for this is the low value of viscosity chosen, which may not be representative of the true viscosity of flows through the filtration systems.

Table 4.1: Setpoints for fluid flux for all 3 algorithms with low viscosity

Viscosity	Cycle	Algorithms		
8.9e-04		fmincon	patternsearch	ga
	1	3.17e-03	3.15e-03	2.29e-03
	2	3.34e-03	3.32e-03	2.88e-03
	3	3.25e-03	3.24e-03	2.59e-03
	4	3.36e-03	3.35e-03	2.71e-03
	5	3.44e-03	3.44e-03	2.97e-03
	6	3.42e-03	3.41e-03	2.77e-03

Table 4.2: Setpoints for filtration time for all 3 algorithms with low viscosity

Viscosity	Cycle	Algorithms		
8.9e-04		fmincon	patternsearch	ga
	1	8.601	8.590	7.326
	2	6.372	6.446	5.887
	3	7.079	7.086	6.505
	4	5.710	5.719	4.987
	5	5.061	5.065	3.580
	6	4.687	4.684	4.058

Table 4.3: Power cost for all 3 algorithms with low viscosity

Viscosity	Cycle	Algorithms		
8.9e-04		fmincon	patternsearch	ga
	1	9.41e03	9.39e03	8.44e03
	2	6.90e03	6.90e03	6.66e03
	3	8.14e03	8.13e03	7.67e03
	4	1.23e04	1.23e04	1.18e04
	5	1.48e04	1.48e04	1.43e04
	6	1.87e04	1.87e04	1.82e04

4.3.2 Varied Viscosity Simulations

For the simulations in the previous section, we kept viscosity at the value for clear water at room temperature. However, in practice, viscosities of untreated effluent flow can be at higher values, and thus we wanted to see how these simulations ran for different values of effluent viscosity. In the following experiments, we ran the run-to-run control problem again for values of viscosity ranging from $1.0e - 03\frac{kg}{m \cdot s}$ to $5.0e - 03\frac{kg}{m \cdot s}$, however we will only present the results for viscosities of $2.0e - 03\frac{kg}{m \cdot s}$ and $5.0e - 03\frac{kg}{m \cdot s}$. For the algorithms fmincon and patternsearch, increasing

the viscosity changed nothing about the results, we obtained the exact same setpoints and power

cost as for the run with the normal viscosity of water. However for `ga`, the values we obtained for

the setpoints differed as viscosity changed, sometimes resulting in large differences in predictions,

such as the much larger than usual filtration time prediction of 35.469 hours for the first cycle.

For the most part, `ga` setpoint predictions for higher viscosities were not much different from the

predictions using the normal viscosity of water, and in fact the cost predictions from `ga` remained

nearly identical for all simulations, minus the one outlier cycle mentioned above.

This shows that viscosity doesn't have much of an effect on the overall predictions of this model.

In fact, if we ran this model with only `fmincon` and `patternsearch`, we would suspect viscosity

had no contribution whatsoever. This is an unexpected outcome, since we know normal filtration

operation at River Mountains definitely does depend on the turbidity and particle counts of the

effluent flow, and both of these are known to increase as the viscosity of water increases. From

this fact, and the fact that setpoint predictions for filtration times in all these runnings were too

small, we determined that the proposed filtration/backwashing model is insufficient to describe the

dynamics inside the filtration systems accurately.

Table 4.4: Setpoints for fluid flux for all 3 algorithms with varied viscosities

Viscosity	Cycle	Algorithms		
2.0e-03		`fmincon`	`patternsearch`	`ga`
	1	3.17e-03	3.15e-03	3.27e-03
	2	3.34e-03	3.32e-03	3.22e-03
	3	3.25e-03	3.24e-03	2.61e-03
	4	3.36e-03	3.35e-03	2.92e-03
	5	3.44e-03	3.44e-03	2.98e-03
	6	3.42e-03	3.41e-03	2.83e-03
5.0e-03		`fmincon`	`patternsearch`	`ga`
	1	3.17e-03	3.15e-03	2.29e-03
	2	3.34e-03	3.32e-03	3.90e-03
	3	3.25e-03	3.24e-03	2.62e-03
	4	3.36e-03	3.35e-03	2.69e-03
	5	3.44e-03	3.44e-03	2.86e-03
	6	3.42e-03	3.41e-03	2.81e-03

Table 4.5: Setpoints for filtration time for all 3 algorithms with varied viscosities

Viscosity	Cycle	Algorithms		
2.0e-03		`fmincon`	`patternsearch`	`ga`
	1	8.601	8.590	35.469
	2	6.372	6.446	3.330
	3	7.079	7.086	5.877
	4	5.710	5.719	4.470
	5	5.061	5.065	3.716
	6	4.687	4.684	3.921
5.0e-03		`fmincon`	`patternsearch`	`ga`
	1	8.601	8.590	7.298
	2	6.372	6.446	2.588
	3	7.079	7.086	6.404
	4	5.710	5.719	5.204
	5	5.061	5.065	4.449
	6	4.687	4.684	3.696

Table 4.6: Power cost for all 3 algorithms with varied viscosities

Viscosity	Cycle	Algorithms		
2.0e-03		`fmincon`	`patternsearch`	`ga`
	1	9.41e03	9.39e03	1.09e04
	2	6.90e03	6.90e03	6.69e03
	3	8.14e03	8.13e03	7.65e03
	4	1.23e04	1.23e04	1.19e04
	5	1.48e04	1.48e04	1.43e04
	6	1.87e04	1.87e04	1.82e04
5.0e-03		`fmincon`	`patternsearch`	`ga`
	1	9.41e03	9.39e03	8.45e03
	2	6.90e03	6.90e03	6.97e03
	3	8.14e03	8.13e03	7.68e03
	4	1.23e04	1.23e04	1.18e04
	5	1.48e04	1.48e04	1.43e04
	6	1.87e04	1.87e04	1.82e04

4.4 DRAM Algorithm for Run-to-Run Control

In addition to the above experiments, we also ran a stochastic algorithm based on Markov Chain Monte Carlo methods known as the Delayed Rejection Adaptive Metropolis (DRAM) algorithm. The details and outline of this algorithm are found in [19, 79]. This algorithm is not deterministic like the algorithms used in the previous section, but rather allows for us to take our setpoints of

operation to be random parameter variables Q with realizations q and with known prior probability densities $\pi_0(q)$. Then, given a measurement v_{obs} of the random cost function depending on our two random variables for fluid flux and filtration time, we wish to find the posterior probability densities $\pi(q|v_{obs})$. This is essentially a formulation of the Bayes' Theorem of Inverse Problems found in [70], which we present here.

Definition 4.4.1. Given p random parameter variables Q with known prior probability densities $\pi_0(q)$, which can be non-informative, let v_{obs} be a realization of the random observation variable Υ. Then the posterior probability density of Q, given the measurements v_{obs}, is

$$\pi(q|v_{obs}) = \frac{\pi(v_{obs}|q)\pi_0(q)}{\pi(v_{obs})} = \frac{\pi(v_{obs}|q)\pi_0(q)}{\int_{\mathbb{R}^p} \pi(v_{obs}|q)\pi_0(q)dq}. \tag{4.20}$$

In the above definition, the likelihood function $\pi(v|q)$ quantifies the probability of obtaining the observation v for a given value q of the random parameter Q. This function depends on the assumptions made for the distributions of the errors in the measurement process. Following [79], we assume the measurement errors are iid and that the errors $\epsilon_i \sim N(0, \sigma^2)$, with σ^2 a fixed value. In this case, the likelihood function is given by

$$\pi(v|q) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-SS_q/2\sigma^2}, \tag{4.21}$$

where

$$SS_q = \sum_{i=1}^{n} [v_i - f_i(q)]^2 \tag{4.22}$$

is the sum of squares error between the measurements v_i and the parameter-dependent model responses $f_i(q)$. The likelihood function in the Bayesian framework represents a residual error between the left and right hand sides of equation (4.20). It is this residual that our algorithm will seek to minimize.

The idea of this algorithm is to construct a Markov chain whose stationary distribution is the posterior density $\pi(q|v)$. To do this, we begin with determining a parameter q^0 which minimizes equation (4.22). From there, using an estimation of the covariance matrix V_0 given by

$$V_0 \simeq \sigma_0^2 [\chi^T(q^0)\chi(q^0)]^{-1}, \tag{4.23}$$

where $\chi(q^0)$ denotes the $n \times p$ sensitivity matrix given by

$$\chi_{ik}(q^0) = \frac{\partial f_i(q^0)}{\partial q^k}, \tag{4.24}$$

we propose a new value $q^* \sim J(q^*|q^0)$, where J is called the proposal or jumping distribution. The notation indicates that J specifies q^* based on the previous value q^0, and it should be noted that J is not to be interpreted as a conditional probability density. Then, with probability $\alpha(q^*|q^0)$, which is determined by properties of the likelihood function and prior probability density, we accept q^*. This is the essence of a standard Metroplis algorithm. However, DRAM is an Adaptive Metropolis algorithm, in that the first k_0 parameters $q^0, ..., q^{k_0-1}$ are computed using the initial covariance matrix V_0, and the next set of k_0 parameters are then determined by an updated chain covariance matrix V_k given by

$$V_k = s_p \text{cov}(q^0, ..., q^{k-1}) + \epsilon I_p. \tag{4.25}$$

Here, s_p is a parameter which depends on the dimension p of the parameter space. We choose the common choice of $s_p = 2.38^2/p$, as outlined in [19]. The length k_0 is chosen to ensure a nonsingular covariance matrix in the initial parts of the chain. In practice, $k_0 = 100$ is usually taken, and we will follow such conventions. The term ϵI_p, with $\epsilon \geq 0$ and I_p the p-dimensional identity matrix, ensures that V_k is positive definite. For our purposes, this will not be a concern and we can take $\epsilon = 0$.

If the guess q^* is not accepted, we do not reject immediately as is the case for standard Metropolis algorithms. Instead, we propose a 2nd guess q^{*2} based on the current updated covariance matrix to give the solver more of a chance to accept a new chain value. This is the principle of delayed rejection. Similar to the first guess, we compute a probability $\alpha_2(q^{*2}|q^0)$ which is now determined by the likelihood function, the prior density of q^0, and now also on the first guess q^*. Then the algorithm accepts or rejects this guess with probability α_2. If the algorithm rejects both candidates, we set $q^1 = q^0$, otherwise, we update q^1 to be either q^* or q^{*2}, whichever was accepted.

We present the steps of the DRAM algorithm here, which are also given in [79].

1. Set design parameters and number of chain iterates M.

2. Determine $q^0 = \arg\min_q \sum_{i=1}^{n}[v_i - f_i(q)]^2$.

3. Set $SS_{q^0} = \sum_{i=1}^{n}[v_i - f_i(q^0)]^2$.

4. Compute initial variance estimate: $s_0^2 = \frac{SS_{q^0}}{n-p}$.

5. Construct covariance estimate $V = s_0^2[\chi^T(q^0)\chi(q^0)]^{-1}$ and R = chol(V), the Cholesky decomposition of V.

6. For $k = 1, ..., M$

   (a) Sample $z_k \sim N(0, I_p)$.

   (b) Construct candidate $q^* = q^{k-1} + Rz_k$.

   (c) Sample $u_\alpha \sim \mathcal{U}(0,1)$.

   (d) Compute $SS_{q^*} = \sum_{i=1}^{n}[v_i - f_i(q^*)]^2$.

   (e) Compute $\alpha(q^*|q^{k-1}) = \min\left(1, e^{-[SS_{q^*} - SS_{q^{k-1}}]/2s_{k-1}^2}\right)$, the probability of accepting the first stage candidate.

(f) If $u_\alpha < \alpha$,

    Set $q^k = q^*, SS_{q^k} = SS_{q^*}$

else

  i. Set the design parameter $\gamma_2 = 0.2$.

  ii. Sample $z_k \sim N(0, I_p)$.

  iii. Construct second-stage candidate $q^{*2} = q^{k-1} + \gamma_2 R_k z_k$.

  iv. Sample $u_\alpha \sim \mathcal{U}(0, 1)$.

  v. Compute $SS_{q^{*2}} = \sum_{i=1}^{n}[v_i - f_i(q^{*2})]^2$.

  vi. Compute $\alpha_2(q^{*2}|q^{k-1}, q^*)$, the probability of accepting the 2nd stage candidate (See [79] for explicit form of α_2).

  vii. If $u_\alpha < \alpha$,

    Set $q^k = q^{*2}, SS_{q^k} = SS_{q^{*2}}$.

  else

    Set $q^k = q^{k-1}, SS_{q^k} = SS_{q^{k-1}}$

  endif

endif

(g) Update $s_k^2 \sim$ Inv-gamma(a_{val}, b_{val}), where $a_{val} = 0.5(n_s + n), b_{val} = 0.5(n_s \sigma_s^2 + SS_{q^k})$.

(h) if mod$(k, k_0) = 1$

  Update $V_k = s_p \text{cov}(q^0, ..., q^k)$

else

  $V_k = V_{k-1}$

endif

(i) Update $R_k = \mathrm{Chol}(V_k)$.

In Figure 4.4, we present the results for the chains that predict the setpoints for the values of filtration cycle time and filtration fluid flux for a single filtration cycle. We simulated the same cycle using the normal value for viscosity $\eta = 8.9e - 4$ as well as for a larger viscosity $\eta = 5.0e - 3$. In each case, we let our initial parameter guess $q^0 = [J_f^0, t_f^0] = [10, 10]$ and the number of chain iterates $M = 100,000$. The first 5,000 iterations were removed from the plot to give the algorithm time to settle and to give a clearer plot of the values. We also present MCMC statistics given at the end of each simulation in Table 4.7.

It is clear that even with many chain iterates, the algorithm still finds newer candidates to accept. Because of the stochastic nature of the algorithm, each run manifests a different chain. For a well-posed problem with a clearly defined solution, DRAM will always yield chains which converge to the same solution, given any initial conditions. However, in our case, with just one initial condition, DRAM will consistently give candidates which bounce around throughout all iterations, never settling on a single point for an extended period. Moreover, the stationary distribution of the chain by the end has standard deviations which are nearly the size of the mean itself, as is seen in Table 4.7. This gives us even more evidence that the run-to-run model is not well suited for this particular application. We also note that changes in viscosity do little to effect the convergence of these simulations, however higher viscosities do correlate with longer predicted filtration times.

During these runs, we kept track of a few convergence metrics, which are summarized in Table 4.7, to analyze how well the DRAM algorithm is converging for this particular simulation. The MC error parses the chain into partitions. The mean of each of these partitions is evaluated, and the standard deviation of these means from the total mean is taken to be the MC error. Because the MC error is relatively small, this indicates that the partition means are close to one another. The τ metric is the integrated auto-correlation time, which is computed in our simulations via Sokal's

adaptive truncated periodogram estimator [33]. This metric gives us the estimated number of steps needed to draw independent samples from the posterior distribution. Smaller values of τ indicate that the chain converges quickly. Given that τ is on the order of 10^4 for both parameters in both simulations, this gives further evidence that these chains do not converge quickly, or at all. The Geweke parameter also partitions the chain into pieces and tests the resulting pieces for similarity. Normally, the first ten percent and the last 50 percent of the chain are chosen for comparison, and we followed this convention for these simulations. A Geweke value close to one indicates that the first and last elements of the chain are from the same distribution. However, for both simulations, we observe Geweke parameters that are more close to zero, which indicates that these simulations are struggling to converge, even after 100,000 iterations.

Interestingly, when plotting the accepted values of J_f and t_f against each other, it is clear that there is a linear correlation between these two parameters. Hence, these simulations show that the filtration time and fluid flux have some relation between them that could be incorporated into a future revision of the run-to-run control model.

Figure 4.3: DRAM algorithm results on single filtration cycle, $\eta = 8.9e - 4$
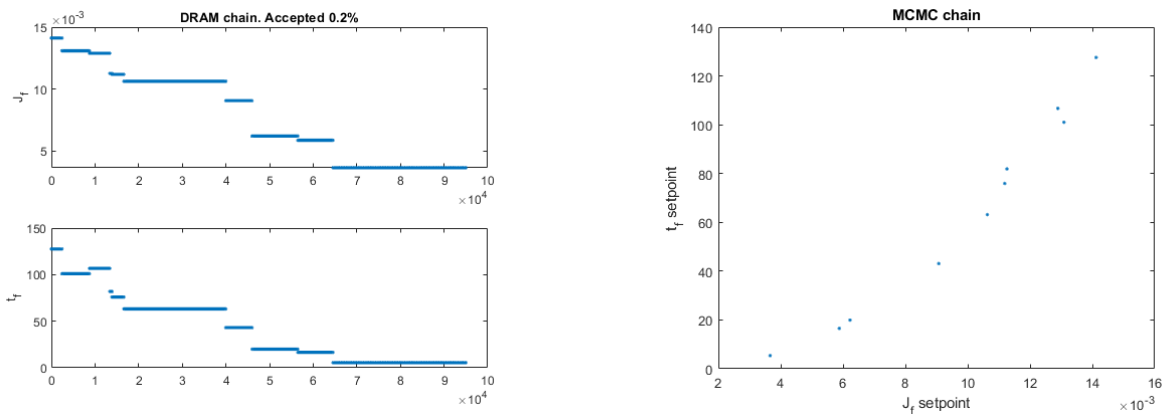
4.5 Problems with Run-to-Run Control

There are many reasons for which this run-to-run model is not accurate enough to describe the

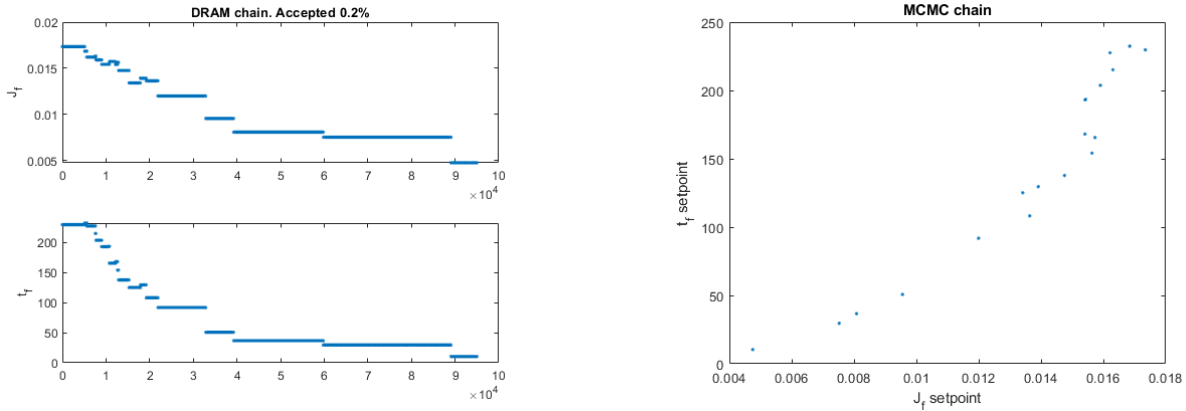Figure 4.4: DRAM algorithm results on single filtration cycle, $\eta = 5.0e-3$

Table 4.7: MCMC statistics for DRAM simulations

Viscosity	Setpoint	Mean	Std. Dev.	MC Error	Tau	Geweke
8.9e-4	J_f	0.0077914	0.003563	0.00081046	12,600	0.12768
8.9e-4	t_f	41.42	36.237	8.2217	12,255	0.027321
5.0e-3	J_f	0.0099432	0.0034773	0.00079164	11,208	0.21414
5.0e-3	t_f	72.139	64.143	14.627	10,889	0.038891

phenomenon of filtration in this setting. One obvious reason is the lack of sufficient plant data to realize the model. This is a common problem for most membrane filtration systems, in particular for the backwashing phase, in which the dynamics of the flow during backwashing are highly erratic. While models have been proposed to describe the dynamics, in this case and many others, there is simply not any reliable input data for the backwashing phase. In particular, the headloss for the membranes at the River Mountains, which is used to determine the TMP, is not recorded at all during the backwashing phases. Knowledge of the TMP is critical during these phases, since it is needed along with fluid flux to compute the resistance of the membrane during backwashing via Darcy's Law. Since this was not known, we could not properly model resistance during backwashing, and thus all contributions to the model from backwash resistance needed to be neglected.

One other possibility is that this model has only been implemented successfully for a membrane bioreactor located in Simmerath, Germany [64]. The filters in this bioreactor have a filtration area of only $29m^2$, while the River Mountains filters each have an area of $180m^2$. In addition,

the filtration media are much different between these applications. The bioreactor uses a PURON hollow fibre module, while the River Mountains filters are gravity based direct filtration units, using the water level pressure to force the water through a filter consisting of a $1.8m$ thick layer of anthracite coal followed by a $0.2m$ thick layer of sand. The difference in size, scope, and method of filtration could be possible reasons that the run-to-run model fails to adequately describe the filtration/backwashing process.

Finally, as shown in the DRAM simulations, there is a linear correlation between the J_f and t_f candidates computed from the chains. This suggests that there is a link between these two state variables that is not explicitly demonstrated within our current model framework. Such a relation could be investigated and incorporated into the run-to-run model itself, which could allow the run-to-run model to produce better convergence results.

Because of the inadequacies of this model, we sought to find a way to relate the measured data we do have from the filtration/backwashing process to the setpoints of operation, namely the filtration and backwashing times, and the flux setpoints for each of these cycles. To deal with these inadequacies, we have also begun devising a machine learning model to probe into the relationship between all the most relevant data available in the SCADA database at River Mountains. Variables being taken into consideration as inputs for the machine learning model are the effluent filtration and backwash fluxes, turbidity, particle count, headloss, and the spread of each of the flows. We wish to determine the importance and inter-relations between these variables and their effects on the setpoints for filtration and backwashing times. This research is currently being conducted at the time of the submitting of this dissertation.

PART III

CONCLUSION

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this dissertation, we computationally studied TRM and its sensitivity equations. These equations were both discretized using finite element in space and second order Crank-Nicolson method in time. We tested the performance of this model with different deconvolution orders, i.e. $N =0$, 1, and 2, using the Shear Layer Roll-up benchmark problem. Our computations using Olson's suggested average filter length scale produced improved results and confirmed the positive effect of such scaling for higher deconvolution order N. The sensitivity computations of TRM with respect to the variation of parameter χ verified the range of this parameter values for a reliable approximated solution. We noted that the sensitivity was lower as the order of deconvolution was increased, and the average filter length scale also contributed to better sensitivities overall for higher orders of deconvolution.

We implemented the EMAC discretization for the TRM model and developed a comparison of it with SKEW. Energy, momentum, and angular momentum balances were derived, which showed that EMAC is a better alternative than SKEW because it not only has an analogue to the energy balance for TRM, but also analogues for the momentum and angular momentum balances. We showed finite element stability and error estimates. In particular, we showed that the error estimate for EMAC is much better than SKEW, as the SKEW error is on the order of $O(e^{\nu^{-3}})$, but EMAC error is on the order of $O(e^{\nu^{-1}})$, which is a significantly smaller order of magnitude in the case of turbulent fluid flow.

We performed several numerical experiments on both EMAC and SKEW to test these theoretical findings. We ran the 2D Taylor Green vortex problem, which confirmed the convergence rates given by our error estimates. A 3D numerical simulation of the Ethier-Steinman problem with

high Reynolds number showed that the error for EMAC is significantly reduced compared to the SKEW, demonstrating that the error estimate for EMAC is much better. We also did experiments on the Lattice Vortex problem, which verified that numerical errors for the EMAC scheme were much smaller than for SKEW scheme, and that EMAC kept the energy, momentum, and angular momentum bounded during the simulation when SKEW could not under the same conditions.

We analyzed a run-to-run control model proposed for modeling membrane filtration processes, and applied it to the filtration/backwashing systems at the River Mountains Facility. Several MATLAB scripts were written to compute setpoints of operation for the filtration/backwashing systems which optimize the overall energy cost of operating these systems. Each of these scripts used different solvers ranging from standard function optimization with constraints to genetic algorithms and MCMC methods. These algorithms were fed input data for several filtration/backwash cycles. After extensively testing these algorithms for their effectiveness, we determined that the proposed run-to-run control model is not adequate for describing the filtration/backwashing phenomenon in the large filtration basins at River Mountains. Future research on the filtration/backwashing systems will be conducted, including the construction of a machine learning model which will help inform us of parameter importance among the measured data available at River Mountains.

BIBLIOGRAPHY

[1] Z. Xiong and J. Zhang. A Batch-to-batch iterative optimal control strategy based on recurrent neural network models. *Journal of Process Control*, 15:11–21, 2005.

[2] O. Lehmkuhl, G. Houzeaux, H. Owen, G. Chrysokentis and I. Rodriguez. A low dissipation finite element scheme for scale resolving simulations of turbulent flows. Journal of Comp. Phys., 390:51–65, 2019.

[3] A. Palha and M. Gerritsma. A mass, energy, enstrophy and vorticity conserving (meevc) mimetic spectral element discretization for the 2D incompressible Navier-Stokes equations. *Journal of Comp. Phys.*, 328:200–220, 2017.

[4] A. Arakawa and V. Lamb. A potential enstrophy and energy conserving scheme for the shallow water equations. *Monthly Weather Review*, 109:18–36, 1981.

[5] S. Breckling, T. Hill and M. Neda. A review of time relaxation methods. *Fluids*, 2:1–20, 2017.

[6] J. Bell, P. Collela and H. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Int. Journal Numer. Meth. Fluids*, 8:537–557, 1989.

[7] N. Adams and S. Stolz. A subgrid-scale deconvolution approach for shock capturing. *Journal of Computational Physics*, 178:391–426, 2001.

[8] B. Wittenmark Adaptive dual control methods: An overview. *IFAC Proceedings*, 28:67–72, 1995.

[9] N. Adams, S. Stolz, and L. Kleiser. An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows. *Physics of Fluids*, 13:997–1015, 2001.

[10] L. Rebholz. An energy and helicity conserving finite element scheme for the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 45:1622–1638, 2017.

[11] D. Bonne and S. Jorgensen. Batch to batch improving control of yeast fermentation. *Computer Aided Chemical Engineering*, 9:621–626, 2001.

[12] L. Breiman, J. Friedman, R. Olshen and C. Stone. *Classification and Regression Trees.* Chapman and Hall / CRC, 1998.

[13] A. Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two dimensional incompressible flow, Part I.. *Journal of Comp. Phys.*, 1:119–143, 1966.

[14] J. Connors. Convergence analysis and computational testing of the finite element discretization of the Navier-Stokes-α model. *Numerical Methods for Partial Differential Equations*, 26:1328–1350, 2010.

[15] S. Merdan and C. Manica. Convergence analysis of the finite element method for a fundamental model in turbulence. *Math. Models and Methods in Applied Sciences*, 22:24pp, 2012

[16] M. Germano. Differential filters for the large eddy numerical simulation of turbulent flows. *Physics of Fluids*, 29:1755–1757, 1986.

[17] M. Neda. Discontinuous time relaxation method for the time-dependent Navier-Stokes equations. *Advances in Numerical Analysis*, 2010(3-4).

[18] R. Abramov and A. Majda. Discrete approximations with additional conserved quantities. *Methods Appl. Anal.*, 10:151–190, 2003.

[19] H. Haario, M. Laine, A. Mira and E. Saksman. DRAM: Efficient adaptive MCMC. *Stat. Comput.*, 16:339–354, 2006.

[20] S. Charnyi, T. Heister, M. Olshanskii and L. Rebholz. Efficient discretizations for the EMAC formulation of the incompressible Navier-Stokes equations. *Applied Numerical Mathematics*, 141:220–233, 2017.

[21] D. Foreman-Mackey, D. Hogg, D. Lang and J. Goodman. Emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific* 125:306–312, 2013.

[22] C. Manica, M. Neda, M. Olshanskii, and L. Rebholz. Enabling numerical accuracy of Navier-Stokes-α through deconvolution and enhanced stability. *ESAIM: Mathematical modelling and numerical analysis*, 45:277–307, 2010.

[23] J. Liu and W. Wang. Energy and helicity preserving schemes for hydro and magnetohydro-dynamics flows with symmetry. *Journal of Comp. Phys.*, 200:8–33, 2004.

[24] O. Lehmkuhl, F. Sacco, B. Paun, T. Iles, P. Iaizzo, G. Houzeaux, M. Vazquez, C. Butakoff and J. Aguado-Sierra. Evaluating the roles of detailed endocardial structures on right ventricular haemodynamics by means of CFD simulation. *International Journal of Numerical Methods in Biomedical Engineering*, 34:1–14, 2018.

[25] R. Ethier and D. Steinman. Exact fully 3D Navier-Stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids*, 19:369–375, 1994.

[26] I. Stanculescu. Existence theory of abstract approximate deconvolution models of turbulence. *Annali dell' Universita di Ferrara*, 54:145–168, 2008.

[27] P. Rosenau Extending hydrodynamics via the regularization of the Chapman–Enskog expansion. *Physical Review A.*, 40:7193–7196, 1989.

[28] P. Fischer and J. Mullen. Filter-based stabilization of spectral element methods. *C. R. Acad. Sci. Paris Ser I Math.*, 332:265–270, 2001.

[29] J. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier-Stokes problem, Part IV: Error analysis for the second order time discretization. *SIAM Journal on Numerical Analysis*, 27:353–384, 1990.

[30] M. Neda and J. Waters. Finite element computations of time relaxation algorithm for flow ensembles. *Applied Engineering Letters*, 1:51–56, 2016.

[31] G. Fix. Finite element models for ocean circulation problems. *SIAM Journal on Applied Mathematics*, 29:371–387, 1975.

[32] L. Davis, M. Neda, F. Pahlevani, and J. Waters. Fluid models and parameter sensitivities: Computations and applications. *International Journal of Novel Ideas: Mathematics*, 1:12–39, 2017.

[33] A. Sokal. *Functional integration: Basics and applications.* Springer US, Boston, MA, 1997.

[34] R. Salmon and L. Talley. Generalizations of Arakawa's jacobian. *Journal of Comp. Phys.*, 83:247–259, 1989.

[35] "The Mathworks, Inc.". *How pattern search polling works - MATLAB and Simulink.* https://www.mathworks.com/help/gads/how-pattern-search-polling-works.html. October 18th, 2021.

[36] "The Mathworks, Inc.". *How the genetic algorithm works.* https://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html. October 18th, 2021.

[37] M. Neda, X. Sun and L. Yu. Increasing accuracy and efficiency for regularized Navier-Stokes equations. *Acta Applicandae Mathematicae*, 118:57–79, 2012.

[38] W. Layton. *Introduction to finite element methods for incompressible, viscous flows.* SIAM Publications, Philadelphia, PA, 2008.

[39] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging.* CRC Press, Boca Raton, FL, 1998.

[40] W. Gao and S. Engell. Iterative set-point optimization of batch chromatography. *Computers and Chemical Engineering*, 29:1401–1409, 2005.

[41] D. Pastrana, J. Cajas, O. Lehmkuhl, I. Rodriguez and G. Houzeaux. Large-eddy simulations of the vortex-induced vibration of a low mass ratio two-degree-of-freedom circular cylinder at subcritical Reynolds number. *Computers and Fluids*, 173:118–132, 2018.

[42] O. Lehmkuhl, F. Sacco, B. Paun, T. Iles, P. Iaizzo, G. Houzeaux, M. Vazquez, C. Butakoff and J. Aguado-Sierra. Left ventricular trabeculations decrease the wall shear stress and increase the intra-ventricular pressure drop in CFD simulations. *Frontiers in Physiology*, 9:1–15, 2018.

[43] L. Rebholz and M. Olshanskii. Longer time accuracy for incompressible Navier-Stokes simulations with the EMAC formulation. *Computer Methods in Applied Mechanics and Engineering*, 372:113369, 2020.

[44] L. Berselli, T. Iliescu and W. Layton. *Mathematics of Large Eddy Simulation of Turbulent Flows.* Heidelberg: Springer, 2006

[45] V. Girault, R. Nochetto and L. Scott. Max-norm estimates for Stokes and Navier-Stokes approximations in convex polyhedra. Numerische Mathematik, 131:771–822, 2015.

[46] E. Olson. Model Error in the LANS-α and NS-α Deconvolution Models of Turbulence. *International Journal of Numerical Analysis and Modeling*, 15:811–833, 2018.

[47] E. Sachs, A. Hu, A. Ingolfsson and P. Langer. Modeling and control of an epitaxial silicon deposition process with step disturbances. *[1991 Proceedings] IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, 104–107, 1991.

[48] R. Temam. *Navier-Stokes equations: Theory and numerical analysis.* AMS Chelsea Publishing, 2001.

[49] F. Hecht. New development in FreeFem++ *Journal of Numerical Mathematics*, 20:251–265, 2012.

[50] R. Martin, M. Soria, O. Lehmkuhl, A. Gorobets and A. Duben. Noise radiated by an open cavity at low Mach number. *International Journal of Aeroacoustics*, 18:647–668, 2019.

[51] Y. Bard. *Nonlinear Parameter Estimation.* New York and London: Academic Press, 1974.

[52] R. Guenanff (2004). *Non-stationary coupling of Navier-Stokes/Euler for the generation and radiation of aerodynamic noises (PhD thesis, Universite Rennes, Rennes, France).*

[53] W. Layton, C. Manica, M. Neda and L. Rebholz. Numerical analysis and computational testing of a high accuracy Leray-deconvolution model of turbulence. Numerical Methods for Partial Differential Equations, 24:555–582, 2008.

[54] S. De, D. Hannasch, M. Neda and E. Nikonova. Numerical analysis and computations of a high accuracy time relaxation fluid flow model. *International Journal of Computer Mathematics*, 89:2353–2373, 2012.

[55] V. Ervin, W. Layton and M. Neda. Numerical analysis of a higher order time relaxation model of fluids. *International Journal of Numerical Analysis and Modeling*, 4:648–670, 2007.

[56] A. Dunca and M. Neda. Numerical analysis of a nonlinear time relaxation model of fluids. *Journal of Mathematical Analysis and Applications*, 420:1095–1115, 2014.

[57] G. Baker, V. Dougalis and O. Karakashian. On a higher order accurate, fully discrete Galerkin approximation to the Navier-Stokes equations. *Math. Comp.*, 39:339–375, 1982.

[58] S. Charnyi, T. Heister, M. Olshanskii and L. Rebholz. On conservation laws of Navier-Stokes Galerkin discretizations. *Journal of Computational Physics*, 337:289–308, 2017.

[59] P. Schroeder, V. John, P. Lederer, C. Lehrenfeld, G. Lube and J. Schoberl. On reference solutions and the sensitivity of the 2D Kelvin-Helmholtz instability problem. *Computers and Mathematics with Applications*, 77:1010–1028, 2019.

[60] G. Taylor. LXXV. On the decay of vortices in a viscous fluid. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 46:671–674, 1923.

[61] K. Jepsen, M. Bram, L. Hansen, Z. Yang and S. Lauridsen. Online backwash optimization of membrane filtration for produced water treatment *Membranes*, 9:68, 2019.

[62] D. Brown and M. Minion. Performance of under-resolved two-dimensional incompressible flow simulations. *Journal of Comp. Phys.*, 122:283–288, 1995.

[63] P. Schroeder and G. Lube. Pressure-robust analysis of divergence-free and conforming FEM for evolutionary incompressible Navier-Stokes flows. *Journal of Numer. Math.* 25:249–276, 2017.

[64] J. Busch and W. Marquardt. Run-to-run control of membrane filtration in wastewater treatment - An experimental study. *IFAC Proceedings*, 2:195–200, 2007.

[65] J. Busch, A. Cruse and W. Marquardt. Run-to-run control of membrane filtration processes. *AIChE Journal*, 53:2316–2328, 2007.

[66] J. Busch and W. Marquardt. Run-to-run control of membrane filtration processes. *IFAC Proceedings*, 39:1003–1008, 2006.

[67] E. Castillo and A. Hurwitz. Run-to-run process control: Literature review and extensions. *Journal of Quality Technology*, 29:184–196, 1997.

[68] M. Neda, F. Pahlevani and J. Waters. Sensitivity analysis and computations of the time relaxation model. *Journal of Advances in Applied Mathematics and Mechanics*, 7:89–115, 2015.

[69] M. Neda and E. Nikonova. Sensitivity computations of time relaxation model with an application in cavity computation. World Academy of Science, Engineering and Technology, 79:542–545, 2011.

[70] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, New York, 2005.

[71] N. Adams, S. Stolz, and L. Kleiser. The approximate deconvolution model for large-eddy simulations of compressible flows and its application to shock-turbulent-boundary-layer interaction. *Physics of Fluids*, 13:2985–3001, 2001.

[72] A. Kolmogorov. The local structure of turbulence in incompressible viscous fluids for very large Reynolds numbers. *Doklady Akademii Nauk SSR*, 30:9–13, 1941.

[73] S. Brenner and R. Scott. *The mathematical theory of finite element methods.* Springer-Verlag, 1994

[74] S. Schochet and E. Tadmor. The regularized Chapman-Enskog expansion for scalar conservation laws. *Archive for Rational Mechanics and Analysis*, 119:95–107, 1992.

[75] A. Labovsky, W. Layton, C. Manica, M. Neda and L. Rebholz. The stabilized, extrapolated trapezoidal finite element method for the Navier-Stokes equations. Computer Methods in Applied Mechanics and Engineering, 198:958–974, 2009.

[76] A. Takhirov, M. Neda and J. Waters. Time relaxation algorithm for flow ensembles *Numerical Methods for Partial Differential Equations*, 32:757–777, 2016.

[77] W. Layton and M. Neda. Truncation of scales by time relaxation. *Journal of Mathematical Analysis and Applications*, 325:788–807, 2007.

[78] L. Kushner, W. Hubbard and R. Parker. Turbidity and Viscosity Measurements on Some Cationic Detergents in Water and in Sodium Chloride Solutions. *Jouranl of Research of the National Bureau of Standards*, 59:113–119, 1957.

[79] R. Smith *Uncertainty Quantification - Theory, Implementation, and Applications.* SIAM Computational Science and Engineering, 2014.

[80] A. Majda and A. Bertozzi. *Vorticity and Incompressible Flow.* Cambridge University Press, Cambridge, England, 2001.

CURRICULUM VITAE

Graduate College

University of Nevada, Las Vegas

Jeffrey Belding

jeffrey.belding@unlv.edu

jeffbelding92@gmail.com

Degrees:

Bachelor of Science, Pure Mathematics, 2015

University of Nevada, Reno in Reno, Nevada

Bachelor of Science, Physics, 2015

University of Nevada, Reno in Reno, Nevada

Awards:

CALMS (California Lake Management Society) 2021 Student Scholarship Winner   May, 2021.

Publications:

Computational Study of the Time Relaxation Model with High Order Deconvolution Operator, **Jeffrey Belding**, Monika Neda, Fran Pahlevani, Results in Applied Mathematics (Results Appl. Math.) 8:100–111, 2020.

An Efficient Discretization for a Family of Time Relaxation Models, **Jeffrey Belding**,

Monika Neda, Rihui Lan, Computer Methods in Applied Mechanics and Engineering (Comp. Meth. in Appl. Mech. and Eng.), 2021 (Under review)

Application of a Run-to-Run Control Model for Membrane Filtration at a Large Urban Water Treatment Facility, **Jeffrey Belding**, 2021 (To be submitted)

Dissertation Title:

Numerical Studies of Regularized Navier-Stokes Equations and an Application of a Run-to-Run Control model for Membrane Filtration at a Large Urban Water Treatment Facility

Dissertation Examination Committee:

Chairperson, Dr. Monika Neda, Ph.D.

Committee Member, Dr. Jichun Li, Ph.D.

Committee Member, Dr. Pengtao Sun, Ph.D.

Committee Member, Dr. Hongtao Yang, Ph.D.

Graduate Faculty Representative, Dr. Pushkin Kachroo, Ph.D.