

A TV-Based Iterative Regularization Method for the Solutions of Thermal Convection Problems

Eric T. Chung and Jeff C.-F. Wong*

Department of Mathematics, Lady Shaw Building, The Chinese University of Hong Kong, Shatin, Hong Kong.

Received 27 February 2012; Accepted (in revised version) 31 January 2013

Available online 27 May 2013

Abstract. Linear/nonlinear and Stokes based-stabilizations for the filter equations for damping out primitive variable (PV) solutions corrupted by uniformly distributed random noises are numerically studied through the natural convection (NC) as well as the mixed convection (MC) environment. The most recognizable filter-scheme is based on a combination of the negative Laplace equation multiplied with the selection of the spatial scale and a linear function in order to preserve the uniqueness of the filtered solution. A more complicated filter-scheme, based on a Stokes problem which couples a filtered velocity and a filtered (artificial) pressure (or Lagrange multiplier) in order to enforce the incompressibility constraint, is also studied. Linear and Stokes based-filters via nested iterative (NI) filters and the consistent splitting scheme (CSS) are proposed for the NC/MC problems. Inspired by the total-variation (TV) model of image diffusion, well preserved feature flow patterns from the corrupted NC/MC environment are obtained by TV-Stokes based-filters together with the CSS. Our experimental results show that our proposed algorithms are effective and efficient in eliminating the unwanted spurious oscillations and preserving the accuracy of thermal convective fluid flows.

AMS subject classifications: 65M60, 76M10

Key words: Linear/nonlinear differential filter, Stokes differential filters, Leray-alpha model, total-variation regularization.

1 Introduction

The differential filter is one of the most straightforward regularization or stabilization methods in Computational Fluid Dynamics (see [7] for a review), where resolved amplitudes of fluctuations of fluid flows can be diminished. Let ζ and $\bar{\zeta}$ be an unfiltered and

*Corresponding author. *Email addresses:* tschung@math.cuhk.edu.hk (E. T. Chung), jwong@math.cuhk.edu.hk (J. C.-F. Wong)

a filtered scalar or vector field variable, respectively. The model equation is based on a partial differential equation of evolution type described as follows (e.g., [8,9]):

Given ξ , find a unique solution $\bar{\xi}$ such that

$$\begin{cases} -\alpha^2 \nabla^2 \bar{\xi} + \bar{\xi} = \xi, & \text{in } \Omega, \\ \bar{\xi} = \xi, & \text{on } \partial\Omega, \end{cases} \quad (1.1)$$

where α , $0 < \alpha$, represents a regularization length (or spatial) scale. By adjusting α , the reduction of the unwanted spurious oscillations corrupted by uniformly distributed random noises, can be accomplished simply by the smoothing filter $(-\alpha^2 \nabla^2 + I)^{-1}$, where ∇^2 and I are the Laplace and identity operators, respectively.

When considering the incompressible flow problems, the extension to Eq. (1.1) is done by adding a filtered pressure gradient term as a Lagrange multiplier and enforcing the incompressibility constraint, often referred to as the Stokes differential filter (e.g., [10,12]). The method is described as follows:

Given ξ , find a pair of unique solutions $(\bar{\xi}, \lambda)$ such that

$$\begin{cases} -\alpha^2 \nabla^2 \bar{\xi} + \bar{\xi} + \nabla \lambda = \xi, & \text{in } \Omega, \\ \nabla \cdot \bar{\xi} = 0, & \text{in } \Omega, \\ \bar{\xi} = \xi, & \text{on } \partial\Omega, \end{cases} \quad (1.2)$$

where the filtered pressure λ is assumed known up to an arbitrary constant. It is known that the choice of the boundary conditions in (1.1) and (1.2) is an open problem. However, those that we used are the most reasonable choices and are widely accepted in literature.

Let us consider the NC as well as the MC environment. Assume at each time, the primitive variable (PV) $\xi = \{u, v, p, T\}$, where u and v are velocities, p is the pressure and T is the temperature, in this study is generated from the error-free one and is assumed to have random noise, as shown in Eq. (1.3)

$$\xi = \xi_{\text{free}} + \omega \hat{\sigma}, \quad (1.3)$$

where the term $\hat{\sigma}$ is the standard deviation of the random errors, which is supposed constant and ω is a random variable with normal distribution, zero mean, and unitary standard deviation, so that Eq. (1.3) describes an unpredictable error in the solution of ξ . In order to recover/preserve the fluid flow details as much as possible, the noise is removed in different successive filtering steps that consist of the following four-step strategy [13]:

At each time, evolve-filter-deconvolve-relax.

The four-step strategy can be sequentially described as follows:

1. The evolution step is used to determine each PV (or unfiltered) solution at the $n+1^{\text{th}}$ time-step from the n^{th} time-step.

2. The filtration step is used to stabilize the solutions and conduct the noise reduction by means of Eq. (1.1) or Eq. (1.2).
3. The deconvolution step is used to enhance the accuracy of the filtered solutions when using Eq. (1.1) or Eq. (1.2) again.
4. The relaxation step is used to reduce the induced numerical diffusion driven by the filtration and deconvolution steps as the time-step size Δt tends to zero (e.g., [14]).

As we shall see below, proper use of the deconvolution and the relaxation steps is required when adding the random noise. Based on our numerical experiments, the filtration step used to reduce noise in numerical data with linear/nonlinear filters would not be efficiently worked out due to the one-step iteration. Inspired by the works of Ingram et al. [15] and Mays [17], in turn, we shall combine the filtration step with the deconvolution step and formulate these two steps as a nested iterative process, coined as *an iterative filter/deconvolve step* that dampens spurious solutions and correctly predicts the flow. Henceforth, the relaxation step will work effectively. A three-step strategy is:

At each time, evolve-‘iterative filter/deconvolve’-relax.

Iterative differential and iterative Stokes-based filters may work fine in a class of large amplitudes of fluctuations of an unfiltered solution. As reported by many others, these diffusion-based filters may introduce too much diffusion when faced with steep gradients and sharp corners of the fluid flow structure, and other applications for example [6]. In this work, the TV based regularization is used; adding the anisotropic filter (or nonlinear diffusion) term in Eqs. (1.1) and (1.2) improves the situation. By introducing a smooth auxiliary (or dual) variable(s) in the shrinkage formulation

$$g(\bar{\zeta}) = \begin{cases} \frac{\nabla \bar{\zeta}}{\|\bar{\zeta}\|_{\text{TV}}}, & \text{if } \|\bar{\zeta}\|_{\text{TV}} \geq \epsilon, \\ 0, & \text{otherwise,} \end{cases} \quad (1.4)$$

where $\|\bar{\zeta}\|_{\text{TV}} := \left| \frac{\partial \bar{\zeta}}{\partial x} \right| + \left| \frac{\partial \bar{\zeta}}{\partial y} \right|$, and $\epsilon > 0$ is the regularized parameter chosen near a zero value, Eqs. (1.1) and (1.2) can be reformulated as:

Given ζ , find a unique solution $\bar{\zeta}$ such that

$$\begin{cases} -\alpha^2 \nabla^2 \bar{\zeta} - \alpha_{\text{TV}}^2 \nabla \cdot (g(\bar{\zeta})) + \bar{\zeta} = \zeta, & \text{in } \Omega, \\ \bar{\zeta} = \zeta, & \text{on } \partial\Omega, \end{cases} \quad (1.5)$$

and given ζ , find a pair of unique solutions $(\bar{\zeta}, \lambda)$ such that

$$\begin{cases} -\alpha^2 \nabla^2 \bar{\zeta} - \alpha_{\text{TV}}^2 \nabla \cdot (g(\bar{\zeta})) + \bar{\zeta} + \nabla \lambda = \zeta, & \text{in } \Omega, \\ \nabla \cdot \bar{\zeta} = 0, & \text{in } \Omega, \\ \bar{\zeta} = \zeta, & \text{on } \partial\Omega, \end{cases} \quad (1.6)$$

where α_{TV} , $0 < \alpha_{TV}$, also represents a regularization length scale. It is well-known that the choice of regularization parameters α and α_{TV} is problem-dependent, and the method for systematically choosing these parameters is currently not known. Thus, we will present the best results by comparing various choices of these parameters. The use of (1.4) is related to the nonlinear diffusion approach introduced in [11]. Instead of using Eq. (1.4) to control diffusion, the authors in [11] use an indicator function to control diffusion.

Using the combination of the consistent splitting scheme (CSS) [18] for solving the momentum equations and mixed finite element (FE) methods for spatial discretization (see e.g., [19]), the aim of the present research is:

- to numerically study linear and nonlinear iterative differential filters (cf. Algorithm 3.1 and Algorithm 3.3 describe Eq. (1.1) and Eq. (1.5), respectively) for cutting down on the unpredictable error in the solutions of $\{p, T\}$, and use iterative Stokes differential filters (cf. Algorithm 3.2 and Algorithm 3.4 describe Eqs. (1.2) and (1.6), respectively) for reducing the unpredictable error in the solution of u at each time-step,
- to propose a decoupling scheme for $\bar{\xi}$ and λ from the system of two iterative Stokes differential filters using the CSS (cf. Eqs. (1.2) and (1.6)), and
- to shed some light on the numerical study of the Leray-alpha model for NC and MC problems (cf. Algorithm 4.1).

We remark that there are other choices of solvers for the NC and MC problems, see for example [3] and [4].

This paper contains six sections. In this first section, the current research in the field of regularization methods of fluid flows is introduced and the motivation/feature for using the proposed method in the thermal convection problem are stated. Next, a two-dimensional transient incompressible NC problem in the presence of the point-source function is introduced in Section 2. In Section 3, the computational algorithm of the Leray-alpha model is shown and the key features of two kinds of iterative regularization techniques are stated. In Section 4, implementations of the numerical procedures of each algorithm are given. A list of worked examples are employed to demonstrate and discuss the results of the proposed algorithms in Section 5. Finally, the overall contribution of this work to the field of iterative regularization methods on NC and MC flow problems is discussed in the Conclusion.

2 Problem description

Let Ω be a bounded domain in the Euclidean space \mathbb{R}^2 with a piecewise smooth boundary $\partial\Omega$. A fixed final time is denoted by t_f . Let $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$, where $\partial\Omega$ consists of two piecewise smooth boundaries. The aim of the following problem is to study the fluid flow structure of the velocity, the pressure and the temperature fields driven by a *specified* strength of a time-varying heat source $G(t)$ in $(0, t_f]$. The governing equations

for the two-dimensional time-dependent NC problem, assuming air to be of constant thermo-physical properties with the Boussinesq approximation considered valid, are the Navier-Stokes equation (x - and y - momentum equations), the continuity equation (or the incompressibility constraint) and the equation of energy. These equations are described by the PV formulation and are normalized using the appropriate non-dimensional parameters, and turn out as follows [20]:

$$\left\{ \begin{array}{ll} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + Pr \nabla^2 \mathbf{u} + Ra Pr T \mathbf{j}, & \text{in } \Omega \times (0, t_f], \\ \nabla \cdot \mathbf{u} = 0, & \text{in } \Omega \times (0, t_f], \\ \mathbf{u} = \mathbf{0}, & \text{on } \partial\Omega \times (0, t_f], \\ \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = \nabla^2 T + G(t) \delta_k(x - x^\dagger) \delta_k(y - y^\dagger), & \text{in } \Omega \times (0, t_f], \\ T|_{\partial\Omega_1} = 0 \text{ and } \frac{\partial T}{\partial n} \Big|_{\partial\Omega_2} = 0, & \text{on } \partial\Omega \times (0, t_f], \end{array} \right. \quad (2.1)$$

with the initial conditions

$$\mathbf{u}(x, t=0) = \mathbf{0} \quad \text{and} \quad T(x, t=0) = 0, \quad \text{in } \Omega, \quad (2.2)$$

where $\mathbf{u}(x, t) = (u, v)$ is the velocity field, $p(x, t)$ is the pressure scalar field, $T(x, t)$ is the temperature scalar field, and (x^\dagger, y^\dagger) is the location of the point source.

The following non-dimensional variables are defined:

$$\begin{aligned} x &= \frac{x'}{d_x}, \quad y = \frac{y'}{d_y}, \quad t = \frac{\kappa t'}{d_y^2}, \quad \mathbf{u} = \frac{d_y \mathbf{u}'}{\kappa}, \quad T = \frac{T' - T'_{\text{cold}}}{T'_{\text{hot}} - T'_{\text{cold}}}, \\ p &= p^* - (T'_{\text{cold}} - T'_{\text{ref}}) \frac{d_y^3}{\kappa^2} \alpha^* g y, \quad p^* = \frac{d_y^2 p'}{\rho \kappa^2}, \end{aligned}$$

and $T'_{\text{ref}} = (T'_{\text{hot}} + T'_{\text{cold}})/2$ is the average temperature of the system, α^* is the thermal expansion coefficient, $Pr = \nu/\kappa$ is the Prandtl number, and $Ra = \beta_T g (T'_{\text{hot}} - T'_{\text{cold}}) d_y^3 / \kappa \nu$ is the Rayleigh number, where β_T is the thermal expansion coefficient, g is the gravitation constant, κ is the thermal diffusivity, ρ is the density, ν is the kinematic viscosity, \mathbf{j} is the unit vector in the y -direction, T'_{hot} is the nominal bottom temperature, T'_{cold} is the dimensional temperature at the top boundary, d_x is half of the width and d_y is half of the depth of the cavity. A two-dimensional square cavity is considered for the present study with the physical dimensions given in Fig. 1.

The dimensionless strength of the heat source $G(t)$ is related to the dimensional strength $G'(t')$ as follows:

$$G(t) = \frac{G'(t') d_y}{(T'_{\text{hot}} - T'_{\text{cold}}) \tilde{k} d_x^3}$$

where \tilde{k} is thermal conductivity. The characteristic temperature of the system T'_{hot} is related to the characteristic magnitude of the dimensional heat source G'_{ref} by the following

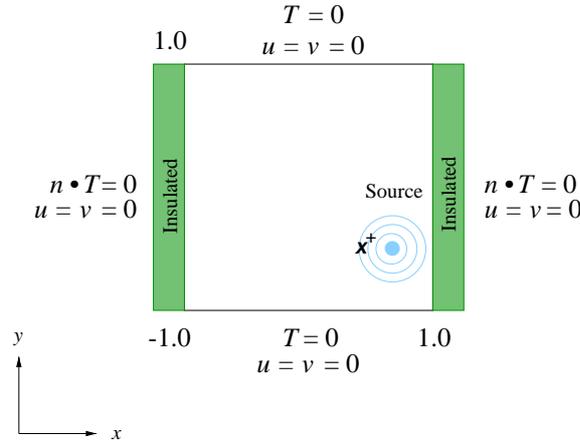


Figure 1: Geometry, coordinates and a source location.

connection:

$$T'_{\text{hot}} - T'_{\text{cold}} = \frac{G'_{\text{ref}}}{\tilde{k}}.$$

Combining Ra and the above result gives

$$Ra = a^* g \frac{d_y^3 G'_{\text{ref}}}{\kappa \nu \tilde{k}}.$$

The function $\delta_k(x - x^+)$, for example, approximates the point source in the cavity and becomes the Dirac delta function as k tends to infinity, and is defined by

$$\delta_k(x - x^+) = \frac{k}{2 \cosh^2(k(x - x^+))}.$$

Then, $\delta_k(y - y^+)$ can be defined similar to $\delta_k(x - x^+)$. In this work, in order to avoid having the point-source distribution become too spiky, $k = 20$ is used.

In what follows, two different types of regularization methods to control the irregular and unstable fluid flows are revisited and proposed.

3 Mathematical formulation

It was Leray in 1934 [21] who introduced the regularization technique of incompressible Newtonian fluid flow that can be described as follows:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla) \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \quad (3.1)$$

where \bar{u} denotes convolution with a Gaussian function with a filter radius $\alpha > 0$. The idea of the Leray-alpha model is to replace convolution with a Gaussian function by the so-called differential filter (or the alpha filter):

$$-\alpha^2 \nabla^2 \bar{u} + \bar{u} = u. \quad (3.2)$$

Its computational advantages were discussed by [7].

One of the main advantages of the differential filter is to filter out unwanted noise from u . However, after filtering, the divergence free condition will be violated, see [1] and [22] for more detailed discussions. Adding the filtered pressure gradient $\nabla \lambda$ as a Lagrangian multiplier allows us to enforce the incompressibility constraint $\nabla \cdot u = 0$. The Leray-alpha model can be formulated as follows:

$$\begin{cases} \frac{\partial u}{\partial t} + (\bar{u} \cdot \nabla) u = -\nabla p + \nu \nabla^2 u + f, \\ \nabla \cdot u = 0, \\ -\alpha^2 \nabla^2 \bar{u} + \bar{u} + \nabla \lambda = u, \\ \nabla \cdot \bar{u} = 0. \end{cases} \quad (3.3)$$

When converting this idea into the numerical solutions of the thermal convection flow problems, the first step in the solution procedure is to discretise the above governing equations, applying conventional differencing to the partial derivative terms. At each time, each PV is determined using the four-step strategy. For time stepping, the second-order backward difference method is used, while all nonlinear terms are treated using a semi-implicit approach and a linear extrapolation involving the pressure and the temperature in the momentum equations, i.e.,

$$w^{n,*} := \begin{cases} w^n, & n = 1, \\ 2w^n - w^{n-1}, & n \geq 2, \end{cases} \quad (3.4)$$

where $w = \{u, p, T\}$. The decoupling scheme of the velocity and the pressure from the momentum equations (cf. Eqs. (3.5) and (3.10)) and the filtered velocity and the filtered pressure from the Stokes differential filter (cf. Eqs. (3.6) and (3.11)) based on the CSS is used in order to reduce the (filtered) pressure boundary layer effect stemming from the (filtered) pressure Poisson equation.

In general form, this can be expressed as:

Momentum Equations

Calculation of velocities

Evolution step

$$\begin{cases} \frac{3u^{n+1}}{2\Delta t} + (\bar{u}^{n,*} \cdot \nabla) u^{n+1} \\ = \frac{4\bar{u}^n - \bar{u}^{n-1}}{2\Delta t} - \nabla \bar{p}^{n,*} + Pr \nabla^2 \bar{u}^{n+1} + Pr Ra \bar{T}^{n,*} j, & \text{in } \Omega \times (0, t_f], \\ u^{n+1}|_{\partial\Omega} = 0, & \text{on } \partial\Omega \times (0, t_f]. \end{cases} \quad (3.5)$$

Filtration step

$$\begin{cases} -\alpha^2 \nabla^2 \widetilde{\mathbf{u}}^{n+1} + \widetilde{\mathbf{u}}^{n+1} + \nabla \widetilde{\lambda}^n = \mathbf{u}^{n+1}, & \text{in } \Omega \times (0, t_f], \\ \widetilde{\mathbf{u}}^{n+1}|_{\partial\Omega} = \mathbf{0}, & \text{on } \partial\Omega \times (0, t_f], \\ (\nabla \widetilde{\psi}^{n+1}, \nabla q) = (\mathbf{u}^{n+1} - \widetilde{\mathbf{u}}^{n+1}, \nabla q), \quad \forall q \in H^1(\Omega), \\ \widetilde{\lambda}^{n+1} = \widetilde{\psi}^{n+1} + \widetilde{\lambda}^n - \alpha^2 \nabla \cdot \widetilde{\mathbf{u}}^{n+1}. \end{cases} \quad (3.6)$$

Deconvolution step

$$\begin{cases} -\alpha^2 \nabla^2 \widetilde{\widetilde{\mathbf{u}}}^{n+1} + \widetilde{\widetilde{\mathbf{u}}}^{n+1} + \nabla \widetilde{\widetilde{\lambda}}^n = \widetilde{\widetilde{\mathbf{u}}}^{n+1}, & \text{in } \Omega \times (0, t_f], \\ \widetilde{\widetilde{\mathbf{u}}}^{n+1}|_{\partial\Omega} = \mathbf{0}, & \text{on } \partial\Omega \times (0, t_f], \\ (\nabla \widetilde{\widetilde{\psi}}^{n+1}, \nabla q) = (\widetilde{\widetilde{\mathbf{u}}}^{n+1} - \widetilde{\mathbf{u}}^{n+1}, \nabla q), \quad \forall q \in H^1(\Omega), \\ \widetilde{\widetilde{\lambda}}^{n+1} = \widetilde{\widetilde{\psi}}^{n+1} + \widetilde{\widetilde{\lambda}}^n - \alpha^2 \nabla \cdot \widetilde{\widetilde{\mathbf{u}}}^{n+1}, \end{cases} \quad (3.7)$$

and

$$D(\widetilde{\widetilde{\mathbf{u}}}^{n+1}) := 2\widetilde{\widetilde{\mathbf{u}}}^{n+1} - \widetilde{\widetilde{\mathbf{u}}}^{n+1}. \quad (3.8)$$

Relaxation step

$$\overline{\mathbf{u}}^{n+1} = (1 - \chi)\mathbf{u}^{n+1} + \chi D(\widetilde{\widetilde{\mathbf{u}}}^{n+1}), \quad (3.9)$$

where χ is a time relaxation parameter.

Calculation of pressure

Evolution step

$$\begin{cases} (\nabla \phi^{n+1}, \nabla q) = \left(\frac{3\overline{\mathbf{u}}^{n+1} - 4\overline{\mathbf{u}}^n + \overline{\mathbf{u}}^{n-1}}{2\Delta t}, \nabla q \right), \quad \forall q \in H^1(\Omega), \\ p^{n+1} = \phi^{n+1} + \overline{p}^{n,*} - Pr \nabla \cdot \overline{\mathbf{u}}^{n+1}. \end{cases} \quad (3.10)$$

Filtration step

$$\begin{cases} -\alpha^2 \nabla^2 \widetilde{p}^{n+1} + \widetilde{p}^{n+1} = p^{n+1}, & \text{in } \Omega \times (0, t_f], \\ \frac{\partial \widetilde{p}^{n+1}}{\partial \mathbf{n}}|_{\partial\Omega} = 0, & \text{on } \partial\Omega \times (0, t_f]. \end{cases} \quad (3.11)$$

Deconvolution step

$$\begin{cases} -\alpha^2 \nabla^2 \widetilde{p^{n+1}} + \widetilde{p^{n+1}} = \widetilde{p^{n+1}}, & \text{in } \Omega \times (0, t_f], \\ \frac{\partial \widetilde{p^{n+1}}}{\partial \mathbf{n}} \Big|_{\partial \Omega} = 0, & \text{on } \partial \Omega \times (0, t_f], \end{cases} \tag{3.12}$$

and

$$D(\widetilde{p^{n+1}}) := 2\widetilde{p^{n+1}} - \widetilde{\widetilde{p^{n+1}}}. \tag{3.13}$$

Relaxation step

$$\overline{p^{n+1}} = (1 - \chi) p^{n+1} + \chi D(\widetilde{p^{n+1}}). \tag{3.14}$$

Energy Equation

Evolution step

$$\begin{cases} \frac{3T^{n+1}}{2\Delta t} + (\overline{\mathbf{u}^{n+1}} \cdot \nabla) T^{n+1} = \frac{4\overline{T^n} - \overline{T^{n-1}}}{2\Delta t} + \nabla^2 T^{n+1}, & \text{in } \Omega \times (0, t_f], \\ T^{n+1} \Big|_{\partial \Omega_1} = 0 \text{ and } \frac{\partial T^{n+1}}{\partial x} \Big|_{\partial \Omega_2} = 0, & \text{on } \partial \Omega \times (0, t_f]. \end{cases} \tag{3.15}$$

Filtration step

$$\begin{cases} -\alpha^2 \nabla^2 \widetilde{T^{n+1}} + \widetilde{T^{n+1}} = T^{n+1}, & \text{in } \Omega \times (0, t_f], \\ \widetilde{T^{n+1}} \Big|_{\partial \Omega_1} = 0 \text{ and } \frac{\partial \widetilde{T^{n+1}}}{\partial x} \Big|_{\partial \Omega_2} = 0, & \text{on } \partial \Omega \times (0, t_f]. \end{cases} \tag{3.16}$$

Deconvolution step

$$\begin{cases} -\alpha^2 \nabla^2 \widetilde{\widetilde{T^{n+1}}} + \widetilde{\widetilde{T^{n+1}}} = T^{n+1}, & \text{in } \Omega \times (0, t_f], \\ \widetilde{\widetilde{T^{n+1}}} \Big|_{\partial \Omega_1} = 0 \text{ and } \frac{\partial \widetilde{\widetilde{T^{n+1}}}}{\partial x} \Big|_{\partial \Omega_2} = 0, & \text{on } \partial \Omega \times (0, t_f], \end{cases} \tag{3.17}$$

and

$$D(\widetilde{\widetilde{T^{n+1}}}) := 2\widetilde{\widetilde{T^{n+1}}} - \widetilde{\widetilde{\widetilde{T^{n+1}}}}. \tag{3.18}$$

Relaxation step

$$\overline{T^{n+1}} = (1 - \chi) T^{n+1} + \chi D(\widetilde{\widetilde{T^{n+1}}}). \tag{3.19}$$

The above three sets of four-step computations works efficiently when there is no random noise involved at each time. Now we assume \mathbf{u}^{n+1} in Eq. (3.5), p^{n+1} in Eq. (3.10) and assume T^{n+1} in Eq. (3.15) contain unpredictable random noise. We then observed the following:

- Not only is the pressure being filtered (cf. Eq. (3.11)), but the temperature is also being filtered (cf. Eq. (3.16)). Hence, using the differential filter to get the smooth pressure and temperature solutions is an indispensable step towards solving the NC flow problems.
- Owing to the one-step iteration, the relaxation step in Eqs. (3.8)-(3.9), Eqs. (3.13)-(3.14) and Eqs. (3.18)-(3.19) also contain noise.

Putting the filtration step and the deconvolution step together to create an iterative process dampens spurious oscillations at each iteration. Algorithm 3.1 and Algorithm 3.2 describe two iterative algorithms for linear and Stokes differential filters, respectively. Here, a few notes are:

- Eqs. (3.6)-(3.7), Eqs. (3.11)-(3.12) and Eqs. (3.16)-(3.17) are formulated as iterative algorithms.
- Computations to Eqs. (3.8) and (3.13)/(3.18) are replaced by Eqs. (4.8) and (4.13), respectively (see Algorithm 4.2 and Algorithm 4.3).
- In these two algorithms, the stopping criterion depends on the desired value of N .
- Two DO-loops in Algorithm 3.2 are considered; an inner loop controls the discrete Stokes differential filter for each j -th iteration and the convergence of (ξ_j, λ_k) is controlled by the choice of K , while an outer loop that is controlled by the choice of N performs a smoothing effect in order to obtain filtered velocity solutions.

Algorithm 3.3 and Algorithm 3.4 describe two iterative algorithms for nonlinear TV and TV Stokes differential filters, respectively. Here, a few notes are:

- The nonlinear diffusion filter term is treated as a semi-implicit form.
- The unfiltered variable ξ_0 with the noise remains unchanged at each time-step. At each iteration, the approximate profile of $\xi_{j,h}$ will be recovered in spite of the corruption term $\xi_{0,h}$.
- To distinguish between Algorithm 3.2 and Algorithm 3.4, the auxiliary pressure Poisson equation in Algorithm 3.4 is

$$(\nabla \psi_k, \nabla q) = (\xi_0 - \xi_j, \nabla q), \quad \forall q \in H^1(\Omega)$$

instead of

$$(\nabla \psi_k, \nabla q) = (\xi_{j-1} - \xi_j, \nabla q), \quad \forall q \in H^1(\Omega).$$

Algorithm 3.1: Solution of the algorithm for the iterative differential filter.

Set $\tilde{\zeta}_0 := \overline{\zeta^{n+1}}$;
 DO $j=1, \dots, N$

$$-\alpha^2 \nabla^2 \tilde{\zeta}_j + \tilde{\zeta}_j = \tilde{\zeta}_{j-1},$$

$$\tilde{\zeta}_j|_{\partial\Omega_1} = 0 \quad \text{and} \quad \frac{\partial \tilde{\zeta}_j}{\partial x}|_{\partial\Omega_2} = 0 \quad \text{or} \quad \frac{\partial \tilde{\zeta}_j}{\partial \mathbf{n}}|_{\partial\Omega} = 0;$$

END DO
 Update $\tilde{\zeta}^{n+1} \leftarrow \tilde{\zeta}_N$.

Algorithm 3.2: Solution of the algorithm for the iterative Stokes differential filter.

Set $\tilde{\boldsymbol{\zeta}}_0 := \overline{\mathbf{u}^{n+1}}$;
 DO $j=1, \dots, N$
 Set $\lambda_{k-1} := 0$
 DO $k=1, \dots, K$

$$-\alpha^2 \nabla^2 \tilde{\boldsymbol{\zeta}}_j + \tilde{\boldsymbol{\zeta}}_j + \nabla \lambda_{k-1} = \tilde{\boldsymbol{\zeta}}_{j-1},$$

$$\tilde{\boldsymbol{\zeta}}_j|_{\partial\Omega} = \mathbf{0},$$

$$(\nabla \psi_k, \nabla q) = (\tilde{\boldsymbol{\zeta}}_{j-1} - \tilde{\boldsymbol{\zeta}}_j, \nabla q), \quad \forall q \in H^1(\Omega),$$

$$\lambda_k = \psi_k + \lambda_{k-1} - \alpha^2 \nabla \cdot \tilde{\boldsymbol{\zeta}}_j;$$

END DO
 END DO
 Update $\mathbf{u}^{n+1} \leftarrow \tilde{\boldsymbol{\zeta}}_N$.

Note that Algorithm 3.1 is used for the temperature and pressure while Algorithms 3.2 is used for the velocity. Moreover, Algorithms 3.1-3.2 are used when the solutions are smooth enough. When the solutions contain sharp gradients, we have to replace Algorithms 3.1-3.2 by Algorithms 3.3-3.4 respectively, which are TV-based methods designed to capture the sharp gradients.

4 Numerical procedure

A complete description of the CSS to solve the thermal convection problems can be found in [19]. In the following, we will provide a brief account of the numerical procedure. Note that it is proved in [18] that the CSS is unconditionally stable.

Algorithm 3.3: Solution of the algorithm for the iterative TV differential filter.

Set $\xi_0 := \overline{\xi^{n+1}}$;
 DO $j=1, \dots, N$

$$-\alpha^2 \nabla^2 \xi_j - \alpha_{\text{TV}}^2 \nabla \cdot \left(\frac{1}{\|\xi_{j-1}\|_{\text{TV}}} \nabla \xi_j \right) + \xi_j = \xi_0,$$

$$\xi_j|_{\partial\Omega_1} = 0 \text{ and } \left(\alpha^2 \frac{\partial \xi_j}{\partial x} + \alpha_{\text{TV}}^2 \frac{1}{\|\xi_{j-1}\|_{\text{TV}}} \frac{\partial \xi_j}{\partial x} \right) \Big|_{\partial\Omega_2} = 0 \text{ or } \left(\alpha^2 \frac{\partial \xi_j}{\partial n} + \alpha_{\text{TV}}^2 \frac{1}{\|\xi_{j-1}\|_{\text{TV}}} \frac{\partial \xi_j}{\partial n} \right) \Big|_{\partial\Omega} = 0;$$

END DO
 Update $\xi^{n+1} \leftarrow \xi_N$.

Algorithm 3.4: Solution of the algorithm for the iterative TV Stokes differential filter.

Set $\xi_0 := \overline{u^{n+1}}$
 DO $j=1, \dots, N$
 Set $\lambda_{k-1} := 0$
 DO $k=1, \dots, K$

$$-\alpha^2 \nabla^2 \xi_j - \alpha_{\text{TV}}^2 \nabla \cdot \left(\frac{1}{\|\xi_{j-1}\|_{\text{TV}}} \nabla \xi_j \right) + \xi_j + \nabla \lambda_{k-1} = \xi_0,$$

$$\xi_j|_{\partial\Omega} = \mathbf{0},$$

$$(\nabla \psi_k, \nabla q) = (\xi_0 - \xi_j, \nabla q), \quad \forall q \in H^1(\Omega),$$

$$\lambda_k = \psi_k + \lambda_{k-1} - \alpha^2 \nabla \cdot \xi_j;$$

 END DO
 END DO
 Update $u^{n+1} \leftarrow \xi_N$.

4.1 Mixed FE solver

The governing PDEs in the PV formulation are approximated using the mixed FE approach. The space of square integrable functions on Ω :

$$L^2(\Omega) := \left\{ q \mid \int_{\Omega} |q|^2 d\Omega < \infty \right\}$$

and the subspace

$$H^1(\Omega) := \left\{ \phi \mid \phi \in L^2(\Omega), \text{grad} \in (L^2(\Omega))^{\text{nd}} \right\}$$

of $L^2(\Omega)$ are defined, where nd is the space dimension of the problem. Let us introduce a regular triangulation of Ω , with a finite number of triangles \mathcal{K}_l , $l = 1, \dots, E_h$, where E_h

stands for the total number of triangles.

The spatial discretization of systems (cf. Eqs. (3.5), (3.10), (3.15), Algorithms 3.1-3.4) replaces the (infinite-dimensional) solution function spaces X , Q and M by finite-dimensional subspaces X_h , Q_h and M_h . We define X_h by

$$X_h = \{ \phi_h \in C^0(\Omega) : \phi_h|_{\mathcal{K}_l} \in \mathbb{P}_2, \forall l = 1, \dots, E_h \},$$

and then define Q_h by

$$Q_h = \{ q_h \in C^0(\Omega) : q_h|_{\mathcal{K}_l} \in \mathbb{P}_1, \forall l = 1, \dots, E_h \}$$

and finally define M_h by

$$M_h = \{ m_h \in C^0(\Omega) : m_h|_{\mathcal{K}_l} \in \mathbb{P}_1, \forall l = 1, \dots, E_h \}.$$

The space \mathbb{P}_r denotes the set of polynomials of degree r in each \mathcal{K}_l , i.e., for $r \geq 0$:

$$\mathbb{P}_r = \left\{ w : \mathcal{K}_l \rightarrow \mathbb{R}, w(x, y) = \sum_{0 \leq i+j \leq r} \alpha_{ij} x^i y^j \right\}.$$

The Hood-Taylor $\mathbb{P}_2/\mathbb{P}_1$ FE pair is used for velocity/auxiliary pressure-pressure, while the \mathbb{P}_2 FE is used for the temperature.

The FE discretisation of the Leray-alpha scheme follows a classical approach: a weak form of Eqs. (3.5), (3.10), (3.15), Algorithms 3.1-3.4 is written and FE approximation subspace $\mathbf{X}_{0,h} \subset \mathbf{X} = (H^1(\Omega))^2$, $Q_h \subset Q = H^1(\Omega)$, $M_h \subset M = L^2(\Omega)$ and $\widehat{X}_{0,n,\nabla T=0,h} \subset \widehat{X} = H^1(\Omega)$ are introduced for the velocities, auxiliary pressure, and temperature, respectively. The discrete NC problem then reduces to obtain $(\mathbf{u}_h, \phi_h, p_h, T_h) \in \mathbf{X}_{0,h} \times Q_h \times M_h \times \widehat{X}_{0,n,\nabla T=0,h}$, as described in Algorithm 4.1. We remark that there are other efficient solvers for the above filters, see for example [5, 27, 28] and [32].

4.2 Features of Algorithm 4.1 as a discrete Leray-alpha scheme

At each time-step, the main features of Algorithm 4.1 can be described as follows:

- Steps 2, 4 and 7 are simply the formulation of CSS.
- Step 3 adopts the weak form of the iterative Stokes differential filter (Algorithm 4.2) for making smoother velocity solutions, while Steps 6 and 8 operate the weak form of the iterative differential filter (Algorithm 4.3) for smoothing the temperature and the pressure solutions.
- In Algorithms 4.2 and 4.3, the relaxation step is added into the DO-loop, as well as a linear interpolation of filtered solutions at the N^{th} iteration and the $(N+1)^{\text{th}}$ iteration.
- Algorithms 4.2 and 4.3 are the finite element discretizations of Algorithms 3.1 and 3.2.

Algorithm 4.1: Discrete Leray-alpha scheme.

1. Given $\mathbf{u}_h^n, \mathbf{u}_h^{n-1}, p_h^n, p_h^{n-1}, T_h^n$ and T_h^{n-1} .
2. Solve for $\overline{\mathbf{u}_h^{n+1}} \in \mathbf{X}_{0,h}$ such that for all $\boldsymbol{\phi}_h \in \mathbf{X}_h$:

$$\begin{aligned} & \frac{3}{2\Delta t} (\overline{\mathbf{u}_h^{n+1}}, \boldsymbol{\phi}_h) + \left(\left[(\mathbf{u}_h^{n,*} \cdot \nabla) \overline{\mathbf{u}_h^{n+1}} + \frac{1}{2} (\nabla \cdot \mathbf{u}_h^{n,*}) \overline{\mathbf{u}_h^{n+1}} \right], \boldsymbol{\phi}_h \right) + Pr (\nabla \overline{\mathbf{u}_h^{n+1}}, \nabla \boldsymbol{\phi}_h) \\ &= \frac{2}{\Delta t} (\mathbf{u}_h^n, \boldsymbol{\phi}_h) + \frac{1}{2\Delta t} (\mathbf{u}_h^{n-1}, \boldsymbol{\phi}_h) + (p_h^{n,*}, \nabla \boldsymbol{\phi}_h) + Pr Ra (T_h^{n,*}, \boldsymbol{\phi}_h j). \end{aligned} \tag{4.1}$$

3. Use Algorithm 4.2 or Algorithm 4.4 to get $\overline{\mathbf{u}_h^{n+1}}$ from $\overline{\mathbf{u}_h^{n+1}}$.
4. Solve for $\varphi_h^{n+1} \in Q_h$ such that for all $q_h \in Q_h$:

$$(\nabla \varphi_h^{n+1}, \nabla q_h) = \left(\frac{3\overline{\mathbf{u}_h^{n+1}} - 4\mathbf{u}_h^n + \mathbf{u}_h^{n-1}}{2\Delta t}, \nabla q_h \right). \tag{4.2}$$

5. Solve for $\overline{p_h^{n+1}} \in M_h$ such that for all $m_h \in M_h$:

$$(\overline{p_h^{n+1}}, m_h) = \left(\left[\varphi_h^{n+1} + p_h^{n,*} - \frac{1}{Re} \nabla \cdot \overline{\mathbf{u}_h^{n+1}} \right], m_h \right). \tag{4.3}$$

6. Use Algorithm 4.3 or Algorithm 4.5 to get p_h^{n+1} from $\overline{p_h^{n+1}}$.
7. Solve for $\overline{T_h^{n+1}} \in \widehat{X}_{0,n,\nabla T=0,h}$ such that for all $\phi_h \in X_h$:

$$\frac{3}{2\Delta t} (\overline{T_h^{n+1}}, \phi_h) + (\nabla \overline{T_h^{n+1}}, \nabla \phi_h) = \frac{2}{\Delta t} (T_h^n, \phi_h) - \frac{1}{2\Delta t} (T_h^{n-1}, \phi_h). \tag{4.4}$$

8. Use Algorithm 4.3 or Algorithm 4.5 to get T_h^{n+1} from $\overline{T_h^{n+1}}$.

4.3 The TV regularization method

In Step 3 and Step 6/8 of Algorithm 4.1, we can replace the differential filters (Algorithms 4.2-4.3) by the TV-based filters (described in the form of differential equations in Algorithms 3.3-3.4) in order to capture the sharp gradients in the solutions, which is the main reason for this paper. In the following, we will describe the numerical implementation of these filters.

- By making use of the anisotropic filter (cf. Eq. (1.4)), in Algorithm 4.2, Eqs. (4.5) and (4.6) change to

$$\begin{aligned} & \alpha^2 (\nabla \boldsymbol{\xi}_{j,h}, \nabla \boldsymbol{\phi}_h) + \alpha_{TV}^2 \left(\frac{1}{\|\boldsymbol{\xi}_{j-1,h}\|_{TV}} \nabla \boldsymbol{\xi}_{j,h}, \nabla \boldsymbol{\phi}_h \right) + (\boldsymbol{\xi}_{j,h}, \boldsymbol{\phi}_h) \\ &= (\lambda_{k-1,h}, \nabla \cdot \boldsymbol{\phi}_h) + (\boldsymbol{\xi}_{0,h}, \boldsymbol{\phi}_h), \end{aligned}$$

Algorithm 4.2: Iterative Stokes differential filter.

-
1. Given $\xi_{0,h} = \overline{u_h^{n+1}} \in X_{0,h}$
 2. DO $j=1, \dots, N$,
 Given $\lambda_{0,h} = 0 \in Q_h$

i. DO $k=1, \dots, K$

a. Solve for $\xi_{j,h} \in X_{0,h}$ such that for all $\phi_h \in X_h$:

$$\alpha^2 (\nabla \xi_{j,h}, \nabla \phi_h) + (\xi_{j,h}, \phi_h) = (\lambda_{k-1,h}, \nabla \cdot \phi_h) + (\xi_{j-1,h}, \phi_h). \quad (4.5)$$

b. Solve for $\psi_{k,h} \in Q_h$ such that for all $q_h \in Q_h$:

$$(\nabla \psi_{k,h}, \nabla q_h) = (\xi_{j-1,h} - \xi_{j,h}, \nabla q_h). \quad (4.6)$$

c. Solve for $\lambda_{k,h} \in M_h$ such that for all $m_h \in M_h$:

$$(\lambda_{k,h}, m_h) = (\psi_{k,h} + \lambda_{k-1,h} - \alpha^2 \nabla \cdot \xi_{j,h}, m_h). \quad (4.7)$$

END

ii. Update

$$\xi_{j,h} = (1 - \chi) \xi_{j-1,h} + \chi \xi_{j,h}. \quad (4.8)$$

END

3. Update

$$u_h^{n+1} \leftarrow \xi_{N,h}. \quad (4.9)$$

and

$$(\nabla \psi_{k,h}, \nabla q_h) = (\xi_{0,h} - \xi_{j,h}, \nabla q_h).$$

We refer to these modifications as Algorithm 4.4.

- Likewise, in Algorithm 4.3, for instance, Eq. (4.10) changes to

$$\alpha^2 (\nabla \xi_{j,h}, \nabla \phi_h) + \alpha_{\text{TV}}^2 \left(\frac{1}{\|\xi_{j-1,h}\|_{\text{TV}}} \nabla \xi_{j,h}, \nabla \phi_h \right) + (\xi_{j,h}, \phi_h) = (\xi_{0,h}, \phi_h)$$

with

$$\xi_j|_{\partial\Omega_1} = 0 \quad \text{and} \quad \left(\alpha^2 \frac{\partial \xi_j}{\partial x} + \alpha_{\text{TV}}^2 \frac{1}{\|\xi_{j-1,h}\|_{\text{TV}}} \frac{\partial \xi_j}{\partial x} \right) \Big|_{\partial\Omega_2} = 0$$

or

$$\left(\alpha^2 \frac{\partial \xi_j}{\partial \mathbf{n}} + \alpha_{\text{TV}}^2 \frac{1}{\|\xi_{j-1,h}\|_{\text{TV}}} \frac{\partial \xi_j}{\partial \mathbf{n}} \right) \Big|_{\partial\Omega} = 0.$$

Algorithm 4.3: Iterative differential filter.

-
1. Given $\zeta_{0,h} = \overline{T_h^{n+1}} \in \widehat{X}_{0,n}$ or $\zeta_{0,h} = \overline{p_h^{n+1}} \in M_h$
 2. DO $j=1, \dots, N$,

- Solve for $\zeta_{j,h} \in \widehat{X}_{0,n, \nabla \zeta=0,h}$ such that for all $\phi_h \in X_h$:

$$\alpha^2(\nabla \zeta_{j,h}, \nabla \phi_h) + (\zeta_{j,h}, \phi_h) = (\zeta_{j-1,h}, \phi_h). \quad (4.10)$$

Or

- Solve for $\zeta_{j+1,h} \in M_h$ such that for all $m_h \in M_h$:

$$\alpha^2(\nabla \zeta_{j,h}, \nabla \phi_h) + (\zeta_{j,h}, m_h) = (\zeta_{j-1,h}, m_h). \quad (4.11)$$

- Update

$$\zeta_{j,h} = (1-\chi)\zeta_{j-1,h} + \chi\zeta_{j,h}. \quad (4.12)$$

END

3. Update

$$T_h^{n+1} \leftarrow \zeta_{N,h} \quad (4.13)$$

or update

$$p_h^{n+1} \leftarrow \zeta_{N,h}. \quad (4.14)$$

We refer to this modification as Algorithm 4.5. To compute the TV-norm, let us define

$$\zeta_{x,h} = \frac{\partial \zeta_{j-1,h}}{\partial x} \quad \text{and} \quad \zeta_{y,h} = \frac{\partial \zeta_{j-1,h}}{\partial y}.$$

Now, at each iteration, depending on the choice of interpolation spaces for each PV, for instance, we solve for $\zeta_{x,h}, \zeta_{y,h} \in X_h$ such that for all $\phi_h \in X_h$:

$$\begin{cases} (\zeta_{x,h}, \phi_h) = \left(\frac{\partial \zeta_{j-1,h}}{\partial x}, \phi_h \right), \\ (\zeta_{y,h}, \phi_h) = \left(\frac{\partial \zeta_{j-1,h}}{\partial y}, \phi_h \right). \end{cases} \quad (4.15)$$

5 Computational results

In this section, noise-removal for NC and MC problems are studied using iterative regularization and TV based algorithms. The motivation for studying this problem is that it acts as a model problem for some important applications arising in atmospheric sciences.

Atmospheric images are obtained by experiments and they inevitably contain noise due to experimental errors. An important step is removing the noise from these images, see for example [16]. By using our proposed algorithms, we are able to removing the noise from the images and at the same time systematically incorporate the underlying physical principles into considerations.

5.1 Natural-convection problems

Suppose the approximate PV solutions at each time-step are added by the noise data. In other words, the random errors are added to the approximate solution. It can be shown in the following equation:

$$\zeta_{i,j}^{\text{noise}} = \zeta_{i,j}^{\text{approx}} + \omega_{i,j} \hat{\sigma} \max |\zeta_{i,j}^{\text{approx}}|, \quad (5.1)$$

where $\zeta_{i,j}^{\text{approx}} = \{u_h, v_h, p_h, T_h\}$ and the subscripts i and j are the grid number of the spatial-coordinate and temporal-coordinate, respectively. The variables $\zeta_{i,j}^{\text{noise}}$ and $\zeta_{i,j}^{\text{approx}}$ in Eq. (5.1) are the corrupted and approximate PV solutions, respectively. Furthermore, $\hat{\sigma} = \{\hat{\sigma}_u, \hat{\sigma}_p, \hat{\sigma}_h\}$ and $\omega_{i,j}$ are the standard deviation of random (measurement) errors and the random number, respectively; these values are fixed at each time-step. In what follows, we drop the i and j superscripts. It is worth mentioning that the errors are accumulated at each time-step computation even though the filtering process is applied.

Here we consider a case where $Pr = 0.71$ and $Ra = 1000$ at $t_f = 1$. The uniform grid used for the present calculation is 129^2 for the velocities and the temperature, and 65^2 for the pressure and the auxiliary pressure, with the time-step size $\Delta t = 0.025$. The relaxation parameter is fixed to $\chi = \Delta t$ and the regularized parameter is fixed to $\epsilon = 10^{-6}$. The fluid flows are triggered by the following choices of source functions that are used as follows:

Problem Set 1: A heat source is located at $(0.75, 0.75)$:

- a sinusoidal curve-like function;

$$G(t) = -\frac{750}{4} t \left(t - \frac{1}{3} \right)^2 (t-1) + 1.0, \quad t \in [0, 1].$$

Problem Set 2: Two heat sources are located at $(0.75, 0.75)$ and $(-0.75, -0.75)$:

- a triangular ramping function;

$$G_1(t) = \begin{cases} \frac{4}{0.3} t + 1.0, & 0 \leq t \leq 0.3, \\ -\frac{2}{0.3} t + 7.0, & 0.3 \leq t \leq 0.6, \\ 3.0, & 0.6 \leq t \leq 1. \end{cases} \quad (5.2)$$

- a single step function;

$$G_2(t) = \begin{cases} 2.0, & 0 \leq t \leq 0.3, \\ 5.0, & 0.3 \leq t \leq 0.7, \\ 2.0, & 0.7 \leq t \leq 1. \end{cases} \quad (5.3)$$

Numerical results are summarized as follows:

- Problem set 1: Three random noise levels $\hat{\sigma} = 0.1$, $\hat{\sigma} = 0.05$ and $\hat{\sigma} = 0.01$ are adopted. We fixed the values of $N = K = 10$ and $\alpha = 0.75$. Fig. 2 illustrates the noise free solutions of the approximate velocities, pressure and temperature. Comparison of one-step iterative ($N = K = 1$) and nested iterative algorithms is shown in Figs. 3 and 4. In these figures, filtered solutions are shown on the left and the un-filtered solutions are shown on the right. We immediately noticed that noise reduction over the approximate velocities was accomplished very well by Algorithms 4.2-4.3. We also reported the results of the filtered pressure and temperature solutions since these solutions were driven by the filtered velocities. Figs. 5 and 6 showed the numerical performances of the NI methods on the filtered pressure and temperature solutions. As $\hat{\sigma}$ decreases, the profiles of these solutions are close to the exact one although weak zig-zag-like patterns exist at its four boundaries. It can be seen that when filtered pressure solutions are subjected to Neumann's boundary conditions, the pressure profiles are well reconstructed.

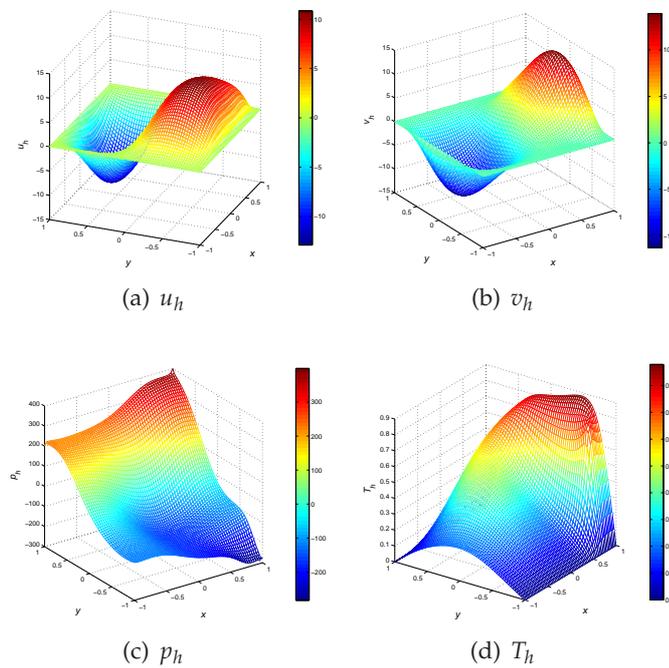


Figure 2: Noise-free solutions for approximate velocity, pressure and temperature solutions at $t_f = 1$.

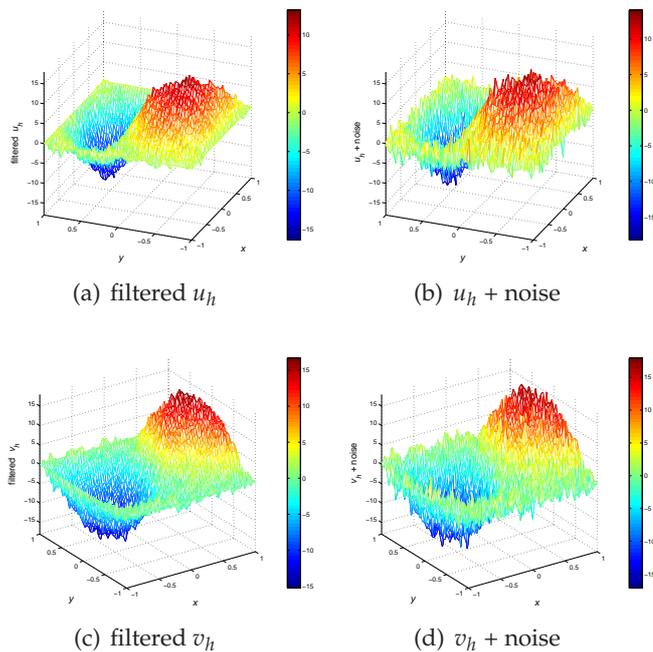


Figure 3: Reconstructed velocity solutions when the field is corrupted by $\hat{\sigma}=0.1$ at $t_f=1$ using a one-step iteration. The reconstructions are obtained by Algorithms 4.2-4.3, and $N=K=1$.

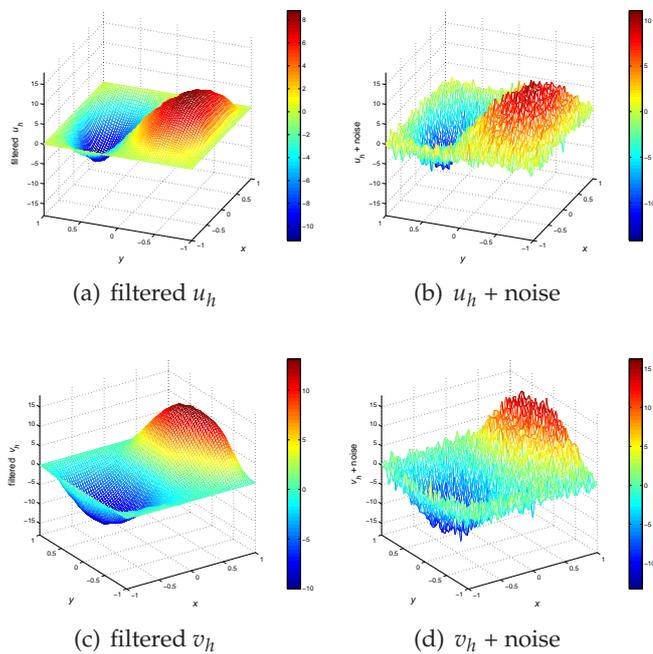


Figure 4: Reconstructed velocity solutions when the field is corrupted by $\hat{\sigma}=0.1$ at $t_f=1$. The reconstructions are obtained by Algorithms 4.2-4.3, and $N=K=10$.

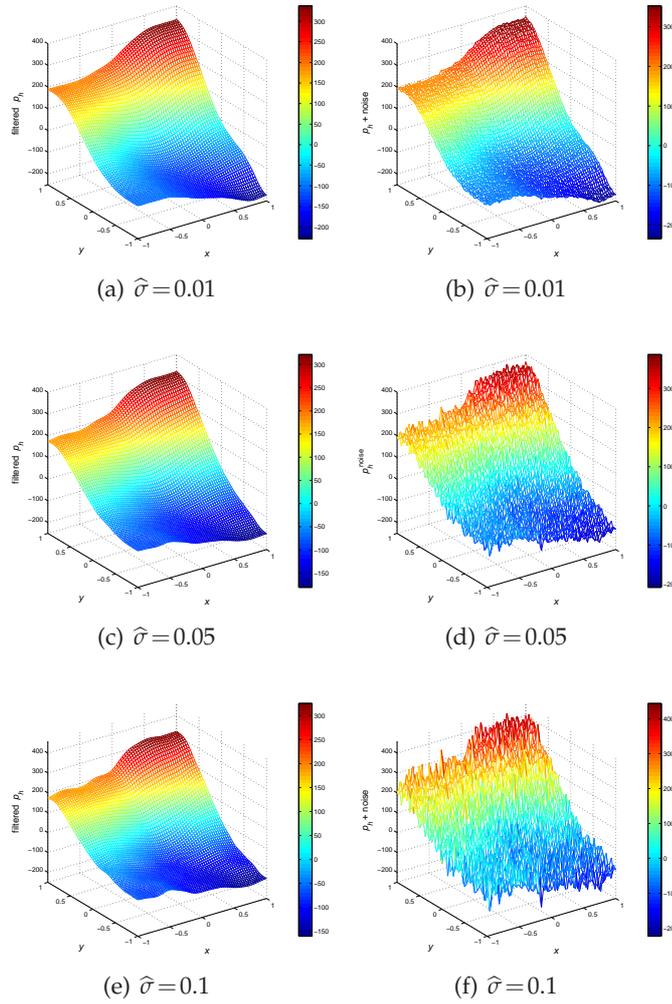


Figure 5: Reconstructed pressure solutions when the field is corrupted by different levels of $\hat{\sigma}$ at $t_f=1$. The reconstructions are obtained by Algorithms 4.2-4.3, and $N=K=10$.

- Problem set 2: We fixed the values of $N=K=17$ and $\alpha=0.75$. We compare the performance of Algorithms 4.2-4.3 and Algorithms 4.4-4.5 when the solutions are corrupted by $\hat{\sigma}_u=0.005$, $\hat{\sigma}_p=0.005$, and $\hat{\sigma}_h=0.05$. For illustrative purposes, the results of the filtered temperature solutions are reported. Fig. 7 illustrates the noise free solutions of approximate temperature involved with two heat sources. In order to measure the effectiveness of these algorithms, the following relative error is used:

$$\zeta_{*,h} := \frac{\|\zeta_{\text{approx},h} - \zeta_{*,h}\|}{\|\zeta_{\text{approx},h}\|},$$

where $\|\star\| = \sqrt{\sum_{i=1}^{\text{DOF}} \star_i^2}$ (DOF = 129^2 or 65^2), $\zeta_{*,h}$ is a filtered solution and the sub-

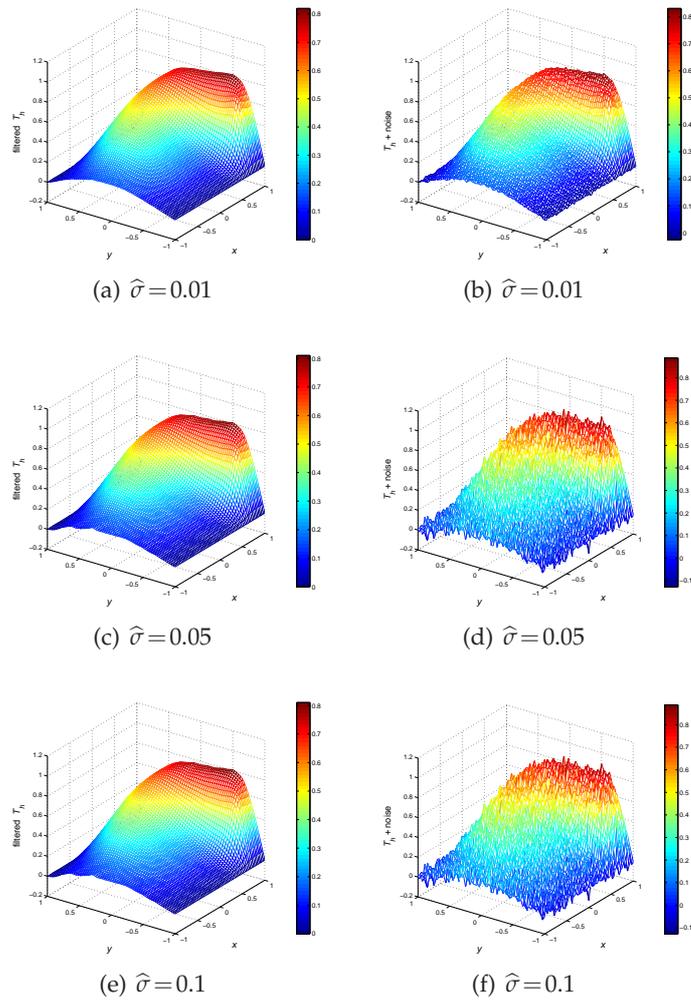


Figure 6: Reconstructed temperature solutions when the field is corrupted by different levels of $\hat{\sigma}$ at $t_f=1$. The reconstructions are obtained by Algorithms 4.2-4.3, and $N=K=10$.

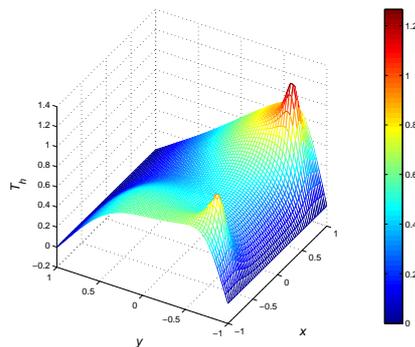


Figure 7: Noise-free solution for the approximate temperature solution at $t_f=1$.

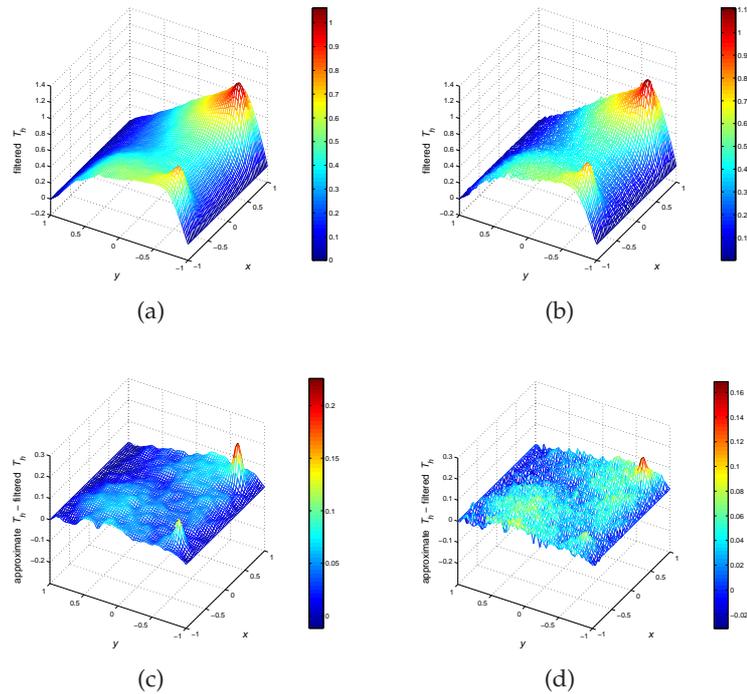


Figure 8: Reconstructed temperature solutions when the field is corrupted by $\hat{\sigma}_h = 0.05$ at $t_f = 1$. (a) Solution obtained by Algorithms 4.2-4.3. (b) Solution obtained by the TV-based algorithms (Algorithm 4.4-4.5). (c) Absolute error for the solution obtained by Algorithms 4.2-4.3. (d) Absolute error for the solution obtained by the TV-based algorithms.

script $*$ = {NI,TV} represents the choice of Algorithms 4.2-4.3 and Algorithms 4.4-4.5 with Algorithm 4.1, respectively. It can be seen that

- Use of the TV norm algorithm results in the capture of two peaks of filtered temperature profiles, while the use of the iterative regularization algorithm slightly smooths out the filtered temperature solution, as illustrated by the comparison of Fig. 8(a) with Fig. 8(b),
- In Figs. 8(c) and (d), the magnitude of the error difference between the approximate and filtered temperature solutions when compared with that of the iterative regularization algorithm is smaller. The relative error $T_{NI,h}$ is $8.4457E-2$, while the relative error $T_{TV,h}$ is $8.0223E-2$.
- Hence, the TV algorithm performs well.

5.2 Mixed convection problems

A system of non-dimensional MC equations in the PV formulation is used as follows [19]:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \frac{Gr}{Re^2} T \mathbf{j}, & \text{in } \Omega \times (0, t_f], \\ \nabla \cdot \mathbf{u} = 0, & \text{in } \Omega \times (0, t_f], \\ \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = \frac{1}{Pr Re} \nabla^2 T, & \text{in } \Omega \times (0, t_f], \end{cases} \quad (5.4)$$

with the initial conditions

$$\mathbf{u}(x, t=0) = \mathbf{0} \quad \text{and} \quad T(x, t=0) = 0, \quad \text{in } \Omega, \quad (5.5)$$

and the left- and right-hand walls are insulated (or adiabatic) and the boundary conditions for velocity and temperature on $\Omega \times (0, t_f]$ are

$$\begin{cases} T = T_{\text{hot}}, \quad u = 0, \quad v = U_0, & \text{for } x = 0, \quad 0 \leq y \leq 1, \\ T = T_{\text{cold}}, \quad u = 0, \quad v = 0, & \text{for } x = 1, \quad 0 \leq y \leq 1, \\ \frac{\partial T}{\partial y} = 0, \quad u = 0, \quad v = 0, & \text{for } y = 0, \quad 0 \leq x \leq 1, \\ \frac{\partial T}{\partial y} = 0, \quad u = 0, \quad v = 0, & \text{for } y = 1, \quad 0 \leq x \leq 1. \end{cases} \quad (5.6)$$

We select $T_{\text{cold}} = 0$, $T_{\text{hot}} = 1$ and $U_0 = 1$. For an adiabatic wall no heat transfer is permitted through the lower and upper boundaries.

In [19], we investigated the steady state of MC flows for the following choices of parameters: the Prandtl number $Pr = 0.71$, the Grashof number $Gr = 10^6$ and the Reynolds number $Re = 3000$ with $t_f = 400$ and $\Delta t = 0.0025$. We use these values as an initial condition of the fluid flows.

Due to the space limitation, the approximate pressure is only reported in Fig. 9, where two sharp pressure profiles appear at the top of right- and left-corners due to the boundary conditions for the constant velocity value U_0 . We compare the performance of Algorithms 4.2-4.3 and Algorithms 4.4-4.5 when the solutions are corrupted

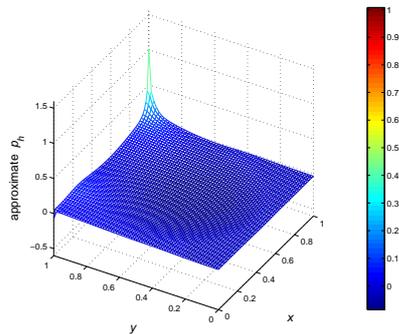


Figure 9: Noise-free solution for the approximate pressure solution at $t_f = 400$.

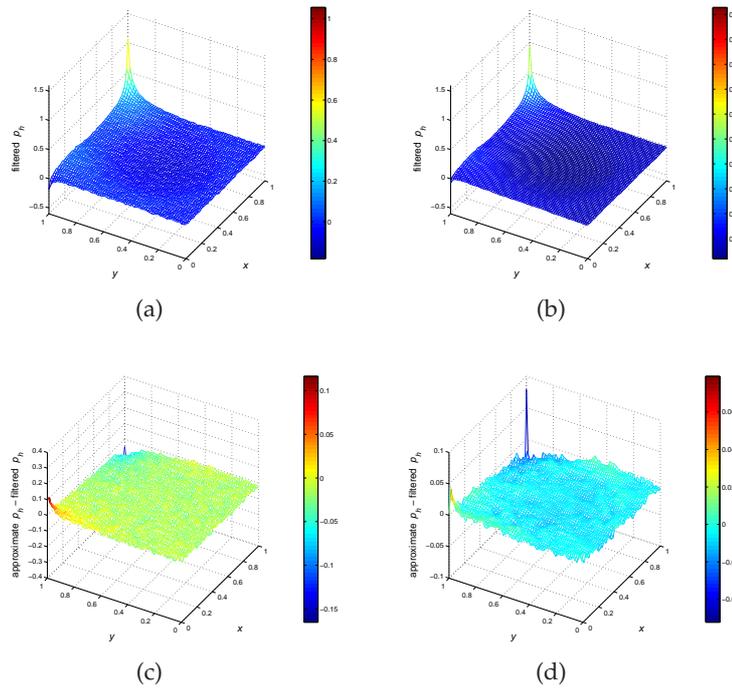


Figure 10: Reconstructed pressure solutions when the field is corrupted by $\hat{\sigma}_p=0.01$ at $t_f=400+\Delta t$. (a) Solution obtained by Algorithms 4.2-4.3. (b) Solution obtained by the TV-based algorithms (Algorithms 4.4-4.5). (c) Absolute error for the solution obtained by Algorithms 4.2-4.3. (d) Absolute error for the solution obtained by the TV-based algorithms.

by $\hat{\sigma}_u = \hat{\sigma}_p = \hat{\sigma}_T = 0.01$. We fixed the values of $N = K = 10$ with $\Delta t = 0.025$. In Fig. 10, two sets of regularization length scales are used:

- $\alpha_u = \alpha_p = \alpha_T = 0.2$ and $\alpha_{u,TV} = \alpha_{p,TV} = \alpha_{T,TV} = 0$ in the left panel;
- $\alpha_u = \alpha_p = \alpha_T = 0.2$ and $\alpha_{u,TV} = 1.3, \alpha_{p,TV} = 3.95, \alpha_{T,TV} = 1.5$ in the right panel.

We conclude that

- In Figs. 10(a) and (b), it can be seen that both pressure profiles filtered out the noise.
- In Figs. 10(c) and (d), the range of the difference values in the left panel is 0.2820, and in the right panel it is 0.1312; the magnitude of the error difference between the approximate and filtered pressure solutions is smaller when Algorithms 4.4-4.5 are used. The relative error $p_{TV,h}$ is $1.4262E-1$, while the relative error $p_{NI,h}$ is $3.9450E-1$.
- Hence, using the TV based algorithm with the diffusion term gives the most promising result.

6 Conclusions

We have proposed two iterative regularization algorithms for smoothing out the approximate solutions when the data are corrupted by noise. Our findings are:

- Iterative differential and Stokes-differential filters may be advantageous for the noise removal when the profiles of the approximate solutions do not exhibit abrupt changes. When the solutions contain discontinuities or sharp gradients, these filters produce too much smoothing and hence some important features of the solutions are also removed. As a result, the solutions obtained are inaccurate.
- The TV based model with iterative differential and Stokes-differential filters has a remarkable effect on the stability of the solutions. This iterative model is very promising for noise removal when the solution profiles have sharp corners and a steep gradient. However, the addition of the global TV term affects the accuracy of the solution in smooth regions. A remedy for this is to apply a local version of this TV regularization, namely, we only regularize the solution at locations where a sharp gradient is detected. A related work can be found in [2].

There are a number of possible future works which stem from this work:

- The choice of suitable regularization parameters versus each time-step size in a more realistic/physical situation should further explored. See, for example, [23].
- There are no restrictions that prevent the present study from being extended to other regularization models. See, for example, Clark-alpha model [24], Lagrangian-averaged Navier-Stokes-alpha model [25] and Navier-Stokes-alpha-beta model [26].
- A comparative study of different TV-based models should be further studied. See, for example, split Bregman method [29], TV-Stokes model [30] and Alternating direction method [31].
- The use of a local TV based filter to enhance the accuracy of the solution. It was shown to be successful in a related problem, see [2].

References

- [1] A. Bowers, L. Rebholz, A. Takhirov and C. Trenchea, Improved accuracy in regularization models of incompressible flow via adaptive nonlinear filtering, To appear in International Journal for Numerical Methods in Fluids.
- [2] H. N. Chan and E. T. Chung, A staggered discontinuous Galerkin method with local TV regularization for the Burgers equation, Submitted.
- [3] E. T. Chung and C. S. Lee, A staggered discontinuous Galerkin method for the convection-diffusion equation, J. Numer. Math., 20 (2012), 1–31.

- [4] E. T. Chung and W. T. Leung, A sub-grid structure enhanced discontinuous Galerkin method for multiscale diffusion and convection-diffusion problems, *Commun. Comput. Phys.*, 14 (2013), 370–392.
- [5] E. T. Chung, H. H. Kim and O. Widlund, Two-level overlapping Schwarz algorithms for a staggered discontinuous Galerkin method, *SIAM J. Numer. Anal.*, 51 (2013), 47–67.
- [6] E. T. Chung, T. F. Chan and X. C. Tai, Electrical impedance tomography using level set representation and total variational regularization, *J. Comput. Phys.*, 205 (2005), 357–372.
- [7] B. J. Geurts and D. D. Holm, Leray and LANS- α modeling of turbulent mixing, *J. Turbulence*, 7 (2006), 1–33.
- [8] J. S. Mullen and P. F. Fischer, Filtering techniques for complex geometry fluid flows, *Commun. Numer. Meth. Eng.*, 15 (1999), 9–18.
- [9] A. Dunca, Optimal design of fluid flow using subproblems reduced by large eddy simulation, Report, DEP, Argonne National Lab., Argonne, IL, 2002.
- [10] W. Layton, C. Manica, M. Neda and L. Rebholz, Numerical analysis and computational testing of a high accuracy Leray-deconvolution model of turbulence, *Numer. Meth. Part. Differential Equations*, 24 (2007), 555–582.
- [11] W. Layton, L. Rebholz and C. Trenchea, Modular nonlinear filter stabilization of methods for higher Reynolds numbers flow, *J. Math. Fluid Mech.*, to appear.
- [12] J. Connors, Convergence analysis and computational testing of the finite element discretization of the Navier-Stokes alpha model, *Numer. Meth. Part. Differential Equations*, 26 (2010), 1328–1350.
- [13] V. Ervin, W. Layton and M. Neda, Numerical analysis of filter based stabilization for evolution equations, Submitted.
- [14] V. J. Ervin, W. J. Layton and M. Neda, Numerical analysis of a higher order time relaxation model of fluids, *Int. J. Numer. Anal. Mod.*, 4 (2007), 648–670.
- [15] R. Ingram, W. Layton and N. Mays, A superconvergence theory for a family of iterated regularization methods, Submitted.
- [16] Y. Mao and J. Gilles, Non rigid geometric distortions correction-application to atmospheric turbulence stabilization, To be appeared in *Inverse Problems and Imaging*.
- [17] N. Mays, Iterated Regularization Methods for Solving Inverse Problems, Ph.D. Thesis, University of Pittsburgh, 2011.
- [18] J. L. Guermond and J. Shen, A new class of truly consistent splitting schemes for incompressible flows, *J. Comput. Phys.*, 192 (2003), 262–276.
- [19] J. C. F. Wong, Numerical simulation of two-dimensional laminar mixed-convection in a lid-driven cavity using the mixed finite element consistent splitting scheme, *Int. J. Numer. Methods Heat Fluid Flows*, 17 (2007), 46–93.
- [20] H. M. Park and O. Y. Chung, On the solution of an inverse natural convection problem using various conjugate gradient methods, *Int. J. Numer. Meth. Eng.*, 47 (2000), 821–842.
- [21] J. Leray, Essai sur le mouvement d'un fluide visqueux emplissant l'espace, *Acta Math.*, 63 (1934), 193–248.
- [22] M. van Reeuwijk, H. J. J. Jonker and K. Hanjalic, Incompressibility of the Leray-alpha model for wall-bounded flows, *Phys. Fluids*, (2006), 018103.
- [23] K. A. Scott and F. S. Lien, Application of the NS- α model to a recirculating flow, *Flow Turbulence Combust.*, 84 (2010), 167–192.
- [24] C. Cao, D. D. Holm and E. S. Titi, On the Clark- α model of turbulence: global regularity and long-time dynamics, *J. Turbulence*, 6 (2005), 1–11.
- [25] J. P. Graham, D. Holm, P. D. Mininni and A. Pouquet, Three regularization models of the

- Navier-Stokes equations, *Phys. Fluids*, 20 (2008), 035107.
- [26] T. Y. Kim, M. Neda, L. G. Rebholz and E. Fried, A numerical study of the Navier-Stokes- $\alpha\beta$ model, Submitted.
 - [27] H. H. Kim, E. T. Chung and C. S. Lee, A BDDC algorithm for a class of staggered discontinuous Galerkin methods, Submitted.
 - [28] H. H. Kim, E. T. Chung and C. S. Lee, A staggered discontinuous Galerkin method for the Stokes system, Submitted.
 - [29] T. Goldstein and S. Osher, The split Bregman method for $L1$ regularized problems, *SIAM J. Imaging Sci*, 2 (2009), 323–343.
 - [30] J. Hahn, C. Wu and X. Tai, Augmented Lagrangian method for generalized TV-Stokes model, *J. Sci. Comput.*, 50 (2012), 235–264.
 - [31] Y. Xiao and J. Yang, A fast algorithm for total variation image reconstruction from random projections, Submitted.
 - [32] L. Zhong, E. T. Chung and C. Liu, Some efficient techniques for the symmetric discontinuous Galerkin approximation of second order elliptic problems, Submitted.