Mathematical Modelling and Analysis
Volume 28, Issue 1, 71–90, 2023
https://doi.org/10.3846/mma.2023.15482

An Improved SIMPLEC Scheme for Fluid Registration

Mohamed Alahyane[a], Abdelilah Hakim[b], Amine Laghrib[c] and Said Raghay[b]

[a] *L2EP Laboratory, University of Lille*
59000 Lille, France
[b] *LAMAI, Faculté des Sciences et Techniques*
B.P 549, Av.Abdelkarim Elkhattabi, Guéliz Marrakech, Morocco
[c] *EMI, Faculty of Sciences and Techniques (M'Ghila), Sultan Moulay Slimane University*
P.O. Box. 523, Béni Mellal, Morocco

E-mail(*corresp.*): laghrib.amine@gmail.com

Received August 23, 2021; revised November 5, 2022; accepted November 25, 2022

Abstract. The image registration is always a strongly ill-posed problem, a stable numerical approach is then desired to better approximate the deformation vectors. This paper introduces an efficient numerical implementation of the Navier Stokes equation in the fluid image registration context. Although fluid registration approaches have succeeded in handling large image deformations, the numerical results are sometimes inconsistent and unexpected. This is related, in fact, to the used numerical scheme which does not take into consideration the different properties of the continuous operators. To take into account these properties, we use a robust numerical scheme based on finite volume with pressure correction. This scheme, which is called by the Semi-Implicit Method for Pressure-Linked Equation-Consistent (SIMPLEC), is known for its stability and consistency in fluid dynamics context. The experimental results demonstrate that the proposed method is more efficient and stable, visually and quantitatively, compared to some classical registration methods.

Keywords: image registration, fluid registration, SIMPLEC, Navier Stokes equations.

AMS Subject Classification: 58F15; 58F17.

1 Introduction

Image registration is a widely used technique in image processing; its main principle is to compute geometrical correspondences between two or more given images [7]. It is considered as a useful tool in numerous applications: including astronomy, robotics and especially in bio-medical imaging [19, 25].

Generally, image registration problems can be classified into two types: intensity-based methods and feature-based methods. The first type methods are based on the image's gray spaces [2, 3, 20], while the second ones exploit the difference features existing between the given images, such as regions and edges, to find the correspondences in the image's feature spaces [23]. In most cases, the image registration task is formulated as an optimization problem involving a distance measure to evaluate the similarity. However, due to different illuminations (grey-levels) of the image regions, the pairing cannot be achieved successfully, which makes the image registration an ill-posed problem. An additive regularization term is then added, motivated by the nature of the transformation. In fact, each regularizer produces a different registration model, and this choice is very vital for the solution's existence and uniqueness. One of the classical regularizers is the elastic regularization [4]. Even if this model has been addressed in several image registration problems [24], it is still limited, since it does not handle largely deformed data sets [25]. Another weakness of the linear elasticity regularization is that it does not take in consideration discontinuous displacement. A more convincing choice for preserving discontinuities of the displacement was the total variation-based regularization (TV), see [28]. The TV model regularizes the velocity/displacement/mapping, in contrast, it gives unsatisfactory registration results when the displacements are very smooth. A more robust regularization is proposed in [10], which takes into account the advantages of the stopped diffusion of the total variation registration models. Recently, an improved discontinuity-preserving model was proposed in [30] motivated by several regularization techniques used in vector-valued image denoising, which gives promising results; but didn't tackle large deformations. In the context of non-parameteric registration, Christensen [6] developed one of the most effective non parametric registration called the fluid registration [8, 9]. The purpose of the fluid image registration is to compute the deformation $u(x,t)$ at time t for a given force field f, while the deformed image is considered in a viscous fluid. The deformations are then governed by the Navier Stokes equations for momentum conservation (where the pressure is neglected). This equation is given by

$$\begin{cases} \nu \Delta v(x,t) + (\mu+\nu)\nabla(\nabla \cdot v(x,t)) = f(x,t,u(x,t)), \\ v(x,t) = \partial_t u(x,t) + v(x,t) \cdot \nabla u(x,t). \end{cases} \tag{1.1}$$

The component Δv is called the viscous term, while the component term $\nabla(\nabla \cdot v(x,t))$ allows for contraction and expansion of the fluid. The second part of (1.1) defining material derivative of the displacement u. The constants μ and ν are the Lamé coefficients.

Although the fluid image registration appears easy to solve, in reality it is quite complicated. One of the key complications is the choice of demons

force [15]. Moreover, fluid image registration is not based on a direct optimization approach. Also, since we deal with a set of nonlinear partial differential equations (PDE), great care must be taken in the choice of the discretization method. In addition, when displacement u is discontinuous, the fluid image registration gives poor results since it skips the discontinuity regions. Another well-known drawback of this method is the high computational cost. Indeed, once the force field f is fixed, the first part of Equation (1.1) is solved for $v(x,t)$ using the successive over-relaxation (SOR) scheme [6], which computes an accurate fluid model at the expense of a large computational time. Then, an explicit Euler scheme is used to advance u in time t. The method requires at each iteration the computation of the Jacobian matrix of the displacement field, which is also computationally very expensive. This framework is thus time-consuming, which motivates the search of faster implementations. Recently, a new fluid approach was proposed using the Semi-Implicit Method for Pressure-Linked Equation (SIMPLE) scheme, where the pressure is well defined in the context of image registration [1]. However, if the registered images are not with the same modalities and in the presence of noise, this method becomes inconsistent with the apparition of some artefacts. Hence the introduction of another more accurate scheme.

The contributions in this paper are summarized below.

Firstly, we define the pressure term, in the image registration context, as the effect of each region with respect to the nearest one, which has not been treated by Christensen in the previous model (1.1).

Secondly, we propose a stable, consistent and fast numerical implementation of the Navier Stokes equations for an incompressible fluid to resolve the fluid image registration problem. Which allows to introduce ideas from computational fluid dynamics in the image analysis context [29].

The use of the finite difference approximation in the image registration is very limited since it does not take into account the continuous and discontinuous operator properties. An alternative and robust discretization scheme is then needed. In this paper, we propose a finite volume scheme type to handle different properties of the fluid registration in the discretization process of the proposed Navier Stokes equation [13]. The use of this scheme can efficiently take into account the defined pressure term in the registered image. After the success of the Semi-Implicit Method for Pressure Linked Equations [26] algorithm in solving the Navier Stokes equations and many other problems in computational fluid dynamics (CFD) [14,18,26], the Semi-Implicit Method for Pressure Linked Equations-Consistent (SIMPLEC) [12] is also proposed with good consistency. However, no such attempt has been proposed to solve the fluid image registration problem. Then, in this work we introduce the SIMPLEC algorithm as a discretization approach to solve the Navier-Stokes equation applied to image registration problems. We can see that the numerical results confirm that our proposed scheme is more efficient compared with other classical methods.

This paper is organized as follows: Section 2 explains the main concepts of the image registration and introduces the proposed Navier Stokes equation for solving the fluid registration problem. In Section 3, we propose the SIM-

PLEC discretization scheme. Finally, numerical results and comparisons of our approach with classical registration methods are presented in Section 4.

2 Mathematical model of fluid registration

The registration problem is usually based on a minimization problem between two images, template image and reference image [21]. Given two images $\mathcal{T}$, $\mathcal{R}$: $\Omega \subset \mathbb{R}^n \to \mathbb{R}$, compactly supported on Ω (a bounded convex domain), where n denotes spatial dimension of the given images (in our case $n = 2$). The purpose of registration is to look for a transformation $u : \Omega \subset \mathbb{R}^n$ such that ideally $\mathcal{T}(u(x))$ looks like $\mathcal{R}(x)$ as much as possible for all $x \in \Omega$. In summary, the desired transformation u is achieved by minimizing a so-called distance measure $\mathcal{D}$. Since this problem is ill-posed, an appropriate regularization $\mathcal{S}$ is used.
The variational formulation of the image registration problem consists of finding a minimizer u of

$$\mathcal{J}(u) = \mathcal{D}(\mathcal{T}, \mathcal{R}; u) + \mathcal{S}(u) \quad \text{for} \quad u \in \mathcal{A}, \tag{2.1}$$

where $\mathcal{A}$ denotes the set of admissible transformations.

The regularization term $\mathcal{S}$ is in general based on the gradient of u, noted ∇u. There is in fact many choices of the regularization term, the widely used are: diffusion registration [22] and elastic registration [21]. One of the more successful regularization choices was the fluid one [11], which is formulated such as an elastic potential of $\partial_t u$. To solve this problem, the Euler-Lagrange identity is used, which coincides with the resolution of the Navier-Lamé equation. There are also other regularizations derived from the diffusion one, such as the curvature term and the hyperelastic energy [5], which do not have good physical motivations.

Based on the proprieties of this fluid regularization, we propose a dynamic equation for the incompressible Newtonian fluids (2.2), which are governed by the Navier Stokes equations. This equation coupled the velocity vector field $v(x,t)$ to a scalar pressure $p(x,t)$ such as

$$\begin{cases} \dfrac{\partial v}{\partial t}(x,t) + v(x,t) \cdot \nabla v(x,t) = -\nabla p(x,t) + \nu \Delta v(x,t) + f(x,t,u(x,t)), \\ \nabla \cdot v(x,t) = 0, \end{cases} \tag{2.2}$$

where the velocity $v(x,t)$ is calculated according to the displacement $u(x,t)$ such as

$$v(x,t) = \partial_t u(x,t) + v(x,t) \cdot \nabla u(x,t). \tag{2.3}$$

The main analogy followed in this paper is the parallel between the incompressible Newtonian fluid and the image velocity of each pixel $v(x,t)$ under the image registration concept. The introduced Navier Stokes equation (2.2) is well posed in the image registration task. In fact, the parameter ν is supposed to be the factor of the diffusion in the imaging problems. The pressure p is also modelled in the imaging task, which represents the effect of each region on

the nearest one in the image during the registration process. In fact, since the pressure is given in gradient form, it may represent the contours. We suppose in the following that p is an external force. The term $\nabla \cdot v = 0$ is well posed since the pixels are incondensable. Finally, f is supposed to be the external forces obtained by the gradient of the distance $\mathcal{D}$ between the two images; $\mathcal{T}$ and $\mathcal{R}$. A classical choice for this distance is the sum square difference (SSD) measure defined as

$$\mathcal{D}_{SSD}(\mathcal{T}, \mathcal{R}; u) = \int_{\Omega} (\mathcal{T}(u(x)) - \mathcal{R}(x))^2 \, dx. \tag{2.4}$$

Hence, the external forces are computed such as:

$$f(x, t, u(x, t)) = \nabla \mathcal{T}(u(x, t)) \left(\mathcal{R}(x) - \mathcal{T}(u(x, t))\right).$$

For this choice, we can effectively assure the existence of a solution to the problem (2.1) using techniques in [11]. There are many advantages in the use of the Navier Stokes equation. Firstly, we don't have any problem with the existence of a solution, since it is well-developed in the literature [29]. Secondly, there are many and stable numerical approaches that we can use to resolve this equation derived from a classical example of fluid dynamics.
For a mathematical transparency of the Navier Stokes equation (2.2), a convenient way is to consider it nondimensional form [16]. This is obtained by introducing a reference length L_* and a reference time T_*, then we set

$$v^{'} = \frac{v}{V_*}, \quad x^{'} = \frac{x}{L_*}, \quad t^{'} = \frac{t}{T_*}, \quad P^{'} = \frac{p}{V_*^2}, \quad f^{'} = \frac{L}{V_*^2} f,$$

where V_* is the reference velocity defined as $V_* = \frac{L_*}{T_*}$. By a substitution into (2.2), we obtain for $v^{'}$, $P^{'}$, $f^{'}$ the following equation

$$\begin{cases} \dfrac{\partial v^{'}}{\partial t} + (v^{'} \cdot \nabla) v^{'} = -\nabla P^{'} + \frac{1}{\mathrm{Re}} \Delta v^{'} + f^{'}(\,.\,, u(\,.\,)), \\ \nabla \cdot v^{'} = 0, \end{cases} \tag{2.5}$$

where R_e is a nondimensional constant called the Reynolds number defined as $\frac{1}{\mathrm{R_e}} = \frac{\nu}{L_* V_*}$, and to avoid the complexity of the notations, we keep the same notation in (2.2), i.e., we substitute $v^{'}$, $P^{'}$, $f^{'}$ by v, p, f. Then, we rewrite the Equation (2.5) as follows:

$$\begin{cases} \dfrac{\partial v}{\partial t}(x, t) + (v(x, t) \cdot \nabla) v(x, t) = -\nabla p(x, t) + \frac{1}{\mathrm{Re}} \Delta v(x, t) + f(x, t, u(x, t)), \\ \nabla \cdot v(x, t) = 0. \end{cases} \tag{2.6}$$

In the following section, we discuss the proposed discretization scheme for the Navier Stokes equation (2.6), in the image registration context.

3 Discretization

To solve the Navier Stokes equation (2.6), we use a finite volume discretization based on a staggered control volume illustrated in the Figure 1. To calculate the

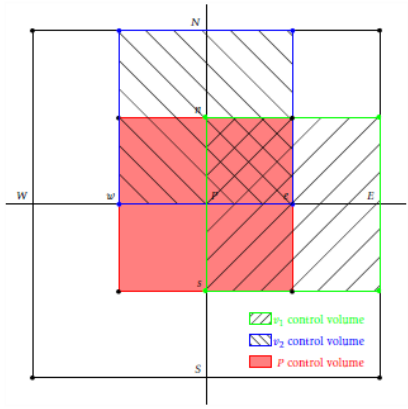


Figure 1. Control volume of v and p.

variables v_1 (the horizontal component of velocity), v_2 (the vertical component of velocity) and p (Pressure), we use three staggered mesh [27]. The control volumes for v_1 and v_2 are displaced with respect to the control volume for the continuity equation. In the Figure 1, "P" denotes the node at which the partial differential equation is approximated, and "E", "W", "N", "S" are its neighbours. Cell faces "e", "w" for v_1 and "n", "s" for v_2 are in the midway between the nodes.

Firstly, we rewrite the momentum equation of (2.6) in a differential form for both velocity components

$$\begin{aligned}
&\frac{\partial v_1}{\partial t}+\frac{\partial}{\partial x}\left(v_1v_1-\frac{1}{\mathrm{R}_e}\frac{\partial v_1}{\partial x}\right)+\frac{\partial}{\partial y}\left(v_2v_1-\frac{1}{\mathrm{R}_e}\frac{\partial v_1}{\partial y}\right)=-\frac{\partial p}{\partial x}+f_1(.,u_1(.),u_2(.)),\\
&\frac{\partial v_2}{\partial t}+\frac{\partial}{\partial x}\left(v_1v_2-\frac{1}{\mathrm{R}_e}\frac{\partial v_2}{\partial x}\right)+\frac{\partial}{\partial y}\left(v_2v_2-\frac{1}{\mathrm{R}_e}\frac{\partial v_2}{\partial y}\right)=-\frac{\partial p}{\partial y}+f_2(.,u_1(.),u_2(.)),\\
&\frac{\partial v_1}{\partial x}+\frac{\partial v_2}{\partial y}=0.
\end{aligned}\tag{3.1}$$

To solve this system, we discretize each equation separately and we use the SIMPLE method. The discretization momentum equation for v_1 is derived by integrating the first equation of (3.1) over the control volume corresponding to v_1, using the points (E,P,n,s) shown in Figure 1 and over the time interval; from t to $t+\Delta t$. Thus, using the fact that the velocity v_1 and v_2 do not depend on the vertical and horizontal components, respectively, under each volume control face, we have then

$$\begin{aligned}
&\frac{vol}{\Delta t}\left(v_1-v_1^0\right)+\int_s^n\left(v_1v_1-\frac{1}{\mathrm{R}_e}\frac{\partial v_1}{\partial x}\right)_E-\left(v_1v_1-\frac{1}{\mathrm{R}_e}\frac{\partial v_1}{\partial x}\right)_P dy\\
&+\int_P^E\left(v_2v_1-\frac{1}{\mathrm{R}_e}\frac{\partial v_1}{\partial y}\right)_n-\left(v_2v_1-\frac{1}{\mathrm{R}_e}\frac{\partial v_1}{\partial y}\right)_s dx=vol\,f_1-\Delta y\left(p_E-p_P\right)
\end{aligned}\tag{3.2}$$

with $vol=\Delta x\Delta y$ and v_1^0 is the previous iteration of v_1 (in general it represents v_1 at $t=0$). To simplify the study of this problem, we introduce some new

entities defined as

$$J_E = \int_s^n \left(v_1 v_1 - \frac{1}{\mathrm{R}_e} \frac{\partial v_1}{\partial x} \right)_E dy, \quad J_W = \int_s^n \left(v_1 v_1 - \frac{1}{\mathrm{R}_e} \frac{\partial v_1}{\partial x} \right)_P dy,$$

$$J_N = \int_P^E \left(v_2 v_1 - \frac{1}{\mathrm{R}_e} \frac{\partial v_1}{\partial y} \right)_n dx, \quad J_S = \int_P^E \left(v_2 v_1 - \frac{1}{\mathrm{R}_e} \frac{\partial v_1}{\partial y} \right)_s dx.$$

If we substitute this new entities in the Equation (3.2), we have then

$$\frac{vol}{\Delta t} \left(v_1 - v_1^0 \right) + J_E - J_W + J_N - J_S = vol f_1 - \Delta y \left(p_E - p_P \right). \tag{3.3}$$

Also, the integration of the third continuity equation in (3.1) over the corresponding control volume and time gives

$$\int_t^{t+\Delta t} \int_P^E \int_s^n \mathrm{R}_e \left(\frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} \right) dy dx dt = 0,$$

which it is equivalent to

$$F_E - F_W + F_N - F_S = 0, \tag{3.4}$$

where $F_E = \Delta y \left(\mathrm{R}_e v_1 \right)_e$, $F_W = \Delta y \left(\mathrm{R}_e v_1 \right)_w$, $F_N = \Delta x \left(\mathrm{R}_e v_2 \right)_n$, $F_S = \Delta x \left(\mathrm{R}_e v_2 \right)_s$. Using the Equations (3.3) and (3.4), we find

$$\begin{aligned} (3.3) - v_{1P} \cdot (3.4) \Rightarrow & \frac{vol}{\Delta t} v_{1P} + J_E - F_E \cdot v_{1P} - J_W + F_W \cdot v_{1P} + J_N - F_N \cdot v_{1P} \\ & - J_S + F_S \cdot v_{1P} = vol f_1 - \Delta y \left(p_E - p_P \right) + \frac{vol}{\Delta t} v_{1P}^0. \end{aligned} \tag{3.5}$$

On the other hand, we suppose that the flow is unidirectional (since the registration is concentrated in a particular direction), laminar (since the pixels are not mixed) with a constant pressure on each grid (the effect of the neighbors pixels is neglected in a fixed grid). Thus, the first equation of (3.1) becomes

$$v_1 \frac{\partial v_1}{\partial x} - \frac{1}{\mathrm{R}_e} \frac{\partial^2 v_1}{\partial x^2} = 0. \tag{3.6}$$

Also, the second equation representing the vertical direction becomes

$$v_2 \frac{\partial v_1}{\partial y} - \frac{1}{\mathrm{R}_e} \frac{\partial^2 v_1}{\partial y^2} = 0. \tag{3.7}$$

To compute the entities (J_E, J_W, J_N and J_S), we use the solution of the two Equations (3.6) and (3.7). Firstly, we have to rewrite the Equation (3.6) under each grid as follow

$$\begin{cases} (v_1)_E \frac{\partial v_1}{\partial x} - \frac{1}{\mathrm{R}_e} \frac{\partial^2 v_1}{\partial x^2} = 0, \\ x = 0, \qquad v_1 = v_{1P}, \\ x = \Delta x, \qquad v_1 = v_{1E}. \end{cases} \tag{3.8}$$

Also, the expression of the Equation (3.7) in each grid is given by

$$\begin{cases} (v_2)_n \frac{\partial v_1}{\partial y} - \frac{1}{\mathrm{R}_e} \frac{\partial^2 v_1}{\partial y^2} = 0, \\ y = 0, \qquad v_1 = v_{1S}, \\ y = \Delta y, \qquad v_1 = v_{1N}. \end{cases} \tag{3.9}$$

The solution of the Equation (3.8) is calculated using this expression

$$v_1 = v_{1P} + (v_{1E} - v_{1P}) \left(\frac{\exp((\mathrm{R}_e v_1)_E x) - 1}{\exp((\mathrm{R}_e v_1)_E \Delta x) - 1} \right). \tag{3.10}$$

In addition, the solution of (3.9) is given by

$$v_2 = v_{1P} + (v_{1N} - v_{1P}) \left(\frac{\exp((\mathrm{R}_e v_1)_n y) - 1}{\exp((\mathrm{R}_e v_1)_n \Delta y) - 1} \right).$$

We assume that $P_e = (\mathrm{R}_e v_1)_E \Delta x$, where P_e is called the Peclet number, representing the local report on the boundary portion of the volume control for inertial and viscous forces. The Equation (3.10) is now described as

$$v_1 = v_{1P} + (v_{1E} - v_{1P}) \left(\frac{\exp((\mathrm{R}_e v_1)_E x) - 1}{\exp(P_e) - 1} \right).$$

We can now calculate the expression of J_E using the Equation (3.10)

$$J_E = v_{1P} \frac{(\mathrm{R}_e v_1)_E \exp(P_e)}{\exp(P_e) - 1} - v_{1E} \frac{(\mathrm{R}_e v_1)_E}{\exp(P_e) - 1},$$

injecting the expression of J_E in (3.5), we have

$$\begin{aligned} J_E - v_{1P} F_E &= \Delta y \left(v_{1P} \frac{(\mathrm{R}_e v_1)_E \exp(P_e)}{\exp(P_e) - 1} - v_{1E} \frac{(\mathrm{R}_e v_1)_E}{\exp(P_e) - 1} \right) - v_{1P} F_E \\ &= v_{1P} \frac{F_E}{\exp(P_e) - 1} - v_{1E} \frac{F_E}{\exp(P_e) - 1} = A_E (v_{1P} - v_{1E}). \end{aligned}$$

By the same way, we can find the others expressions $(J_i - v_{1P} \cdot F_i)$, $i \in (W, N, S)$ given as follow

$$\begin{aligned} -J_W + v_{1P} F_W &= v_{1P} \frac{F_W exp(P_w)}{exp(P_w) - 1} - v_{1W} \frac{F_W exp(P_w)}{exp(P_w) - 1} = A_W (v_{1P} - v_{1W}), \\ J_N - v_{1P} F_N &= v_{1P} \frac{F_N}{exp(P_n) - 1} - v_{1N} \frac{F_N}{exp(P_n) - 1} = A_N (v_{1P} - v_{1N}), \\ -J_S + v_{1P} F_S &= v_{1P} \frac{F_S exp(P_s)}{exp(P_s) - 1} - v_{1S} \frac{F_S exp(P_s)}{exp(P_s) - 1} = A_S (v_{1P} - v_{1S}). \end{aligned}$$

Therefore, (3.5) amounts to

$$\begin{aligned} &\frac{vol}{\Delta t} v_{1P} + A_E (v_{1P} - v_{1E}) + A_W (v_{1P} - v_{1W}) + A_N (v_{1P} - v_{1N}) + A_S (v_{1P} - v_{1S}) \\ &\quad = vol f_1 - \Delta y \, (p_E - p_P) + \frac{vol}{\Delta t} v_{1P}^0. \end{aligned}$$

Finally, the discretization equation can be written as

$$A_P v_{1P} = A_E v_{1E} + A_W v_{1W} + A_N v_{1N} + A_S v_{1S} + b_1 + \Delta y\,(p_P - p_E)\,, \quad (3.11)$$

where

$$A_E = \frac{F_E}{exp(P_e - 1)};\quad P_e = (\mathrm{R}_e v_1)_E \Delta x;\quad A_W = \frac{F_W exp(P_w)}{exp(P_w - 1)};\quad P_w = (\mathrm{R}_e v_1)_P \Delta x,$$
$$A_N = \frac{F_N}{exp(P_n - 1)};\quad P_n = (\mathrm{R}_e v_2)_n \Delta y;\quad A_S = \frac{F_S exp(P_s)}{exp(P_s - 1)};\quad P_s = (\mathrm{R}_e v_2)_s \Delta y,$$
$$A_P = A_E + A_W + A_N + A_S + \frac{vol}{\Delta t},\quad b_1 = vol f_1 + \frac{vol}{\Delta t} v_{1P}^0.$$

The momentum equations for the other direction is obtained with the same way, see [26] for more details. So the discretization for v_2 is given as

$$A_P v_{2P} = A_E v_{2E} + A_W v_{2W} + A_N v_{2N} + A_S v_{2S} + b_2 + \Delta x\,(p_P - p_N)\,, \quad (3.12)$$

where $b_2 = vol f_2 + \frac{vol}{\Delta t} v_{2P}^0$. Since we have calculated the different momentum equations, we need to correct the pressure in each step to verify that $\nabla \cdot v(x,t) = 0$.

3.1 The pressure correction equation

The main idea of this step is to improve the guessed pressure p^* such that the velocity field will progressively converge to the solution of the continuity equation (3.1). Generally, the correct pressure p is obtained using $p = p^* + p^{'}$, where $p^{'}$ is the corrected pressure. Then, we have to compute the new velocity components corresponding to the new corrected pressure $p^{'}$. The two corresponding velocity corrections, denoted by $v_1^{'}$ and $v_2^{'}$ respectively, are obtained by

$$v_1 = v_1^* + v_1^{'}, \qquad v_2 = v_2^* + v_2^{'}. \quad (3.13)$$

For any guessed pressure p^*, the velocities v_1^* and v_2^*, obtained by solving the momentum equations (3.11)–(3.12), satisfy

$$\begin{cases} A_P v_{1P}^* = \sum_{nb} A_{nb} v_{1nb}^* + b_1 + \Delta y\,(p_P^* - p_E^*)\,, \\ A_P v_{2P}^* = \sum_{nb} A_{nb} v_{2nb}^* + b_2 + \Delta x\,(p_P^* - p_N^*)\,. \end{cases} \quad (3.14)$$

The velocities v_1 and v_2 are obtained through (3.11)–(3.12) using the correct pressure p, which satisfy the continuity condition. The correction of the guessed pressure by $p^{'} = p - p^*$ is therefore necessary to correct the velocities v_1^* and v_2^* by (3.13). The relation between $p^{'}$ and $v^{'}$ is then obtained by subtracting the Equation (3.14) from (3.11)–(3.12)

$$\begin{cases} A_P v_{1P}^{'} = \sum_{nb} A_{nb} v_{1nb}^{'} + \Delta y \left(p_P^{'} - p_E^{'}\right), \\ A_P v_{2P}^{'} = \sum_{nb} A_{nb} v_{2nb}^{'} + \Delta x \left(p_P^{'} - p_N^{'}\right). \end{cases} \quad (3.15)$$

In the SIMPLEC algorithm, the terms $\sum_{nb} A_{nb} v_{1P}^{'}$ and $\sum_{nb} A_{nb} v_{2P}^{'}$ are subtracted from both sides of Equation (3.15). This yields

$$\begin{cases} \left(A_P - \sum_{nb} A_{nb}\right) v_{1P}^{'} = \sum_{nb} A_{nb} \left(v_{1nb}^{'} - v_{1P}^{'}\right) + \Delta y \left(p_P^{'} - p_E^{'}\right), \\ \left(A_P - \sum_{nb} A_{nb}\right) v_{2P}^{'} = \sum_{nb} A_{nb} \left(v_{2nb}^{'} - v_{2P}^{'}\right) + \Delta x \left(p_P^{'} - p_N^{'}\right). \end{cases} \tag{3.16}$$

To introduce a consistent approximation, the terms $\sum_{nb} A_{nb} \left(v_{1nb}^{'} - v_{1P}^{'}\right)$ and $\sum_{nb} A_{nb} \left(v_{2nb}^{'} - v_{2P}^{'}\right)$ are neglected. Replacing $v_{1P}^{'}$ and $v_{2P}^{'}$ by $v_{1P} - v_{1P}^{*}$ and $v_{2P} - v_{2P}^{*}$ respectively, the Equations (3.16) become

$$\begin{cases} v_{1P} = v_{1P}^{*} + d_1 (p_P^{'} - p_E^{'}), \\ v_{2P} = v_{2P}^{*} + d_2 (p_P^{'} - p_N^{'}), \end{cases} \tag{3.17}$$

where

$$d_1 = \Delta y / \big(A_P - \sum_{nb} A_{nb}\big), \quad d_2 = \Delta x / \big(A_P - \sum_{nb} A_{nb}\big).$$

Using the fact that v_{1P} and v_{2P} satisfy the Equation (3.17), if we replace these expressions in Equation (3.4), we obtain the following correction pressure equation:

$$A_P p_P^{'} = A_E p_E^{'} + A_W p_W^{'} + A_N p_N^{'} + A_S p_S^{'} + b_p, \tag{3.18}$$

where

$$A_E = \Delta y \mathrm{R}_e d_1, \quad A_W = \Delta y \mathrm{R}_e d_1, \quad A_N = \Delta x \mathrm{R}_e d_2, \quad A_S = \Delta x \mathrm{R}_e d_2,$$
$$A_P = A_E + A_W + A_N + A_S, \quad b_p = \Delta y \mathrm{R}_e \left((v_1^*)_w - (v_1^*)_e\right) + \Delta x \mathrm{R}_e \left((v_2^*)_s - (v_2^*)_n\right).$$

To solve these equations we use the algebrical equation form to obtain the fields verifying the conservation equations [26]. Finally, we summarize the proposed method in the Algorithm 1.

Algorithm 1 The proposed finite volume algorithm for image registration

Input: v_1^*, v_2^* and the pressure field p^*, the Reynolds number.
Output: The velocities v_1, v_2 and the pressure p.

1: **procedure**
2: Solve the momentum equations (3.14) to obtain v_1^* and v_2^*.
3: Solve the Equation (3.18) for $p^{'}$.
4: Calculate p by adding $p^{'}$ to p^*.
5: Calculate v_1 and v_2 from their previous values using the velocity-correction formula (3.17).
6: Treat the corrected pressure p as a new guessed pressure p^*, return to step 2.
7: Repeat the whole procedure until convergence is reached.

3.2 Approximation of displacement u

The next step is to compute the displacement $u = (u_1, u_2)$ from the associated velocities v_1 and v_2, calculated through Algorithm 1, based on (2.3) and an Euler scheme [1]. For each grid point $x_i \in \mathbb{R}^2$ with a fixed index $i = (i_1, i_2) \in \mathbb{N}^2$ we have $u^k(x_i) \in \mathbb{R}^2$, and we set

$$u_i^k = \left(u_i^{k,1}, u_i^{k,2}\right) \in \mathbb{R}^2,$$

where $u_i^{k,1}$ and $u_i^{k,2}$ are, respectively, the first and second component approximations of u. We also set

$$v_i^k = \left(v_i^{k,1}, v_i^{k,2}\right) \in \mathbb{R}^2,$$

which is the velocity approximation, where $v_i^{k,1}$ is the first component and $v_i^{k,2}$ is the second one.

As the first step, to approximate the ∇u term, a centred finite difference approximation is used to compute the Jacobian matrix J_i^k of u_i^k. Secondly, for the partial time derivative $\partial_t u$, we use a forward finite difference approximation. Therefore, the displacement and the velocity are connected through the following Euler scheme which is performed for all i

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \left(I_d - J_i^k\right) v_i^k. \tag{3.19}$$

The proposed algorithm to compute the transformations u is finally summarized in Algorithm 2.

Algorithm 2 The proposed algorithm for displacement u computation

Input: v_1^k, v_2^k, u_1^k, u_2^k, the time-step $\tau = 0.001$.
Output: The displacements u_1^{k+1}, u_2^{k+1}.

1: **procedure**
2: Compute of the Jacobian Matrix J_i^k.
3: Solve the Equations (3.19) to obtain u_1^{k+1} and u_2^{k+1}.
4: If the relative change of the distance measure is brought below a user supplied tolerance $tol = 10^{-5}$, the iteration is stopped,

$$\frac{\left\|R - T^k\right\|_F - \left\|R - T^{k+1}\right\|_F}{\left\|R - T^{k+1}\right\|_F} \leq tol.$$

5: If else, calculate the new v_1^{k+1} and v_2^{k+1} from the Algorithm 1, then return to step 2.

4 Results and discussion

In this section, we test the performance of the proposed image registration model. In fact, we have tested our algorithm on a large image registration

Left: *Example 1* (64 x 64) and right: *Example 2* (256 x 256)

Left: *Example 3* (256 x 256) and right: *Example 4* (512 x 512)

Figure 2. Four representative data sets of registration problems. Left column: Reference image, right column: Template image.

benchmark. We present only four tests chosen with different size and deformation. To measure the robustness of the proposed algorithm, we compare it with some competitive approaches, including the fluid registration model $\mathcal{R}^{Fluid}$ [11], the modified total variation (TV) regularization $\mathcal{R}^{MTV}$ [17], the SIMPLE model [1] and also the improved discontinuity-preserving model $\mathcal{R}^{IDP}$ proposed in [30]. In the Figure 2, we present the reference and template image for the four used tests.

Our aim is to find the deformation between the template and the reference images, and recovering the reference image by registering the template one. In Figures 3–6, we represent the registered template image using the proposed approach compared with the other methods using the difference between the reference and template images and also the deformed grid for each test.

If we focus on what happens in the error between the template image and the reference one, in all examples, we can see that the proposed method is more efficient than the other image registration methods. To evaluate the performance of the proposed approach with respect to noise reduction, we use two measures such as peak signal to noise ratio (PSNR) and the structural similarity (SSIM). The PSNR is a popular metric used to measure the quality of the estimated image, while the SSIM is a complementary measure, which gives an indication of image quality based on known characteristics of the human visual system [31]. The PSNR measures signal strength relative to noise in the image and is defined by

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right),$$

where the MSE is the mean squared error defined by

$$MSE = \frac{1}{MN}\Sigma_{i=1}^{M}\Sigma_{j=1}^{N}(Y(i,j) - X(i,j))^2.$$

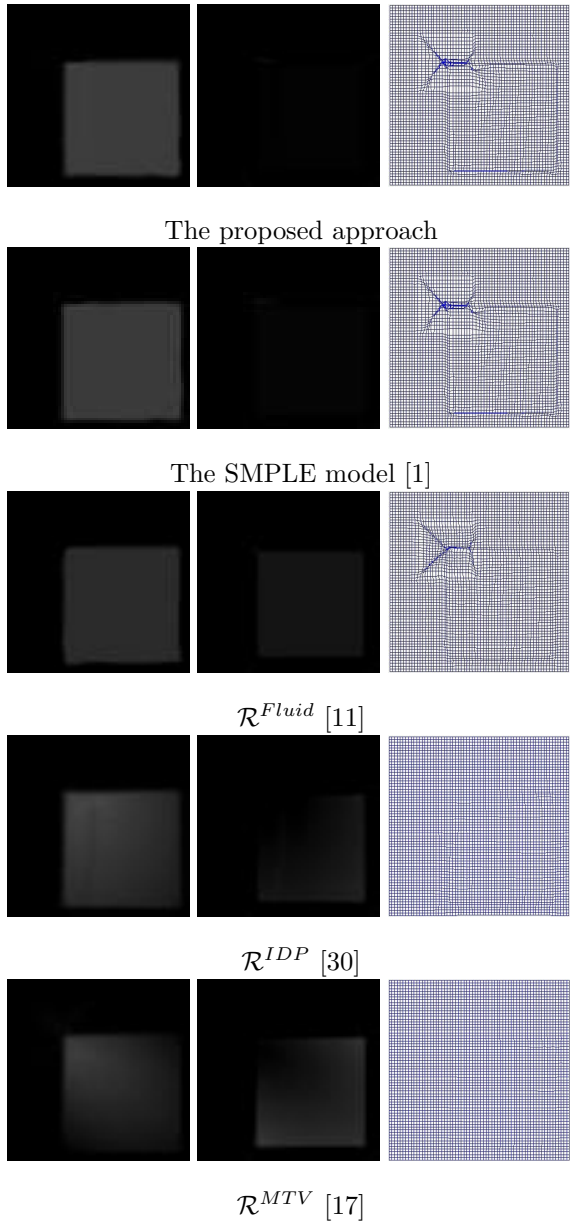
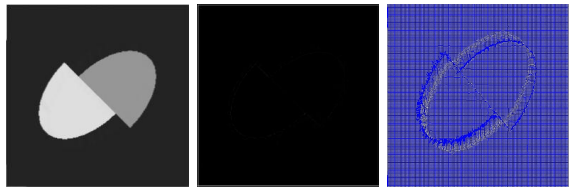
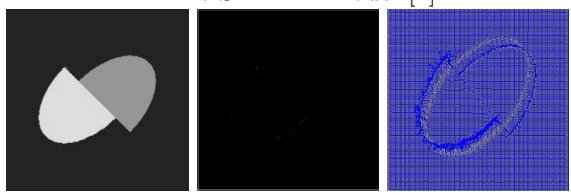


Figure 3. In each row we present the obtained result by each method for *Example 1*; left: the registered template image; the middle: difference between reference and obtained image and in the right column we plot the deformation field.

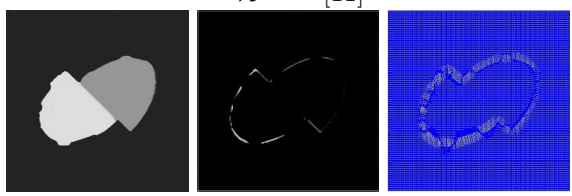


The proposed approach

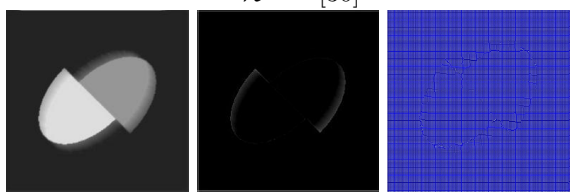
The SIMPLE model [1]



$\mathcal{R}^{Fluid}$ [11]



$\mathcal{R}^{IDP}$ [30]



$\mathcal{R}^{MTV}$ [17]

Figure 4. In each row we present the obtained result by each method for *Example 2*; left: the registered template image; the middle: difference between reference and obtained image and in the right column we plot the deformation field.

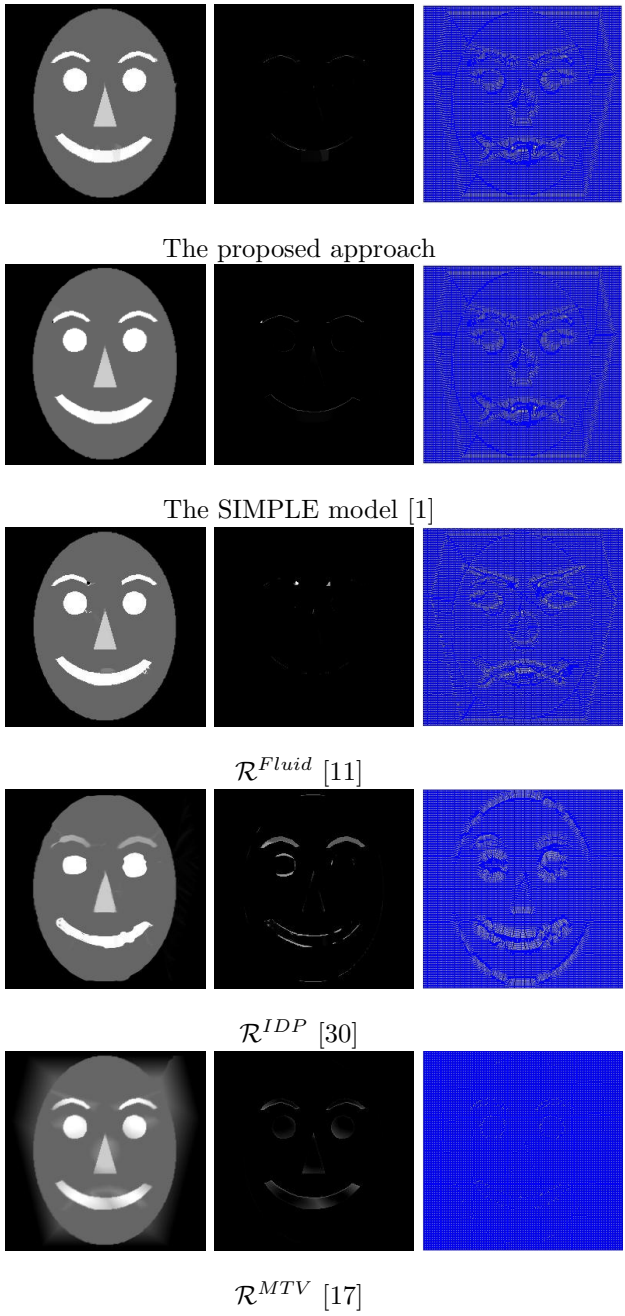


Figure 5. In each row we present the obtained result by each method for *Example 3*; left: the registered template image; the middle: difference between reference and obtained image and in the right column we plot the deformation field.

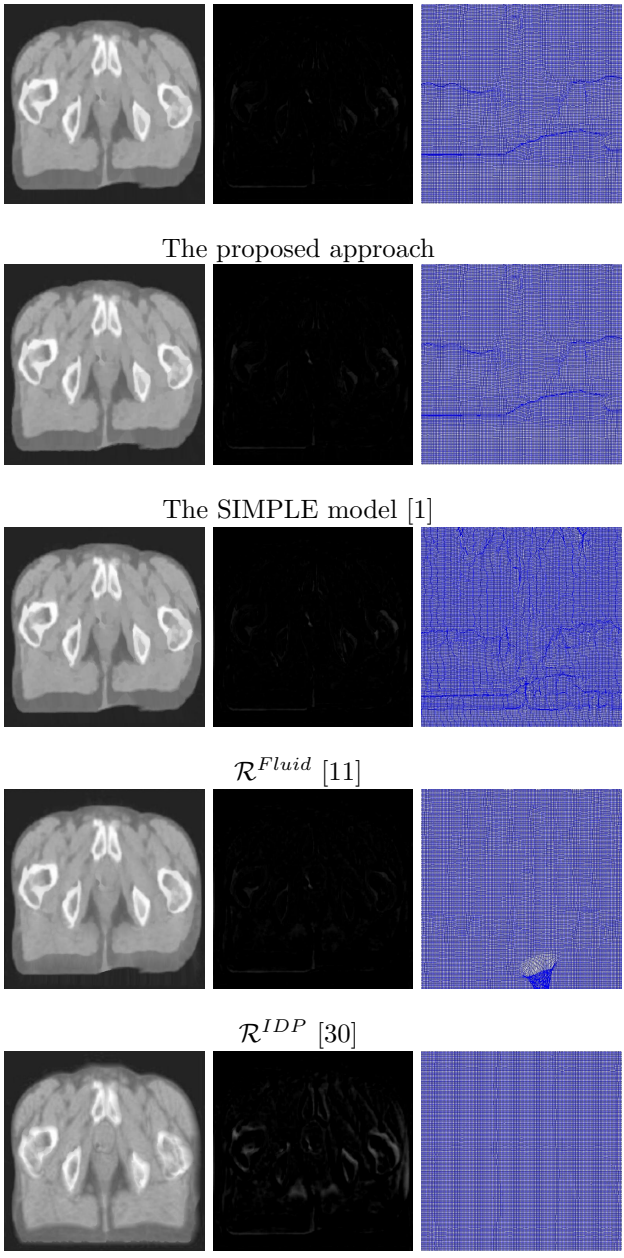


Figure 6. In each row we present the obtained result by each method for *Example 4*; left: the registered template image; the middle: difference between reference and obtained image and in the right column we plot the deformation field.

The $SSIM$ is calculated on multiple windows of given image, i.e. the measurement between two windows x and y of size $N \times N$ is defined by

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)(2\,cov_{xy} + c_3)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)(\sigma_x\sigma_y + c_3)},$$

where the variables, respectively, defined for x and y as follows: μ_x and μ_y, mean; σ_x^2 and σ_y^2, variance; cov_{xy}, covariance; $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ are two stabilizing constants; and L the dynamics of the pixel values, 255 for 8-bit encoded image. This metric gives an indication on the quality of the image based on the known characteristics of human visual system. Moreover, to measure the quality of the registered images, the relative reduction of the dissimilarity $\mathrm{rel} \cdot SSD$ is used

$$\mathrm{rel} \cdot SSD = \frac{D(u)}{D_{\mathrm{stop}}} \times 100\%,$$

where u is the current optimal value and D_{stop} is the value of $D(u)$ at $u = 0$.

In Table 1, the SSIM, PSNR and $\mathrm{rel} \cdot SSD$ values are calculated for all the used images. The best results are represented by a bold number. Always, the proposed method outperforms the other in terms of both PSNR and SSIM.

Table 1. PSNR, SSIM and $\mathrm{rel} \cdot SSD$ results obtained using the fluid image registration and proposed approach to the four above examples. In *bold* the highest value of each row is shown.

Image		Method				
	Metric	$\mathcal{R}^{MTV}$	$\mathcal{R}^{IDP}$	$\mathcal{R}^{Fluid}$	Model [1]	proposed
Example 1	PSNR	22.7103	27.4564	26.1098	32.1215	32.5905
64×64	SSIM	0.7782	0.9027	0.9003	0.9482	0.9542
	rel.SSD	88.14%	59.43%	88.98%	9.20%	2.81%
Example 2	PSNR	22.6853	21.3302	27.2033	34.8440	27.5385
256×256	SSIM	0.9470	0.9458	0.9568	0.9899	0.9583
	rel.SSD	80.56%	55.92%	12.54%	7.02%	4.32%
Example 3	PSNR	20.4743	22.5388	30.0480	28.8180	30.3619
256×256	SSIM	0.6765	0.8408	0.9470	0.9544	0.9527
	rel.SSD	87.40%	62.61%	16.78%	16.60%	6.42%
Example 4	PSNR	21.3706	32.3857	33.4616	33.5446	33.5525
512×512	SSIM	0.6912	0.9297	0.9478	0.9483	0.9486
	rel.SSD	73.16%	22.14%	18.47%	18.45%	16.89%

The full code for the proposed model is implemented in the MATLAB 2013. Typically, the execution of the main implemented programme requires an average of $1 \sim 5$ minutes on a 3 GHz Pentium Quad core computer for 256×256 grey-scale images; the runtime increases for larger image sizes. While the execution of the fluid image registration takes about $2 \sim 7$ minutes in the same

conditions. However, for the two other regularizations : $\mathcal{R}^{IDP}$ and $\mathcal{R}^{MTV}$, the execution time is sometimes less compared to that recorded by our method.

Finally, we want to check the speed of the convergence during time iterations of the proposed approach compared to the SIMPLE model [1]. To do that, we run the two codes for 500 time iterations for the *Example 3* registration process and we compute the relative error in each iteration. The obtained results are presented in Figure 7 which shows that the approach based on SIMPLEC algorithm converge faster than the one based on SIMPLE [1].

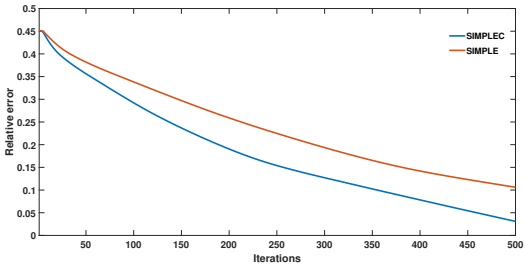


Figure 7. The relative error for the registration process of the *Example 3* using the proposed approach and compared to the SIMPLE model [1].

5 Conclusions

A consistent numerical scheme for the fluid image registration based on Navier Stokes equation was introduced in the image registration context. A finite volume-based scheme using the SIMPLEC algorithm was performed to avoid the errors arising from the discretization part. The performance of this approach has been tested using different examples. The proposed approach outperforms some competitive ones both visually and using different criteria.

Acknowledgements

We are grateful to the anonymous referee for the corrections and useful suggestions that have improved this article.

References

[1] M. Alahyane, A. Hakim, A. Laghrib and S. Raghay. Fluid image registration using a finite volume scheme of the incompressible Navier Stokes equation. *Inverse Problems and Imaging*, **12**(5):1055–1081, 2018. https://doi.org/10.3934/ipi.2018044.

[2] M. Alahyane, A. Hakim, A. Laghrib and S. Raghay. A fast approach of nonparametric elastic image registration problem. *Mathematical Methods in the Applied Sciences*, **42**(18):7059–7075, 2019. https://doi.org/10.1002/mma.5810.

[3] M. Alahyane, A. Hakim, A. Laghrib and S. Raghay. A lattice Boltzmann method applied to the fluid image registration. *Applied Mathematics and Computation*, **349**:421–438, 2019. https://doi.org/10.1016/j.amc.2018.12.051.

[4] C. Broit. *Optimal registration of deformed images*. University of Pennsylvania, 1981.

[5] M. Burger, J. Modersitzki and L. Ruthotto. A hyperelastic regularization energy for image registration. *SIAM Journal on Scientific Computing*, **35**(1):B132–B148, 2013. https://doi.org/10.1137/110835955.

[6] G.E. Christensen. Deformable shape models for anatomy, 1994.

[7] G.E. Christensen. WE-H-202-04: Advanced medical image registration techniques. *Medical Physics*, **43**(6):3845–3845, 2016. https://doi.org/10.1118/1.4958005.

[8] G.E. Christensen, R.D. Rabbitt and M.I. Miller. A deformable neuroanatomy textbook based on viscous fluid mechanics. In *27th Ann. Conf. on Inf. Sciences and Systems*, pp. 211–216. Citeseer, 1993.

[9] G.E. Christensen, R.D. Rabbitt and M.I. Miller. Deformable templates using large deformation kinematics. *IEEE transactions on image processing*, **5**(10):1435–1447, 1996. https://doi.org/10.1109/83.536892.

[10] N. Chumchob and K. Chen. A variational approach for discontinuity-preserving image registration. *Proceedings of ICMA-CU*, pp. 266–282, 2010.

[11] E. D'Agostino, F. Maes, D. Vandermeulen and P. Suetens. A viscous fluid model for multimodal non-rigid image registration using mutual information. *Medical image analysis*, **7**(4):565–575, 2003. https://doi.org/10.1016/S1361-8415(03)00039-2.

[12] J.P. Van Doormaal and G.D. Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical heat transfer*, **7**(2):147–163, 1984. https://doi.org/10.1080/01495728408961817.

[13] R. Eymard, T. Gallouët and R. Herbin. *Finite volume methods*, volume 7. Elsevier, 2000.

[14] J.H. Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.

[15] B. Fischer and J. Modersitzki. Ill-posed medicine-an introduction to image registration. *Inverse Problems*, **24**(3):034008, 2008. https://doi.org/10.1088/0266-5611/24/3/034008.

[16] C. Foias, O. Manley, R. Rosa and R. Temam. *Navier-Stokes equations and turbulence*, volume 83. Cambridge University Press, 2001.

[17] C. Frohn-Schauf, S. Henn and K. Witsch. Multigrid based total variation image registration. *Computing and Visualization in Science*, **11**(2):101–113, 2008. https://doi.org/10.1007/s00791-007-0060-2.

[18] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations: theory and algorithms*, volume 5. Springer Science & Business Media, 2012.

[19] E. Haber and J. Modersitzki. Numerical methods for volume preserving image registration. *Inverse problems*, **20**(5):1621, 2004. https://doi.org/10.1088/0266-5611/20/5/018.

[20] H. Han. A fractional-order decomposition model of image registration and its numerical algorithm. *Computational and Applied Mathematics*, **39**(2):1–19, 2020. https://doi.org/10.1007/s40314-020-1066-3.

[21] A. Laghrib and A. Hakim S. Raghay M. El Rhabi. Robust super resolution of images with non-parametric deformations using an elastic registration. *Appl. Math. Sci*, **8**:8897–8907, 2014. https://doi.org/10.12988/ams.2014.49751.

[22] A. Laghrib, A. Ghazdali, A. Hakim and S. Raghay. A multi-frame super-resolution using diffusion registration and a nonlocal variational image restoration. *Computers & Mathematics with Applications*, **72**(9):2535–2548, 2016. https://doi.org/10.1016/j.camwa.2016.09.013.

[23] A. Mang and G. Biros. An inexact Newton–Krylov algorithm for constrained diffeomorphic image registration. *SIAM journal on imaging sciences*, **8**(2):1030–1069, 2015. https://doi.org/10.1137/140984002.

[24] T. Mansi, X. Pennec, M. Sermesant, H. Delingette and N. Ayache. iLogDemons: A demons-based registration algorithm for tracking incompressible elastic biological tissues. *International journal of computer vision*, **92**(1):92–111, 2011. https://doi.org/10.1007/s11263-010-0405-z.

[25] J. Modersitzki. *FAIR: flexible algorithms for image registration*, volume 6. SIAM, 2009.

[26] S. Patankar. *Numerical heat transfer and fluid flow*. CRC press, 1980.

[27] S.V. Patankar. A calculation procedure for two-dimensional elliptic situations. *Numerical Heat Transfer*, **4**(4):409–425, 1981. https://doi.org/10.1080/01495728108961801.

[28] L.I. Rudin, S. Osher and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, **60**(1-4):259–268, 1992.

[29] R. Temam. *Navier-Stokes equations: theory and numerical analysis*, volume 343. American Mathematical Soc., 2001.

[30] J. Zhang, K. Chen and B. Yu. An improved discontinuity-preserving image registration model and its fast algorithm. *Applied Mathematical Modelling*, **40**(23-24):10740–10759, 2016. https://doi.org/10.1016/j.apm.2016.08.009.

[31] W. Zhou, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, **13**:600–612, April 2004. https://doi.org/10.1109/TIP.2003.819861.