



HAL
open science

A Splitting in Time Scheme and Augmented Lagrangian Method for a Nematic Liquid Crystal Problem

F. Guillén González, Jonas Koko

► **To cite this version:**

F. Guillén González, Jonas Koko. A Splitting in Time Scheme and Augmented Lagrangian Method for a Nematic Liquid Crystal Problem. *Journal of Scientific Computing*, 2015. hal-02081397

HAL Id: hal-02081397

<https://hal.science/hal-02081397v1>

Submitted on 27 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Splitting in Time Scheme and Augmented Lagrangian Method for a Nematic Liquid Crystal Problem

F. GUILLÉN-GONZÁLEZ *

Departamento de Ecuaciones Diferenciales y Análisis Numérico & IMUS

Universidad de Sevilla, Aptdo. 1160, E-41080 Sevilla, Spain

J. KOKO

ISIMA/LIMOS, Université Blaise Pascal – CNRS UMR 6158

Campus des Cézeaux – BP 10125, F-63173 Aubière cedex, France

December 18, 2014

Abstract

We study the numerical approximation of nematic liquid crystal flows governed by a Ericksen-Leslie problem. This problem couples the incompressible Navier-Stokes dynamic with a gradient flow system related to the orientation unitary vector of molecules. First, a two sub-step viscosity-splitting time scheme is proposed. The first sub-step couples diffusion and convection terms whereas the second one is concerned with diffusion terms and constraints (divergence free and unit director field). Then, in the first sub-step we use a Gauss-Seidel decoupling algorithm, and in the second sub-step, we use Uzawa type algorithms on augmented Lagrangian functionals to overcome the divergence free constraint and the unit director field constraint. From the computational point of view, it is a fully decoupled linear scheme (where all systems to solve are for scalar variables). Some numerical experiments in 2D are carried out by using only linear finite elements in space, confirming the viability and the convergence of our scheme.

Keywords: Ericksen-Leslie’s nematic model, splitting in time schemes, Augmented Lagrangian, mixed formulation.

*F. G-G. is partially supported by the Spanish project MTM2012-32325 and also acknowledges support by the University of Clermont-Ferrand II during his stay as a “Professeur invité”.

1 Introduction

The liquid crystal model

We consider the following simplified Ericksen-Leslie's problem in $Q = (0, T) \times \Omega$ modeling a nematic liquid crystal fluid filling a bounded domain $\Omega \subset \mathbb{R}^N$ ($N = 2$ or 3 in practice) with boundary $\partial\Omega$ and $\Sigma = (0, T) \times \partial\Omega$:

$$\partial_t d + (u \cdot \nabla)d - \gamma \Delta d + q d = 0, \quad (1)$$

$$|d| = 1, \quad (2)$$

$$\partial_t u + (u \cdot \nabla)u - \nu \Delta u + \nabla p + \lambda (\nabla d)^t \Delta d = f, \quad (3)$$

$$\nabla \cdot u = 0, \quad (4)$$

$$u|_{\Sigma} = 0, \quad d|_{\Sigma} = l, \quad (5)$$

$$u|_{t=0} = u_0, \quad d|_{t=0} = d_0. \quad (6)$$

The unknowns of this problem are:

- $d : Q \rightarrow \mathbb{R}^N$ the orientation of liquid crystal molecules with $q : Q \rightarrow \mathbb{R}$ the Lagrange multiplier associated with the unitary constraint (2),
- $u : Q \rightarrow \mathbb{R}^N$ the fluid velocity and $p = \tilde{p} + \lambda |\nabla d|^2 / 2$ the Lagrange multiplier associated with the incompressibility constraint (4), where $\tilde{p} : Q \rightarrow \mathbb{R}$ is the pressure.

The data are:

- $f : Q \rightarrow \mathbb{R}^N$ the external force,
- $l : \Sigma \rightarrow \mathbb{R}^N$ the Dirichlet boundary data for d , and
- $u_0, d_0 : \Omega \rightarrow \mathbb{R}^N$ the initial data.

Finally, $\gamma, \nu, \lambda > 0$ are given constants. Hereafter, $|d|$ and $|\nabla d|$ denotes the euclidean norm of the vector $d \in \mathbb{R}^N$ or the matrix $\nabla d \in \mathbb{R}^{N \times N}$, respectively. $(\nabla d)^t$ denotes the transpose matrix of ∇d .

Eventually, the Dirichlet condition $d|_{\Sigma} = l$ can be changed by the homogeneous Neumann condition

$$\partial_n d|_{\Sigma} = 0$$

where n is the unit outward normal to $\partial\Omega$.

It is easy to deduce, multiplying (1) by d and applying (2), that $q = -\gamma |\nabla d|^2$, hence the Lagrange multiplier q has an explicit expression in function of d . Nevertheless, we are going to keep the unknown q because in the fully-discrete numerical schemes this explicit expression will be lost.

Known results

First numerical results related to problem (1)-(6) are based on the discretization of a penalized problem of (1)-(6) by means of a Ginzburg-Landau functional depending on the penalized parameter ε (see for instance [5, 15, 16, 14, 9, 10]). All these schemes suffer from the disadvantage of being sensitive to the choice of the penalty parameter ε .

In [3], *Becker, Feng* and *Prohl* considered two nonlinear fully discrete C^0 -finite element methods. The first scheme discretizes the Ginzburg-Landau penalized problem and it is unconditionally energy-stable (conserving decreasing the energy associated with the continuous problem) and converges towards the penalized problem. The second algorithm in [3] discretizes directly (1)-(6) and is unconditionally energy-stable, although the convergence when the discrete time and space parameters go to zero remains as an open problem. This second algorithm is based on a reformulation of the d -system (1) by using vectorial products, arriving at a non-linear and fully coupled scheme, which is implemented in practice via the iterative Newton's method.

On the other hand, a finite element scheme based on a saddle-point formulation of the director vector is proposed in [1], allowing to consider the limit problem (1)-(6) and the penalized problem (using a Ginzburg-Landau functional) in a unified way. In this case, a linear time semi-implicit algorithm is introduced which is unconditionally stable (satisfying a discrete energy inequality), although the resulting scheme is fully coupled.

Finally, for the director vector only problem (i.e. problem (1)-(2) with $u = 0$), *Glowinski, Lin* and *Pan* describe in [5], a splitting in time scheme based on the Chorin-Temam projection method for fluids. Moreover, a scheme is given for the limit problem (1)-(2) and another one for the corresponding penalized version.

Novelty of the paper

In this paper we design a new fully discrete algorithm approximating directly the problem (1)-(6), which is completely different from the second scheme presented in [3] and the schemes given in [5, 1]. This new scheme is based on a viscosity-splitting scheme in time, an iterative fixed-point method for the coupled non-symmetric problem without constraint and the augmented Lagrangian algorithm in space associated with two symmetric mixed problems:

- a Stokes type problem for the fluid part;
- a director vector problem with the unitary constraint, introducing the Lagrange multiplier associated to the non-convex constraint $|d| = 1$.

Then, our scheme decouples the Lagrange multipliers with respect to the primal variables as velocity and director vector. Moreover, the vectorial systems for the velocity and director vector will be decoupled by components. To our knowledge, this is the first fully decoupled

linear scheme applied to the Ericksen-Leslie's problem (1)-(6). Some numerical computations are presented performing the convergence and viability of this new scheme.

Organization of the paper

The rest of the paper is organized as follows. In Section 2 a first time scheme is presented, splitting the convection nonlinear terms from the constraints (incompressibility and director vector in the unit sphere). Then, fully decoupled strategies for the three different problems appearing in the previous time-splitting scheme will be described in Sections 3, 4 and 5, respectively. Finally, the numerical results are presented in Section 6, by using only linear finite elements in space.

2 A viscosity-splitting time discrete scheme

The time interval $[0, T]$ is divided into M subintervals of equal length $k = T/M$ (by simplicity), considering the partition $\{t_n = nk\}_{n=1}^M$. We would like to compute $\{u^n, p^n, d^n, q^n\}$ as approximations of $\{u(t_n), p(t_n), d(t_n), q(t_n)\}$.

For this, let us start with the following time discrete scheme, which is related to the so-called viscosity-splitting algorithm, see [2, 11, 12] for the Navier-Stokes case:

Initialization. Let $u^0 = u_0$ and $d^0 = d_0$.

Step $n \geq 1$. Given (u^{n-1}, d^{n-1}) , we compute (u^n, p^n, d^n, q^n) via the following two sub-step scheme.

Sub-step 1. Given (u^{n-1}, d^{n-1}) , compute $(\tilde{u}^n, \tilde{d}^n)$ as the solution of the coupled linear system

$$\frac{1}{k}(\tilde{d}^n - d^{n-1}) + (\tilde{u}^n \cdot \nabla)d^{n-1} - \gamma \Delta \tilde{d}^n = 0, \quad (7)$$

$$\begin{aligned} \frac{1}{k}(\tilde{u}^n - u^{n-1}) + (u^{n-1} \cdot \nabla)\tilde{u}^n + \frac{1}{2}(\nabla \cdot u^{n-1})\tilde{u}^n \\ - \nu \Delta \tilde{u}^n + \lambda(\nabla d^{n-1})^t \Delta \tilde{d}^n = f^n. \end{aligned} \quad (8)$$

Sub-step 2. Given $(\tilde{u}^n, \tilde{d}^n)$, compute (u^n, p^n) and (d^n, q^n) as the solution of the following two uncoupled systems

$$(u^n, p^n) \quad s.t. \quad \frac{1}{k}(u^n - \tilde{u}^n) - \nu \Delta(u^n - \tilde{u}^n) + \nabla p^n = 0, \quad \nabla \cdot u^n = 0, \quad (9)$$

$$(d^n, q^n) \quad s.t. \quad \frac{1}{k}(d^n - \tilde{d}^n) - \gamma \Delta(d^n - \tilde{d}^n) + q^n d^n = 0, \quad |d^n| = 1, \quad (10)$$

Obviously, systems (7)-(8), (9) and (10) must be endowed with the exact boundary conditions related to (5):

$$\tilde{u}^n|_{\partial\Omega} = 0 = u^n|_{\partial\Omega} \quad \text{and} \quad \tilde{d}^n|_{\partial\Omega} = l(t_n) = d^n|_{\partial\Omega}.$$

In order to describe a fully decoupled the above time discretization scheme, we will analyze separately the three different previous subproblems: the coupled problem (7)-(8) of sub-step 1, and the two decoupled problems (9) and (10) of sub-step 2.

3 Solution of the problem (7) – (8)

For simplicity, in this Section we will use the notations $u = \tilde{u}^n$, $d = \tilde{d}^n$, $u^* = u^{n-1}$, $d^* = d^{n-1}$ and $f = f^n$. Hence, given (u^*, d^*) , the problem (7)-(8) can be rewritten as:

Find (u, d) such that

$$\frac{1}{k}(u - u^*) + (u^* \cdot \nabla)u + \frac{1}{2}(\nabla \cdot u^*)u - \nu \Delta u + \lambda(\nabla d^*)^t \Delta d = f. \quad (11)$$

$$\frac{1}{k}(d - d^*) + u \cdot \nabla d^* - \gamma \Delta d = 0. \quad (12)$$

As in [9], we subtract from d a “lifting” function \bar{d}^n as a solution of the Laplace-Dirichlet problem

$$\begin{aligned} \Delta \bar{d}^n &= 0, \quad \text{in } \Omega, \\ \bar{d}^n &= l(t_n), \quad \text{on } \partial\Omega. \end{aligned}$$

If we set

$$\hat{d} = d - \bar{d}^n,$$

then \hat{d} is the solution of the problem $\Delta \hat{d} = \Delta d$ and $\hat{d} = 0$ on $\partial\Omega$. Again as in [9], we also introduce the auxiliary variable

$$w = -\sqrt{\lambda} \Delta d = -\mu \Delta d = -\mu \Delta \hat{d},$$

where $\mu = \sqrt{\lambda}$. Then (u, d) is a solution of (11)-(12) if, and only if, $(u, d = \hat{d} + \bar{d}^n, w)$ is a solution of the coupled system

$$\frac{1}{k}(u - u^*) + (u^* \cdot \nabla)u + \frac{1}{2}(\nabla \cdot u^*)u - \nu \Delta u - \mu(\nabla d^*)^t w = f. \quad (13)$$

$$\frac{\mu}{k}(d - d^*) + \mu(u \cdot \nabla)d^* + \gamma w = 0, \quad (14)$$

$$\mu \Delta \hat{d} + w = 0. \quad (15)$$

To introduce the variational formulation of (13)-(15), we define the following trilinear form

$$c(u^*, u, \bar{u}) = ((u^* \cdot \nabla)u, \bar{u}) + \frac{1}{2}(\nabla \cdot u^*, u \cdot \bar{u}), \quad \forall u^*, u, \bar{u} \in H_0^1(\Omega)^N.$$

Hereafter we denote by (\cdot, \cdot) the scalar product in $L^2(\Omega)$. The variational formulation of (13)-(15) can be written as follows:

Find $(u, d = \widehat{d} + \bar{d}^n, w) \in H_0^1(\Omega)^N \times H^1(\Omega)^N \times L^2(\Omega)^N$ such that

$$\frac{1}{k}(u - u^*, \bar{u}) + \nu(\nabla u, \nabla \bar{u}) + c(u^*, u, \bar{u}) - \mu((\nabla d^*)^t w, \bar{u}) = (f, \bar{u}), \quad \forall \bar{u} \in H_0^1(\Omega)^N, \quad (16)$$

$$\frac{\mu}{k}(d - d^*, \bar{w}) + \mu((u \cdot \nabla) d^*, \bar{w}) + \gamma(w, \bar{w}) = 0, \quad \forall \bar{w} \in L^2(\Omega)^N, \quad (17)$$

$$\mu(\nabla \widehat{d}, \nabla \bar{d}) - (w, \bar{d}) = 0, \quad \forall \bar{d} \in H_0^1(\Omega)^N. \quad (18)$$

To solve (16)-(18), we use a successive iterative method studied in [9] (see Algorithm 1). Note that problem (19) in Algorithm 1 is equivalent to N decoupled convection-diffusion equations, one for each component of the velocity field $u^{(\ell)}$. Hence N linear non-symmetric systems depending on the time step must be solved. On the other hand, problem (20) in Algorithm 1 decouples each component of the pair $(\widehat{d}^{(\ell+1)}, w^{(\ell+1)})$. Moreover, since $w^{(\ell+1)}$ will be approximated by P_0 finite elements, we can eliminate each component of $w^{(\ell+1)}$ from $(20)_1$ in function of the corresponding component of $\widehat{d}^{(\ell+1)}$ in $(20)_2$ by the so-called ‘‘static condensation’’ process. Then, problem (20) is equivalent to solving N linear symmetric systems independent from the time step.

Algorithm 1 Successive iterative algorithm for the coupled problem (7)-(8)

Initialization: Let $w^{(0)}$ be given ($w^{(0)} = w^{n-1}$ if $n \geq 2$ or $w^{(0)} = -\mu \Delta \widehat{d}^0$ if $n = 1$).

Step $\ell \geq 0$: Let $w^{(\ell)} \in L^2(\Omega)^N$ be given.

1. Compute $u^{(\ell)} \in H_0^1(\Omega)^N$ such that for all $\bar{u} \in H_0^1(\Omega)^N$:

$$\frac{1}{k}(u^{(\ell)}, \bar{u}) + \nu(\nabla u^{(\ell)}, \nabla \bar{u}) + c(u^*, u^{(\ell)}, \bar{u}) = (f, \bar{u}) + \frac{1}{k}(u^*, \bar{u}) + \mu((\nabla d^*)^t w^{(\ell)}, \bar{u}). \quad (19)$$

2. Compute $(\widehat{d}^{(\ell+1)}, w^{(\ell+1)}) \in H_0^1(\Omega)^N \times L^2(\Omega)^N$ such that for all $(\bar{d}, \bar{w}) \in H_0^1(\Omega)^N \times L^2(\Omega)^N$:

$$\begin{cases} \frac{\mu}{k}(\widehat{d}^{(\ell+1)}, \bar{w}) + \gamma(w^{(\ell+1)}, \bar{w}) = \frac{\mu}{k}(d^* - \bar{d}^n, \bar{w}) - \mu(u^{(\ell)} \cdot \nabla d^*, \bar{w}), \\ \mu(\nabla \widehat{d}^{(\ell+1)}, \nabla \bar{d}) - (w^{(\ell+1)}, \bar{d}) = 0. \end{cases} \quad (20)$$

3. We stop iterating (19)-(20) as soon as

$$\frac{\|u^{(\ell)} - u^{(\ell-1)}\|_{L^2}^2 + \|d^{(\ell)} - d^{(\ell-1)}\|_{L^2}^2 + \|w^{(\ell)} - w^{(\ell-1)}\|_{L^2}^2}{\|u^{(\ell)}\|_{L^2}^2 + \|d^{(\ell)}\|_{L^2}^2 + \|w^{(\ell)}\|_{L^2}^2} \leq \varepsilon^2. \quad (21)$$

4 Solution of the problem (9)

By using the notations $u = u^n$ and $p = p^n$, the problem (9) is rewritten as:

Find (u, p) such that

$$\frac{1}{k}(u - \tilde{u}^n) - \nu(\Delta u - \nu\Delta\tilde{u}^n) + \nabla p = 0, \quad \nabla \cdot u = 0, \quad u|_{\partial\Omega} = 0. \quad (22)$$

Let us introduce the bilinear and linear forms

$$\begin{aligned} \mathbf{a}_1(u, \bar{u}) &= \frac{1}{k}(u, \bar{u}) + \nu(\nabla u, \nabla \bar{u}), \quad \forall u, \bar{u} \in H_0^1(\Omega)^N \\ \boldsymbol{\tau}_1(\bar{u}) &= \frac{1}{k}(\tilde{u}^n, \bar{u}) + \nu(\nabla \tilde{u}^n, \nabla \bar{u}), \quad \forall \bar{u} \in H_0^1(\Omega)^N \end{aligned}$$

and the (quadratic and convex) functional $F_1 : H_0^1(\Omega)^N \rightarrow \mathbb{R}$ defined by

$$F_1(\bar{u}) = \frac{1}{2}\mathbf{a}_1(\bar{u}, \bar{u}) - \boldsymbol{\tau}_1(\bar{u})$$

In (22), the pressure p is the Lagrange multiplier associated with the divergence constraint $\nabla \cdot u = 0$. Therefore, (22) is the (strong) saddle-point formulation of the following constrained minimization problem:

Find $u \in H_0^1(\Omega)^N$ with $\nabla \cdot u = 0$ such that:

$$F_1(u) \leq F_1(\bar{u}), \quad \forall \bar{u} \in H_0^1(\Omega)^N \text{ with } \nabla \cdot \bar{u} = 0. \quad (23)$$

The saddle-point (u, p) of the constrained problem (23) can be computed using Uzawa/conjugate gradient algorithm operating in the space $H^1(\Omega) \cap L_0^2(\Omega)$ for the pressure, where

$$L_0^2(\Omega) = \left\{ p \in L^2(\Omega), \int_{\Omega} p dx = 0 \right\}.$$

This space $H^1(\Omega) \cap L_0^2(\Omega)$ is endowed with the scalar product $(\nabla p, \nabla q)$.

The resulting conjugate gradient algorithm is then preconditioned by the discrete operator equivalent of $(-\Delta)$, see e.g. [6, 7]. The corresponding Uzawa/conjugate gradient algorithm is described in Algorithm 2. Note that the vectorial problems (24) and (25) in Algorithm 2 are decoupled by components.

5 Solution of the problem (10)

Again for simplicity, in this Section we set $d = d^n$ and $q = q^n$. Hence, problem (10) is rewritten as:

Find (u, q) such that

$$\frac{1}{k}(d - \tilde{d}^n) - \gamma\Delta d + qd = -\gamma\Delta\tilde{d}^n, \quad |d| = 1, \quad d|_{\partial\Omega} = l^n. \quad (30)$$

To simplify we introduce the forms

$$\begin{aligned} \mathbf{a}_2(d, \bar{d}) &= \frac{1}{k}(d, \bar{d}) + \gamma(\nabla d, \nabla \bar{d}), \quad \forall d, \bar{d} \in H^1(\Omega)^N \\ \boldsymbol{\tau}_2(\bar{d}) &= \frac{1}{k}(\tilde{d}^n, \bar{d}) + \gamma(\nabla \tilde{d}^n, \nabla \bar{d}), \quad \forall \bar{d} \in H^1(\Omega)^N \end{aligned}$$

Algorithm 2 Uzawa/conjugate gradient algorithm for the Stokes problem (22)

Initialization: Let $p^{(0)} \in L_0^2(\Omega)$ be given ($p^{(0)} = p^{n-1}$ if $n \geq 2$ or $p^{(0)} = 0$).

1. Compute $u^{(0)} \in H_0^1(\Omega)^N$ via

$$\mathbf{a}_1(u^{(0)}, \bar{u}) = \boldsymbol{\tau}_1(\tilde{u}^{n-1}, \bar{u}) + (p^{(0)}, \nabla \cdot \bar{u}), \quad \forall \bar{u} \in H_0^1(\Omega)^N. \quad (24)$$

2. Compute $g^{(0)} \in L_0^2(\Omega)$ via

$$(\nabla g^{(0)}, \nabla \bar{g}) = (\nabla \cdot u^{(0)}, \bar{g}), \quad \forall \bar{g} \in L_0^2(\Omega).$$

3. Set $\bar{p}^{(0)} = g^{(0)}$

Step $\ell \geq 0$: Let $p^{(\ell)}, u^{(\ell)}, g^{(\ell)}, \bar{p}^{(\ell)}$ be given.

1. Compute $\bar{u}^{(\ell)} \in H_0^1(\Omega)^N$ via

$$\mathbf{a}_1(\bar{u}^{(\ell)}, \bar{u}) = (\bar{p}^{(\ell)}, \nabla \cdot \bar{u}), \quad \forall \bar{u} \in H_0^1(\Omega)^N. \quad (25)$$

2. Compute $\bar{g}^{(\ell)} \in L_0^2(\Omega)$ via

$$(\nabla \bar{g}^{(\ell)}, \nabla \bar{g}) = (\nabla \cdot \bar{u}^{(\ell)}, \bar{g}), \quad \forall \bar{g} \in L_0^2(\Omega). \quad (26)$$

3. Compute the step size

$$t^{(\ell)} = \frac{(\nabla g^{(\ell)}, \nabla \bar{g}^{(\ell)})}{(\bar{p}^{(\ell)}, \nabla \cdot \bar{u}^{(\ell)})}. \quad (27)$$

4. Update $p^{(\ell)}, g^{(\ell)}$ and $u^{(\ell)}$

$$p^{(\ell+1)} = p^{(\ell)} - t^{(\ell)} \bar{p}^{(\ell)}, \quad g^{(\ell+1)} = g^{(\ell)} - t^{(\ell)} \bar{g}^{(\ell)}, \quad u^{(\ell+1)} = u^{(\ell)} - t^{(\ell)} \bar{u}^{(\ell)}. \quad (28)$$

5. If $|\nabla g^{(\ell+1)}|_{L^2}^2 |\nabla g^{(0)}|_{L^2}^{-2} \leq \varepsilon^2$, then $p^n = p^{(\ell+1)}$ and $u^n = u^{(\ell+1)}$; else compute

$$\rho^{(\ell)} = \frac{|\nabla g^{(\ell+1)}|_{L^2}^2}{|\nabla g^{(\ell)}|_{L^2}^2}.$$

Compute the new conjugate direction:

$$\bar{p}^{(\ell+1)} = g^{(\ell+1)} + \rho^{(\ell)} \bar{p}^{(\ell)}. \quad (29)$$

6. We stop iterating (25)-(29) as soon as

$$\frac{(\nabla g^{(\ell)}, \nabla \bar{g}^{(\ell)})}{(\nabla g^{(0)}, \nabla \bar{g}^{(0)})} \leq \varepsilon.$$

and the (quadratic and convex) functional $F_2 : H^1(\Omega)^N \rightarrow \mathbb{R}$, defined by

$$F_2(\bar{d}) = \frac{1}{2} \mathbf{a}_2(\bar{d}, \bar{d}) - \boldsymbol{\tau}_2(\bar{d})$$

In (30), the term qd is the Lagrange multiplier associated with the restriction $|d| = 1$. Therefore, (30) is the (strong) saddle-point formulation of the following constrained minimization problem (with non-convex restriction):

Find $d \in H_{l^n}^1(\Omega)^N$ with $|d| = 1$, such that:

$$F_2(d) \leq F_2(\bar{d}), \quad \forall \bar{d} \in H_{l^n}^1(\Omega)^N \text{ with } |\bar{d}| = 1, \quad (31)$$

where $H_{l^n}^1(\Omega)^N$ is the affine subspace

$$H_{l^n}^1(\Omega)^N = \{d \in H^1(\Omega)^N : d|_{\partial\Omega} = l^n\}.$$

The augmented Lagrangian formulation combines the advantages of both the Lagrange multiplier method and the penalty method without suffering from disadvantage of either. Moreover, there is no need to increase the penalty parameter to infinity.

Let us introduce the set of point-wise constraint

$$C = \{d : d \in L^2(\Omega)^N; \quad |d| = 1 \text{ a.e. in } \Omega\}$$

and its characteristic function

$$\chi_C(d) = \begin{cases} 0 & \text{if } d \in C \\ +\infty & \text{if } d \notin C. \end{cases}$$

Then, the problem (31) is equivalent to

$$d \in H_{l^n}^1(\Omega)^N \text{ s.t. } F_2(d) + \chi_C(d) \leq F_2(\bar{d}) + \chi_C(\bar{d}), \quad \forall \bar{d} \in H_{l^n}^1(\Omega)^N. \quad (32)$$

We associate with (32) the Augmented Lagrangian functional $\mathcal{L}_r : H_{l^n}^1(\Omega)^N \times L^2(\Omega)^N \times L^2(\Omega)^N$, defined by

$$\mathcal{L}_r(d, e, \mu) = F_2(d) + \chi_C(e) + \frac{r}{2} |d - e|_{L^2}^2 + (\mu, d - e), \quad (33)$$

with $r > 0$ an ‘‘augmentation’’ parameter. A saddle-point (d, e, μ) of (33), can be approximated by a (standard) Uzawa-type algorithm of the form ($\mu^{(0)}$ given):

1. Given $\mu^{(\ell)}$, find $(d^{(\ell)}, e^{(\ell)})$ such that

$$\mathcal{L}_r(d^{(\ell)}, e^{(\ell)}, \mu^{(\ell)}) \leq \mathcal{L}_r(\bar{d}, \bar{e}, \mu^{(\ell)}), \quad \forall (\bar{d}, \bar{e}) \in (H_{l^n}^1)^N \times L^2(\Omega)^N.$$

2. Update the multiplier

$$\mu^{(\ell+1)} = \mu^{(\ell)} + r(d^{(\ell)} - e^{(\ell)}).$$

For obvious decomposition properties, we prefer the following (Uzawa) block-relaxation type algorithm [4]. Starting from $(e^{(0)}, \mu^{(0)}) \in L^2(\Omega)^N \times L^2(\Omega)^N$ ($e^{(0)} = \mu^{(0)} = d^{n-1}$), we compute successively $d^{(\ell+1)}$, $e^{(\ell+1)}$ and $\mu^{(\ell+1)}$ as follows:

$$d^{(\ell+1)} \in (H_{I_n}^1)^N, \quad \mathcal{L}_r(d^{(\ell+1)}, e^{(\ell)}, \mu^{(\ell)}) \leq \mathcal{L}_r(\bar{d}, e^{(\ell)}, \mu^{(\ell)}), \quad \forall \bar{d} \in (H_{I_n}^1)^N, \quad (34)$$

$$e^{(\ell+1)} \in L^2(\Omega)^N, \quad \mathcal{L}_r(d^{(\ell+1)}, e^{(\ell+1)}, \mu^{(\ell)}) \leq \mathcal{L}_r(d^{(\ell+1)}, \bar{e}, \mu^{(\ell)}), \quad \forall \bar{e} \in L^2(\Omega)^N, \quad (35)$$

$$\mu^{(\ell+1)} = \mu^{(\ell)} + r(d^{(\ell+1)} - e^{(\ell+1)}). \quad (36)$$

Obviously, the discrete approximation space for $e^{(\ell)}$ and $\mu^{(\ell)}$ will be the same as for the director field $d^{(\ell)}$.

Taking into account some simplifications, (34)-(36) lead to Algorithm 3, where implementation of (35) and (36) are made of explicit manner. Moreover, the vectorial problem (37) of Algorithm 3 is decoupled by components.

Algorithm 3 Uzawa block-relaxation algorithm for the subproblem (10)

$\ell = 0$ Given $(e^{(0)}, \mu^{(0)}) \in L^2(\Omega)^N \times L^2(\Omega)^N$ ($e^{(0)} = \mu^{(0)} = d^{n-1}$)

$\ell \geq 0$ Compute successively $d^{(\ell+1)}$, $e^{(\ell+1)}$ and $\mu^{(\ell+1)}$ as follows

1. Compute $d^{(\ell+1)} \in (H_{I_n}^1)^N$ such that

$$\mathbf{a}_2(d^{(\ell+1)}, \bar{d}) + r(d^{(\ell+1)}, \bar{d}) = \boldsymbol{\tau}_2(\bar{d}) + (re^{(\ell)} - \mu^{(\ell)}, \bar{d}), \quad \forall \bar{d} \in (H_0^1)^N. \quad (37)$$

2. Compute the auxiliary unknown

$$e^{(\ell+1)} = \frac{rd^{(\ell+1)} + \mu^{(\ell)}}{|rd^{(\ell+1)} + \mu^{(\ell)}|}.$$

3. Update the multiplier

$$\mu^{(\ell+1)} = \mu^{(\ell)} + r(d^{(\ell+1)} - e^{(\ell+1)}).$$

4. We iterate until

$$\frac{\|d^{(\ell+1)} - d^{(\ell)}\|_{L^2(\Omega)}}{\|d^{(\ell+1)}\|_{L^2(\Omega)}} \leq \varepsilon.$$

6 Numerical experiments

We assume that Ω is a $2D$ polygonal domain which can be entirely triangulated by a family \mathfrak{T}_h furnished by a bigger family \mathfrak{T}_{2h} dividing each triangle of \mathfrak{T}_{2h} into four triangles by

means of the edge-midpoints. Finite element spaces for $(\tilde{u}_h, \tilde{d}_h)$, (u_h, p_h, w_h) and (d_h, e_h, μ_h) are chosen as in Figure 1, where all approximations are over \mathfrak{T}_h except for pressure which is approximated in \mathfrak{T}_{2h} (this approximation for the velocity/pressure is called P_1 -iso- P_2/P_1 which is known to satisfy the Babuska-Brezzi *inf-sup* condition [8]).

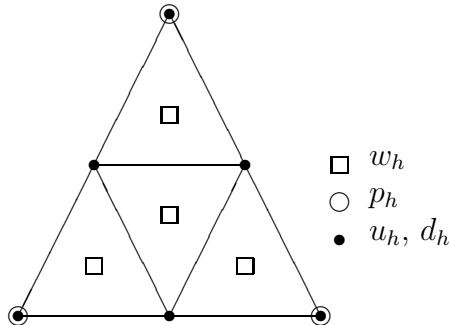


Figure 1: Finite element spaces

All numerical results were carried out using uniform meshes of the square $\Omega = (0, 1)^2$ and vectorized Matlab codes [13]. As linear solver, we use the generalized minimum residual method (**gmres**) for the non symmetric systems of Algorithm 1 and the preconditioned conjugate gradient method (**pcg**) for symmetric (positive definite) systems (i.e. all linear systems in Algorithms 2 and 3). The preconditioner matrices are obtained by incomplete factorizations.

6.1 Test case with exact stationary solution

We present, in this subsection, computations on a test case with known solution, proposed by Prohl [17]. Contrary to [17], now all numerical simulations are carried out without removing the convection term in Equation (1). The constants λ , ν and γ are set to unity, and we adjust right-hand sides in (1)-(3) and (5) such that the following functions be an exact solution:

$$u = \begin{pmatrix} x^2(1-x)^2(2y-6y^2+4y^3) \\ -y^2(1-y)^2(2x-6x^2+4x^3) \end{pmatrix}, \quad (38)$$

$$p = x^2 - \frac{1}{3}, \quad (39)$$

$$d = \begin{pmatrix} \frac{1}{2}x \\ (1 - \frac{1}{4}x^2)^{1/2} \end{pmatrix}. \quad (40)$$

The exact solution (38)-(40) is approximated as a stationary solution of the corresponding evolution problem, using the time-stepping scheme (7)-(8), (9)-(10), with initialization equal to zero. We assume that a stationary solution is reached if the relative $L^2(\Omega)$ -error for

(u_h, p_h, d_h) is less than 10^{-8} . The time step is taken as $k = 0.001$. Tolerance parameter ε for the iterative algorithms (Algorithm 1, Algorithm 2 and Algorithm 3) is set to $\varepsilon = 10^{-6}$.

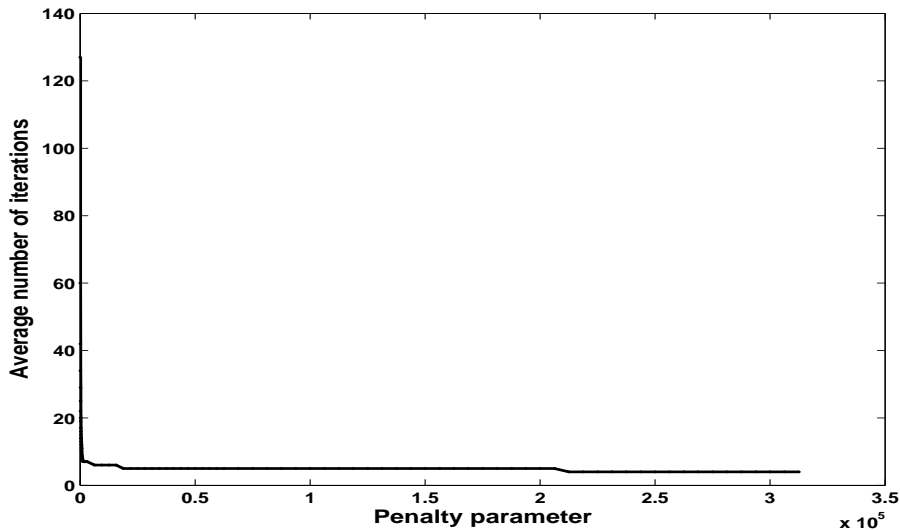


Figure 2: Average number of iterations in Algorithm 3 versus the augmentation parameter r .

The augmented Lagrangian Algorithm 3 is very sensitive to the choice of the augmentation parameter r . In our simulations, we assume that

$$r = \alpha \frac{h}{k}, \quad (41)$$

where h is the mesh size and k the time step. We make this (empirical) choice to make the augmentation parameter independent from the mesh size. Figure 2 shows the average number of iterations in Algorithm 3 (required to reach the stationary state) versus the augmentation parameter r , using the mesh size $h = 1/16$. We can notice that the average number of iterations in Algorithm 3 is large for small values of r ($r < 100$, i.e. $\alpha < 15$). For sufficiently large values of r ($\alpha > 400$), the average number of iterations in Algorithm 3 is between 5 and 6. Note that large values of r can lead to numerical instabilities (zero pivots can appear during complete or incomplete factorizations). Table 1 shows that, with (41) for $\alpha = 500$, the average number of iterations in Algorithm 3 is virtually independent from the mesh size.

Mesh size h	1/8	1/16	1/32	1/64	1/128
Number of iterations	5	5	5	6	6

Table 1: Average number of iterations in Algorithm 3 for various mesh sizes, with $r = 500 h/k$

To reduce the number of unnecessary iterations in Algorithm 3, we have computed the min and max values for $|d_h|$ using several values of the tolerance ε . The results are summarized in

Tolerance ε in Algorithm 3	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
$\min d_h $	0.999987	0.999987	0.999999	0.999999	0.999999
$\max d_h $	1.000402	1.000402	1.000000	1.000000	1.000000

Table 2: Min and max values for $|d_h|$ with respect to ε in Algorithm 3, $h = 1/32$.

Table 2. We can notice that we do not need high accuracy in Algorithm 3. Indeed, according to the mesh size, the norm of the director field can be considered as satisfactory as soon as $\varepsilon = 10^{-2}$.

Table 3 displays the $L^2(\Omega)$ and $H^1(\Omega)$ -errors for (u_h, p_h, d_h) , which confirms numerically the convergence of the proposed scheme with respect to the mesh size h .

h	$\ d_h - d\ _{L^2}$	$\ d_h - d\ _{H^1}$	$\ u_h - u\ _{L^2}$	$\ u_h - u\ _{H^1}$	$\ p_h - p\ _{L^2}$
1/8	6.0709×10^{-4}	7.8064×10^{-3}	9.4204×10^{-4}	1.4222×10^{-2}	7.9048×10^{-3}
1/16	2.3554×10^{-4}	3.8769×10^{-3}	2.4905×10^{-4}	7.0873×10^{-3}	2.4985×10^{-3}
1/32	9.8581×10^{-5}	1.9395×10^{-3}	6.2446×10^{-5}	3.5321×10^{-3}	8.7538×10^{-4}
1/64	4.5747×10^{-5}	9.6495×10^{-4}	1.5677×10^{-5}	1.7533×10^{-3}	3.4502×10^{-4}
1/128	1.6159×10^{-5}	4.7330×10^{-4}	4.0809×10^{-6}	8.8102×10^{-4}	1.1856×10^{-4}

Table 3: $L^2(\Omega)$ and $H^1(\Omega)$ -errors towards the stationary solution (38)-(40)

6.2 Test case with exact time-dependent solution

Again, the constants λ , ν and γ are set to unity and we adjust right-hand sides in (1)-(3) and (5)-(6) such that the exact solution is (cf. Prohl [17]):

$$u = (1 + t^3) \begin{pmatrix} x^2(1-x)^2(2y - 6y^2 + 4y^3) \\ -y^2(1-y)^2(2x - 6x^2 + 4x^3) \end{pmatrix},$$

$$d = \begin{pmatrix} tx \\ (1 - t^2x^2)^{1/2} \end{pmatrix},$$

$$p = t(x^2 - \frac{1}{3}).$$

Tolerance parameter ε for the iterative algorithms (Algorithm 1, Algorithm 2 and Algorithm 3) are set to $\varepsilon = 10^{-6}$. As in the previous Section, we take the penalty parameter in Algorithm 3 as $r = \alpha h/k$. Figure 3 shows the average number of iterations in Algorithm 3 versus the augmentation parameter r for $h = 1/16$ and two time steps: $k = 1/10$ and $k = 1/160$. One can notice that for $k = 1/10$, the number of iterations in Algorithm 3

is lower than 10 if $\alpha > 3 \times 10^4$. In contrast, for $k = 1/160$, the number of iterations in Algorithm 3 is about 5-6 if $\alpha > 50$. We can conclude that if the time step is not small enough, α must be chosen large. This property is illustrated in Table 4. For $k = 1/160$, the number of iterations is virtually independent of the mesh size while it is decreasing with respect to the mesh size for $k = 1/10$.

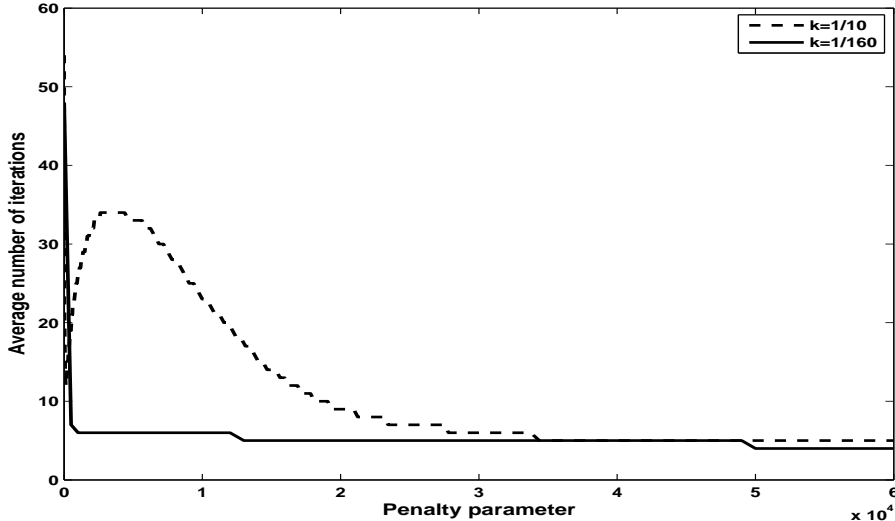


Figure 3: Average number of iterations in Algorithm 3 versus the augmentation parameter r , for $k = 1/10$ (discontinuous line) and $k = 1/160$ (continuous line), and $h = 1/16$.

h	1/8	1/16	1/32	1/64	1/128
Iterations for $k = 1/10$	37	32	18	13	12
Iterations for $k = 1/160$	4	5	5	5	5

Table 4: Average number of iterations in Algorithm 3 for $k = 1/10$ and $k = 1/160$, with $r = 3000 h/k$

Table 5 displays the $L^2(\Omega)$ and $H^1(\Omega)$ -errors at $t = 0.5$ for u_h , p_h and d_h and confirms numerically the convergence of the proposed scheme with respect to the time step k . We notice that the approximation of the pressure is better in the stationary problem than in the evolution case.

6.3 Liquid crystals on a square slab

We consider a test problem, derived from [5], with $\Omega = (0, 1)^2$, $f = 0$, $(\gamma, \nu, \lambda) = (1, 1, 10^{-6})$ and the Dirichlet boundary conditions $u = 0$ and $d = (\cos(p\theta), \sin(p\theta))$, where p is an integer, $\cos \theta = (x - 1/2)/r$, $\sin \theta = (y - 1/2)/r$ and $r = ((x - 1/2)^2 + (y - 1/2)^2)^{1/2}$ for any (x, y) on the boundary. In fact, integer p is the number of cycles given by the director boundary

ℓ	$\ d_h - d\ _{L^2}$	$\ d_h - d\ _{H^1}$	$\ u_h - u\ _{L^2}$	$\ u_h - u\ _{H^1}$	$\ p_h - p\ _{L^2}$
1	2.9303×10^{-4}	2.2657×10^{-3}	1.4089×10^{-5}	5.0818×10^{-4}	8.5832×10^{-2}
2	1.9684×10^{-4}	1.4923×10^{-3}	6.4825×10^{-6}	4.9937×10^{-4}	8.4722×10^{-2}
3	1.2060×10^{-4}	9.2021×10^{-4}	2.5101×10^{-6}	4.9715×10^{-4}	8.4157×10^{-2}
4	6.4724×10^{-5}	5.4118×10^{-4}	6.5534×10^{-7}	4.9670×10^{-4}	8.3880×10^{-2}
5	3.0190×10^{-5}	3.4011×10^{-4}	5.8558×10^{-7}	4.9661×10^{-4}	8.3746×10^{-2}

Table 5: $L^2(\Omega)$ and $H^1(\Omega)$ -errors at $t = 0.5$ and time-steps $k = 0.1 \times 2^{1-\ell}$ ($h = 1/256$)

data. The initial values u_0 and d_0 have the same formulas except that (x, y) is taken in the whole domain Ω .

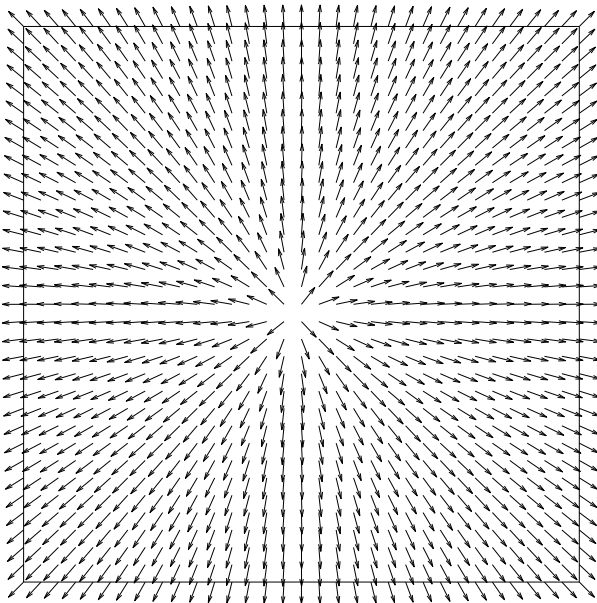


Figure 4: Director field on a square liquid crystal slab for $p = 1$, $t_{stat} = 0.143$.

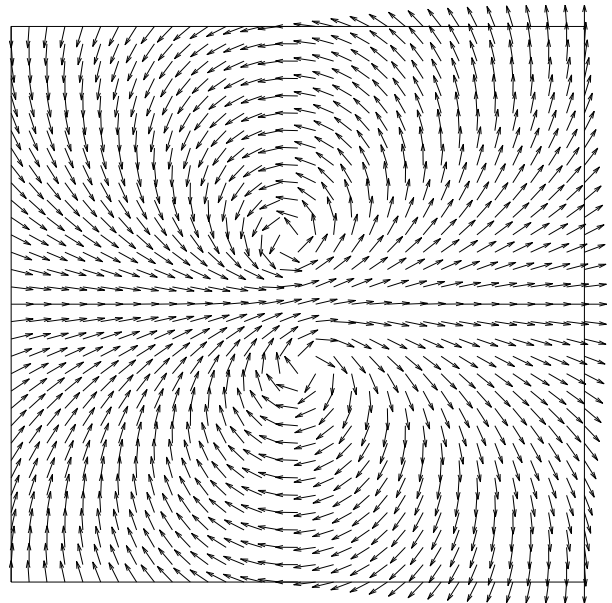


Figure 5: Director field on a square liquid crystal slab for $p = 2$, $t_{stat} = 0.229$.

The time step is $k = 0.001$, the mesh size is $h = 1/32$ and, after some tests, we take the augmentation parameter as $r = 320 h/k$. In order to reduce the number of unnecessary iterations in Algorithm 3, we choose the tolerance in the augmented Lagrangian Algorithm 3 as $\varepsilon = 10^{-4}$. In the other iterative algorithms the tolerance is $\varepsilon = 10^{-6}$. We assume that a stationary solution is reached if the $L^2(\Omega)$ -error for (u_h, d_h, p_h) is less than 10^{-5} and the corresponding time is denoted by t_{stat} .

Figures 4-7 depict the reached stationary director fields. Note that for $p = 1, 2, 3$ and 4 , then one, two, three and four singularities are obtained, respectively. We have also computed the stationary director fields for $p = 5, 6, 7$ and 8 . As was pointed out in [5], it seems hard to predict the number of point singularities for $p \geq 7$. Notice that, our results for $p > 1$

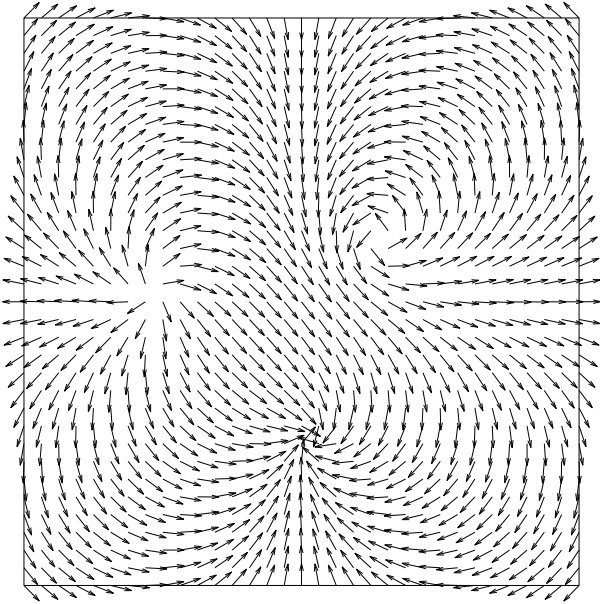


Figure 6: Director field on a square liquid crystal slab for $p = 3$, $t_{stat} = 0.253$.

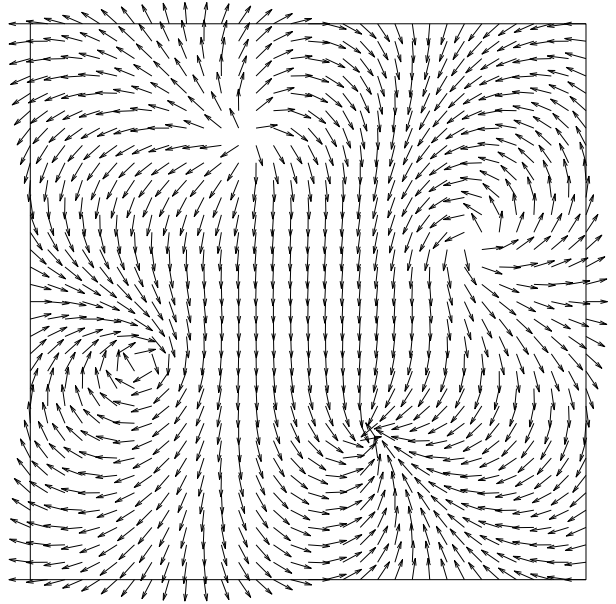


Figure 7: Director field on a square liquid crystal slab for $p = 4$, $t_{stat} = 0.609$.

are rather different from those of [5] from a qualitative point of view, because pictures of director vector given in [5] and in Figures 4-7 are different, and also from a quantitative one, because for instance for $p = 2$ we obtain 2 point-singularities while 3 point-singularities are reached in [5]. Nevertheless, the computations of this paper are not fully comparable to those obtained in [5] where a liquid crystal model involving only the director vector is considered, while our results are obtained for a coupled fluid/director field model.

Finally, Table 6 shows the average number of iterations in Algorithm 3 versus p .

p	1	2	3	4	5	6	7	8
Iterations	9	13	16	20	22	19	17	21

Table 6: Average number of iterations in Algorithm 3 versus p , with $r = 320 h/k$.

Conclusion

We have designed a fully splitting and decoupled in time linear algorithm for nematic liquid crystal flows with explicit treatment of the unitary constraint for the director field by an augmented Lagrangian technique. This algorithm allows us to use only P_1 finite elements in space, choosing the stable pair P_1 -iso P_2/P_1 for the velocity-pressure approximation.

We study the numerical behavior of the scheme, observing that the fixed-point iterative procedure (19)-(20) works well, because the convergence criterion (21) is satisfied after few ($\approx 4 - 5$) iterations. Moreover, since the augmented Lagrangian Algorithm 3 is sensitive

with respect to the augmentation parameter r , we propose an empirical rule for choosing this parameter depending on the quotient of mesh size over time step. Nevertheless, how to make an automatic choice of the optimal value of this augmentation parameter in Algorithm 3 is still an open question.

Acknowledgments

The authors are very grateful to the anonymous referees for their helpful and valuable suggestions and remarks, which greatly improved the earlier version of this paper.

References

- [1] BADIA S., GUILLÉN-GONZÁLEZ F. and GUTIÉRREZ-SANTACREU J.V., *Finite element approximation of nematic liquid crystal flows using a saddle-point structure*. Journal of Computational Physics 230 (2011) 1686-1706.
- [2] BLASCO J. and CODINA R., *Error estimates for an operator-splitting method for incompressible flows*. Appl. Num. Math., **51** (2004), 1-17.
- [3] BECKER R., FENG X. and PROHL A., *Finite element approximations of the Ericksen-Leslie model for nematic liquid crystal flow*. SIAM J. Numer. Anal. 46 (2008), no. 4, 1704-1731.
- [4] GLOWINSKI R. and LE TALLEC P., *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics*. Studies in Applied Mathematics. SIAM, Philadelphia, 1989.
- [5] GLOWINSKI R., LIN P. and PAN X.-B., *An operator-splitting method for a liquid crystal model*. Comput. Physics Comm. 152 (2003), 242-252.
- [6] GLOWINSKI R., PAN T.W. and PERIAUX J., *Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies*. *Comput. Meth. Appl. Mech. Eng.*, 151 (1998), 181-184.
- [7] GLOWINSKI R., PAN T.W., HESLA T.I. and JOSEPH D., *A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow*. *Comput. Meth. Appl. Mech. Eng.*, 184 (2000), 241-267.
- [8] GIRAULT V. and RAVIART P.A., *Finite element methods for Navier-Stokes equations: theory and algorithms*. Berlin, Springer-Verlag, 1986.
- [9] GIRAULT V. and GUILLÉN-GONZÁLEZ F., *Mixed formulation, approximation and decoupling algorithm for a nematic liquid crystals model*. *Math. Comput.* 80 (2011), no. 274, 781-819.

- [10] GUILLÉN-GONZÁLEZ F. and GUTIÉRREZ-SANTACREU J.V., A linear mixed finite element scheme for a nematic Ericksen-Leslie liquid crystal model. *ESAIM: M2AN*. 47 (2013), no. 5, 1433-1464.
- [11] GUILLÉN-GONZÁLEZ F. and REDONDO-NEBLE M.V., New error estimates for a viscosity-splitting scheme in time for the three-dimensional Navier-Stokes equations. *IMA J. Num. Anal.* 31 (2011), no. 2, 556-579.
- [12] GUILLÉN-GONZÁLEZ F. and REDONDO-NEBLE M.V., Spatial error estimates for a finite element viscosity-splitting scheme for the Navier-Stokes equations *Int. J. Num. Anal. Methods*, 10, no. 4 (2013), 826-844.
- [13] KOKO J., *Calcul Scientifique avec Matlab*, Ellipses, Paris, 2009.
- [14] LIN P. and LIU C., Simulations of singularity dynamics in liquid crystal flows: A C^0 finite element approach. *Journal of Computational Physics* 215 (2006) 348-362.
- [15] LIU C. and WALKINGTON N., Approximation of liquid crystal flows. *SIAM J. Numer. Anal.*, 37 (2000), no. 3, 726-741.
- [16] LIU C. and WALKINGTON N., Mixed methods for the approximation of liquid crystal flows. *M2AN*, 36 (2002), no. 2, 205-222.
- [17] PROHL A., *Computational Micro-magnetism*. Advances in Numerical Mathematics. Teubner, 2001.