

# Helmholtz Decomposition of Vector Fields Using an Optimal Preconditioned Conjugate Gradient Algorithm

Jorge Lopez

UJAT, Tabasco, México

Email: jorge.lopez@ujat.mx

**How to cite this paper:** Lopez, J. (2023) Helmholtz Decomposition of Vector Fields Using an Optimal Preconditioned Conjugate Gradient Algorithm. *Journal of Applied Mathematics and Physics*, 11, 1337-1348. <https://doi.org/10.4236/jamp.2023.115086>

**Received:** April 14, 2023

**Accepted:** May 22, 2023

**Published:** May 25, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

In this article, we study numerically a Helmholtz decomposition methodology, based on a formulation of the mathematical model as a saddle-point problem. We use a preconditioned conjugate gradient algorithm, applied to an associated operator equation of elliptic type, to solve the problem. To solve the elliptic partial differential equations, we use a second order mixed finite element approximation for discretization. We show, using 2-D synthetic vector fields, that this approach, yields very accurate solutions at a low computational cost compared to traditional methods with the same order of approximation.

## Keywords

Helmholtz Decomposition, Self-Adjoint Operator, Optimal Preconditioning, Finite Element

## 1. Introduction

Several problems and applications require a deep knowledge of a vector field over a region. An important tool in this context is the Helmholtz decomposition theorem (HDT) that guarantees, under certain hypothesis [1] [2], a unique decomposition of a vector field  $\mathbf{u}$  defined in a domain  $\Omega$  into its solenoidal  $\mathbf{u}_s$  and irrotational  $\mathbf{u}_p$  components. This is,

$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_p, \text{ in } \Omega. \quad (1)$$

This decomposition is important, for example, because the properties like incompressibility and vorticity can be studied directly by studying these components.

The application areas of the HD include (but are not restricted to) electromagnetism, linear elasticity, fluid mechanic [3], astrophysics [4], geophysics [5], computer vision and robotic [6] [7] [8], image processing [9]. The HDT also establishes that the Helmholtz decomposition components can be obtained in terms of a vector potential (the solenoidal) and a scalar potential (the irrotational), solution of certain Poisson equations with different kinds of boundary conditions, one of them being  $\mathbf{u}_s \cdot \mathbf{n} = 0$ , on  $\Gamma = \partial\Omega$ . These elliptic equations cannot be solved explicitly, so it is necessary to apply numerical methods to solve it. The most common mesh dependant methods are the finite element methods (FEM) [6] [10] [11], Finite difference methods (FDM) [9] [12], Galerkin Formulation Using Finite Differences [13], Fourier and Wavelets Domains [14] [15] [16]. Among the meshless methods, we can mention the Smoothed Particle Hydrodynamics (SPH) [17] [18] [19] and the Statistical Learning Using Matrix-Valued Kernels [20] [21] [22].

The problem we are considering in this work is the numerical decomposition, in a given domain  $\Omega$ , of a vector field  $\mathbf{u}$  into its solenoidal and irrotational components with the more general boundary condition:  $\mathbf{u}_s \cdot \mathbf{n} = 0$  is satisfied only in  $\Gamma_N$ , a section of  $\Gamma$  where the solenoidal vector field is tangential to the boundary. This kind of boundary conditions appears, for example when we want to adjust vector fields from experimental or incomplete data. Let  $u$  be given and defined in the Lipschitz bounded domain  $\Omega$ , with boundary  $\Gamma = \Gamma_N \cup \Gamma_D$ . We want to solve numerically the following model:

$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_p, \text{ in } \Omega, \quad (2)$$

$$\nabla \cdot \mathbf{u}_s = 0, \text{ in } \Omega, \quad (3)$$

$$\mathbf{u}_p = \nabla p \text{ in } \Omega, \quad (4)$$

$$\mathbf{u}_s \cdot \mathbf{n} = 0, \text{ on } \Gamma_N, \quad (5)$$

where  $p$  is a scalar potential,  $\mathbf{n}$  is the unitary outer vector, normal to the boundary  $\Gamma$  and  $\Gamma_D = \Gamma - \Gamma_N$ .

We apply a methodology based on the solution of saddle point problems to find simultaneously the pair  $(\mathbf{u}_s, p)$  (and consequently  $\mathbf{u}_p = \mathbf{u} - \mathbf{u}_s$ ) by using an optimal preconditioned conjugate gradient algorithm combined with a mixed finite element method to solve elliptic problems. The associated stiffness matrix components are approximated by a trapezoidal rule that produce a diagonal matrix so that the solution of the stiffness system is easy. This property together with the use of an optimal preconditioned resulted in a very fast algorithm.

The structure of this article is as follows: In Section 2, we consider the mathematical formulation of the problem and we describe the classical elliptic problems for scalar potentials allowing the decomposition. In Section 3, we describe the variational version of the problem. After reformulating the problem, we describe a preconditioned conjugate gradient (*PCG-algorithm*), where a mixed finite element method is used to solve the elliptic subproblems at each iteration. In Section 4, we present and discuss the numerical results, and finally, in Section

5, we give some concluding remarks.

## 2. Mathematical Formulation of the Helmholtz Decomposition

The Helmholtz decomposition theorem [2], establishes that if  $\Omega \subset \mathbb{R}^d$  ( $d \geq 2$ ) is a bounded Lipschitz domain with boundary  $\Gamma$ , then, each  $\mathbf{u} \in \mathbf{L}_2(\Omega)$  have a unique Helmholtz decomposition

$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_p, \tag{6}$$

if we know that the solenoidal component satisfies  $\mathbf{u}_s \cdot \mathbf{n} = 0$  in  $\Gamma$ , i.e.

$\mathbf{u}_s \in H_{div}(\Omega)$  and  $\mathbf{u}_p \in G(\Omega)$ , where

$$H_{div}(\Omega) = \{w \in \mathbf{H}(div; \Omega) : \nabla \cdot \mathbf{w} = 0 \text{ \& } \mathbf{w} \cdot \mathbf{n} = 0 \text{ in } \Gamma\}, \tag{7}$$

$$\mathbf{H}(div; \Omega) = \{\mathbf{v} \in \mathbf{L}_2(\Omega) : \nabla \cdot \mathbf{v} \in L_2(\Omega)\} \tag{8}$$

and

$$G(\Omega) = \{\mathbf{v} \in \mathbf{L}_2(\Omega) : \exists p \in L_2(\Omega) \text{ such that } \mathbf{v} = \nabla p\}. \tag{9}$$

Spaces  $H_{div}(\Omega)$  and  $G(\Omega)$  are orthogonal in  $\mathbf{L}_2(\Omega)$ .

If  $\mathbf{u} \in \mathbf{H}(div; \Omega)$  and the boundary condition  $\mathbf{u}_s \cdot \mathbf{n} = 0$  in  $\Gamma$  is given, the irrotational vector field  $\mathbf{u}_p = \nabla p$  can be calculated by obtaining the scalar field  $p(\mathbf{x}) \in \mathbf{H}^1(\Omega) / \mathbb{R}$ , as solution of the Poisson problem with Neumann boundary condition

$$\begin{aligned} -\Delta p &= -\nabla \cdot \mathbf{u}, \text{ in } \Omega, \\ \nabla p \cdot \mathbf{n} &= \mathbf{u} \cdot \mathbf{n}, \text{ on } \Gamma, \end{aligned} \tag{10}$$

where  $\mathbf{H}^1(\Omega) / \mathbb{R} = \{v \in \mathbf{H}^1(\Omega) : \int_{\Omega} v(\mathbf{x}) \, d\mathbf{x} = 0\}$ .

If we first solve the elliptic problem (10) to find  $p(\mathbf{x})$  then we calculate  $\mathbf{u}_p = \nabla p$  and finally we calculate  $\mathbf{u}_s = \mathbf{u} - \mathbf{u}_p$ .

In order to describe our methodology for finding the Helmholtz decomposition, we consider a given vector field  $\mathbf{u} \in \mathbf{H}(div; \Omega)$  satisfying (2)-(5). From (2), we have

$$\mathbf{u} - \mathbf{u}_s = \mathbf{u}_p,$$

or

$$\mathbf{u} - \mathbf{u}_s = \nabla p, \text{ for some } p \in L_2(\Omega),$$

From this,

$$\int_{\Omega} (\mathbf{u} - \mathbf{u}_s) \cdot \mathbf{v} \, d\mathbf{x} = \int_{\Omega} \nabla p \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{H}(div; \Omega).$$

Since

$$\int_{\Omega} \nabla \lambda \cdot \mathbf{v} \, d\mathbf{x} = \int_{\Gamma} \lambda \mathbf{v} \cdot \mathbf{n} \, d\gamma - \int_{\Omega} \lambda \nabla \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in H(div; \Omega), \forall \lambda \in L_2(\Omega),$$

we can write (if  $p = 0$  in  $\Gamma_D$  and  $\mathbf{v} \cdot \mathbf{n} = 0$  on  $\Gamma_N$ )

$$\int_{\Omega} (\mathbf{u} - \mathbf{u}_s) \cdot \mathbf{v} \, d\mathbf{x} = - \int_{\Omega} p \nabla \cdot \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N,$$

where  $\mathbf{V}_N$  is the space of vector fields defined as

$$\mathbf{V}_N = \{ \mathbf{v} \in H(\text{div}; \Omega) : \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \Gamma_N \}. \tag{11}$$

Considering condition (3) we conclude that the pair  $(\mathbf{u}_s, p) \in \mathbf{V}_N \times L_2(\Omega)$ , satisfies

$$\begin{cases} \int_{\Omega} (\mathbf{u} - \mathbf{u}_s) \cdot \mathbf{v} \, d\mathbf{x} = - \int_{\Omega} p \nabla \cdot \mathbf{v} \, d\mathbf{x}, & \forall \mathbf{v} \in \mathbf{V}_N, \\ \int_{\Omega} q \nabla \cdot \mathbf{u}_s \, d\mathbf{x} = 0, & \forall q \in L_2(\Omega). \end{cases} \tag{12}$$

Next section is devoted to find the pair  $(\mathbf{u}_s, p) \in \mathbf{V}_N \times L_2(\Omega)$ , satisfying (12).

### 3. A Preconditioned Conjugate Gradient Algorithm

Following [23], assume that  $(\mathbf{u}_s, p)$  is solution of problem (12) with

$$\mathbf{u} - \mathbf{u}_s = \mathbf{u}_p. \tag{13}$$

Considering that  $\nabla \cdot \mathbf{u}_s = 0$ , it follows that  $\nabla \cdot \mathbf{u}_p = \nabla \cdot \mathbf{u}$ , so we have that  $p$  satisfies the following operational equation

$$Ap = \nabla \cdot \mathbf{u}, \tag{14}$$

where  $A : L_2(\Omega) \rightarrow L_2(\Omega)$  is the operator defined by

$$A\mu = \nabla \cdot \mathbf{u}_\mu, \tag{15}$$

with  $\mathbf{u}_\mu$  the solution of

$$\begin{cases} \mathbf{u}_\mu \in \mathbf{H}(\text{div}; \Omega), \\ \int_{\Omega} \mathbf{u}_\mu \cdot \mathbf{v} \, d\mathbf{x} = - \int_{\Omega} \mu \nabla \cdot \mathbf{v} \, d\mathbf{x}, & \forall \mathbf{v} \in \mathbf{V}_N. \end{cases} \tag{16}$$

Note that subscript  $s$  in (13) have not the same meaning as  $p$  and  $\mu$  in (16). The properties of operator  $A$  (linear, self-adjoint and strongly elliptic, [23]), allow us to solve equation (14) by using a conjugate gradient algorithm operating in  $L_2(\Omega)$ . Also, we can use the optimal preconditioner  $B^{-1} : L_2(\Omega) \rightarrow L_2(\Omega)$  defined by

$$Bq = \phi_q, \tag{17}$$

where  $\phi_q$  solves the problem

$$\int_{\Omega} \nabla \phi_q \cdot \nabla \psi \, d\mathbf{x} = \int_{\Omega} q \psi \, d\mathbf{x}, \quad \forall \psi \in H_D^1(\Omega), \tag{18}$$

$$\phi_q = 0 \text{ on } \Gamma_D, \tag{19}$$

$$\nabla \phi_q \cdot \mathbf{n} = 0 \text{ on } \Gamma_N, \tag{20}$$

with

$$H_D^1(\Omega) = \{ \psi \in H^1(\Omega) : \psi = 0 \text{ on } \Gamma_D \}. \tag{21}$$

Next we enumerate the three stages of the preconditioned conjugate gradient algorithm to solve the equation  $Ap = b$  :

- 1) **Initialization:**  $p^0$  given,  $g^0 = Ap^0 - b$ ,  $\hat{g}^0 = Bg^0$ ,  $d^0 = -\hat{g}^0$ .
- 2) **Descent:** For  $m \geq 0$ , assuming we know  $p^m, g^m, \hat{g}^m, d^m$ , find  $p^{m+1}, g^{m+1}, \hat{g}^{m+1}, d^{m+1}$  by

$$p^{m+1} = p^m + \alpha_m d^m \text{ where } \alpha_m = \frac{\langle g^m, \hat{g}^m \rangle}{\langle d^m, Ad^m \rangle}.$$

$$g^{m+1} = g^m + \alpha_m Ad^m,$$

$$\hat{g}^{m+1} = \hat{g}^m + \alpha_m B(Ad^m).$$

**3) Test of convergence and new conjugate direction:**

If  $\langle g^{m+1}, \hat{g}^{m+1} \rangle \leq \varepsilon \langle g^0, \hat{g}^0 \rangle$ , take  $p = p^{m+1}$  and stop.

Else  $d^{m+1} = -\hat{g}^{m+1} + \beta_m d^m$  with  $\beta_m = \frac{\langle g^{m+1}, \hat{g}^{m+1} \rangle}{\langle g^m, \hat{g}^m \rangle}$

Do  $m = m + 1$  and return to 2.

Using Equations (15)-(16), which define operator  $A$ , and Equations (17)-(18) which define operator  $B$ , the detailed preconditioned conjugate gradient algorithm (PCG) to obtain  $p$  and  $\mathbf{u}_s$  (and  $\mathbf{u}_p$ ) is as follows:

**Initialization**

1) Given  $p^0 \in L_2(\Omega)$ , solve

$$\begin{cases} \mathbf{u}_p^0 \in \mathbf{H}(\text{div}; \Omega), \\ \int_{\Omega} (\mathbf{u}_p^0) \cdot \mathbf{v} \, dx = -\int_{\Omega} p^0 \nabla \cdot \mathbf{v} \, dx, \forall \mathbf{v} \in \mathbf{V}_N. \end{cases}$$

2) Let  $g^0 = -\nabla \cdot \mathbf{u}^0$ , where  $\mathbf{u}^0 = -(\mathbf{u}_p^0 - \mathbf{u})$ .

3) Solve

$$\begin{cases} \phi^0 \in H_D^1(\Omega), \\ \int_{\Omega} (\nabla \phi^0) \cdot \nabla \psi \, dx = \int_{\Omega} g^0 \psi \, dx, \forall \psi \in H_D^1(\Omega). \end{cases}$$

4) Let  $\hat{g}^0 = \phi^0$ ,  $d^0 = -\hat{g}^0$ .

**Descent**

For  $m \geq 0$ , assuming  $p^m$ ,  $g^m$ ,  $\hat{g}^m$ ,  $d^m$ ,  $\mathbf{u}^m$  are known, compute  $p^{m+1}$ ,  $g^{m+1}$ ,  $\hat{g}^{m+1}$ ,  $d^{m+1}$  and  $\mathbf{u}^{m+1}$ , using the following steps:

5) Solve

$$\begin{cases} \mathbf{u}^m \in \mathbf{H}(\text{div}; \Omega), \\ \int_{\Omega} (\mathbf{u}^m) \cdot \mathbf{v} \, dx = -\int_{\Omega} d^m \nabla \cdot \mathbf{v} \, dx, \forall \mathbf{v} \in \mathbf{V}_N. \end{cases}$$

6) Let  $\bar{g}^m = \nabla \cdot \mathbf{u}^m$ .

7) Let  $\alpha_m = \int_{\Omega} g^m \hat{g}^m \, dx / \int_{\Omega} \bar{g}^m d^m \, dx$ .

8) Solve

$$\begin{cases} \bar{\phi}^m \in H_D^1(\Omega), \\ \int_{\Omega} (\nabla \bar{\phi}^m) \cdot \nabla \psi \, dx = \int_{\Omega} \bar{g}^m \psi \, dx, \forall \psi \in H_D^1(\Omega). \end{cases}$$

9) Set

$$p^{m+1} = p^m + \alpha_m d^m,$$

$$\mathbf{u}^{m+1} = \mathbf{u}^m - \alpha_m \mathbf{u}^m,$$

$$g^{m+1} = g^m + \alpha_m \bar{g}^m,$$

$$\hat{g}^{m+1} = \hat{g}^m + \alpha_m \bar{\phi}^m.$$

**Test of convergence and new descent direction**

If  $\int_{\Omega} g^{m+1} \hat{g}^{m+1} d\mathbf{x} / \int_{\Omega} g^0 \hat{g}^0 d\mathbf{x} < \varepsilon$ , then do  $p = p^{m+1}$ ,  $\mathbf{u}_s = \mathbf{u}^{m+1}$  and stop. Otherwise, do the following:

- 10) Compute  $\beta_m = \int_{\Omega} g^{m+1} \hat{g}^{m+1} d\mathbf{x} / \int_{\Omega} g^m \hat{g}^m d\mathbf{x}$
- 11) Set  $d^{m+1} = -\hat{g}^{m+1} + \beta_m d^m$ .
- 12) Do  $m = m + 1$  and return to 5.

Note that in this algorithm,  $\mathbf{u}_s$  and  $p$  are computed simultaneously.

To approximate the functions belonging to the spaces  $\mathbf{V}_N$  and  $L_2(\Omega)$ , we make use of the *Bercovier-Pironneau* finite element approximation using two triangulation of  $\Omega$ , the coarse mesh  $\mathcal{T}_{2h}$  and the fine mesh  $\mathcal{T}_h$  obtained from the coarse triangulation through a regular subdivision of each triangle  $T \in \mathcal{T}_{2h}$ , as shown in **Figure 1**.

The vector-valued functions such as  $\mathbf{u}_s$  are approximated in the fine mesh and the scalar functions, such as  $p$  are approximated in the coarse mesh. Since  $\mathbf{u}_s$  is obtained on the fine mesh, its resolution is the same as that obtained when solving the elliptics problems (10) in  $\mathcal{T}_h$ .

To measure the global difference between the exact solenoidal field  $\mathbf{u}_s$  and the computed solenoidal field  $\mathbf{u}_{sh}$  we take the relative error

$$e_{sr} = \frac{\|\mathbf{u}_s - \mathbf{u}_{sh}\|_2}{\|\mathbf{u}_s\|_2}. \tag{22}$$

Also, we computed the  $L^2$ -norm of the divergence of  $\mathbf{u}_{sh}$ ,  $\|\nabla \cdot \mathbf{u}_{sh}\|_{L_2(\Omega)}$ , which we denote by *ndiv* in the next section.

### 4. Numerical Results

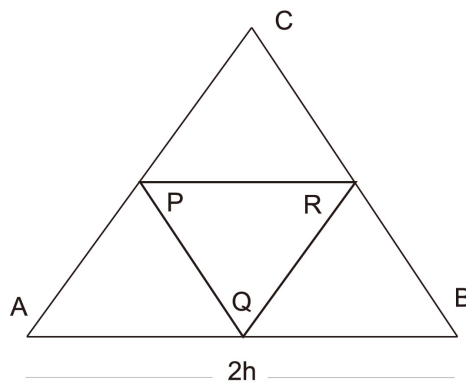
To show the performance of the *PCG-algorithm* we chose two synthetic vector fields. In Example 1 we consider a vector field satisfying the boundary condition

$$\mathbf{u}_s \cdot \mathbf{n} = 0, \text{ on } \Gamma. \tag{23}$$

In Example 2 we consider a vector field satisfying the boundary condition

$$\mathbf{u}_s \cdot \mathbf{n} = 0, \tag{24}$$

but only in  $\Gamma_N$ , a section of  $\Gamma$ .



**Figure 1.** Element in  $\mathcal{T}_{2h}$ : triangle ABC. Elements in  $\mathcal{T}_h$ : triangles AQP, PRC, PQR and QBR.

All numerical calculations were done in a Toshiba PC: Portege R705, Windows 7, 64 Bits, Intel Processor CORE i3, 2.27 GHz, 3 GB Ram.

**Example 1.** We consider the 2D vector field

$$\mathbf{u}(x, y) = \mathbf{u}_s(x, y) + \mathbf{u}_c(x, y), \Omega = (-1, 1) \times (-1, 1), \tag{25}$$

for which the solenoidal and irrotational components are

$$\mathbf{u}_s(x, y) = (-\sin(\pi x)\cos(\pi y), \cos(\pi x)\sin(\pi y)), \tag{26}$$

$$\mathbf{u}_c(x, y) = (\pi \cos(\pi(x+y)), \pi \cos(\pi(x+y))) = \nabla(\sin(\pi(x+y))). \tag{27}$$

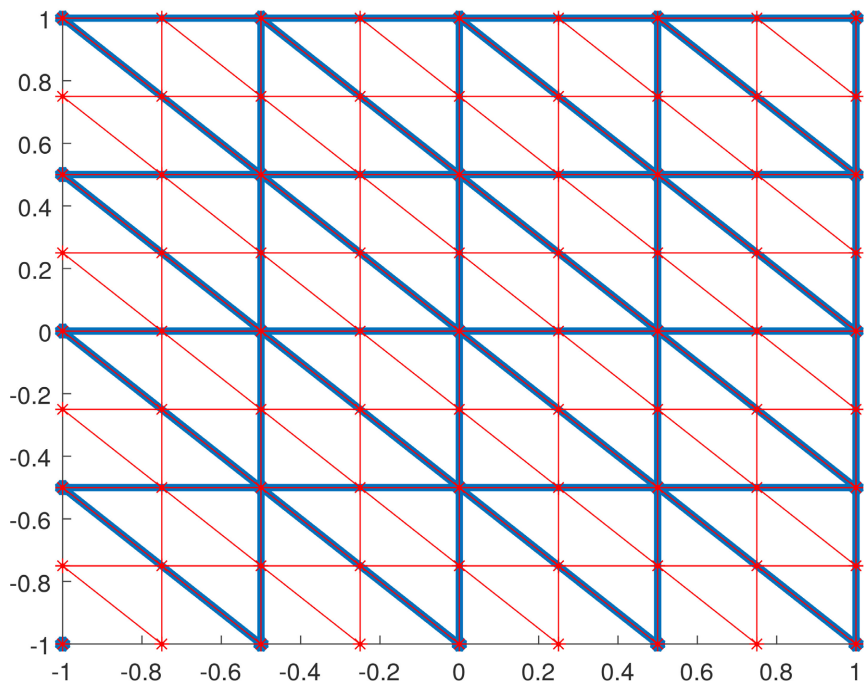
In this case,

$$\mathbf{u}_s \cdot \mathbf{n} = 0 \text{ in } \Gamma_N = \Gamma. \tag{28}$$

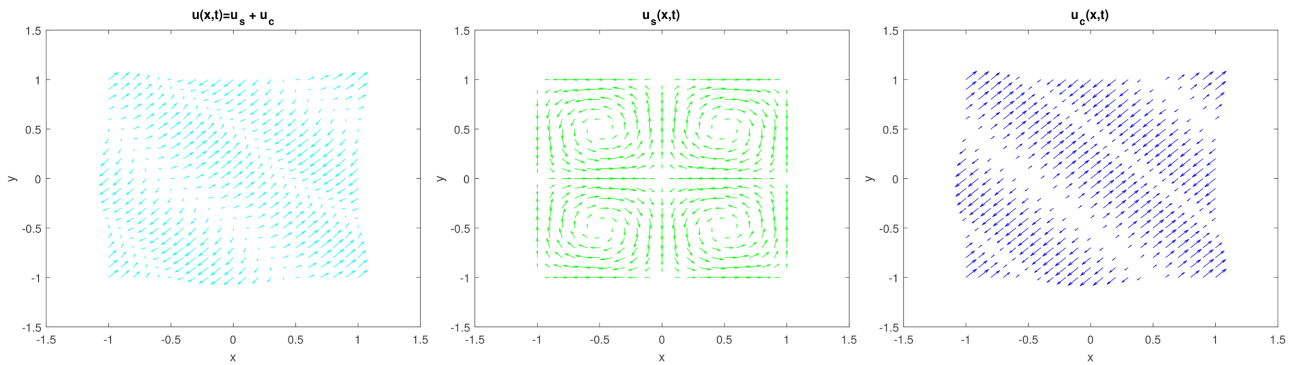
Consequently,  $\Gamma_D = \emptyset$ . To solve the elliptic problems in the PCG algorithm we use simultaneously a coarse triangular mesh  $\mathcal{T}_{2h}$  (blue in **Figure 2**) to approximate the scalar potential  $p$  and a fine triangular mesh  $\mathcal{T}_h$  (red in **Figure 2**) to approximate the solenoidal component.

In **Figure 3**, we show the exact vector fields  $\mathbf{u}(x, y)$ ,  $\mathbf{u}_s(x, y)$  and  $\mathbf{u}_c(x, y)$ .

We want to see how well we can approximate the solenoidal component and the scalar potential  $p$  applying the PCG algorithm. A summary of the numerical results is shown in **Table 1**, where we show the relative error  $e_{sr}$  of the solenoidal component, the  $L^2$ -norm of the divergence  $ndiv$  of the solenoidal component, as well as the number of iterations to get convergence, up to the given tolerance ( $\varepsilon = 10^{-4}$ ), for several mesh sizes.



**Figure 2.** An example of a  $5 \times 5$  coarse mesh and a  $9 \times 9$  refined mesh. The scalar potential  $p$  is approximated in the coarse mesh  $\mathcal{T}_{2h}$  (blue). The solenoidal component  $\mathbf{u}_s$  is approximated in the fine mesh  $\mathcal{T}_h$  (red). On each blue triangle there are four red triangles.



**Figure 3.** Exact vector fields for Example 1. Left:  $\mathbf{u}$ . Middle: solenoidal component  $\mathbf{u}_s$ . Right: irrotational component  $\mathbf{u}_c$ .

**Table 1.** Numerical results for Example 1.  $\varepsilon = 10^{-4}$ .

$\mathcal{T}_h$	<i>CG-algorithm</i>			<i>PCG-algorithm</i>		
	$e_r$	<i>ndiv</i>	iters.	$e_r$	<i>ndiv</i>	iters.
$33 \times 33$	3.84E-2	1.65E-3	9	6.87E-2	4.55E-3	4
$65 \times 65$	1.65E-2	3.25E-4	18	8.50E-2	3.03E-3	3
$129 \times 129$	1.18E-2	1.22E-4	34	5.99E-2	1.08E-3	3
$257 \times 257$	7.57E-3	3.38E-5	67	8.90E-2	6.89E-4	2

In **Figure 4**, we show the stream lines for the exact (left) and approximated (right) solenoidal vector fields. In **Figure 5**, we show the exact (left) and approximated (right)  $p$ .

It is important to mention that the error bounds in **Table 1** were obtained using in the stop criterion of the conjugate gradient algorithm the relaxed value  $\varepsilon = 10^{-4}$  and the results can be improved if we use a more restrictive  $\varepsilon$ , as is shown in the next example.

**Example 2.** Here we consider the 2D vector field  $\mathbf{u}(x, y) = (x, 0)$

$$\mathbf{u}(x, y) = \mathbf{u}_s(x, y) + \mathbf{u}_c(x, y) = (x, 0), \Omega = (1, 2) \times (0, 1), \tag{29}$$

for which the solenoidal and irrotational components are

$$\mathbf{u}_s(x, y) = (x, -y), \tag{30}$$

$$\mathbf{u}_c(x, y) = (0, y) = \nabla p(x, y) = \nabla(y^2/2 + c), c \text{ any constant.} \tag{31}$$

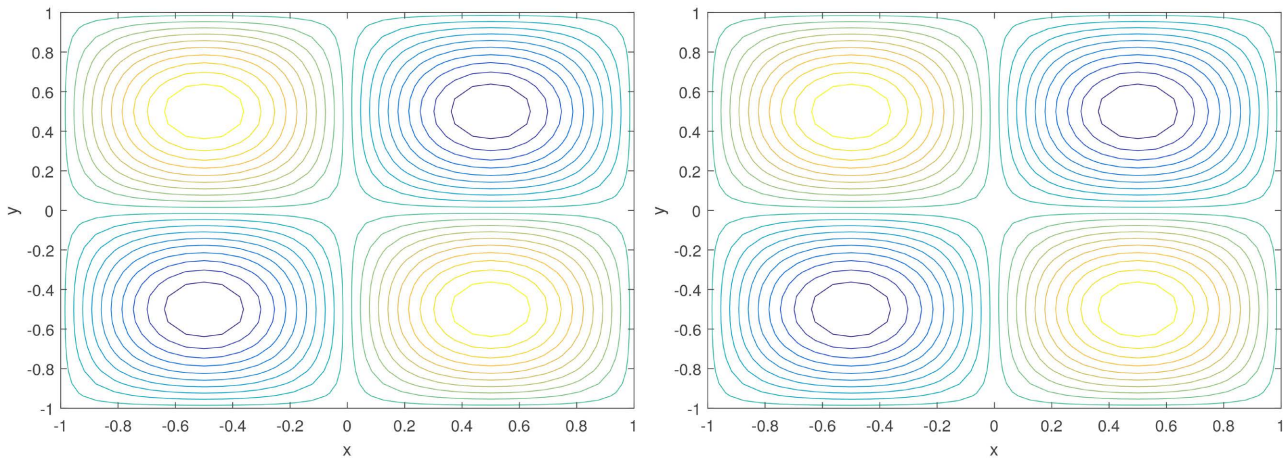
In this example,

$$\mathbf{u}_s \cdot \mathbf{n} = 0 \text{ on } \Gamma_N = \{(x, y) \in \partial\Omega : y = 0\}, \tag{32}$$

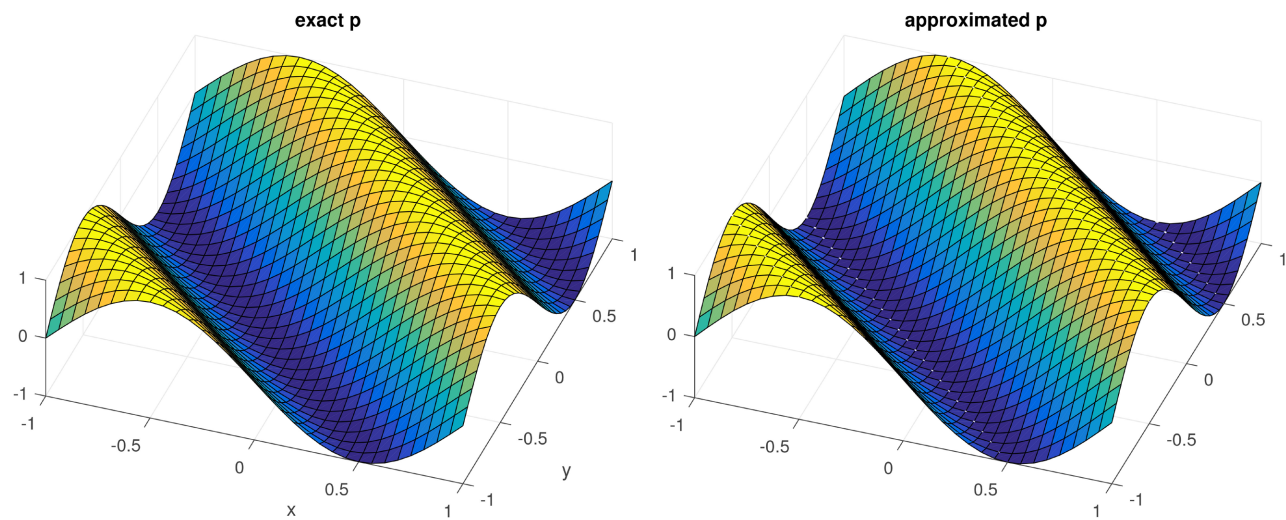
so that,  $\Gamma_D$  is the top and vertical boundary of  $\Omega$ . Our formulation implies that  $p = 0$  in  $\Gamma_D$ , but this is only possible in the top part of  $\Gamma_D$ , and we do not want (and we are not able) to impose additional boundary conditions in  $p$ , so we impose exact boundary conditions in  $\mathbf{u}_s$  in the vertical boundary. **Table 2** shows the numerical results obtained for this test problem.

For comparison reasons we have repeated the calculations in example 2 but using a more restrictive stop tolerance of  $\varepsilon = 10^{-12}$  for the conjugate gradient





**Figure 4.** Stream lines for the exact (left) and approximated (right) solenoidal vector fields for Example 1, in a velocity mesh of  $65 \times 65$  (a coarse mesh of  $33 \times 33$ ).



**Figure 5.** Exact (left) and approximated (right)  $p$  for Example 1, in a coarse mesh of  $33 \times 33$  (a velocity mesh of  $65 \times 65$ ).

**Table 2.** Numerical results for Example 2.  $\varepsilon = 10^{-4}$ .

$T_h$	CG-algorithm			PCG-algorithm		
	$e_r$	$ndiv$	iters.	$e_r$	$ndiv$	iters.
$33 \times 33$	1.16E-3	1.24E-5	84	1.82E-3	6.19E-5	2
$65 \times 65$	5.04E-4	3.29E-6	170	6.40E-4	1.09E-5	2
$129 \times 129$	1.88E-4	9.04E-7	346	2.25E-4	1.93E-6	2
$257 \times 257$	9.25E-5	2.23E-7	695	9.93E-5	3.40E-7	2

method with and without preconditioning. We also solved a problem equivalent to (10) but with exact (not diagonal) stiffness matrix. A summary of the numerical results with the three algorithms is shown in **Table 3**, where we show the relative solenoidal error  $e_r$ , average of divergence  $mdiv$ , as well as the number of iterations to get convergence, up to the given tolerance ( $\varepsilon = 10^{-12}$ , in the CG

**Table 3.** Numerical results for Example 3.  $\varepsilon = 10^{-12}$ .

$T_h$	<i>CG-algorithm</i>				<i>PCG-algorithm</i>				<i>E-alg</i>
	$e_r$	<i>mdiv</i>	iters.	CPUt	$e_r$	<i>mdiv</i>	iters.	CPUt	CPUt
$33 \times 33$	3.1E-4	-7.6E-09	260	0.25	1.9E-3	-3.4E-6	6	0.03	0.01
$65 \times 65$	1.1E-4	-5.0E-10	498	1.18	6.9E-4	-4.5E-7	7	0.085	0.06
$129 \times 129$	4.0E-5	-2.7E-11	854	7.24	2.4E-4	-5.9E-8	7	0.26	0.73
$257 \times 257$	1.4E-5	-1.6E-12	1688	60.69	8.6E-5	-7.5E-9	7	1.96	15.75

and PCG algorithms) for different mesh sizes. This table also contains the CPU time (sec) spent in each numerical experiment, and, in particular, in the last column the CPU time spent by solving the elliptic problem equivalent to (10). For this elliptic problem the results for  $e_r$  and *mdiv* were not as good as those of the CG algorithms.

It is clear that the *PCG-algorithm* performs much better than the other algorithms. Another nice feature of the performance of the *PCG-algorithm* in this example is that the number of iterations is independent of the mesh size (7 iterations in each case). Also, the relative error between the computed and the exact solenoidal vector field is of the same order for the CG and PCG algorithms, with the largest difference occurring on the top boundary. For this example, we observe a loss of accuracy on the mean divergence when the *PCG-algorithm* is employed. Anyway, the mean divergence obtained with the *PCG-algorithm* is still very accurate, from a practical point of view, since we are using polynomial of degree 1 to approximate the unknowns.

## 5. Conclusions

We have applied an optimal preconditioner for the conjugate gradient algorithm, to solve the operator equation associated with the saddle-point formulation of the Helmholtz decomposition problem. Numerical results show that the new preconditioned conjugate gradient algorithm, and its stable approximation by a mixed finite element discretization, preserve the good properties of the non-preconditioned conjugate gradient algorithm, but it is much faster, and produces much better results than the methods based on the elliptic problem (10). Some additional nice properties of this algorithm are:

- It enforces very accurately mass conservation in the computed vector fields, considering that the approximation is second order.
- The number of iterations is reduced from several hundreds to less than 7 in the examples considered in this article. Also, the number of iterations of the *PCG-algorithm* is almost independent with respect to mesh refinement, while the number of iterations in the non-preconditioned algorithm doubles at each mesh refinement in most cases. Therefore, there is a substantial reduction in the computational time in both examples. However, this behavior must be corroborated in 3-D problems with adaptive meshes in complex

domains, and with millions of degrees of freedom, in order to have a better idea of the performance of the method with realistic scenarios.

- It is not necessary to impose boundary conditions on the multiplier  $p$ , as it is done with the elliptic approaches. Furthermore, post-processing the scalar potential to find the vectorial field from the multiplier is not required, since the multiplier and the solenoidal vector field are found simultaneously by the algorithm.

The application of the methodology presented here to the more realistic three-dimensional case is an extension of the present work.

## Acknowledgements

Sincere thanks to the members of JAMP for their professional attitude of high quality.

## Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

## References

- [1] Agashe, S. (2021) New Formulas and Results for 3-Dimensional Vector Fields. *Applied Mathematics*, **12**, 1058-1096. <https://doi.org/10.4236/am.2021.1211069>
- [2] Bhatia, H., Norgard, G., Pascucci, V. and Bremer, P. (2012) The Helmholtz-Hodge Decomposition—A Survey. *IEEE Transactions on Visualization and Computer Graphics*, **19**, 1386-1404. <https://doi.org/10.1109/TVCG.2012.316>
- [3] Ahusborde, E., Azaez, M., Caltagirone, J.P., Gerritsma, M. and Lemoine, A. (2014) Discrete Hodge Helmholtz Decomposition. *Monografías Matemáticas Garca de Galdeano*, **39**, 1-10.
- [4] Mansour, N.N., Kosovichev, A., Georgobiani, D., Wray, A. and Miesch, M. (2004) Turbulence Convection and Oscillations in the Sun. *Proceedings of the SOHO14/GONG Workshop*, New Haven, 12-16 July 2004, 164-171.
- [5] Akram, M. and Michel, V. (2010) Regularisation of the Helmholtz Decomposition and its Application to Geomagnetic Field Modelling. *GEM—International Journal on Geomathematics*, **1**, 101-120. <https://doi.org/10.1007/s13137-010-0001-y>
- [6] Guo, Q., Mandal, M.K., Liu, G. and Kavanagh, K.M. (2006) Cardiac Video Analysis Using Hodge-Helmholtz Field Decomposition. *Computers in Biology and Medicine*, **36**, 1-20. <https://doi.org/10.1016/j.compbiomed.2004.06.011>
- [7] Gao, H., Mandal, M., Guo, G. and Wan, J. (2010) Singular Point Detection Using Discrete Hodge Helmholtz Decomposition in Fingerprint Images. 2010 *IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, 14-19 March 2010, 1094-1097. <https://doi.org/10.1109/ICASSP.2010.5495348>
- [8] Guo, Q., Mandal, M.K. and Li, M.Y. (2005) Efficient Hodge-Helmholtz Decomposition of Motion Fields. *Pattern Recognition Letters*, **26**, 493-501. <https://doi.org/10.1016/j.patrec.2004.08.008>
- [9] Hinkle, J., Fletcher, P.T., Wang, B., Salter, B. and Joshi, S. (2009) 4D MAP Image Reconstruction Incorporating Organ Motion. In: Prince, J.L., Pham, D.L. and Myers, K.J., Eds., *Information Processing in Medical Imaging*, Springer, Heidelberg,

- 676-687. [https://doi.org/10.1007/978-3-642-02498-6\\_56](https://doi.org/10.1007/978-3-642-02498-6_56)
- [10] Polthier, K. and Preu, E. (2003) Identifying Vector Fields Singularities Using a Discrete Hodge Decomposition. In: Hege, H.C. and Polthier, K., Eds., *Visualization and Mathematics III*, Springer, Heidelberg, 113-134. [https://doi.org/10.1007/978-3-662-05105-4\\_6](https://doi.org/10.1007/978-3-662-05105-4_6)
- [11] Tong, Y., Lombeyda, S., Hirani, A. and Desbrun, M. (2003) Discrete Multiscale Vector Field Decomposition. *ACM Transactions on Graphics*, **22**, 445-452. <https://doi.org/10.1145/882262.882290>
- [12] Foster, N. and Metaxas, D. (1997) Modeling the Motion of a Hot, Turbulent Gas. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, 3-8 August 1997, 181-188. <https://doi.org/10.1145/258734.258838>
- [13] Ingber, M.S. and Kempka, S.N. (2001) A Galerkin Implementation of the Generalized Helmholtz Decomposition for Vorticity Formulations. *Journal of Computational Physics*, **169**, 215-237. <https://doi.org/10.1006/jcph.2001.6724>
- [14] Deriaz, E. and Perrier, V. (2006) Divergence-Free and Curl-Free Wavelets in 2D and 3D: Application to Turbulent Flows. *Journal of Turbulence*, **7**, 1-37. <https://doi.org/10.1080/14685240500260547>
- [15] Deriaz, E. and Perrier, V. (2008) Direct Numerical Simulation of Turbulence Using Divergence-Free Wavelets. *Multiscale Modeling and Simulation*, **7**, 1101-1129. <https://doi.org/10.1137/070701017>
- [16] Deriaz, E. and Perrier, V. (2008) Orthogonal Helmholtz Decomposition in Arbitrary Dimension Using Divergence-Free and Curl-Free Wavelets. *Applied and Computational Harmonic Analysis*, **26**, 249-269. <https://doi.org/10.1016/j.acha.2008.06.001>
- [17] Colin, F., Egli, R. and Lin, F. (2006) Computing a Null Divergence Velocity Field Using Smoothed Particle Hydrodynamics. *Journal of Computational Physics*, **217**, 680-692. <https://doi.org/10.1016/j.jcp.2006.01.021>
- [18] Petronetto, F., Paiva, A., Lage, M., Tavares, G., Lopes, H. and Lewiner, T. (2010) Meshless Helmholtz-Hodge Decomposition. *IEEE Transactions on Visualization and Computer Graphics*, **16**, 338-349. <https://doi.org/10.1109/TVCG.2009.61>
- [19] Stam, J. and Fiume, E. (1995) Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes. *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Technique*, Los Angeles, 6-11 August 1995, 129-136. <https://doi.org/10.1145/218380.218430>
- [20] Fuselier, E.J. (2006) Refined Error Estimates for Matrix-Valued Radial Basis Functions. Ph.D. Thesis, Texas A&M University, College Station.
- [21] Macedo, I. and Castro, R. (2010) Learning Divergence-Free and Curl-Free Vector Fields with Matrix-Valued Kernels. Technical Report, Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil.
- [22] Narcowich, F.J. and Ward, J.D. (1994) Generalized Hermite Interpolation via Matrix-Valued Conditionally Positive Definite Functions. *Mathematics of Computation*, **63**, 661-687. <https://doi.org/10.1090/S0025-5718-1994-1254147-6>
- [23] López-López, J., Juárez, L.H. and Sandoval, M.L. (2016) Improving the Reconstruction of Vector Fields Using Mixed Finite Element Methods and Optimal Preconditioning. *Numerical Methods for Partial Differential Equations*, **32**, 1137-1154. <https://doi.org/10.1002/num.22044>