On numerical results of an inverse Cauchy problem modelling the air flow in the bronchial tree

immediate

Abstract

This paper is concerned by the resolution of an inverse Cauchy problem governed by Stokes equation modelling the air flow in the lungs. It consists in determining the air velocity and pressure on the artificial boundaries of the bronchial tree, this data completion problem is one of highly ill-posed problems in the sense of Hadamard [12]. This gives a great importance of their formulations. The main idea of this work is to develop some regularizing, stable and fast iterative algorithms based on domain decomposition method. We discuss the efficiency and the feasibility of the proposed approach by some numerical tests. Finally, we exploit Robin-Robin algorithm for the numerical simulation of the air flow in the bronchial tree configuration.

1 Introduction

Recently, data completion problems and their applications in measurement and data imaging technologies give rise constantly to a great scientific interest. In particular, the development and the improvement of new diagnostic techniques for assessment of lung pathology, namely in case of such disorders of the lungs mechanism as asthma and chronic obstructive pulmonary disease (COPD) [20]. The spirometry can be considered as the most common test of the lung function which is including the forced expiration manoeuvre. It consists in as deep and strong as possible expiration preceded by maximum inhalation. It has been shown that the registered maximum expiratory flow-volume (MEFV) curve is highly reproducible, effort-independent and simultaneously sensitive to respiratory disorders [14]. For many reasons, namely, the respiratory system complexity or limitations of existing diagnostic techniques, the problems of deeper insight into the lung structure and function as well as development and testing of new diagnostic techniques cannot be addressed experimentally. A complementary approach consists in conducting research via mathematical modelling. Computer-implemented models enable a wide range of simulations mimicking natural lung behaviour, including pathological alternations. Finding answers to specific questions about the system under investigation using numerical simulations is called the forward problem solving. On the contrary, applying mathematical models to determine system parameters from the raw measurement data, as e.g. retrieving changes in airway mechanics from the MEFV curve. This situation take part of an inverse problem solving. In this context, since the MEFV curve is in fact based on the formula of the volume flow rate which is related necessarily to the air velocity and the pressure in the airways of the bronchial tree. So it's interesting to determine the couple velocity-pressure especially at the inaccessible part called the artificial boundaries. This will be our framework for the numerical study of this data completion problem.

This class of problems occur when we deal with partial differential equations for which boundary conditions are incomplete. More precisely, it appears in the case where boundary data are over-specified on a part of the boundary, while no data are available on its remaining part. Thus data completion inverse problems consist in recovering the lacking boundary data. They are typically ill-posed in Hadamard's sense [12], since they are known by their severe instability. That is small perturbation on data generates big error on the solution. This gives great importance to their formulation. Despite the complexity of these problems, it still attracting the interest of the scientific community. This explains the intense and immense contributions in the literature, as well as various approaches, to the theoretical and numerical

solutions. In particular, a special attention is given to the proposed approaches for solving these problems. So, several numerical methods have been investigated in the last three decades. We mention the method of quasi-reversibility, first introduced by Lions and al [19], and many research works have been conducted to propose different regularization approaches, we can not list all of them. The most popular one is Tikhonov's regularization method [25], which transforms the original ill-posed problem into a well posed one by minimizing the L^2-norm of the solution subjected to constraint equations. Then, other methods have been proposed to regularize the Cauchy problem, we can mention for instance the alternating method [5,18], the universal regularization method [17], the technic of fundamental solution [8], the fixed point approach inspired from a domain decomposition like methods [7] etc. Some iterative methods are based on the use of a sequence of well-posed problems [15, 16, 18] and others on the minimization of a cost functional [2, 6].

We are interested in solving a Cauchy inverse problem governed by the Stokes equation in spirometry which is modelling the air flow inside the bronchial tree [4]. So, we aim to reconstruct the air velocity and the pressure on the artificial boundaries based on the measurements available on the other part of the boundary. The approach adopted here is based on the relationship between the Cauchy problem and domain decomposition method proposed recently in [7]. Thereby the Neumann-Neumann and Robin-Robin algorithms will be investigated for the resolution comparing with the KMF algorithm firstly introduced by Kozlov-Maz'iya-Fomin [18].

This paper is outlined as follow. The next section is devoted to the formulation of the inverse Cauchy problem for the Stokes system. In section 3, we are interested by the numerical study of alternating methods based on DDM where a stopping criterion is proposed. In section 4, we present a comparison between different algorithms by introducing some numerical tests using the $\mathbb{P}_{1Bubble}/\mathbb{P}_1$ finite element [3] for the direct discrete problems. Finally, we are interested by the numerical simulation of the bronchial tree configuration where Robin-Robin algorithm is considered for the resolution. This choice is approved by the comparison established between different algorithms, some numerical results of the air velocity and the pressure on the artificial boundaries of the bronchial tree are presented. The numerical stability of the solution is also discussed.

2 Inverse problem setting for the air flow

We consider the inverse Cauchy problem modelling the air flow inside the bronchial tree, by simulating approximatively the structure of such tree as 2D geometric domain (see figure 1). The inverse problem consists in determining the couple velocity-pressure (u, p) over the inaccessible boundaries Γ_i, for $i = 1, ..., N$ (Here $N = 8$), solution of

$$\begin{cases} -\mu \operatorname{div}(D(u)) + \nabla p = f & \text{dans} \quad \Omega \\ \operatorname{div}(u) = 0 & \text{dans} \quad \Omega \\ u = h & \text{sur} \quad \Gamma_0 \\ \sigma(u)n = g & \text{sur} \quad \Gamma_0 \\ u = 0 & \text{sur} \quad \Gamma_l \end{cases} \tag{1}$$

where $D(u)$ is the deformation tensor defined by

$$D(u) = \frac{1}{2}(\nabla u + (\nabla u)^t),$$

$\sigma(u)$ is the Cauchy tensor given by :

$$\sigma(u) = 2\mu D(u) - p\, I, \quad I \text{ is the identity matrix}$$

n is the normal vector out-directed from Ω, $f \in L^2(\Omega)^d$, $g \in H_{00}^{-\frac{1}{2}}(\Gamma_3)^d$ and $h \in H^{\frac{1}{2}}(\Gamma_1)^d$ are given functions where the space $H^{\frac{1}{2}}(\Gamma)^d$ can be defined as follow.

$$H^{\frac{1}{2}}(\Gamma)^d = \{ \quad v_{/\Gamma} \quad / \quad v \in H^1(\Omega)^d \}$$

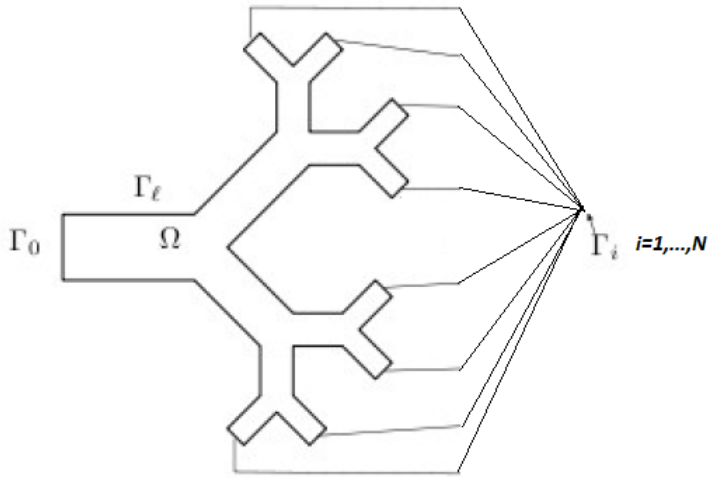


Figure 1: Diagram simulating the bronchial tree studied

While $H_{00}^{-\frac{1}{2}}(\Gamma_0)^d$ is the dual space of :

$$H_{00}^{\frac{1}{2}}(\Gamma_0)^d = \{ \quad \varphi \in H^{\frac{1}{2}}(\Gamma_0)^d \quad / \quad \exists v \in H^1(\Omega)^d \quad \text{such that} \quad v_{/\Gamma_0} = \varphi, \quad v_{/\partial\Omega/\Gamma_0} = 0, \}$$

and μ is the kinematic viscosity coefficient. In the problem 1, we note the existence of two conditions of type Dirichlet and Neumann over the accessible boundary Γ_0 (the proximal part), we suppose that the air velocity is null on the boundary Γ_l, while the solution is completely unknown on the artificial boundaries Γ_i for $i = 1, ..., 8$. We consider firstly a rectangular domain for the resolution, before dealing with the main problem.

3 Numerical study of alternating method based on DDM

In this section, we will present some algorithms for the Cauchy problem resolution derived from the domain decomposition techniques. Namely, Neumann-Neumann and Robin-Robin algorithms. The KMF algorithm is also considered here which allow us to make a comparison between different algorithms. Let $\Omega =]0, 1[\times]0, 1[$, where the boundary $\partial\Omega = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ (See Figure 2).

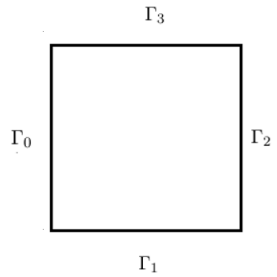


Figure 2: Domain Ω.

We consider the problem : find $(u,p)_{|\Gamma_0}$ solution of the following inverse Cauchy problem

$$(P)\begin{cases} -2\mu\,\mathrm{div}(D(u)) + \nabla p = f & \text{in} \quad \Omega \\ \mathrm{div}(u) = 0 & \text{in} \quad \Omega \\ u = h_1 & \text{on} \quad \Gamma_1 \\ u = h_2 & \text{on} \quad \Gamma_2 \\ \sigma(u)n = g_1 & \text{on} \quad \Gamma_2 \\ \sigma(u)n = g_2 & \text{on} \quad \Gamma_3 \end{cases} \tag{2}$$

where f, g_1, g_2, h_1, h_2 are given functions. In the problem 2, the solution on Γ_0 is unknown and must be determined from the boundary conditions on Γ_1, Γ_2 and Γ_3. In order to reconstruct these unknown boundary conditions based on the supplementary data provided on the other boundaries, we opt for a methodology based on the domain decomposition operating mode.

The decomposition domain like algorithms that we use to solve the Cauchy problem requires the resolution of the following well posed problems. Given an initial approximation $v^{(n)}$ of u on Γ_0, so we solve

$$(P^{(2n)})\begin{cases} -2\mu\,\mathrm{div}(D(u^{2n})) + \nabla p^{2n} = f & \text{in} \quad \Omega \\ \mathrm{div}(u^{2n}) = 0 & \text{in} \quad \Omega \\ \sigma(u^{2n})n = v^{(n)} & \text{on} \quad \Gamma_0 \\ u^{(2n)} = h_1 & \text{on} \quad \Gamma_1 \\ \sigma(u^{2n})n = g_1 & \text{on} \quad \Gamma_2 \\ \sigma(u^{2n})n = g_2 & \text{on} \quad \Gamma_3 \end{cases} \tag{3}$$

then by taking $v^{(n+1)} = u^{(2n)}$, we solve

$$(P^{(2n+1)})\begin{cases} -2\mu\,\mathrm{div}(D(u^{2n+1})) + \nabla p^{2n+1} = f & \text{in} \quad \Omega \\ \mathrm{div}(u^{2n+1}) = 0 & \text{in} \quad \Omega \\ \sigma(u^{(2n+1)})n = \sigma(u^{(2n)})n & \text{on} \quad \Gamma_0 \\ u^{(2n+1)} = h_1 & \text{on} \quad \Gamma_1 \\ u^{(2n+1)} = h_2 & \text{on} \quad \Gamma_2 \\ \sigma(u^{2n+1})n = g_2 & \text{on} \quad \Gamma_3 \end{cases} \tag{4}$$

We start by recalling the classical KMF algorithm.

Algorithm 1. ***Classical KMF***

1. *Given $\varepsilon > 0$ and an initial approximation $v^{(0)}$ of u on Γ_0 satisfying the compatibility conditions.*
2. *Solve the problem $P^{(2n)}$*
3. *Solve the problem $P^{(2n+1)}$*
4. *Actualize $v^{(n+1)} = u^{(2n-1)}, \qquad n \geq 1$*
5. *If $\|u^{(2n)} - u^{(2n+1)}\|_{L^2(\Gamma_0)} \leq \varepsilon$ stop. Else go to step 2.*

In the following, the KMF algorithm can be relaxed with dynamic θ proposed by Jourhmane and Nachaoui [15]. Then we have

Algorithm 2. ***Relaxed KMF with dynamic θ***

1. *Given $\varepsilon > 0$ and an initial approximation $v^{(0)}$ of u on Γ_0 satisfying the compatibility conditions.*
2. *Solve the problem $P^{(2n)}$.*

3. *Solve the problem* $P^{(2n+1)}$.

4. *Compute* $e^{(2n)} = u_{\Gamma_0}^{(2n)} - u_{\Gamma_0}^{(2(n-1))}$, $e^{(2n+1)} = u_{\Gamma_0}^{(2n+1)} - u_{\Gamma_0}^{(2(n-1))}$.

5. $\theta^{(n)} = \frac{<e^{(2n)}, e^{(2n)} - e^{(2n+1)}>}{\|e^{(2n)} - e^{(2n+1)}\|_{L^2(\Gamma_0)}} \qquad n \geq 1.$

6. *Compute* $v^{(n)} = \theta^{(n)} u^{(2n-1)} + (1 - \theta^{(n)}) v^{(n-1)} \qquad , n \geq 1$

7. *If* $\|u^{(2n)} - u^{(2n+1)}\|_{L^2(\Gamma_0)} \leq \varepsilon$ *stop. Else go to step 2.*

Now, we present the algorithms derived from the DD method. Namely Neumann-Neumann and Robin-Robin algorithms.

Algorithm 3. ***Algorithm derived from Neumann-Neumann DD method***

1. *For* $n = 0$ *give* $s_1, s_2 > 0$ *two positive averaging coefficients such that* $s_1 + s_2 > 0$, $\varepsilon > 0$, $\theta > 0$ *and an initial approximation* $v^{(n)}$ *of* u *on* Γ_0 *satisfying compatibility conditions.*

2. *Solve the problem* $P^{(2n)}$.

3. *Solve the problem* $P^{(2n+1)}$ *with the following condition* $u^{(2n+1)} = v_n$ *on* Γ_0.

4. *Solve the problem* $P^{(2n)}$ *for* $\Psi^{(2n)}$ *with the following condition on* Γ_0

$$\Psi^{(2n)} = s_1 \sigma(u^{(2n)}) n - s_2 \sigma(u^{(2n+1)}) n.$$

5. *Solve the problem* $P^{(2n+1)}$ *for* $\Psi^{(2n+1)}$ *with the following condition on* Γ_0

$$\Psi^{(2n+1)} = s_1 \sigma(u^{(2n)}) n - s_2 \sigma(u^{(2n+1)}) n.$$

6. *Compute* $v^{(n)} = v^{(n)} - \theta(s_1 \Psi_{|\Gamma_0}^{(2n)} - s_2 \Psi_{|\Gamma_0)}^{(2n+1)}) \qquad , n \geq 1.$

7. *If* $\|u^{(2n)} - u^{(2n+1)}\|_{L^2(\Gamma_0)} \leq \varepsilon$ *stop. Else* $n = n + 1$ *go to step 2.*

Algorithm 4. ***Algorithm derived from Robin-Robin DD method***

1. *For* $n = 0$ *give* $\varepsilon > 0$, $\theta > 0$ *and an initial approximation* $v^{(n)}$ *of* u *on* Γ_0 *satisfying the compatibility conditions.*

2. *Solve the problem* $P^{(2n)}$ *with the following condition on* Γ_0

$$\sigma(u^{(2n)}) n + \gamma_1 u^{(2n)} = \sigma(u^{(2n+1)}) n + \gamma_1 u^{(2n+1)}$$

3. *Solve the problem* $P^{(2n+1)}$ *with the following condition on* Γ_0

$$\sigma(u^{(2n+1)}) n - \gamma_2 u^{(2n+1)} = \sigma(u^{(2n)}) n - \gamma_2 u^{(2n)}$$

4. *If* $\|u^{(2n)} - u^{(2n+1)}\|_{L^2(\Gamma_0)} \leq \varepsilon$ *stop. Else* $n = n + 1$ *go to step 2.*

where $v^{(0)}$ *is given, and* γ_1 *and* γ_2 *are non-negative acceleration parameters satisfying* $\gamma_1 + \gamma_2 > 0$.

3.1 Proposed stopping criterion

The classical stopping criterion commonly used is the successive difference between the solution of actual iteration and the preview one on Γ_0 *defined by* $\|u^{(2n)} - u^{(2n-2)}\|_{\Gamma_0}$. *According to the relationship between Cauchy problem and domain decomposition methods in rectangular domain, it's more significant that at the convergence of proposed algorithms the solutions* $u^{(2n)}$ *and* $u^{(2n+1)}$ *must be identical on all the boundary* $\partial\Omega$. *This allows us to develop a more efficient stopping criterion than the usual ones. So, it's more logical to use a stopping criterion on the norm of the gap* $u^{(2n)} - u^{(2n+1)}$ *in* L^2 *on all the boundary* $\partial\Omega$, *where* $(u^{(2n)}, p^{(2n)})$ *is the solution of* $P^{(2n)}$ *and* $(u^{(2n+1)}, p^{(2n+1)})$ *is the solution of* $P^{(2n+1)}$. *We denote by* $\Gamma_c := \Gamma_2 \cup \Gamma_3 \cup \Gamma_0$, *the proposed stopping criterion is*

$$\|u^{(2n)} - u^{(2n+1)}\|_{0,\Gamma_c} \tag{5}$$

In the sequel, in order to understand the behavior of this stopping criterion compared to the others, we seek to approach the solution of the inverse Cauchy problem (2) defined by the following data. In this example which is the same one used in [7], we aim to approximate the solution of the inverse problem (2) associated to the following data :

- *Dirichlet data* Γ_1, *noted by* $h_{1|\Gamma_1} = (h_1^{(1)}, h_1^{(2)})$, *is given by:*

$$\begin{cases} h_1^{(1)}(x,y) = \cosh(x)\sinh(0), \\ h_1^{(2)}(x,y) = -\cosh(0)\sinh(x). \end{cases} \tag{6}$$

- *The Dirichlet and Neumann data over* Γ_2, *noted respectively by* $h_{2|\Gamma_2} = (h_2^{(1)}, h_2^{(2)})$ *and* $g_{1|\Gamma_2} = (g_1^{(1)}, g_1^{(2)})$, *are given by :*

$$\begin{cases} h_2^{(1)}(x,y) = \cosh(1)\sinh(y), \\ h_2^{(2)}(x,y) = -\cosh(y)\sinh(1), \end{cases} \tag{7}$$

$$\begin{cases} g_1^{(1)}(x,y) = \mu\,\sinh(x)\sinh(y) - \left(y\cosh(x) - \dfrac{\sinh(1)}{1}\right), \\ g_1^{(2)}(x,y) = -\mu\,\cosh(x)\cosh(y). \end{cases} \tag{8}$$

- *The Neumann data over* Γ_3, *noted by* $g_{2|\Gamma_3} = (g_2^{(1)}, g_2^{(2)})$, *is given by :*

$$\begin{cases} g_2^{(1)}(x,y) = \mu\,\cosh(y)\cosh(x), \\ g_2^{(2)}(x,y) = -\mu\,\sinh(y)\sinh(x) - \left(y\cosh(x) - \dfrac{\sinh(1)}{2}\right). \end{cases} \tag{9}$$

- *The source term over* Ω *noted by* $f = (f_1, f_2)$ *which is given by :*

$$\begin{cases} f_1(x,y) = -2\mu\,\cosh(x)\sinh(y) + y\sinh(x), \\ f_2(x,y) = 2\mu\,\cosh(y)\sinh(x) + \cosh(x), \end{cases} \tag{10}$$

where $\mu = 0.1$,

We reconstruct the velocity and the pressure over the inaccessible part of the boundary Γ_0, *by using the above data associated to the following exact solution :*

$$\begin{cases} ue_1(x,y) = \cosh(x)\sinh(y), \\ ue_2(x,y) = -\cosh(y)\sinh(x), \\ pe(x,y) = y\cosh(x) - \frac{\sinh(1)}{2}. \end{cases} \tag{11}$$

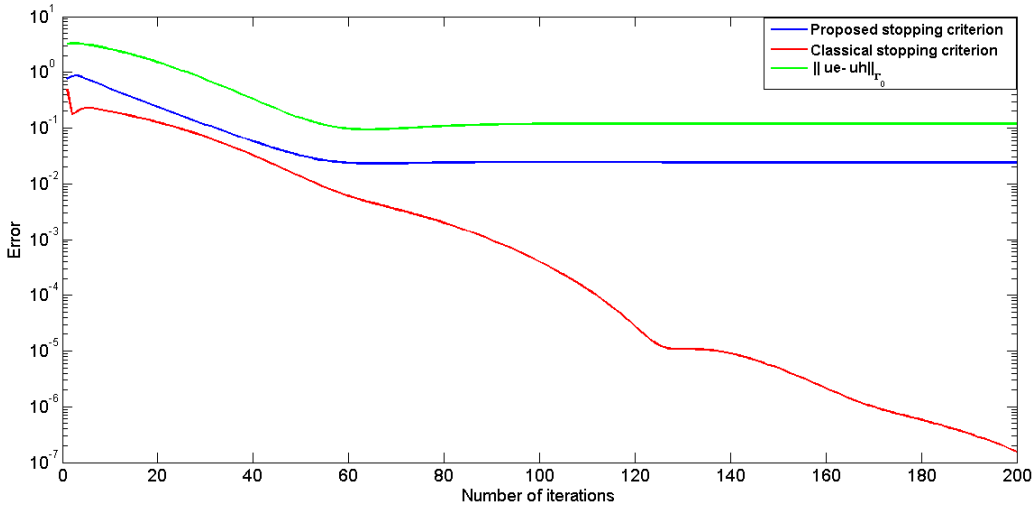


Figure 3: The behavior of proposed stopping criterion, the classical stopping criterion and the error on Γ_c obtained by the algorithm classical KMF

The approximation is made by the following algorithms and $\mathbb{P}_{1Bubble}/\mathbb{P}_1$ finite elements method. In order to illustrate the efficiency of the stopping criterion, denote by n_e the number of iterations made by an algorithm when the criterion is satisfied. As clearly shown in the figure 3, the classical stopping criterion does not follow the behavior of the error $(u_e - u_h)_{\Gamma_0}$, which explains the large number of iterations in order to achieve the convergence. While the proposed stopping criterion follows faithfully the behavior of error. This allows to stop the algorithm with the best precision and with a minimum number of iterations equal to $ne = 59$ iterations in this case.

In order to examine our proposed stopping criterion for the relaxed KMF with dynamic θ, we present in the figure 4 the comparison of the gape between the exact velocity u_e and the approximate one u_h obtained by the algorithm 2.

We remark from Figure 4 that the new stopping criterion has accurately reflects the behavior of the error on Γ_0. Furthermore, the number of iterations at the convergence is reduced, with $ne = 38$ iterations. Beside, the proposed stopping criterion follow the same behavior as the usual error calculated on the boundary part Γ_0 for Neumann-Neumann and Robin-Robin algorithms, this is clearly seen in the figure 5.

Remark 1. *We note that all the iterative methods previously mentioned give arise to a common behavior of the approximation process. Indeed, the error made on inaccessible boundary decreases until a threshold, then rises a little before its becoming stationary. Classical stopping criterion does not detect this threshold. Therefore, these stopping criterion stop the algorithms in the phase where the error stagnates. This observation prompted us to propose this stopping criterion which is more responsive and reflects the real behavior of the error on inaccessible part of the boundary. Consequently, this enable us to reduce significantly the computational cost by reducing the iterations number.*

In the following, we present comparison results concerning the accuracy of the reconstructed solution produced by KMF algorithm and each of the algorithms we proposed.

4 Numerical results and discussion

In order to prove the efficiency of the approach based on the domain decomposition methods, we present some numerical results obtained by KMF algorithm and some domain decomposition algorithms of type "Neumann-Neumann" and "Robin-Robin" to resolve the Cauchy problem governed by Stokes equation.

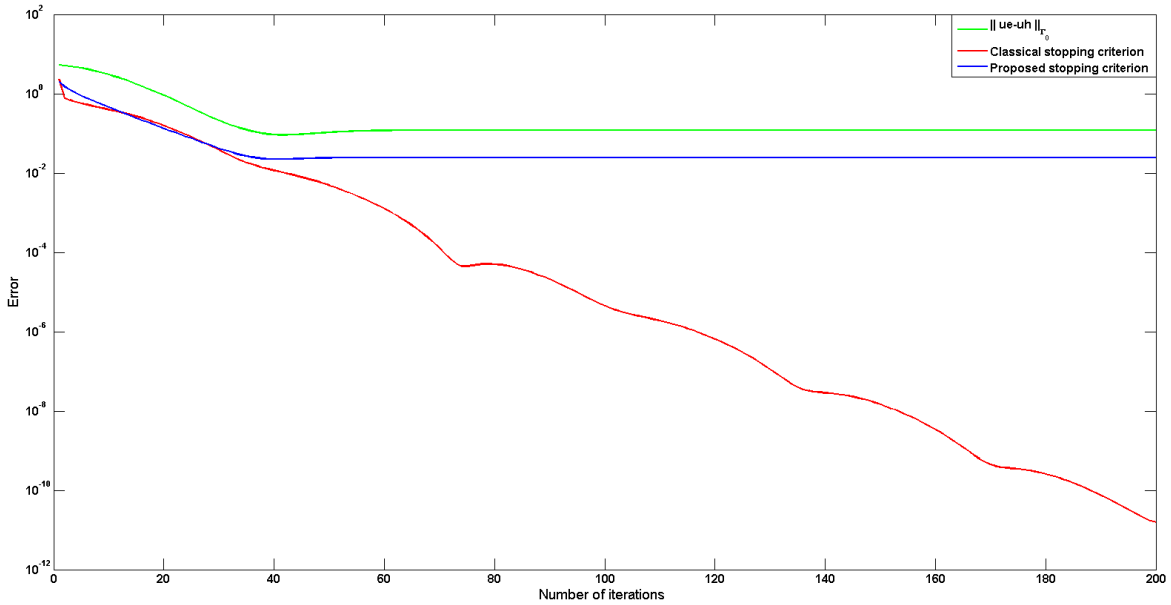
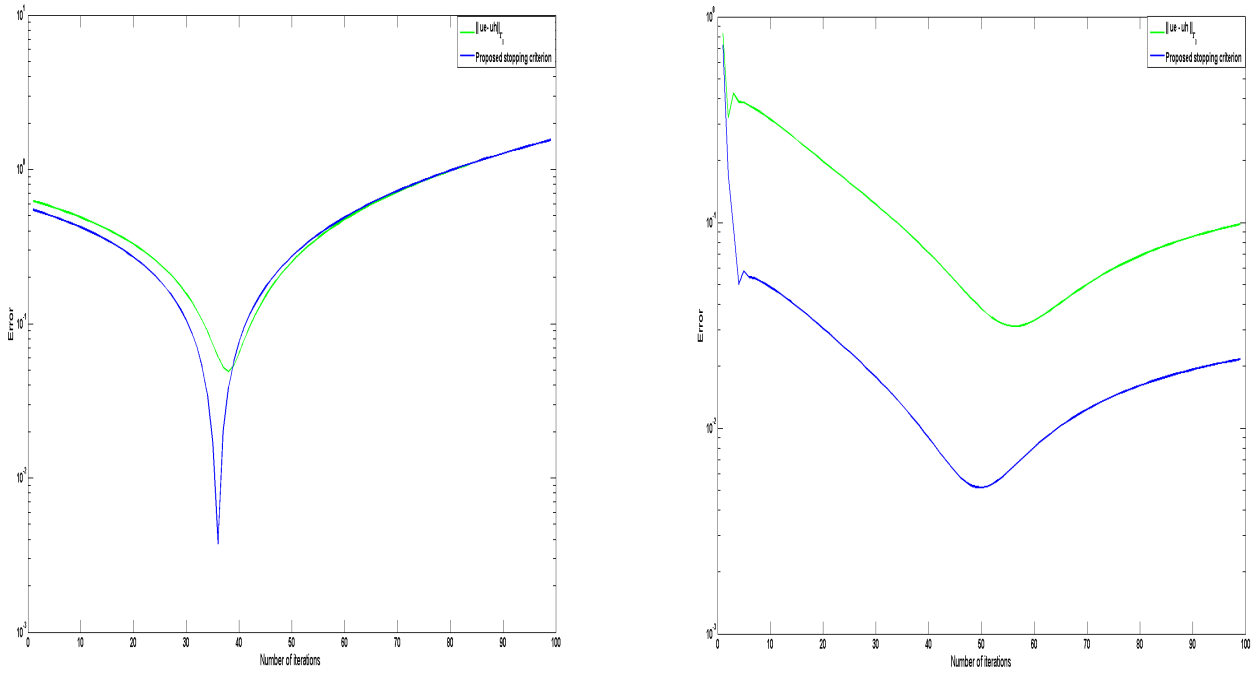


Figure 4: The behavior of proposed stopping criterion, the classical stopping criterion andthe error on Γ_c obtained by the algorithm KMF with dynamic θ



(a) Neumann-Neumann algorithm

(b) Robin-Robin algorithm

Figure 5: The proposed stopping criterion and the error on Γ_0 produced by the Neumann-Neumann and Robin-Robin algorithms

We aim to show that the methods deriving from the decomposition of the domain offer us robust, efficient and very useful alternatives for solving the inverse Cauchy problem. However the algorithms in question involve a number of parameters which are namely γ_1 and γ_2 for the "Robin-Robin" algorithm and θ, s_1 and s_2 for the "Neumann-Neumann" algorithm. For this, a sensitivity analysis is introduced, indeed the evolution of the error in function of these parameters is presented in order to choose the optimal ones.

4.1 Numerical tests

4.1.1 Exemple 1

We still consider the same example presented before. So, based on the data (6), (7), (8), (9) *and* (10) *and by chosen a compatible initial data, we present some numerical results showing the comparison between the exact and the approximate solutions by different algorithms. In the figures 6, 7 and 8, we present the exact and the approximate solutions associated to the two components of the velocity and the pressure. These obtained results indicate the accuracy of the solutions achieved by KMF, "Neumann-Neumann" and "Robin-Robin" algorithms. The comparison between different algorithms in function of the iterations*

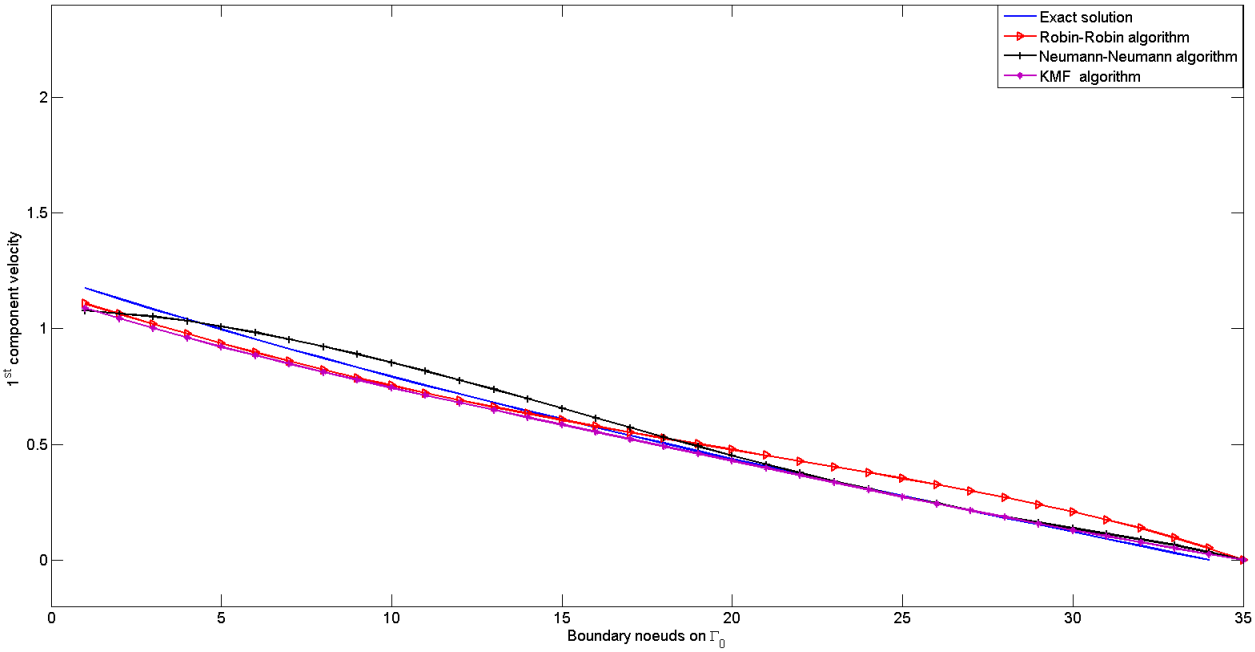


Figure 6: The exact and the approximate solutions of the first component of the velocity u_1.

number as well as the execution time was illustrated in the table 1, respectively for $\theta = 1.8$, $s_1 = 0.1$ and $s_2 = 1$ in the case of the "Neumann-Neumann" algorithm, for $\gamma_1 = 0.3$ and $\gamma_2 = 0.1$ for the "Robin-Robin" algorithm. In fact, this choice of the parameters was approved by a study of the error variation of the velocity and the pressure compared to those parameters. We can note that in addition to the accuracy obtained, by the algorithms derived from the domain decomposition technique, it offers an efficient and fast alternative approach allowing to solve numerically the inverse Cauchy problem. Regarding the choice of the adequate parameters θ, s_1 and s_2 used in "Neumann-Neumann" algorithm or γ_1 and γ_2 intervening in "Robin-Robin" algorithm, we give some guidelines based on a sensibility study, so we present firstly in the figures 9 and 10 the evolution of the velocity and the pressure errors compared to the values of s_1 and s_2 for a fixed value $\theta = 1$. We remark that for a fixed value $\theta = 1$, the "Neumann-Neumann" algorithm converges for (s_1, s_2) belongs to the interval $[0, 0.2] \times [0.6, 1]$ and diverges elsewhere. From these figures,

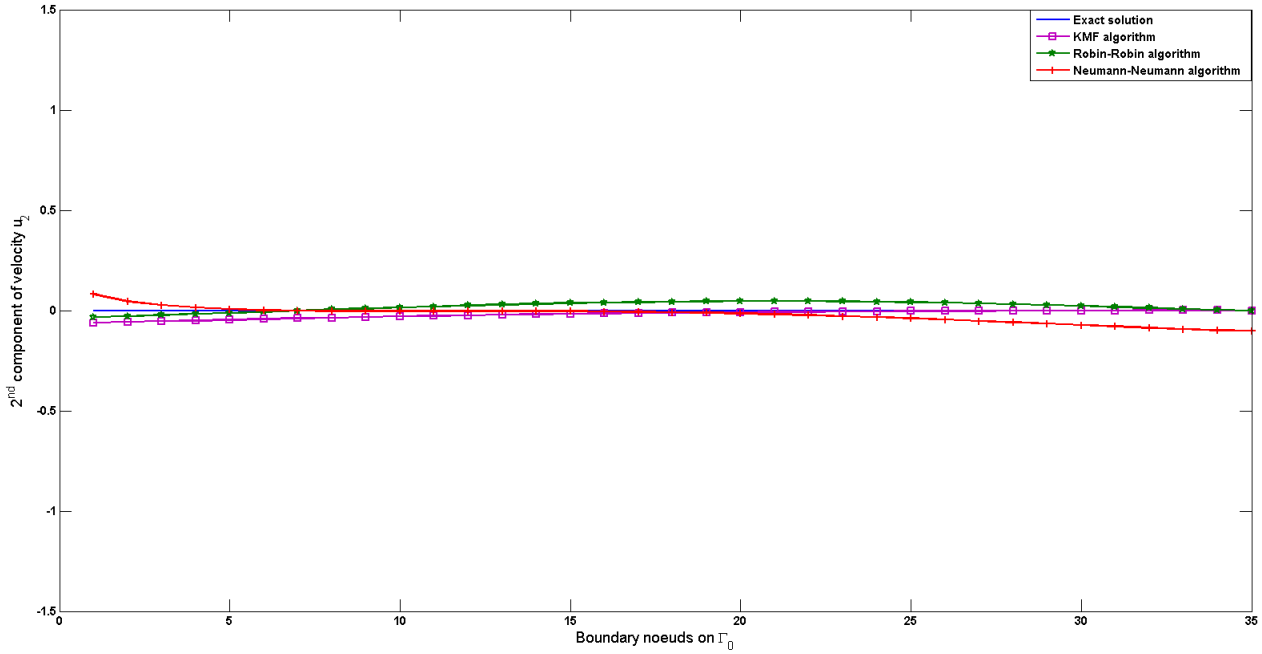


Figure 7: The exact and the approximate solutions of the second component of the velocity u_2.

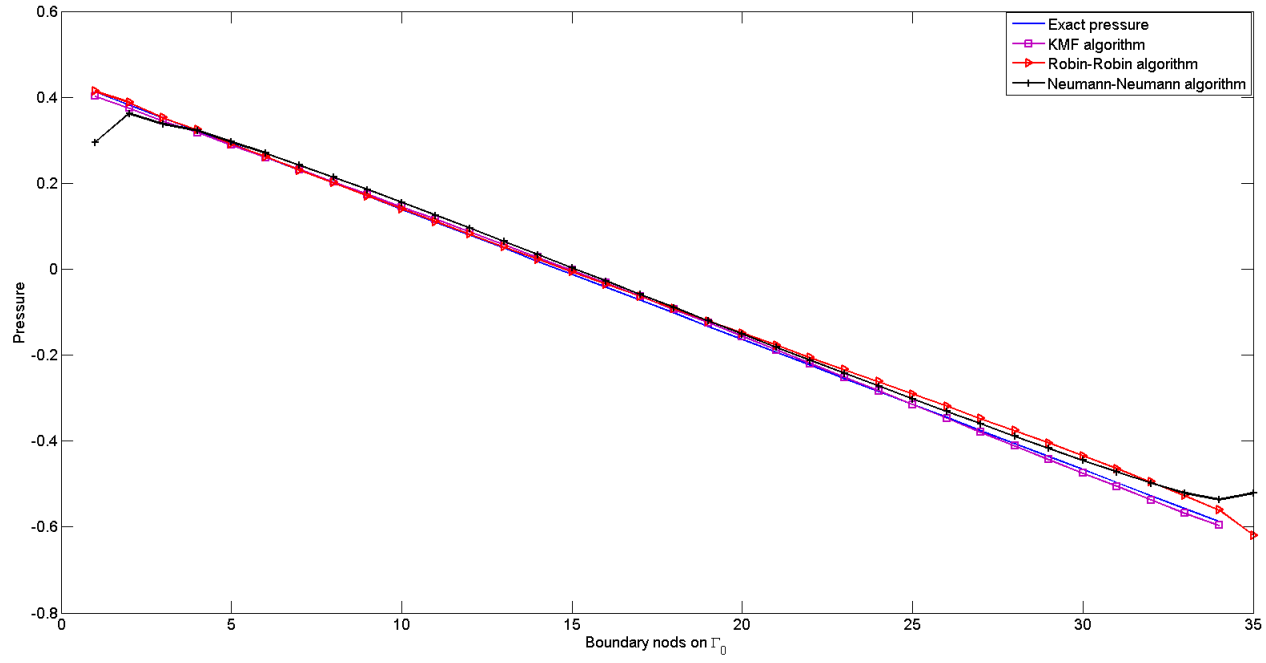


Figure 8: The exact and the approximate solutions of the pressure p.

the optimal value of (s_1, s_2) *correspondent to the optimal error associated to the velocity and the pressure is* $(0.1; 1)$. *Now, by fixing the value of* (s_1, s_2) *at the optimal one and varying* θ, *we present in the figure*

	Velocity error over Γ_0	iterations number	execution time (s)
KMF algorithm	0.034	64	93.92
Neumann-Neumann algorithm	0.035	55	60.11
Robin-Robin algorithm	0.031	37	19.5

Table 1: Comparison different algorithms for the parameters $h = \frac{1}{32}, \theta = 1.8$, $s_1 = 0.1, s_2 = 1, \gamma_1 = 0.3, \gamma_2 = 0.1$

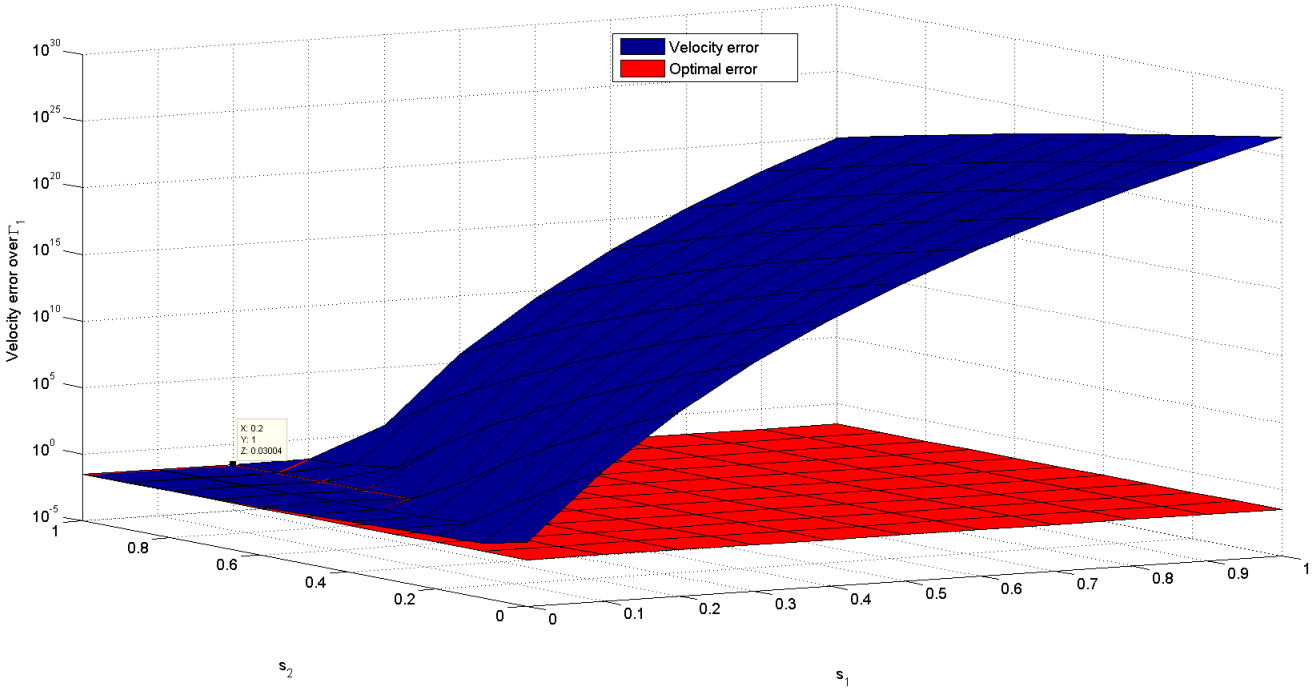


Figure 9: Evolution of the velocity error in function of s_1 and s_2 values for $\theta = 1$

11 the variation of the error in function of θ, we can then deduce that the optimal value is $\theta = 1.8$.

The sensitivity study of the parameters of "Robin-Robin" algorithm was studied in [7], where we can conclude from the evolution of the velocity and the pressure errors in function of γ_1 and γ_2 that "Robin-Robin" algorithm converges for (γ_1, γ_2) belongs to the interval $[0.3, 0.6] \times [0, 1]$ and diverges elsewhere. We deduce then that the optimal value of (γ_1, γ_2), correspondent to the optimal error associated to the velocity and the pressure, is $(0.3, 0.1)$.

In the following, we are interested in the numerical convergence of different algorithms previously presented.

4.2 Numerical Convergence

In this section, we study the convergence of the solution in function of the discretization step. Indeed, the Cauchy problem (2) was resolved for the data (6), (7), (8), (9) and (10) iteratively by using KMF, "Neumann-Neumann" and "Robin-Robin" algorithms, in order to reconstruct the solution over the inaccessible part of the boundary. So, by taking the discretization step $h \in \{1/16, 1/32, 1/64, 1/128\}$, we remark that those algorithms are convergent following the comparison results of the velocity and the pressure errors with respect to different values of h which are respectively illustrated in the tables 2, 3 and 4.

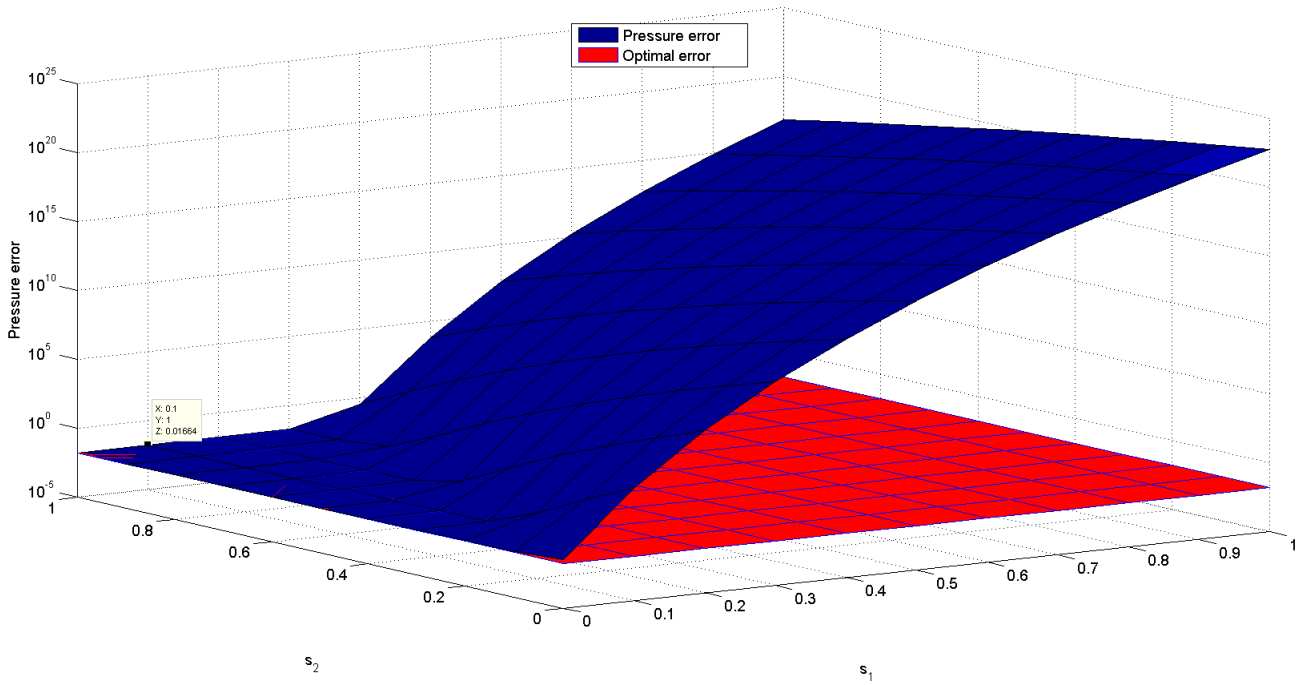


Figure 10: Evolution of the pressure error in function of s_1 and s_2 values for $\theta = 1$

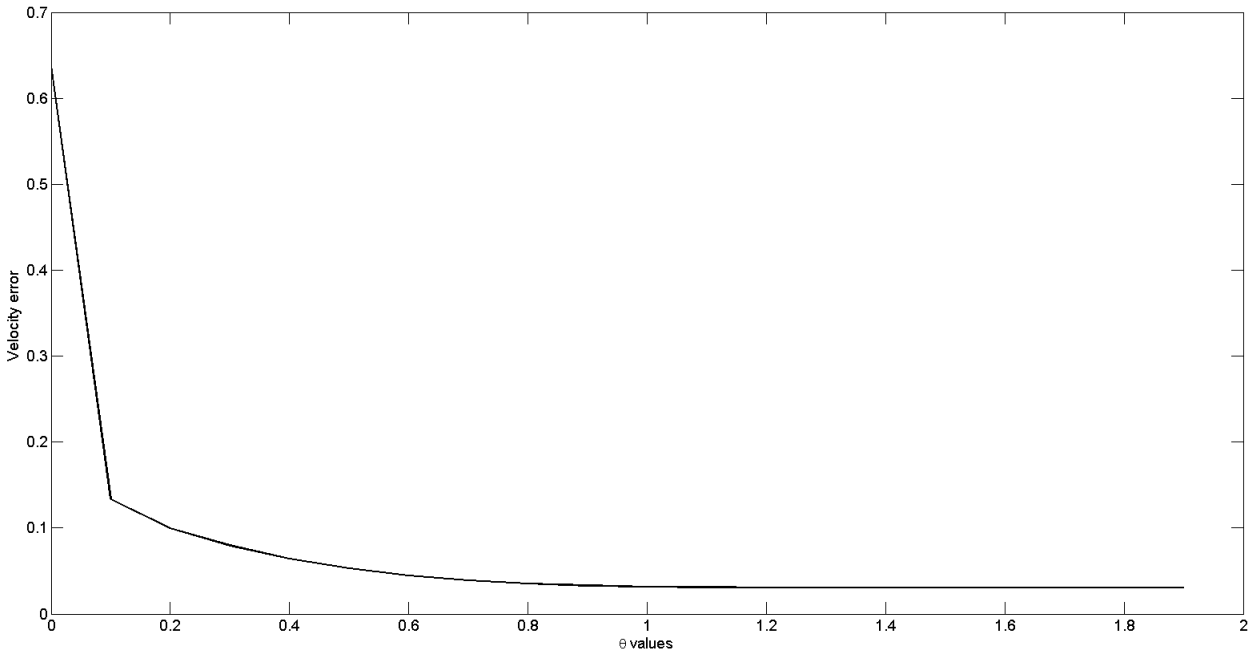


Figure 11: Error evolution in function of θ for $(s_1, s_2) = (0.1, 1)$

We note that error decrease for decreasing values of the discretization step.

h	1/16	1/32	1/64	1/128
Velocity component $u_{1\|\Gamma_0}$ error	$1.84\,10^{-2}$	$8.80\,10^{-3}$	$5.18\,10^{-3}$	$3.50\,10^{-3}$
Velocity component $u_{2\|\Gamma_0}$ error	$1.51\,10^{-2}$	$1.38\,10^{-2}$	$9.24\,10^{-3}$	$7.91\,10^{-3}$
Pressure $p_{\|\Gamma_0}$ error	$1.89\,10^{-2}$	$1.63\,10^{-2}$	$6.91\,10^{-3}$	$3.75\,10^{-3}$

Table 2: The obtained errors by the KMF algorithm for different values of h.

h	1/16	1/32	1/64	1/128
Velocity component $u_{1\|\Gamma_0}$ error	$4.78\,10^{-2}$	$4.18\,10^{-2}$	$3.11\,10^{-2}$	$3.30\,10^{-2}$
Velocity component $u_{2\|\Gamma_0}$ error	$6.63\,10^{-2}$	$3.07\,10^{-2}$	$3.99\,10^{-2}$	$3.96\,10^{-2}$
Pressure $p_{\|\Gamma_0}$ error	$1.78\,10^{-2}$	$1.65\,10^{-2}$	$1.80\,10^{-2}$	$2.07\,10^{-2}$

Table 3: The obtained errors by the Neumann-Neumann algorithm for different values of h.

h	1/16	1/32	1/64	1/128
Velocity component $u_{1\|\Gamma_0}$ error	$3.55\,10^{-1}$	$3.55\,10^{-2}$	$1.28\,10^{-2}$	$6.07\,10^{-3}$
Velocity component $u_{2\|\Gamma_0}$ error	$1.41\,10^{-1}$	$7.12\,10^{-2}$	$2.69\,10^{-2}$	$1.13\,10^{-2}$
Pressure $p_{\|\Gamma_0}$ error	$3.37\,10^{-1}$	$1.08\,10^{-2}$	$8.51\,10^{-3}$	$4.96\,10^{-3}$

Table 4: The obtained errors by the Robin-Robin algorithm for different values of h.

We are now interested in the comparison between different algorithms KMF, "Neumann-Neumann" and "Robin-Robin" in term of the evolution of the velocity and the pressure errors with respect to the iterations number. So, we present in the figure 12 the variation of the velocity component u_1 error over the inaccessible part of the boundary Γ_0 with respect to the iterations number, the same thing for the pressure in the figure 13.

We conclude from this comparison that the "Robin-Robin" algorithm allows to get an accurate solution with minimal iterations number, indeed , we can achieve the velocity component u_1 error over Γ_0 of 7.9310^{-3} in 56 iterations, and a pressure precision of 3.1410^{-3} in 42 iterations.

4.2.1 Exemple 2

we still consider the same rectangulaire domaine (Figure 2). In this example, we aim to approximate a trigonometric solution of the inverse Cauchy problem (2) associated to the following data:

- *Dirichlet data over Γ_1, noted by $h_{1|\Gamma_1} = (h_1^{(1)}, h_1^{(2)})$, is given by :*

$$\begin{cases} h_1^{(1)}(x,y) = 0, \\ h_1^{(2)}(x,y) = \sin(x). \end{cases} \tag{12}$$

- *Dirichlet and Neumann data over Γ_2, noted respectively by $h_{|\Gamma_2} = (h_2^{(1)}, h_2^{(2)})$ and $g_{1|\Gamma_2} = (g_1^{(1)}, g_1^{(2)})$, which are given by :*

$$\begin{cases} h_{02}^{(1)}(x,y) = -\cos(1)\sin(y), \\ h_{02}^{(2)}(x,y) = \sin(1)\cos(y), \end{cases} \tag{13}$$

$$\begin{cases} g_1^{(1)}(x,y) = \mu(\sin(x)\sin(y)) - 2\pi(\cos(2\pi y) - \cos(2\pi x)), \\ g_1^{(2)}(x,y) = \mu(\cos(x)\cos(y)). \end{cases} \tag{14}$$

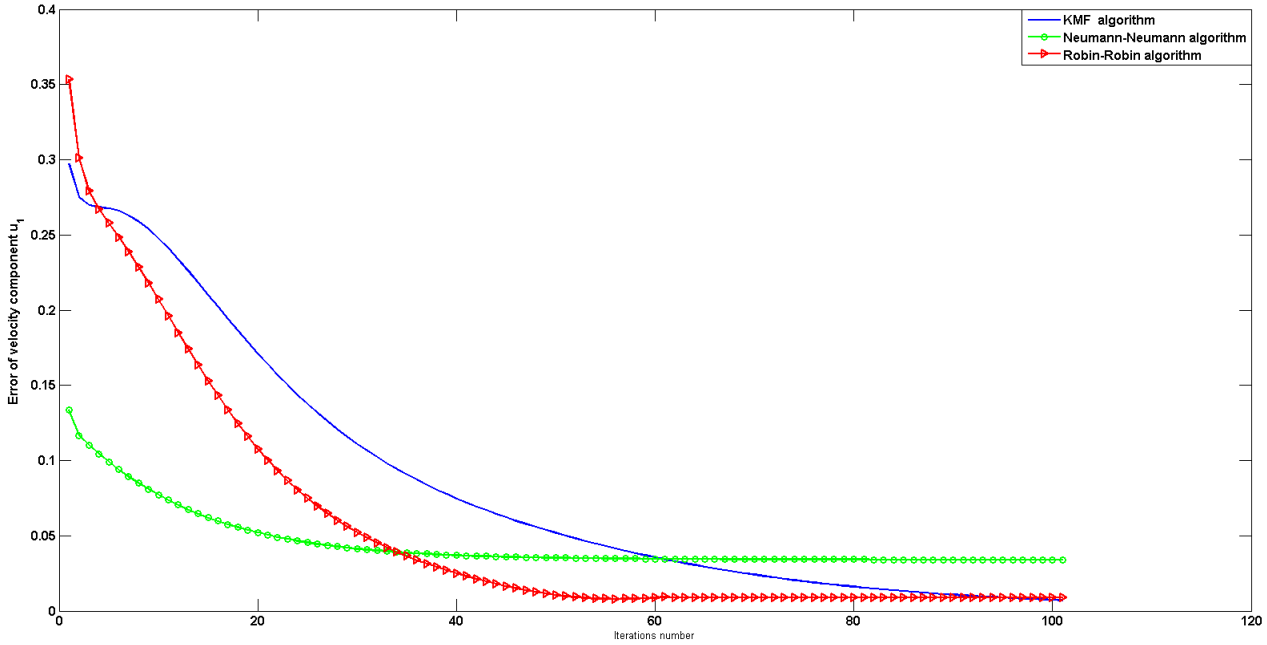


Figure 12: The variation of the velocity component u_1 error over Γ_0 with respect to the iterations number for different algorithms.

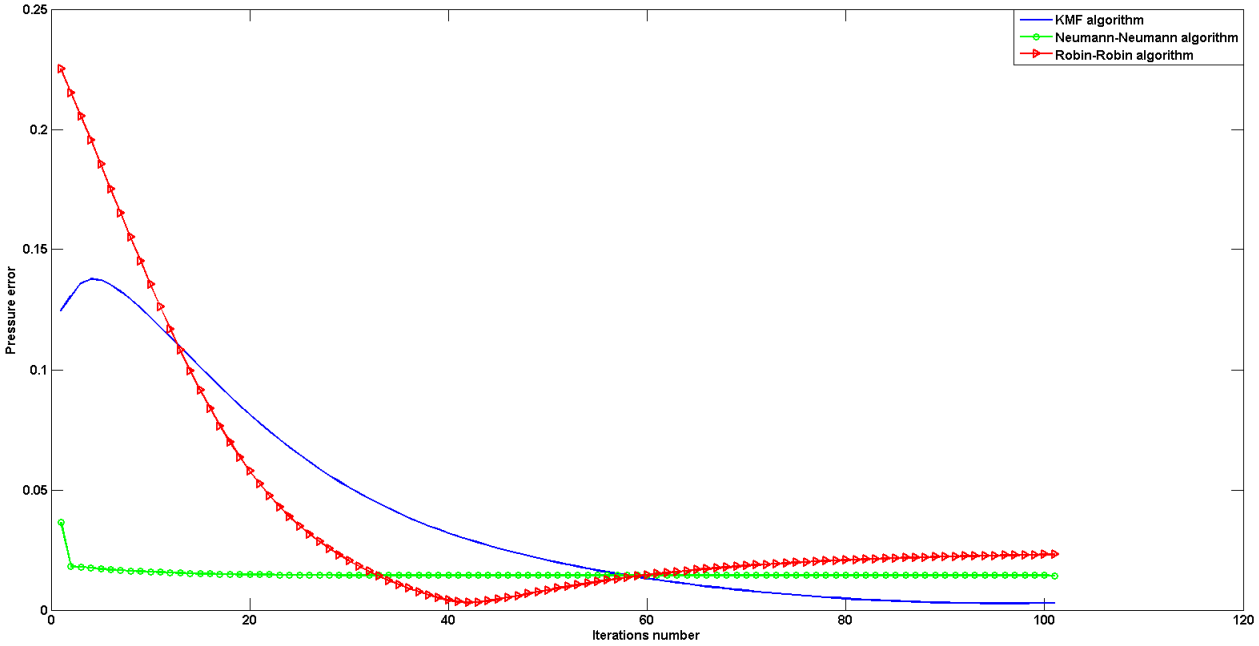


Figure 13: The variation of the pressure p error over Γ_0 with respect to the iterations number for different algorithms.

- *Neumann data over* Γ_3, *noted by* $g_{2|\Gamma_3} = (g_2^{(1)}, g_2^{(2)})$, *is given by :*

$$\begin{cases} g_2^{(1)}(x,y) = -\mu\cos(x)\cos(y), \\ g_2^{(2)}(x,y) = -\mu\sin(x)\sin(y) - 2\pi(cos(2\pi y) - cos(2\pi x)). \end{cases} \tag{15}$$

- *The source term over* Ω *noted by* $f = (f_1, f_2)$, *is given by :*

$$\begin{cases} f_1(x,y) = -2\mu\cos(x)\sin(y) + 4\pi^2\sin(2\pi x), \\ f_2(x,y) = 2\mu\sin(x)\cos(y) - 4\pi^2\sin(2\pi y), \end{cases} \tag{16}$$

where $\mu = 0.1$,

So, we reconstruct the velocity and pressure fields over the part of the boundary Γ_0, *by using the above data associated to the exact solution :*

$$\begin{cases} ue_1(x,y) = -\cos(x)\sin(y), \\ ue_2(x,y) = \sin(x)\cos(y), \\ pe(x,y) = 2\pi(\cos(2\pi y) - \cos(2\pi x)). \end{cases} \tag{17}$$

So, based on the data (12), (13), (14), (15) *and* (16) *and by choosing an compatible initial data, we present some numerical results to evaluate the quality of the approximation. In the figures 14, 15 and 16, we present the exact and the approximate solutions of the two component of the velocity and the pressure over* Γ_0. *We can deduce overall that a good approximation of the reconstructing solutions is obtained by using different algorithms, especially by "Robin-Robin" algorithm.*

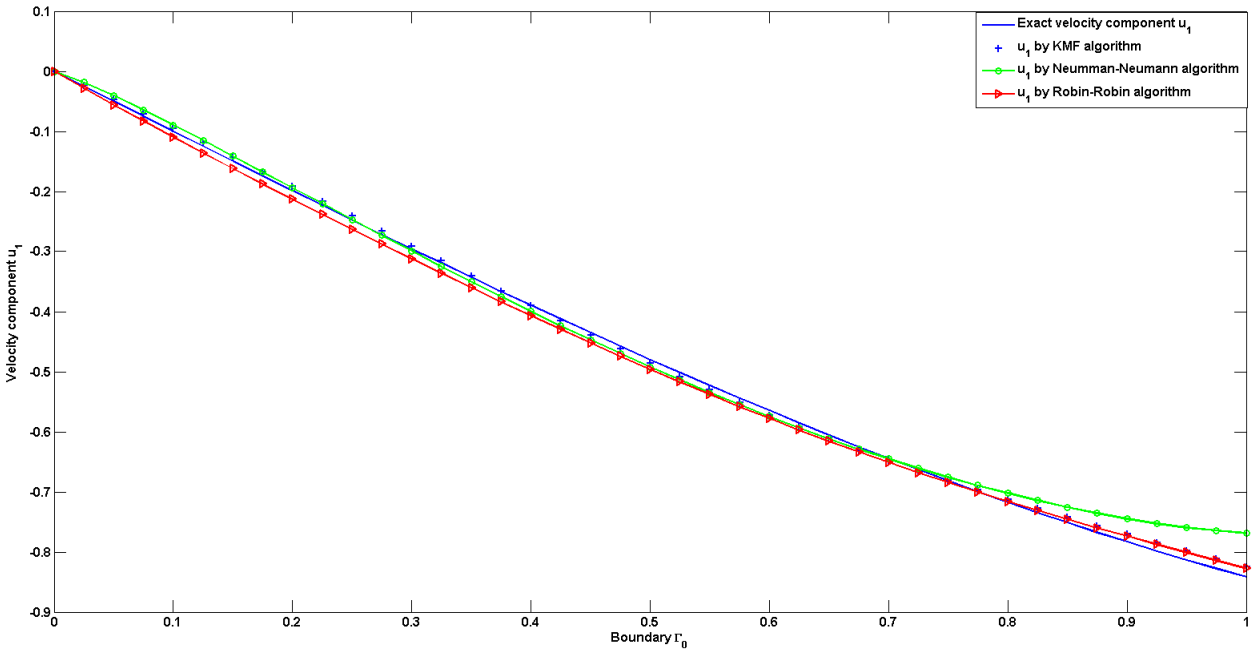


Figure 14: The exact and the approximate solutions associated to the first component of the velocity u_1 over Γ_0.

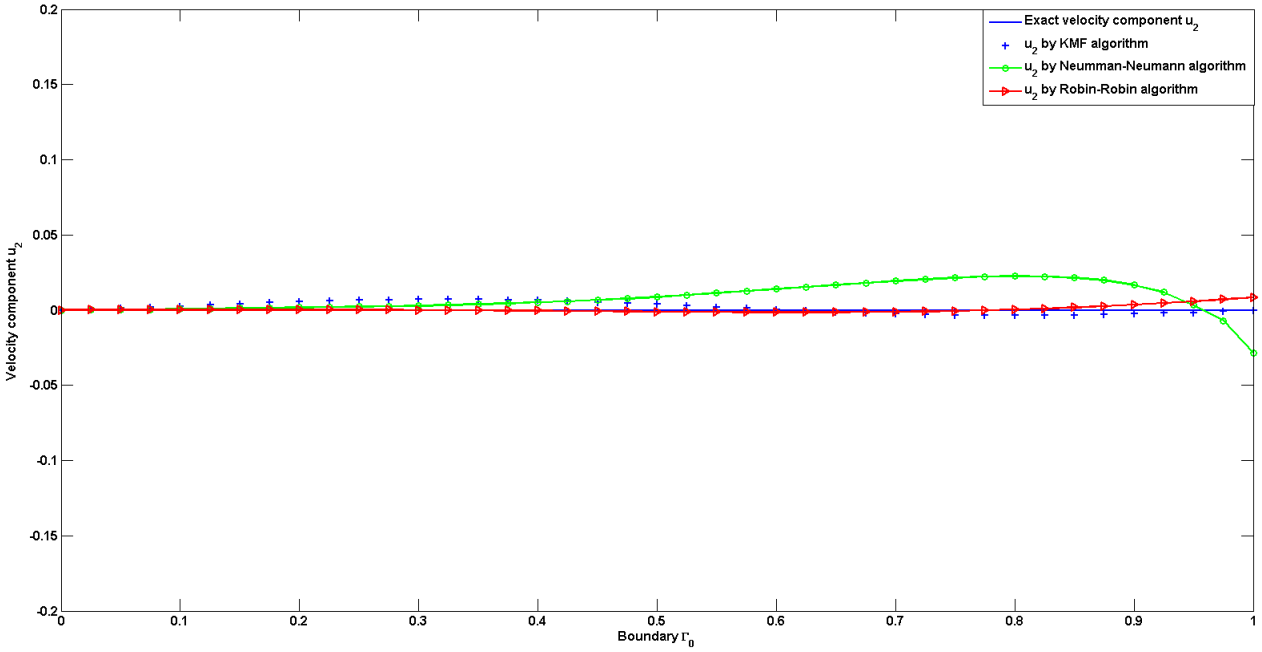


Figure 15: The exact and the approximate solutions associated to the component of the velocity u_2 over Γ_0.

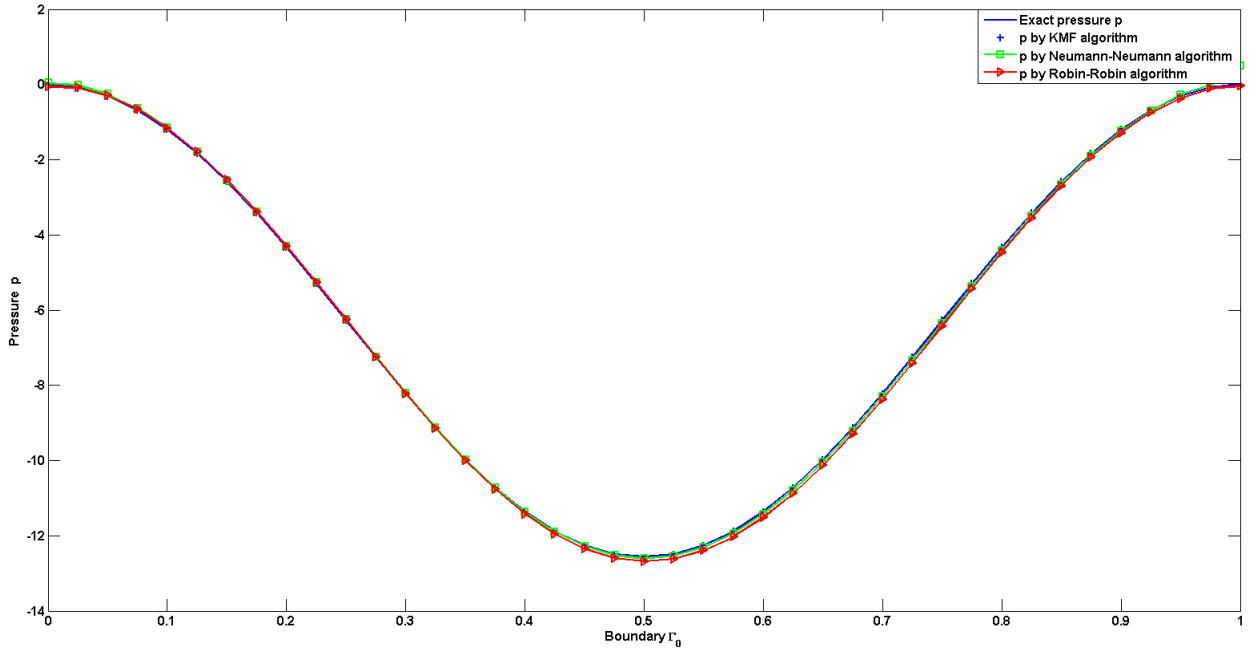


Figure 16: The exact and the approximate solutions associated to the pressure p over Γ_0.

4.2.2 Exemple 3

Now, we are considering a domaine Ω *in the form of a part of ring (Figure 17), which is the boundary is* $\partial\Omega = \Gamma_0 \cup \Gamma_{01} \cup \Gamma_{02} \cup \Gamma_{03}$ *such that*

$$\Gamma_{01} = \{(x,y) \in \mathbb{R}^2 / x = t\ \cos(\frac{\pi}{4}); \quad y = t\ \sin(\frac{\pi}{4}) \quad ; \quad t \in [r_2, r_1]\}$$

$$\Gamma_{02} = \{(x,y) \in \mathbb{R}^2 / x = r_1 \cos(\theta); \quad y = r_1 \sin(\theta) \quad ; \quad \theta \in [\frac{\pi}{4}, \frac{3\pi}{4}]\}$$

$$\Gamma_{03} = \{(x,y) \in \mathbb{R}^2 / x = t \cos(\frac{3\pi}{4}); \quad y = t \sin(\frac{3\pi}{4}) \quad ; \quad t \in [r_2, r_1]\}$$

$$\Gamma_{0} = \{(x,y) \in \mathbb{R}^2 / x = r_2 \cos(\theta); \quad y = r_2 \sin(\theta) \quad ; \quad \theta \in [\frac{\pi}{4}, \frac{3\pi}{4}]\}$$

where $r_2 = 0.4$ *et* $r_1 = 0.8$

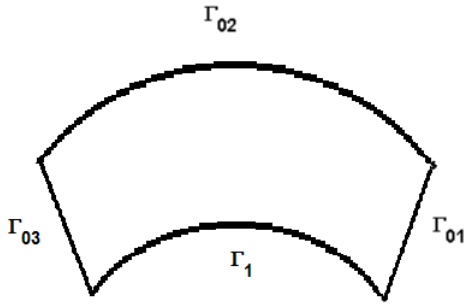


Figure 17: Domain Ω.

In this example, we propose the the reconstruction of a polynomial solution of the inverse Cauchy problem (2) associated to the following data:

- *The Dirichlet data over* Γ_{01}, *noted by* $h_{|\Gamma_{01}} = (h_{01}^{(1)}, h_{01}^{(2)})$, *is given by :*

$$\begin{cases} h_{01}^{(1)}(x,y) = 4y^3 - x^2, \\ h_{01}^{(2)}(x,y) = 4x^3 + 2xy - 1. \end{cases} \tag{18}$$

- *The Dirichlet and Neumann data over* Γ_{02}, *noted respectively by* $h_{|\Gamma_{02}} = (h_{02}^{(1)}, h_{02}^{(2)})$ *and* $g_{|\Gamma_{02}} = (g_{02}^{(1)}, g_{02}^{(2)})$, *are given by :*

$$\begin{cases} h_{02}^{(1)}(x,y) = 4y^3 - x^2, \\ h_{02}^{(2)}(x,y) = 4x^3 + 2xy - 1, \end{cases} \tag{19}$$

$$\begin{cases} g_{02}^{(1)}(x,y) = \mu((-2x) * N.x + (12y^2) * N.y) - (24xy - 12x) * N.x, \\ g_{02}^{(2)}(x,y) = \mu * ((12x^2 + 2y) * N.x + (2x) * N.y) - (24xy - 12x) * N.y. \end{cases} \tag{20}$$

- *The Neumann data over* Γ_{03}, *noted by* $g_{|\Gamma_{03}} = (g_{03}^{(1)}, g_{03}^{(2)})$, *is given by :*

$$\begin{cases} g_{03}^{(1)}(x,y) = \mu((-2x) * N.x + (12y^2) * N.y) - (24xy - 12x) * N.x, \\ g_{03}^{(2)}(x,y) = \mu((12x^2 + 2y) * N.x + (2x) * N.y) - (24xy - 12x) * N.y. \end{cases} \tag{21}$$

- *The source term over* Ω *noted by* $f = (f_1, f_2)$, *is given by :*

$$\begin{cases} f_1(x,y) = -\mu(24y - 2) + 24y - 12, \\ f_2(x,y) = -24\mu x + 24x, \end{cases} \tag{22}$$

where $\mu = 0.01$,

In the same way, we reconstruct the velocity and the pressure over the part of the boundary Γ_0*, by using the above data associated to the follwing exact solution :*

$$\begin{cases} ue_1(x,y) = 4y^3 - x^2, \\ ue_2(x,y) = 4x^3 + 2xy - 1, \\ pe(x,y) = 24xy - 12x. \end{cases} \tag{23}$$

So, based on the data (18), (19), (20), (21) *and* (22) *and by choosing a compatible initial data, we illustrate in the figures 18, 19 and 20, the exact and the approximate solutions of the two components of the velocity and pressure over* Γ_0*. The obtained numerical results, by different algorithms show again the accuracy of the reconstructing solution, in particular by "Robin-Robin" algorithm. From different numerical*

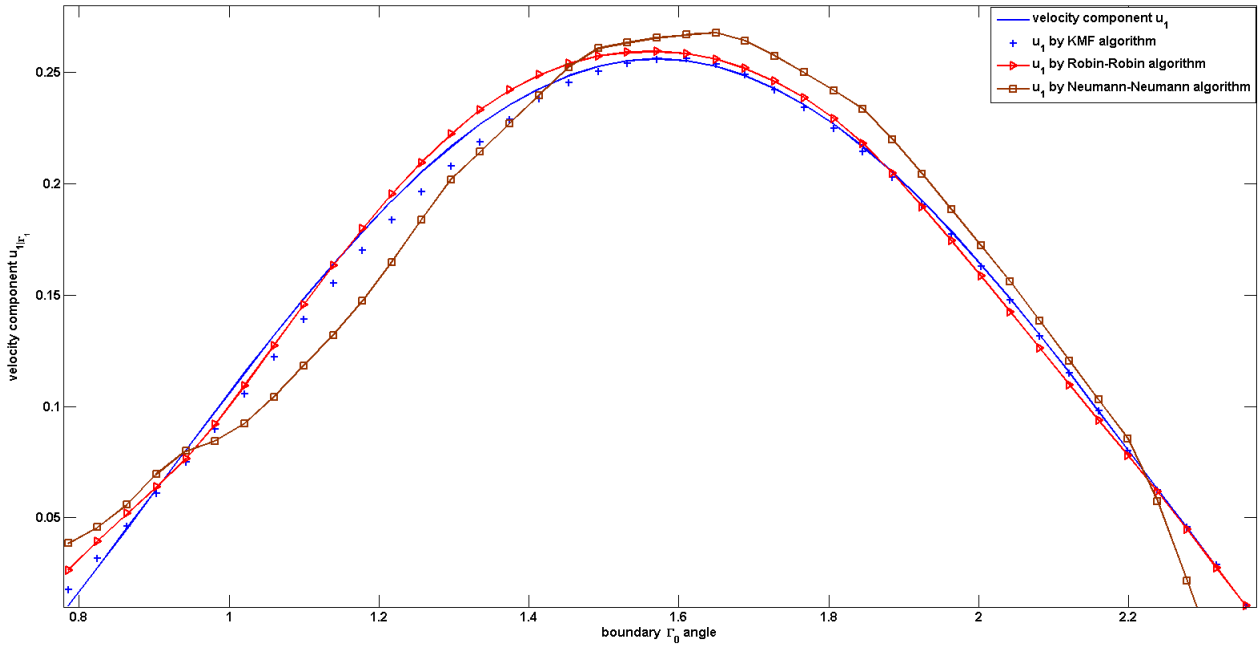


Figure 18: The exact and the approximate solution associated to the first component of the velocity u_1 over Γ_0.

tests presented before, the comparison between different algorithms implemented leads to conclude the performance of the "Robin-Robin" algorithm in term of the accuracy of the solution and fastness compared to the others. So, we opt for this algorithm in the simulation of the inverse Cauchy problem modelling the air flow in the lungs.

5 Numerical simulation of Cauchy's inverse problem modeling the air flow in the bronchial tree

In this section, we are interested in the numerical simulation of the inverse Cauchy problem modelling the air flow in the lungs. this simulation was realized in a 2D domain simulating the bronchial tree. In this context, we opt for "Robin-Robin" algorithm as a way of resolution which we prove its performance compared to the other algorithms previously presented, so we show some numerical results of the approximation of the velocity and the pressure fields over the artificial boundaries of the tree, by using a finite elements method of type $\mathbb{P}_{1Bulle}/\mathbb{P}_1$*. Finally, we study the numerical stability of the solution.*

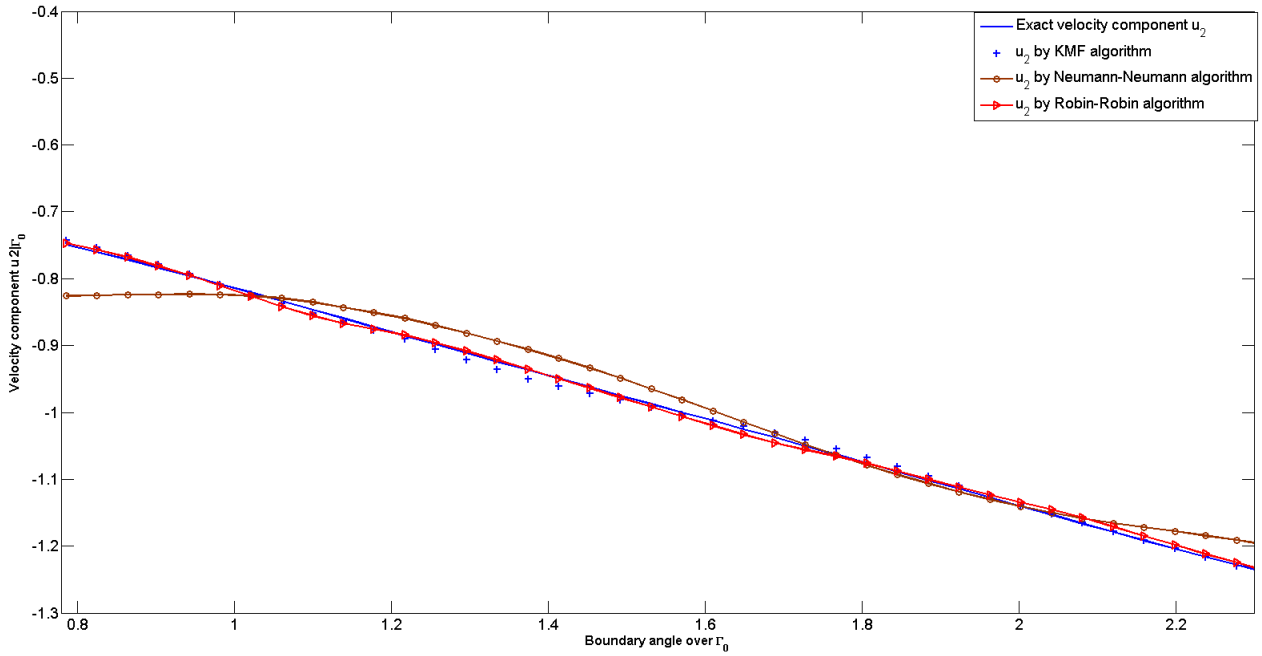


Figure 19: The exact and the approximate second component of the velocity u_2 over Γ_0.

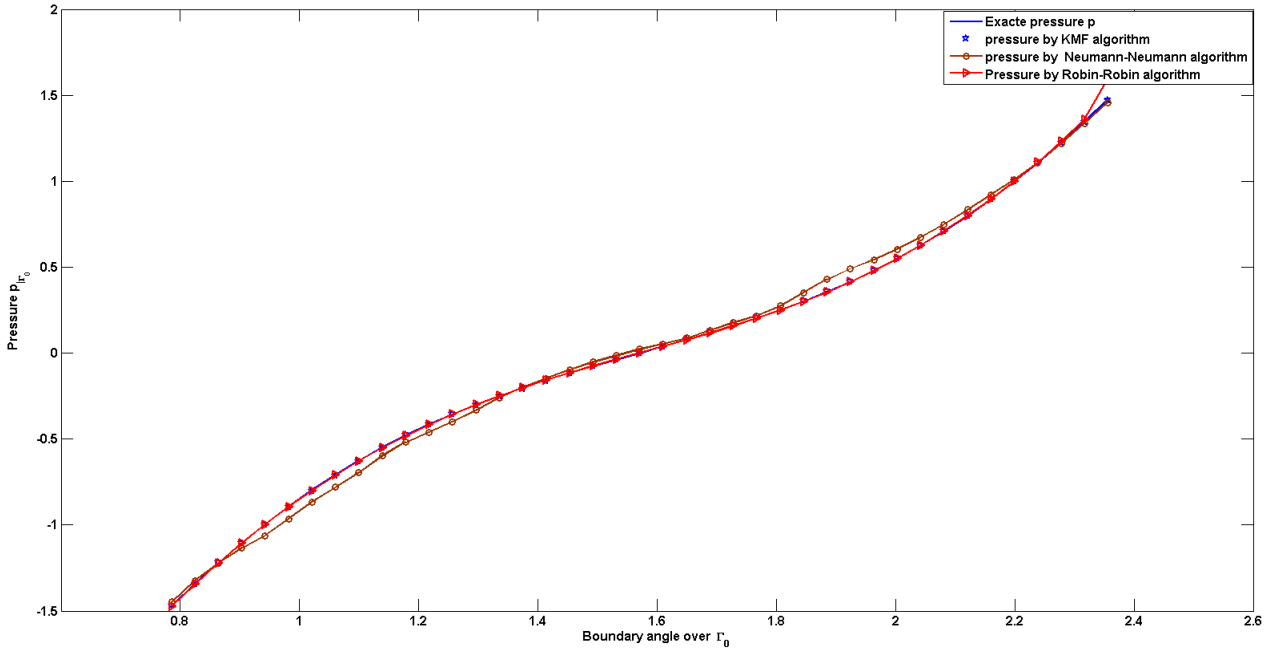


Figure 20: The exact and the approximate pressure p over Γ_0.

As we have mentioned before "Robin-Robin" algorithm has been chosen in order to propose a numerical approximation of the problem, this algorithm requires the resolution of succession of direct problems

(Stokes problem), which are approximated by the finite elements of type $\mathbb{P}_{1Bulle}/\mathbb{P}_1$. So, we need to consider a regular mesh of the bronchial tree (see the figure 21).

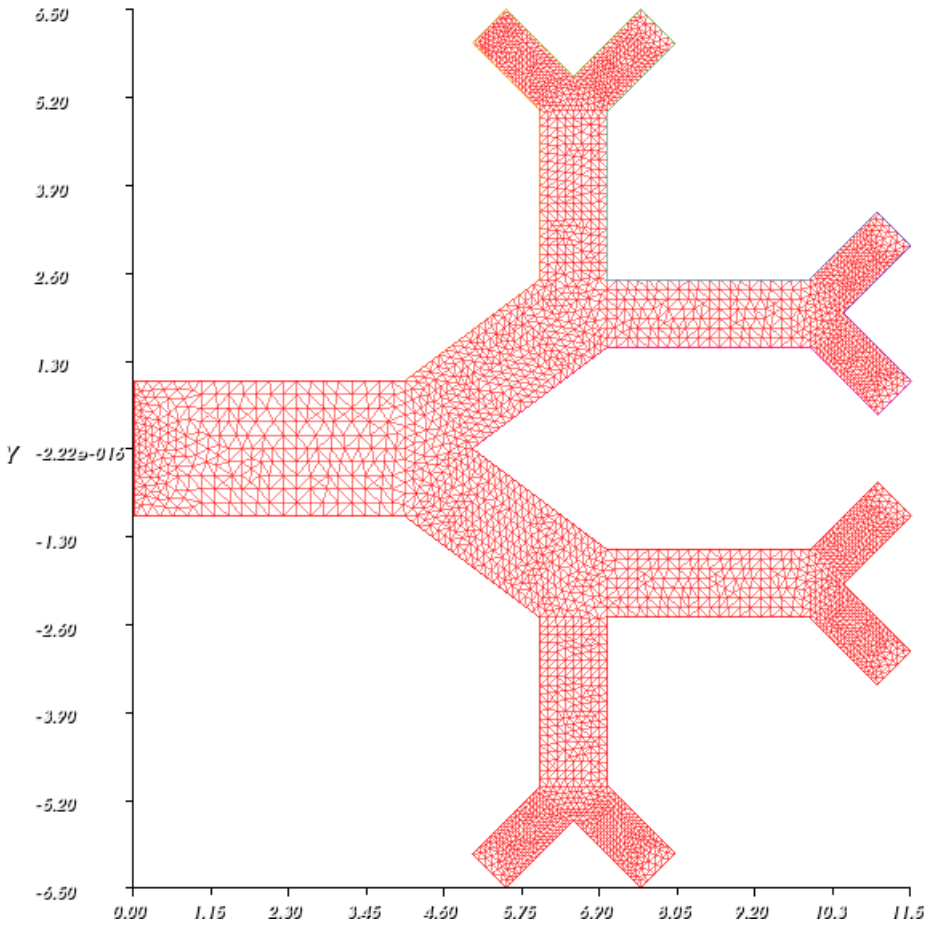


Figure 21: 2D mesh of the bronchial tree

6 Inverse problem resolution by the Robin-Robin algorithm

We recall the Robin-Robin algorithm used for a such configuration as follow

Algorithm 5. *:*

1. *Choose the two acceleration parameters γ_1 and γ_2, such that $\gamma_1 + \gamma_2 > 0$ and precision $\varepsilon > 0$.*
2. *For $k = 0$, choose an initial data $v^{(0)}$ of the velocity u over Γ_1.*
3. *Solve the following problem:*

$$\begin{cases} -2\mu \operatorname{div}(D(u^{(k)})) + \nabla p^{(k)} = f & \text{dans} \quad \Omega, \\ \operatorname{div}(u^{(k)}) = 0 & \text{dans} \quad \Omega, \\ \sigma(u^{(k)})n = g & \text{sur} \quad \Gamma_0, \\ u^{(k)} = 0 & \text{sur} \quad \Gamma_l, \\ \sigma(u^{(k)})n + \gamma_1 u^{(k)} = \sigma(u^{(k+1)})n + \gamma_1 u^{(k+1)} & \text{sur} \quad \Gamma_i, \qquad \text{pour } i = 1, ..8 \end{cases} \tag{24}$$

4. Solve the following problem:

$$\begin{cases} -2\mu\,\mathrm{div}(D(u^{(k+1)})) + \nabla p^{(k+1)} = f & \text{dans} \quad \Omega, \\ \mathrm{div}(u^{(k+1)}) = 0 & \text{dans} \quad \Omega, \\ u^{(k+1)} = h & \text{sur} \quad \Gamma_0, \\ u^{(k+1)} = 0 & \text{sur} \quad \Gamma_l, \\ \sigma(u^{(k+1)})n - \gamma_2 u^{(k+1)} = \sigma(u^{(k)})n - \gamma_2 u^{(k)} & \text{sur} \quad \Gamma_i \qquad \text{pour } i = 1,..8 \end{cases} \tag{25}$$

5. If $\|u^{(k)} - u^{(k+1)}\|_{(L^2(\Gamma_i))^d} \leq \varepsilon$, *we stop. Else* $k = k+1$, *go to the step 3.*

where $v^{(0)} = (0,0)$ *and* $\varepsilon = 10^{-2}$.

6.1 Numerical results

In the figure 22, we present the difference between the exact and the approximate solutions associated to the two components of the velocity and the pressure (u_1, u_2, p) *at the first iteration through the algorithm 5. so, we start with a big error on the inaccessible artificial boundaries* Γ_i, *for* $i = 1, ..., 8$. *Then, we present in the figures 23, 24 and 25 respectively the difference between the exact and the approximate solutions* u_1, u_2 *and* p, *over the simulating bronchial tree, at the convergence of the algorithm 5. Now, The exact and the approximate solutions associated to the first component of the velocity* u_1 *are illustrated in the figure 26, while the second component of the velocity* u_2 *is presented in the figure 27, finally the exact and the approximated pressures are illustrated in the figure 28, it is clear that we obtain a good approximation of the reconstructing solutions by the algorithm 5. Now, in order to investigate the accuracy of the solution* (u_1, u_2, p) *reconstruct over the artificial boundaries* Γ_i *for* $i = 1, ..., 8$. *We present firstly the component of the velocity* u_1 *on the boundaries* Γ_i *for* $i = 1, 2, 3, 4$ *in the figure 29, and on the boundaries* Γ_i *for* $i = 5, 6, 7, 8$, *in the figure 30. We present also the second component of the velocity* u_2 *on the boundaries* Γ_i *for* $i = 1, 2, 3, 4$ *in the figure 31, and on the rest of the boundaries in the figure 32. Finally, the pressure is also illustrated on the artificial boundaries* Γ_i *for* $i = 1, 2, 3, 4$ *in the figure 33, and on the rest of the inaccessible boundary in the figure 34. Thus we can note that the results obtained are very satisfactory, they show well the quality of approximation of the reconstructed solution on the inaccessible boundary. In the following, we are interested to numerical stability of the reconstruct solution by the algorithm 5.*

6.2 Numerical stability

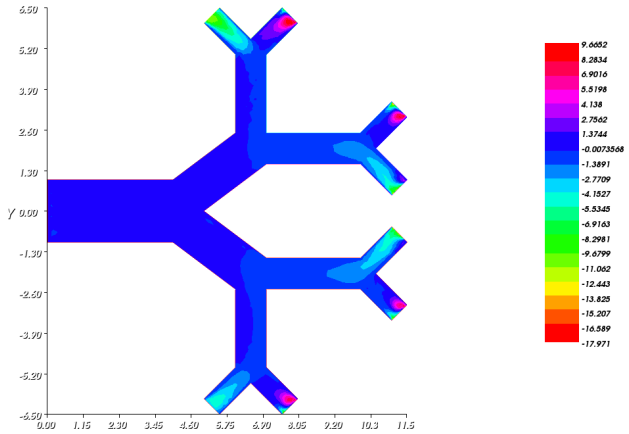
As it was already mentioned before, this class of inverse Cauchy problems are ill-posed problems in the sense of Hadamard [12]. So the stability of the solution of such problems is not an easy task.

In this section, we investigate numerically the stability of the algorithm 5. More precisely, we will study the behavior of of such algorithm in the presence of small perturbations in the data. Indeed, the Dirichlet and Neumann boundary data on Γ_0 *are perturbed to simulate measurement errors such that*

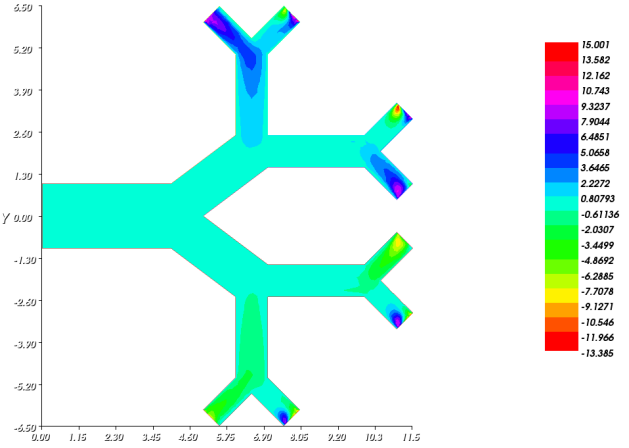
$$g^\delta = g + \delta g \ \text{ on } \quad \Gamma_{02} \qquad h^\delta = h + \delta h \ \text{ on } \quad \Gamma_0,$$

where δ *is the noise level, and* $\delta g = g * \delta * (2 * rand - 1)$ *and* $\delta h = h * \delta * (2 * rand - 1)$ *are Gaussian noises with mean zero. They are generated by an appropriate function rand which returns a pseudo-random value drawn from a uniform distribution on* $[0, 1]$. *The associate perturbed inverse problem reads: find* (u^δ, p^δ) *solution of*

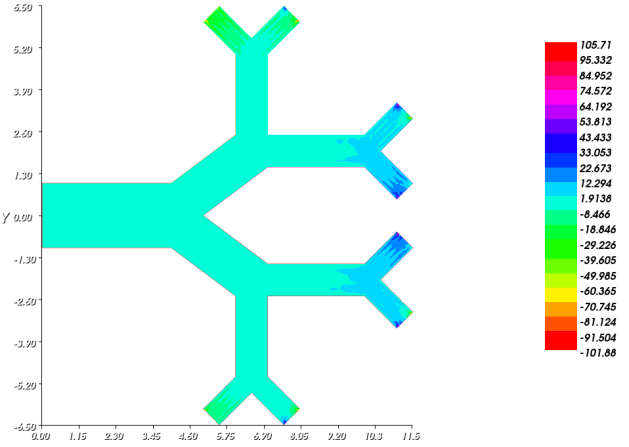
$$\begin{cases} -2\mu\,\mathrm{div}(D(u^\delta)) + \nabla p^\delta = f & \text{dans} \quad \Omega, \\ \mathrm{div}(u^\delta) = 0 & \text{dans} \quad \Omega, \\ u = h^\delta & \text{sur} \quad \Gamma_0, \\ \sigma(u^\delta)n = g^\delta & \text{sur} \quad \Gamma_0, \\ u^\delta = 0 & \text{sur} \quad \Gamma_l, \end{cases} \tag{26}$$



(a) 1st component of the velocity u_1



(b) 2nd component of the velocity u_2



(c) The pressure p

Figure 22: The difference between the exact and the approximate velocity and the pressure (u_1, u_2, p) at the first iteration by the algorithm 5

.

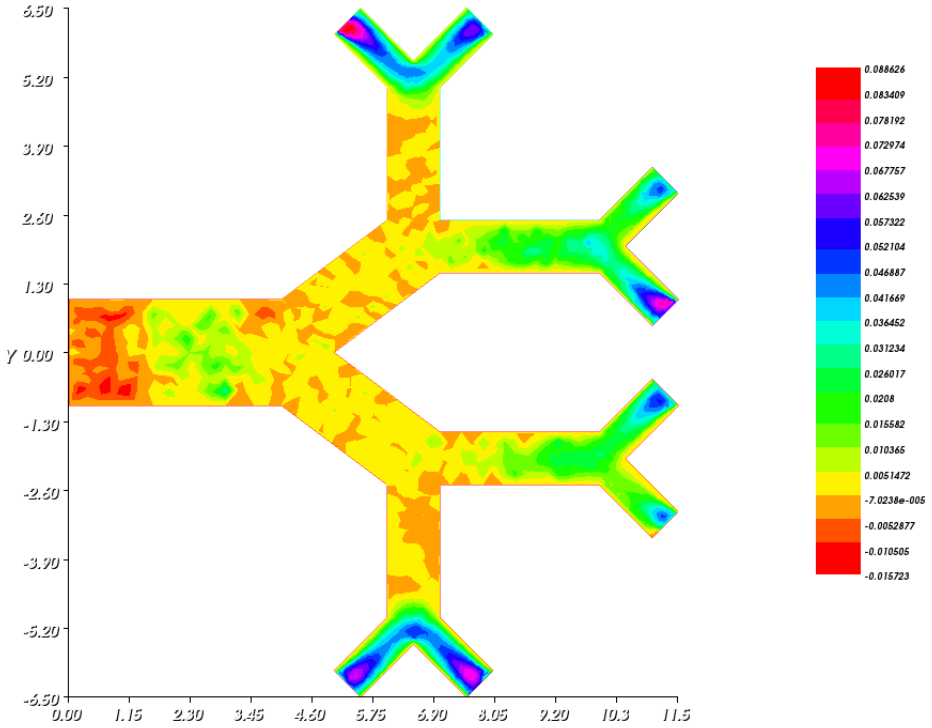


Figure 23: the difference between the exact and the approximate velocities u_1 over the simulating bronchial tree at the convergence of the algorithm 5

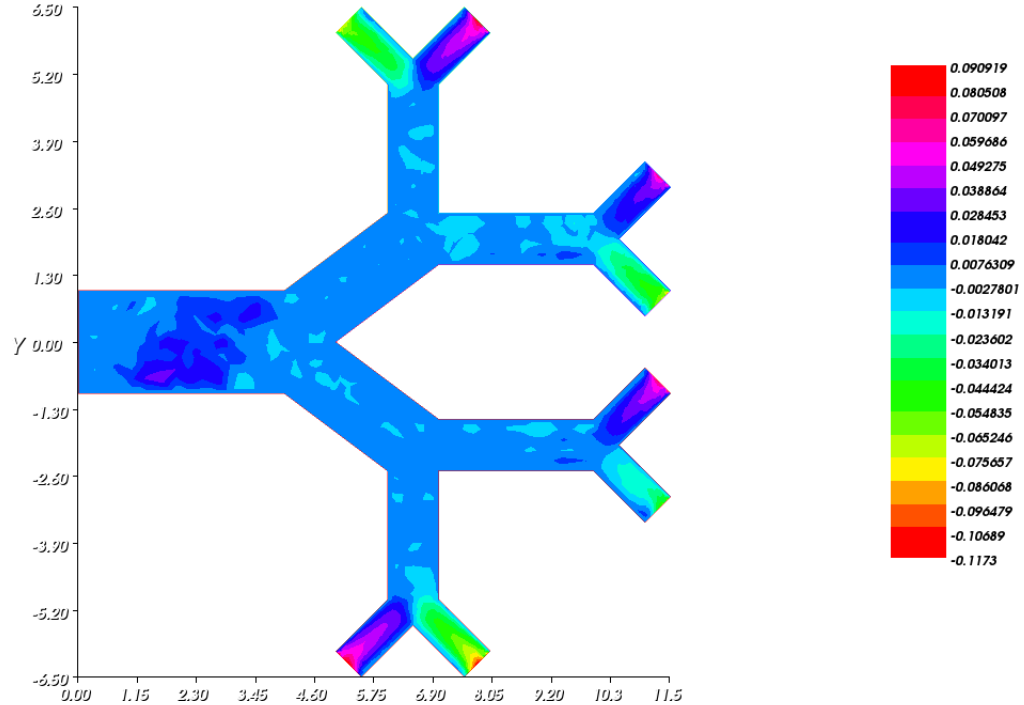


Figure 24: the difference between the exact and the approximate velocities u_2 over the simulating bronchial tree at the convergence of the algorithm 5

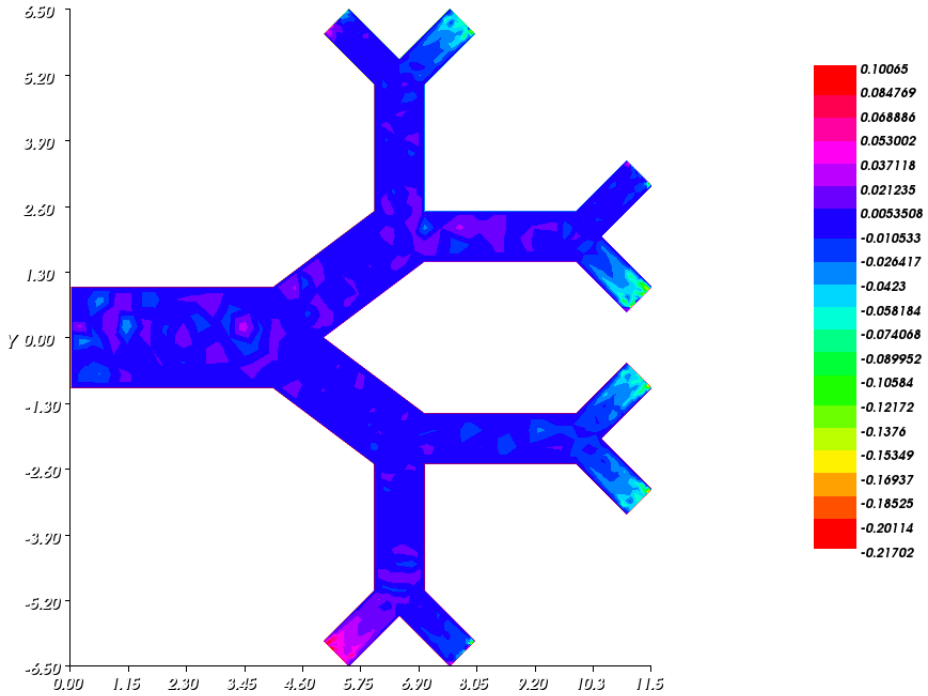


Figure 25: the difference between the exact and the approximate pressures p over the simulating bronchial tree at the convergence of the algorithm 5

In order to show that our proposed methods are numerically stable, we introduce a uniformly distributed noise with different values of the level $\delta = 0.001, 0.01, 0.05$ *and* 0.1.

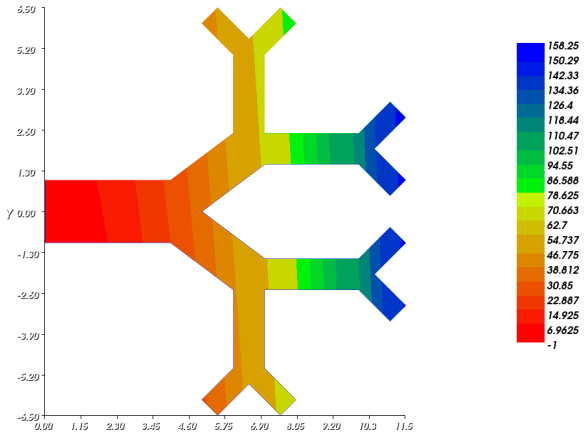
First, we show in the figures 35, 36 and 37 the difference between the exact, the approximate and perturbed solutions for a noise level $\delta = 0.1\%$ *associated to the velocity components and the pressure. In the figures 38, 39 and 40, we illustrate also the difference between the exact and the perturbed solutions for a noise level* $\delta = 10\%$. *These obtained results show that a small perturbations do not affect the error between the exact and the approximate solutions compared to the results obtained without noise.*

Now, let us examine the stability of the solution on the inaccessible artificial boundaries, we present in the figures 41, 42 and 43 the exact, approximate and the perturbed solutions by different noise levels $0.1\%, 1\%, 5\%$ *et* 10% *for the first component of the velocity* u_1 *on the boundaries* Γ_i, *for* $i = 1, ..., 8$. *Next, regarding the stability of the second component of the velocity* u_2 *on the inaccessible boundaries, for this we present prsentons the exact, approximate and the perturbed solutions associated to the same noise levels on* Γ_1 *and* Γ_2 *in the figure 44, on* Γ_3 *and* Γ_4 *in the figure 45. Finally, the exact, approximate and the perturbed pressure for the noise levels* 1%, 5% *and* 10% *on the boundaries* Γ_1 *and* Γ_2 *are illustrated in the figure 46, on* Γ_3 *and* Γ_4 *in the figure 47, and on* Γ_i, *for* $i = 5, 6, 7, 8$ *in the figure 48.*

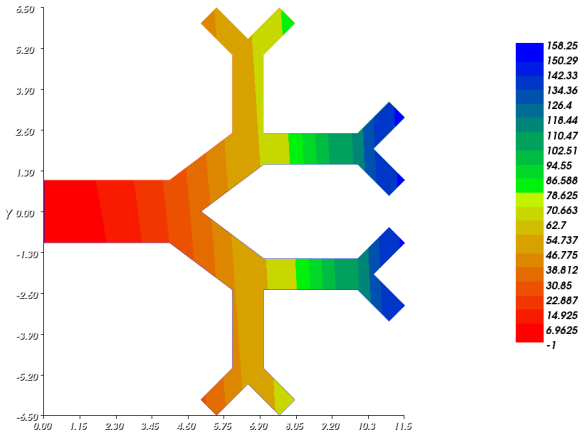
As we can see from these figures, the obtained solution (u_h^δ, p_h^δ) *with* $\delta = 0.01$ *and the obtained one without noise are almost identical, while the curve the exact solution is close to those of the solutions with* $\delta = 0.01$, 0.05 *and* 0.10. *Moreover, the error satisfy the following stability estimate*

$$\begin{aligned} \|u_{exacte} - u_{approche}\|_{(L^2(\Gamma_i))^d} &\leq \delta \ pour \qquad i = 1, ..., 8 \\ \|p_{exacte} - p_{approche}\|_{(L^2(\Gamma_i))^d} &\leq \delta \ pour \qquad i = 1, ..., 8 \end{aligned} \tag{27}$$

which shows that the algorithm 5 is stable. This confirms that the proposed algorithm can be considered as a regularizing one.

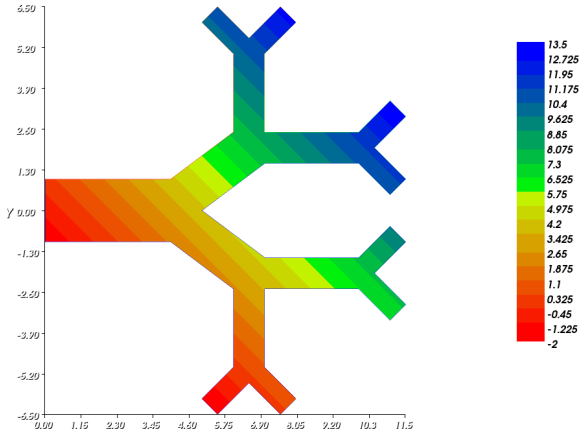


(a) The exact u_1

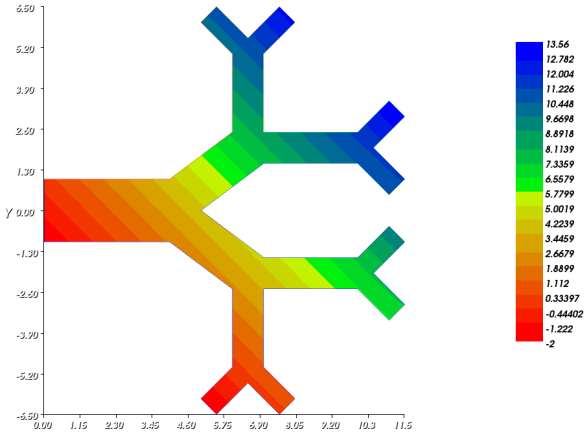


(b) The approximate u_1

Figure 26: The exact and the approximate component of the velocity u_1

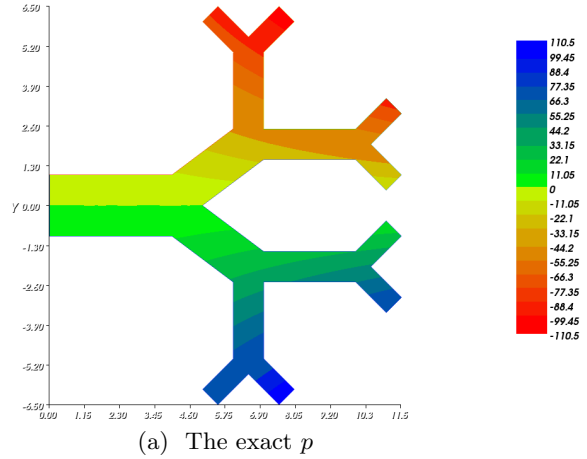


(a) The exact u_2

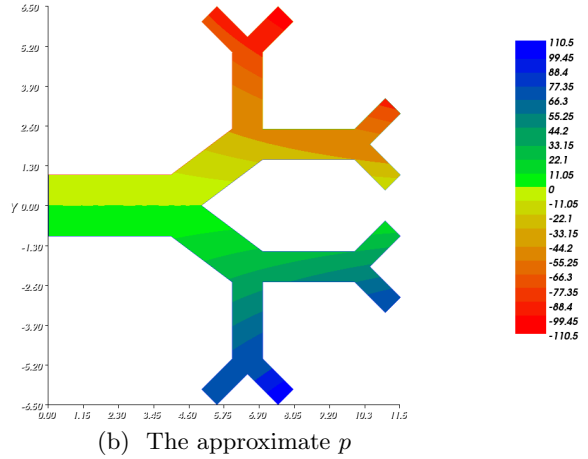


(b) The approximate u_2

Figure 27: The exact and the approximate component of the velocity u_2



(a) The exact p



(b) The approximate p

Figure 28: The exact and the approximate pressure p

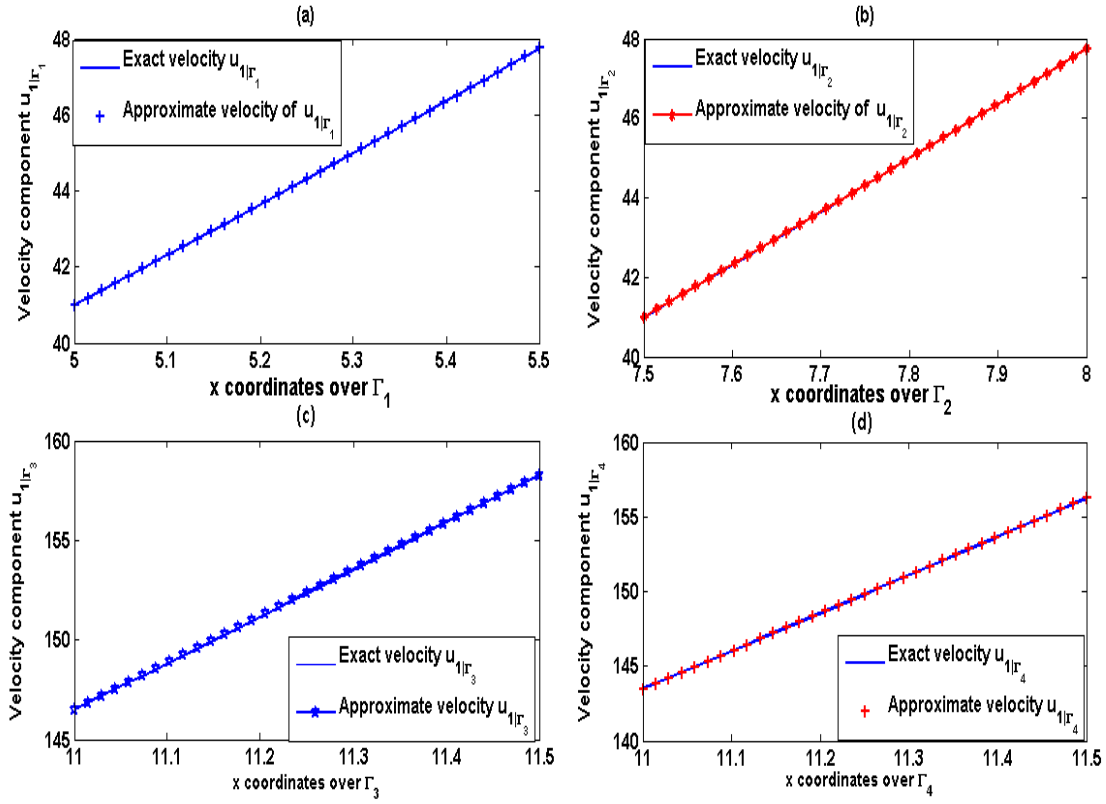


Figure 29: The component of the velocity u_1 on the artificial boundaries Γ_i for $i = 1, 2, 3, 4$.

7 Conclusion

In this paper, we are interested by a numerical simulation of an inverse Cauchy problem governed by Stokes equation modelling the air flow in the bronchial tree. The approach that we are opting for is based on the fact that the suggested Dirichlet-Neumann domain decomposition algorithm is exactly the KMF algorithm widely used in the literature. This established relationship allowed us to develop a new stopping criterion which is more responsive and reflects the real behavior of the error on the non accessible part of the boundary. This encouraged us to expand the scope of the study to apply other methods of domain decomposition for solving this inverse Cauchy problem. The obtained numerical results by the $P_{1Bubble}/P_1$ *finite element approximations shows that the proposed methods accurately reflects the behavior of the error on non accessible the boundary. An other advantage of these approaches is that they give the opportunity to reduce the execution time. Thus, we have shown the effectiveness of these methods in terms of quality (accuracy of approximation) and quantitative (computation cost). Finally, we realize the numerical simulation of the inverse problem modelling the air flow in a two dimension domain simulating the bronchial tree, by using the Robin-Robin algorithm, in order to recover the solution on the artificial boundaries. The numerical stability was discussed.*

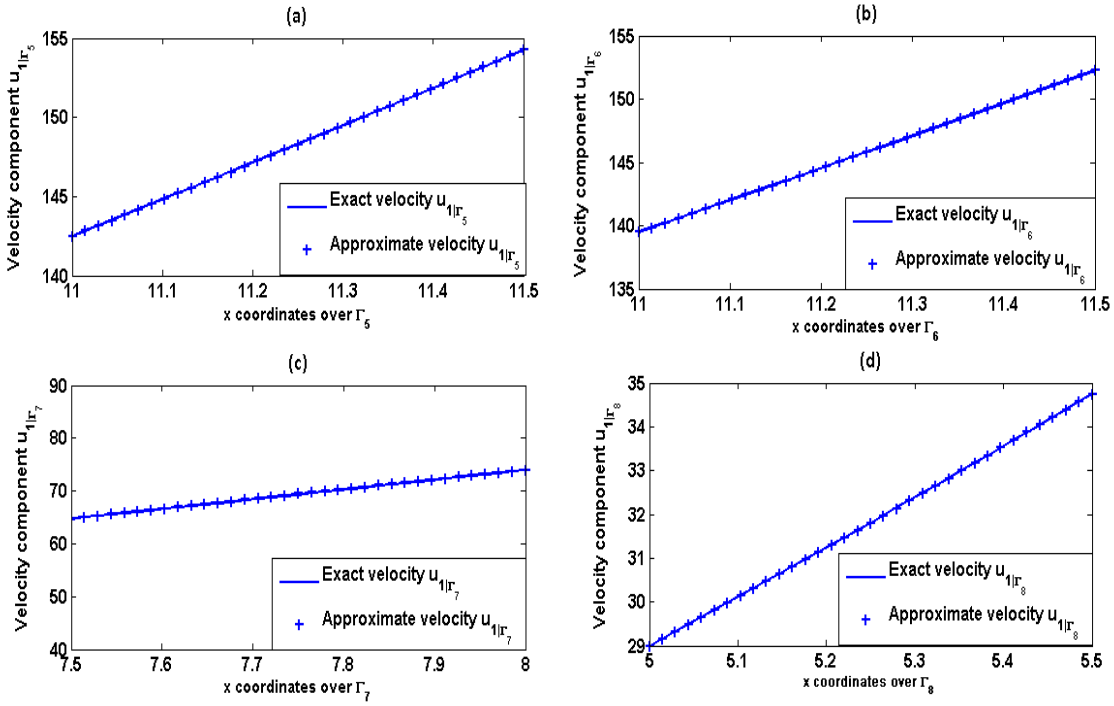


Figure 30: The component of the velocity u_1 on the artificial boundaries Γ_i for $i = 5, 6, 7, 8$.

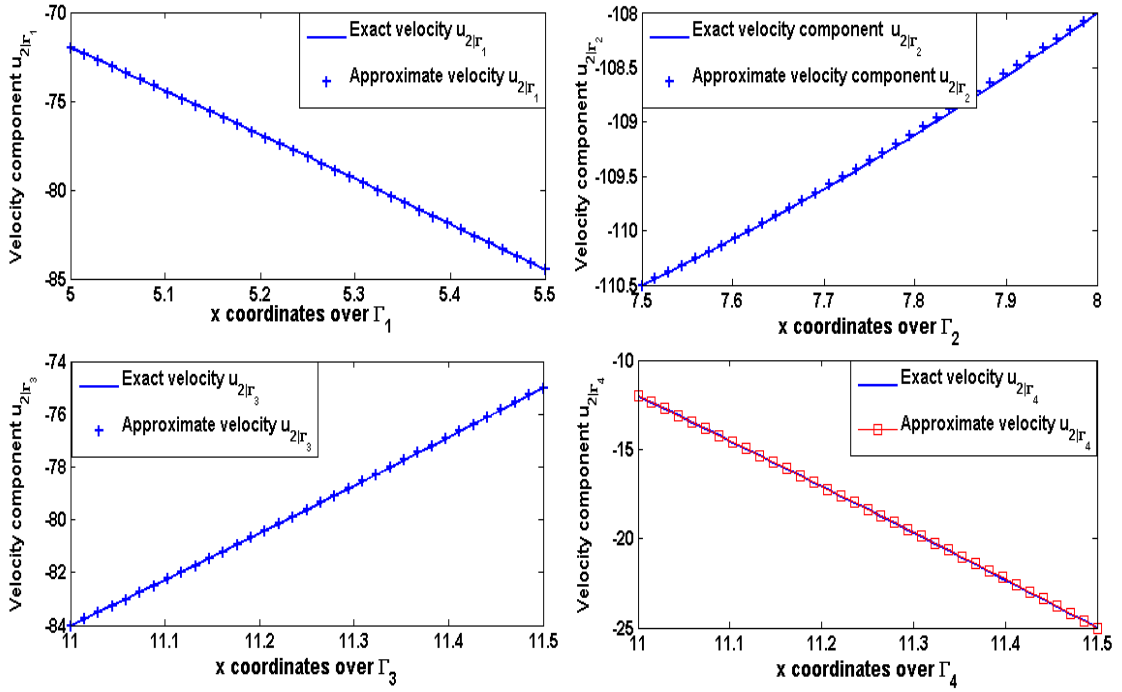


Figure 31: The component of the velocity u_2 on the artificial boundaries Γ_i for $i = 1, 2, 3, 4$.

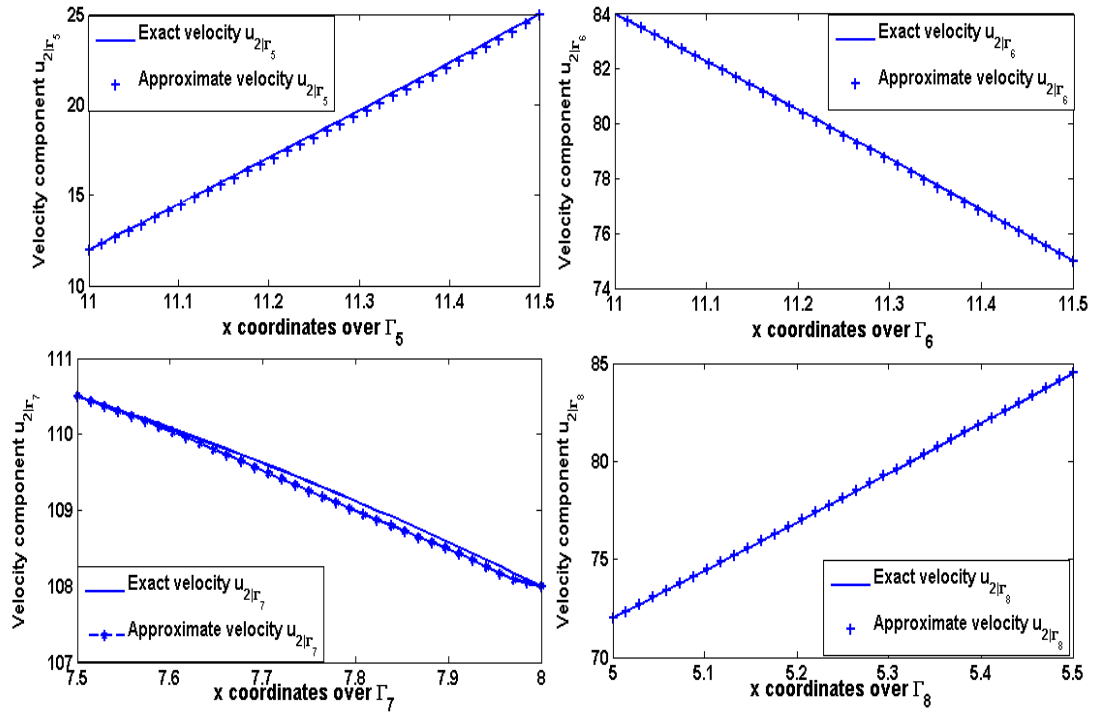


Figure 32: The component of the velocity u_2 on the artificial boundaries Γ_i for $i = 5, 6, 7, 8$.

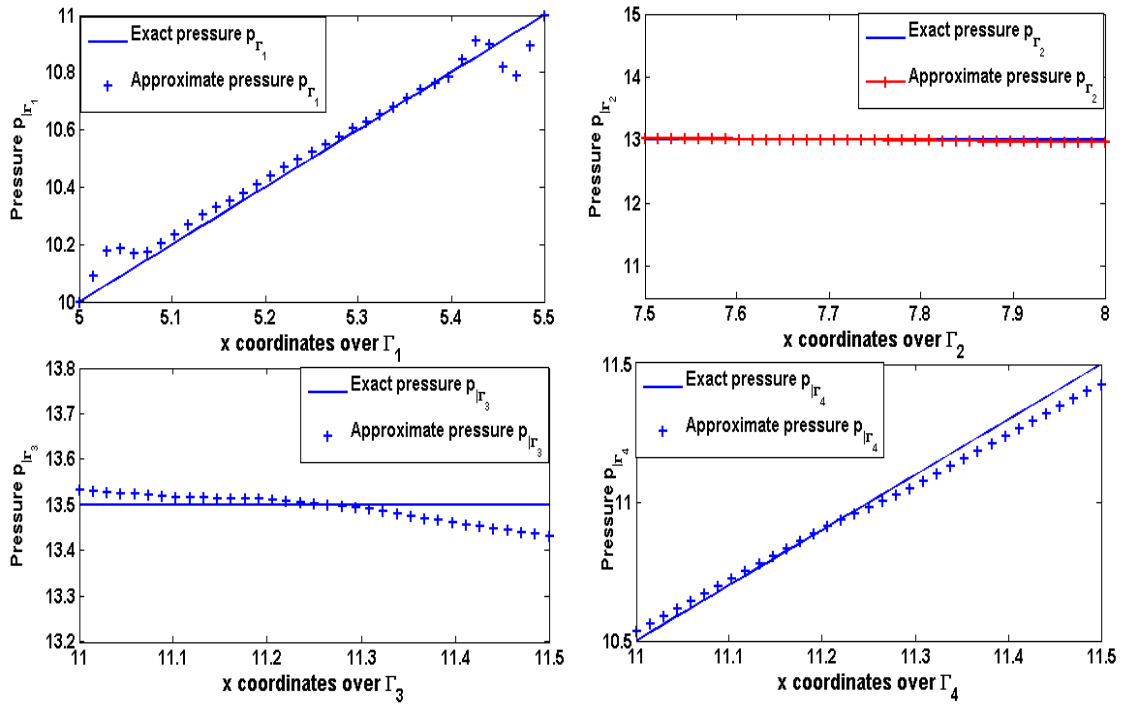


Figure 33: The pressure p on the artificial boundaries Γ_i for $i = 1, 2, 3, 4$.

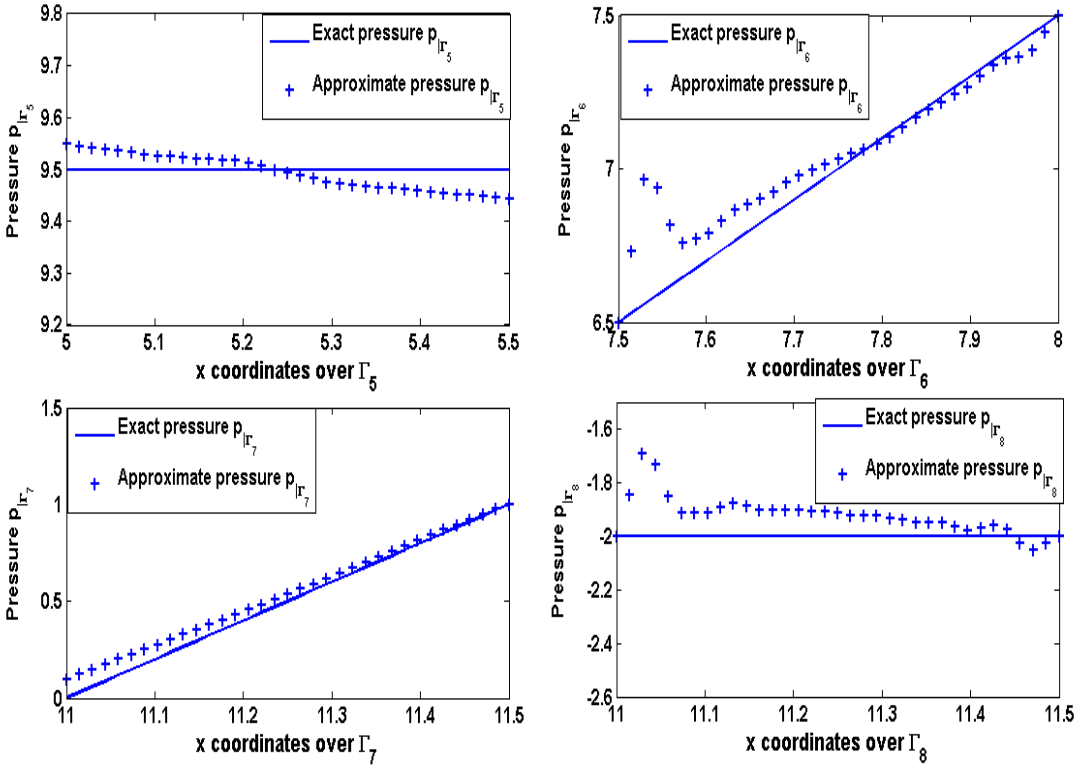


Figure 34: The pressure p on the artificial boundaries Γ_i for $i = 5, 6, 7, 8$.

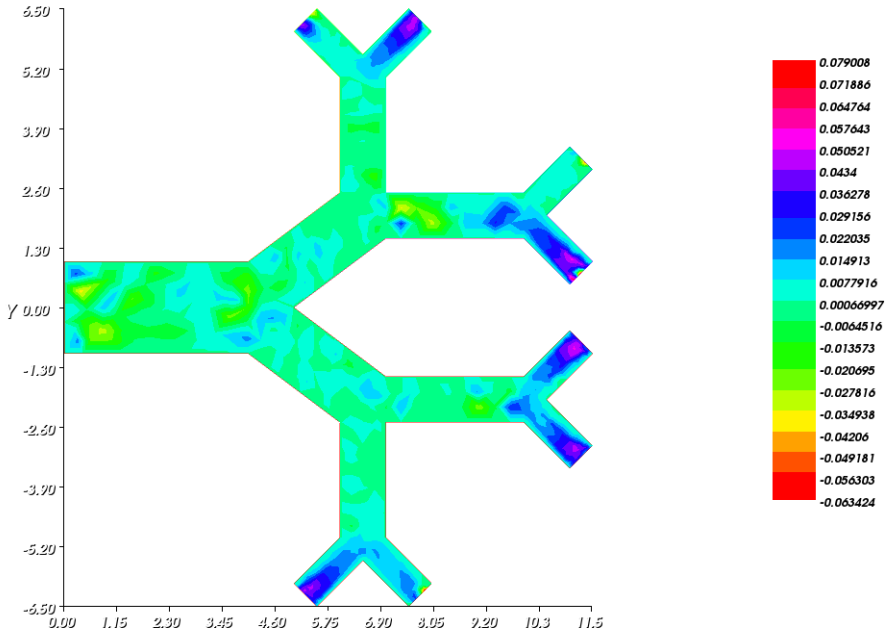


Figure 35: The difference between the exact, the approximate and the perturbed solutions for a noise level $\delta = 0.1\%$ associated to the first component of the velocity u_1.

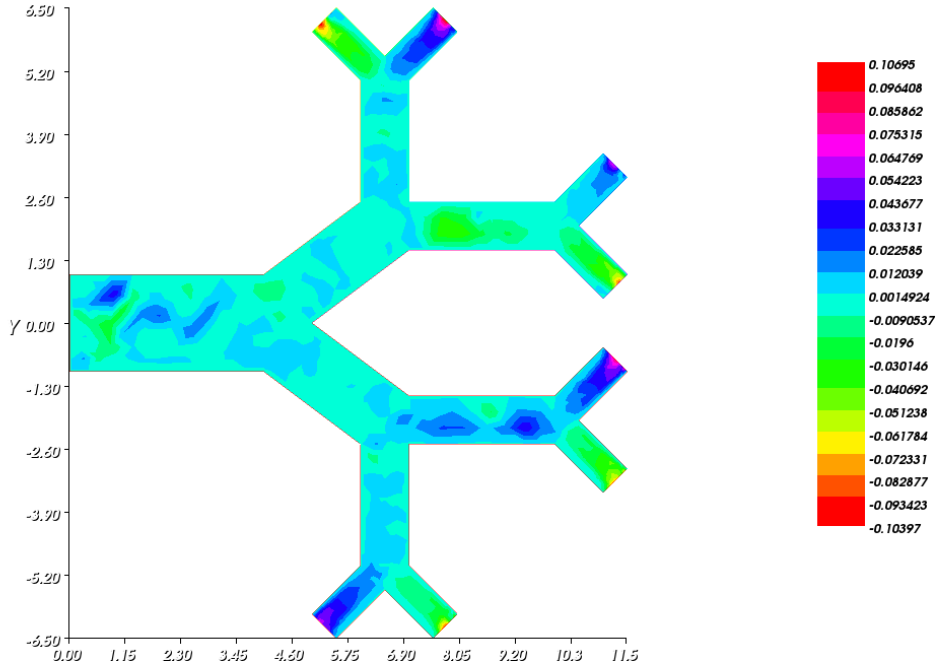


Figure 36: The difference between the exact, the approximate and the perturbed solutions for a noise level $\delta = 0.1\%$ associated to the second component of the velocity u_2.

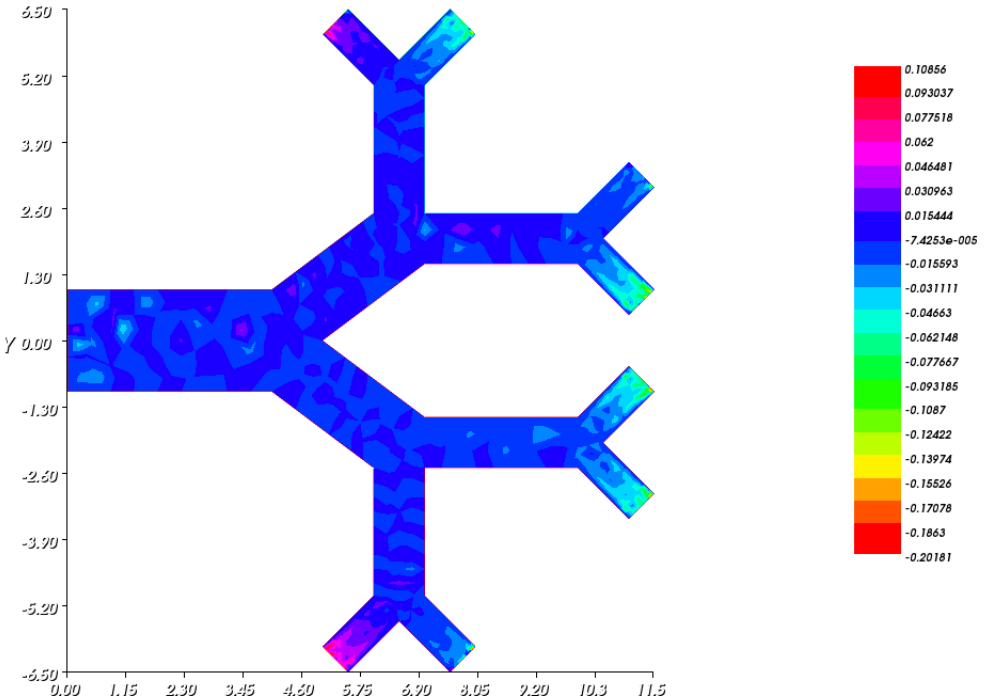


Figure 37: The difference between the exact, the approximate and the perturbed solutions for a noise level $\delta = 0.1\%$ associated to the pressure p.

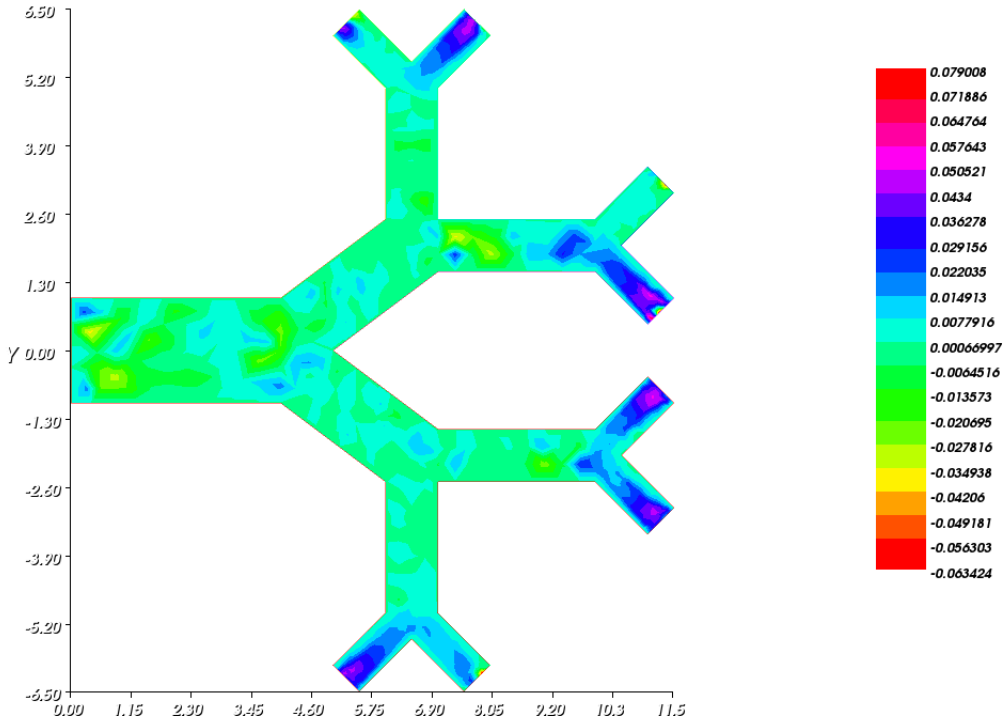


Figure 38: The difference between the exact, the approximate and the perturbed solutions for a noise level $\delta = 10\%$ associated to the first component of the velocity u_1.

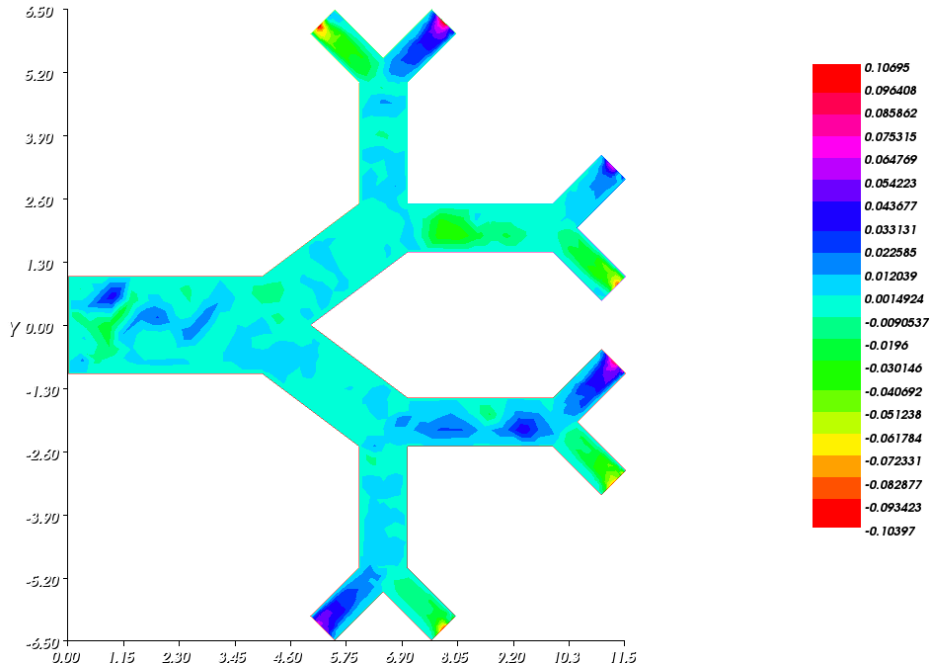


Figure 39: The difference between the exact, the approximate and the perturbed solutions for a noise level $\delta = 10\%$ associated to the second component of the velocity u_2.

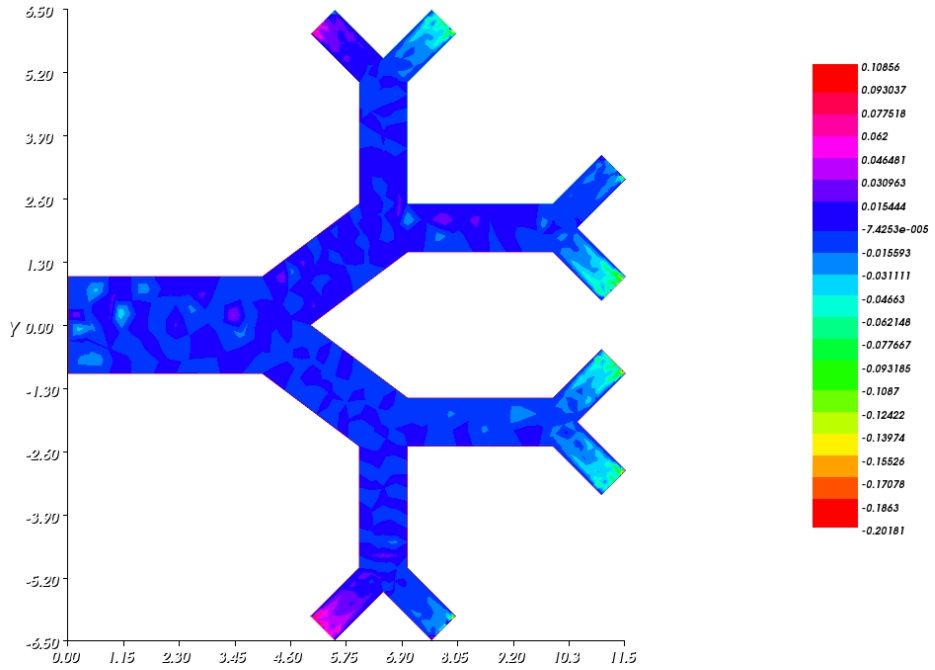


Figure 40: The difference between the exact, the approximate and the perturbed solutions for a noise level $\delta = 10\%$ associated to the pressure p.

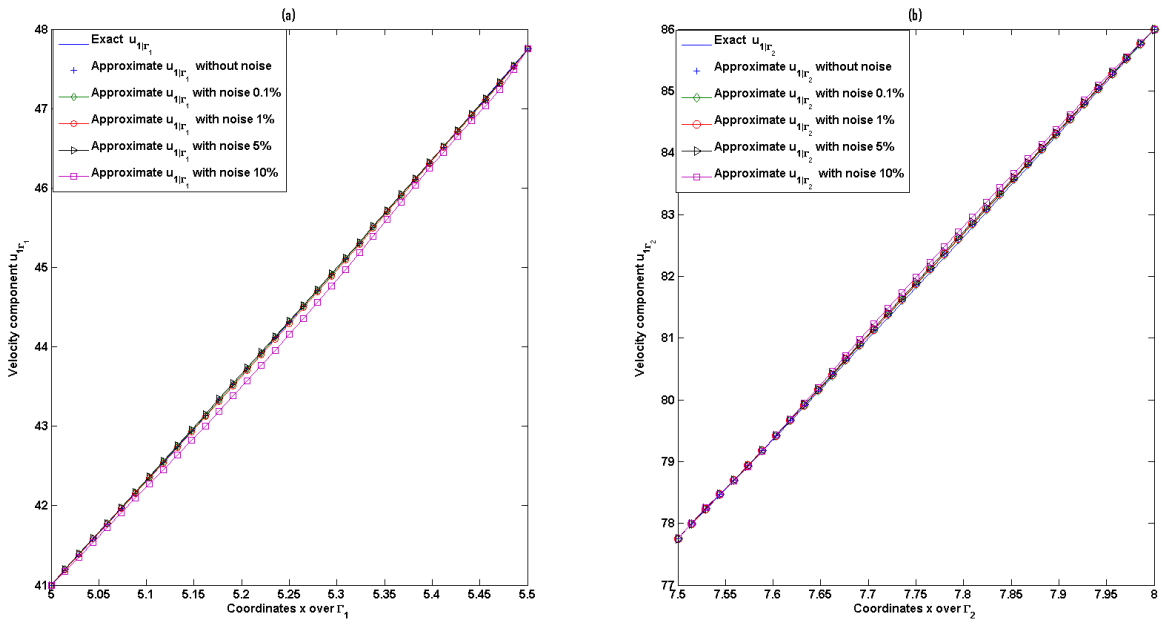


Figure 41: The exact, approximate and the perturbed solutions by different noise levels $0.1\%, 1\%, 5\%$ et 10% for the first component of the velocity u_1 on the boundaries Γ_1 and Γ_2

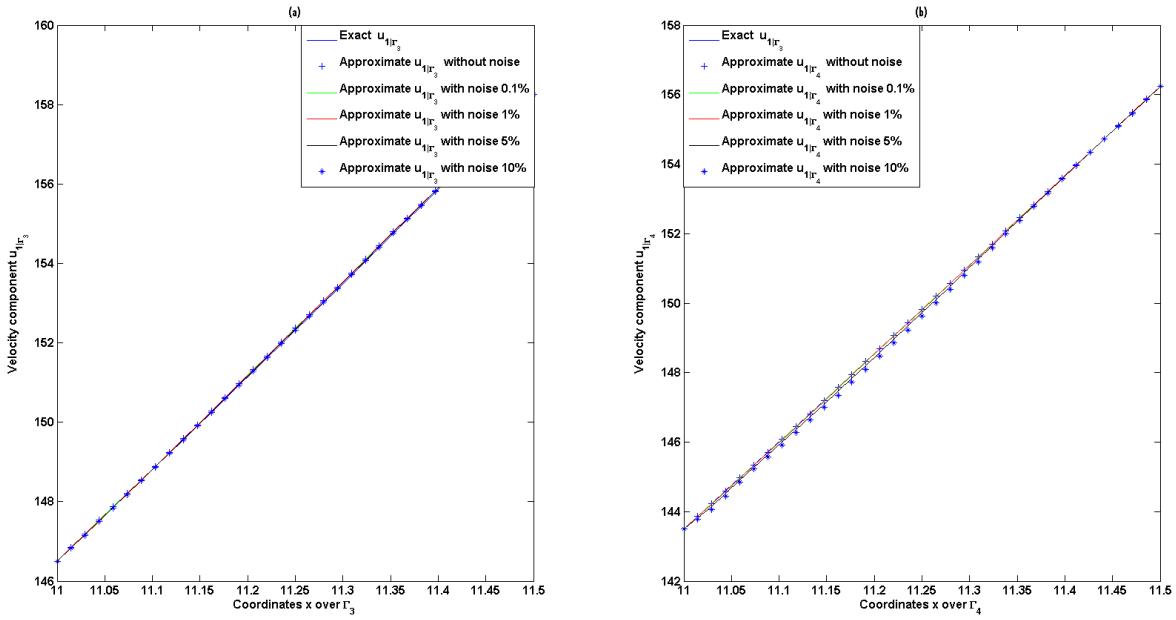


Figure 42: The exact, approximate and the perturbed solutions by different noise levels $0.1\%, 1\%, 5\%$ et 10% for the first component of the velocity u_1 on the boundaries Γ_3 and Γ_4

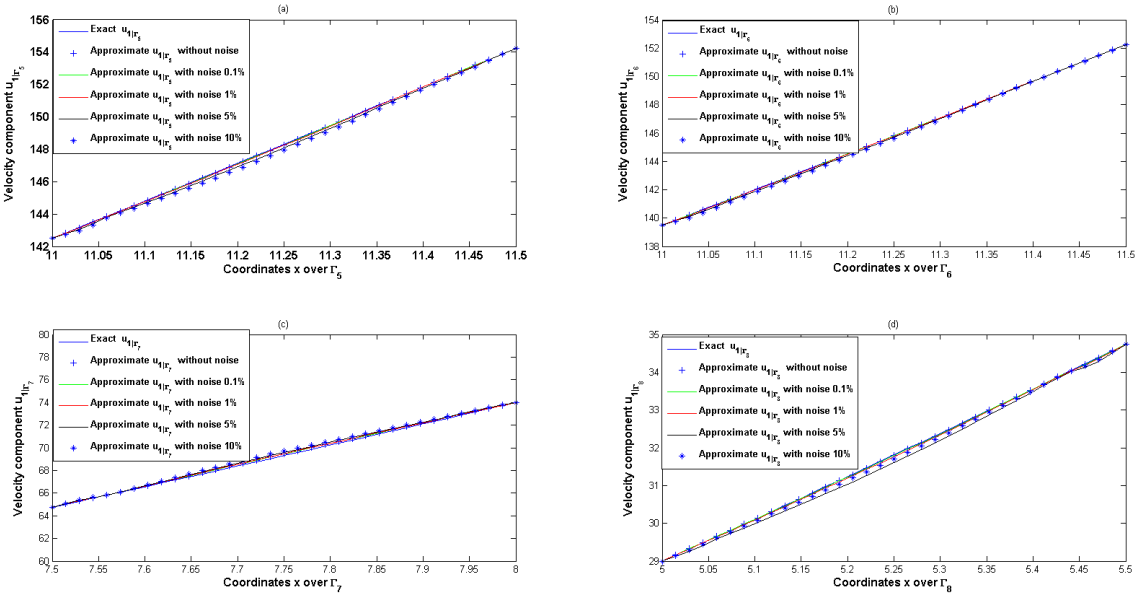
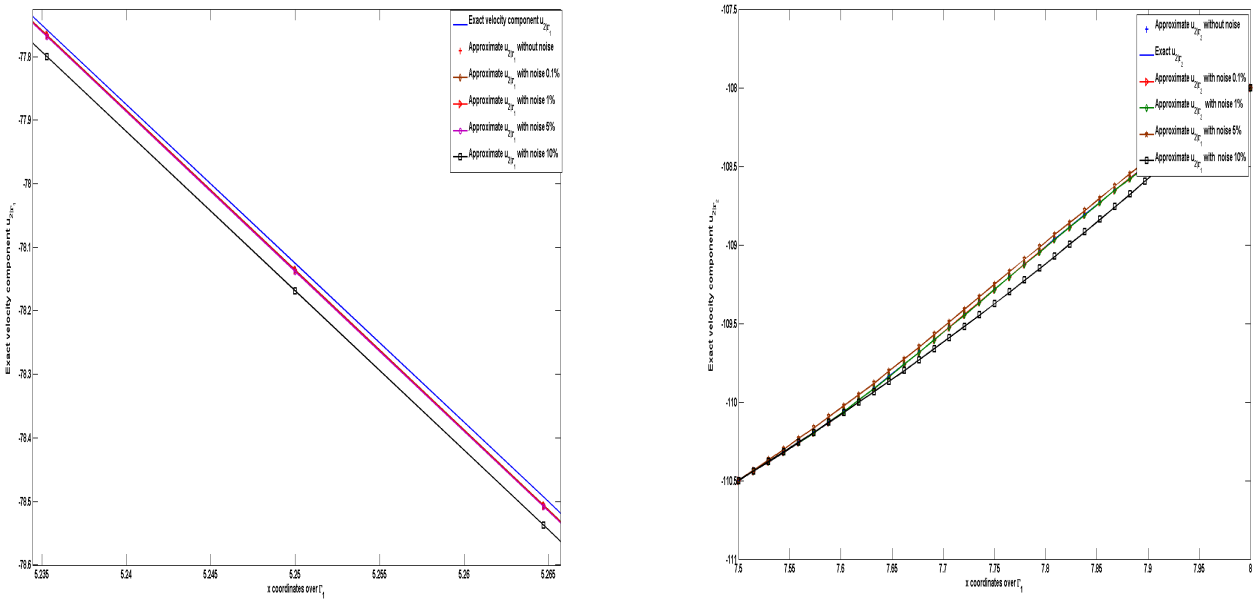
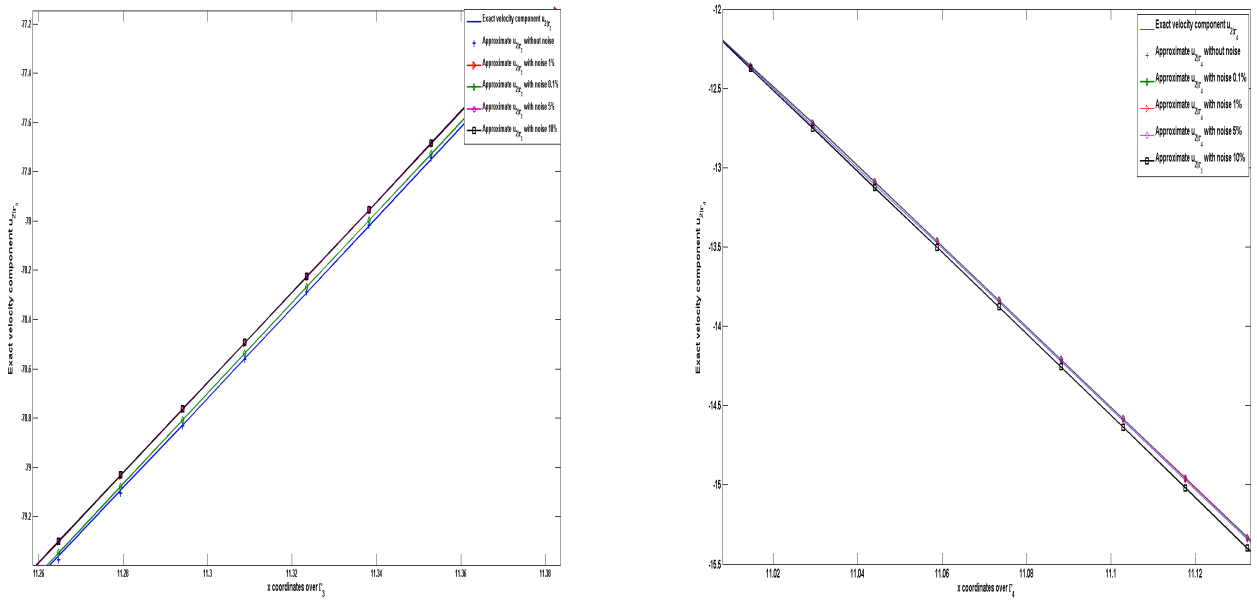


Figure 43: The exact, approximate and the perturbed solutions by different noise levels $0.1\%, 1\%, 5\%$ et 10% for the first component of the velocity u_1 on the boundaries Γ_i for $i = 5, 6, 7, 8$



(a) the boundary Γ_1 (b) the boundary Γ_2

Figure 44: The exact, approximate and the perturbed solutions by different noise levels $0.1\%, 1\%, 5\%$ et 10% for the second component of the velocity u_2 on the boundaries Γ_1 and Γ_2



(a) the boundary Γ_3 (b) the boundary Γ_4

Figure 45: The exact, approximate and the perturbed solutions by different noise levels $0.1\%, 1\%, 5\%$ et 10% for the second component of the velocity u_2 on the boundaries Γ_3 and Γ_4

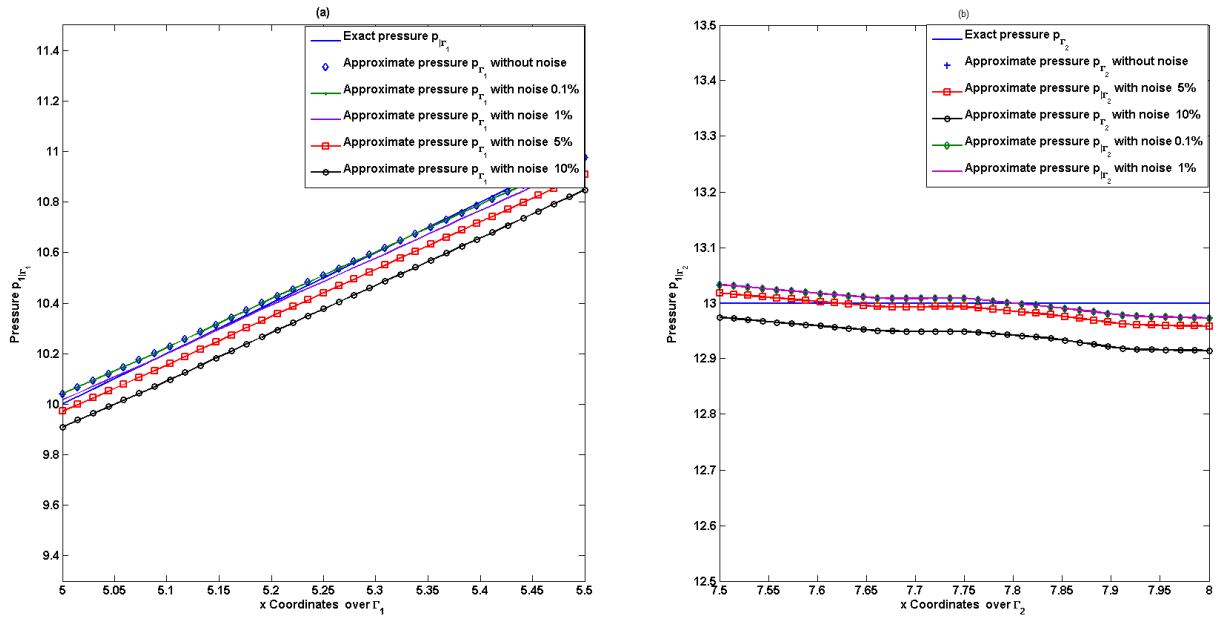


Figure 46: The exact, approximate and the perturbed solutions by different noise levels $1\%, 5\%$ et 10% for the pressure p on the boundaries Γ_1 and Γ_2

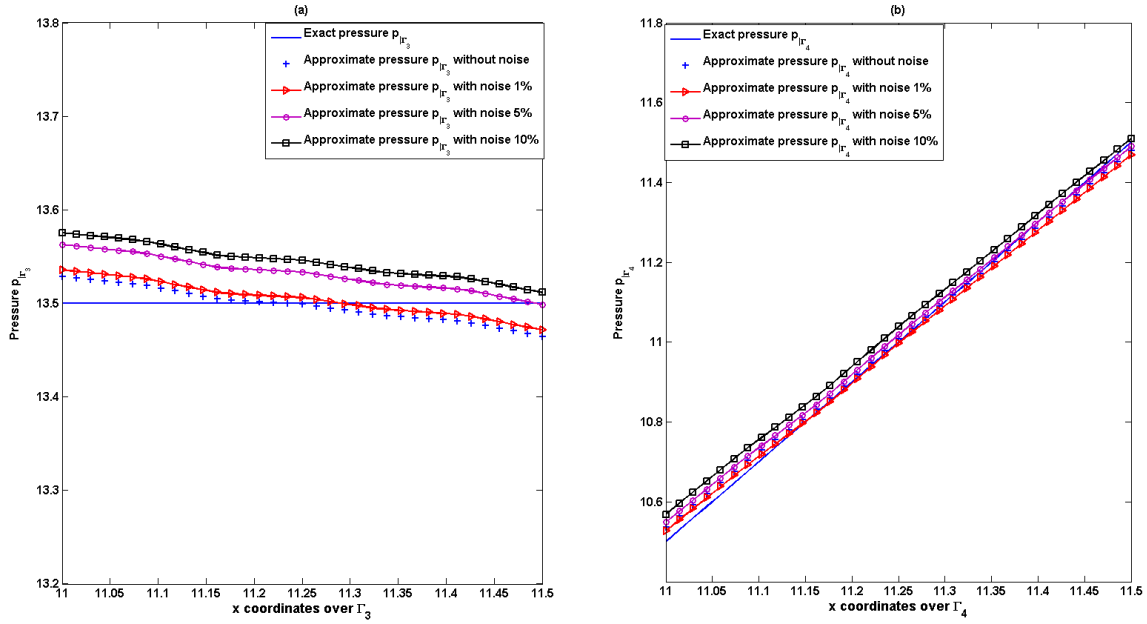


Figure 47: The exact, approximate and the perturbed solutions by different noise levels $1\%, 5\%$ et 10% for the pressure p on the boundaries Γ_3 and Γ_4

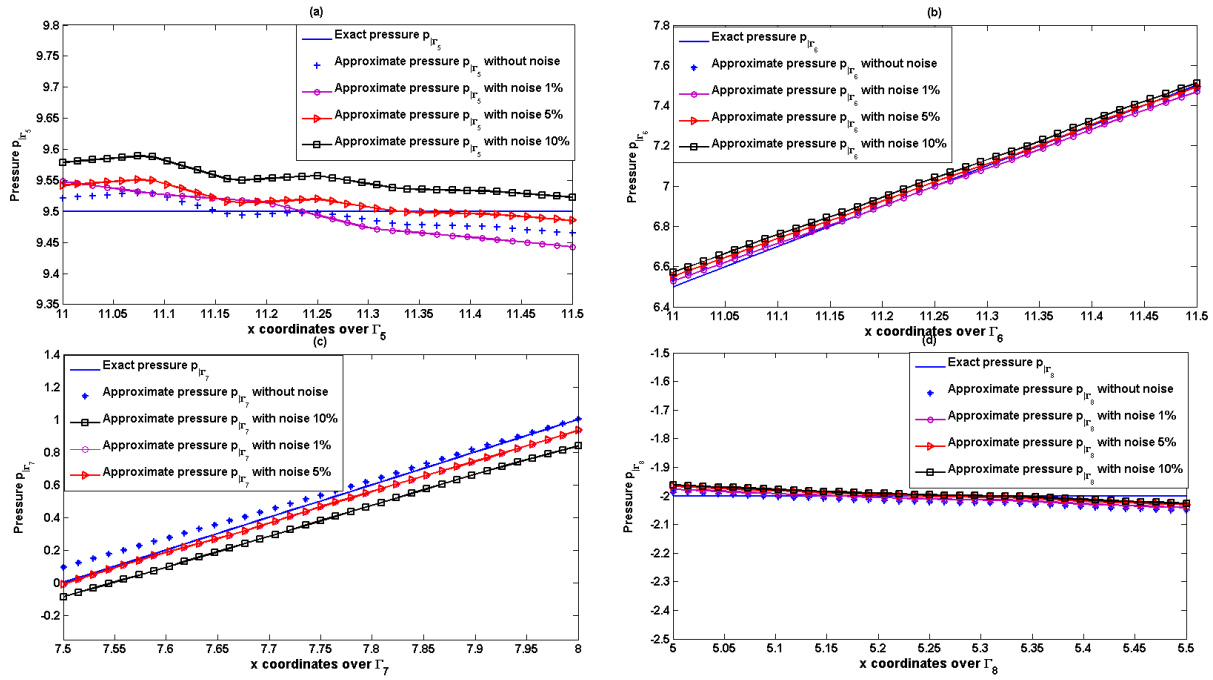


Figure 48: The exact, approximate and the perturbed solutions by different noise levels $1\%, 5\%$ et 10% for the pressure p on the boundaries Γ_i for $i = 5, 6, 7, 8$

References

[1] *A. B. Abda, I. B. Saad, and M. Hassine. Recovering boundary data: the cauchy stokes system.* Applied Mathematical Modelling, *37(1-2):1–12, 2013.*

[2] *S. Andrieux, T. Baranger, and A. B. Abda. Solving cauchy problems by minimizing an energy-like functional.* Inverse problems, *22(1):115, 2006.*

[3] *D. N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations.* Calcolo, *21(4):337–344, 1984.*

[4] *L. Baffico, C. Grandmont, and B. Maury. Multiscale modeling of the respiratory tract.* Mathematical Models and Methods in Applied Sciences, *20(01):59–93, 2010.*

[5] *F. Berntsson, V. A. Kozlov, L. Mpinganzima, and B. O. Turesson. An accelerated alternating procedure for the Cauchy problem for the Helmholtz equation.* Computers & Mathematics with Applications, *68(1):44–60, 2014.*

[6] *F. Caubet, J. Dardé, and M. Godoy. On the data completion problem and the inverse obstacle problem with partial cauchy data for laplaces equation.* ESAIM: Control, Optimisation and Calculus of Variations, *25:30, 2019.*

[7] *A. Chakib, A. Nachaoui, M. Nachaoui, and H. Ouaissa. On a fixed point study of an inverse problem governed by stokes equation.* Inverse Problems, *35(1):015008, 2018.*

[8] *G. Fairweather and A. Karageorghis. The method of fundamental solutions for elliptic boundary value problems.* Advances in Computational Mathematics, *9(1-2):69, 1998.*

[9] *L. Formaggia, J.-F. Gerbeau, F. Nobile, and A. Quarteroni. Numerical treatment of defective boundary conditions for the navier–stokes equations.* SIAM Journal on Numerical Analysis, *40(1):376–401, 2002.*

[10] *L. Formaggia, A. Veneziani, and C. Vergara. A new approach to numerical solution of defective boundary value problems in incompressible fluid dynamics.* SIAM Journal on Numerical Analysis, *46(6):2769–2794, 2008.*

[11] *L. Formaggia, A. Veneziani, and C. Vergara. Flow rate boundary problems for an incompressible fluid in deformable domains: formulations and solution methods.* Computer Methods in Applied Mechanics and Engineering, *199(9-12):677–688, 2010.*

[12] *J. Hadamard.* Lectures on Cauchy's problem in linear partial differential equations. *New York: Dover, 1953.*

[13] *J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible navier–stokes equations.* International Journal for Numerical Methods in Fluids, *22(5):325–352, 1996.*

[14] *R. E. Hyatt, D. P. Schilder, and D. L. Fry. Relationship between maximum expiratory flow and degree of lung inflation.* Journal of Applied Physiology, *13(3):331–336, 1958.*

[15] *M. Jourhmane and A. Nachaoui. An alternating method for an inverse cauchy problem.* Numerical Algorithms, *21(1-4):247, 1999.*

[16] *M. Jourhmane and A. Nachaoui. Convergence of an alternating method to solve the cauchy problem for poisson's equation.* Applicable analysis, *81(5):1065–1083, 2002.*

[17] M. V. Klibanov. Carleman estimates for the regularization of ill-posed Cauchy problems. Applied Numerical Mathematics*, 94:46–74, 2015.*

[18] V. A. Kozlov, V. G. Maz'Ya, and A. V. Fomin. An iterative method for solving the Cauchy problem for elliptic equations. Comput. Maths. Math. Phys.*, 31(1):45–52, 1991.*

[19] R. Lattés and J. L. Lions. The method of quasi-reversibility: applications to partial differential equations*. Elsevier, New York, 1969.*

[20] C. J. Murray and A. D. Lopez. Evidence-based health policy–lessons from the global burden of disease study. Science*, 274(5288):740–743, 1996.*

[21] A. Quarteroni, A. Veneziani, and C. Vergara. Geometric multiscale modeling of the cardiovascular system, between theory and practice. Computer Methods in Applied Mechanics and Engineering*, 302:193–252, 2016.*

[22] C. Tajani, J. Abouchabaka, and O. Abdoun. Data recovering problem using a new kmf algorithm for annular domain. American Journal of Computational Mathematics*, 2(2):88, 2012.*

[23] C. Tajani, J. Abouchabaka, and O. Abdoun. Kmf algorithm for solving the cauchy problem for helmholtz equation. Appl. Math. Sci*, 6:4577–4587, 2012.*

[24] C. Tajani, H. Kajtih, and A. Daanoun. Iterative method to solve a data completion problem for biharmonic equation for rectangular domain. Annals of West University of Timisoara-Mathematics and Computer Science*, 55(1):129–147, 2017.*

[25] A. N. Tikhonov, V. I. Arsenin, and F. John. Solutions of ill-posed problems*, volume 14. Winston Washington, DC, 1977.*