A STABILIZED FINITE ELEMENT METHOD FOR STREAM
FUNCTION VORTICITY FORMULATION OF NAVIER-STOKES
EQUATIONS

MOHAMED ABDELWAHED, NEJMEDDINE CHORFI, MAATOUG HASSINE

Communicated by Vicentiu Radulescu

Abstract. We studied the solvability of the two-dimensional stream function-vorticity
formulation of the Navier-Stokes equations. We use the time discretization and
the method of characteristics order one for solving a quasi-Stokes system that
we discretize by a piecewise continuous finite element method. A stabilization
technique is used to overcome the loss of optimal error estimate. Finally a
parallel numerical algorithm is presented and tested.

1. Introduction

The flow of an incompressible viscous fluid in a two dimensional domain \( \Omega \) is
characterized by two variables: its velocity \( u \) and its pressure \( p \). It is described by
Navier-Stokes problem

\[
\frac{\partial u}{\partial t} + u \nabla u - \nu \Delta u + \nabla p = f \quad \text{in } \Omega, \ t > 0,
\]
\[
\nabla \cdot u = 0 \quad \text{in } \Omega, \ t > 0.
\]

(1.1)

This system corresponds to the equation of the conservation of the quantity of
movement and the equation of conservation of mass. Here, \( \nu \) is the kinematic
viscosity of the fluid and \( f \) is a given function corresponding to the forces applied
to the fluid. In general, we add to this system a boundary condition on the border
\( \Gamma \) of the domain \( \Omega \), like

\[
u = u_d \quad \text{on } \Gamma
\]

(1.2)

known as the Dirichlet condition.

This formulation, called velocity-pressure formulation, can be rewritten by intro-
ducing two other scalar functions, called the vorticity (noted by \( \omega \)) and the stream
function (noted by \( \psi \)) \[2 \ 5 \ 8 \ 10 \ 11 \ 16 \ 17 \ 18\]. The link between these functions
is given by the relations:

\[
\omega = \nabla \times u \quad \text{and} \quad u = \nabla \times \psi.
\]

(1.3)
Then we obtain the following $\psi - \omega$ formulation equivalent to (1.1),

$$\begin{align*}
\frac{\partial \omega}{\partial t} + u \nabla \omega - \nu \Delta \omega &= \nabla \times f \quad \text{in } \Omega, \ t > 0, \\
\omega + \Delta \psi &= 0 \quad \text{in } \Omega, \ t > 0.
\end{align*}$$

(1.4)

The velocity boundary condition (1.2) implies that the function $\psi$ and its normal derivative $\frac{\partial \psi}{\partial n}$ are fixed on the boundary $\Gamma$ and given by

$$\psi = \psi_d \quad \text{and} \quad \frac{\partial \psi}{\partial n} = g \quad \text{on } \Gamma.$$  

Such a formulation has two main advantages. The first one is related to the automatic satisfaction of the divergence free condition. The second one concerns the reduction of the number of equations.

We propose to solve problem (1.4). A time discretization of this system using the characteristics method [7, 12, 14], leads to study, at each time step $\Delta t$, the following system, called quasi-Stokes problem.

$$\begin{align*}
\omega + \Delta \psi &= 0 \quad \text{in } \Omega \\
-\Delta \omega - \lambda \Delta \psi &= F \quad \text{in } \Omega
\end{align*}$$

(1.5)

where $\lambda = \frac{1}{\nu \Delta t}$ and $F = \lambda \omega^p + \frac{1}{\nu} \nabla \times f$ with $\omega^p(t, x) = \omega(t - \Delta t, x(t - \Delta t))$.

The resolution of the system (1.5) by a direct approach leads to the loss of one order in the error estimates. To optimize the behavior of our approach, we adapt the regularization-stabilization technique, introduced in [1, 3], to the Navier-Stokes equations. The main step of the proposed approach is to write the decomposition $\psi - \omega$ in a natural variational framework.

Such a decomposition permit us to built a linear piecewise stabilized finite element method having a good behavior and to obtain an optimal error estimate.

The numerical resolution of the obtained linear system is performed by the Bi-gradient Conjugate Stabilized method [13]. After an algorithmic analysis identifying the dependence between the different tasks and data involved, an implementation under MPI (Message Passing Interface) has been done [16]. We obtain then a parallel stabilized algorithm for the Navier-Stokes problem.

We start this paper by presenting respectively the time and spatial discretization of the considered problem. Then we describe the stabilized method and its advantage. The numerical analysis of the obtained system and the validation of the stabilization technique is presented in section 3. In addition we present a parallel algorithm. The performance of the proposed method is illustrated by some numerical results.

2. Discrete problem

We present in this section the time and spatial approximations of problem (1.4).

2.1. Time discretization. Let $T$ a fixed positive real and $f \in C(0, T, H^{-1}(\Omega))$. We consider the regular partition of $[0, T]$ into $N$ equal subintervals $[t_{i-1}, t_i]$, $1 \leq i \leq N$ with $t_0 < t_1 < \ldots < t_N = T$ and $\delta t = t_i - t_{i-1} = \frac{T}{N}$.

Let $\psi^{n+1} = \psi(., t^{n+1})$ and $\omega^{n+1} = \omega(., t^{n+1})$ being the approximations of $\psi$ and $\omega$ at time $t^{n+1} = (n+1)\delta t$. The time discretization of the problem (1.4) is given
by: Find $\psi^{n+1}$ and $\omega^{n+1}$ solutions to

\[
\left(\frac{\partial \omega}{\partial t} + u \nabla \omega\right)^{n+1} - \nu \Delta \omega^{n+1} = \nabla \times f^{n+1} \quad \text{in } \Omega \\
\omega^{n+1} + \Delta \psi^{n+1} = 0 \quad \text{in } \Omega \\
\psi^{n+1} = \psi_d \quad \text{on } \Gamma \\
\frac{\partial \psi^{n+1}}{\partial n} = g \quad \text{on } \Gamma 
\]

(2.1)

The total derivative of the function omega is approximated using characteristics scheme [7, 12] as follows

\[
\left(\frac{\partial}{\partial t} + u \nabla\right)\omega(x, t^{n+1}) \approx \frac{\omega(x, t^{n+1}) - \omega(X(x, t^{n+1}; t^n), t^n)}{\delta t} = \frac{\omega^{n+1} - \omega^n \circ X^n}{\delta t} 
\]

(2.2)

where $X^n(x) = X(x, t^{n+1}; t^n)$ is the position at time $t^n$ of particle of fluid which is at point $x$ at time $t^{n+1}$. System (2.1) becomes

\[
\omega^{n+1} + \Delta \psi^{n+1} = 0 \quad \text{in } \Omega \\
\Delta \omega^{n+1} + \frac{1}{\nu \delta t} \Delta \psi^{n+1} = -\frac{1}{\nu} \left(\nabla \times f^{n+1} + \frac{\omega^n \circ X^n}{\delta t}\right) \quad \text{in } \Omega \\
\psi^{n+1} = \psi_d \quad \text{on } \Gamma \\
\frac{\partial \psi^{n+1}}{\partial n} = g \quad \text{on } \Gamma 
\]

(2.3)

which is equivalent at each time step to the quasi-Stokes system

\[
\omega + \Delta \psi = 0 \quad \text{in } \Omega \\
\Delta \omega + \lambda \Delta \psi = F \quad \text{in } \Omega \\
\psi = \psi_d \quad \text{on } \Gamma \\
\frac{\partial \psi}{\partial n} = g \quad \text{on } \Gamma 
\]

(2.4)

where $F(\cdot, t^{n+1}) = -\frac{1}{\nu} (\nabla \times f^{n+1} + \frac{\omega^n \circ X^n}{\delta t})$, and $\psi_d, g, \lambda, F$ are given.

2.2. Continuous problem. First, we denote by $H^{-1}(\Omega)$ the dual space of

\[
H^1_0(\Omega) = \{ \theta \in L^2(\Omega) : \nabla \theta \in L^2(\Omega), v|_{\partial \Omega} = 0\}
\]

with the associated norm

\[
\|v\|_{-1, \Omega} = \sup_{\varphi \in H^1_0(\Omega)} \frac{(v, \varphi)_{-1,1,\Omega}}{\|\varphi\|_{1, \Omega}}, \quad \forall v \in H^{-1}(\Omega)
\]

where $\langle \cdot, \cdot \rangle_{-1, \Omega}$ is the dual pairing between $H^{-1}(\Omega)$ and $H^1_0(\Omega)$. We introduce the space

\[
H^{-1}(\Delta; \Omega) = \{ \theta \in L^2(\Omega) : \Delta \theta \in H^{-1}(\Omega)\}
\]

equipped with the norm

\[
\|\theta\|_{-1, \Delta, \Omega} = (\|\theta\|^2_{2, \Omega} + \|\Delta \theta\|^2_{-1, \Omega})^{1/2}, \quad \forall \theta \in H^{-1}(\Delta; \Omega)
\]

For simplicity of the analysis, we consider in the following $\psi_d = 0$ and $g = 0$. A variational formulation of problem (2.4) is given by: Find $(\omega, \psi) \in H^{-1}(\Delta; \Omega) \times
$H_0^1(\Omega)$ such that
\[ \int_{\Omega} \theta \omega d\Omega + \langle \Delta \theta, \psi \rangle_{-1,1,\Omega} = 0 \quad \forall \theta \in H^{-1}(\Delta; \Omega) \tag{2.5} \]
\[ \langle \Delta \omega, \eta \rangle_{-1,1,\Omega} - \lambda \int_{\Omega} \nabla \eta \nabla \psi d\Omega = \int_{\Omega} F \eta d\Omega \quad \forall \eta \in H_0^1(\Omega) \]

**Theorem 2.1.** For all $f \in H^{-1}(\Omega)$, problem (2.5) has a unique solution $(\omega, \psi) \in H^{-1}(\Delta; \Omega) \times H_0^1(\Omega)$ and there exists a constant $C > 0$ such that:
\[ |\psi|_{1,\Omega} + ||\omega||_{-1,\Delta,\Omega} \leq C ||f||_{-1,\Omega} \tag{2.6} \]
Moreover, the solution $(\omega, \psi)$ of (2.5) is a solution of (2.4).

For a proof of the above theorem see [4].

**Remark 2.2.** Using the decomposition
\[ \omega = \omega^0 + \omega^*, \tag{2.7} \]
problem (2.5) can be rewritten as: Find $\omega^0 \in H_0^1(\Omega)$ such that
\[ \int_{\Omega} \nabla \omega^0 \nabla \eta d\Omega = \langle F, \eta \rangle_{-1,1,\Omega}, \quad \forall \eta \in H_0^1(\Omega), \tag{2.8} \]
and find $\omega^* \in H^{-1}(\Delta; \Omega)$, $\psi \in H_0^1(\Omega)$ such that
\[ \int_{\Omega} \theta \omega^* d\Omega - \int_{\Omega} \nabla \psi \nabla \theta d\Omega = - \int_{\Omega} \theta \omega^0 d\Omega \quad \forall \theta \in H^1(\Omega), \tag{2.9} \]
\[-\langle \Delta \omega^*, \eta \rangle_{-1,1,\Omega} + \lambda \int_{\Omega} \nabla \psi \nabla \eta d\Omega = 0 \quad \forall \eta \in H_0^1(\Omega) \]
The advantage of this decomposition is to get more regularization on $\omega^*$, Indeed $-\Delta \omega^* = \lambda \Delta \psi \in L^2(\Omega)$ while $-\Delta \omega^0 = f \in H^{-1}(\Omega)$ and $-\Delta \omega = f + \lambda \Delta \psi \in H^{-1}(\Omega)$.

Next, we shall discretize problem (2.8)-(2.9) using the decomposition defined by (2.7). This particularity related to $\omega^*$ will improve the behavior of the discrete method.

**2.3. Numerical approximation.** Let $(\mathcal{T}_h)_h$ be a regular family of decomposition of $\Omega$ in triangles $K$ [9, 15]. For each triangle $K$, we denote respectively by $h_K$ its diameter and $h = \max_{K \in \mathcal{T}_h} h_K$.

We associate to each decomposition $\mathcal{T}_h$, the set $\mathcal{C}_h$ of the internal edges. Then, for every edge $T$ of $\mathcal{C}_h$, there exists two triangles $K$ and $K'$ of $\mathcal{T}_h$ such that
\[ K \neq K' \quad \text{and} \quad T = \partial K \cap \partial K'. \tag{2.10} \]
We define the jump of the normal derivative on each edge $T$ of $\mathcal{C}_h$, by
\[ [\partial_n v]_T = \begin{cases} \partial_{nK}^K v_K + \partial_{nK'} v_{K'}, & \text{p.p. on } T \quad \text{if } T = \partial K \cap \partial K', K, K' \in \mathcal{T}_h \vspace{1mm} \\ 0, & \text{on } T \quad \text{if } T = \partial K \cap \Gamma, K \in \mathcal{T}_h, \end{cases} \tag{2.11} \]
where $v_K = v|_K$ and $\partial_{nK}^K v_K$ is the normal derivative of $v_K$ on $\partial K$, $K \in \mathcal{T}_h$, and the discrete scalar product and semi-norm
\[ \langle \theta_h, \delta_h \rangle_h = \sum_{T \in \mathcal{T}_h} \text{mes } T \int_T [\partial_n \theta_h][\partial_n \delta_h] dT, \quad \forall \theta_h \in X_h, \delta_h \in X_h, \tag{2.12} \]
\[ |\theta_h|_h = \left( \langle \theta_h, \theta_h \rangle_h \right)^{1/2} \quad \forall \theta_h \in X_h, \]
where } mes \{ T \} \text{ denotes the length of the edge of } T \text{ for } T \in \mathcal{C}_h.

We consider the discrete spaces

\[ X_h = \{ \theta \in C^0(\Omega) : \forall K \in \mathcal{T}_h, \theta |_K \in P_1(K) \} \subset H^{-1}(\Delta; \Omega) \]

\[ X_h^0 = X_h \cap H^1_0(\Omega) \]

where } P_1(K) \text{ is the space of piecewise linear functions defined on } K. \text{ We denote by } \omega_h, \psi_h \text{ and } F_h \text{ respectively the approximations of } \omega, \psi \text{ and } F \text{ in } X_h. \text{ The classical discrete variational formulation of problem (2.8)-(2.9) is written}

\[ a(\omega_h, \theta_h) + \delta(\theta_h, \psi_h) = 0 \quad \forall \theta_h \in X_h \]

\[ \int_{\Omega} \omega_h \cdot \theta_h \, d\Omega \]

where } a, \delta \text{ and } \delta \text{ are the following three bilinear forms

\[ a(\omega_h, \theta_h) = \int_{\Omega} \omega_h \cdot \theta_h \, d\Omega, \quad \forall \omega_h, \theta_h \in X_h, \]

\[ \delta(\theta_h, \psi_h) = \int_{\Omega} \nabla \theta_h \cdot \nabla \psi_h \, d\Omega, \quad \forall \theta_h, \psi_h \in X_h, \]

\[ d(\varphi_h, \eta_h) = \lambda \int_{\Omega} \nabla \varphi_h \cdot \eta_h \, d\Omega, \quad \forall \varphi_h, \eta_h \in X_h \times X_h. \]

We remark that the compatibility constant between spaces } X_h \text{ and } X_h^0 \text{ is independent on } h \text{ but the associated bilinear form } a(\cdot, \cdot) \text{ to } X_h \text{ is elliptic for the } L^2(\Omega) \text{ norm. Consequently the coercivity constant depends on } h. \text{ To make up for such inconvenient, we use an approximation method based on a technique of stabilization described in } [2, 3]. \text{ It consists in changing this principal form } a(\cdot, \cdot) \text{ into another form } a_h(\cdot, \cdot) \text{ by addition a stabilizing term as

\[ a_h(\delta_h, \theta_h) = a(\delta_h, \theta_h) + \beta A_h(\delta_h, \theta_h) \quad \forall \delta_h, \theta_h \in X_h, \]

\[ \text{with } A_h(\delta_h, \theta_h) = (\delta_h, \theta_h) \text{ and } \beta \geq 0 \text{ is a parameter to be chosen.}

\textbf{Discrete problem.} Using stabilization technique, introduced in (2.14), the discrete variational formulation of (2.8)-(2.9) (which is equivalent to (2.5)) can be rewritten as

\[ \int_{\Omega} \nabla \omega_h^\ast \cdot \nabla \eta_h \, d\Omega = \langle f, \eta_h \rangle_{-1,1,\Omega}, \quad \forall \eta_h \in X_h^0, \]

\[ \int_{\Omega_h} \omega_h^\ast \theta_h \, d\Omega_h + \beta \langle \omega_h^\ast, \theta_h \rangle_{h} - \int_{\Omega_h} \nabla \psi_h \cdot \nabla \theta_h \, d\Omega_h = - \int_{\Omega_h} \omega_h^0 \theta_h \, d\Omega_h \quad \forall \theta_h \in X_h \]

\[ - \int_{\Omega_h} \nabla \omega_h^\ast \cdot \nabla \eta_h \, d\Omega_h - \lambda \int_{\Omega_h} \nabla \psi_h \cdot \nabla \eta_h \, d\Omega_h = 0 \quad \forall \eta_h \in X_h^0 \]

\[ \omega_h^\ast \in X_h^0, \omega_h \in X_h^0, \omega_h^\ast \in X_h, \psi_h \in X_h^0 \]

\[ (\psi_h, \omega_h) \text{ is an approximation of } (\psi, \omega). \]

\textbf{Theorem 2.3.} \textit{The discrete problem (2.15) has a unique solution.}

For a proof of the above theorem see [4].
Error estimates.

**Theorem 2.4.** If \( \omega^* \in H^2(\Omega) \) and \( \omega^0 \in H^2(\Omega) \), there exist \( s \in [\frac{1}{2}, 1] \) and \( C > 0 \), independent of \( h, \beta \) and \( \lambda \), such that

\[
\| \omega^* - \omega_h^* \|_{0, \Omega} + \sqrt{\lambda} |\psi - \psi_h|_{1, \Omega} \leq C (h^{s+1} |\omega^0|_{2, \Omega} + e_h),
\]

(2.16)

\[
|\psi - \psi_h|_{1, \Omega} \leq C h |\psi|_{2, \Omega} + C (1 + \sqrt{\beta})(h^{s+1} |\omega^0|_{2, \Omega} + e_h)
\]

(2.17)

where

\[
e_h = (\lambda h \sqrt{\beta} + \frac{\lambda}{\sqrt{\lambda}}) |\psi|_{2, \Omega} + (\lambda h + \sqrt{\lambda}) h |\omega^*|_{2, \Omega}
\]

The proof of the above theorem is an adaptation to the quasi-Stokes problem of the proof presented in [3] for the biharmonic problem. We omit it.

The parameter \( \beta \) is selected in such way that error estimates in theorem 2.4 becomes optimal. In the case of \( \sqrt{\beta} = \frac{1}{\sqrt{\lambda}} \),

\[
e_h = h \sqrt{\lambda} |\psi|_{2, \Omega} + (h^{s+1} + \frac{h}{\sqrt{\lambda}}) |\omega^*|_{2, \Omega}
\]

and we obtain the following error estimates.

**Corollary 2.5.** Under hypothesis of theorem 2.4

\[
\| \omega^* - \omega_h^* \|_{0, \Omega} \leq C (h^{s+1} + \frac{h}{\sqrt{\lambda}}),
\]

(2.19)

\[
|\psi - \psi_h|_{1, \Omega} \leq C \left( h + \frac{h^{s+1}}{\sqrt{\lambda}} \right),
\]

(2.20)

where the constant \( C > 0 \) is independent of \( h \) and \( \lambda \).

This choice of parameter \( \beta \) leads to an optimal error estimates, we get a behavior in \( O(h) \). We remark that if \( \lambda \) becomes very small, the error estimate on \( \psi \) is better than that on \( \omega \).

3. Numerical implementation

3.1. Linear system. Let \( n \) the number of mesh nodes, \( \{\theta_i\}_{i=1}^n \) be a base of \( X_h \) and \( \{\eta_j\}_{i=1}^n \) a base of \( X_0^m \). The system (2.13) is written for all \( j = 1, \ldots, n \)

\[
\sum_{i=1}^n \omega_i a(\theta_i, \theta_j) + \beta \sum_{i=1}^n \omega_i A_h(\theta_i, \theta_j) + \sum_{i=1}^n \psi_i b(\theta_j, \eta_i) = 0
\]

(3.1)

\[
\sum_{i=1}^n \omega_i b(\theta_i, \eta_j) - \sum_{i=1}^n \psi_i d(\eta_i, \eta_j) = \int_\Omega f \eta_j d\Omega
\]

which is equivalent to the linear system

\[
MX = N
\]

(3.2)

where

\[
M = \begin{pmatrix} A + \beta S & B^T \\ B & -D \end{pmatrix}, \quad X = \begin{pmatrix} \omega \\ \psi \end{pmatrix}, \quad N = \begin{pmatrix} 0 \\ F \end{pmatrix}
\]

- The matrix \( A = (a_{i,j})_{1 \leq i,j \leq n} \) with \( a_{i,j} = a(\theta_i, \theta_j) \) is computed from the bilinear form \( a(\cdot, \cdot) \).
• The matrix $S = (s_{i,j})_{1 \leq i,j \leq n}$ with $s_{ij} = A_h(\theta_i, \theta_j)$ represents the bilinear form $A_h(\cdot, \cdot)$.
• The matrix $B = (b_{i,j})_{1 \leq i,j \leq n}$ with $b_{ij} = b(\eta_i, \eta_j)$.
• $D = -\lambda B$.
• $F$ is associated to the second member with $F_j = \int_\Omega F_{\eta_j} \, d\Gamma$.

For the storage of these matrices, we used the Morse compact method keeping only non zero terms. For the resolution, we implement a parallel Bi-gradient Conjugate Stabilized (BICGSTAB) algorithm [13].

3.2. Parallel algorithm. The parallel implementation of the BICGSTAB algorithm was done with fortran and MPI library for communications [16]. We present in the following some numerical results to show the performance of the parallelized solver. The simulations are performed on a cluster of 8 PC’s and for different Meshes (Mesh 1 with $4 \times 10^4$ nodes, Mesh 2 with $16 \times 10^4$ nodes and Mesh 3 with $64 \times 10^4$ nodes). We remark that the number of unknowns is the double of the number of nodes and that from the meshes 1 to the 2 and from 2 to 3, we resolved a problem with four times more data.

Table 1. Computing time with respect to number of processors

<table>
<thead>
<tr>
<th>Mesh</th>
<th>1 proc</th>
<th>2 proc</th>
<th>4 proc</th>
<th>8 proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.05</td>
<td>11.13</td>
<td>6.55</td>
<td>4.02</td>
</tr>
<tr>
<td>2</td>
<td>245.85</td>
<td>128.04</td>
<td>70.85</td>
<td>37.7</td>
</tr>
<tr>
<td>3</td>
<td>2113.36</td>
<td>1061.98</td>
<td>548.9</td>
<td>297.65</td>
</tr>
</tbody>
</table>

From table 1 we remark that the execution time increases when we raise the number of unknowns and decreases when the number of processors increases. We deduce that more the mesh is finer or more the number of unknown is important, more the gain of time becomes significant. This is proportional to the treated case and is often polluted by the communication time between processors which becomes more significant when the number of processors increases.

To evaluate the efficiency of the algorithm, we compute the speed-up which represents the ratio between the execution times for $n$ processors and 1 processor. The optimal speed-up for $n$ processors is $n$. We present in table 2 the obtained speed-up for each mesh.

Table 2. Speed-up

<table>
<thead>
<tr>
<th>Mesh</th>
<th>2 proc</th>
<th>4 proc</th>
<th>8 proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.89</td>
<td>3.21</td>
<td>5.25</td>
</tr>
<tr>
<td>2</td>
<td>1.92</td>
<td>3.47</td>
<td>6.52</td>
</tr>
<tr>
<td>3</td>
<td>1.99</td>
<td>3.85</td>
<td>7.1</td>
</tr>
</tbody>
</table>

We remark that the speed-up increases and approaches its optimal when the number of unknowns increases. Indeed, for 2 processors we obtain a good speed-up for mesh 1 (1.89) which reaches its optimal value (1.99) for mesh 3. On the other hand for 8 processors we obtain bad speed-up (5.25) for mesh 1 but which becomes better (7.1) for mesh 3. This result is a consequence of the communication
time between processors which becomes very important compared to the calculation volume when we increase the number of processors. Therefore, the communication cost degrades the performance of the parallel algorithm.

3.3. Numerical validation. We present in this section a validation of the presented stabilization technique on the quasi-Stokes problem \((2.4)\). We consider the domain of computation \(\Omega = [0, 1] \times [0, 1]\) and the following analytical stream function solution

\[
\psi(x, y) = 3x \sin(\pi x) \cos(\pi y) \quad 0 < x, y < 1.
\]

Figure 1 shows the total error \(\|\omega_1 - \omega_{1h}\|_{-1, \Delta, \Omega} + |\psi_1 - \psi_{1h}|_{1, \Omega}\) with respect to the stabilization term \(\beta\). We remark that the lowest error for this case is obtained for \(\beta = 0.1\).

Figure 2 shows the slope of the total error according to the parameter \(\beta\). We remark that this slope is included in the interval \([1.4, 2]\) which confirms the obtained error estimate.

Figures 3 and 4 present a comparison between the exact and computed solutions. Figures 3 (a) and 4 (a) show respectively the exact stream function and vorticity...
isovalues. We give also a comparison between the stabilized method corresponding to $\beta = 0.1$ (figures 3(c) and 4(c)) and the classical finite element method corresponding to $\beta = 0$ (figures 3(b) and 4(b)). One can see the contribution of the stabilization term $\beta$ on the quality of the solution. This contribution is more clear for the vorticity (figure 3(c)).

**Conclusion.** A stabilized finite element method for the Navier-Stokes equations written in stream function-vorticity formulation is presented in this work. In order to optimize the computing time, a parallel implementation for the obtained solver is presented. The proposed approach will be extended in a forthcoming work to a three dimensional Navier-Stokes equations. we denote that the proposed algorithm is general and can be used for many engineering problems related to the Navier-Stokes equations.

**Acknowledgements.** The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for funding this Research group No (RG-1435-026).
References


Mohamed Abdelwahed
Department of Mathematics, College of Sciences, King Saud University, Riyadh, Saudi Arabia
E-mail address: mabdelwahed@ksu.edu.sa

Nejmeddine Chorfi
Department of Mathematics, College of Sciences, King Saud University, Riyadh, Saudi Arabia
E-mail address: nchorfi@ksu.edu.sa

Maatoug Hassine
Department of Mathematics, College of Sciences, Monastir University, Tunisia
E-mail address: maatoug_hassine@yahoo.fr