J. KSIAM Vol.15, No.3, 233–251, 2011

AN OVERVIEW OF BDF2 GAUGE-UZAWA METHODS FOR INCOMPRESSIBLE FLOWS

JAE-HONG PYO

DEPARTMENT OF MATHEMATICS, KANGWON NATIONAL UNIVERSITY, REPUBLIC OF KOREA
E-mail address: jhpyo@kangwon.ac.kr

ABSTRACT. The Gauge-Uzawa method [GUM] in [9] which is a projection type algorithm to solve evolution Navier-Stokes equations has many advantages and superior performance. But this method has been studied for backward Euler time discrete scheme which is the first order technique, because the classical second order GUM requests rather strong stability condition. Recently, the second order time discrete GUM was modified to be unconditionally stable and estimated errors in [12]. In this paper, we contemplate several GUMs which can be derived by the same manner within [12], and we dig out properties of them for both stability and accuracy. In addition, we evaluate an stability condition for the classical GUM to construct an adaptive GUM for time to make free from strong stability condition of the classical GUM.

1. INTRODUCTION AND THE FIRST ORDER GAUGE-UZAWA METHOD

Given an open bounded polyhedral domain Ω in $\mathbb{R}^d$, with $d = 2$ or 3, we consider the time-dependent Navier-Stokes Equations of incompressible fluids:

$$\begin{aligned} \mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \mu\triangle\mathbf{u} &= \mathbf{f}, \qquad \text{in } \Omega, \\ \nabla\cdot\mathbf{u} &= 0, \qquad \text{in } \Omega, \\ \mathbf{u}(0,\mathbf{x}) &= \mathbf{u}^0, \quad\ \text{in } \Omega, \end{aligned} \tag{1.1}$$

with vanishing Dirichlet boundary condition $\mathbf{u} = \mathbf{0}$ on $\partial\Omega$ and pressure mean-value $\int_\Omega p = 0$. The primitive variables are the (vector) velocity $\mathbf{u}$ and the (scalar) pressure p. The viscosity $\mu = Re^{-1}$ is the reciprocal of the Reynolds number Re. Hereafter, vectors are denoted in boldface.

Received by the editors May 30, 2011; Revised August 25, 2011; Accepted in revised form September 15, 2011.

1991 *Mathematics Subject Classification.* 65M12, 65M15, 76D05.

Key words and phrases. Projection method, Gauge method, Uzawa method, Gauge-Uzawa method, Navier-Stokes equation, incompressible fluids.

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2007-331-C00043).

The coupling of pressure p and velocity $\mathbf{u}$ in the momentum equation is responsible for compatibility conditions between the spaces for $\mathbf{u}$ and p, characterized by the celebrated inf-sup condition, and associated numerical difficulties [3]. In the late 60's, projection methods which are decoupling methods and to reduce the computational cost were introduced independently by Chorin [1] and Temam [15]. And the methods quickly gained popularity in the computational fluid dynamics community, and over the years, an enormous amount of efforts have been devoted to develop more accurate and efficient projection type schemes, we refer to [11, 4] for comprehensive and up-to-date review on this subject.

The first order Gauge-Uzawa method [GUM] has been constructed in [9] to solve (1.1) and enhanced in [10, 14] to solve more complicated problems which are the Boussinesq equations and the non-constant density fluid problems. The first order GUM to solve Navier-Stokes equations in [9] can be summarized as

Algorithm 1 (The first order Gauge-Uzawa Method)**.** *Set* $\phi^0 = 0$ *and* $s^0 = 0$ *and repeat for* $1 \le n \le N = [\frac{T}{\tau} - 1]$.

Step 1: *Find* $\widehat{\mathbf{u}}^{n+1}$ *as the solution of*

$$\frac{\widehat{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\tau} + (\mathbf{u}^n \cdot \nabla)\, \widehat{\mathbf{u}}^{n+1} + \mu s^n - \mu \triangle \widehat{\mathbf{u}}^{n+1} = \mathbf{f}(t^{n+1}),$$
$$\widehat{\mathbf{u}}^{n+1}|_\Gamma = \mathbf{0}.$$

Step 2: *Find* ϕ^{n+1} *as the solution of*

$$-\triangle \phi^{n+1} = \nabla\cdot \widehat{\mathbf{u}}^{n+1},$$
$$\partial_{\boldsymbol{\nu}} \phi^{n+1}|_\Gamma = 0.$$

Step 3: *Update* $\mathbf{u}^{n+1}$ *and* s^{n+1} *by*

$$\mathbf{u}^{n+1} = \widehat{\mathbf{u}}^{n+1} + \nabla \phi^{n+1},$$
$$s^{n+1} = s^n - \nabla \cdot \widehat{\mathbf{u}}^{n+1}.$$

Pressure: *Compute pressure whenever necessary as*

$$p^{n+1} = -\frac{\phi^{n+1}}{\tau} + \mu \triangle \phi^{n+1}. \tag{1.2}$$

It is proved in [9] that the Algorithm 1 is unconditionally stable and hold error bounds

$$\tau \sum_{n=1}^{N} \left(\left\| \mathbf{u}(t^{n+1}) - \mathbf{u}^{n+1} \right\|_0^2 + \left\| \mathbf{u}(t^{n+1}) - \widehat{\mathbf{u}}^{n+1} \right\|_0^2 \right) \le C\tau^2,$$
$$\tau \sum_{n=1}^{N} \left(\left\| \mathbf{u}(t^{n+1}) - \widehat{\mathbf{u}}^{n+1} \right\|_1^2 + \left\| p(t^{n+1}) - p^{n+1} \right\|_0^2 \right) \le C\tau.$$

We has been concentrated to extend GUM to the second order scheme using the second order backward Euler formula [BDF2]. In [13], we evaluate the normal mode solutions for several projection methods and then conclude that GUM with BDF2 is more accurate algorithm than any other projection methods. But we couldn't prove stability condition and evaluate errors via energy estimate for the classical GUM. Recently, an unconditionally stable GUM has been constructed without loss consistency to (1.1) and analyzed errors by energy estimate in [12]. Also they proved that the new GUM is equivalent to the rotational form of pressure correction method in [5]. The errors have been evaluated in [5] for the Stokes equations and it is extended to the Navier-Stokes equations in[12]. The new GUM in [12] modifies the pressure equation to be able to evaluate errors, but the numerical accuracy of the method is not better than the classical GUM. Both GUMs consist to (1.1) and have same origin, but they have totally different numerical behavior. So we are interested in other definitions for the pressure equation. In this paper, we will introduce 3 different GUMs by imposing reasonable and representative pressure equations (2.4), (2.5), and (2.6) below and perform numerical tests to find out properties of them. In addition, we prove an stability constraint for the classical GUM to construct an adaptive GUM for time which is called θ scheme .

We now define notations. Let $H^s(\Omega)$ be the Sobolev space with s derivatives in $L^2(\Omega)$, $\mathbf{L}^2(\Omega) = (L^2(\Omega))^d$ and $\mathbf{H}^s(\Omega) = (H^s(\Omega))^d$, where $d = 2, 3$. Let $\|\cdot\|_0$ denote the $\mathbf{L}^2(\Omega)$ norm, and $\langle \cdot\,,\,\cdot \rangle$ the corresponding inner product. Let $\|\cdot\|_s$ denote the norm of $H^s(\Omega)$ for $s \in \mathbb{R}$.

This paper is organized as follows. We introduce the BDF2 GUMs and discuss about properties of them in §2. We also introduce θ scheme for Navier-Stokes equations in §3 to the classical GUM. We perform stability proof in §4. We finally conclude in §5 with numerical tests.

2. The second order Gauge-Uzawa Algorithms and stabilizations

In this section, we consider 3 GUMs and discuss about advantages and disadvantages of them. Because GUM imposing BDF2 has been introduced in [13, 12], we will reconstruct briefly from BDF2 time discrete Stokes equations:

$$\frac{3\mathbf{u}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\tau} + \nabla p^{n+1} - \mu \mathbf{u}^{n+1} = \mathbf{f}(t^{n+1}). \tag{2.1}$$

We introduce 2 artificial variables $\widehat{\mathbf{u}}^{n+1}$ and ϕ^{n+1} satisfying

$$\widehat{\mathbf{u}}^{n+1} = \mathbf{u}^{n+1} - \nabla\left(\phi^{n+1} - 2\phi^n + \phi^{n-1}\right). \tag{2.2}$$

Inserting (2.2) into (2.1) yields

$$\begin{aligned}&\frac{3\widehat{\mathbf{u}}^{n+1}-4\mathbf{u}^{n}+\mathbf{u}^{n-1}}{2\tau}+\nabla p^{n+1}-\mu\widehat{\mathbf{u}}^{n+1}\\&+\nabla\left(\frac{3\phi^{n+1}-6\phi^{n}+3\phi^{n-1}}{2\tau}-\mu\triangle\left(\phi^{n+1}-2\phi^{n}+\phi^{n-1}\right)\right)=\mathbf{f}(t^{n+1}).\end{aligned}\tag{2.3}$$

In order to end derive GUM, we have to define pressure equation to split (2.3) to 2 decoupled equations. Because the performance of GUM depends sensitively on the definition, we consider 3 reasonable and representative pressure equations:

- the classical GUM uses

$$\frac{3\phi^{n+1}-4\phi^{n}+\phi^{n-1}}{2\tau}-\mu\triangle\phi^{n+1}=-p^{n+1}\tag{2.4}$$

- the stabilized GUM hires

$$\frac{3\phi^{n+1}-3\phi^{n}}{2\tau}-\mu\triangle\left(\phi^{n+1}-\phi^{n}\right)=-p^{n+1}\tag{2.5}$$

- the 2nd order extrapolate pressure GUM imposes

$$\frac{3\phi^{n+1}}{2\tau}-\mu\triangle\phi^{n+1}=-p^{n+1}\tag{2.6}$$

If we impose (2.4), then (2.3) becomes

$$\begin{aligned}\frac{3\widehat{\mathbf{u}}^{n+1}-4\mathbf{u}^{n}+\mathbf{u}^{n-1}}{2\tau}-\nabla\left(\frac{\phi^{n}-\phi^{n-1}}{\tau}-\mu\triangle\left(2\phi^{n}-\phi^{n-1}\right)\right)&\\-\mu\triangle\widehat{\mathbf{u}}^{n+1}=\mathbf{f}(t^{n+1}).&\end{aligned}\tag{2.7}$$

In light of $\nabla\cdot\mathbf{u}^{n+1}=0$, (2.2) leads us

$$-\triangle\phi^{n+1}=-2\triangle\phi^{n}+\triangle\phi^{n-1}+\nabla\cdot\widehat{\mathbf{u}}^{n+1}.$$

To deal with the third order term $\nabla\triangle\phi^{n}$, which is a source of trouble due to lack of commutativity of the differential operators at the discrete level, we introduce the variable $s^{n+1}=\triangle\phi^{n+1}$ and then

$$s^{n+1}=2s^{n}-s^{n-1}-\nabla\cdot\widehat{\mathbf{u}}^{n+1}.$$

We now added up convection term in (2.7) with using the second order extrapolate technique $\mathbf{u}^{*}=2\mathbf{u}^{n}-\mathbf{u}^{n-1}$. Also we can use any other suitable higher order technique. Finally, we arrive at the second order GUM via gathering above equations.

Algorithm 2 (The classical second order Gauge-Uzawa Method)**.** *Set* $\mathbf{u}^1$*,* ϕ^1*, and* s^1 *using the first-order GUM Algorithm 1. Repeat for* $1\le n\le N=[\frac{T}{\tau}-1]$.

Step 1: *Set* $\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ *and find* $\widehat{\mathbf{u}}^{n+1}$ *as the solution of*

$$\begin{aligned}\frac{3\widehat{\mathbf{u}}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\tau} - \nabla\left(\frac{\phi^n - \phi^{n-1}}{\tau} - \mu\left(2s^n - s^{n-1}\right)\right)& \\ +(\mathbf{u}^* \cdot \nabla)\widehat{\mathbf{u}}^{n+1} - \mu\triangle\widehat{\mathbf{u}}^{n+1} &= \mathbf{f}(t^{n+1}), \\ \widehat{\mathbf{u}}^{n+1}|_\Gamma &= \mathbf{0}.\end{aligned} \tag{2.8}$$

Step 2: *Find* ϕ^{n+1} *as the solution of*

$$\begin{aligned}&-\triangle\phi^{n+1} = -\triangle\left(2\phi^n - \phi^{n-1}\right) + \nabla \cdot \widehat{\mathbf{u}}^{n+1}, \\ &\partial_{\boldsymbol{\nu}}\phi^{n+1}|_\Gamma = 0.\end{aligned} \tag{2.9}$$

Step 3: *Update* $\mathbf{u}^{n+1}$ *and* s^{n+1} *by*

$$\begin{aligned}&\mathbf{u}^{n+1} = \widehat{\mathbf{u}}^{n+1} + \nabla\left(\phi^{n+1} - 2\phi^n + \phi^{n-1}\right), \\ &s^{n+1} = 2s^n - s^{n-1} - \nabla \cdot \widehat{\mathbf{u}}^{n+1}.\end{aligned} \tag{2.10}$$

Pressure: *Compute pressure whenever necessary as*

$$p^{n+1} = -\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\tau} + \mu s^{n+1}. \tag{2.11}$$

Remark 2.1 (The initial setting for Algorithm 2)**.** *In the classical GUM Algorithm 2, the pressure equation* (2.11) *is the BDF2 time discrete scheme of the heat equation, which has the same form as the pressure equation* (1.2) *in the first order GUM Algorithm 1. It means that Algorithms 1 and 2 consist each other. So one natural way to set initial values at* t^1 *to launch Algorithm 2 by using Algorithm 1.*

The errors of Algorithm 2 decay fully second order rate in the numerical tests in §5, but this method suffers from weak stability. We will prove the following theorem in §4.

Theorem 1 (Stability of Algorithm 2)**.** *If* τ *is small enough to hold*

$$\tau\mu^2\|\nabla s^n\|_0^2 < M, \quad \forall 1 < n < N, \tag{2.12}$$

then the a priori bound of GU Algorithm 2 holds

$$
\begin{aligned}
&\left\|\mathbf{u}^{N+1}\right\|_0^2 + \left\|2\mathbf{u}^{N+1} - \mathbf{u}^N\right\|_0^2 + 2\left\|\nabla\left(\phi^{N+1} - \phi^N\right)\right\|_0^2 + \tau\mu\sum_{n=1}^{N}\left\|\nabla\widehat{\mathbf{u}}^{n+1}\right\|_0^2 \\
&+ \sum_{n=1}^{N}\left(\left\|\mathbf{u}^{n+1} - 2\mathbf{u}^n + \mathbf{u}^{n-1}\right\|_0^2 + 4\left\|\nabla\left(\phi^{n+1} - 2\phi^n + \phi^{n-1}\right)\right\|_0^2\right) \\
&+ 2\tau\mu\left\|s^{N+1} - s^N\right\|_0^2 \le \left\|\mathbf{u}^1\right\|_0^2 + \left\|2\mathbf{u}^1 - \mathbf{u}^0\right\|_0^2 + 2\left\|\nabla\left(\phi^1 - \phi^0\right)\right\|_0^2 \\
&\qquad + C\frac{\tau}{\mu}\sum_{n=1}^{N}\left\|\mathbf{f}(t^{n+1})\right\|_{-1}^2 + 2\tau\mu\left\|s^1 - s^0\right\|_0^2 + M.
\end{aligned}
\tag{2.13}
$$

The stability constraint (2.12) is rather strong condition, and so Algorithm 2 needs to choose small enough τ. Also (2.12) is an unusual constraint, because of including amount of numerical value s^n. However, we can evaluate $\|\nabla s^n\|_0$ before start computing $n+1$ time step, so we can derive an adaptive Algorithm 5 in §3 to control time marching step size.

Recently, an stabilized GUM has been introduced in [12] by imposing the pressure equation (2.5) which makes (2.3) yield

$$
\begin{aligned}
&\frac{3\widehat{\mathbf{u}}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\tau} - \mu\widehat{\mathbf{u}}^{n+1} \\
&- \nabla\left(\frac{3\phi^n - 3\phi^{n-1}}{2\tau} - \mu\triangle\left(\phi^n - \phi^{n-1}\right)\right) = \mathbf{f}(t^{n+1}).
\end{aligned}
\tag{2.14}
$$

Because the functions of ϕ in 3 equations (2.2), (2.5), and (2.14) are formed by the subtraction of 2 consecutive functions, we use simple notation $\psi^{n+1} := \phi^{n+1} - \phi^n$. Owing to divergence free condition $\nabla\cdot\mathbf{u}^{n+1} = 0$, (2.5) gives

$$
\begin{aligned}
-\triangle\psi^{n+1} &= -\triangle\left(\phi^{n+1} - \phi^n\right) \\
&= -\triangle\left(\phi^n - \phi^{n-1}\right) + \nabla\cdot\widehat{\mathbf{u}}^{n+1} = -\triangle\psi^n + \nabla\cdot\widehat{\mathbf{u}}^{n+1}.
\end{aligned}
\tag{2.15}
$$

To deal with the third order term $\nabla\triangle\phi^n$, which is a source of trouble due to lack of commutativity of the differential operators at the discrete level, we denote $q^{n+1} := \triangle\psi^{n+1}$. So (2.15) can be rewritten by

$$q^{n+1} = q^n - \nabla\cdot\widehat{\mathbf{u}}^{n+1},$$

which is connected with the Uzawa iteration.

Algorithm 3 (The stabilized Gauge-Uzawa Method). *Compute* $\mathbf{u}^1$ *and* p^1 *via any first order projection method and set* $\psi^1 = \frac{-2\tau}{3}p^1$ *and* $q^1 = 0$. *Repeat for* $1 \le n \le N = [\frac{T}{\tau} - 1]$.

Step 1: *Set* $\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ *and find* $\widehat{\mathbf{u}}^{n+1}$ *as the solution of*

$$\frac{3\widehat{\mathbf{u}}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\tau} + \nabla p^n + (\mathbf{u}^* \cdot \nabla)\widehat{\mathbf{u}}^{n+1} - \mu\triangle\widehat{\mathbf{u}}^{n+1} = \mathbf{f}(t^{n+1}),$$
$$\widehat{\mathbf{u}}^{n+1}|_\Gamma = \mathbf{0}.$$

Step 2: *Find* ψ^{n+1} *as the solution of*

$$-\triangle\psi^{n+1} = -\triangle\psi^n + \nabla\cdot\widehat{\mathbf{u}}^{n+1},$$
$$\partial_{\boldsymbol{\nu}}\psi^{n+1}|_\Gamma = 0.$$

Step 3: *Update* $\mathbf{u}^{n+1}$ *and* q^{n+1} *by*

$$\mathbf{u}^{n+1} = \widehat{\mathbf{u}}^{n+1} + \nabla\left(\psi^{n+1} - \psi^n\right)$$
$$q^{n+1} = q^n - \nabla\cdot\widehat{\mathbf{u}}^{n+1}.$$

Step 4: *Update pressure* p^{n+1} *by*

$$p^{n+1} = -\frac{3\psi^{n+1}}{2\tau} + \mu q^{n+1}.$$

In [12], they conclude that Algorithm 3 in continuous level is equivalent to the rotational form of pressure correction method which is introduced in [5]. It is proved in [12] that Algorithm 3 is unconditionally stable. In addition, the errors for Algorithm 3 is evaluated in [5] for Stokes equations and extended for Navier-Stokes equations in [12] under following assumptions:

Assumption 1 (Regularity of Ω)**.** *Let* $\{\mathbf{v}, r\}$ *be the unique solution of the steady Stokes equation*

$$-\triangle\mathbf{v} + \nabla r = \mathbf{w}, \qquad in\ \Omega,$$
$$\nabla\cdot\mathbf{v} = 0, \qquad in\ \Omega,$$
$$\mathbf{v} = 0, \qquad on\ \partial\Omega.$$

Then $\{\mathbf{v}, r\}$ *satisfies*

$$\|\mathbf{v}\|_2 + \|r\|_1 \le C\|\mathbf{w}\|_0.$$

We remark that the validity of Assumption 1 is known if $\partial\Omega$ is of class $\mathbf{C}^2$ [2, 6], or if $\partial\Omega$ is a two-dimensional convex polygon [7], and is generally believed for convex polyhedral [6].

Assumption 2 (Initial setting)**.** *Let* $(\mathbf{u}(t^1), p(t^1))$ *be the exact solution of* (1.1) *at* $t = t^1$. *The initial value* $(\mathbf{u}^1, p^1)$ *satisfies*

$$\left\|\mathbf{u}(t^1) - \mathbf{u}^1\right\|_0 \le C\tau^2 \quad and \quad \left\|\mathbf{u}(t^1) - \mathbf{u}^1\right\|_1 + \left\|p(t^1) - p^1\right\|_0 \le C\tau.$$

Lemma 2.2 (Error estimates)**.** *Suppose the exact solution of* (1.1) *is smooth enough. If Assumptions 1 and 2 hold, then the errors of Algorithms 3 will be bounded by*

$$\tau \sum_{n=1}^{N} \left(\left\| \mathbf{u}(t^{n+1}) - \mathbf{u}^{n+1} \right\|_0^2 + \left\| \mathbf{u}(t^{n+1}) - \widehat{\mathbf{u}}^{n+1} \right\|_0^2 \right) \leq C\tau^4,$$
$$\tau \sum_{n=1}^{N} \left(\left\| \mathbf{u}(t^{n+1}) - \widehat{\mathbf{u}}^{n+1} \right\|_1^2 + \left\| p(t^{n+1}) - p^{n+1} \right\|_0^2 \right) \leq C\tau^2.$$

Furthermore, if assumption 2 hold, then we have

$$\left\| \nabla \cdot \widehat{\mathbf{u}}^{n+1} \right\|_0 \leq C\tau^{\frac{3}{2}}.$$

The errors of Algorithm 3 decay fully second order rate in the numerical tests in §5, but the form of pressure term in (2.8) seems like the first order extrapolation. So we are interested in more accurate pressure scheme via imposing pressure equation (2.6) which makes (2.3) become

$$\frac{3\widehat{\mathbf{u}}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\tau} + \nabla\left(2p^n - p^{n-1}\right) - \mu\triangle\widehat{\mathbf{u}}^{n+1} = \mathbf{f}(t^{n+1}).$$

So we arrive at the new GUM which hires the 2nd order extrapolation on pressure.

Algorithm 4 (The accurate pressure Gauge-Uzawa Method)**.** *Set* $\mathbf{u}^1$*,* ϕ^1*, and* s^1 *using the first-order GUM Algorithm 1. Repeat for* $1 \leq n \leq N = [\frac{T}{\tau} - 1]$*.*

Step 1: *Set* $\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ *and find* $\widehat{\mathbf{u}}^{n+1}$ *as the solution of*

$$\begin{aligned} \frac{3\widehat{\mathbf{u}}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\tau} &+ \nabla\left(2p^n - p^{n-1}\right) \\ +(\mathbf{u}^* \cdot \nabla)\widehat{\mathbf{u}}^{n+1} - \mu\triangle\widehat{\mathbf{u}}^{n+1} &= \mathbf{f}(t^{n+1}), \\ \widehat{\mathbf{u}}^{n+1}|_\Gamma &= \mathbf{0}. \end{aligned} \tag{2.16}$$

Step 2: *Find* ϕ^{n+1} *as the solution of* (2.9)*.*
Step 3: *Update* $\mathbf{u}^{n+1}$ *and* s^{n+1} *by* (2.10)*.*
Step 4: *Compute pressure by addition of 2 functions as*

$$p^{n+1} = -\frac{3\phi^{n+1}}{2\tau} + \mu s^{n+1}. \tag{2.17}$$

Remark 2.3 (The property of Algorithm 4)**.** *The pressure in* (2.16) *is imposed the 2nd order extrapolate scheme, so Algorithm 4 performs more accurate than Algorithms 2 and 3. But, in the view of* (2.17)*, the second order scheme* $2p^n - p^{n-1}$ *includes* $2s^n - s^{n-1}$ *which is the same as in Algorithm 2, so this algorithm also requests the same stability constraint* (2.12) *as in Algorithm 2.*

3. THE θ TIME DISCRETE GAUGE-UZAWA METHOD

Algorithms 2 and 4 request high computational cost due to the stability condition (2.12). In order to solve this difficulty, we observe the fact that the applicable τ depends on the amount of $\|s_h^n\|_0^2$ and so we do not need to impose so small τ if $\|s_h^n\|_0^2$ is small enough. Since we can calculate $\|\nabla s^n\|_0$ after finishing n-th step computation, we can determine the time marching size of $(n+1)$-th step by using (2.12). Because BDF2 scheme is a 2 step method, we need to impose an adaptive technique which is called θ scheme to use non-uniform time marching method. We define θ scheme with standard notations. Let $t^1, t^2, \cdots, t^n, \cdots$ be node points on time and let $\tau^n = t^n - t^{n-1}$ and $\tau^{n+1} = \frac{M}{\mu^2\|\nabla s^n\|_0^2}$, where M is a fixed positive real number. Then τ^{n+1}, for all n, satisfies (2.12). We denote $\theta := \frac{\tau^{n+1}}{\tau^n}$ as a rate indicator of variation of time step size. In addition we note that 2 step extrapolation on non-uniform grid is $z^{n+1} = (1+\theta)z^n - \theta z^{n-1}$, for any function z. Then (2.2) becomes, for the case of non-uniform mesh on time,

$$\widehat{\mathbf{u}}^{n+1} = \mathbf{u}^{n+1} - \nabla\left(\phi^{n+1} - (1+\theta)\phi^n + \theta\phi^{n-1}\right).$$

Finally, we arrive at the following θ-scheme by elementary numerical technique.

Algorithm 5 (The θ-scheme of the classical Gauge-Uzawa method)**.** *Set initial values using a first-order Gauge-Uzawa method with a suitable $\tau = \tau^*$ and $\phi^0 = 0$. And choose a positive real number M. Start the following iteration with $n = 1$ until to reach terminal time T.*

Step 1: *Define $\tau = t^n - t^{n-1}$ and $\tau^{n+1} = \frac{M}{\mu^2\|\nabla s^n\|_0^2}$. Set*

$$\theta = \frac{\tau^{n+1}}{\tau}. \tag{3.1}$$

If $\tau^{n+1} \geq \tau^*$, then set $\tau^{n+1} = \tau^*$. And set $t^{n+1} = t^n + \tau^{n+1}$.

Step 2: *Set $\mathbf{u}^* = (1+\theta)\mathbf{u}^n - \theta\mathbf{u}^{n-1}$ and find $\widehat{\mathbf{u}}^{n+1}$ as the solution of*

$$\frac{(2\theta+1)\widehat{\mathbf{u}}^{n+1} - (\theta+1)^2\mathbf{u}^n + \theta^2\mathbf{u}^{n-1}}{\theta(\theta+1)\tau} + (\mathbf{u}^*\cdot\nabla)\widehat{\mathbf{u}}^{n+1} - \mu\triangle\widehat{\mathbf{u}}^{n+1}$$
$$-\nabla\left(\frac{\phi^n - \phi^{n-1}}{\tau} - \mu\left((\theta+1)s^n - \theta s^{n-1}\right)\right) = \mathbf{f}(t^{n+1}),$$
$$\widehat{\mathbf{u}}^{n+1}|_\Gamma = \mathbf{0}.$$

Step 3: *Find ϕ^{n+1} as the solution of*

$$-\triangle\phi^{n+1} = -\triangle\left((\theta+1)\phi^n - \theta\phi^{n-1}\right) + \nabla\cdot\widehat{\mathbf{u}}^{n+1},$$
$$\partial_{\boldsymbol{\nu}}\phi^{n+1}|_\Gamma = 0.$$

Step 4: *Update* $\mathbf{u}^{n+1}$ *and* s^{n+1} *by*

$$\mathbf{u}^{n+1} = \widehat{\mathbf{u}}^{n+1} + \nabla\left(\phi^{n+1} - (\theta+1)\phi^{n} + \theta\phi^{n-1}\right),$$
$$s^{n+1} = (\theta+1)s^{n} - \theta s^{n-1} - \nabla\cdot\widehat{\mathbf{u}}^{n+1}.$$

Pressure: *Compute pressure whenever necessary as*

$$p^{n+1} = -\frac{(2\theta+1)\phi^{n+1} - (\theta+1)^2\phi^{n} + \theta^2\phi^{n-1}}{\theta(\theta+1)\tau} + \mu s^{n+1}.$$

4. Proof of stabilities

In this section, we prove the Theorems 1. We start to prove main theorem via the following well-known lemmas in [3].

Lemma 4.1 (Orthogonality between divergence free function and curl free function)**.** *Let* $\mathbf{v}\in\mathbf{H}^1(\Omega)$ *and* $q\in\mathbf{L}^2(\Omega)$*. If* $\nabla\cdot\mathbf{u}=0$ *and* $\mathbf{u}\cdot\boldsymbol{\nu}=0$ *on* $\partial\Omega$*, then*

$$\langle\mathbf{u}\,,\,\nabla q\rangle = 0.$$

Lemma 4.2 (Inner product with convection term)**.** *Let* $\mathbf{u},\mathbf{v},\mathbf{w}\in\mathbf{H}^1(\Omega)$ *and* $\nabla\cdot\mathbf{u}=0$*. If*

$$\mathbf{u}\cdot\boldsymbol{\nu}=0 \quad \text{or} \quad \mathbf{v}=\mathbf{0} \quad \text{on } \partial\Omega,$$

then

$$\int_\Omega(\mathbf{u}\cdot\nabla)\mathbf{v}\cdot\mathbf{w}d\mathbf{x} = -\int_\Omega(\mathbf{u}\cdot\nabla)\mathbf{w}\cdot\mathbf{v}d\mathbf{x}$$

and

$$\int_\Omega(\mathbf{u}\cdot\nabla)\mathbf{v}\cdot\mathbf{v}d\mathbf{x} = 0.$$

The following elementary but crucial relation is derived in [8, 9].

Lemma 4.3 (div-grad relation)**.** *If* $\mathbf{v}\in\mathbf{H}^1_0(\Omega)$*, then*

$$\|\nabla\cdot\mathbf{v}\|_0 \le \|\nabla\mathbf{v}\|_0.$$

We now introduce a well know lemma to treat time derivative terms.

Lemma 4.4 (Inner product of time derivative terms)**.** *For any sequence* $\{z\}_{n=0}^N$*, we have*

$$\begin{aligned}2\left\langle 3z^{n+1}-4z^{n}+z^{n-1}\,,\,z^{n+1}\right\rangle &= \left\|z^{n+1}\right\|_0^2 + \left\|2z^{n+1}-z^{n}\right\|_0^2\\ &+\left\|z^{n+1}-2z^{n}+z^{n-1}\right\|_0^2 - \left\|z^{n}\right\|_0^2 - \left\|2z^{n}-z^{n-1}\right\|_0^2,\end{aligned} \tag{4.1}$$

$$2\left\langle z^{n+1}-z^{n}\,,\,z^{n+1}\right\rangle = \left\|z^{n+1}\right\|_0^2 - \left\|z^{n}\right\|_0^2 + \left\|2z^{n+1}-z^{n}\right\|_0^2, \tag{4.2}$$

and

$$2\left\langle z^{n+1} - z^n\,,\, z^n\right\rangle = \left\|z^{n+1}\right\|_0^2 - \|z^n\|_0^2 - \left\|2z^{n+1} - z^n\right\|_0^2. \tag{4.3}$$

We now start to prove stability of Algorithm 2 with rewriting (2.8) by using (2.10) as

$$\begin{aligned}
&\frac{3\mathbf{u}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\tau} + (\mathbf{u}^* \cdot \nabla)\widehat{\mathbf{u}}^{n+1} - \mu\triangle\widehat{\mathbf{u}}^{n+1} \\
&- \nabla\left(\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\tau} - \mu\left(2s^n - s^{n-1}\right)\right) = \mathbf{f}(t^{n+1}).
\end{aligned}$$

We now multiply by $4\tau\widehat{\mathbf{u}}^{n+1}$ and use (4.1) to get

$$\begin{aligned}
&\left\|\mathbf{u}^{n+1}\right\|_0^2 + \left\|2\mathbf{u}^{n+1} - \mathbf{u}^n\right\|_0^2 + \left\|\mathbf{u}^{n+1} - 2\mathbf{u}^n + \mathbf{u}^{n-1}\right\|_0^2 \\
&\quad -\|\mathbf{u}^n\|_0^2 - \left\|2\mathbf{u}^n - \mathbf{u}^{n-1}\right\|_0^2 + 4\tau\mu\left\|\nabla\widehat{\mathbf{u}}^{n+1}\right\|_0^2 = \sum_{i=1}^{3} A_i,
\end{aligned} \tag{4.4}$$

where

$$\begin{aligned}
A_1 &= 2\left\langle\nabla\left(3\phi^{n+1} - 4\phi^n + \phi^{n-1}\right),\, \widehat{\mathbf{u}}^{n+1}\right\rangle, \\
A_2 &= 4\tau\mu\left\langle 2s^n - s^{n-1}\,,\, \nabla\cdot\widehat{\mathbf{u}}^{n+1}\right\rangle, \\
A_3 &= 4\tau\left\langle\mathbf{f}(t^{n+1})\,,\, \widehat{\mathbf{u}}^{n+1}\right\rangle.
\end{aligned}$$

We note here that the convection term is vanished by Lemma 4.2. In conjunction with (2.10) and (4.2), Lemma 4.1 yields

$$\begin{aligned}
A_1 =& -2\left\langle\nabla\left(3\phi^{n+1} - 4\phi^n + \phi^{n-1}\right),\, \nabla\left(\phi^{n+1} - 2\phi^n + \phi^{n-1}\right)\right\rangle \\
=& -2\left\|\nabla\left(\phi^{n+1} - \phi^n\right)\right\|_0^2 + 2\left\|\nabla\left(\phi^n - \phi^{n-1}\right)\right\|_0^2 - 4\left\|\nabla\left(\phi^{n+1} - 2\phi^n + \phi^{n-1}\right)\right\|_0^2.
\end{aligned}$$

In light of the second equation in (2.10), A_2 becomes

$$\begin{aligned}
A_2 =& -4\tau\mu\left\langle 2s^n - s^{n-1}\,,\, s^{n+1} - 2s^n - s^{n-1}\right\rangle \\
=& -4\tau\mu\left\langle s^n - s^{n-1}\,,\, s^{n+1} - 2s^n - s^{n-1}\right\rangle - 4\tau\mu\left\langle s^n\,,\, s^{n+1} - 2s^n - s^{n-1}\right\rangle \\
=& A_{2,1} + A_{2,2}.
\end{aligned}$$

If we use Lemma 4.3, we can readily get

$$\left\|s^{n+1} - 2s^n + s^{n-1}\right\|_0^2 = \left\|\nabla\cdot\widehat{\mathbf{u}}^{n+1}\right\|_0^2 \leq \left\|\nabla\widehat{\mathbf{u}}^{n+1}\right\|_0^2$$

and use (4.3) to deduce

$$\begin{aligned}
A_{2,1} &= -2\tau\mu\left(\left\|s^{n+1} - s^n\right\|_0^2 - \left\|s^n - s^{n-1}\right\|_0^2 - \left\|s^{n+1} - 2s^n + s^{n-1}\right\|_0^2\right) \\
&\leq -2\tau\mu\left(\left\|s^{n+1} - s^n\right\|_0^2 - \left\|s^n - s^{n-1}\right\|_0^2\right) + 2\tau\mu\left\|\nabla\widehat{\mathbf{u}}^{n+1}\right\|_0^2.
\end{aligned}$$

We use (2.10) and (2.9) and then apply integral by parts to obtain

$$\begin{aligned} A_{2,2} &= -4\tau\mu \left\langle s^n \, , \, s^{n+1} - 2s^n - s^{n-1} \right\rangle \\ &= 4\tau\mu \left\langle s^n \, , \, \nabla \cdot \widehat{\mathbf{u}}^{n+1} \right\rangle \\ &= -4\tau\mu \left\langle s^n \, , \, \triangle \left(\phi^{n+1} - 2\phi^n - \phi^{n-1} \right) \right\rangle \\ &\le C\tau^2\mu^2 \|\nabla s^n\|_0^2 + \left\| \nabla \left(\phi^{n+1} - 2\phi^n - \phi^{n-1} \right) \right\|_0^2. \end{aligned}$$

In other hand, Hölder inequality yields

$$A_3 \le C\frac{\tau}{\mu} \left\| \mathbf{f}(t^{n+1}) \right\|_{-1}^2 + \tau\mu \left\| \nabla \widehat{\mathbf{u}}^{n+1} \right\|_0^2.$$

Inserting above estimates into (4.4) derives

$$\begin{aligned} &\left\| \mathbf{u}^{n+1} \right\|_0^2 + \left\| 2\mathbf{u}^{n+1} - \mathbf{u}^n \right\|_0^2 + \left\| \mathbf{u}^{n+1} - 2\mathbf{u}^n + \mathbf{u}^{n-1} \right\|_0^2 - \left\| 2\mathbf{u}^n - \mathbf{u}^{n-1} \right\|_0^2 - \|\mathbf{u}^n\|_0^2 \\ &+ 2 \left(\left\| \nabla \left(\phi^{n+1} - \phi^n \right) \right\|_0^2 - \left\| \nabla \left(\phi^n - \phi^{n-1} \right) \right\|_0^2 \right) + 3 \left\| \nabla \left(\phi^{n+1} - 2\phi^n + \phi^{n-1} \right) \right\|_0^2 \\ &+ \tau\mu \left\| \nabla \widehat{\mathbf{u}}^{n+1} \right\|_0^2 + 2\tau\mu \left(\left\| s^{n+1} - s^n \right\|_0^2 - \left\| s^n - s^{n-1} \right\|_0^2 \right) \\ &\le C\frac{\tau}{\mu} \left\| \mathbf{f}(t^{n+1}) \right\|_{-1}^2 + C\tau^2\mu^2 \|\nabla s^n\|_0^2. \end{aligned}$$

In light of (2.12), we obtain

$$\tau^2\mu^2 \sum_{n=1}^{N} \|\nabla s^n\|_0^2 \le \tau\mu^2 \max_{n=1}^{N} \|\nabla s^n\|_0^2 \le M,$$

and so we arrive at (2.13) via summing for n from 1 to N.

5. Numerical experiments

In this section, we carried out numerical experiments to compare to theoretical results. We perform with known solutions to check accuracy and compute driven cavity flows to check stability.

5.1. Mesh analysis with polynomial example. The first example is performance with known exact solution which is imposed forcing term the exact solution to become

$$\begin{aligned} u &= \cos(t) \left(x^2 - 2x^3 + x^4 \right) \left(2y - 6y^2 + 4y^3 \right), \\ v &= -\cos(t) \left(y^2 - 2y^3 + y^4 \right) \left(2x - 6x^2 + 4x^3 \right), \\ p &= \cos(t) \left(x^2 + y^2 - \frac{2}{3} \right). \end{aligned} \tag{5.1}$$

Tables 1, 2, and 3 are mesh analysis of Algorithms 2, 3, and 4, respectively. We impose $\mu = 1$ and $\tau = h$ in these tests. We define notations $\mathbf{E}$ and e as error functions for velocity and pressure, respectively.

h	1/16	1/32	1/64	1/128	1/256
$\|\mathbf{E}\|_0$	9.83161e-05	3.05559e-05	8.13503e-06	2.01614e-06	5.01679e-07
	Order	1.685977	1.909231	2.012552	2.006759
$\|\mathbf{E}\|_{L^\infty}$	0.000926506	7.64737e-05	1.98964e-05	4.93618e-06	1.231e-06
	Order	3.598765	1.942456	2.011041	2.003564
$\|\mathbf{E}\|_1$	0.00581341	0.00048326	0.000119246	2.94906e-05	7.41107e-06
	Order	3.588513	2.018859	2.015614	1.992501
$\|e\|_0$	0.00969686	0.000591138	0.000140191	3.43644e-05	8.54948e-06
	Order	4.035951	2.076101	2.028407	2.007006
$\|e\|_{L^\infty}$	0.3328	0.0193977	0.00225387	0.000559215	0.000155937
	Order	4.100698	3.105409	2.010929	1.842440

TABLE 1. Error decay of Algorithm 2 with exact solution (5.1).

h	1/16	1/32	1/64	1/128	1/256
$\|\mathbf{E}\|_0$	0.000766413	0.000231018	6.42159e-05	1.70043e-05	4.38265e-06
	Order	1.730117	1.847003	1.917031	1.956024
$\|\mathbf{E}\|_{L^\infty}$	0.00197285	0.000618625	0.000180551	5.0349e-05	1.36219e-05
	Order	1.673144	1.776659	1.842371	1.886035
$\|\mathbf{E}\|_1$	0.0123447	0.00406798	0.00126886	0.000382078	0.000112936
	Order	1.601507	1.680780	1.731594	1.758362
$\|e\|_0$	0.0145993	0.0047309	0.00146086	0.000430436	0.000123065
	Order	1.625713	1.695297	1.762947	1.806378
$\|e\|_{L^\infty}$	0.207157	0.0865286	0.0379158	0.0156967	0.00626727
	Order	1.259476	1.190378	1.272338	1.324552

TABLE 2. Error decay of Algorithm 3 with exact solution (5.1).

In Table 1, the classical GUM which is Algorithm 2 performs 2nd order accuracy for most of spatial error, but the error in $L^\infty(0, 1 : L^\infty(\Omega))$ displays perturbation. So we perform other example in §5.2. Because this oscillation disappear in Table 1 which is the test with (5.2), we expect that the error in $L^\infty(0, 1 : L^\infty(\Omega))$ will also converge to 2 and that the accuracy is fully 2nd order in all spaces.

h	1/16	1/32	1/64	1/128	1/256
$\Vert\mathbf{E}\Vert_0$	4.8541e-05	5.69019e-06	6.71056e-07	8.76035e-08	1.12158e-08
	Order	3.092655	3.083972	2.937373	2.965456
$\Vert\mathbf{E}\Vert_{L^\infty}$	0.000396898	3.33674e-05	1.86666e-06	2.59669e-07	4.14608e-08
	Order	3.572257	4.159908	2.845713	2.646854
$\Vert\mathbf{E}\Vert_1$	0.00193225	0.000207701	1.57598e-05	3.00764e-06	1.19194e-06
	Order	3.217702	3.720187	2.389545	1.335320
$\Vert e\Vert_0$	0.00308668	0.000337233	1.44083e-05	2.24362e-06	3.33836e-07
	Order	3.194238	4.548774	2.683000	2.748617
$\Vert e\Vert_{L^\infty}$	0.101547	0.021122	0.00017087	7.51221e-05	1.44142e-05
	Order	2.265329	6.949703	1.185590	2.381747

TABLE 3. Error decay of Algorithm 4 with exact solution (5.1).

Table 2 is the error decay of Algorithm 3. The error for velocity in $L^\infty(0,1:\mathbf{L}^2(\Omega))$ converges to order 2, and it consists to theoretical result in Lemma 2.2. But errors in other spaces seem to lose order as well as in Table 5 which is the test with (5.2). So we conclude that Algorithm 3 is unconditionally stable, but includes oscillation except in $L^\infty(0,1:\mathbf{L}^2(\Omega))$.

Table 3 is the error decay of Algorithm 4. As mentioned in Remark 2.3, we can see overconvergence for Algorithm 4. This is due to higher order extrapolation of pressure. Algorithm 4 performs better accuracy than other, but this method includes rather big oscillations.

5.2. **Mesh analysis with triangular functions.** Because the test in §5.1 includes abnormal symptom, we carry out numerical test again with the same example in [4]:

$$\begin{aligned} u =&\pi\sin(t)\sin(2\pi y)\sin^2(\pi x),\\ v =&-\sin(t)\sin(2\pi x)\sin^2(\pi y),\\ p =&-\sin(t)\cos(\pi x)\sin(\pi y). \end{aligned} \tag{5.2}$$

Tables 4, 5, and 6 are mesh analysis of Algorithms 2, 3, and 4, respectively. We also impose the same conditions as in §5.1.

Table 4 is the numerical result for the classical GUM which is Algorithm 2. In contrast in Table 1, the convergence rates for all errors are 2. In Table 5, the velocity error in $L^\infty(0,1:\mathbf{L}^2(\Omega))$ for Algorithm 3 converges to 0 with rate 2 and this result consists to Table 2 and Lemma 2.2. For the numerical results Table 2 and Table 5, we conclude that Algorithm 3 lose accuracy for other errors. However, this algorithm is unconditionally stable in [12].

h	1/16	1/32	1/64	1/128	1/256
$\|\mathbf{E}\|_0$	0.000374286	5.83689e-05	1.14962e-05	2.65413e-06	6.50019e-07
	Order	2.680869	2.344043	2.114846	2.029685
$\|\mathbf{E}\|_{L^\infty}$	0.000835976	0.000166414	3.70314e-05	8.70111e-06	2.11636e-06
	Order	2.328685	2.167956	2.089478	2.039614
$\|\mathbf{E}\|_1$	0.0532202	0.0133404	0.00333751	0.000834537	0.000208653
	Order	1.996172	1.998958	1.999724	1.999870
$\|e\|_0$	0.00220608	0.000464423	0.000115444	2.90385e-05	7.28256e-06
	Order	2.247974	2.008246	1.991154	1.995449
$\|e\|_{L^\infty}$	0.0200976	0.00145889	0.00035159	8.72965e-05	2.18391e-05
	Order	3.784080	2.052905	2.009898	1.999010

TABLE 4. Error decay of Algorithm 2 with exact solution (5.2).

h	1/16	1/32	1/64	1/128	1/256
$\|\mathbf{E}\|_0$	0.000666171	0.000171224	4.67914e-05	1.26051e-05	3.30152e-06
	Order	1.960008	1.871570	1.892236	1.932805
$\|\mathbf{E}\|_{L^\infty}$	0.00164411	0.000487813	0.000161503	4.89765e-05	1.39134e-05
	Order	1.752907	1.594767	1.721399	1.815615
$\|\mathbf{E}\|_1$	0.053935	0.0136668	0.00347407	0.000886726	0.00022716
	Order	1.980546	1.975977	1.970067	1.964780
$\|e\|_0$	0.00578956	0.00229765	0.000860856	0.000292507	9.15238e-05
	Order	1.333295	1.416315	1.557301	1.676252
$\|e\|_{L^\infty}$	0.061034	0.0380297	0.0183978	0.00794517	0.00322284
	Order	0.682487	1.047593	1.211383	1.301746

TABLE 5. Error decay of Algorithm 3 with exact solution (5.2).

An remarkable result is Table 6. Big oscillations in Table 3 are disappeared in Table 6. But the difference for errors between Algorithms 2 and 4 also reduced in this test. So we can conclude that Algorithm 4 is more accurate than others, but that this method includes perturbations.

5.3. **Driven cavity flow.** The goal of this section is to check stability of Algorithm 3 via computing the driven cavity flow. In order to assert that Algorithm 3 is unconditionally stable, we perform this test under extremely weak stable conditions: $\mu = 1/10,000$, $h = 1/256$, and $\tau = 0.5$. In general, τ has to be less than h to make hold stable condition. The figure 1 is the numerical result of Algorithm 3. Even

h	1/16	1/32	1/64	1/128	1/256
$\|\mathbf{E}\|_0$	0.000337692	4.3098e-05	5.61456e-06	7.92313e-07	1.34147e-07
	Order	2.970015	2.940376	2.825031	2.562256
$\|\mathbf{E}\|_{L^\infty}$	0.00047177	6.12214e-05	7.979e-06	1.20303e-06	2.15262e-07
	Order	2.945976	2.939756	2.729535	2.482507
$\|\mathbf{E}\|_1$	0.0531745	0.0133294	0.00333457	0.000833776	0.00020846
	Order	1.996123	1.999039	1.999769	1.999889
$\|e\|_0$	0.000582949	0.000128133	2.64715e-05	5.60035e-06	1.23071e-06
	Order	2.185728	2.275130	2.240851	2.186026
$\|e\|_{L^\infty}$	0.00542067	0.00191244	0.000394239	9.06402e-05	1.89593e-05
	Order	1.503057	2.278272	2.120848	2.257245

TABLE 6. Error decay of fixed Algorithm 4 with exact solution (5.2).

though we impose very big $\tau = 0.5$, it is still stable, we conclude Algorithm 3 is unconditionally stable. The figure 1 includes lots of oscillation and it is due to big

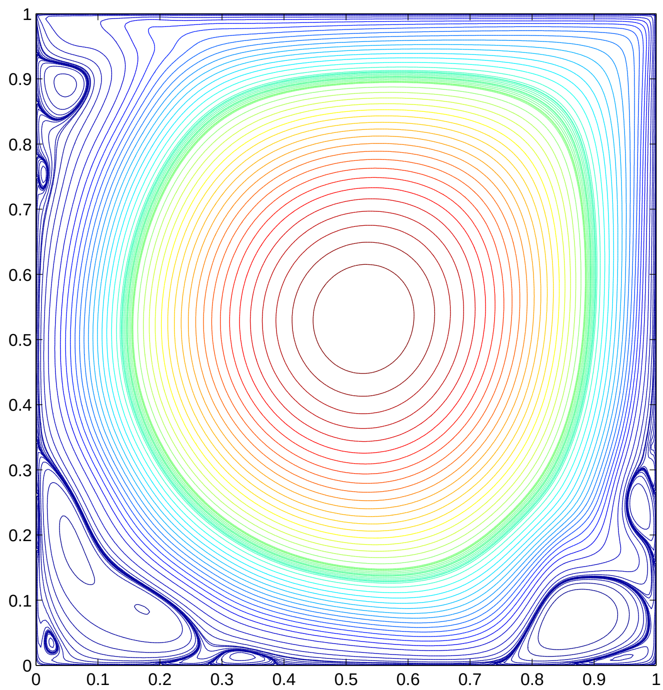

FIGURE 1. Driven cavity for 3 with $\mu = 1/10,000$, $h = 1256$, $\tau = 0.5$.

$\tau = 0.5$.

In contrast Algorithm 3, Algorithms 2 and 4 have to hold the stability constraint (2.12) and so requires very small τ for high viscosity problems. Because the stability condition (2.12) depends on H^1-norm of s^n which is a part of pressure, the range of stability also depends on examples. So the applicability of both algorithms is limited only for simple problems. Therefore we perform the test with Algorithm 5 which is an adaptive method for time. Even though the time marching size τ on Algorithm 5 is adjusted to make hold (2.12) at each step, the algorithm is not free from the constraint because of $\tau \to 0$ as $\nabla s^n \to \infty$. We have tried to impose high Reynolds numbers as big as possible, and the biggest number is only around $\mathbf{u} = 1/1,000$ due to the time consuming. We note that the determinator equivalent to the stability constraint (2.12), and so $\tau^* = \min_n\{\tau^n = t^n - t^{n-1}\}$. Figure 2 is simulation of driven cavity flow for

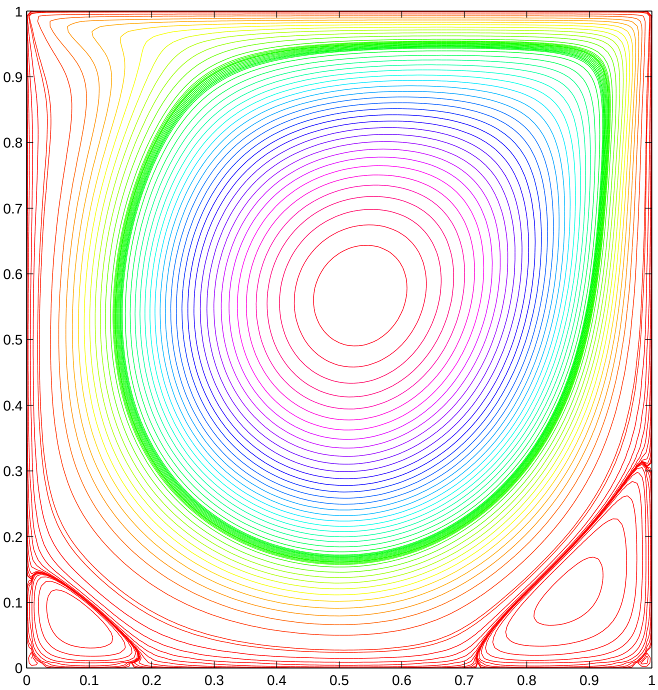

FIGURE 2. Driven cavity for 5 with $\mu = 1/1,000$ and $h = 1/128$

Algorithm 5 with $\mu = 1/1,000$, $M = 0.5$, and $h = 1/128$. The initial time step size is given $\tau^1 = 0.5$, τ^n for all $1 < n < N$ is computed by (3.1), and τ^* is determined 0.008.

Remark 5.1 (The role of M). *In Figure 2, numerical result and stability do not depend much on the number of M, if $M \leq 5,000$. But this scheme becomes unstable for $M \geq 5,000$. We fix $M = 0.5$ which is the same as $\tau^1 = 0.5$ in this test. If we impose bigger M, time marching size increases little bit in a few initial steps and then return to the similar step size to the case of $M = 0.5$.*

6. Conclusion

The classical GUM Algorithm 2 has been considered as one of the most accurate method and displays fully 2nd order error decay at the tests in §5.1 and §5.2. But the method suffers from rather strong stability constraint (2.12) which is proved in Theorem 1. In order to overcome the strong condition, we construct Algorithm 5 which is an adaptive method for time to make hold the condition (2.12) at each time step. We obtain a driven cavity flows in Figure 2, and so we conclude that the condition (2.12) is correct for Algorithm 2 because both Algorithms 2 and 5 are equivalent for the uniform mesh on time. However, the numerical result in Figure 2 is only for the case $\mu = 1/1,000$ which is not big enough to apply to real industrial problems.

In contrast Algorithm 2, the stabilized GUM Algorithm 3 is unconditionally stable scheme and it is verified by numerical test in Figure 1. But we found out that the algorithm lose convergence order at the tests in §5.1 and §5.2.

Algorithm 4 which is newly constructed scheme displays the best accurate performance in §5.1 and §5.2, but the scheme includes perturbation on the tests and requires the strong stability constraint (2.12) as we mentioned in Remark 2.3.

Therefore we conclude that the most reliable method is the classical GUM Algorithm 2, Algorithm 3 is unconditionally stable, and the most accurate method is 4.

References

[1] A.J. Chorin, *Numerical solution of the Navier-Stokes equations*, Math. Comp., 22 (1968) 745–762.

[2] P. Constantin and C. Foias, *Navier-Stokes Equations*, The University of Chicago Press (1988).

[3] V. Girault, and P.A. Raviart, *Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag (1986).

[4] J.L. Guermond, P. Minev, and Jie Shen. *An overview of projection methods for incompressible flows*, Comput. Methods Appl. Mech. Engrg., 195 (2006), 6011–6045.

[5] J.L. Guermond and J. Shen *On the error estimates of rotational pressure-correction projection methods*, Math. Comp., 73 (2004), 1719-1737.

[6] J.G. Heywood and R. Rannacher, *Finite element approximation of the non-stationary Navier-stokes problem. I. regularity of solutions and second-order error estimates for spatial discretization*, SIAM J. Numer. Anal., 19 (1982), 275–311.

[7] R.B. Kellogg and J.E. Osborn, *A regularity result for the stokes problems in a convex polygon*, J. Funct. Anal., 21 (1976), 397–431.

[8] R.H. Nochetto and J.-H. Pyo, *Error estimates for semi-discrete gauge methods for the Navier-Stokes equations*, Math. Comp., 74, (2005), 521–542.

[9] R.H. Nochetto and J.-H. Pyo, *A finite element Gauge-Uzawa method. Part I : the Navier-Stokes equations*, SIAM J. Numer. Anal., 43, (2005), 1043–1068.

[10] R.H. Nochetto and J.-H. Pyo, *A finite element Gauge-Uzawa method. Part II : Boussinesq Equations*, Math. Models Methods Appl. Sci., 16, (2006), 1599–1626.

[11] A. Prohl, *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations*, B.G.Teubner Stuttgart (1997).

[12] J.-H. Pyo, *Error estimates for the second order semi-discrete stabilized Gauge-Uzawa method for the Navier-Stokes equations*, to appear.

[13] J.-H. Pyo and J. Shen, *Normal mode analysis of second-order projection methods for incompressible flows*, Discrete Contin. Dyn. Syst. Ser. B, 5, (2005), 817–840.

[14] J.-H. Pyo and J. Shen, *Gauge Uzawa methods for incompressible flows with Variable Density*, J. Comput. Phys., 211, (2007), 181–197.

[15] R. Temam, *Sur l'approximation de la solution des equations de Navier-Stokes par la methode des pas fractionnaires. II*, Arch. Rational Mech. Anal., 33 (1969), 377–385.