Some experiments with adaptive penalty methods in the numerical solution of the incompressible Navier-Stokes equations

by

Astrid Berge

BS, University of Washington, 2018

Submitted to the Graduate Faculty of

the Dietrich School of Arts & Sciences in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2020

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS & SCIENCES

This thesis was presented

by

Astrid Berge

It was defended on

December 16, 2020

and approved by

Professor William Layton, Department of Mathematics

Professor Michael Neilan, Department of Mathematics

Professor Catalin Trenchea, Department of Mathematics

Some experiments with adaptive penalty methods in the numerical solution of the incompressible Navier-Stokes equations

Astrid Berge, M.S.

University of Pittsburgh, 2020

This paper presents and tests an adaptive scheme for the penalty method. First, the penalty method is introduced as an optimization method and then applied to the Navier-Stokes equations. The energy equation, proof of stability and consistency error of the penalty method is given. Some computational tests of the penalty method in FEniCS are presented, showing the first-order convergence, with plots of the error. A sample code to recreate the results is included. Next, the idea of an adaptive method is discussed and presented in the case of the penalty method, adapted from a recent paper [6]. The energy equation and inequality are given, showing stability. Numerical experiments in FEniCS are presented for a fixed timestep and varying ε, as well as a test of the doubly adaptive scheme.

Table of contents

List of tables

List of figures

1.0 Introduction

The incompressible Navier-Stokes equations are a set of nonlinear partial differential equations in fluid mechanics, describing the evolution of a velocity field of a fluid as a function of space and time. For a region $\Omega \subset \mathbb{R}^3$, where $x \in \Omega$ and $0 < t \leq T$, with given fluid viscosity ν, velocity $u(x,t)$, pressure $p(x,t)$ and body force $f(x,t)$, the Navier-Stokes equations read

$$u_t + u \cdot \nabla u - \nu \Delta u + \nabla p = f(x,t)$$
$$\nabla \cdot u = 0$$

with the following no-slip boundary condition,

$$u = 0 \text{ on } \partial\Omega, \int_\Omega p dx = 0 \text{ for } 0 < t \leq T$$

The first equation (1) of the Navier-Stokes equations is called the *momentum equation* while the second equation (2) gives the incompressibility constraint. The momentum equation describes conservation of *linear momentum* and the second equation describes the conservation of mass.

1.1 Computational setup

The semi-discretized NSE suppresses the spatial discretization and replaces the time derivative with a backward Euler approximation:

$$\frac{u^{n+1} - u^n}{\Delta t} + u^n \cdot \nabla u^{n+1} - \nu \Delta u^{n+1} - \nabla p^{n+1} = f^{n+1} \tag{1}$$

$$\nabla \cdot u^{n+1} = 0 \tag{2}$$

Now, solving the NSE is equivalent to solving a sequence of n linear systems. In order to use the Finite Element Method in FEniCS [1] to solve the NSE, it is necessary to derive the variational formulation.

Define

$$X = H_0^1(\Omega) = \{u \in L^2 : \nabla u \in L^2, u|_{\partial\Omega} = 0\}$$
$$Q = L_0^2(\Omega) = \{p \in L^2 : \int_\Omega p dx = 0\}$$

Let $(v, q) \in (X, Q)$ be arbitrary. We take an inner product of (1) with v and multiply (2) by q and integrate over Ω, applying integration by parts to terms with second-order derivatives:

$$\int_\Omega u_t \cdot v dx + \int_\Omega (u \cdot \nabla u) \cdot v dx + \nu \int_\Omega \nabla u : \nabla v dx - \int_\Omega p(\nabla \cdot v) dx = \int_\Omega f \cdot v dx$$
$$\int_\Omega (\nabla \cdot u) q dx = 0$$

Then we seek u, p such that the above equations hold for all $v \in X, q \in Q$. Applied to the semi-discrete problem,

$$\int_\Omega \frac{u^{n+1} - u^n}{\Delta t} \cdot v dx + \int_\Omega \left(u^{n+1} \cdot \nabla u^{n+1}\right) \cdot v dx + \nu \int_\Omega \nabla u^{n+1} : \nabla v dx$$
$$- \int_\Omega p^{n+1}(\nabla \cdot v) dx = \int_\Omega f^{n+1} \cdot v dx, \text{ for all } v \in X$$
$$\int_\Omega \left(\nabla \cdot u^{n+1}\right) \cdot q = 0, \text{ for all } q \in Q$$

1.1.1 Necessary bounds and inequalities

For $u, v, w, \in X$, define the trilinear form $b^*(u, v, w) := \frac{1}{2}(u \cdot \nabla v, w) - \frac{1}{2}(u \cdot \nabla w, v)$. This trilinear form is skew-symmetric, so that

$$b^*(u, v, w) = -b^*(u, w, v)$$

The following lemma is from [2].

Lemma 1. For any $u, v, w \in X$,

$$b^*(u, v, w) = \int_\Omega u \cdot \nabla v \cdot w dx + \frac{1}{2} \int_\Omega (\nabla \cdot u) v \cdot w dx$$

For any $u, v \in X$, $b^*(u, v, v) = 0$. For $M = \sup_{u,v,w \in X} \frac{(u \cdot \nabla v, w)}{||\nabla u|| ||\nabla v|| ||\nabla w||}$, we have $b^*(u, v, w) \leq M \|\nabla u\| \|\nabla v\| \|\nabla w\|$.

The following two items are from [5].

Theorem 1 (Young's inequality)**.** For any $a, b \geq 0$ and $p, q > 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$ then

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}$$

In particular for $p = q = 2$,

$$ab \leq \frac{a^2}{2} + \frac{b^2}{2}$$

Definition 1 (dual norm)**.** The *dual space* of X, $X^* = H^{-1}(\Omega)$, is the closure of $L^2(\Omega)$ in $\|\cdot\|_{-1}$ where

$$\|f\|_{-1} := \sup_{v \in X} \frac{(f, v)}{\|\nabla v\|}$$

2.0 Penalty methods for the NSE

2.1 Definition and formulation

A *penalty method* is a method for solving a constrained optimization problem

$$\text{minimize } f(x)$$
$$\text{subject to } x \in S$$

for some $f(x)$ continuous, where S is a constraint set [8]. A penalty method converts this problem to an unconstrained problem by adding a penalty term $cP(x)$ for a given $c > 0$:

$$\text{minimize } f(x) + cP(x)$$

where $P(x)$ is continuous, nonnegative, and $P(x) = 0$ exactly when $x \in S$ [8]. The penalty method penalizes solutions that violate the constraint, so that an optimal solution x will be in a region where $P(x)$ is small.

In the case of the NSE, we seek to minimize a related energy functional subject to the incompressibility constraint [9]. The penalty method for solving the Navier-Stokes equations belongs to the broader class of artificial compressibility methods, along with other methods such as the artificial compression method and the projection method [10]. The incompressibility constraint

$$\nabla \cdot u = 0$$

creates a coupling of the pressure and velocity terms, which makes the system more complicated to solve numerically. The penalty method alters the incompressibility constraint, adding an εp term for some given, typically small, ε:

$$u_t + u \cdot \nabla u - \nu \Delta u + \nabla p = f^{n+1} \tag{3}$$

$$\varepsilon p + \nabla \cdot u = 0 \tag{4}$$

The variational formulation for the penalty problem is given as follows: Find $(u, p) \in (X, Q)$ such that for any $(v, q) \in (X, Q)$, the equations

$$\int_\Omega u_t \cdot v dx + \int_\Omega (u \cdot \nabla u) \cdot v dx + \nu \int_\Omega \nabla u : \nabla v dx - \int_\Omega p(\nabla \cdot v) dx = \int_\Omega f \cdot v dx$$
$$\varepsilon \int_\Omega pq dx + \int_\Omega (\nabla \cdot u) q dx = 0$$

hold.

The variational formulation may be equivalently derived in another way. Using the second equation (4), we now have the pressure p in terms of the velocity u:

$$p = -\frac{1}{\varepsilon} \nabla \cdot u$$

Now, we can substitute $p = -\frac{1}{\varepsilon}\nabla \cdot u$ into (3) and then find the variational formulation by integrating over Ω and applying integration by parts to terms with second-order derivatives. The following definition is from [9]. Define

$$H^1_\varepsilon(\Omega) := \{u : u \in H^1_0(\Omega), ||u||_{1,\varepsilon} = ||u||_1 + \frac{1}{\varepsilon} \int_\Omega (\nabla \cdot u)(\nabla \cdot u) dx\}$$

where $|| \cdot ||_1$ is the norm in $X = H^1(\Omega)$. Now we seek $u \in H^1_\varepsilon(\Omega)$ such that:

$$\int_\Omega u_t \cdot v dx + \int_\Omega (u \cdot \nabla u) \cdot v dx + \nu \int_\Omega \nabla u : \nabla v dx + \frac{1}{\varepsilon} \int_\Omega (\nabla \cdot u)(\nabla \cdot v) dx = \int_\Omega f \cdot v dx \quad (5)$$

holds for all $v \in H^1_\varepsilon(\Omega), q \in Q$. When we solve the NSE using the penalty method in the numerical experiments to follow, rather than simultaneously seeking the velocity u and pressure p in their respective spaces, we first solve for u in (5) above and then proceed to update the pressure p at each step. Using a backward Euler time discretization, the problem becomes

`Find` $u^{n+1} \in H^1_\varepsilon(\Omega)$:

$$\int_\Omega \frac{u^{n+1} - u^n}{\Delta t} \cdot v dx + \int_\Omega (u^n \cdot \nabla u^{n+1}) \cdot v dx + \nu \int_\Omega \nabla u^{n+1} : \nabla v dx$$
$$+ \frac{1}{\varepsilon} \int_\Omega (\nabla \cdot u^{n+1})(\nabla \cdot v) dx = \int_\Omega f^{n+1} \cdot v dx$$

`Update` p^{n+1}:

$$\int_\Omega p^{n+1} q dx = -\frac{1}{\varepsilon} \int_\Omega \nabla \cdot u^{n+1} q dx$$

2.2 Stability and consistency error for the penalty method

2.2.1 Energy equality for the penalty method

Consider the penalty method applied to the semi-discretized NSE.

$$\frac{u^{n+1}-u^n}{\Delta t} + u^n \cdot \nabla u^{n+1} + \frac{1}{2}(\nabla \cdot u^n)u^{n+1} - \nu \Delta u^{n+1} - \nabla p^{n+1} = f^{n+1} \tag{6}$$

$$\varepsilon p^{n+1} + \nabla \cdot u^{n+1} = 0 \tag{7}$$

First, for any solution u to the incompressible NSE, we have that $\frac{1}{2}(\nabla \cdot u) = 0$, so the $\frac{1}{2}(\nabla \cdot u^n)u^{n+1}$ term is added to the momentum equation in order to formulate the explicitly skew-symmetrized trilinear form of the nonlinear term. We will now derive the energy equality for the method. The energy equality describes the energy in the system at the final time.

First, we take the dot product of the first equation with u^{n+1}, the dot product of the second equation with p^{n+1}, and integrate over the spatial domain Ω:

$$\int_\Omega \left(\frac{u^{n+1}-u^n}{\Delta t}\right) \cdot u^{n+1} + (u^n \cdot \nabla u^{n+1}) \cdot u^{n+1} + \big(\frac{1}{2}(\nabla \cdot u^n)\big) \cdot u^{n+1} - \nu \Delta u^{n+1} \cdot u^{n+1}$$

$$-\nabla p \cdot u^{n+1} dx = \int_\Omega f^{n+1} \cdot u^{n+1} dx$$

$$\int_\Omega \varepsilon p^{n+1} \cdot p^{n+1} + (\nabla \cdot u^{n+1}) \cdot p^{n+1} dx = 0$$

We now investigate and simplify each term, making sure to multiply through by Δt. Looking at the first term,

$$\int_\Omega \big(u^{n+1} - u^n\big) \cdot u^{n+1} dx = \int_\Omega u^{n+1} \cdot u^{n+1} - u^n \cdot u^{n+1} dx$$

Using the polarization identity $(x, y) = \frac{1}{2}(\|y\|^2 + \|x\|^2 - \|x-y\|^2)$, we then have

$$= \frac{1}{2}(\left\|u^{n+1}\right\|^2 - \|u^n\|^2 + \left\|u^{n+1} - u^n\right\|^2)$$

Using integration by parts and the divergence theorem,

$$-\int_\Omega \nu \Delta u^{n+1} \cdot u^{n+1} dx = \nu \left\|\nabla u^{n+1}\right\|^2$$

$$\int_\Omega \nabla p^{n+1} \cdot u^{n+1} dx = -\int_\Omega p^{n+1} \cdot (\nabla \cdot u^{n+1}) dx$$

Also note that by definition, $\int_\Omega p^{n+1} \cdot p^{n+1} dx = \|p^{n+1}\|^2$. Now, to simplify the nonlinear term, we use skew-symmetry as given by Lemma 1:

$$\int_\Omega \frac{1}{2}((\nabla \cdot u^n)u^{n+1}, u^{n+1}) + (u^n \cdot \nabla u^{n+1}, u^{n+1}) dx = 0$$

Now, rewriting the equations using these simplifications, we have

$$\begin{gathered}
\frac{1}{2}(\left\|u^{n+1}\right\|^2 - \|u^n\|^2 + \left\|u^{n+1} - u^n\right\|) + \Delta t \nu \left\|\nabla u^{n+1}\right\|^2 \\
-\Delta t \int_\Omega p^{n+1} \cdot \nabla \cdot u^{n+1} dx = \Delta t \int_\Omega f^{n+1} \cdot u^{n+1} dx \\
\Delta t \Bigg(\left\|p^{n+1}\right\|^2 + \int_\Omega (\nabla \cdot u^{n+1}) \cdot p^{n+1} dx \Bigg) = 0
\end{gathered}$$

Adding the above equations, we then have

$$\Delta t \varepsilon \left\|p^{n+1}\right\|^2 + \frac{1}{2}(\left\|u^{n+1}\right\|^2 - \|u^n\|^2 + \left\|u^{n+1} - u^n\right\|) + \Delta t \nu \left\|\nabla u^{n+1}\right\|^2 = \Delta t \int_\Omega f^{n+1} \cdot u^{n+1} dx$$

Now summing from 0 to N, the term $\sum_{n=0}^N \frac{1}{2}(\|u^{n+1}\|^2 - \|u^n\|^2 + \|u^{n+1} - u^n\|)$ is telescoping and hence

$$\sum_{n=0}^N \frac{1}{2}(\left\|u^{n+1}\right\|^2 - \|u^n\|^2 + \left\|u^{n+1} - u^n\right\|) = \frac{1}{2}(\left\|u^{N+1}\right\|^2 - \left\|u^0\right\|^2 + \sum_{n=0}^N \left\|u^{n+1} - u^n\right\|^2)$$

We then have the **energy equality** for the semi-discretized NSE with a penalty term:

$$\begin{gathered}
\frac{1}{2}(\left\|u^{N+1}\right\|^2 + \sum_{n=0}^N \left\|u^{n+1} - u^n\right\|^2) + \Delta t \nu \sum_{n=0}^N \left\|\nabla u^{n+1}\right\|^2 + \Delta t \varepsilon \sum_{n=0}^N \left\|p^{n+1}\right\|^2 \\
= \frac{1}{2}\left\|u^0\right\|^2 + \Delta t \sum_{n=0}^N \int_\Omega f^{n+1} \cdot u^{n+1} dx
\end{gathered}$$

2.2.2 Energy inequality and stability of the method

To derive the energy inequality, we look at the right hand side of the energy equality:

$$\frac{1}{2}\left\|u^0\right\|^2+\Delta t\sum_{n=0}^{N}\int_\Omega f^{n+1}\cdot u^{n+1}dx$$

For each n,

$$\int_\Omega f^{n+1}\cdot u^{n+1}dx=\frac{\int_\Omega f^{n+1}\cdot u^{n+1}dx\,\|\nabla u\|}{\|\nabla u\|}$$

By the definition of the dual norm $\|\cdot\|_{-1}$, it follows that

$$\frac{\int_\Omega f^{n+1}\cdot u^{n+1}dx\,\|\nabla u\|}{\|\nabla u\|}\le\sup_{u\in X}\frac{(f,u)}{\|\nabla u\|}=\|f\|_{-1}\|u\|$$

and by Young's inequality,

$$\le\frac{\nu}{2}\|\nabla u\|^2+\frac{1}{2\nu}\|f\|_{-1}^2$$

Hence, it follows that

$$\Delta t\sum_{n=0}^{N}\int_\Omega f\cdot u^{n+1}dx\le\frac{\Delta\nu}{2}\sum_{n=0}^{N}\left\|\nabla u^{n+1}\right\|^2+\frac{\Delta t}{2\nu}\sum_{n=0}^{N}\left\|f^{n+1}\right\|_{-1}^2$$

Applying this to the right hand side of the energy equality, we find the **energy inequality**:

$$\frac{1}{2}\left\|u^{n+1}\right\|^2+\frac{1}{2}\sum_{n=0}^{N}\left\|u^{n+1}-u^n\right\|^2+\frac{\Delta t\nu}{2}\sum_{n=0}^{N}\left\|\nabla u^{n+1}\right\|^2+\varepsilon\Delta t\sum_{n=0}^{N}\left\|p^{n+1}\right\|^2$$

$$\le\frac{1}{2}\left\|u^0\right\|^2+\frac{\Delta t}{2\nu}\sum_{n=0}^{N}\left\|f^{n+1}\right\|_{-1}^2$$

On the left hand side, we have the numerical dissipation term from the backward Euler time discretization, $\frac{1}{2}\sum_{n=0}^{N}\left\|u^{n+1}-u^n\right\|^2$ as well the viscous dissipation $\frac{\Delta t\nu}{2}\sum_{n=0}^{N}\left\|\nabla u^{n+1}\right\|^2$ and the energy $\frac{1}{2}\left\|u^{n+1}\right\|^2+\varepsilon\Delta t\sum_{n=0}^{N}\left\|p^{n+1}\right\|^2$. This shows the stability of the method, since we have for each $0\le n\le N$

$$\frac{1}{2}\left\|u^{n+1}\right\|^2+\varepsilon\Delta t\sum_{n=0}^{N}\left\|p^{n+1}\right\|^2\le C$$

for a constant C, so the solutions u and p are bounded.

2.2.3 Consistency error of the penalty method

Fix a solution to the NSE $(u, p) \in (X, Q)$. Now, by Taylor's theorem:

$$u^n = u^{n+1} - \Delta t u_t^{n+1} + \frac{\Delta t^2}{2} u_{tt}^{n+1} + \mathcal{O}(\Delta t^3)$$

We have the penalty method applied to the semi-discretized NSE with the explicitly skew-symmetrized nonlinear term

$$\frac{u^{n+1} - u^n}{\Delta t} + u^n \cdot \nabla u^{n+1} + \frac{1}{2}\Big(\nabla \cdot u^n\Big) u^{n+1} - \nu \Delta u^{n+1} + \nabla p^{n+1} = f^{n+1}$$

$$\varepsilon p^{n+1} + \nabla \cdot u^{n+1} = 0$$

as well as the incompressible NSE evaluated at $t = t_{n+1}$

$$u_t^{n+1} + u^{n+1} \cdot \nabla u^{n+1} - \nu \Delta u^{n+1} + \nabla p^{n+1} = f^{n+1}$$

$$\nabla \cdot u^{n+1} = 0$$

To obtain the consistency error, we subtract the incompressible NSE from the model (noting that, as above, for a solution u to the NSE, $\frac{1}{2}(\nabla \cdot u^n)u^{n+1} = 0$):

$$\tau(u) := \frac{u^{n+1} - u^n}{\Delta t} - u_t^{n+1} + (u^n - u^{n+1}) \cdot \nabla u^{n+1}$$

$$\tau(p) := \varepsilon p^{n+1}$$

Using the Taylor expansion of u, we then have that

$$\frac{u^{n+1} - u^n}{\Delta t} - u_t^{n+1} = -\frac{\Delta t}{2} u_{tt} + \mathcal{O}(\Delta t^2) = \mathcal{O}(\Delta t)$$

and by Cauchy-Schwarz,

$$(u^n - u^{n+1}) \cdot \nabla u^{n+1} \leq \left\| u^{n+1} - u^n \right\| \left\| \nabla u^{n+1} \right\|$$

Since $u \in X$, it follows that $\|\nabla u^{n+1}\| \leq C_1$ for some constant C_1, and

$$\left\| u^n - u^{n+1} \right\| = \left\| \Delta t u_t^{n+1} \right\| - \frac{\Delta t^2}{2} u_{tt}^{n+1} + \mathcal{O}(\Delta t^3) \leq C_2 \left\| \Delta t u_t^{n+1} \right\|$$

so that for $C = C_1 C_2$

$$\nabla u^{n+1} \leq \left\| u^{n+1} - u^n \right\| \left\| \nabla u^{n+1} \right\| \leq C \left\| \Delta t u_t^{n+1} \right\| \leq C\Delta t = \mathcal{O}(\Delta t)$$

So the consistency error $\tau(u) = \mathcal{O}(\Delta t)$. Looking at the consistency error for the pressure:

$$\left\| \varepsilon p^{n+1} \right\| \leq C\varepsilon = \mathcal{O}(\varepsilon)$$

since $p \in Q$, hence has a bounded norm.

Choosing the penalty parameter $\varepsilon = \mathcal{O}(\Delta t)$, we then have

$$\tau(p) = \mathcal{O}(\varepsilon) = \mathcal{O}(\Delta t)$$

Now, this suggests that the method is *first order*. Under more stringent regularity conditions, the following result from [10] holds for approximations $u(t_n)$ of the velocity u at the nth time-step produced by the penalty method:

Theorem 2. For all $n \leq N + 1$,

$$\sqrt{t_n} \left\| u(t_n) - u^n \right\| + t_n \left\| u(t_n) - u^n \right\|_1 \leq C(\Delta t + \varepsilon)$$

A similar result from [10] holds for the pressure.

2.3 Convergence test of the penalty method with Taylor-Green test problem

Consider the unit square $\Omega = (0,1) \times (0,1) \subset \mathbb{R}^2$ and $0 \leq t \leq 1.0$, where Ω is discretized by a mesh with $N = 100$ nodes on each edge, and the viscosity $\nu = 1/100$. We have the Taylor-Green solution:

$$u = (-\cos(\pi x)\sin(\pi y)e^{-2\pi^2 \nu t}, \sin(\pi x)\cos(\pi y)e^{-2\pi^2 \nu t})$$
$$p = \frac{-1}{4}(\cos(2\pi x) + \cos(2\pi y))e^{-4\pi^2 \nu t}$$

We apply the fully discretized penalty method to this problem, using the Finite Element Method (FEM) for the spatial discretization in FEniCS [1]. The solution space is the Taylor-Hood element space $\mathbb{P}_2/\mathbb{P}_1$: piecewise quadratic polynomial solutions for the velocity and piecewise linear solutions for the pressure. To examine convergence, we pick timesteps $\Delta t \in [2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}]$ and plot for each timestep the error $e(\Delta t)$ against the timestep Δt on a log-log plot. The error is calculated as follows:

$$\text{Velocity error: } \max_n \|u_n - u_{exact}\|_2 \, (\ell^\infty L^2)$$
$$\text{Pressure error: } \sum_n \|p_n - p_{exact}\|_2 \, (\ell^1 L^2)$$

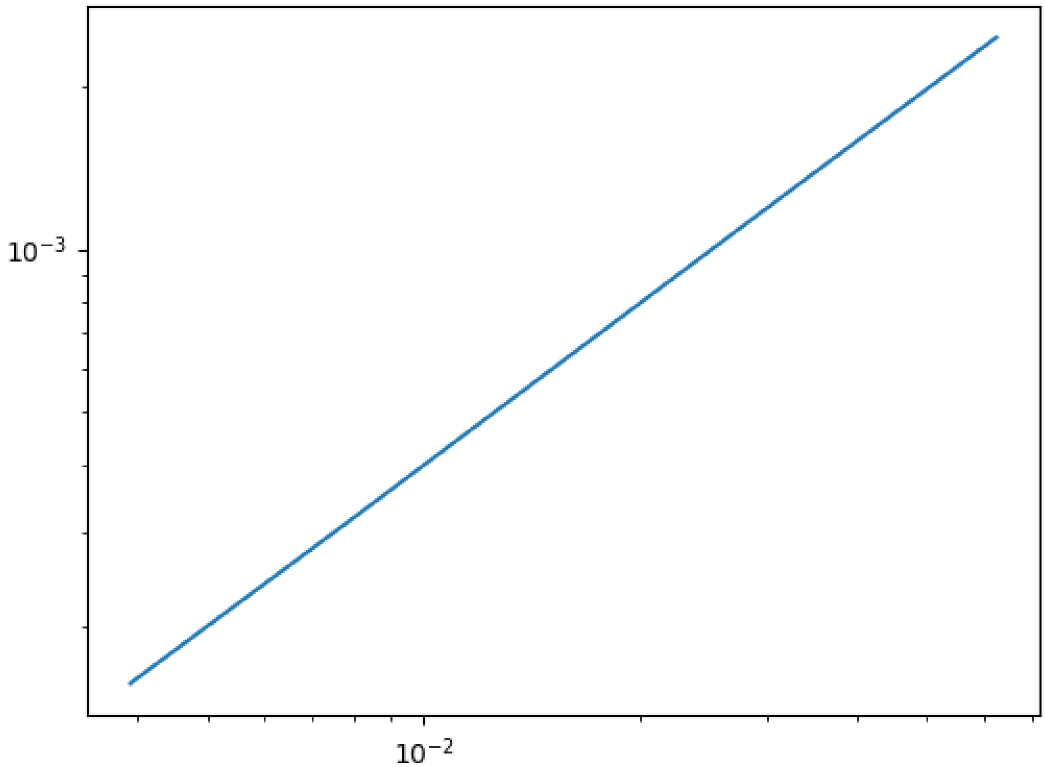

Figure 1: Log-log plot of the velocity error for the Taylor-Green problem, with the timestep Δt on the x-axis and the error on the y-axis.

This demonstrates the first-order convergence of the velocity in the penalty method, since there is a line of slope 1 on the log-log plot.

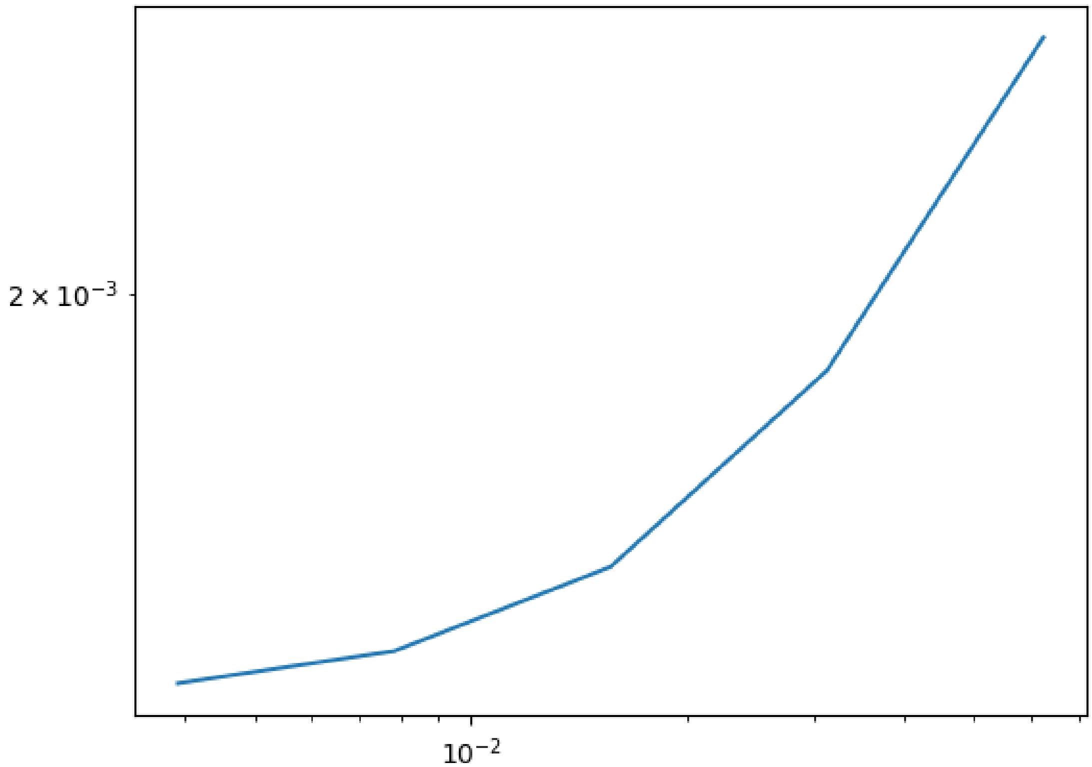

Figure 2: Log-log plot of the pressure error for the Taylor-Green problem, with the timestep Δt on the x-axis and the error on the y-axis.

2.3.1 Convergence test and error plots of the penalty method for a different test problem

Consider as above the unit square $(0,1)\times(0,1) \subset \mathbb{R}^2$, discretized by a mesh with $N = 100$ nodes on each edge, and $0 \le t \le 1$. Let $\nu = 1$. Consider the following test problem, due to [3]

$$u = \pi \sin t(\sin 2\pi y \sin^2 \pi x, -\sin 2\pi x \sin^2 \pi y)$$

$$p = \cos t \cos \pi x \sin \pi y$$

We apply the fully discretized penalty method to this problem, using the Finite Element Method (FEM) for the spatial discretization. The solution space is the Taylor-Hood element space $\mathbb{P}_2/\mathbb{P}_1$. As above, to examine convergence, we pick timesteps $\Delta t \in [2^{-6}, 2^{-7}, 2^{-8}, 2^{-9}, 2^{-10}]$. The log-log plots of convergence follow.

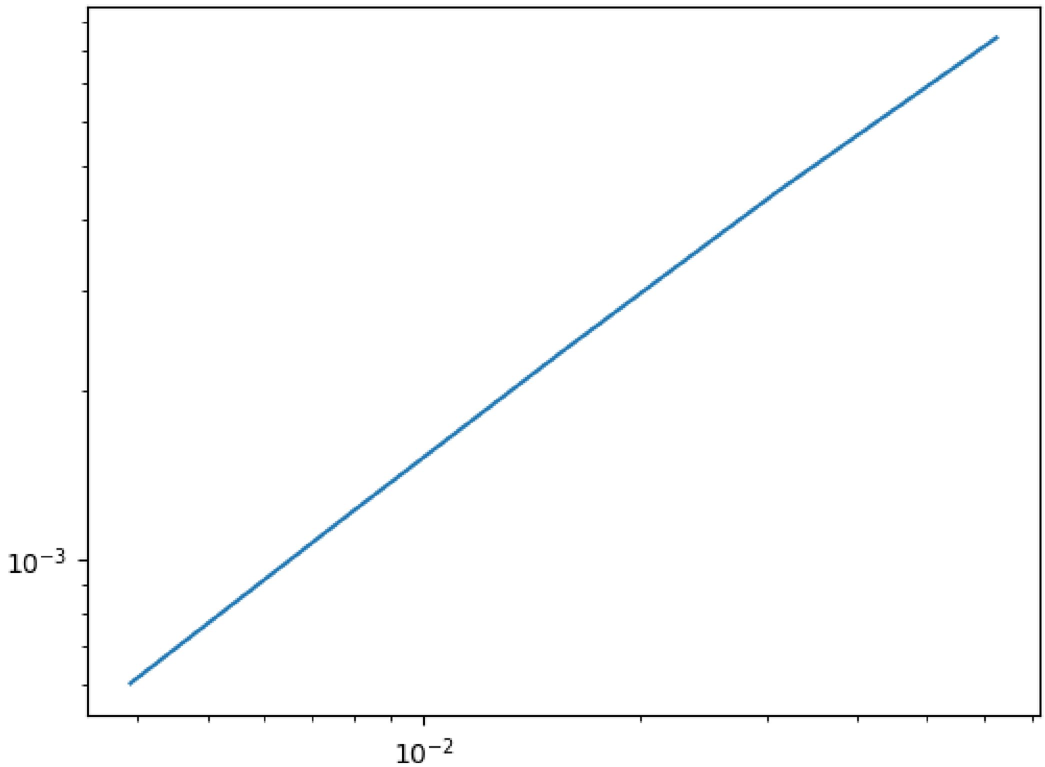

Figure 3: Log-log plot of the velocity error for the Guermond-Shen-Minev problem, with the timesteps Δt on the x-axis and the error on the y-axis. This demonstrates the first-order convergence of the velocity in the penalty method, since there is a line of slope 1 on the log-log plot.

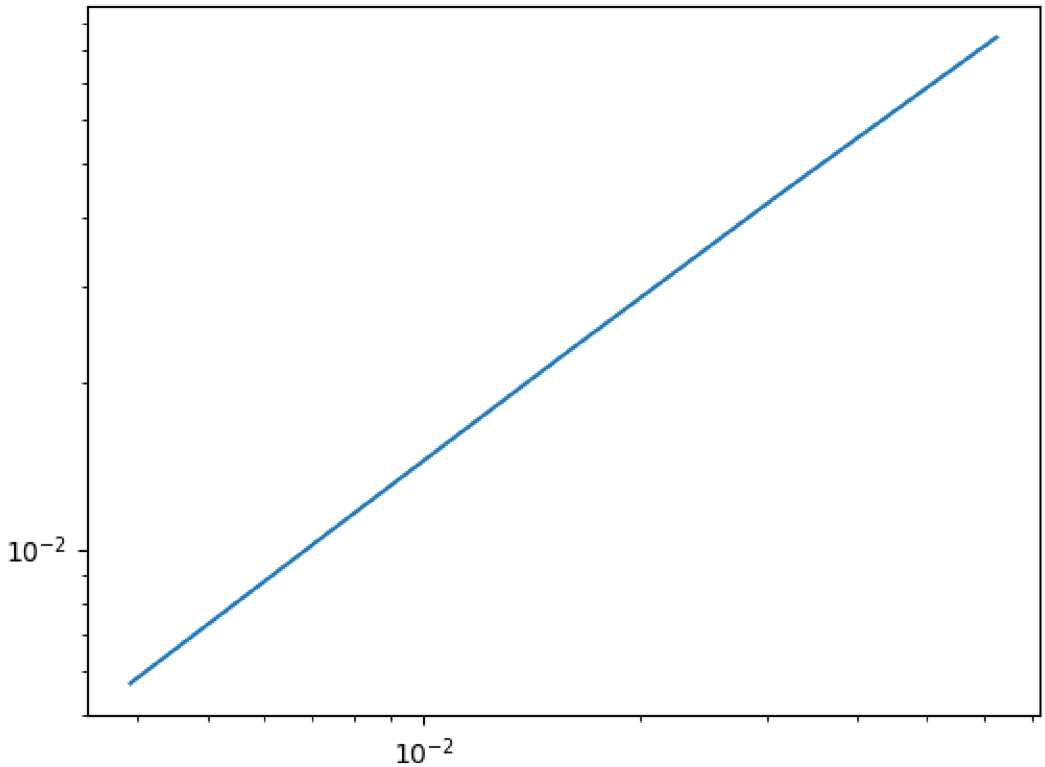

Figure 4: Log-log plot of the pressure error for the Guermond-Shen-Minev problem, with the timesteps Δt on the x-axis and the error on the y-axis. This demonstrates the first-order convergence of the pressure in the penalty method, since there is a line of slope 1 on the log-log plot.

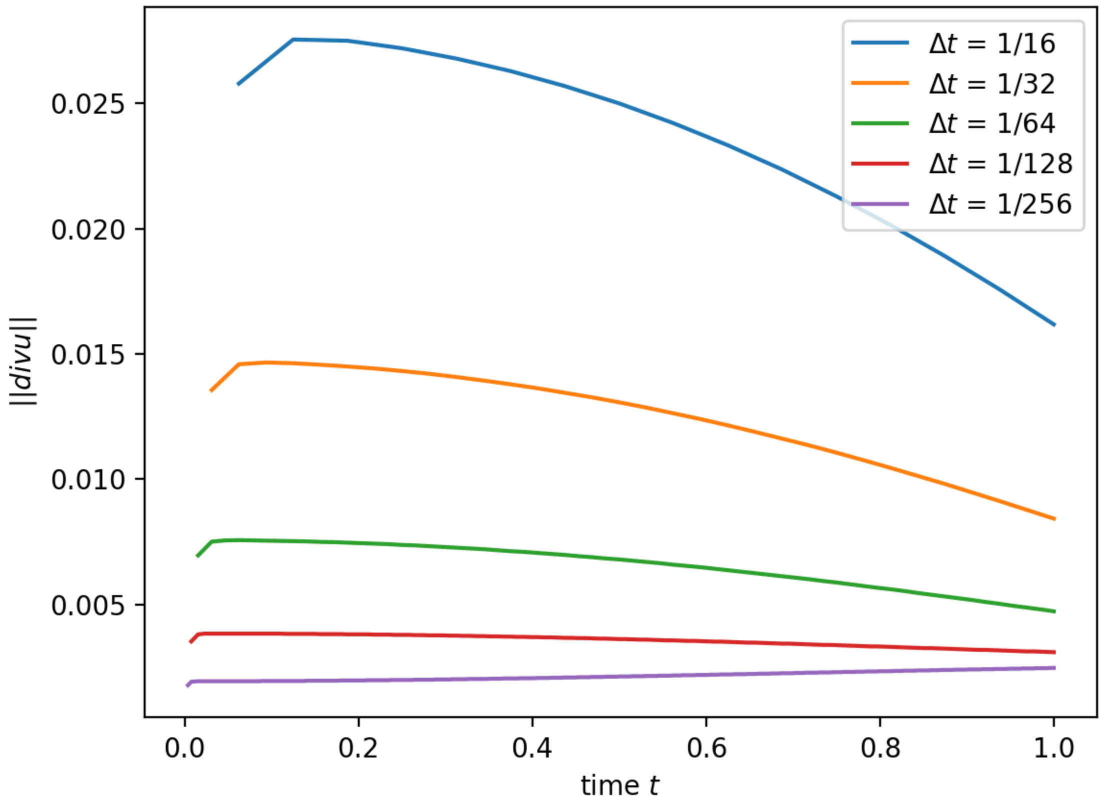

Figure 5: For the Guermond-Shen-Minev problem, a plot of $\|\nabla \cdot u^{n+1}\|$ over time t, with $0 \leq t \leq 1$. The error for different timesteps Δt is marked on the legend.

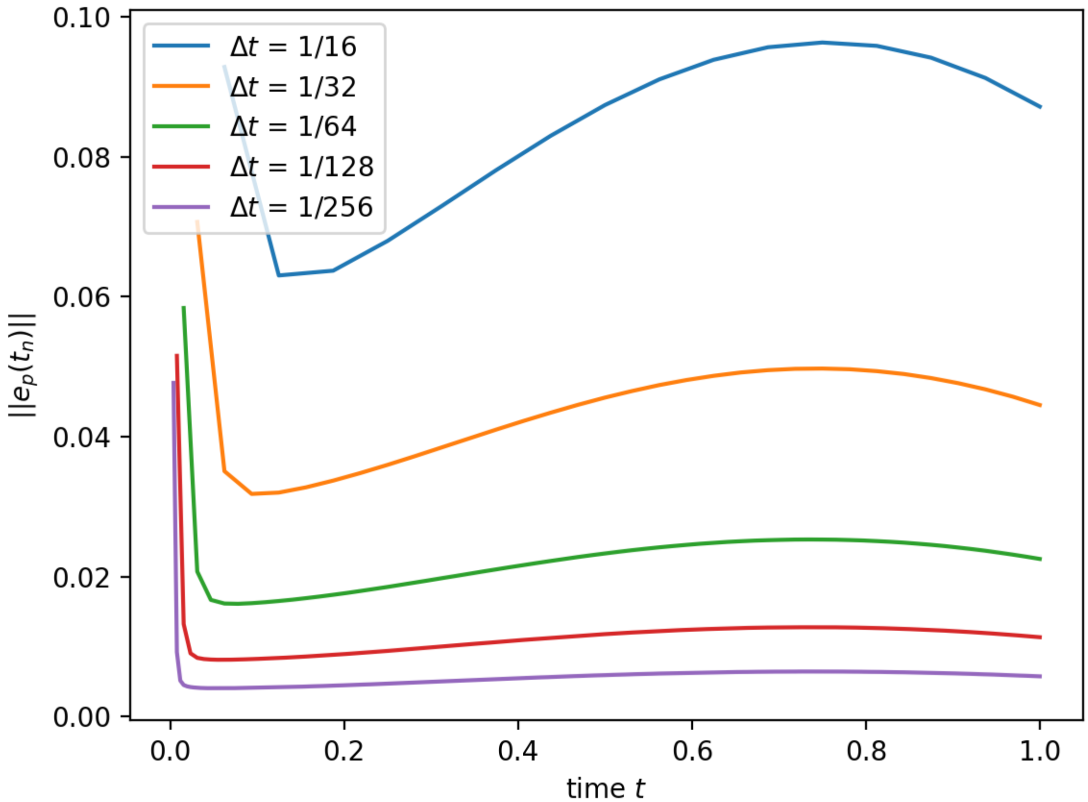

Figure 6: For the Guermond-Shen-Minev problem, a plot of the pressure error over time t, with $0 \leq t \leq 1$. The error for different timesteps Δt is marked on the legend.

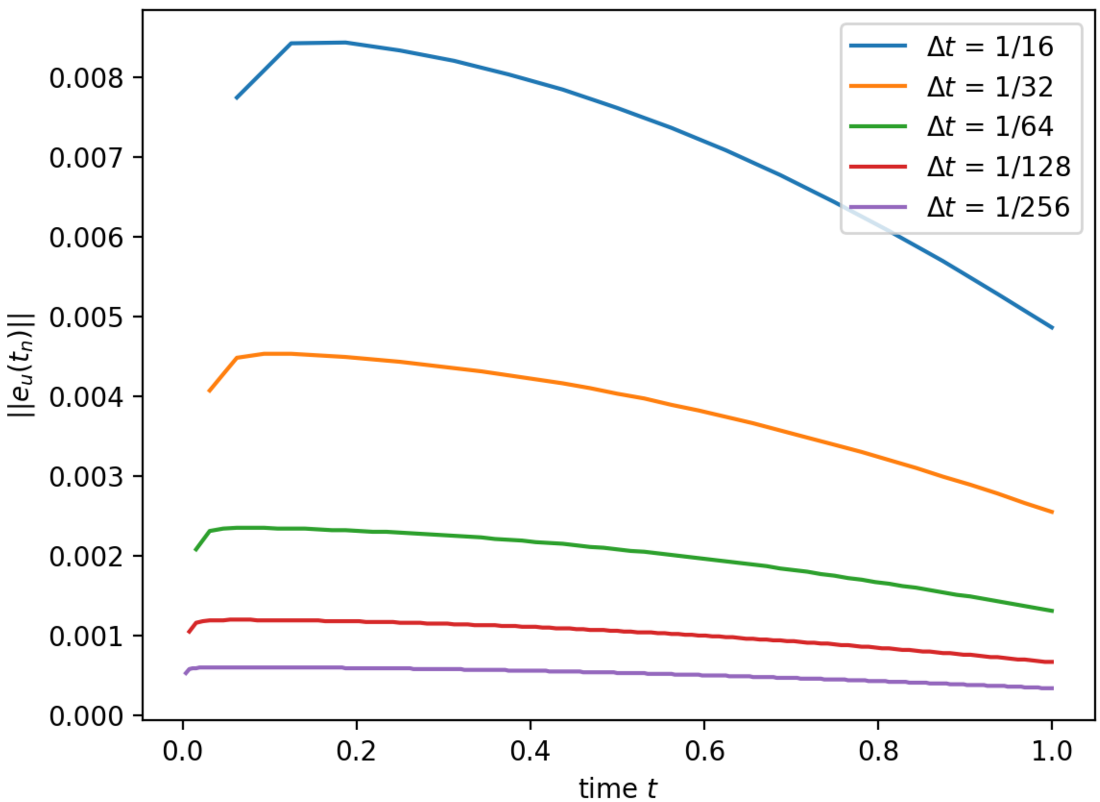

Figure 7: For the Guermond-Shen-Minev problem, a plot of the velocity error over time t, with $0 \leq t \leq 1$. The error for different timesteps Δt is marked on the legend.

2.3.2 A sample code in FEniCS

The following code in FEniCS implements the penalty method, with the Taylor-Green test problem, on the unit square $(0,1) \times (0,1)$ and $0 \leq t \leq 1$.

```
from dolfin import *
import numpy as np
import sympy as sp


N = 100 # Mesh width
```

```
Re = 100.0 # Reynolds number
nu = 1./Re # Viscosity
Pi = np.pi
T_0 = 0.0 # Starting time
T_N = 1.0 # Final time
t = T_0
DT = 0.01 # Timestep size
eps = DT # Choosing eps = O(DT)
TOL = 1.0e-10

# Creating the mesh in FEniCS
mesh = UnitSquareMesh(N,N)

# P2/P1 Taylor-Hood element space

X = VectorFunctionSpace(mesh,'Lagrange',2)
Q = FunctionSpace(mesh,'Lagrange',1)

# Test and Trial functions
u = TrialFunction(X)
p = TrialFunction(Q)

v = TestFunction(X)
q = TestFunction(Q)

unPlus1 = Function(X)
pnPlus1 = Function(Q)

un = Function(X)
pn = Function(Q)
```

```
# Viscous term

def a(u,v):
    return inner(nabla_grad(u),nabla_grad(v))

# Nonlinear term

def b(u,v,w):
    return 0.5 * ( inner ( dot(u, nabla_grad(v)), w)  \
    - inner( dot(u, nabla_grad(w) ), v) )


x,y,s = sp.symbols('x[0] x[1] s') # where s is a variable for time

# exact solution u = (u1, u2)

u1_exact = -sp.cos(Pi * x) * sp.sin(Pi * y) \
* sp.exp(-2.0 * (Pi**2) * nu * s)
u2_exact = sp.sin(Pi * x) * sp.cos(Pi * y) \
* sp.exp(-2.0 * (Pi**2) * nu * s)

# exact solution p
p_exact = (-1./4)*(sp.cos(2*Pi*x)+sp.cos(2*Pi*y))*sp.exp(-4.0*(Pi**2)*nu*s)

# Generating forcing functions f1, f2

f1 = u1_exact.diff(s, 1) + u1_exact * u1_exact.diff(x, 1) + \
u2_exact * u1_exact.diff(y, 1) - \
nu * sum( u1_exact.diff(xi, 2) for xi in (x,y) ) + p_exact.diff(x, 1)
```

```
f2 = u2_exact.diff(s, 1) + u1_exact * u2_exact.diff(x, 1) + \
u2_exact * u2_exact.diff(y, 1) - \
nu*sum( u2_exact.diff(xi,2) for xi in (x,y) ) + p_exact.diff(y, 1)

# making these functions FEniCS-friendly

u_exact = Expression((sp.printing.ccode(u1_exact), \
sp.printing.ccode(u2_exact)), degree = 2, s = t)

p_exact = Expression(sp.printing.ccode(p_exact), degree = 1, s = t)

f = Expression((sp.printing.ccode(f1), sp.printing.ccode(f2)), \
degree = 2, s = t)

# Initial conditions

un.assign(interpolate(u_exact, X))

pn.assign(interpolate(p_exact, Q))

# Boundary conditions

def boundary(x,on_boundary):
    return on_boundary

bc_u = DirichletBC(X, u_exact, boundary)

# Main solve
```

```
while t <= T_N + TOL:
    print('Numerical time level t =', t)
    u_exact.s = t # setting the value for s to the current time
    p_exact.s = t
    f.s = t

    # Solving for velocity u
    a_u = (1./DT) * inner(u,v) * dx + b(un,u,v) * dx + \
    nu * a(u,v) * dx + (1./eps) * div(u) * div(v) * dx

    # Solving for pressure p
    a_p = eps * inner(p,q) * dx

    # Right hand side, A_u * u = f
    b_u = (1./DT) * inner(un,v) * dx + inner(f,v) * dx
    #Right hand side, A_p * p = f
    b_p = -div(unPlus1)*q*dx

    A_u = assemble(a_u)
    B_u = assemble(b_u)

    # Applying boundary conditions
    bc_u.apply(A_u, B_u)

    solve(A_u, unPlus1.vector(), B_u)

    A_p = assemble(a_p)
    B_p = assemble(b_p)

    solve(A_p,pnPlus1.vector(), B_p)
```

```
# L2 error
verror = sqrt( assemble(inner(unPlus1 - u_exact, unPlus1 - u_exact) \
* dx) )
perror = sqrt( assemble(inner(pnPlus1 - p_exact, pnPlus1 - p_exact) \
* dx) )

print ('velocity error = ', verror)
print ('pressure error = ', perror)

# Updating un, pn
un.assign(unPlus1)
pn.assign(pnPlus1)

# Stepping forward in time
t += DT
```

3.0 Adaptive penalty methods

3.1 Adaptive methods

Consider the error $e(t_n)$ at the nth timestep and fix a tolerance TOL. An *adaptive method* seeks to optimize a scheme at each timestep by comparing the current error with a computable estimate EST for the error. A simple halving-and-doubling scheme to adapt the timestep has the following form:

`if` $EST > TOL$ `then repeat the step with:`

$$\Delta t \leftarrow \frac{\Delta t}{2}$$

`else if` $EST << TOL$ `continue and on the next iteration:`

$$\Delta t \leftarrow 2\Delta t$$

The aim is to increase efficiency (by increasing the size of Δt and thus reducing computational complexity if the error is much too small) and accuracy (by ensuring that the computed answer at each timestep is below a specified TOL) [7]. In the case of the penalty method, we seek to adapt the timestep Δt as well as the penalty parameter ε.

3.2 A doubly adaptive scheme for the penalty method

3.2.1 Error estimators for adapting ε, Δt

In order to implement an adaptive scheme, reliable estimates of the error are required. The following error estimators and scheme are from [6] and [4].

Consider the previous computed u^n, u^{n-1} and the previous timesteps $\Delta t_n, \Delta t_{n-1}$ and the constants $\tau = \frac{\Delta t_{n+1}}{\Delta t_n}$ and $\alpha = \frac{\tau(1+\tau)}{1+2\tau}$. Define, for each n, $u^* = (1+\tau)u^n - \tau u^{n-1}$. To estimate the consistency error for the momentum equation, consider the backward Euler approximation with $p = -\frac{1}{\varepsilon}\nabla \cdot u^{n+1}$ and find the first-order approximation u_1^{n+1}

$$\frac{u^{n+1} - u^n}{\Delta t_{n+1}} + u^* \cdot \nabla u^{n+1} + \frac{1}{2}(\nabla \cdot u^*)u^{n+1} - \frac{\Delta t_{n+1}}{\varepsilon_{n+1}} \nabla\nabla \cdot u^{n+1} - \nu \Delta u^{n+1} = f^{n+1}$$

Then we have this second-order approximation based on the backward Euler approximation plus a time filter [4],

$$u^{n+1} = u_1^{n+1} - \frac{\alpha}{2}\left(\frac{2\Delta t_n}{\Delta t_n + \Delta t_{n+1}} u_1^{n+1} - 2u_n + \frac{2\Delta t_n}{\Delta t_n + \Delta t_{n+1}} u^{n-1} \right)$$

So the error estimator EST_1 to adapt the timestep Δt is

$$EST_1 = \left\| u^{n+1} - u_1^{n+1} \right\|$$

To adapt ε, consider the estimator EST_2

$$EST_2 = \left\| \nabla \cdot u^{n+1} \right\|$$

where both norms are the L^2 norm.

3.2.2 Pseudocode for doubly adaptive scheme

At each step, we implement a modified halving-and-doubling scheme, from [6]. Say that $EST_2 > TOL_2$ and the current approximation for u^{n+1} is then unacceptable. Then, in a typical halving-and-doubling scheme, we would take

$$\varepsilon_{n+1} = \frac{1}{2}\varepsilon_{n+1}$$

However, in this scheme we take halving ε to be a worst-case scenario. In cases where ε_{n+1} should be shrunk but not so sharply, i.e. if $\frac{TOL_1}{EST_1}$ is close to 1, we take instead

$$\varepsilon_{n+1} = 0.9\varepsilon_{n+1}\frac{TOL_1}{EST_1}$$

to ensure that ε does not vary too much throughout the scheme. The constant 0.9 is the safety factor.

In the FEniCS code, the following adaptive scheme from [6] will be added before the next update u^{n+1}, p^{n+1} are accepted, given a tolerance TOL_1 for the error in the momentum equation and TOL_2 for the error in the continuity equation:

`if` $EST_1 > TOL_1$ `then repeat the current step with`

$$\Delta t_{n+1} = \max\{0.9\Delta t_n\left(\frac{TOL_1}{EST_1}\right)^{1/2}, \frac{1}{2}\Delta t_{n+1}\}$$

`elseif` $EST_2 > TOL_2$ `then repeat the current step with`

$$\varepsilon_{n+1} = \max\{0.9\varepsilon_{n+1}\frac{TOL_2}{EST_2}, \frac{1}{2}\varepsilon_{n+1}\}$$

`else, accept the current approximation and on the next step`

$$\Delta t_{n+2} = \max\{\min\{0.9\Delta t_{n+1}\left(\frac{TOL_1}{EST_1}\right)^{1/2}, 2\Delta t_{n+1}\}, \frac{1}{2}\Delta t_{n+1}\}$$

$$\varepsilon_{n+2} = \max\{\min\{0.9\varepsilon_{n+1}\frac{TOL_2}{EST_2}, 2\varepsilon_{n+1}\}, \frac{1}{2}\varepsilon_{n+1}\}$$

3.2.3 Energy equality for the scheme

Consider the doubly-adaptive penalty method scheme, where now u^n has been replaced by the second-order extrapolation $u^* = (1+\tau)u^n - \tau u^{n-1}$ and ε now varies in time:

$$\frac{u^{n+1} - u^n}{\Delta t} + u^* \cdot \nabla u^{n+1} + \frac{1}{2}\left(\nabla \cdot u^*\right)u^{n+1} - \nu \Delta u^{n+1} + \nabla p^{n+1} = f^{n+1} \tag{8}$$

$$\varepsilon^{n+1} p^{n+1} + \nabla \cdot u^{n+1} = 0 \tag{9}$$

Then, to find the energy equality, we use the same steps as in finding the energy equality for the semi-discretized NSE with penalty method: take inner products with u^{n+1}, p^{n+1}, rewrite the equations using the polarization identity and integration by parts, and sum over time. Letting $\|\cdot\|$ denote the L^2 norm, we find after the aforementioned steps:

$$\frac{1}{2\Delta t_{n+1}}\left(\left\|u^{n+1}\right\|^2 - \|u^n\|^2 + \left\|u^{n+1} - u^n\right\|^2 \right) + \nu \left\|\nabla u^{n+1}\right\|^2 - \int_\Omega p^{n+1} \cdot (\nabla \cdot u^{n+1}) dx$$

$$= \int_\Omega f^{n+1} \cdot u^{n+1} dx$$

$$\varepsilon^{n+1} \left\|p^{n+1}\right\|^2 + \int_\Omega p^{n+1} \cdot (\nabla \cdot u^{n+1}) dx = 0$$

Adding the two equations and multiplying through by $2\Delta t_{n+1}$, we then have:

$$\left\|u^{n+1}\right\|^2 - \|u^n\|^2 + \left\|u^{n+1} - u^n\right\|^2 + 2\Delta t_{n+1}\varepsilon^{n+1} \left\|p^{n+1}\right\|^2 + 2\Delta t_{n+1}\nu \left\|\nabla u^{n+1}\right\|^2$$

$$= 2\Delta t_{n+1} \int_\Omega f^{n+1} \cdot u^{n+1} dx$$

Now, summing from 0 to N and using the fact that the sum $\sum_{n=0}^{N} \left\|u^{n+1}\right\|^2 - \|u^n\|^2$ is telescoping, we arrive at the **energy equality**:

$$\left\|u^{N+1}\right\|^2 + \sum_{n=0}^{N} \left\|u^{n+1} - u^n\right\|^2 + 2\sum_{n=0}^{N} \Delta t_{n+1}\varepsilon^{n+1} \left\|p^{n+1}\right\|^2 + 2\nu \sum_{n=0}^{N} \Delta t_{n+1} \left\|\nabla u^{n+1}\right\|^2$$

$$= 2\sum_{n=0}^{N} \Delta t_{n+1} \int_\Omega f^{n+1} \cdot u^{n+1} dx + \left\|u^0\right\|^2$$

3.2.4 Energy inequality and stability

To find the energy inequality, as before we apply Young's inequality and bound the right-hand side from above. Note that $\varepsilon^{n+1}\left\|p^{n+1}\right\|^2 = \left\|\sqrt{\varepsilon}^{n+1}p^{n+1}\right\|^2$. Using the definition of the dual norm as in 2.1.2,

$$\int_\Omega f^{n+1}\cdot u^{n+1}dx = \frac{\int_\Omega f^{n+1}\cdot u^{n+1}dx\left\|\nabla u^{n+1}\right\|}{\left\|\nabla u^{n+1}\right\|} \leq \left\|f^{n+1}\right\|_{-1}\left\|\nabla u^{n+1}\right\|$$

Using Young's inequality,

$$\left\|f^{n+1}\right\|_{-1}\left\|\nabla u^{n+1}\right\|^2 \leq \frac{\nu}{2}\left\|\nabla u^{n+1}\right\|^2 + \frac{1}{2\nu}\left\|f^{n+1}\right\|_{-1}^2$$

so we have the energy inequality

$$\left\|u^{N+1}\right\|^2 + 2\sum_{n=0}^{N}\Delta t_{n+1}\left\|\sqrt{\varepsilon^{n+1}}p^{n+1}\right\|^2 + \nu\sum_{n=0}^{N}\Delta t_{n+1}\left\|\nabla u^{n+1}\right\|^2 + \sum_{n=0}^{N}\left\|u^{n+1}-u^n\right\|^2$$

$$\leq \frac{1}{\nu}\sum_{n=0}^{N}\Delta t_{n+1}\left\|f^{n+1}\right\|_{-1}^2 + \left\|u^0\right\|^2$$

3.3 Numerical tests of the adaptive scheme

3.3.1 Test with fixed timestep, variable ε

Consider, as above for the penalty method, the unit square $(0,1)\times(0,1)\subset\mathbb{R}^2$, discretized by a mesh with 100 nodes on each edge, with $\nu = 1$. Let $0 \leq t \leq 10$. We have the Guermond-Shen-Minev test problem [3]

$$u = \pi\sin t(\sin 2\pi y\sin^2\pi x, -\sin 2\pi x\sin^2\pi y)$$

$$p = \cos t\cos\pi x\sin\pi y$$

Fix the timestep $\Delta t \in [2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}, 2^{-9}]$, and the tolerance $TOL = 1/100$. We also have the error estimator as above, $EST_1 = \|\nabla\cdot u^{n+1}\|$. Then, for each timestep Δt, we vary ε according to the following adaptive scheme, capping ε at 0.1:

`if` $EST_2 > TOL$ `then repeat the current step with`

$$\varepsilon_{n+1} = \min\{\max\{0.9\varepsilon_{n+1}\frac{TOL}{EST_2}, \frac{1}{2}\varepsilon_{n+1}\}, 0.1\}$$

`else, accept the current approximation and on the next step`

$$\varepsilon_{n+2} = \min\{\max\{\min\{0.9\varepsilon_{n+1}\frac{TOL}{EST_2}, 2\varepsilon_{n+1}\}, \frac{1}{2}\varepsilon_{n+1}\}, 0.1\}$$

The following plots are of the pressure error, velocity error, $\|\nabla \cdot u\|$ as well as the evolution of ϵ, all over time.

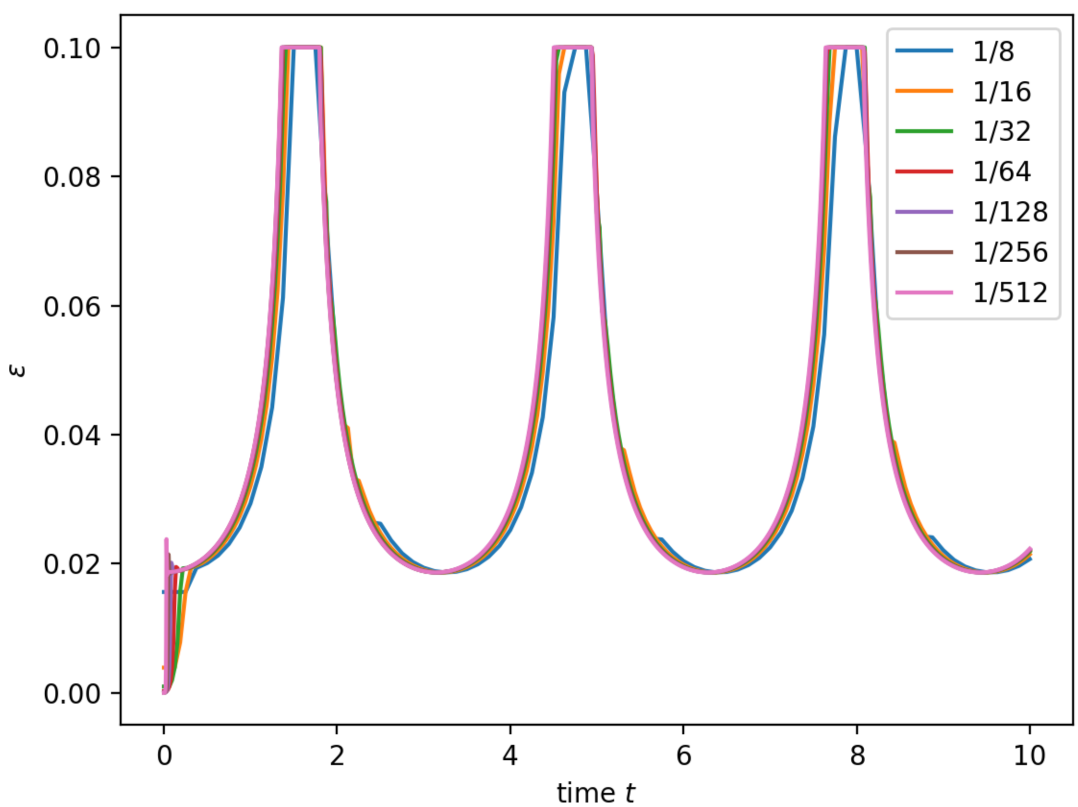

Figure 8: The evolution of ε over time for the fixed-Δt, variable ε test. Note the oscillatory pattern of the evolution: ε caps out at 0.1 as $t = k\frac{\pi}{2}$, where the exact solution is 0.

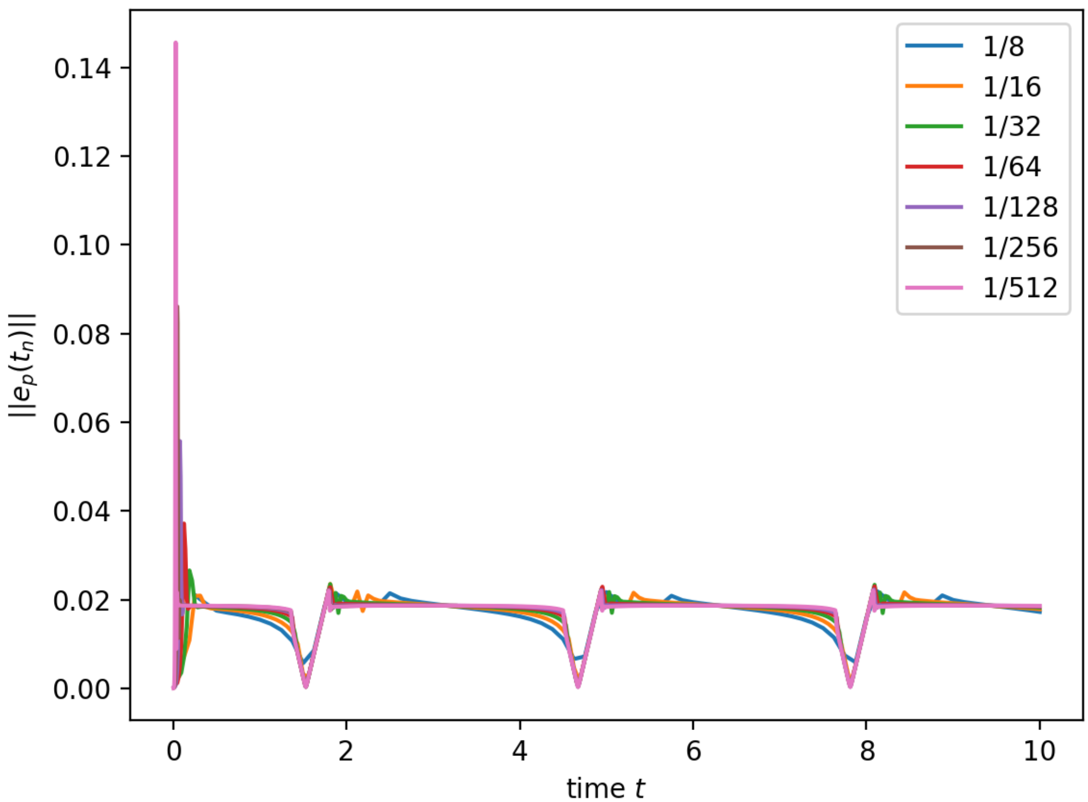

Figure 9: The pressure error for the fixed-Δt, variable ε test. On the y-axis is the error $\|p_n - p_{exact}\|$, and on the x axis is time, $0 \leq t \leq 10$.

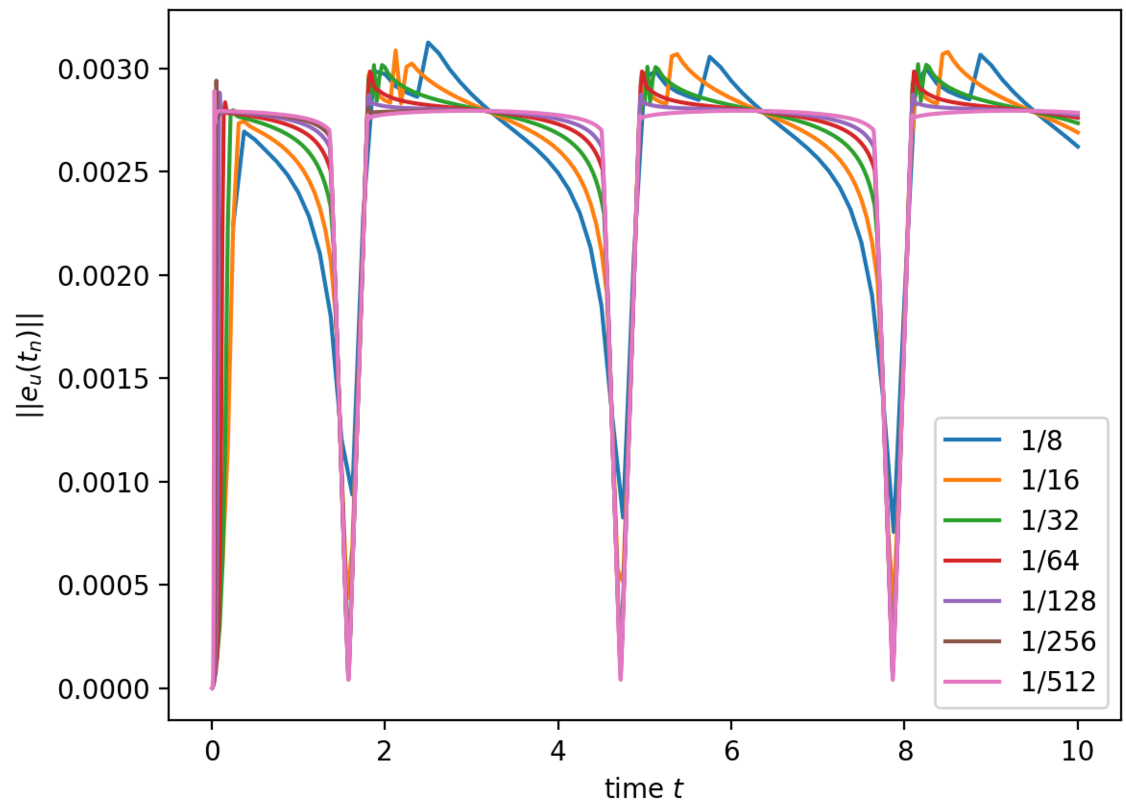

Figure 10: The velocity error for the fixed-Δt, variable ε test. On the y-axis is the error $\|u_n - u_{exact}\|$, and on the x axis is time, $0 \leq t \leq 10$..

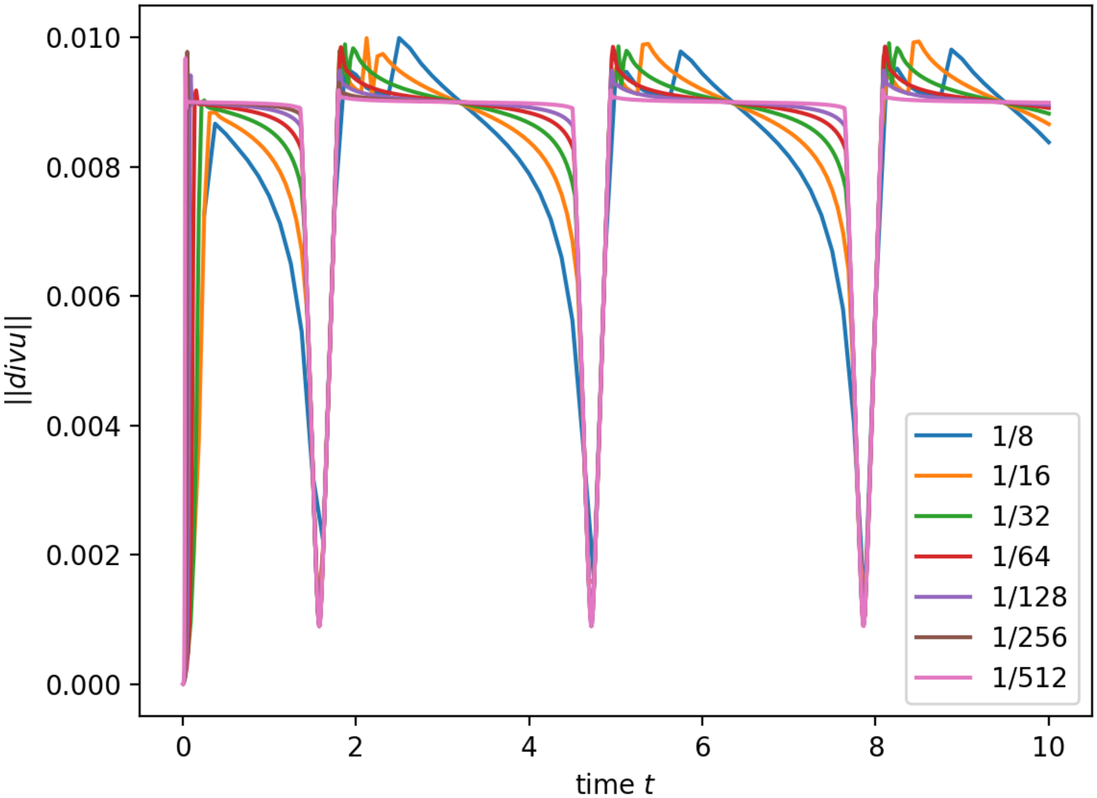

Figure 11: The norm $||\nabla \cdot u||$ for the fixed-Δt, variable ε test.On the y-axis is $\|\nabla \cdot u^{n+1}\|$, and on the x axis is time, $0 \leq t \leq 10$.

3.3.2 Fixed timestep, variable ε with a different error estimator

As before, we fix $\Delta t \in [2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}, 2^{-9}]$ and vary ε. Let the tolerance $TOL = 0.1$. Now, let the error estimator in the continuity equation $EST_2 = \frac{||\nabla \cdot u^{n+1}||}{||u^{n+1}||}$, the relative error. Using the same adaptive scheme as in the previous section to adapt ε, the following graphs compare the performance of using this error estimator $EST_2 = \frac{||\nabla \cdot u^{n+1}||}{||u^{n+1}||}$ versus $EST_1 = ||\nabla \cdot u^{n+1}||$.

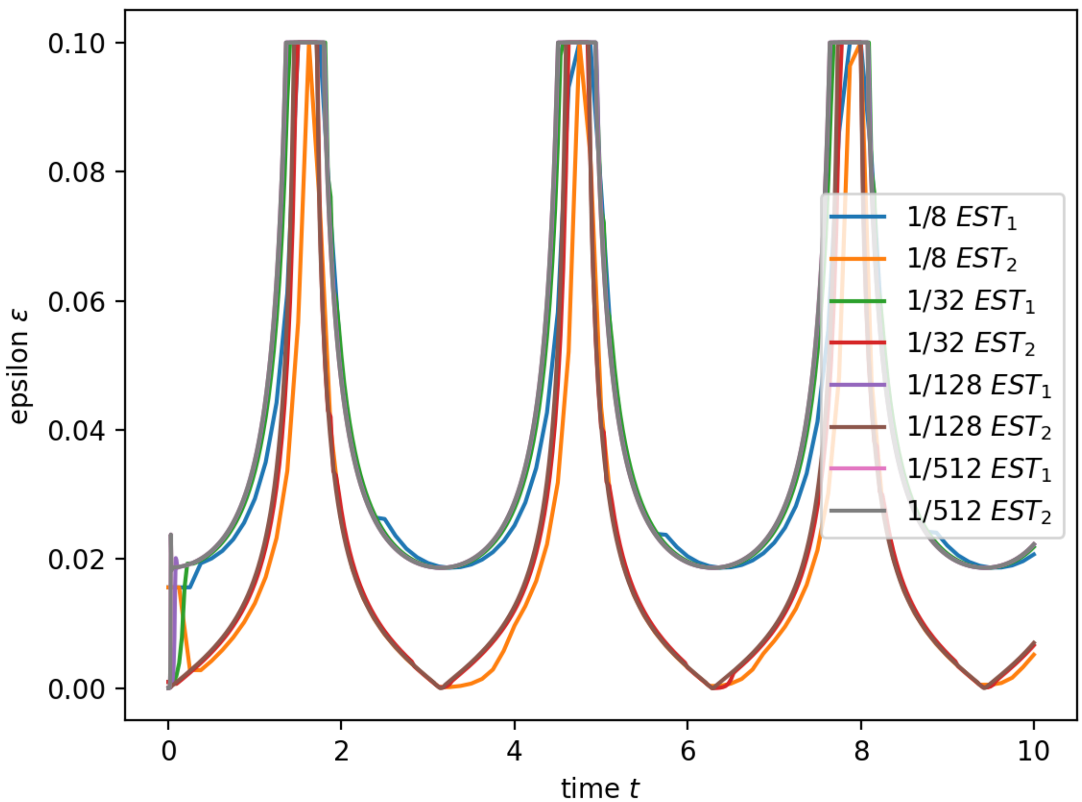

Figure 12: The evolution of ε over time for the fixed-Δt, variable ε test with a relative error estimator. Note the oscillatory pattern of the evolution: ε caps out at 0.1 as $t = k\frac{\pi}{2}$, where the exact solution is 0. The plots for the old error estimator appear clustered above the plot for the new error estimator.

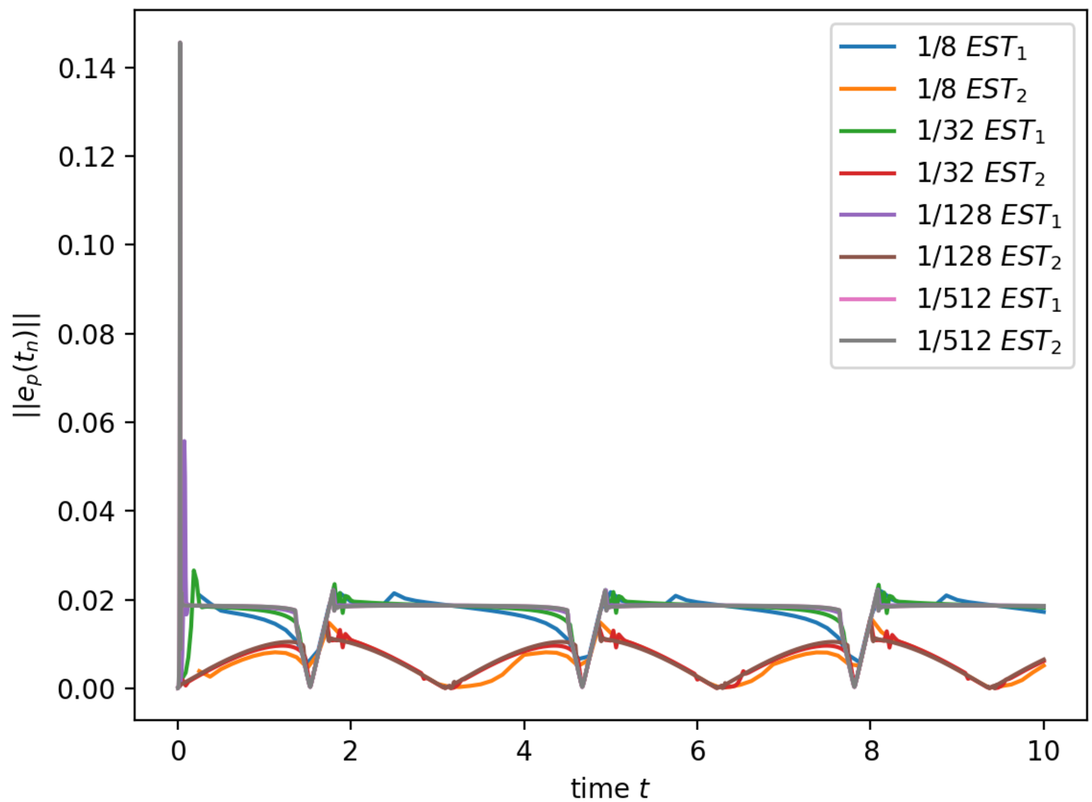

Figure 13: The pressure error for the fixed-Δt, variable ε test with a relative error estimator. On the y-axis is the error $\|p_n - p_{exact}\|$, and on the x axis is time, $0 \le t \le 10$. The plots for the old error estimator appear clustered above the plot for the new error estimator.

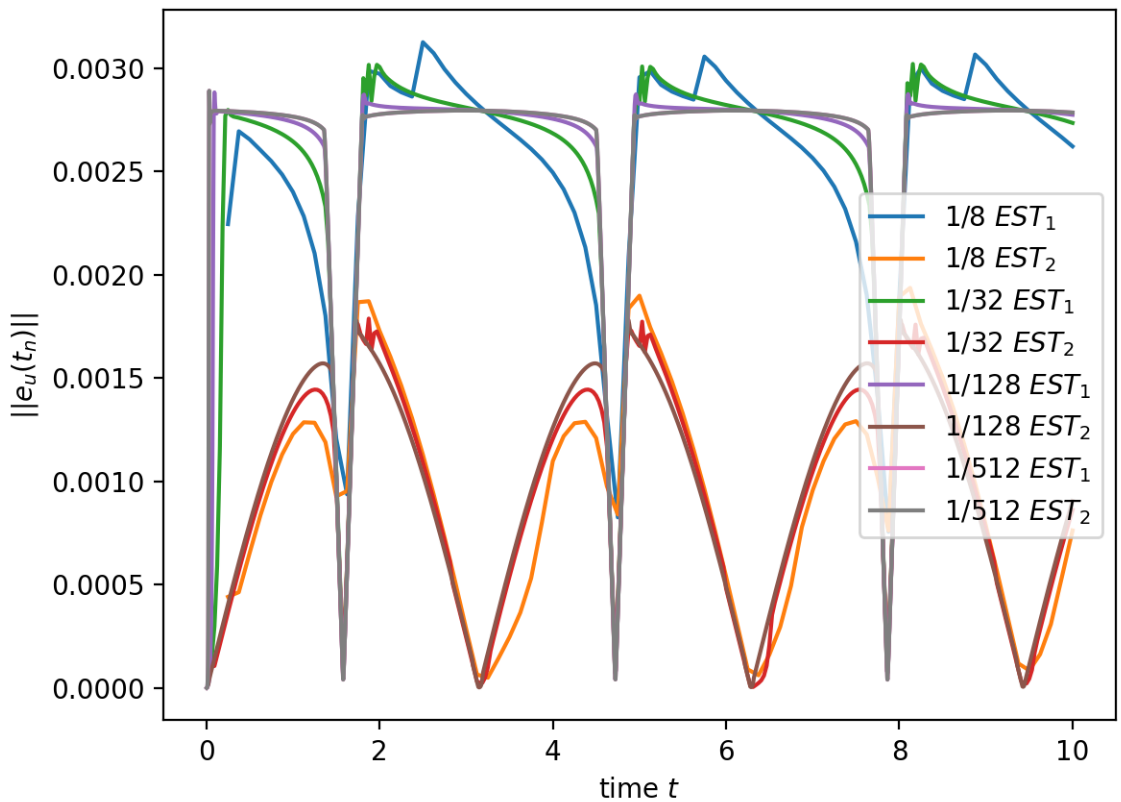

Figure 14: The velocity error for the fixed-Δt, variable ε test with a relative error estimator. On the y-axis is the error $\|u_n - u_{exact}\|$, and on the x axis is time, $0 \leq t \leq 10$. The plots for the old error estimator appear clustered above the plot for the new error estimator.

3.3.3 Test of the doubly adaptive scheme

Consider, as before, the unit square $(0,1) \times (0,1) \subset \mathbb{R}^2$, discretized by a mesh with 100 nodes on each edge, with $\nu = 1$. Let $0 \leq t \leq 2$, and the test problem from [3],

$$u = \pi \sin t(\sin 2\pi y \sin^2 \pi x, -\sin 2\pi x \sin^2 \pi y)$$

$$p = \cos t \cos \pi x \sin \pi y$$

Now, Δt and ε vary autonomously, using the adaptive scheme given in 3.2. Note that the estimator EST_2 for the error in the continuity equation is $||\nabla \cdot u^{n+1}||$. The test was run for 5 iterations from $0 \le t \le 2$, where for each iteration $i = 1, 2, 3, 4, 5$ the tolerance $TOL_1 = TOL_2 = 10^{-0.5*(i+1)}$.

In the following table, the Δt column lists the average timestep over that iteration.

Table 1: Rates of convergence for doubly adaptive test.

Iteration	Δt	$\lvert\lvert\lvert u(t_n) - u\rvert\rvert\rvert_{\ell^\infty L^2}$	Rate	$\lvert\lvert\lvert p(t_n) - p\rvert\rvert\rvert_{\ell^1 L^2}$	Rate
1	0.1	$6.92e-03$	...	$2.68e-01$	...
2	0.0895	$5.65e-03$	1.83	$2.66e-01$	$6.02e-02$
3	0.0623	$2.99e-03$	1.76	$2.44e-01$	$2.43e-01$
4	0.0364	$9.688e-04$	2.10	$2.37e-01$	$5.71e-02$
5	0.0187	$2.91e-04$	1.80	$2.38e-01$	$-7.65e-03$

The following plots are of the evolution of the timestep Δt and evolution of ε, where $0 \le t \le 2$, during the last iteration $i = 5$.

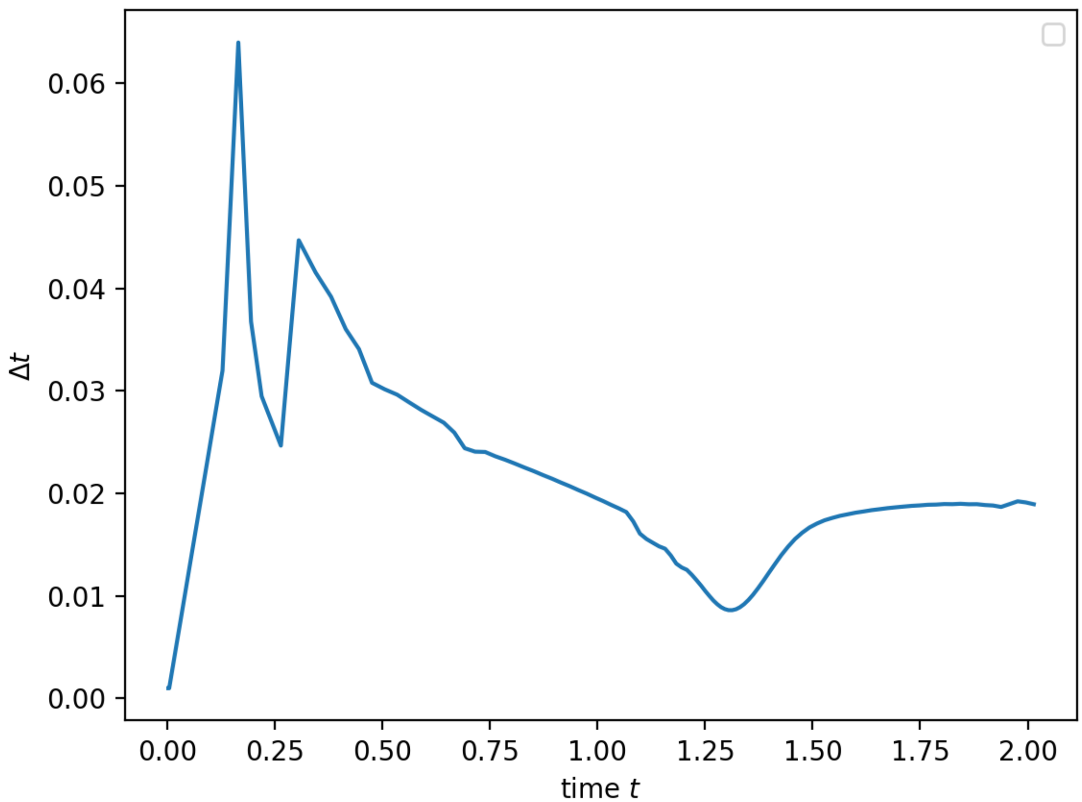

Figure 15: For the doubly adaptive test, a plot of the evolution of the timestep on the final iteration, with $0 \leq t \leq 2$

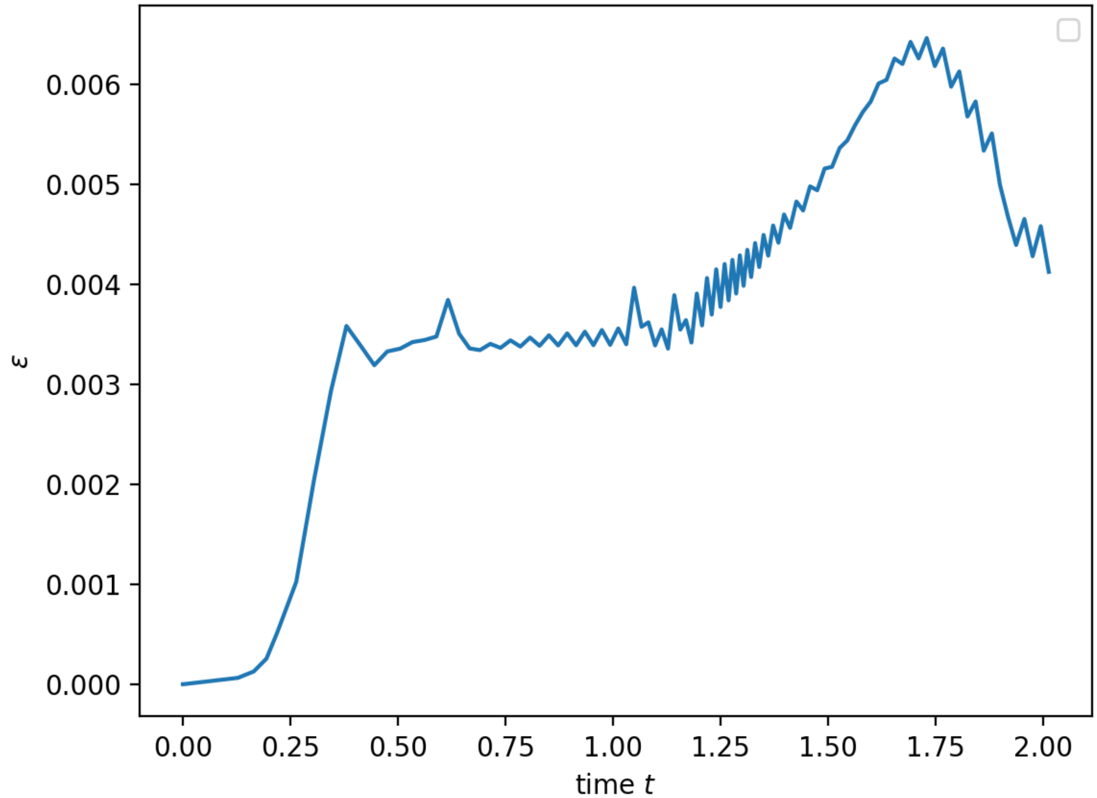

Figure 16: For the doubly adaptive test, a plot of the evolution of ε on the final iteration, with $0 \leq t \leq 2$

The following plots are of the velocity and pressure L^2 error, where $0 \leq t \leq 2$, during the last iteration $i = 5$.

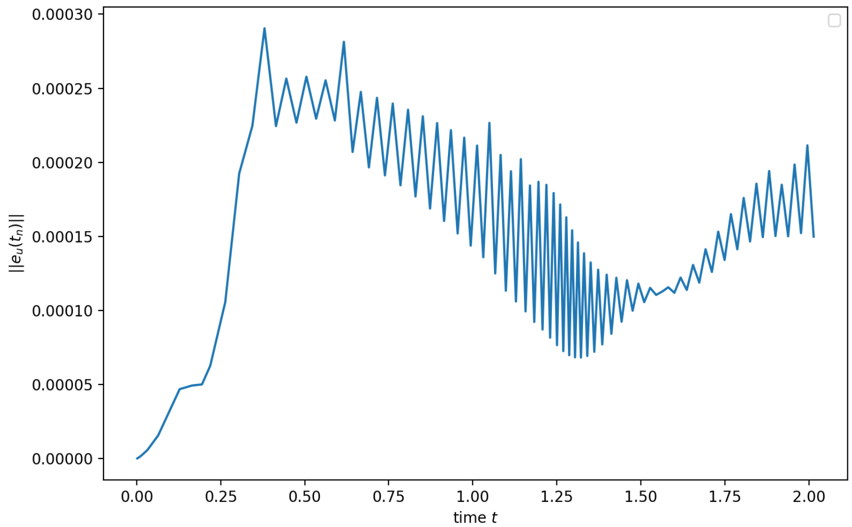

Figure 17: For the doubly adaptive test, a plot of the velocity error in the final iteration, with $0 \leq t \leq 2$ and the L^2 velocity error on the y-axis.

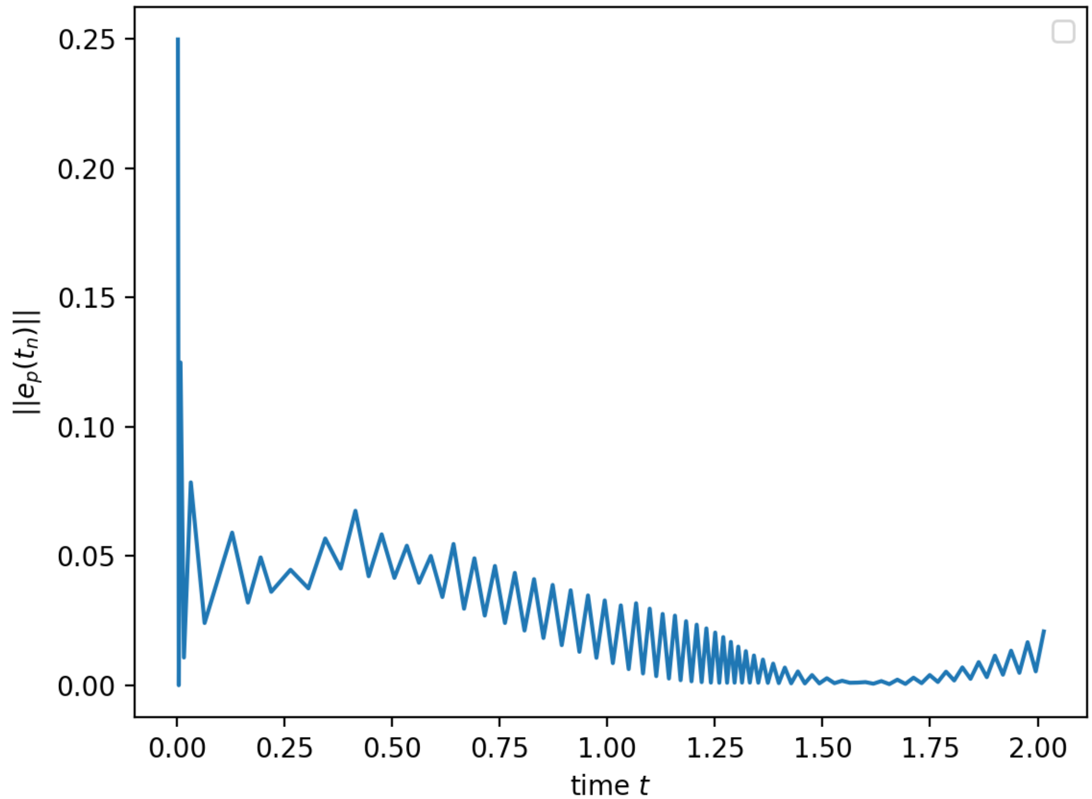

Figure 18: For the doubly adaptive test, a plot of the pressure error in the final iteration, with $0 \leq t \leq 2$ and the L^2 pressure error on the y-axis.

After an initial shock, the pressure error oscillates as before, with a valley in the error graph observed around $t = \pi/2$, where the exact solution $p = 0$.

4.0 Conclusions

The penalty method is a well-studied method that has been used in the numerical solution of flow problems since at least the 1970s [9]. It is a first-order, unconditionally stable method, as shown in error analysis [10] and with the associated energy inequality derived in Section 2.1.2. Computationally, when using the penalty method, the velocity is first solved for using a single solve and then the pressure is updated separately. An adaptive scheme can be added to a method with a few lines of code to change the parameters of the problem autonomously. In the case of the penalty method, two parameters – Δt and ε – can be adapted, using error estimators for the error in the momentum and continuity equations.

Looking at a test problem with oscillatory behavior, when adapting the penalty parameter ε with a fixed timestep Δt, the adaptation of ε appears to capture the behavior of the true solution when looking at graphs of its evolution. When using a relative error estimator as opposed to an absolute error estimator, looking at graphs of the associated error, the relative error estimator appears to perform better. Next, looking at the same test problem with a doubly adaptive scheme, the timestep Δt and ε adapt independently. There are associated increases in the value of ε and dips in the pressure error when the true solution $p = 0$ as before. The adaptive scheme is also shown to be unconditionally stable by deriving the energy equality and inequality, although further analysis of the error is needed.

Future experiments of this scheme could include testing different error estimators EST and different values for the tolerance TOL, as well as using different test problems with varying properties such as a flow between offset circles problem.

Bibliography

[1] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[2] Joseph Anthony Fiordilino and Sarah Khankan. Ensemble timestepping algorithms for natural convection. *Int. J. Numer. Anal. Model.*, 15(4-5):524–551, 2018.

[3] J. L. Guermond, P. Minev, and Jie Shen. An overview of projection methods for incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 195(44-47):6011–6045, 2006.

[4] Ahmet Guzel and William Layton. Time filters increase accuracy of the fully implicit method. *BIT*, 58(2):301–315, 2018.

[5] William Layton. *Introduction to the numerical analysis of incompressible viscous flows*, volume 6. Siam, 2008.

[6] William Layton and Michael McLaughlin. Doubly-adaptive artificial compression methods for incompressible flow. *Journal of Numerical Mathematics*, 2019.

[7] William Layton and Myron Sussman. *Numerical Linear Algebra*. 2014.

[8] David Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*, volume 228. Springer International, 2016.

[9] JN Reddy. On penalty function methods in the finite-element analysis of flow problems. *International Journal for Numerical Methods in Fluids*, 2, 1982.

[10] Jie Shen. On error estimates of the penalty method for unsteady Navier-Stokes equations. *SIAM J. Numer. Anal.*, 32(2):386–403, 1995.