A Nodal-Based Implementation of a Stabilized Finite Element Method for Incompressible Flow Problems

R. Codina

Publication CIMNE Nº PB-156, February 1999

A Nodal-Based Implementation of a Stabilized Finite Element Method for Incompressible Flow Problems

R. Codina

Publication CIMNE Nº PB-156, February 1999

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

A nodal-based implementation of a stabilized finite element method for incompressible flow problems

Ramon Codina

Universitat Politècnica de Catalunya,
Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain
codina@cimne.upc.es

Abstract

The objective of this paper is twofold. First, a stabilized finite element method for the incompressible Navier-Stokes is presented, and several numerical experiments are conducted to check its performance. This method is able to deal with all the instabilities that the standard Galerkin method presents, namely, the pressure instability, the instability arising in convection dominated situations and also the less popular instabilities found when the Navier-Stokes equations have a dominant Coriolis force, or there is a dominant absorption term arising from the small permeability of the medium where the flow takes place.

The second objective is to describe a nodal-based implementation of the finite element formulation introduced. This implementation is based on an a priori calculation of the integrals appearing in the formulation and then the construction of the matrix and right-hand-side vector of the final algebraic system to be solved. After appropriate approximations, this matrix and this vector can be constructed directly for each nodal point, without the need to loop over the elements and thus making the calculations much faster. In order to be able to do this, all the variables have to be defined at the nodes of the finite element mesh, not on the elements. This is also so for the stabilization parameters of the formulation. However, doing this gives rise to questions regarding the consistency and the conservation properties of the final scheme that are addressed in this paper.

Key Words: Finite elements, incompressible Navier-Stokes equations, stabilization methods, mesh graph

Contents

1 Introduction

This paper presents a finite element formulation for solving the generalized incompressible Navier-Stokes equations, including Coriolis forces and the effect of the permeability of the medium. The formulation is based on the sub-grid scale concept, which was introduced for the scalar convection-diffusion equation in [1] and generalized to systems of convection-diffusion-reaction equations in [2]. This formulation applied to the generalized Navier-Stokes equations is presented also in [3], where the linearized problem is analyzed. The numerical analysis undertaken in this reference shows that the method is optimally convergent and that it is able to deal with the instability problems of the standard Galerkin approach. Here, some standard benchmark problems are presented using this formulation in situations outside the scope of the classical analysis, such as thermally coupled flows or flow of nonlinear materials. Likewise, the method is extended to transient problems.

The numerical instabilities of the Galerkin method that the present formulation circumvents come from very different sources. The first one is the classical inf-sup condition for the velocity-pressure finite element interpolations, which is needed in order to have pressure stability (see, e.g. [4]). The second is also classical and concerns the oscillations found when the flow is dominated by convection, that is, the cell Reynolds number is large. Finally, when the viscosity is small compared either to the Coriolis forces or to the absorption effects coming from the medium permeability, numerical oscillations may also appear (see [3] for a more detailed description of the numerical instabilities arising in these cases). The stabilized formulation presented here is able to deal with all these numerical problems, allowing in particular the use of equal velocity-pressure interpolation, assumed throughout in the paper. The idea of using a stabilized method able to deal with the pressure instability and convection dominated flows is old and is in fact the origin of the Galerkin/least-squares method [5]. Extensions of this method to the transient incompressible Navier-Stokes equations are presented for example in [6, 7], among others.

Although the flexibility, generality and sound theoretical foundations of finite element methods applied to fluid flow have been widely acknowledged, they have been also blamed for being difficult to implement, leading to time consuming numerical codes. Perhaps this is why many commercial codes use finite volume rather than finite element methods, despite of the fact that the latter offer a wider range of possibilities, including higher order approximations. The second objective of this paper is to present a non-standard implementation of the finite element formulation presented herein. After computing the volume integrals of the products of the shape functions and its derivatives, the system matrix and force vector of the resulting algebraic system are obtained from them. This is done at each iteration and at each time step, without the need to recompute volume integrals any more. However, some approximations are required to do this. These approximations and their implications are described in this paper.

Whereas for a standard finite element implementation the normal flow of the calculation involves a loop over the elements, the calculation of the element contributions (to the system matrix and to the force vector) and the assembly of these into the global arrays, the flow of the calculations for the algorithm presented here is very different. The system matrix is formed of block matrices corresponding to the nodal points that can be obtained directly, without any reference to the elements. This is done by looping over the nodes of the finite element mesh and then over the nodes connected to a given nodal point. This involves the storage of the graph of the mesh, as well as the graph of the boundary mesh when the contributions from the boundaries need to be accounted for. It has to be stressed that, in principle, this approach is independent of the element type, and it is not restricted to linear simplicial elements as the edge-based implementations described for example in [8, 9, 10].

The nodal-based implementation of the finite element formulation necessitates of some approximations. First of all, all the variables need to be defined at each nodal point and the integrals where they appear have to be computed making using of their nodal values only. This is in particular true for the stabilization parameters of the formulation. Whereas taking them as constant over each element leads unnoticedly to the consistency and the conservativity of the formulation, these two properties might be lost when the stabilization paramenters are considered to vary continuously through the interpolation of their nodal values.

The paper is organized as follows. In the following section, the general problem for a thermally coupled incompressible fluid is presented. The numerical approximation is discussed in Section 3, which includes a simple time discretization, the linearization of the equations and the finite element approximation using a stabilized formulation. Section 4 presents the nodal-based implementation of this formulation, including the approximations needed, a discussion about the consistency and conservativity of the scheme, a description of the way to store the mesh and boundary graphs and the final algorithm itself. Numerical examples are presented in Section 5. They have been chosen as representative of the approximations involved in the scheme. Results turn out to be very similar to those obtained using a standard element-based implementation, but requiring much less CPU time.

2 Problem statement

In this section we shall consider the flow problem for an incompressible fluid in laminar regime and taking into account several physical effects. These include the fact that the reference frame where the computational domain is attached varies in time, the coupling of the Navier-Stokes equations with the heat transport equation through Boussinesq assumption, the permeability of the medium where the fluid flows, and a generalized Newtonian behavior of this fluid, allowing its viscosity to depend on the invariants of the strain rate tensor.

The equations describing the problem are

$$\partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\,\boldsymbol{\omega}\times\boldsymbol{u} - 2\nabla\cdot[\nu\boldsymbol{\varepsilon}(\boldsymbol{u})] + \nabla p + \sigma\boldsymbol{u} + \beta\boldsymbol{g}\vartheta = \boldsymbol{f}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

$$\partial_t\vartheta + (\boldsymbol{u}\cdot\nabla)\vartheta - \nabla\cdot(\kappa\nabla\vartheta) = S, \tag{3}$$

to be solved in $\Omega\times(0,t_{\text{fin}})$, where $\Omega\subset\mathbb{R}^{n_{\text{sd}}}$ ($n_{\text{sd}}=2$ or 3 being the space dimension) is the computational domain and $[0,t_{\text{fin}}]$ is the time interval to be considered. In (1)-(3), $\boldsymbol{u}$ denotes the velocity field, p is the kinematic pressure (i.e., the pressure divided by the density), ϑ is the temperature, ν is the kinematic viscosity, which may depend on the invariants of the symmetrical part of the velocity gradient $\boldsymbol{\varepsilon}(\boldsymbol{u})$, $\boldsymbol{\omega}$ is the velocity of rotation of the frame of reference (and thus $2\,\boldsymbol{\omega}\times\boldsymbol{u}$ is the Coriolis force), σ is the inverse of the permeability of the medium, β is the thermal expansion coefficient, $\boldsymbol{g}$ is the gravity acceleration vector, $\boldsymbol{f}$ is the vector of body forces, κ is the thermal diffusivity (that is, the thermal conductivity divided by the heat capacity) and S is the heat source. The density ρ_0 is assumed constant to obtain equations (1)-(3).

In the most general case, the force vector $\boldsymbol{f}$ in (1) contains the acceleration terms coming from the temporal variation of the reference basis and the reference buoyancy forces from the Boussinesq assumption, that is

$$\boldsymbol{f} = \boldsymbol{g}(1+\beta\vartheta_0) - \boldsymbol{a}_{\text{fr}} - \dot{\boldsymbol{\omega}}\times\boldsymbol{x} - \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{x}). \tag{4}$$

In this equation, ϑ_0 is the reference temperature from which buoyancy forces are computed, $\boldsymbol{a}_{\text{fr}}$ is the acceleration of the frame of the reference measured from an inertial system and expressed in the moving reference, and the vector of position in this moving reference has been denoted by $\boldsymbol{x} = (x_1,x_2,x_3)\equiv(x,y,z)$ in the 3D case. Here and below, a Cartesian coordinate system is assumed.

The rheological behavior that will be assumed for the fluid in one of the numerical examples is the power law, which is perhaps the most common constitutive equation for generalized Newtonian fluids. The expression of this law is

$$\nu = \rho_0^{-1}K_0\left[4I_2(\boldsymbol{\varepsilon})\right]^{(n-1)/2}. \tag{5}$$

Here, K_0 and n are physical constants (K_0 is the material consistency and n the rate sensitivity) and $I_2(\boldsymbol{\varepsilon})$ is the second invariant of the strain rate tensor, $I_2(\boldsymbol{\varepsilon}) := \boldsymbol{\varepsilon}:\boldsymbol{\varepsilon}/2$, the colon standing for the double contraction of second order tensors.

The reasons for considering such a variety of physical phenomena in the model are the following. Firstly, Coriolis forces and permeability effects have been introduced only because of generality. When they dominate, numerical oscillations may appear, global for the former, reduced to boundary layers for the latter. The stabilized method presented here allows to remove them, as it was shown in [3].

The nodal-based implementation described later on is based on a particular way of writing the convective terms in the Navier-Stokes and the heat equation. In order to see the effect on both, it is interesting to solve thermally coupled flows.

Finally, the possibility of dealing with nonlinear materials will be used to demonstrate the effect of another approximation that will be done for the viscous term. This is especially relevant not only in the case of a non Newtonian behavior of the fluid, but also in the case of turbulent flows. The most common turbulent models make use of the Boussinesq assumption, which leads to the introduction of an additional viscosity to be added to the physical one. This turbulent viscosity varies from point to point according to the law that the turbulence model being employed determines. Nonlinear fluids as those considered in this paper can be considered as representative of fluids with non constant viscosity.

In order to write the boundary conditions for equations (1)-(3), consider the boundary $\Gamma = \partial\Omega$ split into two sets of disjoint components as $\Gamma = \overline{\Gamma_{\rm dv} \cup \Gamma_{\rm nv}}$ and also as $\Gamma = \overline{\Gamma_{\rm dt} \cup \Gamma_{\rm nt}}$, where $\Gamma_{\rm dv}$ and $\Gamma_{\rm dt}$ are the parts of the boundary with Dirichlet type boundary conditions for the velocity and the pressure, respectively, and $\Gamma_{\rm nv}$ and $\Gamma_{\rm nt}$ are those where Neumann type conditions are prescribed. If the Cauchy stress tensor (divided by the density) is written as $\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\nu\boldsymbol{\varepsilon}(\boldsymbol{u})$ and prescribed values are represented by an over-bar, the boundary conditions to be considered are

$$\boldsymbol{u}(\boldsymbol{x},t) = \bar{\boldsymbol{u}}(\boldsymbol{x},t) \qquad \text{on } \Gamma_{\rm dv}, \tag{6}$$

$$\boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{x},t) = \bar{\boldsymbol{t}}(\boldsymbol{x},t) \qquad \text{on } \Gamma_{\rm nv}, \tag{7}$$

$$\vartheta(\boldsymbol{x},t) = \bar{\vartheta}(\boldsymbol{x},t) \qquad \text{on } \Gamma_{\rm dt}, \tag{8}$$

$$\kappa\,\boldsymbol{n}\cdot\nabla\vartheta(\boldsymbol{x},t) = \bar{h}(\boldsymbol{x},t) \qquad \text{on } \Gamma_{\rm nt}, \tag{9}$$

for $t \in (0, t_{\rm fin})$.

To close the problem, initial conditions have to be appended to equations (1)-(3) and the boundary conditions (6)-(9). They are of the form $\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}^0(\boldsymbol{x})$, $\vartheta(\boldsymbol{x},0) = \vartheta^0(\boldsymbol{x})$ for $\boldsymbol{x} \in \Omega$, where $\boldsymbol{u}^0(\boldsymbol{x})$ is a given initial velocity and $\vartheta^0(\boldsymbol{x})$ a given initial temperature.

3 Numerical approximation

3.1 Time discretization

Let us consider now the temporal discretization of equations (1)-(3), for which we use the generalized trapezoidal rule. Let $0 = t^0 < t^1 < \ldots < t^N = t_{\rm fin}$ be a partition of the time interval and $\alpha \in [0,1]$. To simplify the notation, we shall take the time step size $\delta t := t^{n+1} - t^n$ constant for all n. Let us also introduce the notation

$$\delta f^n := f^{n+1} - f^n,$$
$$f^{n+\alpha} := \alpha f^{n+1} + (1-\alpha) f^n,$$

$$\delta_t f^n := \frac{\delta f^n}{\delta t},$$

where f is a generic function of time and f^n denotes the value of f at time t^n or an approximation to it.

Let us assume for simplicity that the force vector $\boldsymbol{f}$ and the heat source S are continuous in time. The generalized trapezoidal rule applied to equations (1)-(3) leads to the following time discrete problem: from known $\boldsymbol{u}^n$ and ϑ^n, find $\boldsymbol{u}^{n+1}$, p^{n+1} and ϑ^{n+1} such that

$$\begin{aligned}
&\delta_t \boldsymbol{u}^n + (\boldsymbol{u}^{n+\alpha} \cdot \nabla)\boldsymbol{u}^{n+\alpha} + 2\,\boldsymbol{\omega}^{n+\alpha} \times \boldsymbol{u}^{n+\alpha} - 2\nabla \cdot [\nu^{n+\alpha}\boldsymbol{\varepsilon}(\boldsymbol{u}^{n+\alpha})] + \nabla p^{n+1} \\
&\qquad + \sigma \boldsymbol{u}^{n+\alpha} + \boldsymbol{g}\beta\vartheta^{n+\alpha} = \boldsymbol{f}^{n+\alpha}, \qquad (10)\\
&\nabla \cdot \boldsymbol{u}^{n+1} = 0, \qquad (11)\\
&\delta_t \vartheta^n + (\boldsymbol{u}^{n+\alpha} \cdot \nabla)\vartheta^{n+\alpha} - \nabla \cdot (\kappa \nabla \vartheta^{n+\alpha}) = S^{n+\alpha}, \qquad (12)
\end{aligned}$$

and satisfying the boundary conditions

$$\begin{aligned}
&\boldsymbol{u}^{n+\alpha}(\boldsymbol{x}) = \bar{\boldsymbol{u}}^{n+\alpha}(\boldsymbol{x}) \qquad \text{on } \Gamma_{\text{dv}}, \qquad (13)\\
&\boldsymbol{n} \cdot \boldsymbol{\sigma}^{n+\alpha}(\boldsymbol{x}) = \bar{\boldsymbol{t}}^{n+\alpha}(\boldsymbol{x}) \qquad \text{on } \Gamma_{\text{nv}}, \qquad (14)\\
&\vartheta^{n+\alpha}(\boldsymbol{x}) = \bar{\vartheta}^{n+\alpha}(\boldsymbol{x}) \qquad \text{on } \Gamma_{\text{dt}}, \qquad (15)\\
&\kappa\, \boldsymbol{n} \cdot \nabla \vartheta^{n+\alpha}(\boldsymbol{x}) = \bar{h}^{n+\alpha}(\boldsymbol{x}) \qquad \text{on } \Gamma_{\text{nt}}. \qquad (16)
\end{aligned}$$

Observe that the problem can be posed in terms of $\boldsymbol{u}^{n+\alpha}$ and $\vartheta^{n+\alpha}$ rather than $\boldsymbol{u}^{n+1}$ and ϑ^{n+1} by using the fact that $\delta f^n = \left(f^{n+\alpha} - f^n\right)/\alpha$, for any function f.

The values of interest of the parameter α are $\alpha = 1/2$ and $\alpha = 1$, corresponding to the Crank-Nicholson and the backward Euler schemes, respectively. Both are unconditionally stable, although the former is expected to be second order accurate whereas only a first order approximation can be expected for the latter. Under certain regularity assumptions, this is known to hold at least for the standard Galerkin method [11], although no analysis is available for the stabilized formulation presented below (see [12] for a similar approach to the transient problem for the convection-diffusion equation).

3.2 Weak form

Once the time discretization has been carried out, we are left with the boundary value problem defined by the differential equations (10)-(12) and the boundary conditions (13)-(16). This problem will be discretized in space using a finite element method. It is therefore necessary to obtain first the variational form of the boundary value problem. For that, let us consider the functional spaces

$$\begin{aligned}
V^{n+\alpha} &= \left\{ \boldsymbol{u}(\boldsymbol{x}) \in H^1(\Omega)^{n_{\text{sd}}} \mid \boldsymbol{u} = \bar{\boldsymbol{u}}^{n+\alpha} \text{ on } \Gamma_{\text{dv}} \right\},\\
V_0 &= \left\{ \boldsymbol{u}(\boldsymbol{x}) \in H^1(\Omega)^{n_{\text{sd}}} \mid \boldsymbol{u} = \boldsymbol{0} \text{ on } \Gamma_{\text{dv}} \right\},
\end{aligned}$$

$$Q = \left\{ q(\boldsymbol{x}) \in L^2(\Omega) \mid \int_\Omega q \, \mathrm{d}\Omega = 0 \text{ if } \Gamma_{\mathrm{nv}} = \emptyset \right\},$$
$$\Psi^{n+\alpha} = \left\{ \vartheta(\boldsymbol{x}) \in H^1(\Omega) \mid \vartheta = \bar{\vartheta}^{n+\alpha} \text{ on } \Gamma_{\mathrm{dt}} \right\},$$
$$\Psi_0 = \left\{ \vartheta(\boldsymbol{x}) \in H^1(\Omega) \mid \vartheta = 0 \text{ on } \Gamma_{\mathrm{dt}} \right\},$$

where, as usual, $L^2(\Omega)$ denotes the space of square integrable functions in the domain Ω and $H^1(\Omega)$ the subspace of $L^2(\Omega)$ of functions with square integrable first derivatives.

Once these functional spaces have been introduced, the week form of the problem defined by (10)-(12) and (13)-(16) reads as follows: find $\boldsymbol{u}^{n+\alpha} \in V^{n+\alpha}$, $p^{n+1} \in Q$ and $\vartheta^{n+\alpha} \in \Psi^{n+\alpha}$ such that

$$\begin{aligned}
&\int_\Omega \boldsymbol{v} \cdot \left[\delta_t \boldsymbol{u}^n + (\boldsymbol{u}^{n+\alpha} \cdot \nabla)\boldsymbol{u}^{n+\alpha} + 2\,\boldsymbol{\omega}^{n+\alpha} \times \boldsymbol{u}^{n+\alpha} + \sigma \boldsymbol{u}^{n+\alpha} + \boldsymbol{g}\beta\vartheta^{n+\alpha} \right] \mathrm{d}\Omega \\
&\quad + \int_\Omega 2\boldsymbol{\varepsilon}(\boldsymbol{v}) : \nu^{n+\alpha} \boldsymbol{\varepsilon}(\boldsymbol{u}^{n+\alpha}) \, \mathrm{d}\Omega - \int_\Omega p^{n+1} \nabla \cdot \boldsymbol{v} \, \mathrm{d}\Omega \\
&\quad = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{f}^{n+\alpha} \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{nv}}} \boldsymbol{v} \cdot \bar{\boldsymbol{t}}^{n+\alpha} \, \mathrm{d}\Gamma, \qquad (17)
\end{aligned}$$

$$\int_\Omega q \nabla \cdot \boldsymbol{u}^{n+\alpha} \, \mathrm{d}\Omega = 0, \qquad (18)$$

$$\begin{aligned}
&\int_\Omega \psi \left[\delta_t \vartheta^n + (\boldsymbol{u}^{n+\alpha} \cdot \nabla)\vartheta^{n+\alpha} \right] \mathrm{d}\Omega + \int_\Omega \kappa \nabla\psi \cdot \nabla\vartheta^{n+\alpha} \, \mathrm{d}\Omega \\
&\quad = \int_\Omega \psi S \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{nt}}} \psi \, h^{n+\alpha} \, \mathrm{d}\Gamma, \qquad (19)
\end{aligned}$$

for all test functions $\boldsymbol{v} \in V_0$, $q \in Q$ and $\psi \in \Psi_0$.

3.3 Linearized equations and iterative coupling

The final step previous to the finite element approximation of the nonlinear variational problem (17)-(19) is to linearize it. In our case, there are two sources of nonlinearity, namely, the convective term of the Navier-Stokes and the heat equations and the fact that the viscosity depends in a nonlinear way on the viscosity through a non Newtonian constitutive model.

Concerning the linearization of the convective term of the Navier-Stokes equations, both the fixed point Picard scheme and the Newton-Raphson method can be considered. Let us denote by the superscript i the iteration counter. Given an approximation $\boldsymbol{u}^{n+\alpha,i}$ to $\boldsymbol{u}^{n+\alpha}$, the next one is computed by approximating

$$\begin{aligned}
&(\boldsymbol{u}^{n+\alpha,i+1} \cdot \nabla)\boldsymbol{u}^{n+\alpha,i+1} \approx (\boldsymbol{u}^{n+\alpha,i} \cdot \nabla)\boldsymbol{u}^{n+\alpha,i+1} \\
&\quad + \lambda_1 \left[(\boldsymbol{u}^{n+\alpha,i+1} \cdot \nabla)\boldsymbol{u}^{n+\alpha,i} - (\boldsymbol{u}^{n+\alpha,i} \cdot \nabla)\boldsymbol{u}^{n+\alpha,i} \right], \qquad (20)
\end{aligned}$$

where $\lambda_1 = 0$ corresponds to the Picard method and $\lambda_1 = 1$ to the Newton-Raphson scheme. The former is first order, whereas the latter is second order but needs an initial guess close enough to the converged solution to converge.

The linearization of the constitutive model (5) is not as easy as that of the convective term. In fact, the expression of the viscosity in terms of the velocity in this case is not even differentiable. In such a situation, the simplest way to deal with the constitutive nonlinearity is to use a Picard-like strategy, taking the viscosity evaluated with the velocity $\boldsymbol{u}^{n+\alpha,i}$ when $\boldsymbol{u}^{n+\alpha,i+1}$ is to be computed. Of course only a linear convergence rate can be expected if this is done, and the use of Newton's scheme for the convective term seems useless. However, this depends on the relative importance of the nonlinearities coming from the convective and the viscous terms. In some situations where the influence of the constitutive law is small and the flow is governed by convection, we have found useful to use $\lambda_1 = 1$ in (20), even though the simple fixed point scheme is used for the viscosity.

The convective term in the heat equation is another nonlinearity of the problem. However, instead of linearizing this term and dealing with the fully coupled problem, with velocity, pressure and temperature as unknowns, we shall use an iterative coupling, as described for example in [13]. The idea is to use the temperature known from the previous iteration in the momentum equation (17), and then use this equation and (18) to compute the velocity and the pressure. With the velocity computed, we can proceed to solve the heat transport equation. Again, only a linear convergence rate can be expected for this iterative scheme. However, if the nonlinear term of the momentum equation drives the iterative scheme, it can be useful to use $\lambda_1 = 1$ in (20).

Having the previous considerations in mind, the fully linearized form of problem (17)-(19), coupling iteratively the heat equation to the momentum and incompressibility equations, is as follows: given a guess $\boldsymbol{u}^{n+\alpha,i}$ for $\boldsymbol{u}^{n+\alpha}$ and $\vartheta^{n+\alpha,i}$ for $\vartheta^{n+\alpha}$, find $\boldsymbol{u}^{n+\alpha,i+1} \in V^{n+\alpha}$, $p^{n+1,i+1} \in Q$ and $\vartheta^{n+\alpha,i+1} \in \Psi^{n+\alpha}$ such that

$$
\begin{aligned}
&\int_\Omega \boldsymbol{v} \cdot \Big\{ \delta_t \boldsymbol{u}^{n,i+1} + 2\,\boldsymbol{\omega}^{n+\alpha} \times \boldsymbol{u}^{n+\alpha,i+1} + \sigma \boldsymbol{u}^{n+\alpha,i+1} + \boldsymbol{g}\beta\vartheta^{n+\alpha,i} \\
&\quad + (\boldsymbol{u}^{n+\alpha,i} \cdot \nabla)\boldsymbol{u}^{n+\alpha,i+1} + \lambda_1 \left[(\boldsymbol{u}^{n+\alpha,i+1} \cdot \nabla)\boldsymbol{u}^{n+\alpha,i} - (\boldsymbol{u}^{n+\alpha,i} \cdot \nabla)\boldsymbol{u}^{n+\alpha,i} \right] \Big\} \, \mathrm{d}\Omega \\
&\quad + \int_\Omega 2\boldsymbol{\varepsilon}(\boldsymbol{v}) : \nu^{n+\alpha,i} \boldsymbol{\varepsilon}(\boldsymbol{u}^{n+\alpha,i+1}) \, \mathrm{d}\Omega - \int_\Omega p^{n+1,i+1} \nabla \cdot \boldsymbol{v} \, \mathrm{d}\Omega \\
&\quad = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{f}^{n+\alpha} \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{nv}}} \boldsymbol{v} \cdot \bar{\boldsymbol{t}}^{n+\alpha} \, \mathrm{d}\Gamma, \qquad (21)
\end{aligned}
$$

$$
\int_\Omega q \nabla \cdot \boldsymbol{u}^{n+\alpha,i+1} \, \mathrm{d}\Omega = 0, \qquad (22)
$$

$$
\begin{aligned}
&\int_\Omega \psi \left[\delta_t \vartheta^{n,i+1} + (\boldsymbol{u}^{n+\alpha,i+1} \cdot \nabla)\vartheta^{n+\alpha,i+1} \right] \mathrm{d}\Omega + \int_\Omega \kappa \nabla\psi \cdot \nabla\vartheta^{n+\alpha,i+1} \, \mathrm{d}\Omega \\
&\quad = \int_\Omega \psi S^{n+\alpha,i+1} \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{nt}}} \psi \, h^{n+\alpha} \, \mathrm{d}\Gamma, \qquad (23)
\end{aligned}
$$

for all test functions $\boldsymbol{v} \in V_0$, $q \in Q$ and $\psi \in \Psi_0$. Here, $\nu^{n+\alpha,i}$ means that the viscosity is evaluated with $\boldsymbol{u}^{n+\alpha,i}$, whereas $S^{n+\alpha,i+1}$ is evaluated with $\boldsymbol{u}^{n+\alpha,i+1}$ (S can depend on $\boldsymbol{u}$

through Joule's effect, for instance). Likewise,

$$\frac{\delta \boldsymbol{u}^{n,i+1}}{\delta t} = \frac{1}{\alpha \delta t}\left(\boldsymbol{u}^{n+\alpha,i+1} - \boldsymbol{u}^{n}\right).$$

After (21)-(23) has been solved, convergence needs to be checked and, if not achieved, set $i \leftarrow i+1$ and solve this problem again.

3.4 Finite element approximation

We are now in a position to undertake the finite element approximation of the linear variational problem (21)-(23). In order to discretize it in space, let $\{\Omega^e\}$ be a finite element partition of the domain Ω, with index e ranging from 1 to the number of elements n_{el}. We denote with a subscript h the finite element approximation to the unknown functions, and by $\boldsymbol{v}_h$, q_h and ψ_h the velocity, pressure and temperature test functions associated to $\{\Omega^e\}$.

The finite element approximation to the functional spaces where the unknowns and the test functions belong are also characterized by a subscript h. A very important point is that we are interested in *using equal interpolation for all the unknowns* (velocity, pressure and temperature). Therefore, all the finite element spaces are assumed to be built up using the standard continuous interpolation functions.

In order to overcome the numerical problems of the standard Galerkin method, a stabilized finite element formulation to solve (21)-(23) is applied. This formulation is presented in [2] for the general case of systems of convection-diffusion-reaction equations, and applied to the incompressible Navier-Stokes equations in [3], where its convergence properties for the linearized problem are analyzed. The bottom line of the method is to test the continuous equations by the standard Galerkin test functions plus perturbations that depend on the operator representing the differential equation being solved. In our case, this operator corresponds to the linearized form of the time-discrete Navier-Stokes equations (10)-(11) and the heat equation (12). In this case, the method consists of finding $\boldsymbol{u}_h^{n+\alpha,i+1} \in V_h^{n+\alpha}$, $p_h^{n+1,i+1} \in Q_h$ and $\vartheta_h^{n+\alpha,i+1} \in \Psi_h^{n+\alpha}$ such that

$$\begin{aligned}
&\int_\Omega \boldsymbol{v}_h \cdot \boldsymbol{r}_{u1}^{n+\alpha,i+1}\,\mathrm{d}\Omega + \int_\Omega 2\boldsymbol{\varepsilon}(\boldsymbol{v}_h) : \nu^{n+\alpha,i}\boldsymbol{\varepsilon}(\boldsymbol{u}_h^{n+\alpha,i+1})\,\mathrm{d}\Omega - \int_\Omega p_h^{n+1,i+1}\nabla\cdot\boldsymbol{v}_h\,\mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\boldsymbol{\zeta}_{u1}^{n+\alpha,i}\cdot\left(\boldsymbol{r}_{u1}^{n+\alpha,i+1} + \boldsymbol{r}_{u2}^{n+\alpha,i+1}\right)\mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\zeta_{u2}^{n+\alpha,i}\, r_p^{n+\alpha,i+1}\,\mathrm{d}\Omega \\
&\quad = \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\left(\boldsymbol{v}_h + \boldsymbol{\zeta}_{u1}^{n+\alpha,i}\right)\cdot\boldsymbol{f}^{n+\alpha}\,\mathrm{d}\Omega + \int_{\Gamma_{\mathrm{nv}}}\boldsymbol{v}_h\cdot\bar{\boldsymbol{t}}^{n+\alpha}\,\mathrm{d}\Gamma, \qquad (24)\\
&\int_\Omega q_h r_p^{n+\alpha,i+1}\,\mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\boldsymbol{\zeta}_p^{n+\alpha,i}\cdot\left(\boldsymbol{r}_{u1}^{n+\alpha,i+1} + \boldsymbol{r}_{u2}^{n+\alpha,i+1}\right)\mathrm{d}\Omega \\
&\quad = \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\boldsymbol{\zeta}_p^{n+\alpha,i}\cdot\boldsymbol{f}^{n+\alpha}\,\mathrm{d}\Omega, \qquad (25)
\end{aligned}$$

$$\int_\Omega \psi_h \cdot r_{\vartheta 1}^{n+\alpha,i+1}\,\mathrm{d}\Omega + \int_\Omega \kappa\nabla\psi_h \cdot \nabla\vartheta_h^{n+\alpha,i+1}\,\mathrm{d}\Omega$$
$$+\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \zeta_\vartheta^{n+\alpha,i} \left(r_{\vartheta 1}^{n+\alpha,i+1} + r_{\vartheta 2}^{n+\alpha,i+1}\right)\mathrm{d}\Omega$$
$$= \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \left(\psi_h + \zeta_\vartheta^{n+\alpha,i}\right) S^{n+\alpha,i+1}\,\mathrm{d}\Omega + \int_{\Gamma_{\mathrm{nt}}} \psi_h\, h^{n+\alpha}\,\mathrm{d}\Gamma, \tag{26}$$

for all test functions $\boldsymbol{v}_h \in V_{0,h}$, $q_h \in Q_h$ and $\psi_h \in \Psi_{0,h}$, where

$$\boldsymbol{r}_{u1}^{n+\alpha,i+1} := \delta_t \boldsymbol{u}_h^{n,i+1} + 2\,\boldsymbol{\omega}^{n+\alpha} \times \boldsymbol{u}_h^{n+\alpha,i+1} + \sigma \boldsymbol{u}_h^{n+\alpha,i+1} + \boldsymbol{g}\beta\vartheta_h^{n+\alpha,i}$$
$$+(\boldsymbol{u}_h^{n+\alpha,i} \cdot \nabla)\boldsymbol{u}_h^{n+\alpha,i+1} + \lambda_1 \left[(\boldsymbol{u}_h^{n+\alpha,i+1} \cdot \nabla)\boldsymbol{u}_h^{n+\alpha,i} - (\boldsymbol{u}_h^{n+\alpha,i} \cdot \nabla)\boldsymbol{u}_h^{n+\alpha,i}\right], \tag{27}$$

$$\boldsymbol{r}_{u2}^{n+\alpha,i+1} := -2\nabla \cdot [\nu^{n+\alpha,i}\boldsymbol{\varepsilon}(\boldsymbol{u}_h^{n+\alpha,i+1})] + \nabla p_h^{n+1,i+1}, \tag{28}$$

$$r_p^{n+\alpha,i+1} := \epsilon p_h^{n+1,i+1} - \lambda_2 \epsilon p_h^{n+1,i} + \nabla \cdot \boldsymbol{u}_h^{n+\alpha,i+1}, \tag{29}$$

$$r_{\vartheta 1}^{n+\alpha,i+1} := \delta_t \vartheta_h^{n,i+1} + (\boldsymbol{u}_h^{n+\alpha,i+1} \cdot \nabla)\vartheta_h^{n+\alpha,i+1}, \tag{30}$$

$$r_{\vartheta 2}^{n+\alpha,i+1} := -\nabla \left(\kappa \cdot \nabla\vartheta_h^{n+\alpha,i+1}\right), \tag{31}$$

the functions $\boldsymbol{\zeta}_{u1}$, ζ_{u2} and $\boldsymbol{\zeta}_p$ are computed within each element as

$$\boldsymbol{\zeta}_{u1} = \tau_u \left\{(\boldsymbol{u}_h \cdot \nabla)\boldsymbol{v}_h + 2\,\boldsymbol{\omega} \times \boldsymbol{v}_h - \sigma \boldsymbol{v}_h + 2\nabla \cdot [\nu\boldsymbol{\varepsilon}(\boldsymbol{v}_h)]\right\}, \tag{32}$$

$$\zeta_{u2} = \tau_p \nabla \cdot \boldsymbol{v}_h, \tag{33}$$

$$\boldsymbol{\zeta}_p = \tau_u \nabla q_h, \tag{34}$$

$$\zeta_\vartheta = \tau_\vartheta \left[(\boldsymbol{u}_h \cdot \nabla)\psi_h + \nabla \cdot (\kappa \nabla \psi_h)\right], \tag{35}$$

and the parameters τ_u, τ_p and τ_ϑ are also computed element-wise as [3, 14]

$$\tau_u = \left[\frac{4\nu}{h^2} + \frac{2|\boldsymbol{u}_h|}{h} + |\,\boldsymbol{\omega}| + \sigma\right]^{-1}, \tag{36}$$

$$\tau_p = 4\nu + 2|\boldsymbol{u}_h|h + |\,\boldsymbol{\omega}|h^2 + \sigma h^2, \tag{37}$$

$$\tau_\vartheta = \left[\frac{4\kappa}{h^2} + \frac{2|\boldsymbol{u}_h|}{h}\right]^{-1}, \tag{38}$$

where h is the element size for linear elements and half of it for quadratics.

There are several remarks to be made to the previous equations:

REMARK 1. It is observed that (28) and (31) (the terms of the original differential equations integrated by parts in the weak form of the problem) involve second derivatives of the unknowns. This is why the integrals involving these terms have to be evaluated element by element.

REMARK 2. The term (28) involves also derivatives of the viscosity, in the case in which it is variable. These are very difficult to incorporate in the formulation, although in the next section it is discussed how to deal with the viscosity variation in the nodal-based implementation presented there.

REMARK 3. In (29) we have introduced a parameter ϵ that corresponds to a penalty parameter for the incompressibility constraint. When $\lambda_2 = 0$ the penalty strategy can be considered the 'classical' one. On the other hand, when $\lambda_2 = 1$ it is seen from (29) that the effect of the penalization disappears when convergence is achieved. This *iterative penalty* method is discussed and analyzed in [15]. The benefit of taking $\lambda_2 = 1$ is that larger values of ϵ may be used with a good approximation of the incompressibility constraint. The use of penalty methods is very useful when pressures are discontinuous, since then they can be eliminated at the element level. When continuous pressures are used, they may help to improve the convergence of the iterative methods if they are used to solve the algebraic system of equations.

REMARK 4. In expressions (32)-(38) the velocity $\boldsymbol{u}_h$, the speed of rotation $\boldsymbol{\omega}$ and the viscosity ν are evaluated at the time step and iteration indicated in (24)-(31).

REMARK 5. It is observed from (32)-(35) that these terms are precisely the adjoints of the (linearized) operators of the differential equations to be solved applied to the test functions (observe the signs of the viscous and permeability terms in (32) and of the diffusive term in (35)). This method corresponds to the algebraic version of the sub-grid scale approach [1, 2] and circumvents *all* the stability problems of the Galerkin method. In particular, in this case it is possible to use equal velocity pressure interpolations, that is, we are not tight to the satisfaction of the inf-sup stability condition. □

4 Nodal-based implementation

4.1 Motivation

In this section, the second objective of this paper is attempted, that is, to present a nodal-based finite element implementation of the stabilized finite element formulation presented in the previous section.

Let n_{pts} be the total number of nodes of the finite element mesh and let N^a be the shape function (i.e., the standard finite element interpolation function) associated to node a, $a = 1, ..., n_{\text{pts}}$. From now on, superscripts a and b will refer to the nodes of the mesh.

To obtain the algebraic version of problem (24)-(26), the standard procedure is to inter-

polate the unknowns as

$$u_{h,i} = \sum_{a=1}^{n_{\rm pts}} N^a U_i^a, \qquad i = 1, ..., n_{\rm sd}$$
$$p_h = \sum_{a=1}^{n_{\rm pts}} N^a P^a,$$
$$\vartheta_h = \sum_{a=1}^{n_{\rm pts}} N^a \Theta^a,$$

where upper characters U, P and Θ are used to denote the nodal values of the corresponding lower case variables (at the time step and iteration of interest). The test functions are then taken as $v_{h,i} = N^b \delta_{ik}$ for $k = 1, ..., n_{\rm sd}$ (δ_{ik} being the Kronecker delta), $q_h = N^b$ and $\psi_h = N^b$, $b = 1, ..., n_{\rm pts}$. After the boundary conditions of Dirichlet type are prescribed, this leads to an algebraic system of equation the solution of which yields the nodal unknowns. The matrix of this algebraic system changes from time step to time step and from iteration to iteration due to three reasons: the convective term, the viscous term and the stabilization parameters given in (36)-(38). All these terms depend on the velocity, and hence on the iteration and time step. The former dependence could be avoided by treating explicitly in time these terms, but this would be at the expense of loosing stability of the time integration.

The time consuming task in the calculation of the matrix of the algebraic system (traditionally referred to as 'stiffness matrix') is the numerical integration involved. However, *it is possible to introduce some approximations that allow to express all the integrals in terms of*

$$\begin{aligned} &\int_\Omega N^a N^b \, {\rm d}\Omega, \\ &\int_\Omega N^a \partial_i N^b \, {\rm d}\Omega, \qquad \int_\Omega \partial_i N^a N^b \, {\rm d}\Omega, \qquad i = 1, ..., n_{\rm sd}, \\ &\int_\Omega \partial_i N^a \partial_j N^b \, {\rm d}\Omega, \qquad i, j = 1, ..., n_{\rm sd}, \end{aligned} \tag{39}$$

for $a, b = 1, ..., n_{\rm pts}$, and, if the second derivatives of the shape functions within an element are not zero (or negligible), also in terms of

$$\begin{aligned} &\sum_{e=1}^{n_{\rm el}} \int_{\Omega^e} N^a \Delta N^b \, {\rm d}\Omega, \qquad \sum_{e=1}^{n_{\rm el}} \int_{\Omega^e} \Delta N^a N^b \, {\rm d}\Omega, \\ &\sum_{e=1}^{n_{\rm el}} \int_{\Omega^e} \Delta N^a \partial_i N^b \, {\rm d}\Omega, \qquad \sum_{e=1}^{n_{\rm el}} \int_{\Omega^e} \partial_i N^a \Delta N^b \, {\rm d}\Omega, \qquad i = 1, ..., n_{\rm sd} \\ &\sum_{e=1}^{n_{\rm el}} \int_{\Omega^e} \Delta N^a \Delta N^b \, {\rm d}\Omega, \end{aligned} \tag{40}$$

for $a, b = 1, ..., n_{\rm pts}$. In the previous expressions, ∂_i denotes the partial derivative with respect to the i-th Cartesian coordinate and Δ is the Laplacian operator. For fixed domains Ω, *all the integrals in (39) and (40) can be computed at the beginning of the run and stored.*

At this point there are two questions to be treated. The first is which are the approximations needed to be really able to use only (39) and (40) to build up the matrix of the algebraic system. This is the subject of Sections 4.2 and 4.3.

The second question is how to store the integrals in (39) and (40). The efficiency of the overall implementation depends on how efficient the storage scheme is. The method we employ is described in Section 4.4.

4.2 Approximation of the viscous and convective terms

As it has been mentioned before, the convective and the viscous terms (when the viscosity depends on the velocity) need to be recomputed at each iteration of each time step. However, it is possible to approximate these two terms so that they can be computed with the integrals appearing in (39).

Convective term

Let us begin with the convective term in (24) and assuming for simplicity a Picard linearization, that is, $\lambda_1 = 0$. The terms appearing when $\lambda_1 = 1$, as well as the convective term in the discrete heat equation (26), can be dealt with in a similar way.

Let $\boldsymbol{a}_h \equiv \boldsymbol{u}_h^{n+\alpha,i}$ and $\boldsymbol{u}_h \equiv \boldsymbol{u}_h^{n+\alpha,i+1}$. When the velocity test function is taken such that $v_{h,i} = N^b \delta_{ik}$, with k fixed ($k = 1, ..., n_{\mathrm{sd}}$), the convective term is

$$\int_\Omega \boldsymbol{v}_h \cdot [(\boldsymbol{a}_h \cdot \nabla)\boldsymbol{u}_h] \, \mathrm{d}\Omega = \sum_{j=1}^{n_{\mathrm{sd}}} \left(\int_\Omega N^b a_{h,j} \partial_j N^a \, \mathrm{d}\Omega \right) U_k^a. \tag{41}$$

The need for an additional approximation arises because of the function $a_{h,j}$ appearing within the integrals. Calling A_j^a the nodal values of this function, the approximation that is proposed here is

$$\sum_{j=1}^{n_{\mathrm{sd}}} \left(\int_\Omega N^b a_{h,j} \partial_j N^a \, \mathrm{d}\Omega \right) \approx A_j^c \sum_{j=1}^{n_{\mathrm{sd}}} \left(\int_\Omega N^b \partial_j N^a \, \mathrm{d}\Omega \right), \tag{42}$$

where $c = b$ or $c = a$. In any case, the convective term will be expressed in terms of the integrals of (39), as desired. The reasons for one choice or the other are discussed next:

Choice $c = b$ in (42). Suppose that there exists a nodal integration rule of order n_{int}. The integration points are then the nodes of the mesh, of coordinates $\boldsymbol{x}^g$, $g = 1, ..., n_{\mathrm{pts}}$. The associated weights for the e-th element are denoted W_e^g, and the number of nodes per element by n_{nod} (the same superscript g is used for the numbering of the element nodes). Interpolating the velocity components in the left-hand-side of (42) and using the fact that $N^a(\boldsymbol{x}^b) = \delta^{ab}$, we have that

$$\sum_{j=1}^{n_{\mathrm{sd}}} \left(\int_\Omega N^b a_{h,j} \partial_j N^a \, \mathrm{d}\Omega \right) = \sum_{j=1}^{n_{\mathrm{sd}}} \sum_{c=1}^{n_{\mathrm{pts}}} A_j^c \left(\int_\Omega N^b N^c \partial_j N^a \, \mathrm{d}\Omega \right)$$

$$
\begin{aligned}
&= \sum_{j=1}^{n_{\mathrm{sd}}} \sum_{c=1}^{n_{\mathrm{pts}}} A_j^c \left(\sum_{e=1}^{n_{\mathrm{el}}} \sum_{g=1}^{n_{\mathrm{nod}}} W_e^g \delta^{bg} \delta^{cg} \partial_j N^a(\boldsymbol{x}^g)\Big|_{\Omega^e} \right) + O(h^{n_{\mathrm{int}}}) \\
&= \sum_{j=1}^{n_{\mathrm{sd}}} A_j^b \left(\sum_{e=1}^{n_{\mathrm{el}}} W_e^b \, \partial_j N^a(\boldsymbol{x}^b)\Big|_{\Omega^e} \right) + O(h^{n_{\mathrm{int}}}) \\
&= \sum_{j=1}^{n_{\mathrm{sd}}} A_j^b \left(\int_\Omega N^b \partial_j N^a \, \mathrm{d}\Omega \right) + O(h^{n_{\mathrm{int}}}),
\end{aligned} \tag{43}
$$

which justifies the use of (42) with $c = b$. It is seen that the error is $O(h^{n_{\mathrm{int}}})$. It is known that the nodal numerical integration is of order $n_{\mathrm{int}} = p + 1$, where p is the order of the standard Lagrange interpolation. Therefore, the error of approximation (42) is the same as the order of the finite element interpolation when $c = b$. However, the argument used to arrive to this conclusion is not valid for the stabilization term, whereas the following can also be applied in this case, as will be shown later.

Choice $c = a$ in (42). Since $\boldsymbol{a}_h$ is approximately divergence free, we can approximate

$$
\int_\Omega \boldsymbol{v}_h \cdot [(\boldsymbol{a}_h \cdot \nabla)\boldsymbol{u}_h] \, \mathrm{d}\Omega \approx \int_\Omega \boldsymbol{v}_h \cdot [\nabla \cdot (\boldsymbol{a}_h \otimes \boldsymbol{u}_h)] \, \mathrm{d}\Omega. \tag{44}
$$

In fact, it is not necessary to consider this as an approximation, since the convective term of the original continuous equations could have been written directly as $\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})$ rather than $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$. What is definitely an approximation is *to interpolate the product $\boldsymbol{a}_h \otimes \boldsymbol{u}_h$ instead of each of the components separately.* Doing this when $v_{h,i} = N^b \delta_{ik}$ yields:

$$
\begin{aligned}
\sum_{i,j=1}^{n_{\mathrm{sd}}} \int_\Omega v_{h,i} \partial_j (a_{h,j} u_{h,i}) \, \mathrm{d}\Omega &\approx \sum_{j=1}^{n_{\mathrm{sd}}} \int_\Omega N^b \partial_j \left(\sum_{a=1}^{n_{\mathrm{pts}}} N^a A_j^a U_k^a \right) \mathrm{d}\Omega \\
&= \sum_{j=1}^{n_{\mathrm{sd}}} \sum_{a=1}^{n_{\mathrm{pts}}} A_j^a \left(\int_\Omega N^b \partial_j N^a \, \mathrm{d}\Omega \right) U_k^a,
\end{aligned} \tag{45}
$$

which justifies (42) for $c = a$. The performance of this approximation, as well of the following, will be checked through numerical experiments. However, it can be anticipated that a certain loss of accuracy can occur, since piecewise polynomial solutions of order p will not be anymore solution of the discrete problem when elements of order p are used, since for such solutions $\boldsymbol{u} \otimes \boldsymbol{u}$ is a polynomial of order $2p$.

Viscous term

Let us consider now the approximation of the viscous term in (24). When $v_{h,i} = N^b \delta_{ik}$, with k fixed, this term is

$$
\begin{aligned}
2 \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{v}_h) : \nu \boldsymbol{\varepsilon}(\boldsymbol{u}_h) \, \mathrm{d}\Omega &= \sum_{i,j=1}^{n_{\mathrm{sd}}} \int_\Omega \nu \partial_i v_{h,j} \left(\partial_i u_{h,j} + \partial_j u_{h,i} \right) \mathrm{d}\Omega \\
&= \sum_{i=1}^{n_{\mathrm{sd}}} \int_{\Omega_b} \nu \partial_i N^b \left(\partial_i u_{h,k} + \partial_k u_{h,i} \right) \mathrm{d}\Omega,
\end{aligned} \tag{46}
$$

where Ω_b is the interior of the support of N^b, that is, the union of the domains of the elements to which node b belongs. To see how these integrals can be approximated, let f and g be two given functions, both bounded and the former with bounded derivatives. Expanding f in Taylor series, the integral of their product in a domain Ω_0 is

$$\int_{\Omega_0} fg\,\mathrm{d}\Omega = f(\bar{\boldsymbol{x}}) \int_{\Omega_0} g\,\mathrm{d}\Omega + \sum_{i=1}^{n_{\mathrm{sd}}} \partial_i f|_{\bar{\boldsymbol{x}}} \int_{\Omega_0} (x_i - \bar{x}_i)\, g\,\mathrm{d}\Omega$$
$$+ \frac{1}{2} \sum_{i,j=1}^{n_{\mathrm{sd}}} \partial_i \partial_j f|_{\bar{\boldsymbol{x}}} \int_{\Omega_0} (x_i - \bar{x}_i)\,(x_j - \bar{x}_j)\, g\,\mathrm{d}\Omega + \ldots$$

If d is the diameter of Ω_0 and $\mathrm{meas}(\Omega_0)$ its measure, we see that

$$\int_{\Omega_0} fg\,\mathrm{d}\Omega = f(\bar{\boldsymbol{x}}) \int_{\Omega_0} g\,\mathrm{d}\Omega + O\left(d^m \mathrm{meas}(\Omega_0)\right), \tag{47}$$

where $m = 2$ if $\bar{\boldsymbol{x}}$ is the center of mass of Ω_0 with 'density' g and $m = 1$ otherwise. The idea is now to apply this to the integrals in (46), with $f \equiv \nu$ and $g \equiv \partial_i N^b \left(\partial_i u_{h,k} + \partial_k u_{h,i}\right)$. Obviously, it would be desirable to know the center of mass of Ω_b with density this function g, which is unknown. *What is proposed here is to take ν in (46) as constant and equal to its value ν^b at node b.* Doing this and interpolating the velocity yields:

$$\sum_{i=1}^{n_{\mathrm{sd}}} \int_{\Omega_b} \nu\, \partial_i N^b \left(\partial_i u_{h,k} + \partial_k u_{h,i}\right) \mathrm{d}\Omega \approx \sum_{i=1}^{n_{\mathrm{sd}}} \nu^b \int_{\Omega_b} \partial_i N^b \left(\partial_i u_{h,k} + \partial_k u_{h,i}\right) \mathrm{d}\Omega$$
$$= \sum_{i=1}^{n_{\mathrm{sd}}} \sum_{a=1}^{n_{\mathrm{pts}}} \left[\nu^b \left(\int_\Omega \partial_i N^b \partial_i N^a \,\mathrm{d}\Omega \right) U_k^a + \nu^b \left(\int_\Omega \partial_i N^b \partial_k N^a \,\mathrm{d}\Omega \right) U_i^a \right]. \tag{48}$$

Again, this expression involves only the integrals in (39).

The way to evaluate ν^b is not absolutely clear. For constitutive laws such as (5), the viscosity depends on the velocity gradients, which are discontinuous for standard C^0 finite element interpolations. In order to obtain nodal values of these velocity gradients, we use a standard least-square smoothing from the element-wise values. If G_{ij}^a, $a = 1, \ldots, n_{\mathrm{pts}}$, are the nodal values of $\partial_i u_{h,j}$, $i, j = 1, \ldots, n_{\mathrm{sd}}$, these are the solution of the linear system

$$\sum_{a=1}^{n_{\mathrm{pts}}} \left(\int_\Omega N^b N^a \,\mathrm{d}\Omega \right) G_{ij}^a = \sum_{a=1}^{n_{\mathrm{pts}}} \left(\int_\Omega N^b \partial_i N^a \,\mathrm{d}\Omega \right) U_j^a, \qquad b = 1, \ldots, n_{\mathrm{pts}}, \tag{49}$$

which only involves the integrals in (39). Moreover, to avoid the need for solving (49), we use a nodal numerical integration rule to compute the integral of the left-hand-side, yielding a diagonal 'mass' matrix. Also, since for quadratic elements there are weights which are zero, they are replaced by those corresponding to the splitting of the quadratic elements into a number of linear elements.

REMARK 6. Since

$$\int_\Omega \partial_i N^b \partial_i N^a \,\mathrm{d}\Omega = \int_{\Omega_b \cap \Omega_a} \partial_i N^b \partial_i N^a \,\mathrm{d}\Omega, \tag{50}$$

it is tempting to evaluate the viscosity at a point in $\Omega_b \cap \Omega_a$ not necessarily node b. However, if ν depends also on node a, the second equality in (48) does not hold and the viscous term would be wrongly approximated, since for 'exact' nodal values U_k^a the discrete variational equation (24) would not be satisfied. This can be understood as a lack of 'consistency' of the numerical formulation. In practice, what we have observed is a completely oscillating behavior of the iterative scheme, leading to converged solutions only for mild nonlinearities.

REMARK 7. It is difficult to assess the quality of the approximation in (48), especially because we are interested in cases with a very complex dependence of the viscosity with the velocity (either through a constitutive equation as (5) or through the use of a turbulence model). In general, only an error of order h ($m = 1$ in (47)) can be expected. However, the order of approximation of the finite element method is also lower for non Newtonian fluids as those governed by (5) than when ν is constant (see for example [16]). The need for the least-square smoothing (49) makes things even more complex. Nevertheless, numerical experiments, one of which is presented in Section 5, show that the resulting numerical method yields results very similar to those of the conventional element-based implementation. □

4.3 Nodal stabilization parameters: Consistency and conservation

The last point that needs to be analyzed is the way in which the stabilization terms appearing in (24)-(26) can be approximated to achieve the goal of using only the integrals in (39) and (40) in the implementation. For the purposes of this section, it suffices to consider the stationary problem with $\boldsymbol{\omega} = \mathbf{0}$ and $\sigma = 0$. Also, $\lambda_1 = \lambda_2 = 0$ are taken. Likewise, attention shall be focused on the Navier-Stokes equations, although the same ideas can be applied to the heat equation.

As before, let $\boldsymbol{a}_h$ be the velocity of the previous iteration and $\boldsymbol{u}_h$ the velocity field that needs to be computed. The first approximation to be considered refers to the viscous term of the element residual in (24). Using the fact that the exact velocity is divergence free, the approximation

$$2\nabla \cdot [\nu \boldsymbol{\varepsilon}(\boldsymbol{u}_h)]|_{\Omega^e} \approx \nu \Delta \boldsymbol{u}_h|_{\Omega^e} + 2\, \nabla \nu \cdot \boldsymbol{\varepsilon}(\boldsymbol{u}_h)|_{\Omega^e} \tag{51}$$

avoids the need for computing and storing all the second derivatives of the shape functions, and only their Laplacian needs to be dealt with. Observe that this does not affect the natural boundary conditions associated to the discrete weak problem, since it is used only for the element-wise evaluation of the viscous term. To simplify the calculations, the second term in (51) is evaluated at the previous iteration, computing nodal values for it by using a least-squares smoothing for the viscosity gradients and the velocity gradients as in (49). These nodal values are then added to those of the external force $\boldsymbol{f}$, which is considered to account for them in the following.

Using approximation (51) and considering the simplified situation described above, the discrete problem to be solved is

$$\int_\Omega \boldsymbol{v}_h \cdot [(\boldsymbol{a}_h \cdot \nabla)\boldsymbol{u}_h]\,\mathrm{d}\Omega + 2\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{v}_h) : \nu\boldsymbol{\varepsilon}(\boldsymbol{u}_h)\,\mathrm{d}\Omega - \int_\Omega p_h \nabla\cdot\boldsymbol{v}_h\,\mathrm{d}\Omega$$
$$+S_{\mathrm{mom},1}(\boldsymbol{v}_h;\boldsymbol{u}_h,p_h) + S_{\mathrm{mom},2}(\boldsymbol{v}_h;\boldsymbol{u}_h) = R_{\mathrm{mom}}(\boldsymbol{v}_h), \tag{52}$$

$$\epsilon\int_\Omega p_h q_h\,\mathrm{d}\Omega + \int_\Omega q_h \nabla\cdot\boldsymbol{u}_h + S_{\mathrm{cont}}(q_h;\boldsymbol{u}_h,p_h) = R_{\mathrm{cont}}(q_h), \tag{53}$$

where the right-hand-side terms $R_{\mathrm{mom}}(\boldsymbol{v}_h)$ and $R_{\mathrm{cont}}(q_h)$ come from the force vector (4) and the surface traction in (7),

$$R_{\mathrm{mom}}(\boldsymbol{v}_h) := \int_\Omega \boldsymbol{v}_h\cdot\boldsymbol{f}\,\mathrm{d}\Omega + \int_{\Gamma_{\mathrm{nv}}} \boldsymbol{v}_h\cdot\bar{\boldsymbol{t}}\,\mathrm{d}\Gamma + \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\tau_u\left[(\boldsymbol{a}_h\cdot\nabla)\boldsymbol{v}_h + \nu\Delta\boldsymbol{v}_h\right]\cdot\boldsymbol{f}\,\mathrm{d}\Omega,$$

$$R_{\mathrm{cont}}(q_h) := \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\tau_u\nabla q_h\cdot\boldsymbol{f}\,\mathrm{d}\Omega,$$

and the stabilization terms are given by

$$S_{\mathrm{mom},1}(\boldsymbol{v}_h;\boldsymbol{u}_h,p_h) = \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\tau_u\left[(\boldsymbol{a}_h\cdot\nabla)\boldsymbol{v}_h + \nu\Delta\boldsymbol{v}_h\right]$$
$$\cdot\left[(\boldsymbol{a}_h\cdot\nabla)\boldsymbol{u}_h - \nu\Delta\boldsymbol{u}_h + \nabla p_h\right]\,\mathrm{d}\Omega, \tag{54}$$

$$S_{\mathrm{mom},2}(\boldsymbol{v}_h;\boldsymbol{u}_h) = \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\tau_p(\nabla\cdot\boldsymbol{v}_h)(\nabla\cdot\boldsymbol{u}_h)\,\mathrm{d}\Omega, \tag{55}$$

$$S_{\mathrm{cont}}(q_h;\boldsymbol{u}_h,p_h) = \sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\tau_u\nabla q_h\cdot\left[(\boldsymbol{a}_h\cdot\nabla)\boldsymbol{u}_h - \nu\Delta\boldsymbol{u}_h + \nabla p_h\right]\,\mathrm{d}\Omega, \tag{56}$$

where the stabilization parameters τ_u and τ_p are given by (36) and (37), respectively.

Suppose for a moment that $\Gamma_{\mathrm{nv}} = \partial\Omega$, that is, all the boundary conditions are of Neumann type (case in which the solution would not be unique) and that we can take the test function $\boldsymbol{v}_h$ constant. Assuming that $\boldsymbol{a}_h$ is divergence free (or using expression (44) for the convective term) equations (52)-(53) imply in this case

$$\int_{\partial\Omega}(\boldsymbol{n}\cdot\boldsymbol{a}_h)\,\boldsymbol{u}_h\,\mathrm{d}\Omega = \int_\Omega \boldsymbol{f}\,\mathrm{d}\Omega + \int_{\partial\Omega}\bar{\boldsymbol{t}}\,\mathrm{d}\Gamma, \tag{57}$$

$$\epsilon\int_\Omega p_h\,\mathrm{d}\Omega + \int_{\partial\Omega}\boldsymbol{n}\cdot\boldsymbol{u}_h\,\mathrm{d}\Gamma = 0, \tag{58}$$

which can be understood as global conservation statements for the momentum and the mass of the fluid contained in the domain Ω. It is in this sense that the finite element method can be considered as 'conservative'.

However, equations (52)-(53) are not enforced for constant test functions $\boldsymbol{v}_h$ and q_h, but only for test functions of the form $\boldsymbol{v}_h = N^b\boldsymbol{e}_k$ and $q_h = N^b$, $b = 1,...,n_{\mathrm{pts}}$, $k = 1,...,n_{\mathrm{sd}}$, where $\boldsymbol{e}_k$ is the unit vector along the x_k coordinate. Since the addition of all the shape

functions N^b is 1, equations (57)-(58) can also be obtained by adding up for $b = 1$ to $b = n_{\text{pts}}$ equation (52) enforced for $\boldsymbol{v}_h = N^b \boldsymbol{e}_k$ and also (53) enforced for $q_h = N^b$, but provided that

$$\sum_{b=1}^{n_{\text{pts}}} S_{\text{mom},1}(N^b \boldsymbol{e}_k; \boldsymbol{u}_h, p_h) = 0, \tag{59}$$

$$\sum_{b=1}^{n_{\text{pts}}} S_{\text{mom},2}(N^b \boldsymbol{e}_k; \boldsymbol{u}_h) = 0, \tag{60}$$

$$\sum_{b=1}^{n_{\text{pts}}} S_{\text{cont}}(N^b; \boldsymbol{u}_h, p_h) = 0. \tag{61}$$

These equations can be considered as the *sufficient conditions for the stabilized finite element method to be conservative* (see [14] for further discussion).

In a standard implementation of the formulation, the stabilization parameters are computed depending only on the element, and not on the test functions being used in (54)-(56). However, the idea of the nodal-based implementation presented here is to use nodal values for all the parameters of the formulation, and in particular for τ_u and τ_p. It can be readily observed from (54)-(56) for $\boldsymbol{v}_h = N^b \boldsymbol{e}_k$ that conditions (59)-(61) *will not hold if* τ_u *and* τ_p *depend on* b. If this happens, it is impossible to assess that the numerical formulation is conservative.

Of special interest is what happens to the continuity equation (53) when a penalty parameter is used. Taking $q_h = N^b$ and adding up for all b yields

$$\sum_{b=1}^{n_{\text{pts}}} \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_u \nabla N^b \cdot [(\boldsymbol{a}_h \cdot \nabla)\boldsymbol{u}_h - \nu \Delta \boldsymbol{u}_h + \nabla p_h] \, \mathrm{d}\Omega + \epsilon \int_{\Omega} p_h \, \mathrm{d}\Omega + \int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{u}_h \, \mathrm{d}\Gamma = 0.$$

If τ_u is independent of b, the first term is zero and thus the mean pressure value is zero when so is the mass flux in the domain Ω. This is an interesting property of penalty methods that does not hold when τ_u changes according to different values of b, since the first term is not necessarily zero in this case. If the mass flux is zero (for instance because of a Dirichlet prescription for the velocity), the outcome is that the smaller the value of ϵ is, the larger the mean pressure value. This behavior has been observed in numerical experiments.

Despite this lack of 'conservation', which has to be acknowledged, the stabilization parameters τ_u and τ_p will be evaluated at node b when $\boldsymbol{v}_h = N^b \boldsymbol{e}_k$ and when $q_h = N^b$. The reason for this is related to the consistency of the scheme, which is discussed next.

Let us consider the stabilization term (54). Taking $\boldsymbol{v}_h = N^b \boldsymbol{e}_k$ and interpolating the velocity and the pressure it is found that

$$S_{\text{mom},1}(N^b \boldsymbol{e}_k; \boldsymbol{u}_h, p_h) = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_u \left[\sum_{i=1}^{n_{\text{sd}}} \left(a_{h,i} \partial_i N^b \right) + \nu \Delta N^b \right]$$
$$\cdot \left[\sum_{a=1}^{n_{\text{pts}}} \sum_{j=1}^{n_{\text{sd}}} \left(a_{h,j} \partial_j N^a \right) U_k^a - \nu \sum_{a=1}^{n_{\text{pts}}} \Delta N^a U_k^a + \sum_{a=1}^{n_{\text{pts}}} \partial_k N^a P^a \right] \mathrm{d}\Omega$$

$$
\begin{aligned}
&= \sum_{a=1}^{n_{\text{pts}}} \left[\sum_{i,j=1}^{n_{\text{sd}}} \int_{\Omega} \tau_u \left(a_{h,i} \partial_i N^b \right) \left(a_{h,j} \partial_j N^a \right) \, \mathrm{d}\Omega \right] U_k^a \\
&- \sum_{a=1}^{n_{\text{pts}}} \left[\sum_{i=1}^{n_{\text{sd}}} \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_u \nu \left(a_{h,i} \partial_i N^b \right) \Delta N^a \, \mathrm{d}\Omega \right] U_k^a \\
&+ \sum_{a=1}^{n_{\text{pts}}} \left[\sum_{i=1}^{n_{\text{sd}}} \int_{\Omega} \tau_u \left(a_{h,i} \partial_i N^b \right) \partial_k N^a \, \mathrm{d}\Omega \right] P^a \\
&+ \sum_{a=1}^{n_{\text{pts}}} \left[\sum_{j=1}^{n_{\text{sd}}} \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_u \nu \Delta N^b \left(a_{h,j} \partial_j N^a \right) \, \mathrm{d}\Omega \right] U_k^a \\
&- \sum_{a=1}^{n_{\text{pts}}} \left[\sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_u \nu^2 \Delta N^b \Delta N^a \, \mathrm{d}\Omega \right] U_k^a + \sum_{a=1}^{n_{\text{pts}}} \left[\sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_u \nu \Delta N^b \partial_k N^a \, \mathrm{d}\Omega \right] P^a. \qquad (62)
\end{aligned}
$$

The goal now is to make the convenient approximations to write this in terms of the integrals in (39)-(40). First of all, observe that the integrals involving the shape functions of nodes a and b can be extended over $\Omega_a \cap \Omega_b$ only, the intersection of the interior of the supports of N^a and N^b. Therefore, if approximation (47) is to be used, the functions taken out of the integrals have to be evaluated either at node a or at node b, or a combination of both. However, it is not a matter of choice. The only possibility to approximate (62) is to take:

- The velocity $\boldsymbol{a}_h$ appearing in the element residual of the differential equation evaluated at node a, and thus the i-th component equal to the nodal value A_i^a. This corresponds to the second approximation of the convective term discussed in the previous section, that is, (42) with $c = a$. The reasons for this choice are still valid in this case, whereas the argument for taking $c = b$ is not valid any more, since now, for example, $\tau_u \left(a_{h,i} \partial_i N^b \right) (\boldsymbol{x}^a) \neq \delta^{ba}$ (see the derivation of (43)).

- The velocity $\boldsymbol{a}_h$ appearing in the operator applied to the test function evaluated at node b, and thus the i-th component equal to the nodal value A_i^b. This is needed for consistency reasons, similar to those given in Remark 6: if $\boldsymbol{a}_h$ depends also on node a, exact nodal values of the velocity would not satisfy the discrete variational equations. However, the same discussion concerning the choice of the stabilization parameters is applicable now: the resulting numerical scheme *will not* be conservative.

- The viscosity evaluated at node b. This is also due to the consistency requirement discussed in the previous section.

- The parameters τ_u and τ_p evaluated at node b. Even though this produces a non-conservative scheme, it is essential to have a consistent numerical method, in the sense that exact solutions of the continuous problem should also be solutions of the discrete one, provided they belong to the finite element space. The expressions we use for these

parameters are (36) and (37) (and obviously (38) when the heat equation is dealt with), taking $\boldsymbol{u}_h$ as the nodal velocity at node b, ν the viscosity at this node and h as the minimum distance from node b to the surrounding nodes.

Using all these approximations in (62) one finds

$$
\begin{aligned}
S_{\mathrm{mom},1}(N^b\boldsymbol{e}_k;\boldsymbol{u}_h,p_h) \approx & \sum_{a=1}^{n_{\mathrm{pts}}}\left[\sum_{i,j=1}^{n_{\mathrm{sd}}} A_i^b A_j^a \tau_u^b \left(\int_\Omega \partial_i N^b \partial_j N^a \,\mathrm{d}\Omega\right)\right] U_k^a \\
& -\sum_{a=1}^{n_{\mathrm{pts}}}\left[\sum_{i=1}^{n_{\mathrm{sd}}} A_i^b \tau_u^b \nu^b \left(\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e} \partial_i N^b \Delta N^a \,\mathrm{d}\Omega\right)\right] U_k^a \\
& +\sum_{a=1}^{n_{\mathrm{pts}}}\left[\sum_{i=1}^{n_{\mathrm{sd}}} A_i^b \tau_u^b \left(\int_\Omega \partial_i N^b \partial_k N^a \,\mathrm{d}\Omega\right)\right] P^a \\
& +\sum_{a=1}^{n_{\mathrm{pts}}}\left[\sum_{j=1}^{n_{\mathrm{sd}}} A_j^a \nu^b \tau_u^b \left(\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e} \Delta N^b \partial_j N^a \,\mathrm{d}\Omega\right)\right] U_k^a \\
& -\sum_{a=1}^{n_{\mathrm{pts}}}\left[(\nu^b)^2 \tau_u^b \left(\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e} \Delta N^b \Delta N^a \,\mathrm{d}\Omega\right)\right] U_k^a \\
& +\sum_{a=1}^{n_{\mathrm{pts}}}\left[\nu^b \tau_u^b \left(\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e} \Delta N^b \partial_k N^a \,\mathrm{d}\Omega\right)\right] P^a .
\end{aligned}
$$

Once again, the objective of using only the integrals in (39)-(40) has been accomplished.

REMARK 8. It is interesting to note that even when $\boldsymbol{a}_h = \boldsymbol{0}$, the final algebraic system of equations *will not be symmetric* if the stabilization parameters change from node to node, that is, if the mesh is not uniform or the viscosity is variable. □

The way to treat the stabilization terms (55) and (56), and also the one coming from the stabilization of the heat equation when this is solved, is exactly the same as for (54): the stabilization parameters must be evaluated at the node associated to the test function being used and the convective velocity and viscosity evaluated as explained above.

4.4 Mesh graph and basic algorithm

Once it has been established that the matrix of the algebraic system can be built up making use of the integrals in (39) and (40), the question that needs to be addressed is how to store these integrals in an efficient way. The technique adopted in this work is to use a *compressed sparse row* (CSR) format to store the $n_{\mathrm{pts}} \times n_{\mathrm{pts}}$ 'virtual' matrix $\boldsymbol{M}_{\mathrm{mesh}}$, whose coefficients are $M_{\mathrm{mesh}}^{ab} = 1$ if nodes a and b are connected, $= 0$ otherwise. This is the matrix of the graph associated to the finite element mesh.

In order to store $\boldsymbol{M}_{\mathrm{mesh}}$ using the CSR format, two arrays are needed. Let *NZD* be the number of nonzero coefficients in $\boldsymbol{M}_{\mathrm{mesh}}$. These two arrays are:

$$
R_{\mathrm{mesh}}(n_{\mathrm{pts}}); \quad R_{\mathrm{mesh}}(a) = \text{Coefficient of } \boldsymbol{M}_{\mathrm{mesh}} \text{ where row number } a \text{ starts}, \tag{63}
$$

$$C_{\text{mesh}}(NZD);\quad C_{\text{mesh}}(I) = \text{Column in } \boldsymbol{M}_{\text{mesh}} \text{ of the coefficient } I. \tag{64}$$

For implementation convenience, it is useful to take R_{mesh} of dimension $n_{\text{pts}} + 1$, with $R_{\text{mesh}}(n_{\text{pts}} + 1) = NZD + 1$ (see the algorithm of Box 1).

The arrays in (63)-(64) allow to access to all the components of the integrals in (39) and (40) when these are stored in $n_{\text{pts}} \times n_{\text{pts}}$ matrices (each component of which can be a n_{sd} vector or a $n_{\text{sd}} \times n_{\text{sd}}$ matrix). Moreover, $R_{\text{mesh}}(n_{\text{pts}})$ and $C_{\text{mesh}}(NZD)$ can also be used to store in CSR format the stiffness matrix of the problem, and thus they define completely its topology. Once they have been computed, the memory for the classical array of nodal connections, which has the list of nodes that each element has, can be freed.

To compute the contributions to the force vector due to the surface traction $\bar{\boldsymbol{t}}$ in (7) or the heat flux $\bar{h}$ in (9), the integrals of the shape functions products over the boundaries are needed. However, these can be obtained from the corresponding integrals over the interior of the domain using the expression

$$\int_{\partial\Omega} n_i N^a N^b \, \mathrm{d}\Gamma = \int_{\Omega} \partial_i N^a N^b \, \mathrm{d}\Omega + \int_{\Omega} N^a \partial_i N^b \, \mathrm{d}\Omega, \qquad i = 1, ..., n_{\text{sd}}, \tag{65}$$

where n_i is the i-th component of the exterior normal to $\partial\Omega$. The left-hand-side of (65) is equal to $n_i(\bar{\boldsymbol{x}})$ multiplied by the integral of $N^a N^b$, where $\bar{\boldsymbol{x}}$ is a point in $\partial\Omega \cap \bar{\Omega}_a \cap \bar{\Omega}_b$. Assuming that this point is the same for all i it is found that

$$\int_{\partial\Omega} N^a N^b \, \mathrm{d}\Gamma \approx \pm \left[\sum_{i=1}^{n_{\text{sd}}} \left(\int_{\partial\Omega} n_i N^a N^b \, \mathrm{d}\Gamma \right)^2 \right]^{1/2}. \tag{66}$$

The sign of this integral is determined from the sign of the integrals in (65).

To store the boundary integrals in (66) it is a waste of memory to use a matrix with the same sparsivity pattern as $\boldsymbol{M}_{\text{mesh}}$. It if preferable to store also the boundary graph, that is, the matrix $\boldsymbol{M}_{\text{boun}}$, of dimensions $n_{\text{bpt}} \times n_{\text{bpt}}$, where n_{bpt} is the number of boundary nodes. If NZB is the number of nonzero coefficients of $\boldsymbol{M}_{\text{boun}}$, the arrays needed to store it are

$$R_{\text{boun}}(n_{\text{bpt}});\quad R_{\text{boun}}(a) = \text{Coefficient of } \boldsymbol{M}_{\text{boun}} \text{ where row number } a \text{ starts}, \tag{67}$$

$$C_{\text{boun}}(NZB);\quad C_{\text{boun}}(I) = \text{Column in } \boldsymbol{M}_{\text{boun}} \text{ of the coefficient } I. \tag{68}$$

As for R_{mesh}, it is useful to take R_{boun} of dimension $n_{\text{bpt}} + 1$ and $R_{\text{boun}}(n_{\text{bpt}} + 1) = NZB + 1$.

Let us see know how the arrays in (63)-(64) (and (67)-(68)) allow to construct the stiffness matrix and force vector of the algebraic problem using the approximations described in the previous section. This algebraic problem has the form

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{G} \\ \boldsymbol{D} & \boldsymbol{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{bmatrix}$$

where $\boldsymbol{U}$ and $\boldsymbol{P}$ are the arrays of nodal unknowns of velocity and pressure, respectively, $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ are the force vectors arising from the body forces and the surface traction, and

matrices $\boldsymbol{K}$, $\boldsymbol{G}$, $\boldsymbol{D}$ and $\boldsymbol{L}$ have the structure

$$\begin{aligned}
&\boldsymbol{K} = \left[\boldsymbol{K}^{ab}\right], \qquad \boldsymbol{K}^{ab} = \left[K_{ij}^{ab}\right], \\
&\boldsymbol{G} = \left[\boldsymbol{G}^{ab}\right], \qquad \boldsymbol{G}^{ab} = \left[G_{i}^{ab}\right], \\
&\boldsymbol{D} = \left[\boldsymbol{D}^{ab}\right], \qquad \boldsymbol{D}^{ab} = \left[D_{i}^{ab}\right]^{\mathrm{t}}, \\
&\boldsymbol{L} = \left[L^{ab}\right], \\
&\boldsymbol{F}_1 = \left[\boldsymbol{F}_1^a\right], \qquad \boldsymbol{F}_1^a = \left[F_{1,i}^a\right], \qquad \boldsymbol{F}_2 = \left[F_2^a\right].
\end{aligned} \tag{69}$$

Indexes a and b in these expressions run from 1 to n_{pts}, whereas i and j run from 1 to n_{sd}.

Box 1: Construction of the arrays of the algebraic system

FOR $b = 1, n_{\mathrm{pts}}$ DO:

- Compute τ_u^b, τ_p^b
- Set $F_{1,k}^b = 0$ for $k = 1, ..., n_{\mathrm{sd}}$, $F_2^b = 0$

FOR $I = R_{\mathrm{mesh}}(b), R_{\mathrm{mesh}}(b+1) - 1$ DO:

- Identify the column node: $a = C_{\mathrm{mesh}}(I)$
- Compute the contributions to the stiffness matrix K_{kl}^{ba}, G_k^{ba}, D_k^{ba}, L^{ba}, for $k, l = 1, ..., n_{\mathrm{sd}}$ (see Box 2)
- Add contribution to the force vectors $F_{1,k}^b$ for $k = 1, ..., n_{\mathrm{sd}}$, F_2^b (see Box 3)

END

END

The arrays (69) can be computed by looping first over each nodal point and then over the nodes connected to it, as indicated in Box 1. This can be done making use of (63)-(64). This algorithm turns out to be very efficient compared to the standard loop over the elements to compute the element contributions. In particular, all gather-scatter operations are avoided, and there is no need to perform the classical assembly operations.

Of course, Dirichlet boundary conditions have to be incorporated before solving the final linear system of equations.

The components of the matrices in (69) are indicated in Box 2. The situation considered

Box 2: Components of the stiffness matrix

$$
\begin{aligned}
K_{kl}^{ba} &= \sum_{i=1}^{n_{\mathrm{sd}}} A_i^a \left(\int_\Omega N^b \partial_i N^a \, \mathrm{d}\Omega \right) \delta_{kl} + \nu^b \sum_{i=1}^{n_{\mathrm{sd}}} \left(\int_\Omega \partial_i N^b \partial_i N^a \, \mathrm{d}\Omega \right) \delta_{kl} \\
&\quad + \nu^b \left(\int_\Omega \partial_l N^b \partial_k N^a \, \mathrm{d}\Omega \right) + \tau_p^b \left(\int_\Omega \partial_k N^b \partial_l N^a \, \mathrm{d}\Omega \right) \\
&\quad + \tau_u^b \sum_{i,j=1}^{n_{\mathrm{sd}}} A_i^b A_j^a \left(\int_\Omega \partial_i N^b \partial_j N^a \, \mathrm{d}\Omega \right) \delta_{kl} \\
&\quad - \nu^b \tau_u^b \sum_{i=1}^{n_{\mathrm{sd}}} A_i^b \left(\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \partial_i N^b \Delta N^a \, \mathrm{d}\Omega \right) \delta_{kl} \\
&\quad + \nu^b \tau_u^b \sum_{i=1}^{n_{\mathrm{sd}}} A_i^a \left(\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \Delta N^b \partial_i N^a \, \mathrm{d}\Omega \right) \delta_{kl} \\
&\quad - (\nu^b)^2 \tau_u^b \left(\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \Delta N^b \Delta N^a \, \mathrm{d}\Omega \right) \delta_{kl} \\
G_k^{ba} &= - \left(\int_\Omega \partial_k N^b N^a \, \mathrm{d}\Omega \right) + \tau_u^b \sum_{i=1}^{n_{\mathrm{sd}}} A_i^b \left(\int_\Omega \partial_i N^b \partial_k N^a \, \mathrm{d}\Omega \right) \\
&\quad + \nu^b \tau_u^b \left(\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \Delta N^b \partial_k N^a \, \mathrm{d}\Omega \right) \\
D_k^{ba} &= \left(\int_\Omega N^b \partial_k N^a \, \mathrm{d}\Omega \right) + \tau_u^b \sum_{i=1}^{n_{\mathrm{sd}}} A_i^a \left(\int_\Omega \partial_k N^b \partial_i N^a \, \mathrm{d}\Omega \right) \\
&\quad - \nu^b \tau_u^b \left(\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \partial_k N^b \Delta N^a \, \mathrm{d}\Omega \right) \\
L^{ba} &= \epsilon \left(\int_\Omega N^b N^a \, \mathrm{d}\Omega \right) + \tau_u^b \sum_{i=1}^{n_{\mathrm{sd}}} \left(\int_\Omega \partial_i N^b \partial_i N^a \, \mathrm{d}\Omega \right)
\end{aligned}
$$

is the same as in subsection 4.3, that is to say, the stationary problem with $\boldsymbol{\omega} = \mathbf{0}$, $\sigma = 0$ and $\lambda_1 = \lambda_2 = 0$. The extension to the most general case, including thermally coupled flows, is straightforward.

The expressions of Box 2 summarize all the approximations introduced in this paper related to the viscous term, the convective term and the stabilization terms.

Box 3: Contribution to the force vectors

$$F^b_{1,k} \leftarrow F^b_{1,k} + \left(\int_\Omega N^b N^a \,\mathrm{d}\Omega\right) f^a_k$$
$$+\tau^b_u \sum_{i=1}^{n_{\mathrm{sd}}} A^b_i \left(\int_\Omega \partial_i N^b N^a \,\mathrm{d}\Omega\right) f^a_k + \tau^b_u \nu^b \left(\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \Delta N^b N^a \,\mathrm{d}\Omega\right) f^a_k$$
$$F^b_2 \leftarrow F^b_2 + \tau^b_u \sum_{i=1}^{n_{\mathrm{sd}}} \left(\int_\Omega \partial_i N^b N^a \,\mathrm{d}\Omega\right) f^a_i$$

- If a and b are boundary nodes:

$$F^b_{1,k} \leftarrow F^b_{1,k} + \left(\int_{\partial\Omega} N^b N^a \,\mathrm{d}\Gamma\right) \bar{t}^a_k$$

In Box 3, the body force and the surface traction have been assumed to be interpolated from their nodal values as

$$\boldsymbol{f} \approx \sum_{a=1}^{n_{\mathrm{pts}}} N^a \boldsymbol{f}^a, \qquad \bar{\boldsymbol{t}} \approx \sum_{a=1}^{n_{\mathrm{bpt}}} N^a \bar{\boldsymbol{t}}^a. \tag{70}$$

5 Numerical examples

In this section we present the results of some numerical examples obtained with the stabilized finite element formulation proposed in this paper. Two types of conclusions can be drawn from them. On the one hand, these examples serve to check the behavior of the stabilized formulation is situations more general than those in which it can be analyzed [3]. These situations include thermally driven flows, flows of nonlinear materials and transient problems. An example of each situation is presented.

On the other hand, these numerical examples also serve to compare the performance of the standard element-based implementation of the stabilized formulation and the nodal-based one presented in Section 4, which involves several additional approximations.

The last two examples are intended to demonstrate that the work developed in Section 4 'makes sense'. Firstly, because the resulting numerical scheme is shown to be (almost) optimally convergent in a numerical test, despite all the approximations needed to arrive to it, and secondly because this scheme turns out to be very efficient, an attribute of which finite element methods are usually blamed for lacking.

5.1 Thermally coupled flow in a cavity

In this example, the convective motion of a fluid enclosed in the square cavity $[0,1]\times[0,1]$ and driven by a temperature gradient will be numerically analyzed. The left vertical wall $x=0$ is heated and maintained at a constant temperature $\vartheta=1$, while the right vertical wall $x=1$ is kept at $\vartheta=0$. Horizontal walls are assumed to be adiabatic, i.e., boundary condition (9) with $\bar{h}=0$ is prescribed. Homogeneous Dirichlet boundary conditions are prescribed everywhere on the boundary for the velocity.

Let L be a characteristic length of the problem and G_ϑ a characteristic temperature gradient. The Grashof number Gr and the Prandtl number Pr are defined as

$$\mathrm{Gr} := \frac{\beta|\boldsymbol{g}|L^3 G_\vartheta}{\nu^2}, \qquad \mathrm{Pr} := \frac{\nu}{\kappa}.$$

Taking $L=1$ and $G_\vartheta=1$ in our case, the physical parameters have been adjusted to yield a Prandtl number $\mathrm{Pr}=0.005$ and a Grashof number $\mathrm{Gr}=3\times10^6$. For this combination of values, there is a unique and stable stationary solution to the Navier-Stokes equations coupled with the heat equation using the Boussinesq assumption [17]. Thus, the stationary version of problem (1)-(3) is solved in this example. The gravity is assumed to point downwards.

The finite element mesh employed to discretize the problem, which is refined near the boundaries, consists of 2684 bilinear (Q_1) elements and 2809 nodal points. The Navier-Stokes equations have been linearized up to first order, and the standard penalty method, with $\epsilon=10^{-6}$, has been used to fix the pressure mean to zero. The convergence tolerance has been taken as 0.01% in the relative L^2 norm.

The velocity field, the pressure contours and the temperature contours are shown in fig. 1-fig. 3. These results are very similar using the standard element-based implementation and the nodal-based one presented in this paper. The comparison is made in fig. 4-fig. 9, where the x- and y-velocity sections, as well as the temperature section at $x=0.5$ and $y=0.5$ (that is, the mid-sections of the cavity) are shown. It is important to remark that this example involves two of the approximations discussed in Section 4, namely, those related to the convective term for both the Navier-Stokes and the heat transport equations, and the approximations needed to deal with their stabilization terms.

5.2 Flow over a cylinder

This example involves the flow past a cylinder, another widely solved benchmark problem. The computational domain is $\bar{\Omega} = [0, 16] \times [0, 8] \setminus D$, with the cylinder D of diameter 1 and centered at $(4, 4)$. The velocity at $x = 0$ is prescribed to $(1, 0)$, whereas at $y = 0$ and $y = 8$ the y-velocity component is prescribed to 0 and the x-component is left free. The outflow (where both the x- and y-components are free) is $x = 16$. The Reynolds number is 100, based on the cylinder diameter and the prescribed inflow velocity. The finite element mesh employed consists of 4000 linear triangles, with 2100 nodal points, and is refined near the cylinder.

In order to obtain the fully developed vortex shedding characteristic of this problem, 90 time steps have been performed with $\delta t = 1$ and $\alpha = 0.5$ (Crank-Nicholson scheme), employing for that the element-based implementation. The convergence tolerance within each time step has been taken as 1% (a single Picard iteration has been needed to converge). The solution thus obtained shows a fully developed periodic flow pattern. These results have been taken as the initial condition for a more accurate calculation, now computed with $\delta t = 0.1$ and requiring a convergence tolerance of 0.01% in the relative L^2 norm. Two or three Newton-Raphson iterations have been performed for each time step, both for the element-based and the nodal-based implementations of the formulation.

The period of the oscillations has been found to be 5.9 time units with both implementations. The values given in references [18] and [19] are 6.0 and 5.6, respectively. In Reference [20], the period obtained with a very fine mesh (3426 Q_2/P_1 elements, 14000 nodal points) is 5.8 time units. See also [21] for results obtained using a similar stabilized formulation.

The streamline snapshots and the pressure contours shown in fig. 10-fig. 13 correspond to $t = 7$ and $t = 10$, that is, approximately half a period ($t = 0$ corresponds to the periodic solution computed as described earlier with a higher tolerance and a higher time step size).

The important point is the comparison of the results obtained with the element-based and the nodal-based implementations. This is done in fig. 14-fig. 19. It is observed there that there are only slight differences in the spatial amplitude of the oscillations, as well as some differences in the results near the cylinder (in particular, in the average pressure level).

5.3 Extrusion of a nonlinear fluid

In this section we present some numerical results obtained for the well-known 4:1 plane extrusion problem. This is a very popular test for non-Newtonian flows, since all the flow features that characterize these fluids are present in this problem.

The computational domain in this case is $[0, 20] \times [0, 4] \cup [20, 40] \times [3, 4]$. The inflow is the section $x = 0$, where the velocity field is prescribed to $(v_x(y), 0)$, where $v_x(y)$ has a parabolic profile with maximum 1 at $y = 4$ and minimum 0 at $y = 0$. The outflow $y = 40$ is left free,

that is, condition (7) with $\bar{\boldsymbol{t}} = \mathbf{0}$ is prescribed. On $y = 4$ the velocity is set to $(1, 0)$ and on the rest of the boundary the no-slip condition is used.

The finite element mesh employed for the space discretization is composed of 2100 bilinear (Q_1) elements, with a total of 2201 nodal points. There are 15 elements in the y-direction from coordinates $y = 3$ to $y = 4$ and only 12 from $y = 0$ to $y = 3$. The concentration of elements in the former zone is needed if one wants to reproduce accurately the shear thinning effect of fluids whose viscosity obeys the power law that we consider now, given by (5). Since the effective viscosity values that result from this law are very high, the convective term of the Navier-Stokes equations is neglected (the flow is assumed to be governed by creep), as well as the buoyancy forces due to temperature (the flow is assumed to be thermally uncoupled).

The values of the physical constants that have been used are (all in SI units): $\rho_0 = 1200$ (density), $K_0 = 10^6$ (material consistency) and $n = 0.4$ (rate sensitivity). For this value of n the effect of the non-constant viscosity is pronounced. Since the expression of the viscosity (5) tends to infinity when $I_2(\boldsymbol{\varepsilon})$ tends to zero, a cut-off value $\nu_c = 10^{12}$ for ν has been introduced. The values of the viscosity for the converged solutions are always below this limit, except in isolated points.

The streamlines and pressure contours for this problem are shown in fig. 20 and fig. 21, respectively. As before, results are very similar using the element-based and the nodal-based implementations. A detailed comparison is made in fig. 22-fig. 25, where the x- and y-velocity components at $x = 30$ and $y = 3.5$ are shown. In these figures, the results obtained using a mesh of 525 biquadratic (Q_2) elements with the same set of nodes as the mesh of bilinear elements are also shown for the nodal-based implementation. It has to be remarked that the element-based one did not converge using Q_2 elements.

5.4 Convergence test

Let us consider now a 2D steady state test with analytical solution to check the behavior in space of the finite element approximation to problem (1)-(3). We take Ω as the unit square and the force term so that the exact solution is $p = 0$ and $\boldsymbol{u}(x, y) = (f(x)g'(y), -f'(x)g(y))$, with $f(x) = x^2(1-x)^2$ and $g(y) = y^2(1-y)^2$. This velocity field vanishes on $\partial\Omega$.

As physical properties we have taken $\nu = 0.001$ and different values of $\omega = |\boldsymbol{\omega}|$ and σ. In particular, results will be shown for $\sigma = 0,\ 1000$ and $\omega = 0,\ 1000$. We have used three uniform finite element meshes (meshes 1, 2 and 3) of 5×5, 10×10 and 20×20 biquadratic elements, so that the element sizes are $h = 0.2$, $h = 0.1$ and $h = 0.05$, respectively. The resulting values of the element Reynolds number are not very high and for this particular example the standard Galerkin approach using a stable velocity-pressure pair, such as the Taylor-Hood element Q_2/Q_1, works for meshes 2 and 3 in the absence of Coriolis force. However, when this force exists, the Galerkin method yields completely oscillatory results all over the computational domain (see [3]).

In fig. 26 we have plotted the convergence of the velocities obtained with the stabilized method as the mesh is refined in the discrete ℓ^2 norm and for different combinations of the values of σ and ω. This error is defined as

$$E = \left[\sum_{a=1}^{n_{\text{pts}}}\sum_{i=1}^{n_{\text{sd}}}(U_i^a - u_i(\boldsymbol{x}^a))^2\right]^{1/2}\left[\sum_{a=1}^{n_{\text{pts}}}\sum_{i=1}^{n_{\text{sd}}}(u_i(\boldsymbol{x}^a))^2\right]^{-1/2},$$

where $\boldsymbol{x}^a$ are the coordinates of the nodes.

The optimal convergence rate that should be expected is 3. From fig. 26 it is seen that this is what is found for the element-based implementation. However, for the nodal-based one it is slightly smaller when $\sigma = \omega = 0$ (it is approximately 2.6). This is due to the approximation made for the convective term, which is the dominant one in this case. However, it has to be noted that also approximation (70) has been employed to interpolate the force vector, and this might have introduced some additional error.

A comparison of results for $c = a$ and for $c = b$ in (42) (and the corresponding approximation for the convective part of the stabilization term) is shown in fig. 27. It is seen there that the choice $c = a$ (employed also in fig. 26) gives better results than $c = b$. The convergence rate obtained using a mesh of Q_1 elements obtained by splitting each Q_2 element into four is also plotted in fig. 27 (using again $c = a$). In this case, convergence is optimal, with rate 2.

5.5 Efficiency test

In this final example, the efficiency of the nodal-based implementation compared to the element-based one is studied. For that, we consider the 3D extension of the the previous example. The domain is the unit cube $\bar{\Omega} = [0,1] \times [0,1] \times [0,1]$ and is discretized using three meshes of P_1, P_2, Q_1 and Q_2 elements.

The interest of this example is to compare the CPU time needed to construct the stiffness matrix of the final linear system for the two implementations of the stabilized method. Although this linear system has not been solved, the force term has been taken so as to obtain as exact solution $\boldsymbol{u}(x,y,z) = (h(z)f(x)g'(y), -h(z)f'(x)g(y), 0)$, with $f(x)$ and $g(y)$ as in the previous example and $h(z) = z(1-z)$. Likewise, $\boldsymbol{\omega} = (1,1,1)$ and $\sigma = 1000$ have been used.

Results are shown in Table 1, where the meshes employed have been identified as follows: 't' corresponds to tetrahedral elements and 'h' to hexahedral, 'l' to linear and 'q' to quadratic interpolations, and '1', '2' or '3' identify the number of nodes of the mesh. These meshes have respectively 7^3, 13^3 and 25^3 nodal points, equally distributed.

For the case of tetrahedral linear elements (P_1), that is, for meshes tl1, tl2 and tl3, two possibilities have been considered, namely, the use of 4 points to perform the numerical integration (which corresponds to 'full' integration) and 1 point (which introduces an important numerical integration error but is possible for this stationary problem). Obviously, this does not affect the CPU time needed for the construction of the stiffness matrix in the nodal-based

implementation, since integrals are computed and stored at the beginning of the calculations. Similarly, for tetrahedral quadratic elements (P_2), both 11 points and 4 points can be used for the numerical integration.

The results shown in Table 1, obtained using a single SGI R10000 processor, show that the nodal-based implementation is clearly much more efficient than the element-based one. Obviously these results are dependent on the particular coding of the corresponding finite element algorithm, but this can not affect the general tendency observed in Table 1.

Let us consider now the memory storage required for both implementations. If the number of nodal points is increased, the asymptotic number of resulting elements is $n_{\rm el} = f_{\rm ele} n_{\rm pts}$, where $f_{\rm ele}$ is a factor that depends on the element type. Similarly, if $f_{\rm con}$ is the average number of nodes to which a node is connected, the number of nonzero coefficients of the mesh graph is $NZD = f_{\rm con} n_{\rm pts}$. Let $\mathcal{M}_{\rm eb} n_{\rm pts}$ be the number of coefficients that need to be stored to compute the stiffness matrix for the element based implementation and $\mathcal{M}_{\rm nb} n_{\rm pts}$ for the nodal based one. Assuming in both cases that no second derivatives need to be computed, the coefficients $\mathcal{M}_{\rm eb}$ and $\mathcal{M}_{\rm nb}$ are:

$$
\begin{aligned}
\mathcal{M}_{\rm eb} &= [(1 \text{ (element of volume)} + n_{\rm sd} n_{\rm nod} \text{ (Cartesian derivatives)})\, n_{\rm int} \\
&\quad + n_{\rm nod} \text{ (nodal connectivities)}]\, f_{\rm ele} \\
\mathcal{M}_{\rm nb} &= 0.5\,(1 \text{ (first integrals in (39))} + n_{\rm sd} \text{ (second integrals in (39))} \\
&\quad + n_{\rm sd} n_{\rm sd} \text{ (last integrals in (39))})\, f_{\rm con}
\end{aligned}
$$

Observe that in the expression of $\mathcal{M}_{\rm nb}$ we have taken into account the symmetries of the intregrals in (39), as well as the fact that (65), which allows to store only one of the integrals in the second row of (39). Likewise, for the element-based implementation the storage of the nodal connectivities has to be accounted for, whereas for the nodal-based one the mesh graph can be considered as part of the memory needed to allocate the stiffness matrix of the problem.

The coefficients $\mathcal{M}_{\rm eb}$ and $\mathcal{M}_{\rm nb}$ for the different types of meshes in terms of the factors $f_{\rm ele}$ and $f_{\rm con}$ are given in Table 2. It is observed there that the nodal-based implementation is less memory demanding for lower order interpolations and for fully integrated quadratic tetrahedra, whereas the memory required is more than for the element-based implementation for quadratic hexahedra.

6 Conclusions

Two different aspects related to the finite element approximation of the incompressible Navier-Stokes equations have been treated in this paper. The first of them is the numerical formulation, which consists of a stabilized method able to deal with a very wide range of flows. It has been shown that the method gives very good results for thermally driven

flows, non Newtonian fluids and transient problems. Particular features of this stabilized formulation are that it is based on multiplying the element residuals of the equations by the adjoint of the linearized Navier-Stokes operator applied to the test functions, and also the design of the stabilization parameters. Both ingredients allow to stabilize very different types of numerical instabilities, namely, those which are classical and arise in convection dominated flows and equal velocity-pressure interpolations, and also the less studied cases of dominating Coriolis forces and small medium permeabilities.

The good numerical performance of the stabilized finite element formulation is maintained when the nodal-based implementation described here is used. The basic idea of this implementation is to characterize the topology of the finite element mesh by the matrix of its graph stored in compressed sparse row format. The arrays needed for this representation can be used also to store the integrals of the shape functions and shape function derivative products, the corner stones of the implementation. All the arrays appearing in the fully discrete problem can expressed in terms of them after performing the appropriate approximations. The final numerical scheme is shown in Boxes 1, 2 and 3. Numerical experiments show that the good numerical results due to the stabilized formulation can be obtained in an efficient manner by means of this implementation.

References

[1] T.J.R. Hughes. Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. *Computer Methods in Applied Mechanics and Engineering*, 127:387–401, 1995.

[2] R. Codina. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, submitted.

[3] R. Codina. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, submitted.

[4] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Springer Verlag, 1991.

[5] T.J.R. Hughes, L.P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: a stable Petrov-Galerkin formulation for the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.

[6] L.P. Franca and S.L. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99:209–233, 1992.

[7] P. Hansbo and A. Szepessy. A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 84:175–192, 1990.

[8] T.Barth. A 3-D upwind Euler solver for unstructured meshes. AIAA-91-1548, 1991.

[9] D. Mavriplis. Three-dimensional unstructured multigrid for the Euler equations. AIAA-91-1549, 1991.

[10] H. Luo, J.D. Baum, R. Löhner, and J. Cabello. Adaptive edge-based finite element schemes for the Euler and Navier-Stokes equations on unstructured grids. AIAA-93-0336, 1993.

[11] J.G. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier-Stokes problem. IV: Error analysis for second-order time discretization. *SIAM Journal on Numerical Analysis*, 27:353–384, 1990.

[12] G. Lube and D. Weiss. Stabilized finite element methods for singularly perturbed parabolic problems. *Applied Numerical Mathematics*, 17:431–459, 1995.

[13] M. Cervera, R. Codina, and M. Galindo. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Engineering Computations*, 13(6):4–30, 1996.

[14] R. Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 156:185–210, 1998.

[15] R. Codina. An iterative penalty method for the finite element solution of the stationary Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 110:237–262, 1993.

[16] J. Baranger and K. Najib. Analyse numérique des écoulements quasi-Newtoniens dont la viscosité obéit à la loi puissance ou à la loi de Carreau. *Numerische Mathematik*, 58:35–49, 1990.

[17] A.A. Mohamad and R. Viskanta. Transient natural convection of low-Prandtl-number fluids in a differentially heated cavity. *International Journal for Numerical Methods in Fluids*, 13:61–81, 1991.

[18] A.N. Brooks and T.J.R. Hughes. Streamline upwind / Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equation. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[19] P.M. Gresho, S.T. Chan, R.L. Lee, and C.D. Upson. A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. Part 2: Applications. *International Journal for Numerical Methods in Fluids*, 4:619–640, 1984.

[20] M.S. Engelman and M.A. Jamnia. Transient flow past a circular cylinder: A benchmark solution. *International Journal for Numerical Methods in Fluids*, 11:985–1000, 1990.

[21] T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.

List of Figures

List of Tables

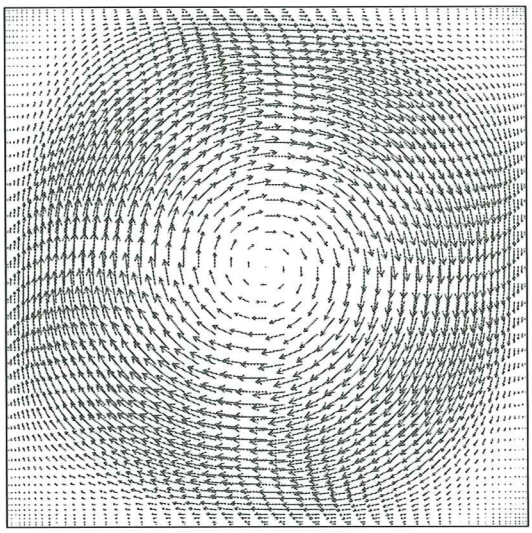

Figure 1: Velocity vectors for Example 1.

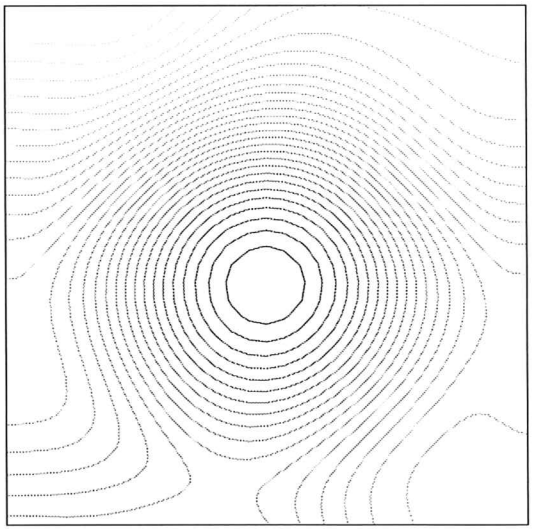

Figure 2: Pressure contours for Example 1.

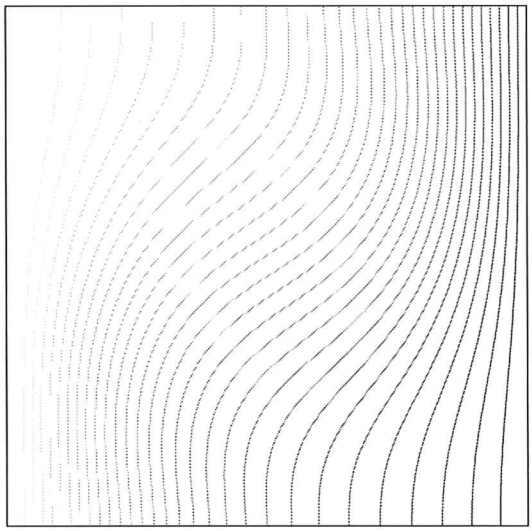

Figure 3: Temperature contours for Example 1.

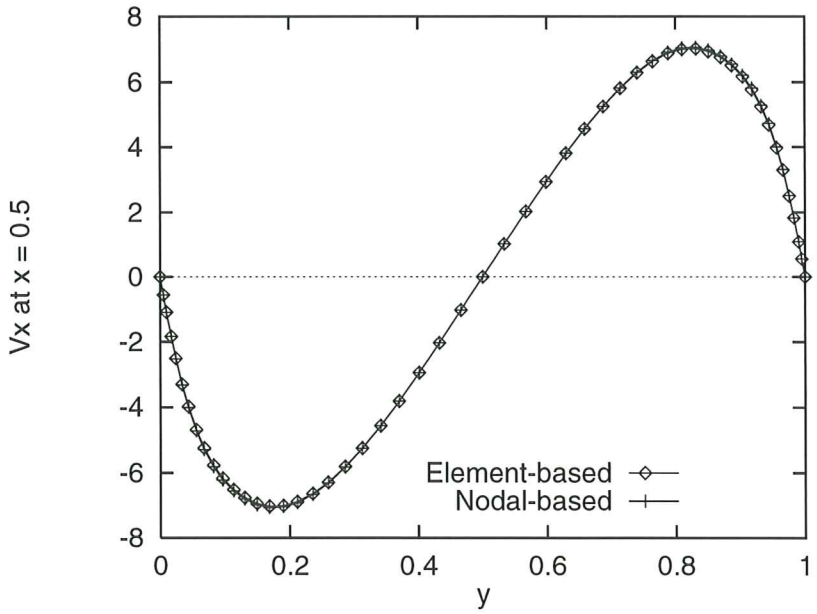

Figure 4: x-velocity at $x = 0.5$ for Example 1.

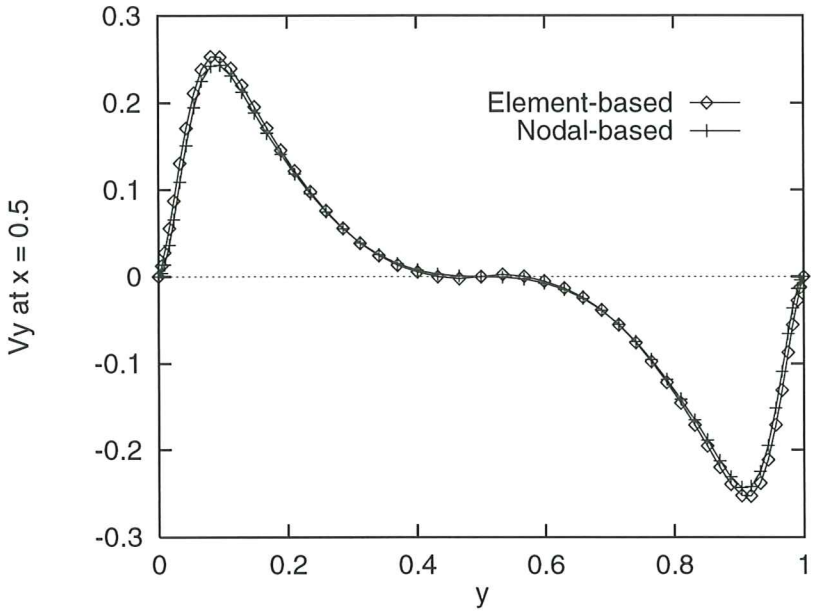

Figure 5: y-velocity at $x = 0.5$ for Example 1.

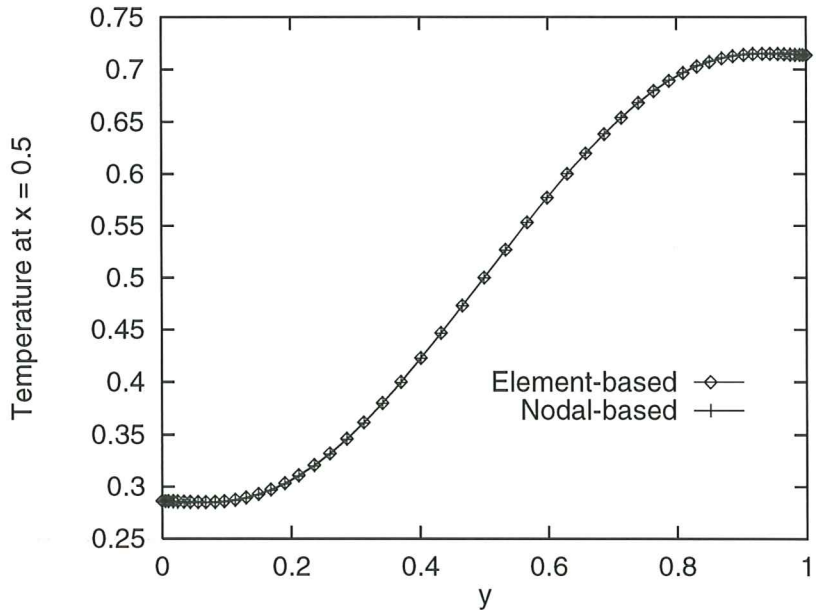

Figure 6: Temperature at $x = 0.5$ for Example 1.

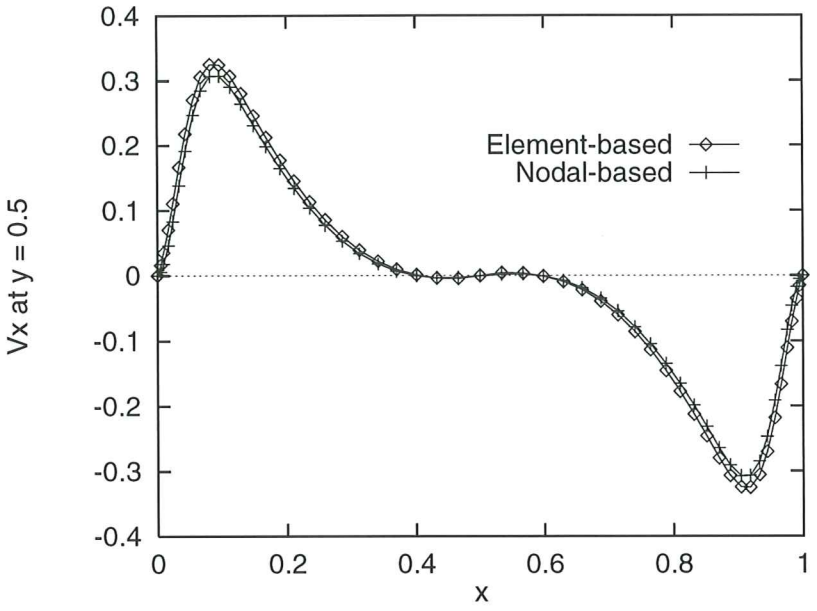

Figure 7: x-velocity at $y = 0.5$ for Example 1.

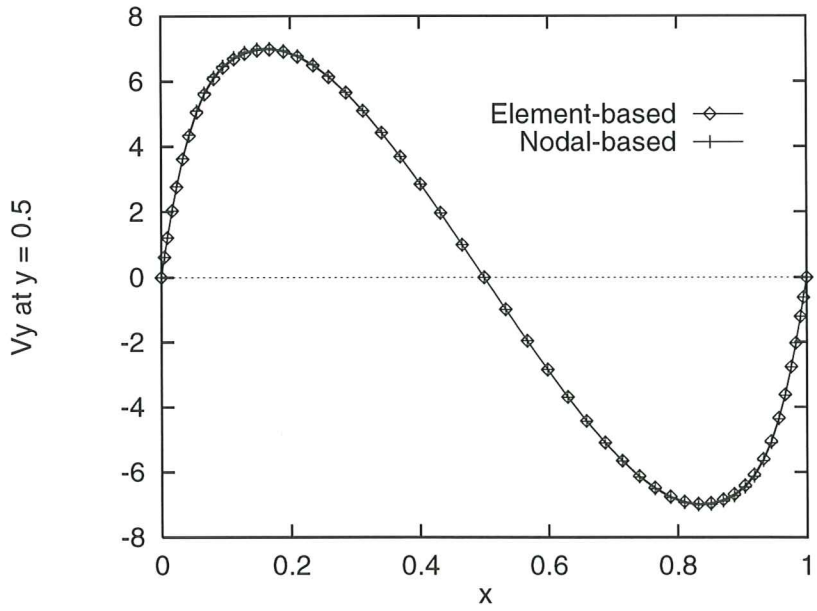

Figure 8: y-velocity at $y = 0.5$ for Example 1.

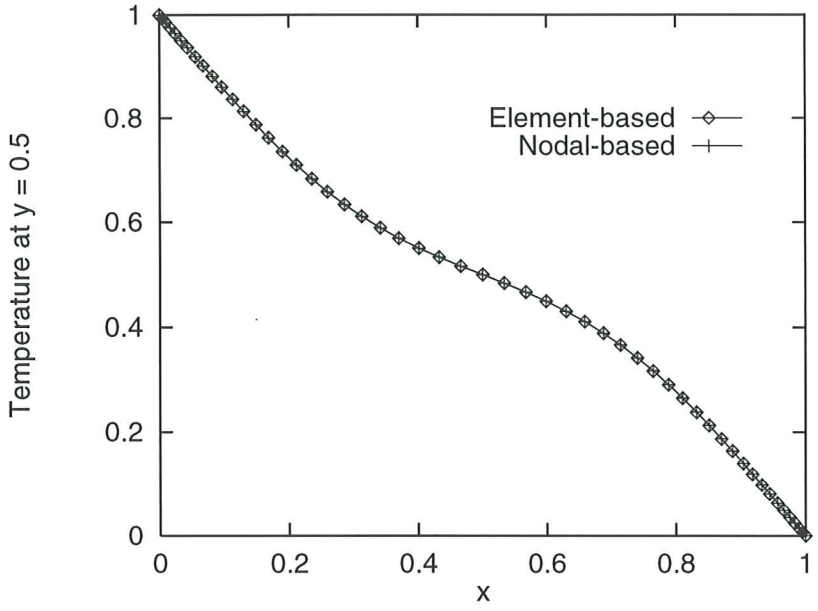

Figure 9: Temperature at $y = 0.5$ for Example 1.

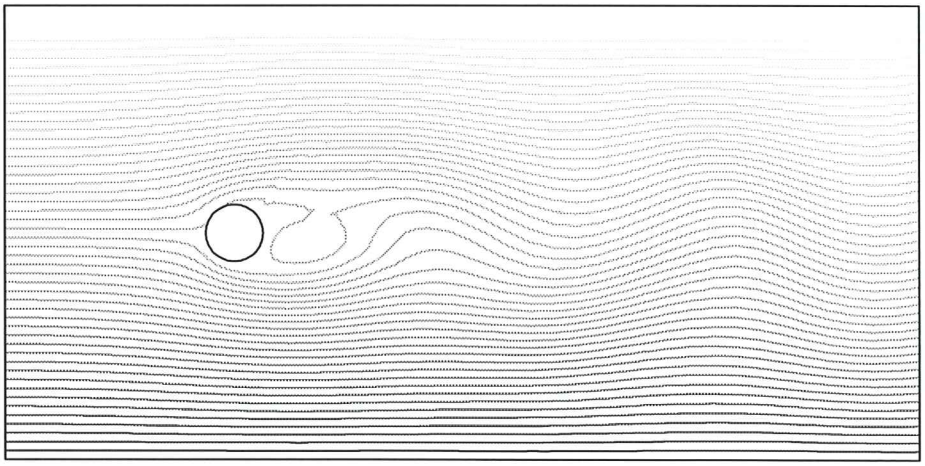

Figure 10: Streamlines at $t = 7$ for Example 2.

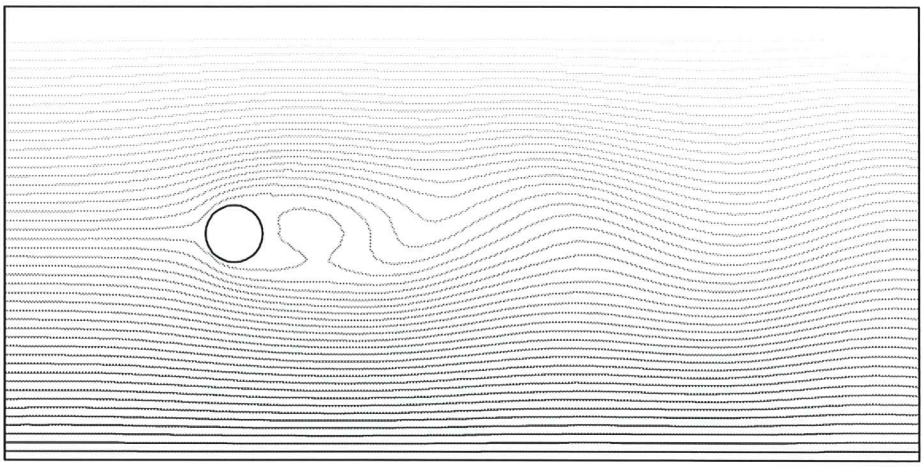

Figure 11: Streamlines at $t = 10$ for Example 2.

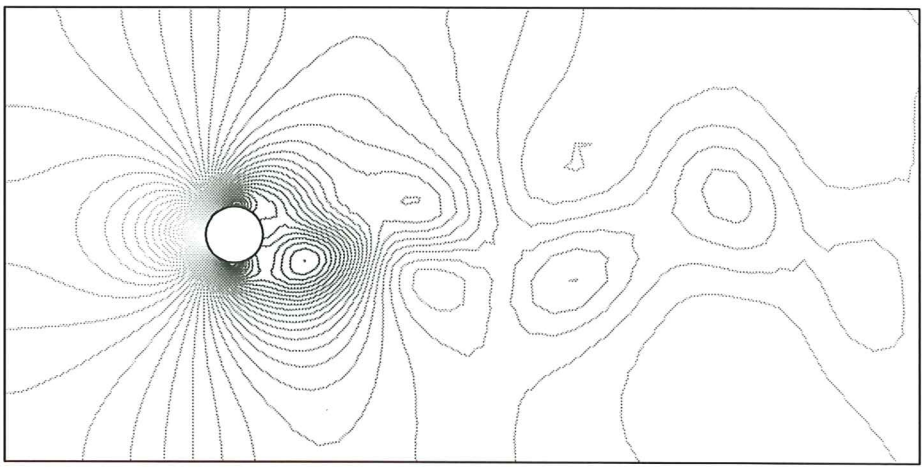

Figure 12: Pressure contours at $t = 7$ for Example 2.

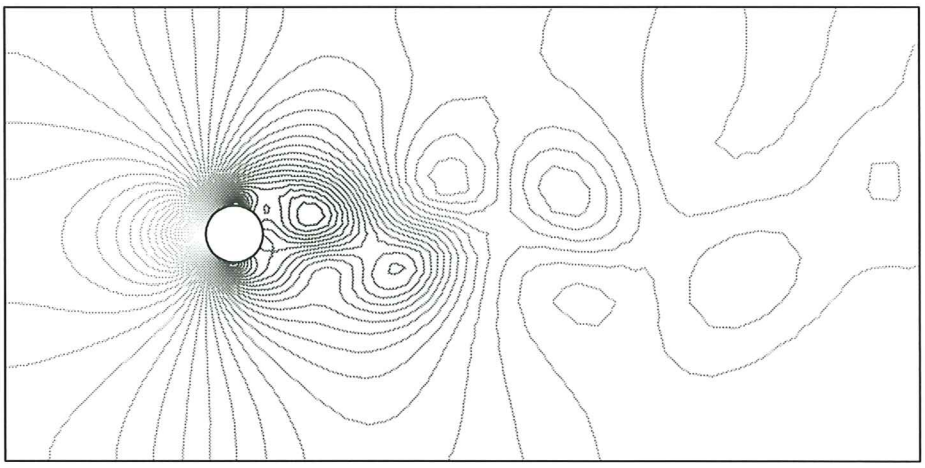

Figure 13: Pressure contours at $t = 10$ for Example 2.

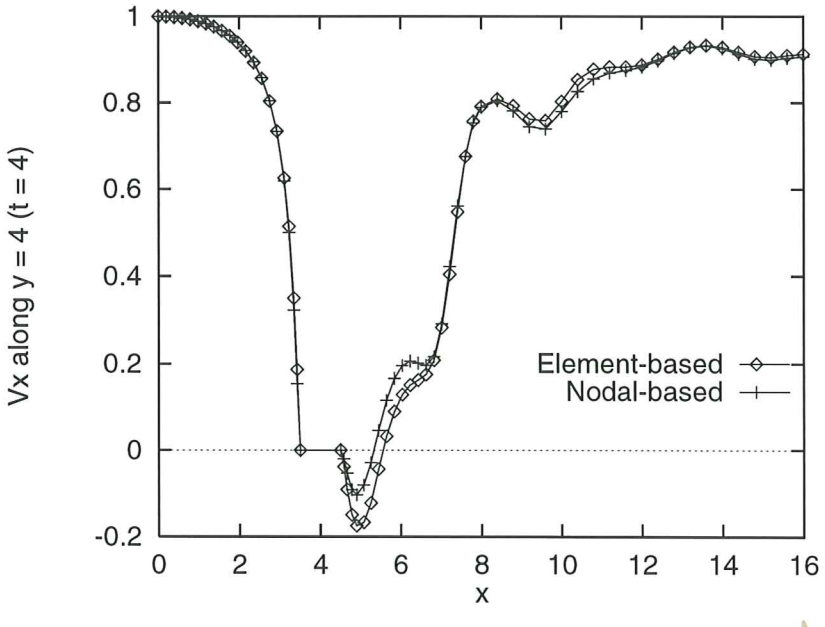

Figure 14: x-velocity at $y = 4$ and $t = 4$ for Example 2.

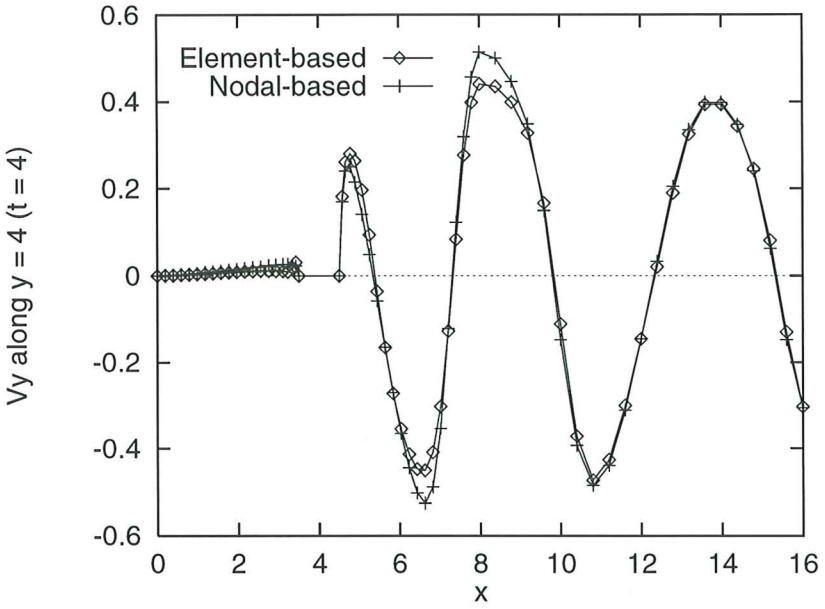

Figure 15: y-velocity at $y = 4$ and $t = 4$ for Example 2.

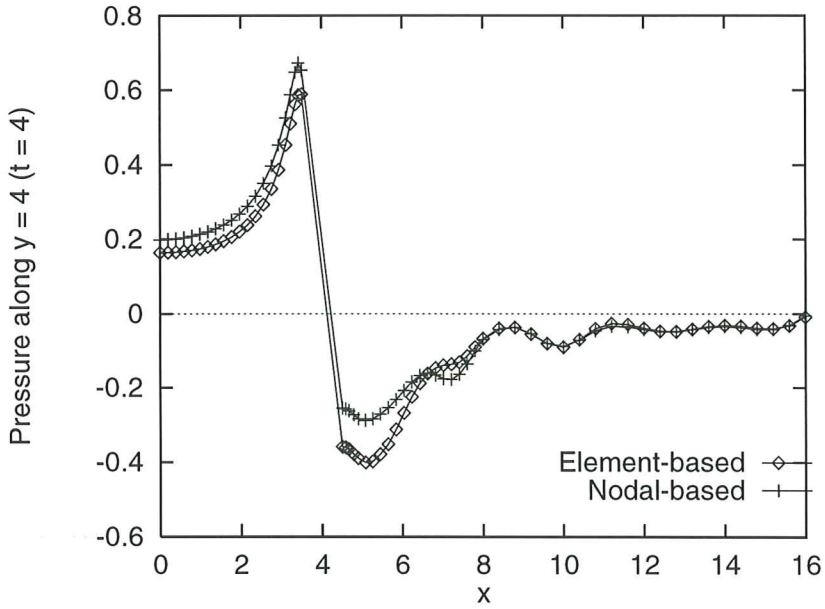

Figure 16: Pressure at $y = 4$ and $t = 4$ for Example 2.

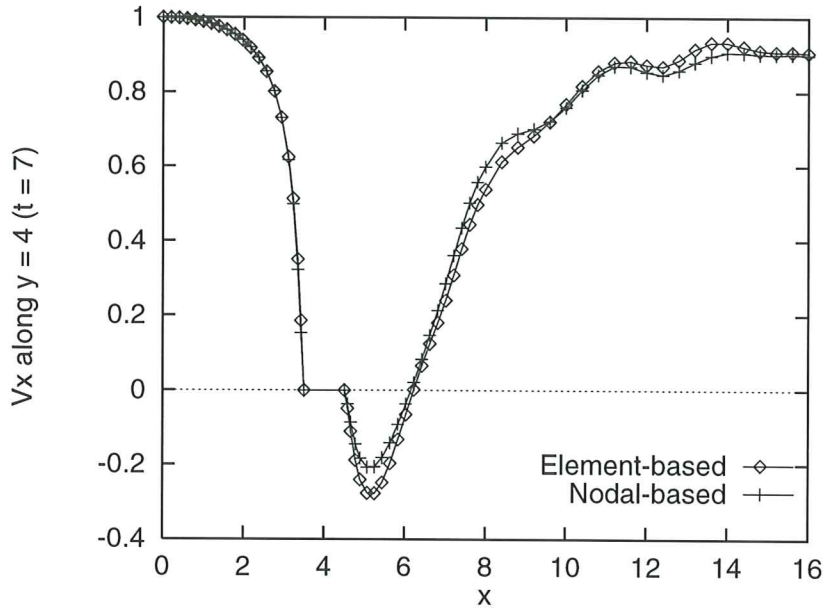

Figure 17: x-velocity at $y = 4$ and $t = 7$ for Example 2.

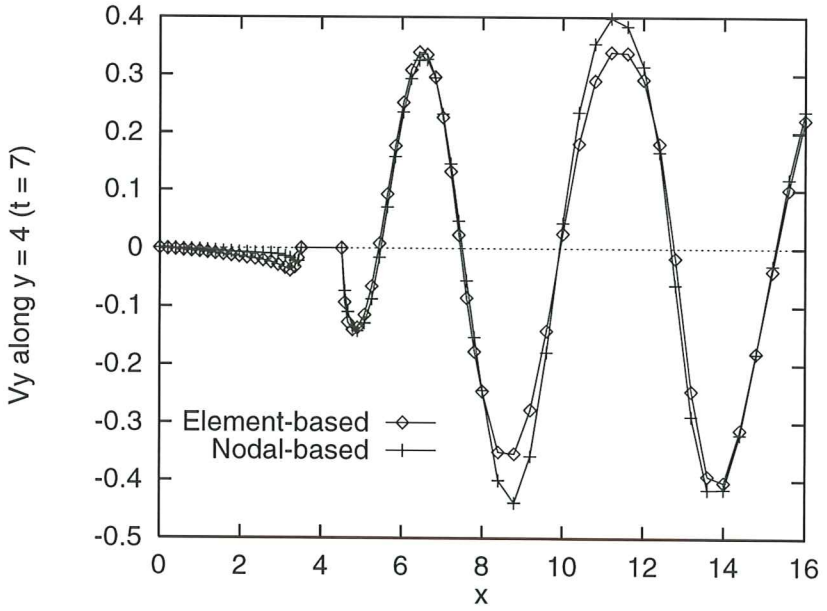

Figure 18: y-velocity at $y = 4$ and $t = 7$ for Example 2.

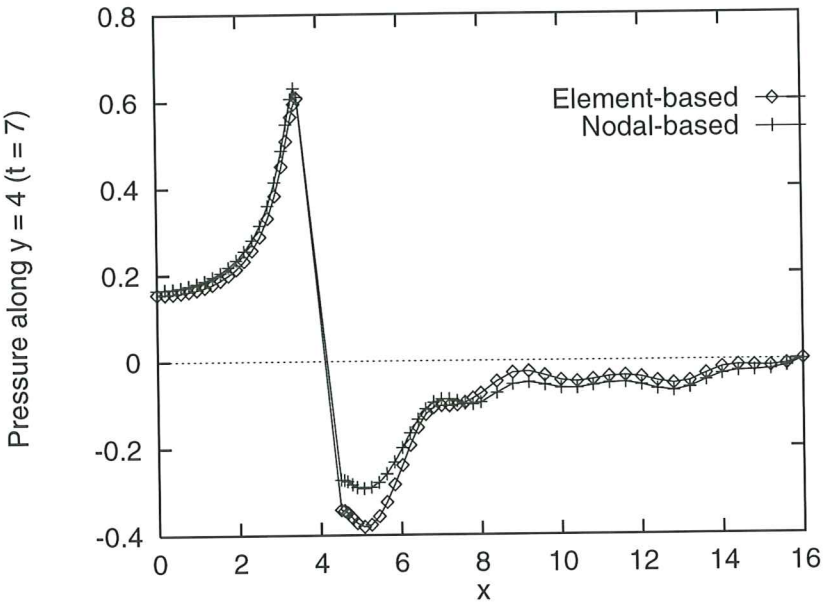

Figure 19: Pressure at $y = 4$ and $t = 7$ for Example 2.

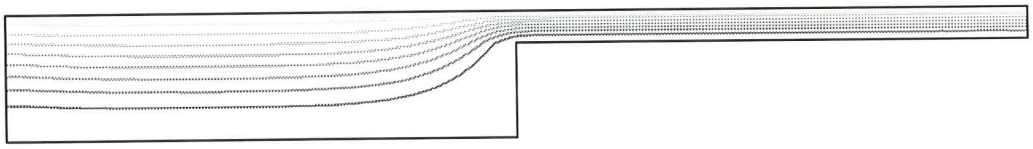

Figure 20: Streamlines for Example 3.

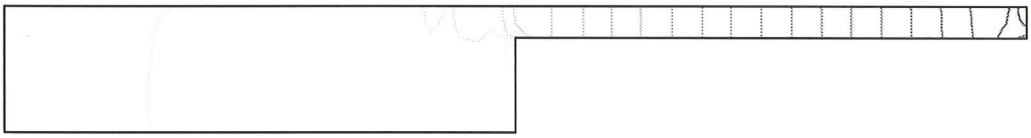

Figure 21: Pressure contours for Example 3.

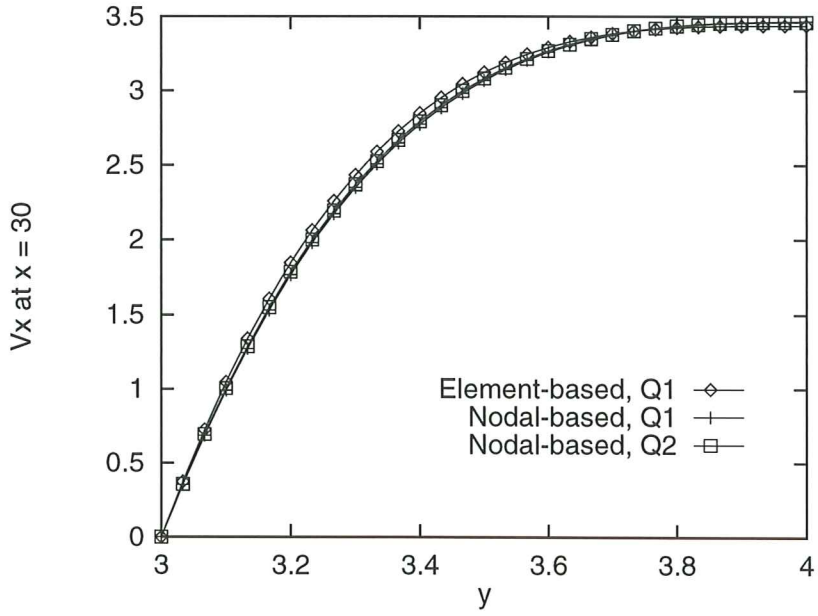

Figure 22: x-velocity at $x = 30$ for Example 3.

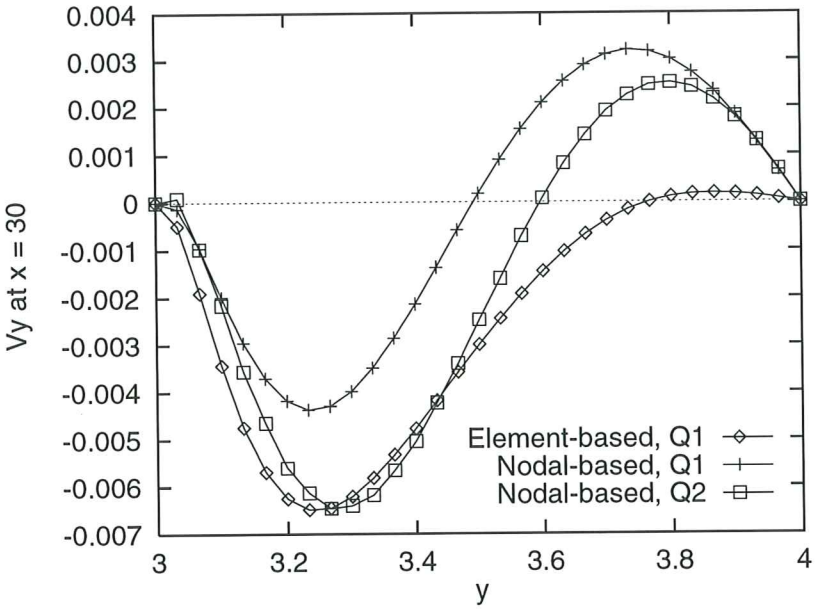

Figure 23: y-velocity at $x = 30$ for Example 3.

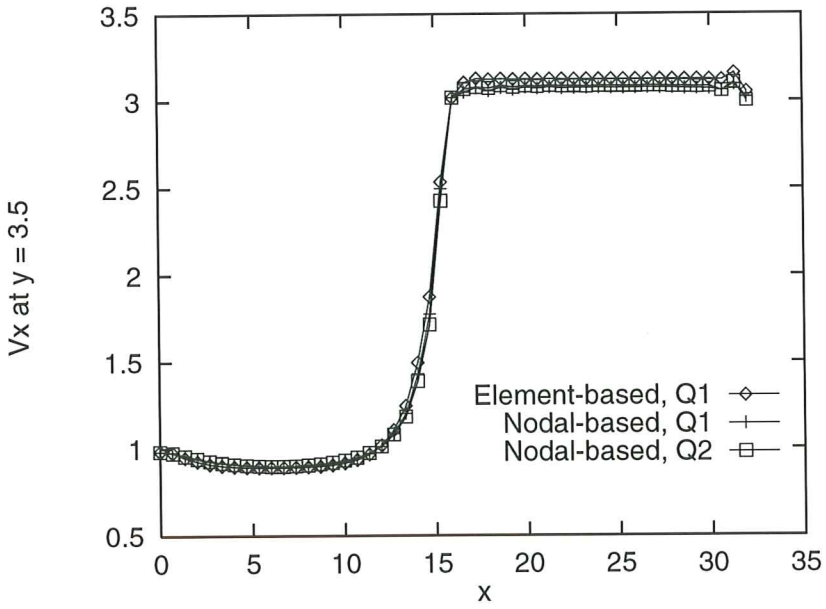

Figure 24: x-velocity at $y = 3.5$ for Example 3.

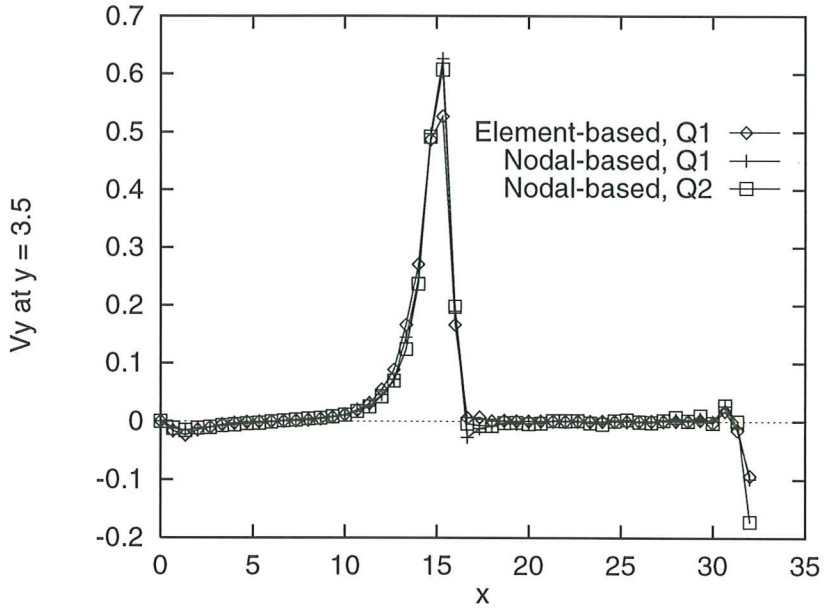

Figure 25: y-velocity at $y = 3.5$ for Example 3.

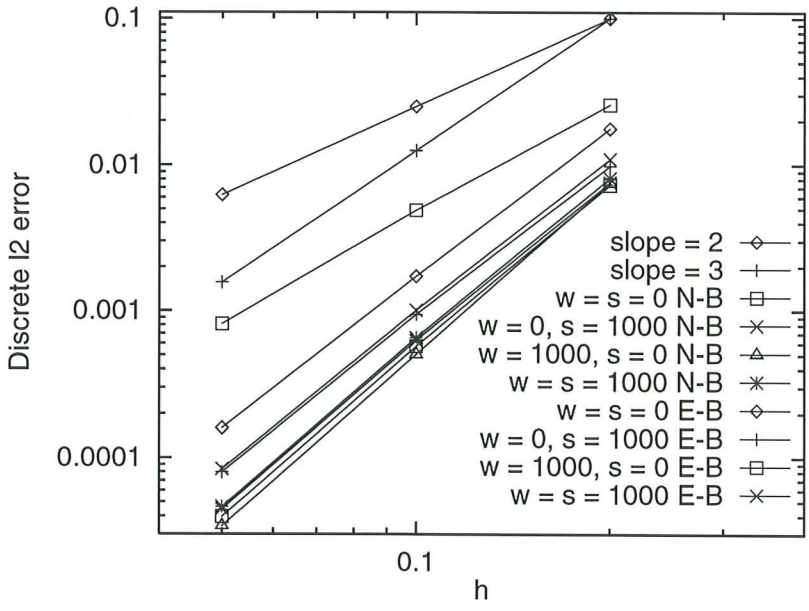

Figure 26: Discrete ℓ^2 errors for Example 4 for different values of $|\boldsymbol{\omega}| \equiv w$ and $\sigma \equiv s$ for the element-based (E-B) and nodal-based (N-B) implementations.

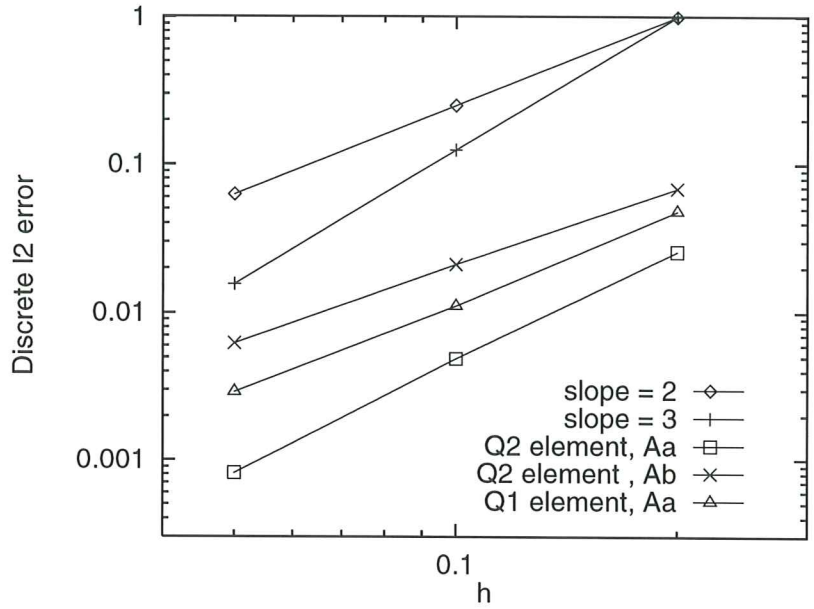

Figure 27: Discrete ℓ^2 errors for Example 4 for Q_2 and Q_1 elements using A^a or A^b in (42).

Mesh	Element-based	Nodal-based	**Factor**
tl1, 4 points	0.94	0.06	**15.67**
tl1, 1 point	0.35	0.06	**5.83**
hl1	0.71	0.09	**7.89**
tq1, 11 points	0.87	0.10	**8.70**
tq1, 4 points	0.43	0.10	**4.30**
hq1	2.48	0.21	**11.81**
tl2, 4 points	7.57	0.44	**17.20**
tl2, 1 point	2.81	0.44	**6.39**
hl2	5.67	0.68	**8.34**
tq2, 11 points	6.89	0.79	**8.72**
tq2, 4 points	3.44	0.79	**4.35**
hq2	19.98	1.87	**10.64**
tl3, 4 points	60.52	3.87	**15.64**
tl3, 1 point	22.65	3.87	**5.85**
hl3	44.60	6.14	**7.26**
tq3, 11 points	55.19	7.10	**7.77**
tq3, 4 points	27.14	7.10	**3.82**
hq3	165.04	15.95	**10.35**

Table 1: Comparison of the CPU time needed to construct the stiffness matrix in Example 5.

Mesh	f_{ele}	$\mathcal{M}_{\text{eb}}$	f_{con}	$\mathcal{M}_{\text{nb}}$
tl, 4 points	6	**336**	9	**58.5**
tl, 1 point	6	**102**	9	**58.5**
hl	1	**208**	27	**175.5**
tq, 11 points	0.75	**263.25**	26.625	**173.0625**
tq, 4 points	0.75	**100.5**	26.625	**173.0625**
hq	0.125	**280.125**	64	**416**

Table 2: Comparison of the asymptotic number of coefficients needed to store derivatives for the element-based implementation and integrals for the nodal-based implementation.