

Implementation of Conservative Schemes in an Edge-Based Finite Element Code



Héctor G. Espinoza Román
CIMNE
Universitat Politècnica de Catalunya

ADVISOR: Ramon Codina i Rovira

Thesis submitted for the degree of
Master of Science in Computational Mechanics

June 2011

1. Reviewer: Name

2. Reviewer:

Day of the defense:

Signature from head of committee:

Abstract

In this work a novel edge-based finite element implementation applied to specific equations is presented. It contains a full description on how we obtained it for the diffusion equation, stabilized convection-diffusion equation and stabilized Navier-Stokes equations. Additionally, classical benchmark problems are solved to show the capabilities of the new implementation.

As the differential equations we are interested in represent conservation statements, it would be desirable that the finite element approximation was exactly conservative (at least globally) independently of the mesh used. The present work revolves around that main objective.

The initial available edge-based approximation is not totally conservative. Of course it becomes more and more conservative as the mesh is refined. It has good h -convergence features and produces 'good solutions' (in the sense that the method does not introduce spurious oscillations and is numerically stable).

On the other hand, the edge-based approximation proposed is exactly globally conservative. Additionally it has good h -convergence features and produces 'good solutions'.

to my family,

who offered unconditional support during the course of this research work.

Acknowledgements

I would like to acknowledge my thesis director, Prof. Ramon Codina, whose expertise and patience helped me a lot during the process of the present research work.

Additionally, I would like to thank the help and opportune advice provided by Javier Principe, Christian Muñoz, Matias Avila and Joan Baiges.

In addition, a very special thanks to the people from CIMNE-TTS where I did my practical training. They were very flexible and facilitated my research work.

I would also like to stress the help of my family, from the distance or in person, were very supportive right from the start of this project.

Finally, I would like to express my gratitude to all the people that compose the Erasmus Mundus programme in Computational Mechanics which were always helpful and supportive. Special thanks to Lelia Zielonka for her help since the start of the master course.

Contents

List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	1
1.3 Element-Based and Edge-Based Implementations	2
1.4 Consistency and Conservation	5
1.4.1 Consistency	5
1.4.2 Conservation	5
1.5 Chapter Overview	7
2 Diffusion Equation	9
2.1 Problem Statement	9
2.1.1 Properties of the Diffusion Equation	10
2.1.2 Thermal conductivity of solids	11
2.1.3 Weak form of the problem	11
2.2 Numerical Approximation	12
2.3 Node-Based Implementation	13
2.3.1 Consistency	14
2.3.2 Conservation	15
2.4 Convergence Test	16

CONTENTS

3	Convection-Diffusion Equation	19
3.1	Problem Statement	19
3.1.1	Properties of the Convective Term	20
3.1.2	Weak Form of the problem	21
3.2	Numerical Approximation	21
3.3	Node-Based Implementation	24
3.3.1	Consistency	27
3.3.2	Conservation	28
3.4	Convergence Test	31
4	Navier-Stokes Equations	41
4.1	Problem Statement	41
4.1.1	Properties of the Navier-Stokes Equation	42
4.1.2	Weak Form of the problem	43
4.1.3	Linearization	44
4.2	Numerical Approximation	44
4.3	Node-Based Implementation	46
4.3.1	Consistency	49
4.3.2	Conservation	51
4.4	Convergence Test	54
5	Applications	59
5.1	Laminar Flow over a Forward-facing step of a Non-Newtonian fluid	59
5.1.1	Problem Description	59
5.1.2	Boundary Conditions	60
5.1.3	Meshing	60
5.1.4	Results	60
5.2	Thermally coupled flow in a cavity	63
5.2.1	Problem Description	63
5.2.2	Boundary Conditions	63
5.2.3	Meshing	63
5.2.4	Results	65

5.3	Flow over a cylinder	68
5.3.1	Problem Description	68
5.3.2	Boundary Conditions	68
5.3.3	Meshing	68
5.3.4	Results	69
5.4	3D Lid Driven Cavity Flow	71
5.4.1	Problem Description	71
5.4.2	Boundary Conditions	71
5.4.3	Meshing	71
5.4.4	Results	73
6	Conclusions	75
6.1	Conclusions	75
6.2	Future Research	76
	References	77

CONTENTS

List of Figures

1.1	Arbitrary computational domain	2
1.2	Discretized Domain	3
1.3	Element-based and Edge-based Algorithms	4
1.4	Tank Flow	6
2.1	Diffusion Equation Convergence Test	17
3.1	Influence of Local Peclet number on numerical instabilities	23
3.2	Triangle meshes for the convergence test	33
3.3	Convection Diffusion Equation Reference Convergence Test	34
3.4	Convection Diffusion Equation Convergence Test	35
3.5	Convection Diffusion Equation Convergence Test	35
3.6	Convection Diffusion Equation Convergence Test	36
3.7	Convection Diffusion Equation Convergence Test	36
3.8	Convection Diffusion Equation Convergence Test	37
3.9	Convection Diffusion Equation Convergence Test	37
4.1	Navier Stokes Convergence Test	55
5.1	Example 1 Domain	59
5.2	Example 1 Boundary Conditions	60
5.3	Example 1 Mesh	60
5.4	Example 1 velocity contour fill	61
5.5	Example 1 pressure contour fill	61
5.6	Example 1 streamlines	61

LIST OF FIGURES

5.7	Example 1 Pressure Contour Fill with finer mesh	62
5.8	Example 2 Velocity Boundary Conditions	64
5.9	Example 2 Temperature Boundary Conditions	64
5.10	Example 2 Mesh	65
5.11	Example 2 Velocity Contour Fill	66
5.12	Example 2 Pressure Contour Fill	66
5.13	Example 2 Streamlines	67
5.14	Example 2 Temperature Contour Fill	67
5.15	Example 3 Boundary Conditions	68
5.16	Example 3 Mesh	69
5.17	Example 3 Streamlines	69
5.18	Example 3 Velocity Contour Fill	70
5.19	Example 3 Pressure Contour Fill	70
5.20	Example 4 Boundary Conditions	72
5.21	Example 4 Mesh	72
5.22	Example 4 y -velocity on a line passing trough points (0.0,0.5,0.5) and (1.0,0.5,0.5)	73
5.23	Example 4 x -velocity on a line passing trough points (0.5,0.00,0.5) and (0.5,1.0,0.5)	74

List of Tables

2.1	Diffusion Convergence Test	17
4.1	Navier Stokes Convergence Test (L2-Error of the velocity)	55

LIST OF TABLES

Chapter 1

Introduction

In this chapter we are going to cover, in a very simplified way, a variety of topics needed to have a general idea to understand the edge-based finite element approximations proposed.

1.1 Motivation

As stated in [2] the lack of conservation of the edge-based implementation led us to implement ad-hoc approximations in order to alleviate this issue.

Some approximations are consistent but not conservative and work very well. Some others are conservative but not consistent and do not work at all. We try to fulfil both conditions using ad hoc approximations and corrections. In section 1.4 the concepts of consistency and conservation are explained in more detail.

1.2 Objectives

The objectives are the following:

- Implement a conservative approximation for the diffusive term in the diffusion equation.
- Implement a conservative approximation for the convective and stabilization terms in the convection-diffusion equation.

1. INTRODUCTION

- Extend the results obtained from the convection-diffusion equation to Navier-Stokes equations.

1.3 Element-Based and Edge-Based Implementations

Element-Based and Edge-Based can be described as methods to calculate the “stiffness matrix” in a finite element code. The final objective is exactly the same and both methods start with identical information. The difference is the flow of the calculations.

Let us assume we are solving a given problem in an arbitrary domain Ω with boundary $\partial\Omega$ as shown in figure 1.1.

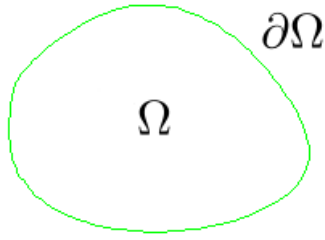


Figure 1.1: Arbitrary computational domain

The information which both methods start is: a partitioned domain denoted Ω^e with e ranging from 1 to the number of elements in which the domain is partitioned and initial/prescribed values of the unknown(s). In figure 1.2 we can see the domain from figure 1.1 discretized using various types of elements.

Let us suppose our generic unknown is u . We denote U^a to the approximate value of the unknown at nodal point \mathbf{x}^a and $u(\mathbf{x}^a)$ to the exact value of the unknown at the nodal point \mathbf{x}^a . Now, looking at the discretization of the domain in figure 1.2, U^a is defined in each of the nodal points; we denote $u_h(\mathbf{x}^a)$ to the approximate solution at any point \mathbf{x}^a .

Using interpolation notation and labelling n_{pts} the number of nodal points in the mesh, we have:

$$x_i = \sum_{a=1}^{n_{pts}} N^a x_i^a \qquad u = \sum_{a=1}^{n_{pts}} N^a U^a \qquad (1.1)$$

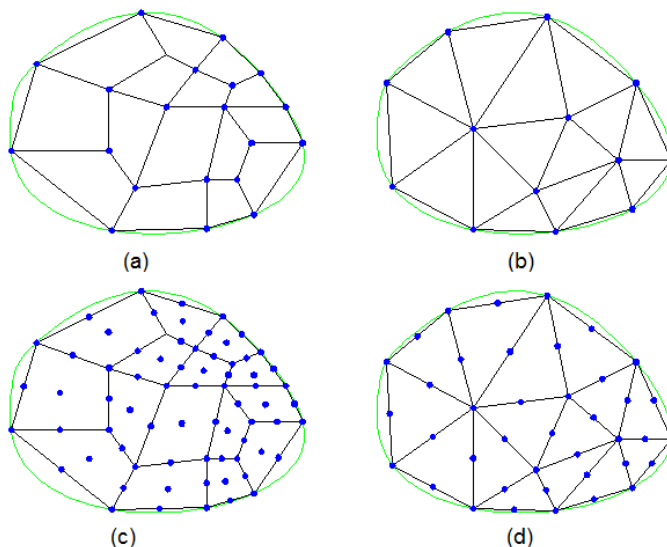


Figure 1.2: Discretized Domain into (a) linear quadrilaterals (Q1 elements, 4 nodes), (b) linear triangles (P1 elements, 3 nodes), quadratic quadrilaterals (Q2 elements, 9 nodes) and (d) quadratic triangles (P2 elements, 6 nodes)

where N^a is a standard finite element interpolation function of given degree p defined in each Ω^e .

Additionally, we interpolate any material property in the same fashion. For example kinematic viscosity ν :

$$\nu = \sum_a^{n_{pts}} N^a \nu^a \tag{1.2}$$

Element-based implementations perform a loop over elements, then a loop over integration points (gauss points) to calculate the element stiffness matrix; finally assemble the element matrix into the global matrix.

On the other hand, edge-based implementations start with pre-assembled global matrices dependent of just the geometry. Then perform a loop over all nodal points, then a loop over the neighbours of the current nodal point and computes the contribution to the stiffness matrix directly, without referring to the elements. To do that, we require to approximate the integrals of interest as a combination of geometry-dependent integrals and nodal values of the properties.

For example, let us suppose we want to approximate the diffusive term in the

1. INTRODUCTION

following way:

$$\int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega \approx \nu \int_{\Omega} \nabla v_h \cdot \nabla u_h \, d\Omega \quad (1.3)$$

where v_h is a function (called test function), u_h is the unknown (treated as a function) and ν is the diffusion coefficient.

As you know, the diffusion coefficient ν is not always constant and can vary; so the approximation done is not exact. The question that arises here is: Where do we evaluate ν in order to be more or less exact? The answer is not very simple and we will try to address that issue in the present work. Something similar happens with the convective term and stabilization terms introduced later on.

In figure 1.3 we can see a model algorithm of element-based and edge-based implementations (Matlab syntax).

<pre>for ielem=1:nelem for iint=1:nint Ke = Ke + Contrib(ielem,iint) end Assemble(K,Ke) end</pre>	<pre>for ipt=1:npts for ineig=1:nneig jpts = Neighbours{ipts}(ineig) K(ipts,jpts) = Contrib(ipts,jpts) end end</pre>
(a)	(b)

Figure 1.3: Element-based and Edge-based Algorithms

In non-linear problems, the global stiffness matrix is recalculated at each iteration of non-linearity. This makes the element-based approach expensive and favours the edge-based approach, because it uses just nodal values and the matrices are already precomputed.

Edge-based implementations can perform better than element-based when no re-meshing or mesh update is necessary. If the mesh changes, edge-based approach might not be the best performing.

1.4 Consistency and Conservation

1.4.1 Consistency

Consistency means that the exact solution of the continuous problem are solutions of the discrete one, provided they belong to the finite element space. the finite element.

For example, let us suppose we are using P1 (triangle) or Q1 (quadrilateral) elements. These elements are of order $p = 1$. An approximation is consistent in this sense if any given discretization is able to reproduce exactly a linear varying (exact) solution.

In the present work we check consistency of the edge-based approximation for the constants.

1.4.2 Conservation

An approximation is conservative if it “conserves” the quantities being transported or diffused. In this sense, conservation means that the amount of the conserved variable that goes into the computational domain is equal to what goes out plus what is generated.

There are many physical magnitudes which follow conservation laws, for instance:

- Conservation of mass-energy
- Conservation of linear momentum
- Conservation of angular momentum
- Conservation of electric charge
- Conservation of mass (when no nuclear reactions happen)
- Conservation of energy (when no nuclear reactions happen)

As an example, let us think in fluid flow. Let us suppose there is one inlet and one outlet in a tank. Additionally, let us suppose the process is steady state, there are not nuclear reactions and velocities are small compared with the speed of light (non-relativistic analysis). Therefore, the mass of fluid inside the tank is constant. As the mass is the conserved quantity in this case, we can infer that the mass that goes into

1. INTRODUCTION

the domain is exactly equal to the mass that comes out. This is the intuitive idea of conservation.

In figure 1.4 we can see the flow problem described before. Fluid enters the tank in the upper-left corner and leaves the domain in the bottom-right corner. Inlet and outlet are both the same size. The colours represent the velocity magnitude.

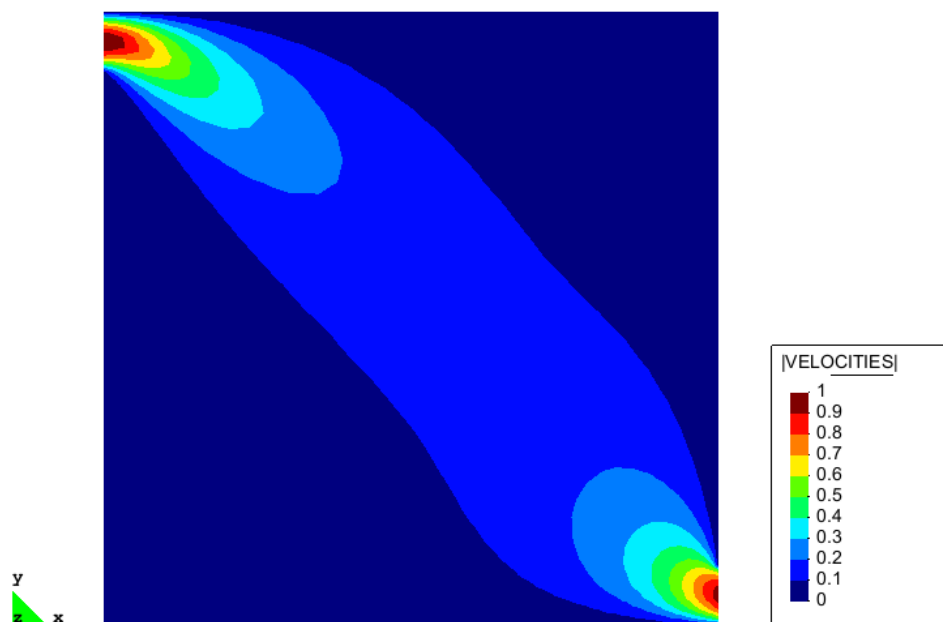


Figure 1.4: Tank Flow - Velocity Contours

As the velocity contours of figure 1.4 were obtained with the consistent implementation, conservation is not ensured. We can see that the outflow is lower than the inflow: mass loss happens.

Conservation can be described in differential form and integral form. Additionally, both descriptions can be related using the divergence theorem:

- **Differential Form:** in the continuous, conservation holds in any control volume. Particularly in a control volume that tends to zero, hence it holds in a point, so we can express a conservation statement in differential form:

i.e. mass conservation: $\nabla \cdot (\rho \mathbf{u}) = 0$, where ρ is the mass density and \mathbf{u} is the velocity at any spatial point \mathbf{x} .

This lead us to the concept of local conservation.

- Integral Form: normally a given discretization does not have enough points to describe a complex solution exactly, hence conservation sometimes does not hold exactly from point to point. A desired feature of a finite element formulation would be “global conservation” and that is what we want with this work.

i.e. mass conservation $\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} d\Gamma = 0$, where \mathbf{u} is the velocity, \mathbf{n} is the unit outward-pointing normal to the boundary $\partial\Omega$ of the domain.

This lead us to the concept of global conservation, and this is actually what we do in the code to check conservation: perform the surface integral.

1.5 Chapter Overview

In the first chapter, a general overview of finite elements, consistency and conservation is given.

In the second, third and fourth chapters we propose some edge-based approximations for the diffusion, convection-diffusion and Navier-Stokes equations respectively. First we start describing the model problem, then various approximations are proposed. After that, we check and/or correct the approximations for consistency and conservation. Finally a convergence test is performed to determine whether optimal convergence rate is obtained or not.

In the fifth chapter, various examples are solved with the best-performing proposed approximations.

Finally, in the last chapter, general conclusions are drawn according to the results obtained.

1. INTRODUCTION

Chapter 2

Diffusion Equation

2.1 Problem Statement

In this chapter we are going to consider the scalar steady-state diffusion equation. From hereafter during this chapter, this equation will serve to model heat transfer by pure conduction, consequently the terminology used for unknowns and boundary conditions will have relation with heat transfer.

The equation that describes the problem is:

$$\nabla \cdot (-\nu \nabla u) = f \tag{2.1}$$

with the following boundary conditions:

$$\begin{aligned} u(\mathbf{x}) &= \bar{u}(\mathbf{x}) && \text{on } \Gamma_D \\ \nu \nabla u \cdot \mathbf{n} &= \bar{h}(\mathbf{x}) && \text{on } \Gamma_N \\ \Gamma_D \cup \Gamma_N &= \partial\Omega \\ \Gamma_D \cap \Gamma_N &= \emptyset \end{aligned} \tag{2.2}$$

to be solved over $\Omega \subset \mathbb{R}^{n_{sd}}$, where n_{sd} is the space dimension, u is the unknown (temperature), f is the forcing term (heat source per unit volume), ν is the diffusion coefficient (heat conductivity), \bar{u} is a prescribed value of the solution, \bar{h} is a prescribed heat flow per unit area and \mathbf{n} is the outward-pointing normal to the boundary.

The prescribed heat flow, \bar{h} , is taken as positive when heat goes into the domain through the boundary. Negative otherwise.

2. DIFFUSION EQUATION

The diffusion coefficient ν is always greater than zero. The forcing term f is taken as positive when heat is added to the domain and negative when heat is extracted from the domain.

In the general case, heat conductivity is not constant and can depend on space coordinates (i.e. a domain composed by different materials) or on the solution itself (i.e. $\nu(\mathbf{x}, u)$) In the present work constant, space dependant and solution dependant diffusion coefficient has been considered to validate de formulation.

2.1.1 Properties of the Diffusion Equation

According to [3] the diffusion process has three basic properties: Conservation, Smoothing and Invariance.

Conservation can be explained examining equation (2.1). Integrating in an arbitrary domain Ω both members we have:

$$\int_{\Omega} \nabla \cdot (-\nu \nabla u) \, d\Omega = \int_{\Omega} f \, d\Omega \quad (2.3)$$

Using the divergence theorem, which relates the volume integral with a surface integral we can write:

$$\int_{\partial\Omega} (-\nu \nabla u) \cdot \mathbf{n} \, d\Gamma = \int_{\Omega} f \, d\Omega \quad (2.4)$$

Equation (2.4) expresses that the net heat flow across the boundary of the domain is equal to the heat generated into the domain. This explains the (energy) conservation property of the diffusion equation.

The smoothing property of the diffusion equation tends to homogenize in time any gradient of the unknown, hence making it smooth. For example, let us assume a 1D problem with no force term and constant diffusion coefficient of equation (2.1).

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \quad (2.5)$$

Equation (2.5) can be interpreted in this way: the time variation of u ($\frac{\partial u}{\partial t}$) is directly proportional to the local curvature of u ($\frac{\partial^2 u}{\partial x^2}$). Therefore, intervals where the local curvature is positive (local minima) will have a positive time derivative, which

means the values will tend to increase. Exactly the same happens in intervals where the local curvature is negative. Consequently, the solution tends to be smooth.

Invariance under reflections and translations refers to the symmetry of the diffusion process. In matrix form this invariance reflects in the symmetry of the diffusion matrix. This can be easily explained using a change of variable. First, let us suppose $u(x)$ is the solution of a particular steady-state diffusion problem. Then, let us take $\bar{x} = -x$ as a new reference system. Then $u(\bar{x})$ is still solution of equation (2.1) because it involves derivation twice, so using the chain rule for derivation the minus sign cancels out.

2.1.2 Thermal conductivity of solids

When analysing heat transfer, the diffusion coefficient ν refers to the heat conductivity. Heat conductivity is the ability of a material to conduct heat energy through it due to a gradient of temperature.

According to the magnitude of the thermal conductivity can be classified in two big groups: conductors and insulators.

- Conductor materials are materials with a high thermal conductivity. They are used to provide a physical barrier between two points and maintain the ability of transferring heat. For example copper is used in heat exchangers or aluminium in heat-sinks to dissipate heat generated in electronic components.
- Insulator materials are materials with low thermal conductivity. They are used to prevent heat transfer from one place to another. For example glass wool prevents ambient heat entering inside the cold environment of a fridge. Elastomeric foams prevent heat loss to the environment in hot water pipes.

2.1.3 Weak form of the problem

Let us define the following functional spaces:

$$V = \{u(\mathbf{x}) \in H^1(\Omega) \mid u = \bar{u} \text{ on } \Gamma_D\}$$

$$V_0 = \{u(\mathbf{x}) \in H^1(\Omega) \mid u = 0 \text{ on } \Gamma_D\}$$

After defining those spaces, the weak form of the problem reads as follows: Find

2. DIFFUSION EQUATION

$u \in V$ such that:

$$\int_{\Omega} \nabla v \cdot \nu \nabla u \, d\Omega = \int_{\Omega} v f \, d\Omega + \int_{\Gamma_N} v \bar{h} \, d\Gamma \quad (2.6)$$

for all test functions $v \in V_0$

2.2 Numerical Approximation

Now we can approximate the weak form of the problem defined in (2.6) using finite elements.

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} . Let u_h be the finite element approximation of the unknown u ; and, v_h the test functions associated with $\{\Omega^e\}$.

Hence, the finite element problem reads: Find $u_h \in V_h$ such that:

$$\int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega = \int_{\Omega} v_h f \, d\Omega + \int_{\Gamma_N} v_h \bar{h} \, d\Gamma \quad (2.7)$$

for all test functions $v_h \in V_{0,h}$

To obtain the algebraic version of the problem we need some definitions. Let n_{pts} be the total number of nodes of the finite element mesh and let N^a be the standard finite element interpolation function associated to node a , where $a = 1, \dots, n_{pts}$.

The unknown is interpolated as:

$$u_h = \sum_{a=1}^{n_{pts}} N^a U^a \quad (2.8)$$

We use upper-case letters (i.e. U^a) to denote the nodal value at node a of the corresponding lower-case variable.

The test functions are taken as:

$$v_h = N^b, \quad b = 1, \dots, n_{pts} \quad (2.9)$$

2.3 Node-Based Implementation

In this section, we are going to express the approximate problem in terms of solely geometrical dependant integrals:

$$\begin{aligned} \int_{\Omega} N^b N^a \, d\Omega \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega, \quad i, j = 1, \dots, n_{sd} \end{aligned} \quad (2.10)$$

for $a, b = 1, \dots, n_{pts}$

With the integrals in (2.10) and taking $v_h = N^b$, we can approximate the diffusive term in various forms:

$$\int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega = \sum_{a=1}^{n_{pts}} \nu^a \left(\sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \right) U^a \quad (2.11a)$$

$$\int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega = \sum_{a=1}^{n_{pts}} \nu^b \left(\sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \right) U^a \quad (2.11b)$$

$$\int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega = \sum_{a=1}^{n_{pts}} \frac{\nu^b + \nu^a}{2} \left(\sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \right) U^a \quad (2.11c)$$

If we consider the matrix form of the problem, the system of equations to solve will be:

$$\underline{\underline{L}} \underline{U} = \underline{\underline{M}} \underline{F} + \underline{B} \quad (2.12)$$

where double underline means a matrix and single underline means a column vector.

Therefore, the diffusion matrix $\underline{\underline{L}}$ will be defined as one of this expressions:

$$L^{ba} = \nu^a \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad (2.13a)$$

$$L^{ba} = \nu^b \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad (2.13b)$$

$$L^{ba} = \frac{\nu^b + \nu^a}{2} \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad (2.13c)$$

2. DIFFUSION EQUATION

2.3.1 Consistency

For simplicity, let us consider problem (2.7) without forcing term. Additionally, let us consider all the boundary conditions are of Dirichlet type: $\bar{u} = 1$. Therefore, $\Gamma_N = \emptyset \wedge \Gamma_D = \partial\Omega$. In this situation the solution is constant and equal to 1 in all nodal points.

This behaviour has to be reproduced exactly by the finite element solution because, even with linear interpolation functions, the exact solution belongs to the finite element space. Consequently, this condition has to hold:

$$\sum_a^{n_{pts}} L^{ba} = 0 \quad \forall b = 1, \dots, n_{pts} \quad (2.14)$$

It can be readily observed that the evaluation method proposed in (2.13b) satisfies this condition, but not the others. So here we propose a correction method to solve this issue: evaluating the main diagonal term of the matrix as the negative of the sum of the off-diagonal terms, so the final form of the matrix $\underline{\underline{L}}$ will be any of the following:

$$L^{ba} = \nu^a \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad \forall a \neq b \quad (2.15a)$$

$$L^{ba} = \nu^b \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad \forall a \neq b \quad (2.15b)$$

$$L^{ba} = \frac{\nu^b + \nu^a}{2} \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad \forall a \neq b \quad (2.15c)$$

and

$$L^{bb} = - \sum_{a \neq b}^{n_{pts}} L^{ba} \quad (2.15d)$$

With the definition provided in (2.15) the consistency condition (2.14) is satisfied by any of the evaluation methods proposed.

Let us prove it.

$$\sum_a^{n_{pts}} L^{ba} = L^{bb} + \sum_{a \neq b}^{n_{pts}} L^{ba} \quad \forall b = 1, \dots, n_{pts}$$

Replacing the definition of L^{bb} as in (2.15d)

$$\sum_a^{n_{pts}} L^{ba} = - \sum_{a \neq b}^{n_{pts}} L^{ba} + \sum_{a \neq b}^{n_{pts}} L^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}$$

This demonstrates (2.14)

2.3.2 Conservation

For simplicity, let us consider problem (2.7) with all boundary conditions of Neumann type (case in which the solution would not be unique). Furthermore, suppose we take the test function v_h constant. Hence, problem (2.7) imply:

$$- \int_{\partial\Omega} \bar{h} \, d\Gamma = \int_{\Omega} f \, d\Omega \quad (2.16)$$

which, in physical terms, represents that the net heat flow leaving the domain (heat out minus heat in) is equal to what is generated inside it (remember that if f is positive, heat energy is added to the domain). This is physically consistent.

However, equation (2.7) is not enforced for constant test functions v_h but only for test functions of the form $v_h = N^b$, $b = 1, \dots, n_{point}$. Since the addition of all shape functions N^b is 1, equation (2.16) can also be obtained taking $v_h = N^b$ in equation (2.7) and adding up over b .

Then, in matrix form, the condition in (2.16) can be expressed as:

$$\sum_b \sum_a L^{ba} U^a = 0 \quad (2.17)$$

in order to have global conservation.

Now let us have a look to the consistent nodal formulations proposed in (2.15). It can be observed that the only formulation that satisfies this condition is (2.15c): Using (2.14) and the symmetry of (2.15c) we can say that $\sum_b L^{ba} = 0$, then:

$$\sum_b \sum_a L^{ba} U^a = \sum_a \left(\sum_b L^{ba} \right) U^a = 0$$

that demonstrates (2.17).

Anyway, in the following section, we will test the convergence of various formulations mentioned in order to determine their convergence properties.

2. DIFFUSION EQUATION

2.4 Convergence Test

Reference [6] shows two a priori error estimates for elliptic partial differential equations. One of the error estimates is in H^1 -norm and the other in L^2 -norm. For simplicity, we are going to use the error estimator in the L^2 -norm.

Let $u \in V$ be the exact solution of the variational problem (2.6) and u_h its approximation using finite elements of order p . Additionally, let $u \in H^{r+1}(\Omega)$ for a given $r > 0$. Then, the following a priori error estimate in the L^2 -norm holds:

$$\| u - u_h \|_{L^2} < Ch^{s+1} | u |_{H^{s+1}}, \quad s = \min\{p, r\} \quad (2.18)$$

where C is a constant independent of u and h .

A brief analysis of (2.18) shows that the minimum convergence rate expected in the L^2 -norm for a degree p finite element interpolation and for a “sufficiently regular” solution is h^{p+1} . So, for linear elements ($p = 1$), we expect a convergence rate of order h^2 .

To make programming even simpler, the L^2 -norm error (normalized) is computed as:

$$E = \left[\sum_{a=1}^{n_{pts}} (U^a - u(\mathbf{x}^a))^2 \right]^{1/2} \left[\sum_{a=1}^{n_{pts}} u(\mathbf{x}^a)^2 \right]^{-1/2} \quad (2.19)$$

where $u(\mathbf{x}^a)$ is the exact solution at \mathbf{x}^a .

In order to create the convergence plots of the proposed nodal approximations, an specific problem with known solution is needed. We have selected a two dimensional steady-state heat conduction problem. We take Ω as the unit square ($[0,1] \times [0,1]$) and the force term in a way such that $u = x^2 y^2 (x-1)^2 (y-1)^2$ is solution of equation (2.1) taking $\nu = 1 + x + y + x^2 + y^2$

We have used four finite element meshes with sizes $h = 0.2, 0.10, 0.05$ and 0.01

In table 2.1 we can see the values obtained and in figure 2.1 we can see the results plotted.

We have denoted as “S” when no consistency correction is used, “C” when the correction proposed in (2.15) is used; and, we use “a”, “b” or “(a+b)/2” to indicate where the diffusion coefficient ν is evaluated.

h	S-a	S-b,C-b	S-(a+b)/2	C-a	C-(a+b)/2
0.200	1.2950	0.2436	0.2609	1.0018	0.2090
0.100	0.1349	0.1005	0.0949	0.0733	0.0522
0.050	0.1050	0.0790	0.0559	0.0719	0.0116
0.010	0.1004	0.0778	0.0414	0.0780	0.0004

Table 2.1: Diffusion Convergence Test

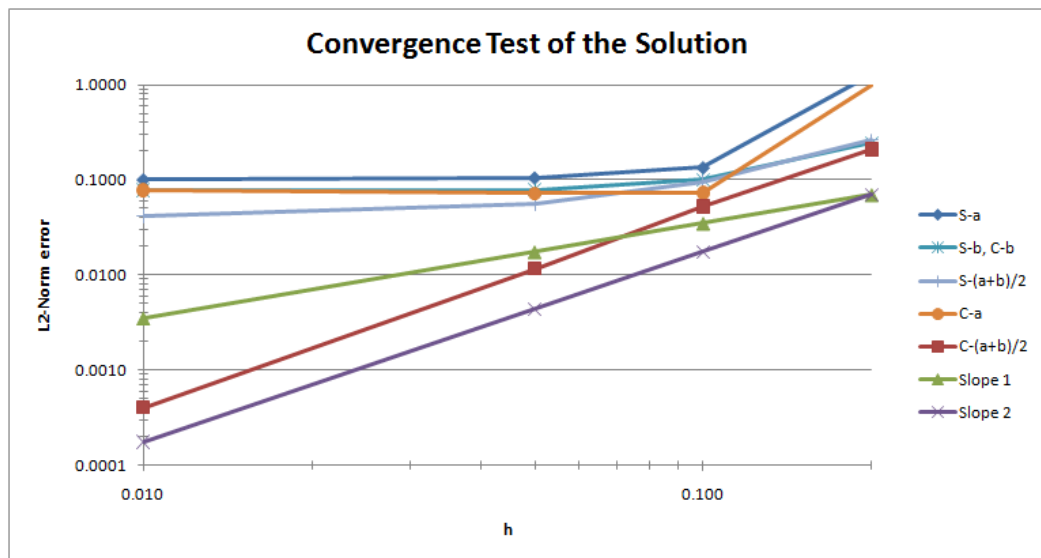


Figure 2.1: Diffusion Equation Convergence Test

2. DIFFUSION EQUATION

As can be seen in figure 2.1, just when diffusion is evaluated in “(a+b)/2” and corrected produces good results in terms of convergence. The other approximations do not work well.

Just to conclude, let us write the final form of the approximation of the diffusive term proposed:

$$L^{ba} = \frac{\nu^b + \nu^a}{2} \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad \forall a \neq b \quad (2.20a)$$

and

$$L^{bb} = - \sum_{a \neq b}^{n_{pts}} L^{ba} \quad (2.20b)$$

Notice that, for a given node b , the evaluation of L^{ba} does not require to loop over all “a” nodes, it just requires to loop over the “a” nodes connected to node b .

Chapter 3

Convection-Diffusion Equation

3.1 Problem Statement

In this chapter we are going to consider the steady-state convection-diffusion equation. From hereafter during this chapter, this equation will serve to model heat transfer by convection and conduction in an incompressible fluid media, consequently the terminology used for unknowns and boundary conditions will have relation with heat transfer.

The equation that describes the problem is:

$$\nabla \cdot (\rho_0 c_0 u \mathbf{a}) + \nabla \cdot (-\nu \nabla u) = f \quad (3.1)$$

with the following boundary conditions:

$$\begin{aligned} u(\mathbf{x}) &= \bar{u}(\mathbf{x}) & \text{on } \Gamma_D \\ \nu \nabla u \cdot \mathbf{n} &= \bar{h}(\mathbf{x}) & \text{on } \Gamma_N \\ \Gamma_D \cup \Gamma_N &= \partial\Omega \\ \Gamma_D \cap \Gamma_N &= \emptyset \end{aligned} \quad (3.2)$$

to be solved over $\Omega \subset \mathbb{R}^{n_{sd}}$, where n_{sd} is the space dimension, u (temperature) is the unknown, f is the forcing term (heat source per unit volume), ν is the diffusion coefficient (thermal conduction coefficient), \mathbf{a} is the convection velocity, ρ_0 is the mass density (constant for our analysis), c_0 is the specific heat capacity (constant for our analysis), \bar{u} is a prescribed value of the solution on the boundary, \bar{h} is a prescribed conductive heat flow per unit area and \mathbf{n} is the outward-pointing normal to the boundary.

3. CONVECTION-DIFFUSION EQUATION

The prescribed heat flow, \bar{h} , is taken as positive when heat goes into the domain through the boundary. Negative otherwise.

The diffusion coefficient ν , the mass density ρ_0 and the specific heat capacity c_0 are considered always greater than zero. The forcing term f is taken as positive when heat is added to the domain and negative when heat is extracted from the domain.

In the general case, the diffusion coefficient is not constant and can depend on space coordinates (i.e. a domain composed by different materials) or on the solution itself (i.e. $\nu(\mathbf{x}, u)$). In the present work constant, space dependant and solution dependant diffusion coefficient has been considered to validate de formulation.

The specific heat capacity c_0 of a material refers to the energy needed to raise the temperature of a unit mass of the material in one temperature unit. In the general case, heat capacity can vary depending on material or on the unknown u itself, but for simplicity, we have considered it constant for our analysis. Specific heat capacity can be referred as c_p (heat capacity at constant pressure) or c_v (heat capacity at constant volume). In this case, as the flow is incompressible and the mesh is fixed, both heat capacities are equal and that is why we refer to them as just specific heat capacity.

As we have mentioned before, we are modelling an incompressible fluid, then the following condition holds:

$$\nabla \cdot \mathbf{a}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \Omega \quad (3.3)$$

3.1.1 Properties of the Convective Term

In chapter 2 we described the properties of the diffusive term. Now we are going just to explain the properties of the convective term.

Extending the concepts from chapter 2, the convection process has two basic properties: Conservation, Invariance to translations and Non-Invariance to reflections.

Conservation can be explained examining equation (3.1), taking $\nu = 0$ and integrating in an arbitrary domain Ω we have:

$$\int_{\Omega} \nabla \cdot (\rho_0 c_0 u \mathbf{a}) \, d\Omega = \int_{\Omega} f \, d\Omega \quad (3.4)$$

Using the divergence theorem, which relates the volume integral with a surface

integral we can re-write the first term of equation (3.4) as:

$$\int_{\partial\Omega} (\rho_0 c_0 u \mathbf{a}) \cdot \mathbf{n} \, d\Gamma = \int_{\Omega} f \, d\Omega \quad (3.5)$$

Equation (3.5) expresses that the net heat flow ($\int_{\partial\Omega} (\rho_0 c_0 u \mathbf{a}) \cdot \mathbf{n} \, d\Gamma$) across the boundary of the domain is equal to the heat generated into the domain. This explains the (energy) conservation property of the convection term.

Invariance under translations can be explained using a change of variable. First, let us suppose $u(x)$ is the solution of a particular steady-state pure convection problem. Then, let us take $\bar{x} = x + \Delta x_0$ as a new reference system. Then $u(\bar{x})$ is still solution of equation (3.1) because it involves first derivatives, so using the chain rule for derivation the constant value Δx_0 cancels out.

Non-Invariance under reflections can be explained using a change of variable. First, let us suppose $u(x)$ is the solution of a particular steady-state pure convection problem. Then, let us take $\bar{x} = -x$ as a new reference system. Then $u(\bar{x})$ is not solution of equation (3.1) because it involves first derivatives, so using the chain rule for derivation the minus sign of the new reference system does not cancel out. This non-invariance to reflections of the convective term reflects in the skew-symmetry of it.

3.1.2 Weak Form of the problem

Let us define the following functional spaces:

$$V = \{u(\mathbf{x}) \in H^1(\Omega) \mid u = \bar{u} \text{ on } \Gamma_D\}$$

$$V_0 = \{u(\mathbf{x}) \in H^1(\Omega) \mid u = 0 \text{ on } \Gamma_D\}$$

After defining those spaces, the weak form of the problem reads as follows: Find $u \in V$ such that:

$$\rho_0 c_0 \int_{\Omega} v (\mathbf{a} \cdot \nabla u) \, d\Omega + \int_{\Omega} \nabla v \cdot \nu \nabla u \, d\Omega = \int_{\Omega} v f \, d\Omega + \int_{\Gamma_N} v \bar{h} \, d\Gamma \quad (3.6)$$

for all test functions $v \in V_0$

3.2 Numerical Approximation

Now we can approximate the weak form of the problem defined in (3.6) using finite elements.

3. CONVECTION-DIFFUSION EQUATION

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} . Let u_h be the finite element approximation of the unknown u ; and, v_h the test functions associated with $\{\Omega^e\}$.

Hence, the finite element problem reads: Find $u_h \in V_h$ such that:

$$\rho_0 c_0 \int_{\Omega} v_h (\mathbf{a} \cdot \nabla u_h) \, d\Omega + \int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega = \int_{\Omega} v_h f \, d\Omega + \int_{\Gamma_N} v_h \bar{h} \, d\Gamma \quad (3.7)$$

for all test functions $v_h \in V_{0,h}$

To obtain the algebraic version of the problem we need some definitions. Let n_{pts} be the total number of nodes of the finite element mesh and let N^a be the standard finite element interpolation function associated to node a , where $a = 1, \dots, n_{pts}$.

The unknown is interpolated as:

$$u_h = \sum_{a=1}^{n_{pts}} N^a U^a \quad (3.8)$$

We use upper-case letters (i.e. U^a) to denote the nodal value at node a of the corresponding lower-case variable.

The test functions are taken as:

$$v_h = N^b, \quad b = 1, \dots, n_{pts} \quad (3.9)$$

One inconvenient that may arise when solving equation (3.7) is the appearance of numerical instabilities. This happens when convection dominates diffusion. The stability in 1D of the Galerkin approximation depends on the Peclet number defined as:

$$Pe = \frac{\rho_0 c_0 |\mathbf{a}| h}{2\nu} \quad (3.10)$$

where h is the mesh size.

In figure 3.1 we can see how the Peclet number influences the numerical instabilities that might appear in a 1D convection-diffusion problem:

According to [6], if $Pe > 1$, numerical instabilities may appear. So we have two options: use a finer mesh (reduce h) or use a stabilization procedure. Here we are going to follow the latter procedure.

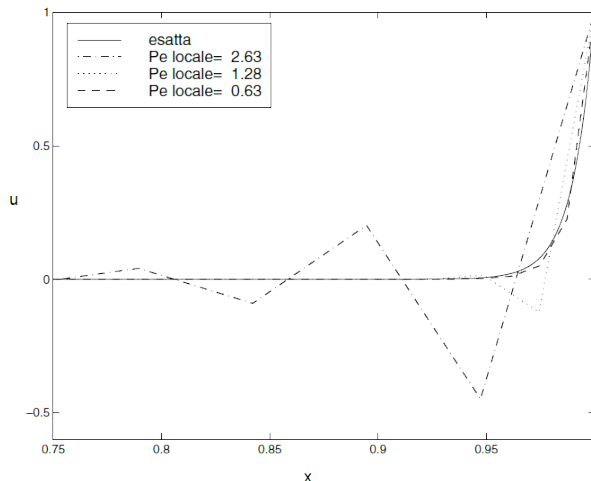


Figure 3.1: Influence of Local Peclet number on numerical instabilities , according to [7]

According to [1], various stabilization methods have been developed over time, such as Streamline Upwind Petrov-Galerkin, Galerkin Least Squares, Subgrid Scale Stabilization, Characteristic Galerkin and Taylor Galerkin.

We are going to use the Subgrid Scale Stabilization method. The reason of that choice is because one of the aims of the edge-based conservative schemes is to be applicable to Navier-Stokes Equations, and for Navier-Stokes we are going to work with an equal velocity-pressure interpolation stabilized using the subgrid scale stabilization method.

Taking:

$$V_h = \{u_h(\mathbf{x}) \in H^1(\Omega) \mid u_h = \bar{u} \text{ on } \Gamma_D\}$$

and

$$V_{0,h} = \{u_h(\mathbf{x}) \in H^1(\Omega) \mid u_h = 0 \text{ on } \Gamma_D\}$$

The weak form of the stabilized problem reads as follows: Find $u \in V$ such that:

$$\begin{aligned} \rho_0 c_0 \int_{\Omega} v_h (\mathbf{a} \cdot \nabla u_h) \, d\Omega + \int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega + \int_{\Omega} \mathcal{P}(v_h) \tau \mathcal{R}(u_h) \, d\Omega \\ = \int_{\Omega} v_h f \, d\Omega + \int_{\Gamma_N} v_h \bar{h} \, d\Gamma \end{aligned} \quad (3.11)$$

for all test functions $v \in V_0$.

3. CONVECTION-DIFFUSION EQUATION

Where $\mathcal{R}(u)$ is the residual of the equation defined as:

$$\mathcal{R}(u) = \mathcal{L}(u) - f = \nabla \cdot (\rho_0 c_0 u \mathbf{a}) + \nabla \cdot (-\nu \nabla u) - f \quad (3.12)$$

$\mathcal{P}(u)$ is the stabilization operator defined as:

$$\mathcal{P}(u) = \nabla \cdot (\rho_0 c_0 u \mathbf{a}) + \nabla \cdot (\nu \nabla u) \quad (3.13)$$

and τ is the stabilization parameter.

As we are just using linear elements ($p = 1$), second derivatives are zero. Then, the final equation becomes:

$$\begin{aligned} \rho_0 c_0 \int_{\Omega} v_h (\mathbf{a} \cdot \nabla u_h) \, d\Omega + \int_{\Omega} \nabla v_h \cdot \nu \nabla u_h \, d\Omega + \rho_0^2 c_0^2 \int_{\Omega} \mathbf{a} \cdot \nabla v_h \tau \mathbf{a} \cdot \nabla u_h \, d\Omega \\ = \int_{\Omega} v_h f \, d\Omega + \int_{\Gamma_N} v_h \bar{h} \, d\Gamma + \rho_0 c_0 \int_{\Omega} \mathbf{a} \cdot \nabla v_h \tau f \, d\Omega \end{aligned} \quad (3.14)$$

The stabilization parameter is taken as variable from point to point and is calculated locally as:

$$\tau = \left[\frac{4\nu}{h^2} + \frac{2\rho_0 c_0 |\mathbf{a}|}{h} \right]^{-1} \quad (3.15)$$

where h is the element size (for linear elements) when the mesh is uniform. If the mesh is non-uniform, h is the maximum distance from the point into consideration to its neighbours.

3.3 Node-Based Implementation

In this section, we are going to express the approximate problem in terms of solely geometrical dependant integrals:

$$\begin{aligned} \int_{\Omega} N^b N^a \, d\Omega \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega, \quad i, j = 1, \dots, n_{sd} \\ \int_{\Omega} N^b \partial_j N^a \, d\Omega, \quad j = 1, \dots, n_{sd} \\ \int_{\Omega} \partial_j N^b N^a \, d\Omega, \quad j = 1, \dots, n_{sd} \end{aligned} \quad (3.16)$$

for $a, b = 1, \dots, n_{pts}$

With the integrals in (3.16) and taking $v_h = N^b$, we can approximate each part of equation (3.14) in various forms.

Notice that we have not written integrals with derivatives higher than first order because just linear elements are analyzed.

To tackle the problem, we are going to divide the problem into a convective part, diffusive part and stabilization part. The aim will be to obtain symmetry in the diffusive and stabilization parts.

Let us consider the matrix form of the problem, the system of equations to solve will be:

$$\underline{\underline{C}}\underline{U} + \underline{L}\underline{U} + \underline{S_U}\underline{U} = \underline{\underline{M}}\underline{F} + \underline{B} + \underline{S_F}\underline{F} \quad (3.17)$$

where double underline means a matrix and single underline means a column vector. $\underline{\underline{C}}$ is the convection matrix, \underline{L} diffusion matrix and $\underline{S_U}$ the stabilization matrix operating over \underline{U} , $\underline{\underline{M}}\underline{F}$ is the forcing term, \underline{B} is the contribution from the Neumann boundary conditions to the right hand side and $\underline{S_F}\underline{F}$ is the stabilization contribution to the forcing vector.

We need to find expressions for the convection, diffusion and stabilization matrices.

The diffusion term is approximated according to the results obtained from chapter 2 as follows:

$$L^{ba} = \frac{\nu^b + \nu^a}{2} \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad \forall a \neq b \quad (3.18a)$$

and

$$L^{bb} = - \sum_{a \neq b}^{n_{pts}} L^{ba} \quad (3.18b)$$

We have proven in chapter 2 that this evaluation of the diffusive term is consistent and conservative, so we do not worry any more about that and centre our efforts in the convective and stabilization terms.

3. CONVECTION-DIFFUSION EQUATION

The convective term can be approximated as any of the following expressions:

$$C^{ba} = \sum_{j=1}^{n_{sd}} \rho_0 c_0 A_j^a \int_{\Omega} N^b \partial_j N^a \, d\Omega \quad (3.19a)$$

$$C^{ba} = \sum_{j=1}^{n_{sd}} \rho_0 c_0 A_j^b \int_{\Omega} N^b \partial_j N^a \, d\Omega \quad (3.19b)$$

$$C^{ba} = \sum_{j=1}^{n_{sd}} \rho_0 c_0 \frac{A_j^a + A_j^b}{2} \int_{\Omega} N^b \partial_j N^a \, d\Omega \quad (3.19c)$$

The stabilization term can be approximated as any of the following expressions:

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^a & A_i^b & A_j^a \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20a)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^a & A_i^a & A_j^b \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20b)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^a & \left(\frac{A_i^a + A_i^b}{2} \right) & \left(\frac{A_j^a + A_j^b}{2} \right) \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20c)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^b & A_i^b & A_j^a \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20d)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^b & A_i^a & A_j^b \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20e)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^b & \left(\frac{A_i^a + A_i^b}{2} \right) & \left(\frac{A_j^a + A_j^b}{2} \right) \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20f)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \frac{\tau^a + \tau^b}{2} & A_i^b & A_j^a \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20g)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \frac{\tau^a + \tau^b}{2} & A_i^a & A_j^b \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20h)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\begin{array}{ccc} \frac{\tau^a + \tau^b}{2} & \left(\frac{A_i^a + A_i^b}{2} \right) & \left(\frac{A_j^a + A_j^b}{2} \right) \\ \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \end{array} \right] \quad (3.20i)$$

The stabilization contribution to the force term can be approximated as:

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^a & A_i^b & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21a)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^a & A_i^a & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21b)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^a & \frac{A_i^a + A_i^b}{2} & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21c)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^b & A_i^b & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21d)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^b & A_i^a & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21e)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \tau^b & \frac{A_i^a + A_i^b}{2} & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21f)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \frac{\tau^a + \tau^b}{2} & A_i^b & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21g)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \frac{\tau^a + \tau^b}{2} & A_i^a & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21h)$$

$$S_F^{ba} = \rho_0 c_0 \sum_a^{n_{pts}} \sum_{i=1}^{n_{sd}} \left[\begin{array}{ccc} \frac{\tau^a + \tau^b}{2} & \frac{A_i^a + A_i^b}{2} & \int_{\Omega} \partial_i N^b N^a \, d\Omega \end{array} \right] \quad (3.21i)$$

3.3.1 Consistency

Let us consider problem (3.14) without forcing term. Additionally, let us consider all the boundary conditions are of Dirichlet type: $\bar{u} = 1$. Therefore, $\Gamma_N = \emptyset \wedge \Gamma_D = \partial\Omega$. In this situation the solution is constant and equal to 1 in all nodal points. Additionally, recall that the convection velocity is divergence free ($\nabla \cdot \mathbf{a}$).

This behaviour has to be reproduced exactly by the finite element solution because, even with linear interpolation functions, the exact solution belongs to the finite element

3. CONVECTION-DIFFUSION EQUATION

space. Consequently, the following conditions have to hold:

$$\sum_a^{n_{pts}} C^{ba} = 0 \quad \forall b = 1, \dots, n_{pts} \quad (3.22a)$$

$$\sum_a^{n_{pts}} L^{ba} = 0 \quad \forall b = 1, \dots, n_{pts} \quad (3.22b)$$

$$\sum_a^{n_{pts}} S_U^{ba} = 0 \quad \forall b = 1, \dots, n_{pts} \quad (3.22c)$$

It can be readily observed that all the evaluation methods of the convective term proposed in (3.19) satisfy this condition. Approximation (3.19a) satisfies this condition because the sum over points a over spatial dimension j of $A_j^a \partial_j N^a$ is $\nabla \cdot \mathbf{a}$ which is zero. Approximation (3.19b) is consistent because we are taking the derivative $\partial_j \sum_a N^a$ which is always zero. Finally, approximation (3.19c) is consistent because is a linear combination of the previous two.

Additionally, for the stabilization term S_U just (3.20d) satisfies the consistency condition. So here we propose a correction method to solve this issue: evaluating the main diagonal term of the stabilization matrix as the negative of the sum of the off-diagonal terms, so the final form of the matrix S_U will be any in (3.20) for the off-diagonal terms and the diagonal terms will be calculated as:

$$S_U^{bb} = - \sum_{a \neq b}^{n_{pts}} S_U^{ba} \quad (3.23)$$

Finally, for the stabilization contribution to the right hand side S_F , as the forcing term is zero, we can not say anything about it, so it keeps for now as defined in (3.21).

With the definitions provided in equations (3.19), (3.20) and (3.21) in combination with (3.23) the consistency condition (3.22) is satisfied by any of the evaluation methods proposed.

3.3.2 Conservation

For simplicity, let us consider the problem (3.14) with all boundary conditions of Neumann type (case in which the solution would not be unique). Furthermore, suppose we

take the test function v_h as constant. Hence, problem (3.14) imply:

$$\int_{\partial\Omega} (\rho_0 c_0 u \mathbf{a} \cdot \mathbf{n} - \bar{h}) \, d\Gamma = \int_{\Omega} f \, d\Omega \quad (3.24)$$

which, in physical terms, represents that the net heat flow that goes out of the domain through the boundary (by means of conduction and mass transport) is equal to the heat added/generated inside the domain. This is physically consistent.

However, equation (3.14) is not enforced for constant test functions v_h but only for test functions of the form $v_h = N^b$, $b = 1, \dots, n_{\text{poin}}$. Since the addition of all shape functions N^b is 1, equation (3.24) can also be obtained taking $v_h = N^b$ in equation (3.14) and adding up over b .

Then, in matrix form, condition (3.24) can be expressed as:

$$\sum_b \sum_a (C^{ba} + L^{ba}) U^a = \int_{\Omega} f \, d\Omega + \int_{\partial\Omega} \bar{h} \, d\Gamma \quad (3.25a)$$

$$\sum_b \sum_a S_U^{ba} U^a = \sum_b \sum_a S_F^{ba} F^a \quad (3.25b)$$

in order to have global conservation.

To be able to see clearly the conservation, we can rewrite the convective term by means of integration by parts:

$$\begin{aligned} \int_{\Omega} v_h \nabla \cdot (\rho_0 c_0 \mathbf{a} u_h) \, d\Omega &= \\ &= -\rho_0 c_0 \int_{\Omega} (\mathbf{a} \cdot \nabla v_h) u_h \, d\Omega + \int_{\partial\Omega} v_h \rho_0 c_0 u_h \mathbf{a} \cdot \mathbf{n} \, d\Gamma \end{aligned} \quad (3.26)$$

With equation (3.26) inserted in (3.14) we recover the global conservation statement presented in (3.24).

In a similar fashion, we can rewrite the approximations of the convective term

3. CONVECTION-DIFFUSION EQUATION

proposed in (3.19):

$$C^{ba} = \rho_0 c_0 \sum_{j=1}^{n_{sd}} \left[-A_j^a \int_{\Omega} (\partial_j N^b) N^a d\Omega + \int_{\partial\Omega} N^b N^a A_j^a n_j^a d\Gamma \right] \quad (3.27a)$$

$$C^{ba} = \rho_0 c_0 \sum_{j=1}^{n_{sd}} \left[-A_j^b \int_{\Omega} (\partial_j N^b) N^a d\Omega + \int_{\partial\Omega} N^b N^a A_j^b n_j^b d\Gamma \right] \quad (3.27b)$$

$$C^{ba} = \rho_0 c_0 \sum_{j=1}^{n_{sd}} \left[-\frac{A_j^a + A_j^b}{2} \int_{\Omega} (\partial_j N^b) N^a d\Omega + \int_{\partial\Omega} N^b N^a \frac{A_j^a + A_j^b}{2} \frac{n_j^a + n_j^b}{2} d\Gamma \right] \quad (3.27c)$$

As the approximations given in (3.27) are exactly equivalent to those given in (3.19), we can see that the condition needed to fulfil (3.25), given that $\sum_a L^{ba} = 0$ and $L^{ba} = L^{ab}$, is:

$$\sum_b \sum_a C^{ba} U^a = \int_{\partial\Omega} \rho_0 c_0 u_h \mathbf{a} \cdot \mathbf{n} d\Gamma \quad (3.28a)$$

$$\sum_b \sum_a S_U^{ba} U^a = \sum_b \sum_a S_F^{ba} F^a \quad (3.28b)$$

It can be clearly seen that condition (3.28a) is fulfilled by only the first form of the convective term (3.27a). In practice what we are going to use is (3.19a) because is equivalent and shorter to write and code.

For the stabilization part we can follow two paths to fulfil conservation. The first path is less restrictive but more difficult to attain; it consists in comply directly with condition (3.28b) (and without forgetting about the consistency condition). The second path, more restrictive but easier to attain, consists in splitting condition (3.28b) into two:

$$\sum_b \sum_a S_U^{ba} U^a = 0 \quad (3.29a)$$

$$\sum_b \sum_a S_F^{ba} F^a = 0 \quad (3.29b)$$

The previous conditions can be rewritten as:

$$\sum_b S_U^{ba} = 0 \quad (3.30a)$$

$$\sum_b S_F^{ba} = 0 \quad (3.30b)$$

Notice that element-based finite element approximations comply with this conditions too, so we are not making our conservative conditions over-restrictive.

From the approximations for S_U with the respective correction for consistency, the only ones that satisfy the requirement (3.30a) are (3.20g) - (3.20i). But to keep analogies with the approximation of the convective term chosen (3.19a), we choose (3.20g).

Finally, for S_F with the stabilization parameter evaluated as the average of nodal points a and b (to keep the analogy with the approximation of S_U chosen), condition (3.30b) is not satisfied, so here we propose yet another correction method: evaluating S_F as (3.21g) for the off diagonal terms and the diagonal as the negative of the sum of the off-diagonal terms:

$$S_F^{aa} = - \sum_{b \neq a} S_F^{ba} \quad \forall a = 1, \dots, n_{pts} \quad (3.31)$$

Notice that the imposition of (3.30a) with the consistency condition produces a symmetric matrix contributing to the non-stabilized main matrix.

Anyway, in the following section, we will test the convergence features of various formulations mentioned in order to determine their convergence properties.

3.4 Convergence Test

Reference [5] shows an a priori error estimator for the stabilized convection-diffusion-reaction equation. In order to calculate that error, we need to define the three bar norm as:

$$||| u |||^2 := \| \nu^{1/2} \nabla u \|_{L_2}^2 + \| \tau^{1/2} \mathbf{a} \cdot \nabla u \|_{L_2}^2 \quad (3.32)$$

We have introduced inside the norm evaluation the values of ν and τ because they are taken as variable inside the domain.

3. CONVECTION-DIFFUSION EQUATION

The error in the three bar norm is defined as:

$$e(h)^2 := ||| u - u_h |||^2 \quad (3.33)$$

The error estimator takes the following form:

$$e(h)^2 \leq \left(\frac{\nu}{h^2} + \frac{|\mathbf{a}|}{h} \right) h^{2(p+1)} |u|_{H^{p+1}}^2 \quad (3.34)$$

where h is the mesh size and p is the order of the polynomial used in the finite element interpolation.

As we are interested in convection dominated situations, ν is very small compared with $|\mathbf{a}|$, then the error will be of order $h^{p+1/2}$.

The error in our code is computed as:

$$E = \left[\sum_{a=1}^{n_{pts}} (||| U^a - u(\mathbf{x}^a) |||^2) \right]^{1/2} \left[\sum_{a=1}^{n_{pts}} \left(\hat{\tau} \hat{L}^{2(p+1)} |u(\mathbf{x}^a)|_{H^{p+1}}^2 \right) \right]^{-1/2} \quad (3.35)$$

where $u(\mathbf{x}^a)$ is the exact solution at \mathbf{x}^a , \hat{L} is the diameter of the computational domain Ω and $\hat{\tau}$ is a normalization parameter defined as:

$$\hat{\tau} := \frac{\max\{\nu\}}{\hat{L}^2} + \frac{\max\{|\mathbf{a}|\}}{\hat{L}} \quad (3.36)$$

In order to create the convergence plots of the proposed nodal approximations, an specific problem with known solution is needed. We have selected a two dimensional steady-state problem. We take Ω as the unit square ($[0,1] \times [0,1]$) and the force term in a way such that $u = x^2 y^2 (x-1)^2 (y-1)^2$ is solution of equation (3.1) taking $\nu = 0.0001$, $\rho_0 = 1$, $c_0 = 1$, $\mathbf{a} = (fg', -f'g)$, $f(x) = x^2$ and $g(y) = y^2$.

We have used four finite element meshes discretized with triangles with uniform sizes of $h = 0.2, 0.10, 0.05$ and 0.01 . In figure 3.2 you can see the meshes used.

We can combine all the approximations with each other and we will end up doing 108 convergence tests. To simplify the process we will just present convergence tests for those formulations that are conservative (and consistent or not consistent). But, to have a reference start point, we will present a convergence test for a consistent formulation that has been used with success many times within the code. This formulation corresponds evaluating the convection velocity in the convective term as A_j^a ;

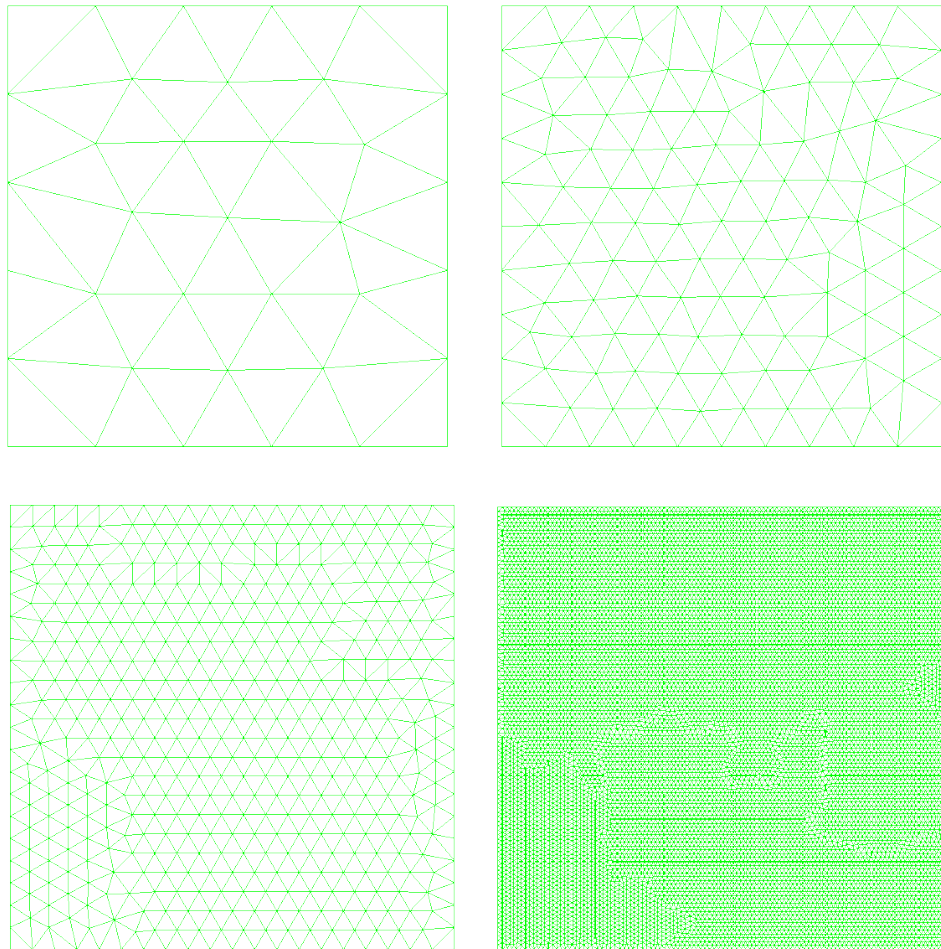


Figure 3.2: Triangle meshes for the convergence test

3. CONVECTION-DIFFUSION EQUATION

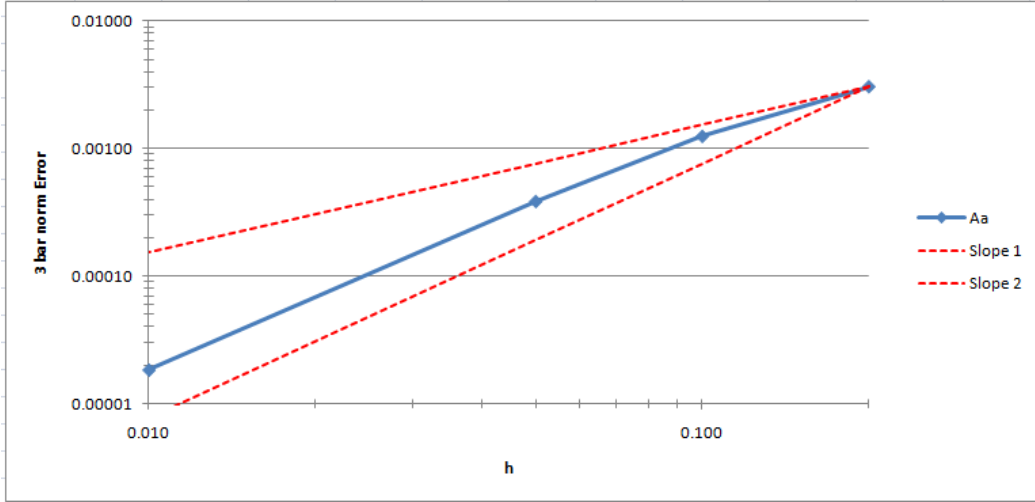


Figure 3.3: Convection Diffusion Equation Reference Convergence Test

an analogous evaluation of the convective velocity in the stabilization terms; and, the stabilization parameter as τ^b in the left and right hand sides.

The results are presented in the following order: first we present the convergence test with the left hand side stabilization parameter τ evaluated at nodal point a without consistency correction. Then with τ evaluated as the average of the values at nodal points a and b with consistency correction. We will denote them as “LHS S Tau a ” and “LHS C Tau ab ”.

The previous left hand side evaluations will be combined with different ways of approximating the convection velocity. First, convection velocity is evaluated at nodal point a , then at nodal point b and finally as the average between a and b . We denote them as “Aa”, “Ab” and “Aab”. Notice that, with the corresponding analogies, the same evaluation of the convection velocity is used for the stabilization terms.

Finally, the force stabilization term is evaluated with τ at nodal point a without conservation correction, then at point a with conservation correction, then at point b with conservation correction and finally as the average between a and b with conservation correction. We thenote them as “RHS S Tau a ”, “RHS C Tau a ”, “RHS C Tau b ” and “RHS C Tau ab ”.

So, in total we will have 6 plots: the combination of the evaluation of τ in the left hand side and the evaluation of the convection velocity. In each of the plots, there will

3.4 Convergence Test

be four curves corresponding to the evaluation of the right hand side stabilization term.

In figures 3.4 through 3.9 we can see the results obtained.

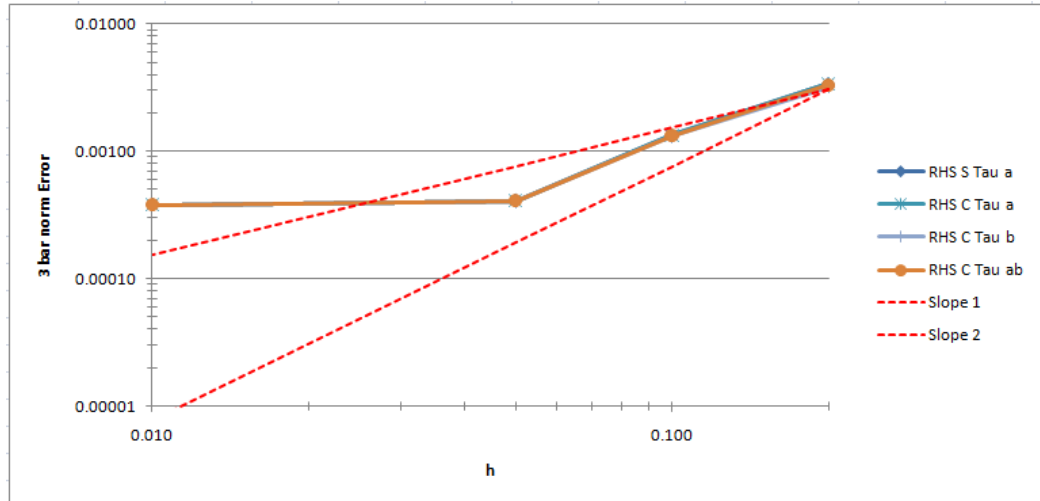


Figure 3.4: Convection Diffusion Equation Convergence Test - LHS S Tau a - Aa

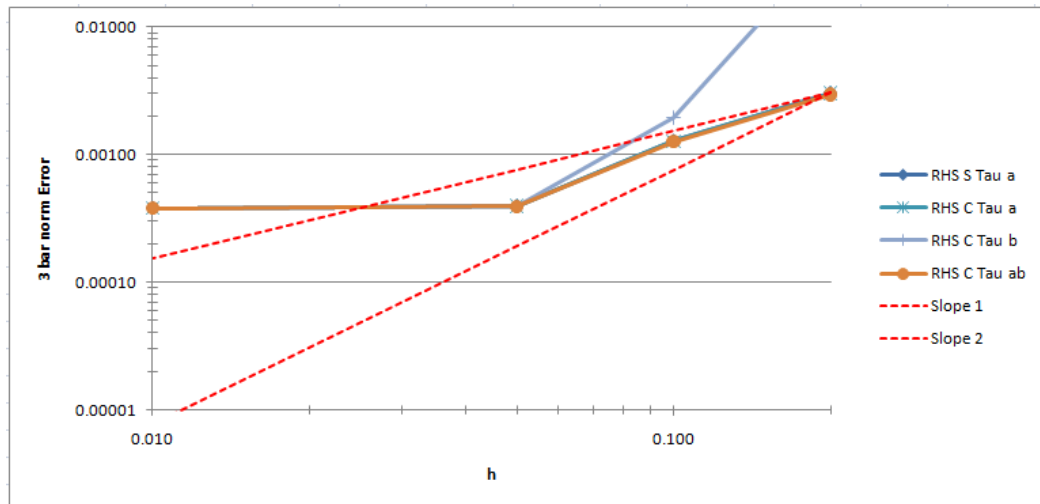


Figure 3.5: Convection Diffusion Equation Convergence Test - LHS S Tau a - Ab

As can be seen in figures 3.4 through 3.9, the evaluation of the left hand side stabilization parameter at node a without correction (LHS S Tau a) does not work well. Evaluating the left hand side stabilization parameter as the average of the values at nodes a and b with consistency correction (LHS C Tau ab) works very well.

3. CONVECTION-DIFFUSION EQUATION

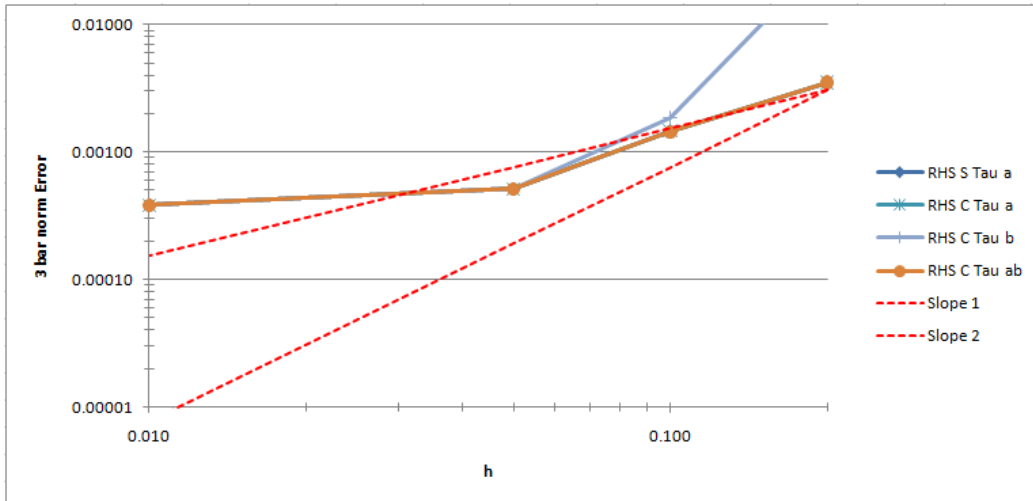


Figure 3.6: Convection Diffusion Equation Convergence Test - LHS S Tau a - Aa

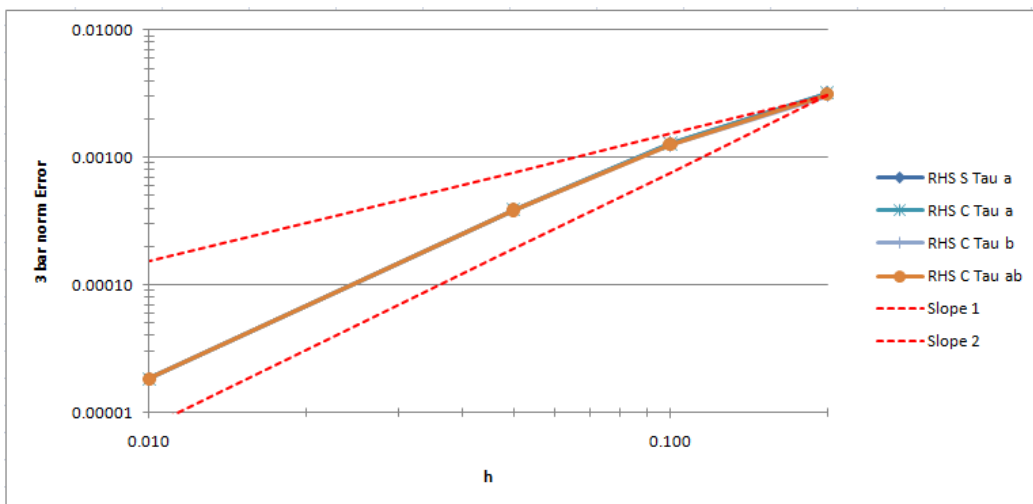


Figure 3.7: Convection Diffusion Equation Convergence Test - LHS C Tau ab - Aa

3.4 Convergence Test

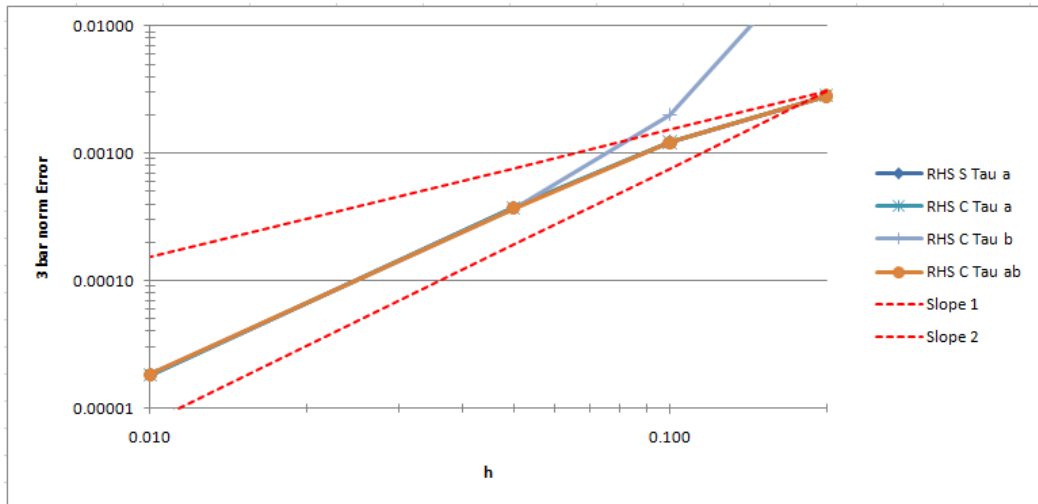


Figure 3.8: Convection Diffusion Equation Convergence Test - LHS C Tau ab - Ab

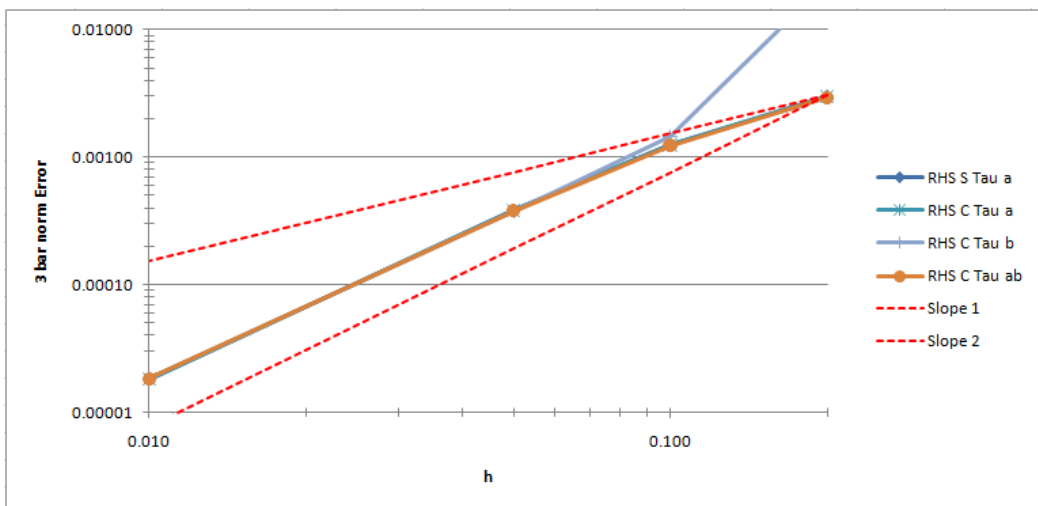


Figure 3.9: Convection Diffusion Equation Convergence Test - LHS C Tau ab - Aab

3. CONVECTION-DIFFUSION EQUATION

Comparing figures 3.7 through 3.9, we can see that the evaluation of the convection velocity as Aa works better than the other evaluations. We prefer this evaluation because of those convergence results and because it represents the interpolation of the product.

Now, we are in figure 3.7. All the evaluations of the right hand side stabilization parameter work pretty much the same. For uniformity, we choose RHS C Tau ab because it uses the same stabilization parameter as the left hand side.

Let us write the final form of the consistent and conservative approximation proposed:

$$C^{ba} = \rho_0 c_0 \sum_{j=1}^{n_{sd}} A_j^a \int_{\Omega} N^b \partial_j N^a \, d\Omega \quad (3.37a)$$

$$L^{ba} = \frac{\nu^b + \nu^a}{2} \sum_{j=1}^{n_{sd}} \int_{\Omega} \partial_j N^b \partial_j N^a \, d\Omega \quad \forall a \neq b \quad (3.37b)$$

$$L^{bb} = - \sum_{a \neq b}^{n_{pts}} L^{ba} \quad (3.37c)$$

$$S_U^{ba} = \rho_0^2 c_0^2 \sum_{i,j=1}^{n_{sd}} \left[\frac{\tau^a + \tau^b}{2} A_i^b A_j^a \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \right] \quad \forall a \neq b \quad (3.37d)$$

$$S_U^{bb} = - \sum_{a \neq b}^{n_{pts}} S_U^{ba} \quad (3.37e)$$

$$M^{ba} F^a = \sum_a \int_{\Omega} N^b N^a \, d\Omega F^a \quad (3.37f)$$

$$B^b = \sum_a \int_{\Gamma_N} N^b N^a \, d\Gamma H^a \quad (3.37g)$$

$$S_F^{ba} = \rho_0 c_0 \sum_{i=1}^{n_{sd}} \left[\frac{\tau^a + \tau^b}{2} A_i^b \int_{\Omega} \partial_i N^b N^a \, d\Omega \right] \quad \forall b \neq a \quad (3.37h)$$

$$S_F^{aa} = - \sum_{b \neq a}^{n_{pts}} S_F^{ba} \quad (3.37i)$$

Notice that, for a given node b , the evaluation of the elements of a matrix (i.e. L^{ba}) does not require to loop over all a nodes, it just requires to loop over the a nodes connected to node b .

Let us remark the implications of the proposed edge-based approximation for the convective term.

The convective term $\rho_0 c_0 \nabla \cdot (u \mathbf{a})$ is written normally as $\rho_0 c_0 (\mathbf{a} \cdot \nabla)u$ because the convection velocity \mathbf{a} is divergence-free. The approximation of the convective term proposed was:

$$\rho_0 c_0 \int_{\Omega} v (\mathbf{a} \cdot \nabla u) \, d\Omega = \rho_0 c_0 \sum_{a=1}^{n_{pts}} \sum_{j=1}^{n_{sd}} A_j^a \int_{\Omega} N^b \partial_j N^a \, d\Omega U^a \quad (3.38)$$

It can be seen as the approximation of the product $u \mathbf{a}$. In other words, the approximation proposed resembles the approximation of $u \mathbf{a}$ instead of just u . This reasoning opens new opportunities for compressible flow with non-constant specific heat capacity. Hence, the general convective part can be written as:

$$\int_{\Omega} v \nabla \cdot (\rho c u \mathbf{a}) \, d\Omega = \sum_{a=1}^{n_{pts}} \sum_{j=1}^{n_{sd}} \rho^a c_0^a A_j^a \int_{\Omega} N^b \partial_j N^a \, d\Omega U^a \quad (3.39)$$

maintaining the consistency and conservation features of this approximation.

Just to finish, special attention should be put into the variation of a material property as function of another (i.e. temperature dependent diffusivity) because all the variations should be thermodynamically feasible. In other words, the relationships between one variable and another should not violate any of the laws of thermodynamic. This complicates the analysis, but restricts it at the same time, because we cannot assume (or require) that a property can variate arbitrarily when keeping other as constant.

3. CONVECTION-DIFFUSION EQUATION

Chapter 4

Navier-Stokes Equations

4.1 Problem Statement

In this chapter we are going to consider the steady-state Navier-Stokes equations. From hereafter during this chapter, this equation will serve to model laminar flow of an incompressible fluid coupled with thermal effects through the Boussinesq assumption, consequently the terminology used for unknowns and boundary conditions will have relation with fluid flow.

The equations that describe the problem are:

$$\mathbf{u} \cdot \nabla \mathbf{u} - 2\nabla \cdot [\nu \boldsymbol{\varepsilon}(\mathbf{u})] + \nabla p + \beta \mathbf{g} \vartheta = \mathbf{f} \quad (4.1a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (4.1b)$$

with the following boundary conditions:

$$\mathbf{u}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x}) \quad \text{on } \Gamma_D$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \bar{\mathbf{t}}(\mathbf{x}) \quad \text{on } \Gamma_N$$

$$\Gamma_D \cup \Gamma_N = \partial\Omega$$

$$\Gamma_D \cap \Gamma_N = \emptyset$$

(4.2)

to be solved over $\Omega \subset \mathbb{R}^{n_{sd}}$, where n_{sd} is the space dimension, \mathbf{u} is the fluid velocity, p is the kinematic pressure (pressure divided by the density), $\boldsymbol{\varepsilon}(\mathbf{u})$ is the symmetric part of the velocity gradient, \mathbf{f} is the forcing term (force per unit mass), ν is the kinematic viscosity, ρ_0 is the mass density (constant for our analysis), β is the volumetric thermal

4. NAVIER-STOKES EQUATIONS

expansion coefficient, ϑ is the temperature, $\bar{\mathbf{u}}$ is a prescribed value of the solution on the boundary, $\bar{\mathbf{t}}$ is a prescribed traction (force per unit area), $\boldsymbol{\sigma}$ is the Cauchy stress tensor divided by the density ($-p\mathbf{I} + 2\nu\boldsymbol{\varepsilon}$) and \mathbf{n} is the outward-pointing normal to the boundary.

The forcing term may contain gravity forces and the reference temperature for the buoyancy forces coming from the Boussinesq assumption:

$$\mathbf{f} = \mathbf{g} (1 + \beta \vartheta_0) \quad (4.3)$$

The thermal expansion coefficient β is taken as positive when the fluid expands due to an increment of temperature.

The kinematic viscosity ν and the mass density ρ_0 are considered always greater than zero.

In the general case, kinematic viscosity is not constant and can depend on space coordinates (i.e. a domain composed by different materials) or on the solution itself (i.e. $\nu(\mathbf{x}, u)$) In the present work constant and solution dependant diffusion coefficient has been considered to validate de formulation. In particular, the power-law rheological behaviour has been implemented in the code:

$$\nu = \rho_0^{-1} K_0 [4I_2(\boldsymbol{\varepsilon})]^{(n-1)/2} \quad (4.4)$$

where K_0 is the material consistency, n is the rate sensitivity and $I_2(\boldsymbol{\varepsilon})$ is the second invariant of the strain rate tensor defined as:

$$I_2(\boldsymbol{\varepsilon}) = \frac{1}{2} \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} \quad (4.5)$$

4.1.1 Properties of the Navier-Stokes Equation

In this section we highlight the conservation properties of the Navier-Stokes equations. These equations conserve mass and conserve linear momentum in all space directions.

Equation (4.1b) expresses mass conservation. It can be integrated over all the domain Ω , rewritten in the following way and converted using the divergence theorem:

$$\begin{aligned} \int_{\Omega} \nabla \cdot (\rho_0 \mathbf{u}) \, d\Omega &= \int_{\Omega} 0 \, d\Omega \\ \int_{\partial\Omega} \rho_0 \mathbf{u} \cdot \mathbf{n} \, d\Gamma &= 0 \end{aligned} \quad (4.6)$$

Equation (4.6) means that the mass flux across the boundary of the domain is null. This makes physical sense because there are not mass sources or sinks in the domain (this does not hold if there are nuclear reactions, but is not the case of our analysis).

Equation (4.1a) is a linear momentum conservation statement at the continuous level. In the same way as we did with the mass conservation equation, it can be integrated in the domain and we obtain:

$$\begin{aligned} \int_{\partial\Omega} (\rho_0 \mathbf{u})(\mathbf{u} \cdot \mathbf{n}) \, d\Gamma &= \int_{\Omega} \rho_0 \mathbf{g} \, d\Omega - \int_{\Omega} \rho_0 \nabla p \, d\Omega - \int_{\Omega} \rho_0 \beta \mathbf{g} (\vartheta - \vartheta_0) \, d\Omega \\ &+ \int_{\Omega} \rho_0 (2 \nabla \cdot (\nu \boldsymbol{\varepsilon}(\mathbf{u}))) \, d\Omega \end{aligned} \quad (4.7)$$

Equation (4.7) states that the net flow of linear momentum across the boundary is equal to the gravitational forces, pressure forces, buoyancy forces and viscous forces acting over each particle of the domain.

4.1.2 Weak Form of the problem

Let us define the following functional spaces:

$$\begin{aligned} V &= \{ \mathbf{u}(\mathbf{x}) \in H^1(\Omega)^{n_{sd}} \mid \mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_{dv} \} \\ V_0 &= \{ \mathbf{u}(\mathbf{x}) \in H^1(\Omega)^{n_{sd}} \mid \mathbf{u} = 0 \text{ on } \Gamma_{dv} \} \\ Q &= \{ q(\mathbf{x}) \in L^2(\Omega)^{n_{sd}} \mid \int_{\Omega} q \, d\Omega = 0 \text{ if } \Gamma_{nv} = \emptyset \} \end{aligned}$$

After defining those spaces, the weak form of the problem reads as follows: Find $\mathbf{u} \in V$, $\vartheta \in \Psi$ and $p \in Q$ such that:

$$\begin{aligned} \int_{\Omega} \mathbf{v} \cdot [\mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{g} \beta \vartheta] \, d\Omega + \int_{\Omega} 2 \nabla \mathbf{v} : \nu \boldsymbol{\varepsilon}(\mathbf{u}) \, d\Omega - \int_{\Omega} p \nabla \cdot \mathbf{v} \, d\Omega = \\ \int_{\Omega} \mathbf{v} \cdot \mathbf{f} \, d\Omega + \int_{\Gamma_{nv}} \mathbf{v} \cdot \bar{\mathbf{t}} \, d\Gamma \end{aligned} \quad (4.8a)$$

$$\int_{\Omega} q \nabla \cdot \mathbf{u} \, d\Omega = 0 \quad (4.8b)$$

for all test functions $\mathbf{v} \in V_0$ and $q \in Q$.

4. NAVIER-STOKES EQUATIONS

4.1.3 Linearization

Problem (4.1) has two sources of non-linearity: the convective term and the viscous term (viscosity depends on \mathbf{u} through (4.4)). To linearize the problem we use the Picard method, so the convective term is computed as:

$$\mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{u}^i \cdot \nabla \mathbf{u}^{i+1} \quad (4.9)$$

i is the iteration counter. \mathbf{u}^i means velocity evaluated with available velocity data and \mathbf{u}^{i+1} means new velocity data.

To linearize viscosity, we use a Picard-like strategy and evaluate the viscous term as:

$$\nu(\mathbf{u}) \varepsilon(\mathbf{u}) = \nu(\mathbf{u}^i) \varepsilon(\mathbf{u}^{i+1}) \quad (4.10)$$

Temperature is taken as something known. If we want to solve Navier-Stokes equations fully coupled with the temperature equation described in chapter 3, we will use an iterative scheme, solving first for temperature, then Navier-Stokes, again temperature and so on. So it is not even needed to linearize the convective part in the temperature equation.

4.2 Numerical Approximation

Now we can approximate the weak form of the problem defined in (4.8) using finite elements and taking into account the linearization defined in (4.9) and (4.10).

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} . Let \mathbf{u}_h be the finite element approximation of the unknown \mathbf{u} ; and, \mathbf{v}_h the test functions associated with $\{\Omega^e\}$.

Here we need to introduce the stabilization concept. As we are going to use equal interpolation for all the unknowns (\mathbf{u} and p), stabilization is not only needed to overcome the numerical instabilities arising when convection is dominant, but also to circumvent the well known inf-sup condition required for the velocity-pressure finite element spaces in order to have pressure stability. Therefore, we are going to use the sub-grid scale stabilized formulation presented in [2].

Hence, the finite element problem reads: Find $\mathbf{u}_h^{i+1} \in V_h$ and $p^{i+1} \in Q_h$ such that:

$$\begin{aligned} \int_{\Omega} \mathbf{v}_h \cdot [\mathbf{u}_h^i \cdot \nabla \mathbf{u}_h^{i+1} + \mathbf{g} \beta \vartheta^i] \, d\Omega + \int_{\Omega} 2 \nabla \mathbf{v}_h : \nu^i \boldsymbol{\varepsilon}(\mathbf{u}_h^{i+1}) \, d\Omega - \int_{\Omega} p_h^{i+1} \nabla \cdot \mathbf{v}_h \, d\Omega \\ + \int_{\Omega} \tau \mathcal{P}_1 \cdot \mathcal{R} \, d\Omega = \int_{\Omega} \mathbf{v}_h \cdot \mathbf{f} \, d\Omega + \int_{\Gamma_{nv}} \mathbf{v}_h \cdot \bar{\mathbf{t}} \, d\Gamma \end{aligned} \quad (4.11a)$$

$$\int_{\Omega} q_h \nabla \cdot \mathbf{u}_h^{i+1} \, d\Omega + \int_{\Omega} \tau \mathcal{P}_2 \cdot \mathcal{R} \, d\Omega = 0 \quad (4.11b)$$

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

$$\mathcal{R} = \mathbf{u}_h^i \cdot \nabla \mathbf{u}_h^{i+1} + \mathbf{g} \beta \vartheta^i - 2 \nabla \cdot [\nu^i \boldsymbol{\varepsilon}(\mathbf{u}_h^{i+1})] + \nabla p^{i+1} - \mathbf{f} \quad (4.12)$$

$$\mathcal{P}_1 = \mathbf{u}_h^i \cdot \nabla \mathbf{v}_h + 2 \nabla \cdot [\nu^i \boldsymbol{\varepsilon}(\mathbf{v}_h)] \quad (4.13)$$

$$\mathcal{P}_2 = \nabla q_h \quad (4.14)$$

$$\tau = \left[\frac{4\nu^i}{h^2} + \frac{2|\mathbf{u}_h^i|}{h} \right]^{-1} \quad (4.15)$$

To obtain the algebraic version of the problem we need some definitions. Let n_{pts} be the total number of nodes of the finite element mesh and let N^a be the standard finite element interpolation function associated to node a , where $a = 1, \dots, n_{pts}$.

The unknowns are interpolated as:

$$u_{h,l} = \sum_{a=1}^{n_{pts}} N^a U_l^a \quad (4.16a)$$

$$p_h = \sum_{a=1}^{n_{pts}} N^a P^a \quad (4.16b)$$

We use upper-case letters (i.e. U_l^a , P^a) to denote the nodal value at node a of the corresponding lower-case variable.

4. NAVIER-STOKES EQUATIONS

The test functions are taken as:

$$v_{h,l} = \delta_{kl} N^b, \quad b = 1, \dots, n_{pts}, \quad k = 1, \dots, n_{sd} \quad (4.17a)$$

$$q_h = N^b, \quad b = 1, \dots, n_{pts} \quad (4.17b)$$

where δ_{kl} is the Kronecker delta, equal to 1 for $k = l$ and 0 otherwise.

From hereafter, we will denote as \mathbf{a} to the velocity in the previous linearization iteration (\mathbf{u}^i) and as \mathbf{u} to the velocity in the current linearization iteration (\mathbf{u}^{i+1}).

4.3 Node-Based Implementation

In this section, we are going to express the approximate problem in terms of solely geometrical dependant integrals:

$$\begin{aligned} & \int_{\Omega} N^b N^a \, d\Omega \\ & \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega, \quad i, j = 1, \dots, n_{sd} \\ & \int_{\Omega} N^b \partial_j N^a \, d\Omega, \quad j = 1, \dots, n_{sd} \\ & \int_{\Omega} \partial_j N^b N^a \, d\Omega, \quad j = 1, \dots, n_{sd} \end{aligned} \quad (4.18)$$

for $a, b = 1, \dots, n_{pts}$

With the integrals in (4.18) and taking $v_{h,l} = \delta_{kl} N^b$ and $q_h = N^b$, we can approximate each part of problem (4.11) in various forms.

Notice that we have not written integrals with derivatives higher than first order because just linear elements are analyzed.

To tackle the problem, we are going to divide it in a convective part, diffusive part and stabilization part. We are going to apply, with the corresponding modifications, the approximations developed in chapters 2 and 3.

Let us consider the matrix form of the problem, the system of equations to solve will be:

$$\begin{bmatrix} \underline{\underline{C}} + \underline{\underline{L}} & \underline{\underline{G}} \\ \underline{\underline{D}} & 0 \end{bmatrix} \begin{bmatrix} \underline{U} \\ \underline{P} \end{bmatrix} + \begin{bmatrix} \underline{\underline{S}}_{1,U} & \underline{\underline{S}}_{1,P} \\ \underline{\underline{S}}_{2,U} & \underline{\underline{S}}_{2,P} \end{bmatrix} \begin{bmatrix} \underline{U} \\ \underline{P} \end{bmatrix} = \begin{bmatrix} \underline{F}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \underline{\underline{S}}_{1,F} & 0 \\ \underline{\underline{S}}_{2,F} & 0 \end{bmatrix} \begin{bmatrix} \underline{F}_1 \\ 0 \end{bmatrix} \quad (4.19)$$

where double underline means a matrix and single underline means a column vector. $\underline{\underline{C}}$ is the convection matrix, $\underline{\underline{L}}$ the viscous matrix, $\underline{\underline{S}}$ the stabilization matrices, $\underline{\underline{D}}$ is the divergence matrix and $\underline{\underline{G}}$ is the gradient matrix.

We need to find expressions for the convection, viscous and stabilization matrices. Divergence and gradient matrices do not represent any difficulty because the node-based implementation is equivalent to the element-based.

To begin, let us define which part of the equation corresponds to the matrix representation.

$$\begin{aligned}
 C &\longrightarrow \int_{\Omega} \mathbf{v}_h \cdot (\mathbf{a}_h \cdot \nabla \mathbf{u}_h) \, d\Omega \\
 L &\longrightarrow \int_{\Omega} 2 \boldsymbol{\varepsilon}(\mathbf{v}_h) : \nu \boldsymbol{\varepsilon}(\mathbf{u}_h) \, d\Omega \\
 G &\longrightarrow - \int_{\Omega} p_h \nabla \cdot \mathbf{v}_h \, d\Omega \\
 D &\longrightarrow \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h \, d\Omega \\
 S_{1,U} &\longrightarrow \int_{\Omega} \tau [(\mathbf{a}_h \cdot \nabla) \mathbf{v}_h] \cdot [(\mathbf{a}_h \cdot \nabla) \mathbf{u}_h] \, d\Omega \\
 S_{1,P} &\longrightarrow \int_{\Omega} \tau [(\mathbf{a}_h \cdot \nabla) \mathbf{v}_h] \cdot [\nabla p_h] \, d\Omega \\
 S_{1,F} &\longrightarrow \int_{\Omega} \tau [(\mathbf{a}_h \cdot \nabla) \mathbf{v}_h] \cdot [\mathbf{f} - \mathbf{g}\beta\vartheta_h] \, d\Omega \\
 S_{2,U} &\longrightarrow \int_{\Omega} \tau [\nabla q_h] \cdot [(\mathbf{a}_h \cdot \nabla) \mathbf{u}_h] \, d\Omega \\
 S_{2,P} &\longrightarrow \int_{\Omega} \tau [\nabla q_h] \cdot [\nabla p_h] \, d\Omega \\
 S_{2,F} &\longrightarrow \int_{\Omega} \tau [\nabla q_h] \cdot [\mathbf{f} - \mathbf{g}\beta\vartheta_h] \, d\Omega
 \end{aligned}$$

Let us denote L_{kl}^{ba} to the element in row b and column a of matrix L corresponding to the k -th moment equation and l -th velocity component.

The viscous term is approximated according to the results obtained from chapter 2. Taking $\mathbf{v}_h = N^b \mathbf{e}_k$ with k fixed with any value from $1, \dots, n_{sd}$ we approximate the

4. NAVIER-STOKES EQUATIONS

viscous term as:

$$L_{kl}^{ba} = \frac{\nu^a + \nu^b}{2} \left[\delta_{kl} \sum_{i=1}^{n_{sd}} \int_{\Omega} \partial_i N^b \partial_i N^a \, d\Omega + \int_{\Omega} \partial_l N^b \partial_k N^a \, d\Omega \right] \quad \forall b \neq a \quad (4.20a)$$

$$L_{kl}^{bb} = - \sum_{a \neq b}^{n_{pts}} L_{kl}^{ba} \quad \forall k, l = 1, \dots, n_{sd} \quad (4.20b)$$

We have proven in chapter 2 that this way of evaluation of the viscous term is consistent and conservative. But this specific expression of the viscous term is not totally symmetric (due to the symmetric velocity gradient) as it was in chapter 2, so some more modifications will be required.

The convective term can be approximated in a similar way as we did in 3. With the appropriate adaptations and taking $\mathbf{v}_h = N^b \mathbf{e}_k$ we have:

$$C_{kl}^{ba} = \delta_{kl} \sum_{j=1}^{n_{sd}} A_j^a \int_{\Omega} N^b \partial_j N^a \, d\Omega \quad (4.21a)$$

In chapter 3 we have proven that this evaluation of the convective term is consistent and conservative.

As we are interested in symmetric forms of the stabilization terms on the main diagonal to be able to fulfil both consistency and conservation at the same time, we are going just to consider the cases when the stabilization parameter τ is evaluated as the average of the values at nodal points a and b .

Assuming we are using just linear elements (P1 or Q1), taking $\mathbf{v}_h = N^b \mathbf{e}_k$ and $q_h = N^b$, the stabilization terms can be approximated as:

$$S_{1,U,kl}^{ba} = \frac{\tau^a + \tau^b}{2} \delta_{kl} \sum_{i,j}^{n_{sd}} A_i^b A_j^a \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \quad (4.22a)$$

$$S_{1,P,k}^{ba} = \frac{\tau^a + \tau^b}{2} \sum_i^{n_{sd}} A_i^b \int_{\Omega} \partial_i N^b \partial_k N^a \, d\Omega \quad (4.22b)$$

$$S_{1,F,kl}^{ba} = \frac{\tau^a + \tau^b}{2} \delta_{kl} \sum_i^{n_{sd}} A_i^b \int_{\Omega} \partial_i N^b N^a \, d\Omega \quad (4.22c)$$

$$S_{2,U,l}^{ba} = \frac{\tau^a + \tau^b}{2} \sum_j^{n_{sd}} A_j^a \int_{\Omega} \partial_l N^b \partial_j N^a \, d\Omega \quad (4.22d)$$

$$S_{2,P}^{ba} = \frac{\tau^a + \tau^b}{2} \sum_i^{n_{sd}} \int_{\Omega} \partial_i N^b \partial_i N^a \, d\Omega \quad (4.22e)$$

$$S_{2,F,l}^{ba} = \frac{\tau^a + \tau^b}{2} \int_{\Omega} \partial_l N^b N^a \, d\Omega \quad (4.22f)$$

4.3.1 Consistency

Let us consider problem (4.11) without forcing term. Additionally, let us consider all the boundary conditions are of Dirichlet type: $\bar{\mathbf{u}} = \mathbf{e}_l$ for any $l = 1, \dots, n_{sd}$. Therefore, $\Gamma_N = \emptyset \wedge \Gamma_D = \partial\Omega$. Furthermore, let us assume pressure is zero everywhere in the domain. In this situation the solution is constant and equal to \mathbf{e}_k in all nodal points.

This behaviour has to be reproduced exactly by the finite element solution because, even with linear interpolation functions, the exact solution belongs to the finite element space. Consequently, the following conditions have to hold:

$$\sum_a^{n_{pts}} C_{kl}^{ba} + L_{kl}^{ba} + S_{1,U,kl}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}, \forall k, l = 1, \dots, n_{sd} \quad (4.23a)$$

$$\sum_a^{n_{pts}} D_l^{ba} + S_{2,U,l}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}, \forall l = 1, \dots, n_{sd} \quad (4.23b)$$

Summing over a , some of the terms are instantly zero. This is the case for D^{ba} , L^{ba} (with consistency correction as presented in (4.20)), C^{ba} (as demonstrated in chapter 3). Hence the conditions (4.23) convert in:

$$\sum_a^{n_{pts}} S_{1,U,kl}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}, \forall k, l = 1, \dots, n_{sd} \quad (4.24a)$$

$$\sum_a^{n_{pts}} S_{2,U,l}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}, \forall l = 1, \dots, n_{sd} \quad (4.24b)$$

Conditions (4.24) can be easily attained correcting the diagonal term as the sum of

4. NAVIER-STOKES EQUATIONS

the off-diagonal terms. So, we have:

$$S_{1,U,kl}^{bb} = - \sum_{a \neq b}^{n_{pts}} S_{1,U,kl}^{ba} = 0 \quad \forall k, l = 1, \dots, n_{sd} \quad (4.25a)$$

$$S_{2,U,l}^{bb} = - \sum_{a \neq b}^{n_{pts}} S_{2,U,l}^{ba} = 0 \quad \forall l = 1, \dots, n_{sd} \quad (4.25b)$$

Now, let us consider problem (4.11) without forcing term. Additionally, let us consider all the boundary conditions are of Dirichlet type: $\bar{\mathbf{u}} = \mathbf{1}$. Therefore, $\Gamma_N = \emptyset \wedge \Gamma_D = \partial\Omega$. Furthermore, let us assume that velocity is constant and equal to one in each of its components everywhere in the domain. In this situation the pressure solution is constant in all nodal points.

This behaviour has to be reproduced exactly by the finite element solution because, even with linear interpolation functions, the exact solution belongs to the finite element space. Assuming conditions (4.23) already hold, the following conditions have to hold:

$$\sum_a^{n_{pts}} G_k^{ba} + S_{1,P,k}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}, \forall k = 1, \dots, n_{sd} \quad (4.26a)$$

$$\sum_a^{n_{pts}} S_{2,P}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts} \quad (4.26b)$$

Summing over a , some of the terms are instantly zero. This is the case for G^{ba} with its corresponding contribution on the boundary. Hence conditions 4.26 convert in:

$$\sum_a^{n_{pts}} S_{1,P,k}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}, \forall k = 1, \dots, n_{sd} \quad (4.27a)$$

$$\sum_a^{n_{pts}} S_{2,P}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts} \quad (4.27b)$$

Conditions (4.24) can be easily attained correcting the diagonal term as the sum of the off-diagonal terms. So, we have:

$$S_{1,P,k}^{bb} = - \sum_{a \neq b}^{n_{pts}} S_{1,P,k}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts}, \forall k = 1, \dots, n_{sd} \quad (4.28a)$$

$$S_{2,P}^{bb} = - \sum_{a \neq b}^{n_{pts}} S_{2,P}^{ba} = 0 \quad \forall b = 1, \dots, n_{pts} \quad (4.28b)$$

Finally, we can not say anything about the stabilization contribution to the forcing term because force is zero.

4.3.2 Conservation

For simplicity, let us consider the problem (4.11) with all boundary conditions of Neumann type (case in which the solution would not be unique). Furthermore, suppose we take the test functions \mathbf{v}_h , q_h as constant. Hence, problem (4.11) imply:

$$\begin{aligned} \int_{\partial\Omega} (\rho_0 \mathbf{u})(\mathbf{u} \cdot \mathbf{n}) \, d\Gamma &= \int_{\Omega} \rho_0 \mathbf{g} \, d\Omega - \int_{\Omega} \rho_0 \nabla p \, d\Omega - \int_{\Omega} \rho_0 \beta \mathbf{g} (\vartheta - \vartheta_0) \, d\Omega \\ &+ \int_{\Omega} \rho_0 [2 \nabla \cdot (\nu \varepsilon(\mathbf{u}))] \, d\Omega \end{aligned} \quad (4.29a)$$

$$\int_{\partial\Omega} \rho_0 \mathbf{u} \cdot \mathbf{n} \, d\Gamma = 0 \quad (4.29b)$$

which, in physical terms, represent linear momentum and mass conservation. This is physically consistent.

However, equations (4.11) are not enforced for constant test functions \mathbf{v}_h , q_h but only for test functions of the form $\mathbf{v}_h = N^b \mathbf{e}_k$, $q_h = N^b$ $b = 1, \dots, n_{\text{point}}$, with k fixed. Since the addition of all shape functions N^b is 1, equations (4.29) can also be obtained taking $v_h = N^b$ in equation (4.11) and adding up over b .

As we have demonstrated in previous chapters, the non-stabilized part of the edge-based approximations represent the conservation statements presented in (4.29). Then, we are just going to focus on the stabilization part. Hence, in matrix form, condition (4.29) can be expressed as:

$$\sum_b \sum_a \sum_l [S_{1,U,kl}^{ba} U_l^a + S_{1,P,k}^{ba} P^a] = \sum_b \sum_a \sum_l S_{1,F,kl}^{ba} F_l^a \quad \forall k = 1, \dots, n_{sd} \quad (4.30a)$$

$$\sum_b \sum_a \sum_l [S_{2,U,l}^{ba} U_l^a + S_{2,P}^{ba} P^a] = \sum_b \sum_a \sum_l S_{2,F,l}^{ba} F_l^a \quad (4.30b)$$

in order to have global conservation.

Now, we have two options: enforcing (4.30) directly or splitting it to make it easier. As element-based finite elements have this property too, we do not see it as over-constraining the approximations but just as trying them to be as similar as the ones

4. NAVIER-STOKES EQUATIONS

obtained from element-based finite element approximations. Then condition (4.30) will become:

$$\sum_b \sum_a \sum_l \left[S_{1,U,kl}^{ba} U_l^a + S_{1,P,k}^{ba} P^a \right] = 0 \quad \forall k = 1, \dots, n_{sd} \quad (4.31a)$$

$$\sum_b \sum_a \sum_l S_{1,F,kl}^{ba} F_l^a = 0 \quad \forall k = 1, \dots, n_{sd} \quad (4.31b)$$

$$\sum_b \sum_a \sum_l \left[S_{2,U,l}^{ba} U_l^a + S_{2,P}^{ba} P^a \right] = 0 \quad (4.31c)$$

$$\sum_b \sum_a \sum_l S_{2,F,l}^{ba} F_l^a = 0 \quad (4.31d)$$

As $S_{1,U}$ and $S_{2,P}$ are symmetric and because they were corrected for consistency, the sum of them over b is zero. Hence condition (4.31) becomes:

$$\sum_b \sum_a S_{1,P,k}^{ba} P^a = 0 \quad \forall k = 1, \dots, n_{sd} \quad (4.32a)$$

$$\sum_b \sum_a \sum_l S_{1,F,kl}^{ba} F_l^a = 0 \quad \forall k = 1, \dots, n_{sd} \quad (4.32b)$$

$$\sum_b \sum_a \sum_l S_{2,U,l}^{ba} U_l^a = 0 \quad (4.32c)$$

$$\sum_b \sum_a \sum_l S_{2,F,l}^{ba} F_l^a = 0 \quad (4.32d)$$

To avoid dealing with the sum over a because it involves the values of the unknowns, we express condition (4.32) as:

$$\sum_b S_{1,P,k}^{ba} = 0 \quad \forall k = 1, \dots, n_{sd} \quad (4.33a)$$

$$\sum_b S_{1,F,kl}^{ba} = 0 \quad \forall k, l = 1, \dots, n_{sd} \quad (4.33b)$$

$$\sum_b S_{2,U,l}^{ba} = 0 \quad \forall l = 1, \dots, n_{sd} \quad (4.33c)$$

$$\sum_b S_{2,F,l}^{ba} = 0 \quad \forall l = 1, \dots, n_{sd} \quad (4.33d)$$

Again, condition (4.33) is not over-restrictive because element-based formulations have this property too.

Notice that, even with consistency correction, condition (4.33) does not hold because the matrices are neither symmetric nor skew-symmetric. Therefore something should be done.

Let us focus on $S_{1,P}^{ba}$. We can interpolate p as we presented in (4.22b) and we will require the consistency condition or we can interpolate ∇p and treat it as an external force evaluated at the previous iteration. So, $S_{1,P}^{ba}$ will be calculated as:

$$S_{1,P,kl}^{ba} = \delta_{kl} \frac{\tau^a + \tau^b}{2} \sum_i^{n_{sd}} A_i^b \int_{\Omega} \partial_i N^b N^a \, d\Omega \quad (4.34)$$

and corrected to ensure conservation:

$$S_{1,P,kl}^{aa} = - \sum_{b \neq a} S_{1,P,kl}^{ba} \quad \forall a = 1, \dots, n_{pts}, \quad \forall k = 1, \dots, n_{sd} \quad (4.35)$$

As the gradients of the solution are undefined on the nodes because we are using linear elements, we will use a standard least-square smoothing from the elemental values. Denoting Gp_i^a to the component i of the gradient of the pressure at node a ($(\nabla p)_i^a$), we can obtain Gp_i^a solving the following system of equations:

$$\sum_{a=1}^{n_{pts}} \left(\int_{\Omega} N^b N^a \right) Gp_i^a = \sum_{a=1}^{n_{pts}} \left(\int_{\Omega} N^b \partial_i N^a \right) P^a, \quad b = 1, \dots, n_{pts} \quad (4.36)$$

Additionally, to avoid solving (4.36), because it is a big system of equations compared with the final system of equations to solve, we are going to use a lumped mass matrix and the system solves explicitly.

Now, let us focus on $S_{2,U}^{ba}$. As we pointed out before, the matrix is not purely symmetric or purely skew-symmetric. Therefore consistency does not implies conservation. We will follow a similar approach as we did with $S_{1,P}^{ba}$. The part $(\mathbf{a} \cdot \nabla) \mathbf{u}$ will be considered as a force term. Nodal values will be obtained using an analogous procedure as we presented in (4.36). Hence, it will be calculated as:

$$S_{2,U,l}^{ba} = \frac{\tau^a + \tau^b}{2} \int_{\Omega} \partial_l N^b N^a \, d\Omega, \quad \forall b \neq a \quad (4.37)$$

with the corresponding conservation correction:

$$S_{2,U,l}^{aa} = - \sum_{b \neq a} S_{2,U,l}^{ba} \quad \forall a = 1, \dots, n_{pts}, \quad \forall k = l, \dots, n_{sd} \quad (4.38)$$

For the viscous matrix L we have followed a similar procedure in order to satisfy the consistency and conservation conditions because it is not strictly symmetric.

4. NAVIER-STOKES EQUATIONS

Now, let us have a look to the stabilization contribution to the forcing term: $S_{1,F}$ and $S_{2,F}$. In order to satisfy condition (4.33), we are going to do a correction: the off-diagonal terms will be calculated as stated before and the diagonal terms will be calculated as the negative of the sum of the off-diagonal terms in the same column. So:

$$S_{1,F,kl}^{aa} = - \sum_{b \neq a} S_{1,F,kl}^{ba} \quad \forall a = 1, \dots, n_{pts}, \forall k, l = 1, \dots, n_{sd} \quad (4.39a)$$

$$S_{2,F,l}^{aa} = - \sum_{b \neq a} S_{2,F,l}^{ba} \quad \forall a = 1, \dots, n_{pts}, \forall l = 1, \dots, n_{sd} \quad (4.39b)$$

In the following section we are going to test the approximation proposed in order to determine its convergence properties.

4.4 Convergence Test

In order to produce a convergence test, we need to import important elements: a reference norm and a model problem solved over different meshes.

For simplicity, the working norm chosen is the L^2 norm. We could have chosen a norm associated to the problem and stabilization method used, but as we get good results in that norm, we stick to it. To make programming simpler, the L^2 -norm error (normalized) is computed as:

$$E = \left[\sum_{a=1}^{n_{pts}} \sum_{i=1}^{n_{sd}} (U_i^a - u_i(\mathbf{x}^a))^2 \right]^{1/2} \left[\sum_{a=1}^{n_{pts}} \sum_{i=1}^{n_{sd}} u_i(\mathbf{x}^a)^2 \right]^{-1/2} \quad (4.40)$$

where $\mathbf{u}(\mathbf{x}^a)$ is the exact solution at \mathbf{x}^a .

In order to create the convergence plots of the proposed nodal approximations, a specific problem with known solution is needed. We have selected a two dimensional steady-state problem. Variable viscosity is not our main target because it is not present in the stabilization terms for linear elements, so we consider constant viscosity. Additionally, as we are interested in convection dominated flows, we take a small viscosity, then $\nu = 0.001$. We take Ω as the unit square ($[0,1] \times [0,1]$) and the force term in a way such that $\mathbf{u} = (f(x)g'(y), -f'(x)g(y))$ and $p = 0$ is solution of equation (4.1) taking $\rho_0 = 1$, $f(x) = x^2(1-x)^2$ and $g(y) = y^2(1-y)^2$.

We have used four finite element meshes discretized with triangles with uniform sizes of $h = 0.2, 0.10, 0.05$ and 0.01 . Meshes are completely unstructured.

In table 4.1 we can see the values obtained and in figure 4.1 we can see the results plotted.

h	0.20	0.10	0.05	0.01
Conservative	0.17405	0.04372	0.010494	0.00048
Only Consistent	0.14788	0.03601	0.010269	0.00048

Table 4.1: Navier Stokes Convergence Test (L2-Error of the velocity)

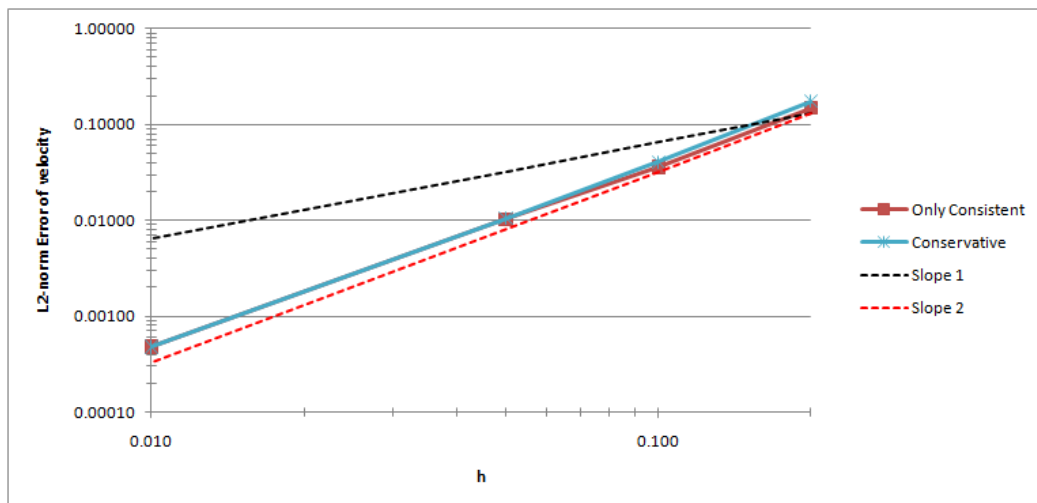


Figure 4.1: Navier Stokes Convergence Test

As can be seen in figure 4.1, the approximations proposed work fine and are consistent and conservative. The convergence rate in $L2$ -norm for linear elements ($p1$ or $q1$) is approximately $h^{1.93}$.

Just to conclude, let us write the final form of the final consistent and conservative

4. NAVIER-STOKES EQUATIONS

approximation proposed:

$$\begin{aligned} \begin{bmatrix} \underline{\underline{C}} + \underline{\underline{L}} & \underline{\underline{G}} \\ \underline{\underline{D}} & 0 \end{bmatrix} \begin{bmatrix} \underline{\underline{U}} \\ \underline{\underline{P}} \end{bmatrix} + \begin{bmatrix} \underline{\underline{S}}_{1,U} & 0 \\ 0 & \underline{\underline{S}}_{2,P} \end{bmatrix} \begin{bmatrix} \underline{\underline{U}} \\ \underline{\underline{P}} \end{bmatrix} = \begin{bmatrix} \underline{\underline{F}}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \underline{\underline{S}}_{1,F} & 0 \\ \underline{\underline{S}}_{2,F} & 0 \end{bmatrix} \begin{bmatrix} \underline{\underline{F}}_1 \\ 0 \end{bmatrix} \\ - \begin{bmatrix} \underline{\underline{L}}_{\text{off}} & \underline{\underline{S}}_{1,P} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \nabla \mathbf{u} \\ \nabla p \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ \underline{\underline{S}}_{2,U} & 0 \end{bmatrix} \begin{bmatrix} (\mathbf{a} \cdot \nabla) \mathbf{u} \\ 0 \end{bmatrix} \end{aligned} \quad (4.41)$$

$$L_{kl}^{ba} = \delta_{kl} \frac{\nu^a + \nu^b}{2} \left[\sum_{i=1}^{n_{sd}} \int_{\Omega} \partial_i N^b \partial_i N^a \, d\Omega + \int_{\Omega} \partial_k N^b \partial_l N^a \, d\Omega \right] \quad (4.42a)$$

$$L_{\text{off},kl}^{ba} = (1 - \delta_{kl}) \frac{\nu^a + \nu^b}{2} \int_{\Omega} \partial_k N^b N^a \, d\Omega \quad (4.42b)$$

$$C_{kl}^{ba} = \delta_{kl} \sum_{j=1}^{n_{sd}} A_j^a \int_{\Omega} N^b \partial_j N^a \, d\Omega \quad (4.42c)$$

$$S_{1,U,kl}^{ba} = \delta_{kl} \frac{\tau^a + \tau^b}{2} \sum_{i,j}^{n_{sd}} A_i^b A_j^a \int_{\Omega} \partial_i N^b \partial_j N^a \, d\Omega \quad \forall a \neq b \quad (4.42d)$$

$$S_{1,U,kl}^{bb} = - \sum_{a \neq b}^{n_{pts}} S_{1,U,kl}^{ba} \quad (4.42e)$$

$$S_{2,P}^{ba} = \frac{\tau^a + \tau^b}{2} \sum_i^{n_{sd}} \int_{\Omega} \partial_i N^b \partial_i N^a \, d\Omega \quad \forall a \neq b \quad (4.42f)$$

$$S_{2,P}^{bb} = - \sum_{a \neq b}^{n_{pts}} S_{2,P}^{ba} \quad (4.42g)$$

$$S_{1,P,kl}^{ba} = \delta_{kl} \frac{\tau^a + \tau^b}{2} \sum_i^{n_{sd}} A_i^b \int_{\Omega} \partial_i N^b N^a \, d\Omega \quad \forall b \neq a \quad (4.42h)$$

$$S_{1,P,kl}^{aa} = - \sum_{b \neq a}^{n_{pts}} S_{1,P,kl}^{ba} \quad (4.42i)$$

$$S_{2,U,l}^{ba} = \frac{\tau^a + \tau^b}{2} \int_{\Omega} \partial_l N^b N^a \, d\Omega \quad \forall b \neq a \quad (4.42j)$$

$$S_{2,U,l}^{aa} = - \sum_{b \neq a}^{n_{pts}} S_{2,U,l}^{ba} \quad (4.42k)$$

$$S_{1,F,kl}^{ba} = \delta_{kl} \frac{\tau^a + \tau^b}{2} \sum_i^{n_{sd}} A_i^b \int_{\Omega} \partial_i N^b N^a \, d\Omega \quad \forall b \neq a \quad (4.42l)$$

$$S_{1,F,kl}^{aa} = - \sum_{b \neq a}^{n_{pts}} S_{1,F,kl}^{ba} \quad (4.42m)$$

$$S_{2,F,l}^{ba} = \frac{\tau^a + \tau^b}{2} \int_{\Omega} \partial_l N^b N^a \, d\Omega \quad \forall b \neq a \quad (4.42n)$$

$$S_{2,F,l}^{aa} = - \sum_{b \neq a}^{n_{pts}} S_{2,F,l}^{ba} \quad (4.42o)$$

Notice that, for a given node b , the evaluation of the matrices does not require to loop over all a nodes, it just requires to loop over the a nodes connected to node b .

4. NAVIER-STOKES EQUATIONS

Chapter 5

Applications

5.1 Laminar Flow over a Forward-facing step of a Non-Newtonian fluid

5.1.1 Problem Description

The problem consists in a 4:1 plane extrusion problem. We are going to use the symmetry of the problem, so we solve just half of it. The computational domain chosen is $[0, 16] \times [0, 4] \cup [16, 32] \times [3, 4]$. Fluid enters from the left (see figure 5.1) and leaves through the right.

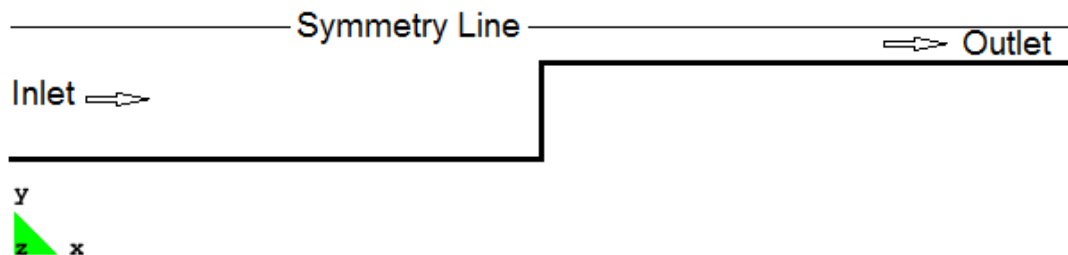


Figure 5.1: Example 1 Domain

The fluid is non-newtonian, its viscosity is not constant a varies according to the power law described in (4.4). The values of the physical constants used for this problem are: $K_0 = 10^6$, $n = 0.4$, $\rho_0 = 1200$. Since for this value of n viscosity varies considerably and viscosity tends to infinity when $I^2(\varepsilon)$ tends to zero, a cut-off value of $\mu_c = 10^{12}$ has been used.

5. APPLICATIONS

5.1.2 Boundary Conditions

Fluid enters with a prescribed velocity $(f(y), 0)$, where $f(y)$ is a parabolic profile with maximum value of 1 at $y = 4$ and minimum value of 0 at $y = 0$. On the walls, the non-slip boundary condition is prescribed $\mathbf{u} = (0, 0)$. On the symmetry line, velocity is left free in the x -direction and velocity is prescribed as zero in the y -direction. Finally, the outlet is left free in the x -direction and velocity is prescribed as zero in the y -direction. In figure 5.2 you can see clearly the boundary conditions.

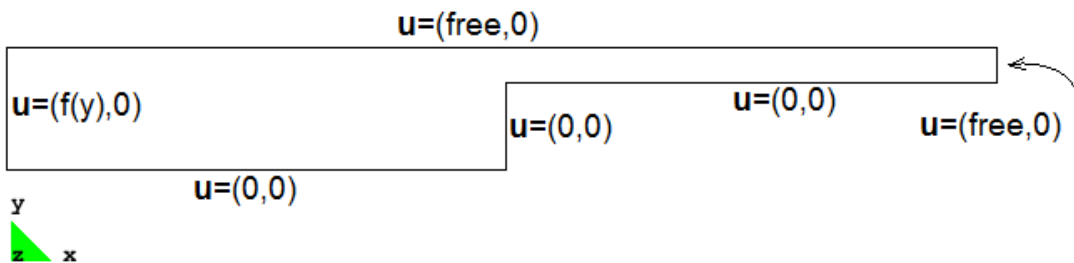


Figure 5.2: Example 1 Boundary Conditions

5.1.3 Meshing

The domain was discretized using 2016 linear quadrilaterals ($q1$) and 2119 nodal points. The mesh can be seen in 5.3

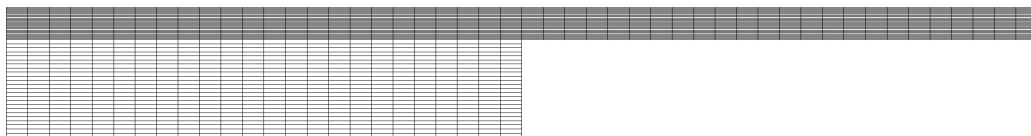


Figure 5.3: Example 1 Mesh

5.1.4 Results

In figure 5.4 you can see the velocity magnitude contour fill. In figure 5.5 you can see the pressure contour fill. In figure 5.6 you can see the streamlines.

As can be seen in the results, the mesh seems too coarse in some zones and the solution is not captured very well. Despite of that, the mass flow entering the domain is the same going out of it, so mass is conserved.

5.1 Laminar Flow over a Forward-facing step of a Non-Newtonian fluid

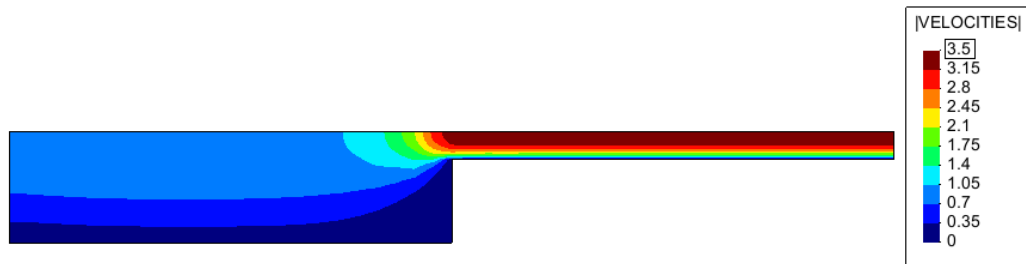


Figure 5.4: Example 1 velocity contour fill

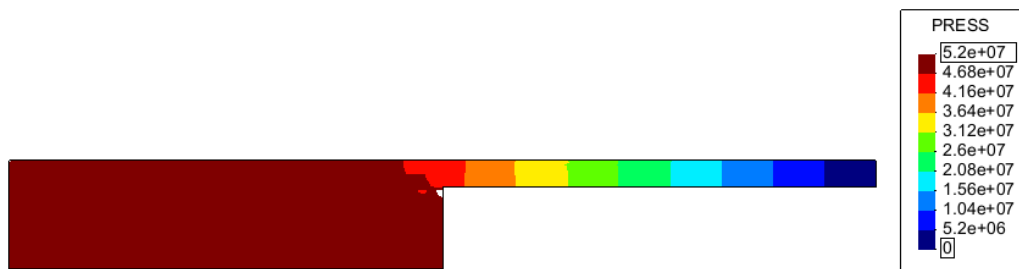


Figure 5.5: Example 1 pressure contour fill

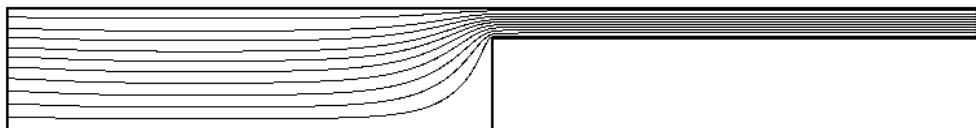


Figure 5.6: Example 1 streamlines

5. APPLICATIONS

Just to finish, we have solved the same problem with a finer mesh of 25000 quadrilaterals and 25401 nodal points and the pressure obtained is shown in figure 5.7

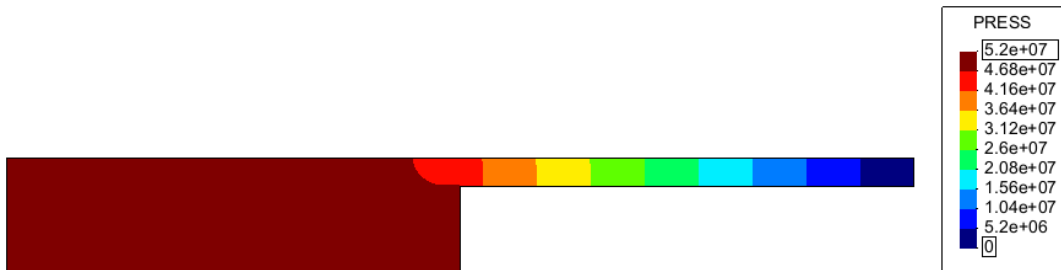


Figure 5.7: Example 1 Pressure Contour Fill with finer mesh

Now it can be seen how the results are much better. We do not show velocity contour or streamlines because are very similar to the ones shown before.

5.2 Thermally coupled flow in a cavity

5.2.1 Problem Description

The problem consists in fluid flow in a square cavity due to thermal effects (natural convection). The computational domain chosen is $[0, 1] \times [0, 1]$. Fluid is inside the cavity and due to the temperature gradient starts to move.

The fluid has constant viscosity μ . Physical properties have been chosen in a way to have a unique and stationary solution. The fluid flow problem and the thermal problem are coupled through the Boussinesq assumption.

Let us denote as L to the characteristic length of the problem and as G_ϑ to the characteristic temperature gradient. Additionally, let us define the Grashof number Gr and Prandtl number Pr :

$$Gr = \frac{\beta |g| L^3 G_\vartheta}{\nu} \quad Pr = \frac{\rho_0 c_0 \nu}{k}$$

Where β is the thermal expansion coefficient, g is the gravity, ν is the kinematic viscosity, ρ is the density and c is the specific heat capacity.

Taking $L = 1$ and $G_\vartheta = 1$, physical properties have been chosen in order to have $Pr = 1$ and $Gr = 1(10)^3$. For this combination of Prandtl and Grashof numbers there is a stable stationary solution, so we can solve the problem in steady-state mode.

5.2.2 Boundary Conditions

Here we need to specify two sets of boundary conditions: Navier-Stokes boundary conditions and thermal boundary conditions. For Navier-Stokes, the non-slip boundary condition has been chosen on all the boundary. For the temperature problem, temperature is prescribed as 1 on $x = 0$ and 0 on $x = 1$. Top and bottom walls have Neumann boundaries with $\bar{h} = 0$ (perfect thermal insulation). In figure 5.8 and 5.9 you can see the boundary conditions.

5.2.3 Meshing

The domain was discretized using 2704 linear quadrilaterals ($q1$) and 2809 nodal points. Mesh was refined near the boundaries to capture the velocity gradients appearing there. The mesh can be seen in figure 5.10.

5. APPLICATIONS

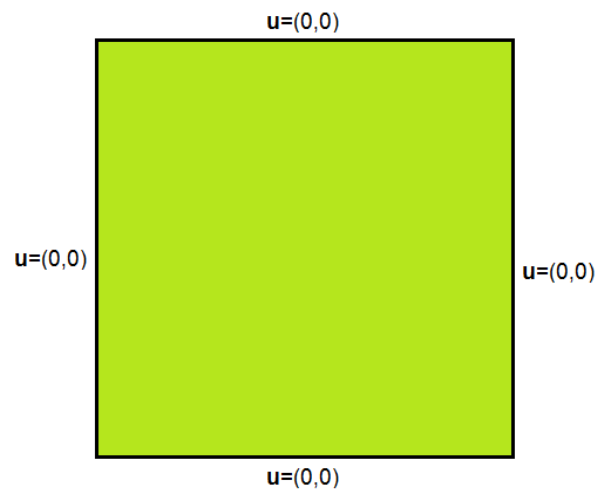


Figure 5.8: Example 2 Velocity Boundary Conditions

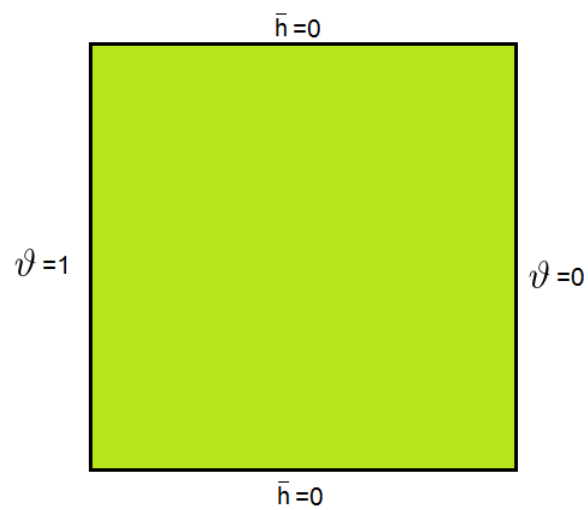


Figure 5.9: Example 2 Temperature Boundary Conditions

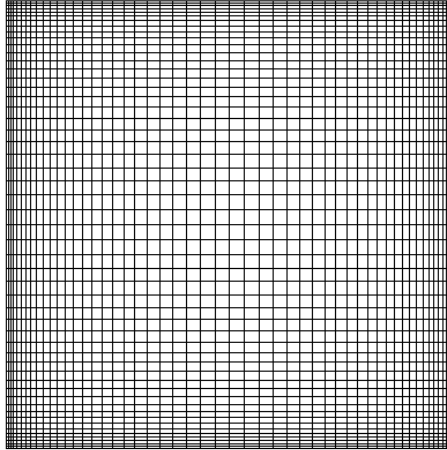


Figure 5.10: Example 2 Mesh

5.2.4 Results

In figure 5.11 you can see the velocity magnitude contour fill. In figure 5.12 you can see the pressure contour fill. In figure 5.13 you can see the streamlines. In figure 5.14 you can see the temperature contour fill.

As can be seen in the results, there are no velocity or pressure instabilities. Additionally, the results obtained are very similar to the ones obtained using an element-based finite element code.

5. APPLICATIONS

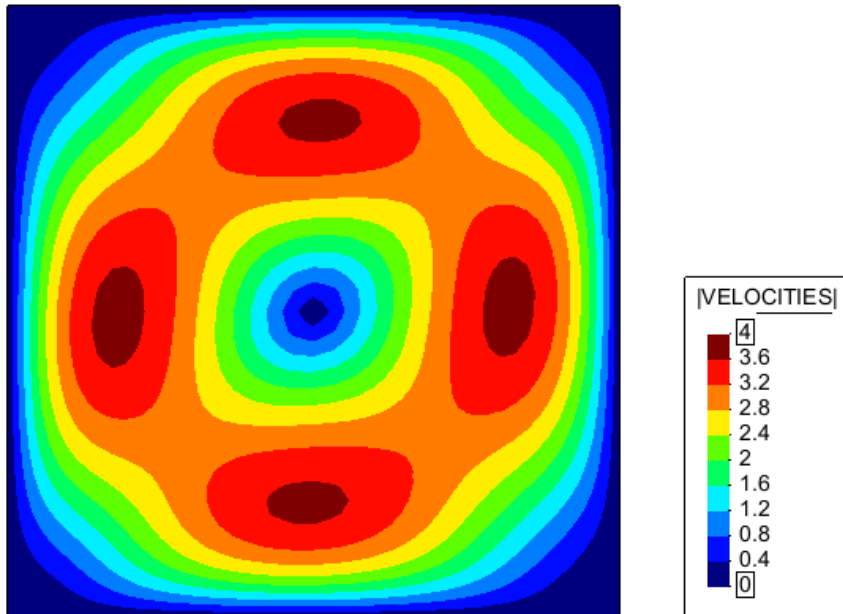


Figure 5.11: Example 2 Velocity Contour Fill

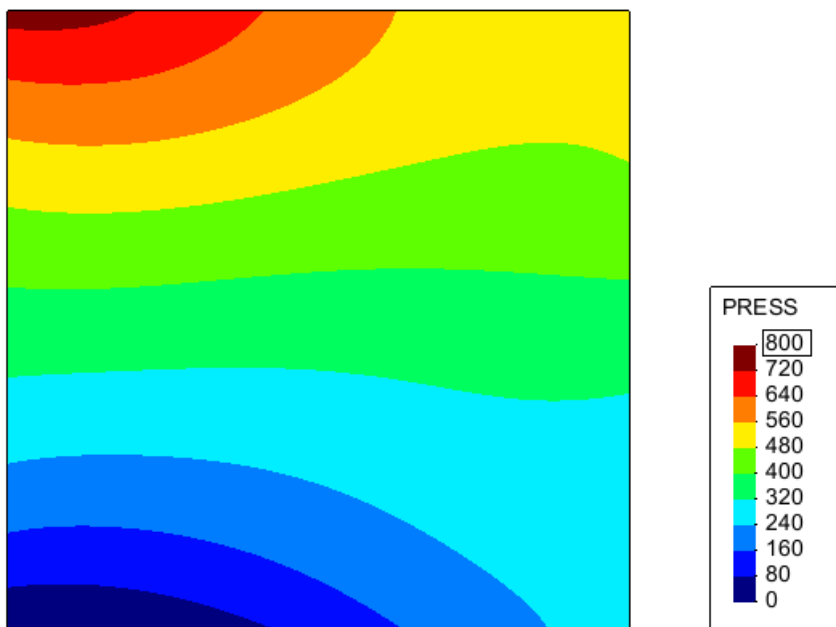


Figure 5.12: Example 2 Pressure Contour Fill

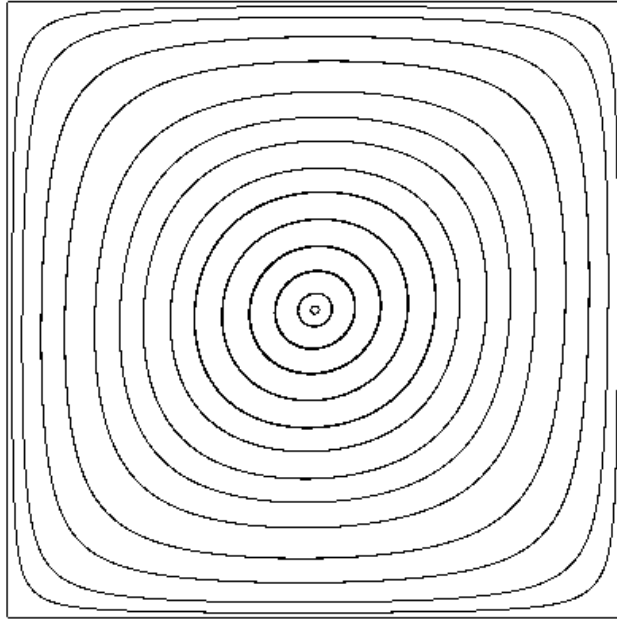


Figure 5.13: Example 2 Streamlines

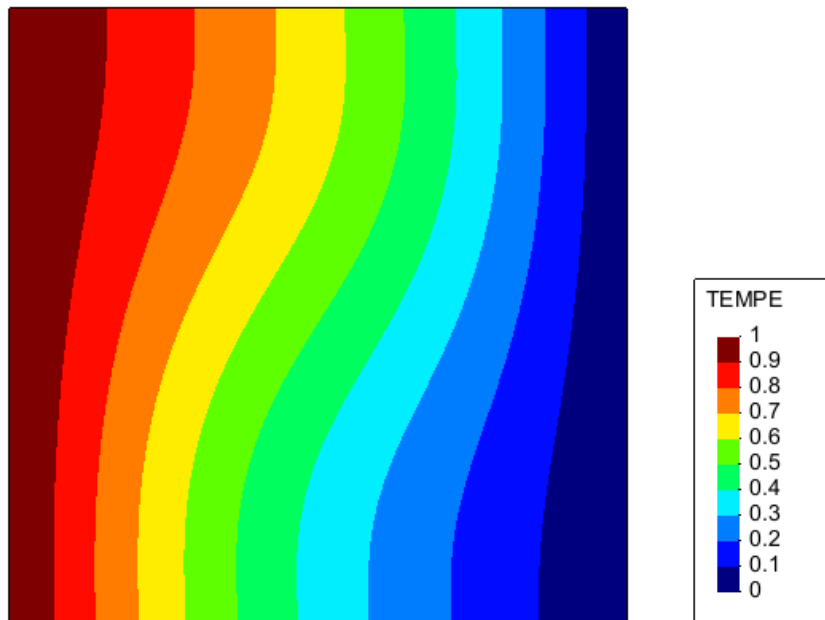


Figure 5.14: Example 2 Temperature Contour Fill

5. APPLICATIONS

5.3 Flow over a cylinder

5.3.1 Problem Description

The problem consists in a fluid around an infinite cylinder. The computational domain chosen is $[0, 16] \times [0, 8] \setminus D$, where D is the cylinder diameter taken as 1. The fluid has constant viscosity μ .

5.3.2 Boundary Conditions

At $x = 0$ a prescribed velocity of $(1, 0)$ is imposed. On the cylinder surface, the non-slip boundary condition is prescribed $\mathbf{u} = (0, 0)$. On $y = 0$ and $y = 8$ velocity is left free in the x -direction and velocity is prescribed as zero in the y -direction. Finally, the outlet ($x = 16$) is left free in the x -direction and y -direction. In figure 5.15 you can see the boundary conditions.

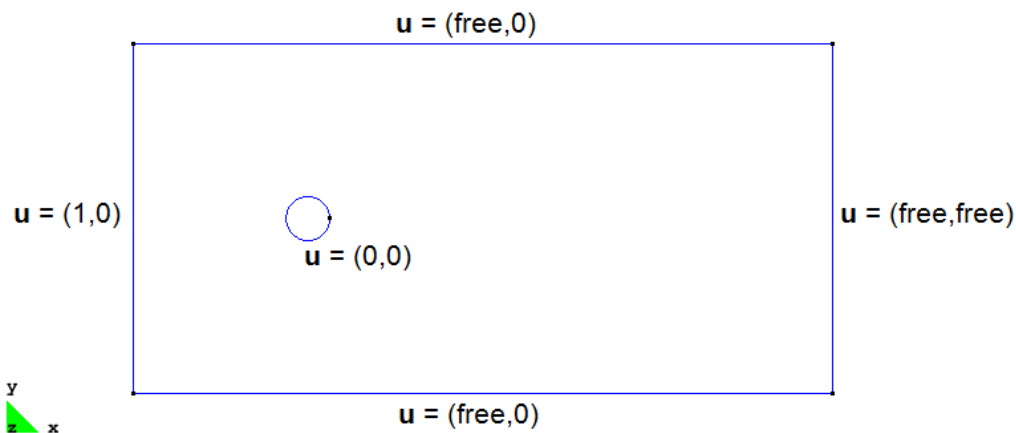


Figure 5.15: Example 3 Boundary Conditions

5.3.3 Meshing

The domain was discretized using 7182 linear triangles ($p1$) and 3844 nodal points. The mesh was refined near the cylinder to capture the velocity gradients appearing there. The mesh can be seen in figure 5.16.

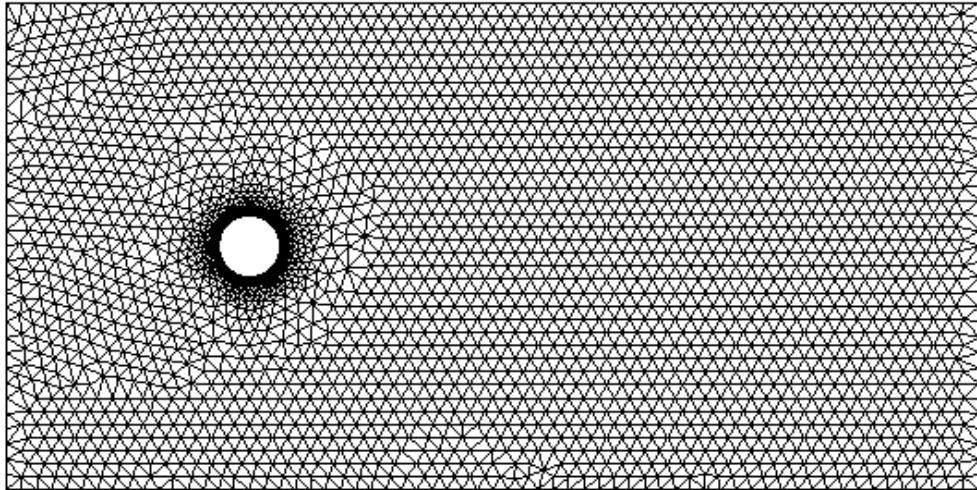


Figure 5.16: Example 3 Mesh

5.3.4 Results

In figure 5.17 we can see the streamlines at given time step. There it can be seen the characteristic vortex shedding.

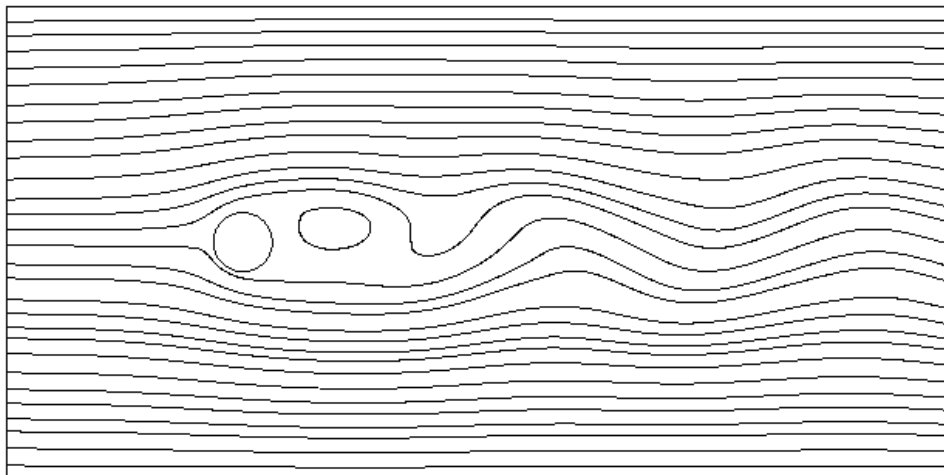


Figure 5.17: Example 3 Streamlines

The period of the oscillations found was 6.1 time units. In the same case, [2] reported a period of 5.9 time units.

In figures 5.18 and 5.19 we can see a velocity and pressure contours at the same

5. APPLICATIONS

time step as figure 5.17

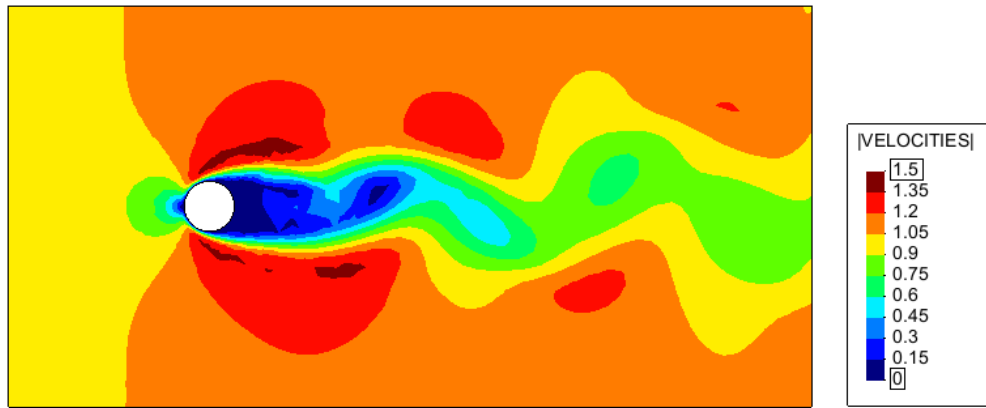


Figure 5.18: Example 3 Velocity Contour Fill

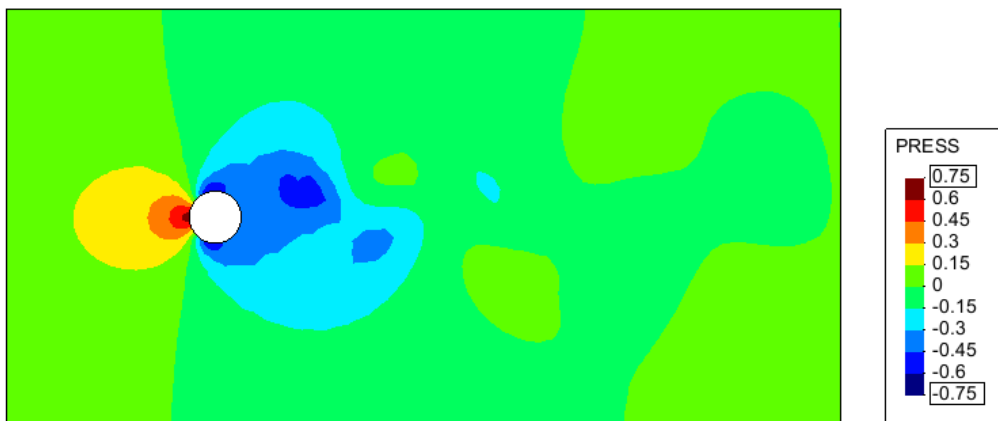


Figure 5.19: Example 3 Pressure Contour Fill

5.4 3D Lid Driven Cavity Flow

5.4.1 Problem Description

Here a classical 3D benchmark problem is addressed. It consists in a fluid enclosed in a cavity of cross sectional aspect ratio $\Gamma = 1$ and span aspect ratio $\Lambda = 1$. The domain chosen is $[0, L] \times [0, L] \times [0, L]$, $L = 1$.

The fluid into consideration has constant viscosity $\mu = 0.01$ and constant density $\rho_0 = 1$.

The Reynolds number is defined as:

$$Re = \frac{\rho_0 U_0 L}{\mu} \quad (5.1)$$

where U_0 is the characteristic velocity and L is the characteristic length. We are going to show the results for a unsteady solution at $Re = 100$.

5.4.2 Boundary Conditions

Fluid motion is induced by a prescribed movement of the wall $y = L$. All the other boundary surfaces have a prescribed velocity $\mathbf{u} = (0, 0, 0)$. In figure 5.20 you can see the boundary conditions.

The velocity boundary condition U was defined in the same way as [4] did. They define the velocity U as:

$$U = U_0 \left(1 - \left(\frac{x}{L}\right)^{18}\right)^2 \left(1 - \left(\frac{z}{L}\right)^{18}\right)^2 \quad (5.2)$$

to avoid discontinuity on the boundary conditions.

5.4.3 Meshing

The domain was discretized using linear hexahedral elements (8-node elements) using $N_x = 24$ divisions in the x -direction, $N_y = 24$ divisions in the y -direction and $N_z = 16$ divisions in the z -direction.

The mesh obtained can be seen in figure 5.21

5. APPLICATIONS

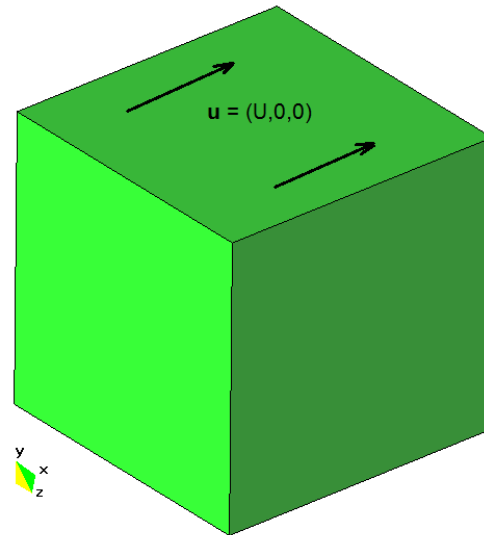


Figure 5.20: Example 4 Boundary Conditions

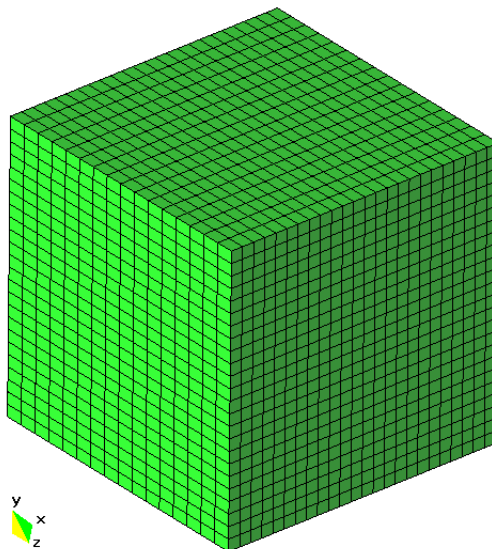


Figure 5.21: Example 4 Mesh

5.4.4 Results

In figure 5.22 we can see a y -velocity profile along a line compared with the results obtained from an element-based implementation. In figure 5.23 we can see a x -velocity profile along a line compared with the results obtained from an element-based implementation.

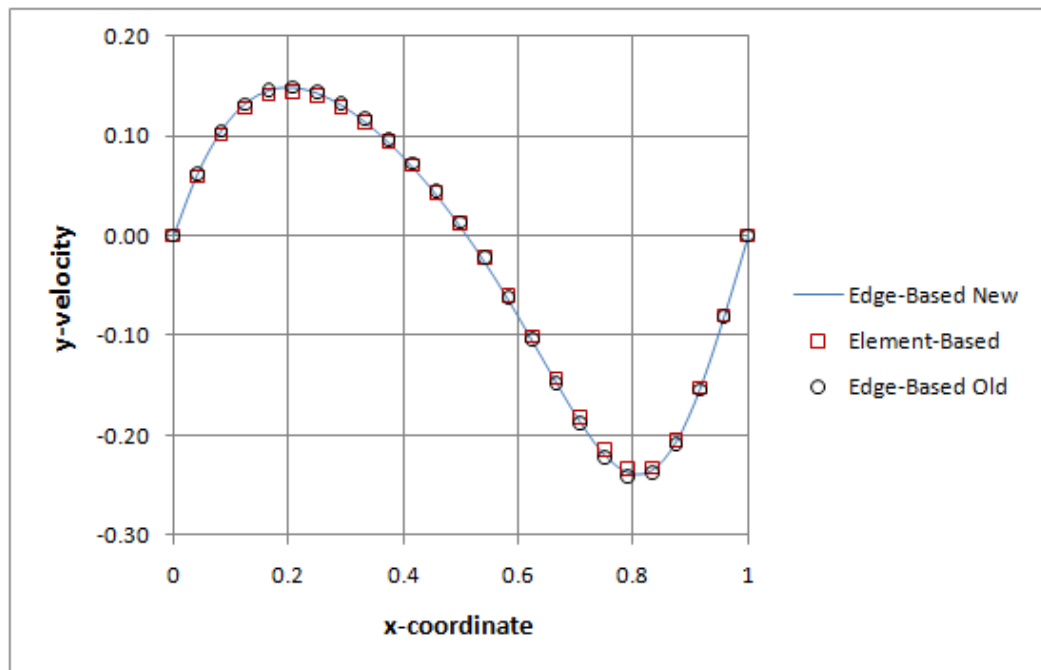


Figure 5.22: Example 4 y -velocity on a line passing through points $(0.0,0.5,0.5)$ and $(1.0,0.5,0.5)$

As can be seen in figures 5.22 and 5.23 the proposed approximations produce results very similar to the element-based results.

5. APPLICATIONS

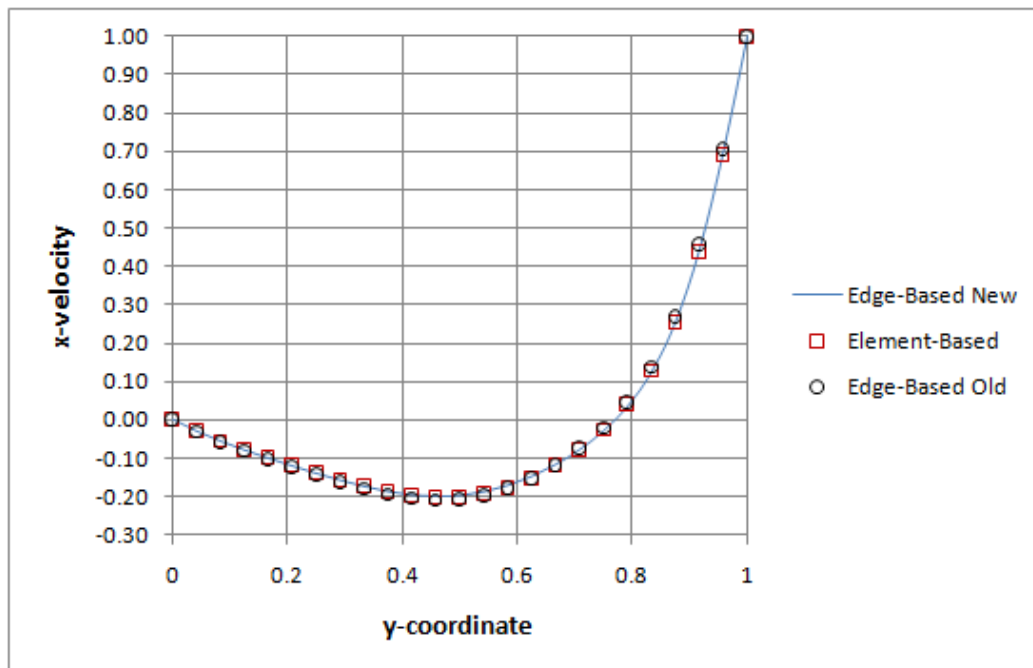


Figure 5.23: Example 4 x -velocity on a line passing trough points $(0.5,0.00,0.5)$ and $(0.5,1.0,0.5)$

Chapter 6

Conclusions

6.1 Conclusions

An exactly global-conservative edge-based approximation was implemented for the diffusion equation. The developments were made for linear elements (p_1 and q_1). It handles very well variable diffusivity. Additionally, it converges with optimal rate for linear elements in its respective norm.

Additionally, the convection-diffusion equation was considered. It is stabilized using the sub-grid scale concept. The edge-based finite element method proposed is exactly conservative (globally) and has optimal convergence rate for linear elements in the respective working norm.

Finally, Navier-Stokes equations were considered. The stabilized formulation allows using an equal velocity-pressure interpolation, so it suits very well the edge-based implementation. An exactly global conservative implementation was proposed: conserves mass and conserves linear momentum. Several numerical examples were solved with the current implementation to illustrate its capabilities and to verify it works as expected.

Notice that the edge-based implementation proposed is globally conservative even for coarse meshes. Coarse meshes are not practical because when we solve a problem, we want a reasonable accuracy in each point of the domain (and we require a fine enough mesh as a consequence). Anyway, we consider that being globally conservative is a plus because the equations solved are conservation statements.

Furthermore, observe that the consistency/conservation corrections were made as

6. CONCLUSIONS

simple as possible to avoid adding too much computational cost to accomplish conservation.

6.2 Future Research

For the Navier-Stokes equations we used a very simple linearization method: Picard Method. Newton-Rapson linearization method can be implemented to improve the non-linear convergence.

The approach followed to develop the conservative edge-based implementations can be extended to quadratic elements ($p2$ or $q2$) or even higher order elements. Special attention should be put in the second-order derivatives that will appear when using higher order elements.

Additionally, improved corrections might be developed to ensure consistency not only for a constant solution, but for a linear solution.

References

- [1] R. CODINA. **Comparison of some finite element methods for solving the diffusion-convection-reaction equation.** *Computer Methods in Applied Mechanics and Engineering*, **156**(1-4):185 – 210, 1998. 23
- [2] R. CODINA. **A nodal-based implementation of a stabilized finite element method for incompressible flow problems.** *International Journal for Numerical Methods in Fluids*, **33**(5):737–766, 2000. 1, 44, 69
- [3] R. GHEZ. *Diffusion Phenomena: Cases and Studies*. Springer, 2001. 10
- [4] E. F. LINS, R. N. ELIAS, G. M. GUERRA, F. A. ROCHINHA, AND A. L. G. A. COUTINHO. **Edge-based finite element implementation of the residual-based variational multiscale method.** *International Journal for Numerical Methods in Fluids*, **61**(1):1–22, 2009. 71
- [5] W. LUBE. **Stabilized finite element methods for singularly perturbed parabolic problems.** *Applied Numerical Mathematics*, **17**(4):431 – 459, 1995. 31
- [6] A. QUARTERONI. *Numerical Approximation of Partial Differential Equations*. Springer, 2008. 16, 22
- [7] A. QUARTERONI. *Numerical Models for Differential Problems*. Sprin, 2009. 23

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such.

The thesis work was conducted from September/2010 to June/2011 under the supervision of Ramon Codina i Rovira at Universitat Politecnica de Catalunya.

Barcelona, June 2011