Master's Thesis

Solution of the Navier-Stokes Equation using the Method of Characteristic Curves

Student
José David Villegas Jiménez

Responsible professor
Manuel Julio García Ruiz

Department of Mechanical Engineering
Universidad EAFIT
Medellín

January, 2014

Abstract

This project deals about the solution of the Navier Stokes Equations by the Method of Characteristics. This method is used to eliminate the convective part of equations of the convection-diffusion type, conducting the material derivative in a Lagrangian manner along the characteristic curves of each node in a fixed grid. Following this approach, the method is able to solve the incompressible Navier Stokes Equations with the advantage of using large timesteps. In the present case, the solution of the well known Lid Driven Cavity Flow problem is obtained for several Reynolds numbers, showing good agreement when compared to solutions obtained by other methods.

Keywords: Characteristic curves, Finite Element Method, Lagrange-Galerkin Method, Backward Euler, Lid Driven Cavity Flow

Contents

Introduction

The description of fluid flow behavior is governed by the Navier Stokes Equation. Although there is not yet an analytical solution for this equation, several numerical techniques have been developed, achieving approximate solutions that have contributed to the solution of problems in different fields of Engineering. Methods such as Finite Differences and Finite Volumes were developed decades ago and have been the inspiration for new methods. The Finite Element Method was intended for structural analysis but it has become popular in fluid dynamics because of its ability to treat boundary conditions and its well defined mathematical basis.

The traditional way to solve the Navier Stokes system in FEM is by using the Galerkin method. The lack of upwinding techniques (adaptative methods according to the direction of propagation of the transported field) evidenced the popularity of Finite Difference and Finite Volume methods. Further research in FEM helped the method to gain popularity due to upwinding techniques, like the Streamline Upwind Petrov Galerkin (SUPG), becoming the most suitable method to deal with instabilities and to improve the finite element solution. Segal (2011)

Other methods to solve the transient incompressible Navier Stokes system are the pressure correction methods, optimization strategies (Uzawa algorithm)Temam (1977), projection schemes, Fractional Step Methods and recently, a group of meshless methods which are becoming popular because complex geometries as well as large deformations, nonlinearities, crack simulations can be treated with ease. The problem with some of these methods is the definition of boundary conditions. These kinds of methods include, the Element-Free Galerkin Method (EFG) Belytschko et al. (1994), Isoparametric Finite Point Method Wang and Takao (2004), Meshless local Petrov-Galerkin (MLPG) Atluri and Zhu (1998), Smooth Particle Hydrodynamics (SPH)Gingold and Monaghan (1977), Moving Particle Semi-Implicit Method (MPS)Koshizuka et al. (1998), among others.

The characteristic based method is followed according to the work done by Zienkiewicz et al. (2005), Zienkiewicz et al. (1999) and contributions of (Nithiarasu (2003), Nithiarasu (2002)). This method is based on a splitting technique proposed by Chorin (1968), related to Fractional Step Methods in which velocity and pressure operators are decoupled. The use of an artificial compressibility method is convenient to overcome computational difficulties when treating the conservation equation Nithiarasu (2003).

The formulation of a Lagrange-Galerkin Method, first introduced by Benque et al. (1980) involving Characteristic curves is covered in the second part of this work. This method is proved to be efficient for convection dominated problems as illustrated by Boukir et al. (1997), Achdou and Guermond (2000), Pironneau (1982) and is stated as a mixture of a traditional Galerkin Method for spatial variables with a Lagrangian formulation for time. This method is set as a mixed formulation in which, a penalty method to solve the system of equations introduces a small perturbation coefficient in the conservation equation, in order to be able to solve the matricial system, achieved after proposing a variational formulation for the problem.

This work provides another approach to the solution of Navier Stokes Equations, presenting the equations in discrete form in a simple manner, treating the temporal term of the equation with a finite different scheme and in contrast to the unstructured triangular grid formulations found in the literature, the use of a structured grid of square elements.

This document is organized as follows. In chapter 1 generalities of the Navier-Stokes Equation (section 1.1) and the concept of characteristic curves (section 1.2) are explained. In chapter 2 some features of the Characteristic Based Method (CBS) are reviewed. The Lagrange-Galerkin or characteristic curve method is introduced in chapter 3, and finally an example of the lid driven cavity flow is solved by the Lagrange-Galerkin procedure and discussed on sections 3.2 and 3.3 respectively.

Chapter 1

Preliminary Concepts

1.1 Navier Stokes Equation

The Navier Stokes equations describe the behavior of viscous fluids and are derived from the conservation laws, namely, the Newton's law of motion, the continuity equation and the energy equation. These coupled set of equations are Hyperbolic Partial Differential Equations of first or second order that describe time-dependent convection dominated problems. The law of momentum conservation states that *the rate of change of momentum of a particle equals the sum of external forces applied to it*, therefore it is important to mention that the deduction of the Navier Stokes equation is based on this law. The left hand side of equation 1.1 shows the rate of change of momentum , while the right hand side shows the sum of body forces, acting throughout the volume of a body (i.e gravitational forces), and contact forces, interacting with the fluid element throughout its surface (i.e viscous forces).

$$\underbrace{\rho\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right)}_{Change\ in\ momentum} = \underbrace{div(\sigma)}_{Surface\ force} + \underbrace{\rho b}_{Body\ force} . \tag{1.1}$$

In order to achieve the Navier Stokes set of equations, a constitutive relation for the stress (σ) must be defined and incorporated; for a Newtonian incompressible fluid ($div(u) = 0$) we have,

$$\sigma = 2\mu D - pI.$$

where $\boldsymbol{\sigma}$ is the stress, $\boldsymbol{\mu}$ is the dynamic viscosity, $\mathbf{D} = \frac{\mathbf{1}}{\mathbf{2}}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathbf{T}}\right)$ is the strain rate or deformation tensor, $\mathbf{p}$ is the pressure and $\mathbf{I}$ is the identity matrix.

The set of equations is now complete:

$$\begin{aligned} \rho\left(\frac{\partial u}{\partial t} + u\cdot\nabla u\right) &= \mu\nabla^2 u + (\mu+\lambda)\,\nabla\left(div(u)\right) - \nabla p + \rho b \\ \nabla\cdot u &= 0. \end{aligned} \tag{1.2}$$

For the case of incompressible fluids, density (ρ) is constant and the second term of 1.2 vanishes.

$$\underbrace{\rho\left(\underbrace{\frac{\partial u}{\partial t}}_{Local\ accel.} + \underbrace{u\cdot\nabla u}_{Convective\ accel.}\right)}_{Material\ derivative} = \underbrace{\mu\nabla^2 u}_{Viscous\ term} - \underbrace{\nabla p}_{Pressure\ term} + \underbrace{\rho b}_{Body\ force\ term}$$

$$\nabla\cdot u = 0. \tag{1.3}$$

In order to complete the Navier Stokes problem, the set of equations must be subjected to initial and boundary conditions. Initial conditions refer to the definition of initial values for the velocity and pressure fields, while boundary conditions are imposed on the boundary $\boldsymbol{\Gamma} = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ of an open fluid domain denoted by $\boldsymbol{\Omega}$.

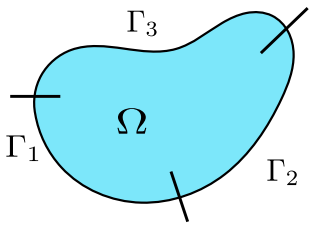


Figure 1.1: Open fluid domain Ω with its boundary $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$

Boundary conditions are of three types as follows,

$$\begin{aligned} Dirichlet \quad & u = g \quad on\ \Gamma_1 \\ Neumann \quad & h = [\sigma]n \quad on\ \Gamma_2 \\ Robin \quad & [\sigma]n = \alpha(u - u_{ref}) \quad on\ \Gamma_3 \end{aligned} \tag{1.4}$$

where g is a defined function or constant, h is the traction or surface force,α is a constant and u_{ref} is a reference velocity.

1.2 The Concept of Characteristic Curve

The scheme of a partial differential equation like the convective transport equation is:

$$\frac{\partial \phi}{\partial t} + a(x,t)\frac{\partial \phi}{\partial x} = Q(x,t) \tag{1.5}$$

where ϕ is the scalar quantity to be transported, a is the change of position in time and Q is the source term. The goal is to find a family of curves in the (x,t) plane in which the problem reduces to an ordinary differential equation. The information in the scalar quantity ϕ can propagate in time by means of the speed of propagation a in function of a point in time and space $a(x,t)$. The direction given by $a = \frac{dx}{dt}$ is the characteristic direction of propagation. If the PDE is linear, this speed of propagation is constant and the characteristic curves are fixed in the space-time plane (x,t). Furthermore, if the linear PDE has constant coefficients, the characteristic curves will be straight lines in plane (x,t) Donea and Huerta (2003).

One can observe that, after transforming the PDE to an homogeneous form, and assuming that the speed of propagation is constant, the equation can be expressed as:

$$\frac{\partial \phi}{\partial t} + a\frac{\partial \phi}{\partial x} = 0. \tag{1.6}$$

In this way we can express the equation 1.6 as $\frac{\partial \phi}{\partial t} = 0$ along curves given by $\frac{\partial x}{\partial t} = a$.

The solution to $\frac{\partial x}{\partial t} = a$ is $x = at + x'$ with x' being a constant of integration. The solution is a straight line that provides transport in time of the spatial profile given by ϕ through the characteristic curves. In a graphical way, the concept of transport by characteristics can be seen on Figure 1.2.

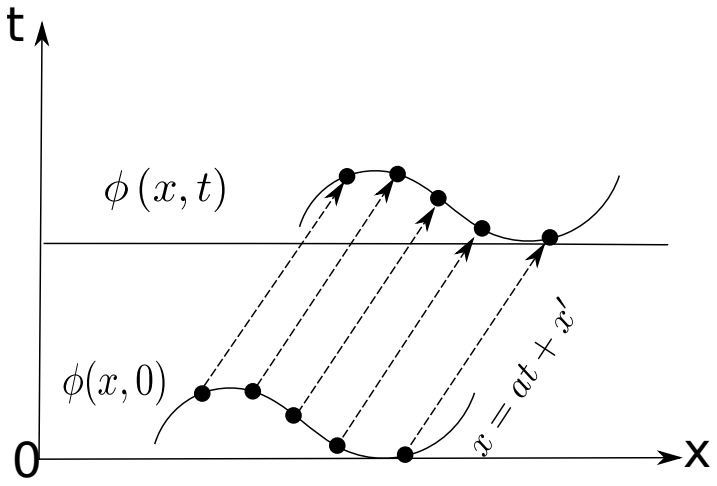


Figure 1.2: Transport of a quantity ϕ through characteristics.

ϕ is constant for the curve so along the characteristic curve $x - at = x'$,when introducing the initial condition $\phi(x,0) = f(x)$, the solution $\phi(x,t)$ in any instant of time t and any position x is

$$\phi(x,t) = f(x') = f(x - at) \tag{1.7}$$

The characteristic based method turns the convection-diffusion problem into a pure diffusion problem so that a spatial discretization procedure by standard Galerkin can be applied to the equation.

Chapter 2

Characteristic-based Split Method (CBS)

The characteristic based method is used to find a solution of fluid dynamics equations for any condition of the flow (compressible or incompressible) by means of a standard Galerkin procedure on characteristic curves. The method called CBS by Zienkiewicz et al. (2005) is able to eliminate the convective part of the equation of momentum conservation after setting it out in a Lagrangian fashion. Further discretization of the convection-diffusion equation will provide ideas for the treatment of Navier Stokes Equations.

2.1 Galerkin Based Characteristic Method

2.1.1 Convection-diffusion equation

After laying out the concept of characteristic curves, the solution of mass and momentum equations (Navier-Stokes Equations) for velocity and pressure variables (u and p) is considered. The main idea is to achieve the solution of these equations following a Galerkin procedure based on characteristics.

In this method, the convection-diffusion equation can be split in two parts, a convective term and a diffusive term that are discretized in a separate style. We have then the

convection-diffusion equation:

$$\underbrace{\frac{\partial \phi}{\partial t}}_{Time\ term} + \underbrace{\frac{\partial}{\partial x_j}(u_j \phi)}_{Convective\ term} - \underbrace{\frac{\partial}{\partial x_i}\left(k \frac{\partial \phi}{\partial x_i}\right)}_{Diffusive\ term} + \underbrace{Q}_{Source\ term} = 0. \tag{2.1}$$

The split is done according to (2.2) as

$$\phi = \phi^* + \phi^{**} \tag{2.2}$$

$$\frac{\partial \phi^*}{\partial t} + U \frac{\partial \phi}{\partial x} = 0 \qquad \frac{\partial \phi^{**}}{\partial t} - \frac{\partial}{\partial x}\left(k \frac{\partial \phi}{\partial x}\right) + Q = 0. \tag{2.3}$$

This split will be explained subsequently while discretizing the Navier Stokes Equations.

The idea of the characteristic based Galerkin method is to perform a time discretization for the convection-diffusion equation in moving coordinates (characteristics). This operation is done having in mind the elimination of the convective term and the use of a standard Galerkin procedure for the spatial discretization of remaining terms.

Taking the convection-diffusion equation on moving coordinates x':

$$\underbrace{\frac{\partial \phi(x')}{\partial t}}_{Time\ term} - \underbrace{\frac{\partial}{\partial x_i'}\left(k \frac{\partial \phi}{\partial x_i'}\right)}_{Diffusive\ term} + \underbrace{Q(x')}_{Source\ term} = 0. \tag{2.4}$$

Discretization of the time term is performed by setting out an approximation of the scalar quantity ϕ in two different states of time: present time ($t = t^n$), and a later time ($t = t^{n+1}$). The approximation uses an operator $\mathcal{L}$ acting on the scalar quantity to be transported on both time moments

$$\frac{\partial \phi}{\partial t} = \mathcal{L}(\phi) \tag{2.5}$$

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi^{n+1} - \phi^n}{\Delta t} = \mathcal{L}(\phi) \qquad \begin{cases} \mathcal{L}(\phi^{n+1}) & Unknown\ (Implicit) \\ \mathcal{L}(\phi^n) & Known\ (Explicit) \end{cases} \tag{2.6}$$

The approximation of the spatial, diffusive and source terms can be done following an implicit, semi-implicit or explicit scheme according to a parameter θ. The most used

scheme is the semi-implicit $\left(\theta = \frac{1}{2}\right)$, often called Crank-Nicolson.

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi^{n+1} - \phi^n}{\Delta t} = \theta \mathcal{L}(\phi^{n+1}) + (1-\theta)\mathcal{L}(\phi^n) \qquad \begin{cases} \theta = 1 & Implicit \\ \theta = \frac{1}{2} & Semi - implicit \\ \theta = 0 & Explicit \end{cases} \tag{2.7}$$

After the discretization process, an expression for the explicit convection-diffusion equation is achieved,

$$\begin{aligned} \phi^{n+1} - \phi^n = -\Delta t \left[\frac{\partial (U_j \phi)}{\partial x_j} - \frac{\partial}{\partial x_i}\left(k \frac{\partial \phi}{\partial x_i} \right) + Q \right]_{t=n} + \\ \frac{\Delta t^2}{2} U_k \frac{\partial}{\partial x_k} \left[\left(\frac{\partial (U_j \phi)}{\partial x_j} \right) - \frac{\partial}{\partial x_i}\left(k \frac{\partial \phi}{\partial x_i} \right) + Q \right]_{t=n}. \end{aligned} \tag{2.8}$$

The advantage of using this equation is that it is written in conservative form, and is explicit because it is evaluated at present time $t = n$.

2.1.2 Navier Stokes equation

In a similar fashion we can discretize the Navier–Stokes equations in time:

In the Navier Stokes Equations the quantity to be transported is the velocity $\phi = \rho u_i = U_i$. The convective term becomes the convective acceleration and the diffusive term is the divergence of the strain forces, which are in fact given by the gradient of the velocity. Finally, the pressure term is treated as a source term as well as the body forces. Hence, the Navier–Stokes equations represent the convection-diffusion of momentum per unit mass

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j}(u_j U_i) - \frac{\partial \tau_{ij}}{\partial x_j} + \underbrace{\frac{\partial p}{\partial x_i} - \rho g_i}_{Source\,term} = 0. \tag{2.9}$$

The scalar quantity is given by $U_i = \rho u_i$.

After discretization of the Navier–Stokes Equations followed by a Crank Nicholson

scheme with $\theta = \frac{1}{2}$, we can obtain an explicit equation for a time instant $t = n$

$$\begin{aligned}\Delta U_i = &- \Delta t \left[\frac{\partial}{\partial x_j}(u_j U_i) - \frac{\partial \tau_{ij}}{\partial x_i} + \frac{\partial p}{\partial x_i} - \rho g_i\right]^n \\ &+ \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left[\frac{\partial}{\partial x_j}(u_j U_i) - \frac{\partial \tau_{ij}}{\partial x_i} + \frac{\partial p}{\partial x_i} - \rho g_i\right]^n.\end{aligned} \tag{2.10}$$

The following step is to split the equation isolating the pressure terms and expressing the change of velocity ΔU_i as the sum of two auxiliary variables ΔU_i^* and ΔU_i^{**}, according to (2.11):

$$\begin{aligned}\Delta U_i = \quad & \Delta t \left[-\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_i} + \rho g_i\right]^n \qquad -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \\ & \underbrace{+\frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left[\frac{\partial}{\partial x_j}(u_j U_i) - \frac{\partial \tau_{ij}}{\partial x_i} - \rho g_i\right]^n}_{\Delta U_i^*} \quad \underbrace{+\frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right)}_{\Delta U_i^{**}}\end{aligned} \tag{2.11}$$

$$\Delta U_i = \Delta U_i^* + \Delta U_i^{**}. \tag{2.12}$$

The pressure terms that were separated are stored in the auxiliary variable ΔU_i^{**}, whereas remaining terms are stored in ΔU_i^*. Higher order terms in the stress tensor are neglected

$$\Delta U_i^* = \Delta t \left[-\frac{\partial (u_j U_i)}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k}\left(\frac{\partial (u_j U_i)}{\partial x_j} - \rho g_i\right)\right]^n. \tag{2.13}$$

$$\Delta U_i^{**} = -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i} \tag{2.14}$$

As there is an additional unknown in the system (the pressure) we need an additional equation to find it. Regarding the mass conservation equation, it is necessary to take it to an alternate form in order to discretize and solve it. Depending on the problem conditions, i.e. a compressible problem, density can be interpolated. Let us use the speed of sound denoted by c

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \boldsymbol{u}) = 0 \tag{2.15}$$

and also

$$\frac{1}{c^2}\frac{\partial p}{\partial t} = -\operatorname{div}(\rho \boldsymbol{u}).$$

Therefore,

$$\frac{\partial \rho}{\partial t} = \frac{1}{c^2}\frac{\partial p}{\partial t} = -\operatorname{div}(\rho \boldsymbol{u}) = -\frac{\partial U_i}{\partial x_i}$$

Using a θ time approximation for the conservation equation

$$\frac{\Delta \rho}{\Delta t} = \left(\frac{1}{c^2}\right)^n \frac{\Delta p}{\Delta t} = -\frac{\partial U_i^{n+\theta_1}}{\partial x_i}, \tag{2.16}$$

which can be expanded as

$$\left(\frac{1}{c^2}\right)^n \frac{\Delta p}{\Delta t} = -\frac{\partial U_i^{n+\theta_1}}{\partial x_i} = -\left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i}{\partial x_i}\right],$$

and splitting the velocity

$$\left(\frac{1}{c^2}\right)^n \frac{\Delta p}{\Delta t} = -\left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial}{\partial x_i}(\Delta U_i^* + \Delta U_i^{**})\right].$$

Using the equation for ΔU^{**} (2.14)

$$\left(\frac{1}{c^2}\right)^n \frac{\Delta p}{\Delta t} = -\left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial}{\partial x_i}\left(\Delta U_i^* + -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i}\right)\right].$$

Since the last term is of superior order, and expanding $p^{n+\theta_2} = p^n + \theta_2 \Delta p$,

$$\left(\frac{1}{c^2}\right)^n \frac{\Delta p}{\Delta t} = -\left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} + -\Delta t \theta_1 \frac{\partial}{\partial x_i}\left(\frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i}\right)\right]. \tag{2.17}$$

The method finishes with the solution of the equations after a Galerkin procedure in the following order:

1. Solve (2.13) to obtain ΔU_i^*.

2. Solve (2.17) to obtain Δp o $\Delta \rho$.

3. Solve (2.14) to establish values for velocity and pressure in t^{n+1}.

If a compressible problem is treated, it becomes necessary to get the energy equation solved.

2.2 Spatial discretization

2.2.1 Step 1

The time discretization process through the characteristic based Galerkin method provides a solution procedure for which it is necessary to follow a spatial discretization of equations using the standard Galerkin method in order to achieve a matricial system of equations. The process starts with the variational treatment of equation 2.13 to be taken to its weak form

$$\Delta U_i^* = \Delta t \left[-\frac{\partial (u_j U_i)}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial (u_j U_i)}{\partial x_j} - \rho g_i \right) \right]^n .$$

The equation can be expressed in vector notation as follows:

$$\Delta \boldsymbol{U}^* = \Delta t \left[-\operatorname{div}(\boldsymbol{u} \otimes \boldsymbol{U}) + \operatorname{div}(\boldsymbol{\tau}) + \boldsymbol{b} + \frac{\Delta t}{2} \left[\nabla (\operatorname{div} (\boldsymbol{u} \otimes \boldsymbol{U}) - \boldsymbol{b}) \right] \boldsymbol{u} \right]^n \tag{2.18}$$

Both sides of the equation are multiplied by an arbitrary function $\boldsymbol{v}$, with the same features and from the same space of $\boldsymbol{u}$, and integrated over the whole domain of analysis Ω.

$$\begin{aligned} \int_\Omega \underbrace{\Delta \boldsymbol{U}^* \cdot \boldsymbol{v}}_{1} = \int_\Omega \Delta t \Bigg[& - \underbrace{\operatorname{div}(\boldsymbol{u} \otimes \boldsymbol{U})}_{2} + \underbrace{\operatorname{div}(\boldsymbol{\tau})}_{3} + \underbrace{\boldsymbol{b}}_{4} \\ & + \frac{\Delta t}{2} \left(\underbrace{\nabla (\operatorname{div}(\boldsymbol{u} \otimes \boldsymbol{U}) - \boldsymbol{b}) \boldsymbol{u}}_{5} \right)^n \Bigg] \cdot \boldsymbol{v}. \end{aligned} \tag{2.19}$$

The suitable form of discretizing the equation is taking each one of the terms separately to the matricial form. Before starting the discretization of each term, it is necessary to define the way the functions U and v are approximated. Functions are approximated multiplying the base functions of the element with the variables at each node . The initial velocity vector u will be interpolated in the same manner as the unknown velocity vector

U

$$U_i(x) = \sum_b \widetilde{U}_i^b N^b(x) \qquad v_i(x) = \sum_a \widetilde{v}_i^a N^a(x) \tag{2.20}$$

with a, b: node number and i: dimension index. $\boldsymbol{u}$ can be expressed in vector notation as

$$\boldsymbol{u} = \sum_{i\ b} \widetilde{u}_i^b N^b(x) \hat{\boldsymbol{e}}_i \tag{2.21}$$

which we will call index approximation of the vector field. And the compact notation which requires to be written as

$$\boldsymbol{N} = [N^1, N^2, \ldots, N^n], \qquad \tilde{\boldsymbol{u}}_i = \begin{bmatrix} u_i^1 \\ \vdots \\ u_i^n \end{bmatrix},$$

where n is the number of nodes. Then

$$u_i = \boldsymbol{N}\tilde{\boldsymbol{u}}_i \qquad \text{and} \qquad \boldsymbol{u} = \boldsymbol{N}\tilde{\boldsymbol{u}}$$

where $\tilde{\boldsymbol{u}}$ is the matrix, form by the nodal values in the $1, 2$ and 3 directions

$$\tilde{\boldsymbol{u}} = \boldsymbol{N}[\tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{u}}_2 \tilde{\boldsymbol{u}}_3] = \boldsymbol{N} \begin{bmatrix} u_1^1 & u_2^1 & u_3^1 \\ \vdots & \vdots & \vdots \\ u_1^n & u_2^n & u_3^n \end{bmatrix}$$

As said before, we must start the process with the first term.

1. $\int_\Omega \Delta U_i^* \cdot v_i$

$$\int_\Omega \Delta U_i^* \cdot v_i \approx \int_\Omega N^b \Delta \widetilde{U}_i^{b*} N^a \widetilde{v}_i^a = \widetilde{v}_i^a \Delta \widetilde{U}_i^{b*} \int_\Omega N^a N^b$$

The product of both basis function vectors gives the mass matrix of the equation, M^{ab}

$$\widetilde{v}_i^a \, M^{ab} \Delta \widetilde{U}_i^{b*} = \left\langle \widetilde{\boldsymbol{v}}_i \, , \, \boldsymbol{M} \Delta \widetilde{\boldsymbol{U}}_i^* \right\rangle .$$

2. $\int_\Omega \operatorname{div}(\boldsymbol{u}\otimes\boldsymbol{U})\cdot\boldsymbol{v}$

   Let us continue with the discretization of the second term denoted by 2 in (2.19):

$$\int_\Omega \operatorname{div}(\boldsymbol{u}\otimes\boldsymbol{U})\cdot\boldsymbol{v}.$$

$$\begin{aligned}\int_\Omega \frac{\partial}{\partial x_j}(u_jU_i)v_i &\approx \int_\Omega \frac{\partial}{\partial x_j}(u_j\widetilde{U}_i^bN^b)\widetilde{v}_i^aN^a\\ &= \widetilde{v}_i^a\widetilde{U}_i^b\underbrace{\int_\Omega \nabla(uN^b)N^a}_{C^{ab}}\\ &= \widetilde{v}_i^a\,C^{ab}\widetilde{U}_i^b\\ &= \left\langle \widetilde{\boldsymbol{v}}_i\,,\,\boldsymbol{C}\widetilde{\boldsymbol{U}}_i\right\rangle.\end{aligned}$$

3. $\int_\Omega \operatorname{div}(\boldsymbol{\tau})\cdot\boldsymbol{v}$

   The third term is the divergence of the deviatoric stress tensor $\boldsymbol{\tau}$. We must begin integrating by parts the inner product of the divergence tensor with the arbitrary function $\boldsymbol{v}$

$$\int_\Omega \frac{\partial\tau_{ij}}{\partial x_j}v_i = \int_\Omega \operatorname{div}(\boldsymbol{\tau})\cdot\boldsymbol{v} = \int_\Omega \operatorname{div}(\boldsymbol{\tau}_i)v_i$$

   where $\boldsymbol{\tau}_i = [\tau_{i1},\tau_{i2},\tau_{i3}]$, is the ith row of the stress tensor. Using the divergence theorem [1]

$$\int_\Omega \operatorname{div}(\boldsymbol{\tau}_i)v_i = -\int_\Omega \langle\boldsymbol{\tau}_i,\nabla v_i\rangle + \int_\Gamma \langle\boldsymbol{\tau}_iv_i,\boldsymbol{n}\rangle.$$

[1] Given $\boldsymbol{F}$ a vector function and g a scalar function, the divergence theorem stands

$$\int_\Omega \operatorname{div}(\boldsymbol{F}g) = \int_\Omega \left(\operatorname{div}(\boldsymbol{F})\;g + \langle\boldsymbol{F},\nabla g\rangle\right) = \int_\Gamma \langle\boldsymbol{F}g,n\rangle.$$

The same result can be obtained using index notation and integration by parts

$$\begin{aligned}\int_\Omega \frac{\partial \tau_{ij}}{\partial x_j} v_i &= -\int_\Omega \tau_{ij} \frac{\partial v_i}{\partial x_j} + \int_\Gamma \tau_{ij} v_i n_j \\ &= -\underbrace{\int_\Omega \nabla \boldsymbol{v} \,:\, \boldsymbol{\tau}}_{a} + \underbrace{\int_\Gamma \langle \boldsymbol{t}, \boldsymbol{v} \rangle}_{b}\end{aligned}$$

where $t_i = \tau_{ij} n_j$ is the traction on the surface

After integration we get two terms, the first one (a) is integrated over the domain and contains a Frobenius inner product of matrices often called double dot product, whereas the other (b) is simply evaluated at the boundary and contains the traction or product between the stress tensor and a normal vector on the surface.

$$A : B = \sum_{ij} A_{ij} B_{ij}$$

a. First, the term integrated over the domain is discretized:

For a Newtonian fluid the following relationship is true,

$$\tau = 2\mu \left(\boldsymbol{\epsilon} - \frac{1}{3} \operatorname{div}(\boldsymbol{\epsilon}) I \right) \qquad \text{with} \qquad \boldsymbol{\epsilon} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T),$$

where μ is the dynamic viscosity, $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}(u)$ is the strain rate and I is the identity matrix. It should be noted that $\boldsymbol{\epsilon}$ and $\boldsymbol{\tau}$ are linear functionals. That is, given $\boldsymbol{u}$ and $\boldsymbol{v}$ in V a vector space of admissible velocities, and given $\alpha \in \mathbb{R}$, then

$$\begin{aligned}&(i) \quad \boldsymbol{\epsilon}(\boldsymbol{u} + \boldsymbol{v}) = \boldsymbol{\epsilon}(\boldsymbol{u}) + \boldsymbol{\epsilon}(\boldsymbol{v}) \\ &(ii) \quad \boldsymbol{\epsilon}(\alpha \boldsymbol{v}) = \alpha \boldsymbol{\epsilon}(\boldsymbol{u}).\end{aligned}$$

The same holds for the divergence and grad operators

$$\begin{aligned}\operatorname{div}(\alpha \boldsymbol{u} + \beta \boldsymbol{v}) &= \alpha \operatorname{div}(\boldsymbol{u}) + \beta \operatorname{div}(\boldsymbol{v}) \\ \nabla(\alpha \boldsymbol{u} + \beta \boldsymbol{v}) &= \alpha \nabla(\boldsymbol{u}) + \beta \nabla(\boldsymbol{v}).\end{aligned}$$

Using the linearity of these operators we can rewrite the strain rate, the divergence and the the stress in terms of the base, (2.21). For the strain rate we

have

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \boldsymbol{\epsilon}(u_i^b N_b \boldsymbol{e}_i) = u_i^b \boldsymbol{\epsilon}(N_b \boldsymbol{e}_i).$$

By definition,

$$\epsilon(\boldsymbol{u})_{rs} = \frac{1}{2}\left(u_{r,s} + u_{r,s}\right),$$

so in the same manner

$$\begin{aligned}\epsilon(N_b \boldsymbol{e}_i)_{rs} &= \frac{1}{2}\left((N_b \boldsymbol{e}_i)_{r,s} + (N_b \boldsymbol{e}_i)_{r,s}\right) \\ &= \frac{1}{2}\left(N_{b,s}\delta_{ir} + N_{b,r}\delta_{is}\right).\end{aligned}$$

Therefore,

$$\epsilon(\boldsymbol{u})_{rs} = u_i^b \epsilon(N_b \boldsymbol{e}_i)_{rs} = \frac{u_i^b}{2}\left(N_{b,s}\delta_{ir} + N_{b,r}\delta_{is}\right). \tag{2.22}$$

In the divergence case

$$\operatorname{div}(\boldsymbol{u}) = u_{,kk},$$

therefore

$$\operatorname{div}(u_i^b N_b \boldsymbol{e}_i) = u_i^b (N_b \boldsymbol{e}_i)_{,kk} = u_i^b N_{b,k}\delta_{ik}, \tag{2.23}$$

and in the gradient case

$$\nabla(\boldsymbol{v}) = v_{r,s},$$

therefore

$$\nabla(v_j^a N_a \boldsymbol{e}_j) = (v_j^a N_a \boldsymbol{e}_j)_{r,s} = v_j^a N_{a,s}\delta_{jr}. \tag{2.24}$$

Merging (2.22), (2.23) (2.24), and using index notation

$$\begin{aligned}\boldsymbol{\sigma} : \nabla \boldsymbol{v} &= 2\mu\left(\boldsymbol{\epsilon}(\boldsymbol{u}) - \frac{1}{3}\operatorname{div}(\boldsymbol{\epsilon})I\right) : \nabla \boldsymbol{v} \\ &= 2\mu\left(\epsilon(\boldsymbol{u})_{rs} - \frac{1}{3}u_{kk}\delta_{rs}\right) v_{r,s} \\ &= \mu u_i^b \left(\left(N_{b,s}\delta_{ir} + N_{b,r}\delta_{is}\right) - \frac{2}{3}N_{b,k}\delta_{ik}\delta_{rs}\right) v_i^a N_{a,s}\delta_{jr}.\end{aligned}$$

Now, this expresion must be integrated over the domain, so it can be expressed

in a compact form as

$$\int_\Omega \boldsymbol{\sigma} : \nabla \boldsymbol{v} = u_i^b v_i^a K_{ji}^{ab}, \tag{2.25}$$

with

$$K_{ij}^{ab} = \int_\Omega \left((N_{b,s}\delta_{ir} + N_{b,r}\delta_{is}) - \frac{2}{3} N_{b,k}\delta_{ik}\delta_{rs} \right) N_{a,s}\delta_{jr}. \tag{2.26}$$

Combining two indexes in one yields

$$\alpha = 2(a-1) + j$$
$$\beta = 2(b-1+i).$$

We can rewrite the expression above in simple matrix notation as

$$\langle \boldsymbol{v}, \boldsymbol{K}\boldsymbol{u} \rangle .$$

b. The term integrated over the boundary Γ contains the traction and is discretized as follows:

$$\begin{aligned} \int_\Gamma \boldsymbol{t} \cdot \boldsymbol{v} &\approx \int_\Gamma t_i N^a \widetilde{v}_i^a \\ &= \widetilde{v}_i^a \int_\Gamma t_i N^a \\ &= \widetilde{v}_i^a \overset{2}{f}{}_i^a \end{aligned}$$

where

$$\int_\Gamma t_i N^a = \overset{2}{f}{}_i^a \tag{2.27}$$

in compact form, and combining subindices the last product can be interpreted as a dot product. The final discrete expression for the term is

$$\int_\Omega \operatorname{div}(\boldsymbol{\tau}) \cdot \boldsymbol{v} \to \langle \boldsymbol{v}, \boldsymbol{K}\boldsymbol{u} \rangle + \left\langle \boldsymbol{v}, \overset{2}{\boldsymbol{f}} \right\rangle \tag{2.28}$$

4. $\int_\Omega \boldsymbol{b} \cdot \boldsymbol{v}$

The fourth term related to body forces, is discretized assuming constant density for

the element

$$\begin{aligned}\int_\Omega \boldsymbol{b}\cdot\boldsymbol{v} &\approx \int_\Omega \rho g_i N^a v_i^a \\ &= v_i^a \int_\Omega \rho g_i N^a \\ &= \widetilde{v}_i^a \ \overset{1}{f}_i^a \\ &= \left\langle \boldsymbol{v}, \overset{1}{\boldsymbol{f}} \right\rangle\end{aligned}$$

where

$$\int_\Omega \rho g_i N^a = \overset{1}{f}_i^a . \tag{2.29}$$

5. $\int_\Omega \nabla(\mathrm{div}(\boldsymbol{u}\otimes\boldsymbol{U}) - \boldsymbol{b})\boldsymbol{u}\cdot\boldsymbol{v}$

The next term to discretize is the stabilizing term denoted by number 5. It should be mentioned that this term has two components: a matrix and a vector. To ease the process, we may take advantage of alternative ways to face up the reduction of this term by using the Frobenius inner product of matrices and integration by parts (First Green identity).

This term can now be discretized following this procedure:

$$\int_\Omega [\nabla \underbrace{(\mathrm{div}\,(\boldsymbol{u}\otimes\boldsymbol{U}) - \boldsymbol{b})}_{\alpha_{ij}}\boldsymbol{u}]\cdot v \tag{2.30}$$

$$\int_\Omega [\nabla\alpha]_{ij} u_j v_i \equiv \int_\Omega [\nabla\alpha] : (v\otimes u) \equiv \int_\Omega \left(\frac{\partial \alpha_i}{\partial x_j}\right) v_i u_j .$$

This term must be integrated by parts:

$$\begin{aligned}\int_\Omega \left(\frac{\partial \alpha_i}{\partial x_j}\right) v_i u_j &= -\int_\Omega \alpha_i \left(\frac{\partial v_i u_j}{\partial x_j}\right) + \int_\Gamma \alpha_i (v_i u_j) n_j \\ &= -\int_\Omega \langle \alpha, \mathrm{div}(v\otimes u)\rangle .\end{aligned}$$

Now, we can replace α for its original expression from (2.30). Operating the inner product, the idea is to discretize resulting terms separately:

$$\int_\Omega \langle \text{div}\,(\boldsymbol{u}\otimes\boldsymbol{U}) - \boldsymbol{b}, \text{div}(\boldsymbol{v}\otimes\boldsymbol{u})\rangle$$

$$\int_\Omega \underbrace{\langle \text{div}\,(\boldsymbol{u}\otimes\boldsymbol{U}), \text{div}(\boldsymbol{v}\otimes\boldsymbol{u})\rangle}_{5.1} \qquad - \int_\Omega \underbrace{\langle \boldsymbol{b}, \text{div}(\boldsymbol{v}\otimes\boldsymbol{u})\rangle}_{5.2}.$$

Let us start with the first integral:

5.1

$$\int_\Omega \langle \text{div}\,(\boldsymbol{u}\otimes\boldsymbol{U}), \text{div}(\boldsymbol{v}\otimes\boldsymbol{u})\rangle \;=\; \int_\Omega \frac{\partial u_j U_i}{\partial x_j}\frac{\partial v_i u_k}{\partial x_k}$$

$$\begin{aligned}
&\approx \int_\Omega \frac{\partial(u_j N^b \widetilde{U}_i^b)}{\partial x_j}\frac{\partial(N^a \widetilde{v}_i^a u_k)}{\partial x_k} \\
&\approx \widetilde{v}_i^a \widetilde{U}_i^b \int_\Omega \frac{\partial(u_j N^b)}{\partial x_j}\frac{\partial(u_k N^a)}{\partial x_k} \\
&\approx \widetilde{v}_i^a \widetilde{U}_i^b \int_\Omega \underbrace{\nabla(uN^b)\nabla(uN^a)}_{K_s^{ab}} \\
&\approx \left\langle \widetilde{v}_i^a \,,\, K_s^{ab}\widetilde{U}_i^b \right\rangle.
\end{aligned}$$

The second integral results in a vector:

5.2

$$\begin{aligned}\int_\Omega \langle \boldsymbol{b}, \operatorname{div}(\boldsymbol{v}\otimes\boldsymbol{u})\rangle &= \int_\Omega \rho g_i \frac{\partial(v_i u_k)}{\partial x_k} \\ &\approx \int_\Omega \rho g_i \frac{\partial(N^a \widetilde{v_i^a} u_k)}{\partial x_k} \\ &\approx \widetilde{v_i^a}\int_\Omega \rho g_i \frac{\partial(N^a u_k)}{\partial x_k} \\ &\approx \widetilde{v_i^a}\int_\Omega \underbrace{\rho g_i \nabla(u N^a)}_{f_s^a} \\ &\approx \langle \widetilde{v_i^a}\ ,\ f_s^a\rangle\,.\end{aligned}$$

At this point, each of the terms of (2.19) is discretized so we can replace the compact form into the equation:

$$\left\langle \widetilde{v}^a\ ,\ M^{ab}\Delta\widetilde{U}_i^{b*}\right\rangle =$$
$$\Delta t\left[-\left\langle \widetilde{v}_i^a\ ,\ C^{ab}\widetilde{U}_i^b\right\rangle - \langle \widetilde{v}_i^a\ ,\ K_\tau^{ab}\widetilde{u}_i^b\rangle + \langle \widetilde{v}_i^a\ ,\ f\rangle + \tfrac{\Delta t}{2}\left\langle \widetilde{v}_i^a\ ,\ K_s^{ab}\widetilde{U}_i^b\right\rangle - \langle \widetilde{v}_i^a\ ,\ f_s^a\rangle\right].$$

Since v is an arbitrary function, we can express the system of equations for $\Delta\widetilde{U}_i^*$ as:

$$M^{ab}\Delta\widetilde{U}_i^{b*} = \Delta t\left[-C^{ab}\widetilde{U}_i^b - K_\tau^{ab}\widetilde{u}_i^b + f + \frac{\Delta t}{2}K_s^{ab}\widetilde{U}_i^b - \frac{\Delta t}{2}f_s^a\right]$$

$$\Delta\widetilde{U}_i^{b*} = -\frac{1}{M^{ab}}\Delta t\left[C^{ab}\widetilde{U}_i^b + K_\tau^{ab}\widetilde{u}_i^b - f - \frac{\Delta t}{2}\left(K_s^{ab}\widetilde{U}_i^b + f_s^a\right)\right].$$

2.2.2 Step 2

Discretization of step 2 is done in a similar way as Step 1. Let us recall the pressure equation

$$\Delta\rho = \left(\frac{1}{c^2}\right)_{t=n}\Delta p = -\Delta t\left[\frac{\partial U_i}{\partial x_i} + \theta_1\frac{\partial\Delta U_i^*}{\partial x_i} - \Delta t\theta_1\left(\frac{\partial^2 p^n}{\partial x_i\partial x_i} + \theta_2\frac{\partial^2\Delta p}{\partial x_i\partial x_i}\right)\right].$$

The goal of step 2 is to find the value of Δp after approximating velocity and pressure as shown in (2.20):

$$\Delta\rho = \left(\frac{1}{c^2}\right)_{t=n} \Delta p = -\Delta t \left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t \theta_1 \frac{\partial}{\partial x_i}\left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right)\right].$$

Multiplying this equation by an arbitrary function v, and integrating over the domain Ω

$$\left(\frac{1}{c^2}\right)_{t=n} \Delta p \cdot v = -\Delta t \left[\frac{\partial U_i}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t \theta_1 \frac{\partial}{\partial x_i}\left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right)\right] \cdot v$$

$$\left(\frac{1}{c^2}\right)_{t=n} \Delta p \cdot N^a v_i^a = -\Delta t \int_\Omega \frac{\partial}{\partial x_i} \left[U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right)\right] \cdot N_p^a v_i^a$$

$$\left(\frac{1}{c^2}\right)_{t=n} \Delta p \cdot N_p^a v_i^a = -\Delta t \int_\Omega N_p^a \frac{\partial}{\partial x_i} \left[U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right)\right] \cdot v_i^a$$

$$\left(\frac{1}{c^2}\right)_{t=n} \Delta p \cdot N_p^a v_i^a = -v_i^a \Delta t \int_\Omega \underbrace{N_p^a}_{F} \underbrace{\frac{\partial}{\partial x_i} \left[U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right)\right]}_{G}.$$

The integration by parts process recalls the equation:

$$\int_\Omega \left(\frac{\partial F}{\partial x_i} G \; + \; F \frac{\partial G}{\partial x_i}\right) d\Omega = \int_\Gamma F G n_i d\Gamma$$

Then, integrating by parts the right side of (2.2.2), we obtain two terms, one over the domain and other over the surface:

$$\left(\frac{1}{c^2}\right)_{t=n} \Delta p \cdot N_p^a v_i^a =$$
$$v_i^a \left\{\Delta t \int_\Omega \frac{\partial N_p^a}{\partial x_i} \left[U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right) - \int_\Gamma N_p^a \left(U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \left(\frac{\partial p^{n+\theta_2}}{\partial x_i}\right) n_i\right)\right]\right\}$$

To simplify the treatment of terms over the domain, let us multiply each term in brackets and rewrite the boundary term as f_p:

$$\left(\frac{1}{c^2}\right)_{t=n} \Delta p \cdot N_p^a v_i^a =$$
$$v_i^a \left[\Delta t \int_\Omega \frac{\partial N_p^a}{\partial x_i} U_i + \theta_1 \Delta t \int_\Omega \frac{\partial N_p^a}{\partial x_i} \Delta U_i^* - \Delta t^2 \theta_1 \int_\Omega \frac{\partial N_p^a}{\partial x_i} \left(\frac{\partial p^{n+\theta_2}}{\partial x_i} \right) \right.$$
$$\left. - \underbrace{\Delta t \int_\Gamma N_p^a \left(U_i + \theta_1 \Delta U_i^* - \Delta t \theta_1 \left(\frac{\partial p^{n+\theta_2}}{\partial x_i} \right) n_i \right)}_{f_p} \right]$$

Interpolation of velocity U gives:

$$\left(\frac{1}{c^2}\right)_{t=n} \Delta p \cdot N_p^a v_i^a =$$
$$v_i^a \left[\Delta t \int_\Omega \frac{\partial N_p^a}{\partial x_i} N^b \widetilde{U}_i^b + \theta_1 \Delta t \int_\Omega \frac{\partial N_p^a}{\partial x_i} N^b \Delta \widetilde{U}_i^{b*} - \Delta t^2 \theta_1 \int_\Omega \frac{\partial N_p^a}{\partial x_i} \left(\frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \right) - f_p \right].$$

Interpolation of pressure p gives:

$$\left(\frac{1}{c^2}\right)_{t=n} N_p^b \widetilde{\Delta p} \cdot N_p^a v_i^a =$$
$$v_i^a \left[\Delta t \int_\Omega \frac{\partial N_p^a}{\partial x_i} N^b \widetilde{U}_i^b + \theta_1 \Delta t \int_\Omega \frac{\partial N_p^a}{\partial x_i} N^b \Delta \widetilde{U}_i^{b*} - \Delta t^2 \theta_1 \int_\Omega \frac{\partial N_p^a}{\partial x_i} \left(\frac{\partial (N_p^b \widetilde{p}^n)}{\partial x_i} \right) - \Delta t^2 \theta_1 \theta_2 \int_\Omega \frac{\partial N_p^a}{\partial x_i} \left(\frac{\partial (N_p^b \widetilde{\Delta p})}{\partial x_i} \right) - f_p \right].$$

Now, it is necessary to arrange together the terms for $\widetilde{\Delta p}$ and $\widetilde{p}$:

$$\left[\underbrace{\left(\frac{1}{c^2}\right)_{t=n} N_p^b N_p^a}_{M^{ab}} + \Delta t^2 \theta_1 \theta_2 \underbrace{\int_\Omega \frac{\partial N_p^a}{\partial x_i}\left(\frac{\partial N_p^b}{\partial x_i}\right)}_{H^{ab}}\right] \widetilde{\Delta p} v_i^a =$$

$$v_i^a \Delta t \left[\int_\Omega \underbrace{\frac{\partial N_p^a}{\partial x_i} N^b}_{G^{ab}} \widetilde{U}_i^b + \theta_1 \int_\Omega \underbrace{\frac{\partial N_p^a}{\partial x_i} N^b}_{G^{ab}} \Delta \widetilde{U}_i^{b*} - \Delta t \theta_1 \int_\Omega \underbrace{\frac{\partial N_p^a}{\partial x_i}\left(\frac{\partial N_p^b}{\partial x_i}\right)}_{H^{ab}} \widetilde{p}^n - f_p\right].$$

Due to the features of the arbitrary function v, we can remove it from the equation, obtaining a system of equations for $\widetilde{\Delta p}$

$$\left[M^{ab} + \Delta t^2 \theta_1 \theta_2 H^{ab}\right] \widetilde{\Delta p} = \Delta t \left[G^{ab} \widetilde{U}_i^b + \theta_1 G^{ab} \Delta \widetilde{U}_i^{b*} - \Delta t \theta_1 H^{ab} \widetilde{p}^n - f_p\right].$$

2.2.3 Step 3

After calculating ΔU_i^* and Δp, it is time to calculate the correction of velocity given in (2.14):

$$\Delta U_i^{**} = -\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i}.$$

Let us remember the split proposed at the beginning of the section:

$$\Delta U_i = \Delta U_i^* + \Delta U_i^{**}.$$

Then, we must accomplish a new discretization process to obtain values for ΔU_i. Multiplying the equation by an arbitrary function v following Galerkin's procedure,

$$\Delta U_i \cdot v = \Delta U_i^* \cdot v + \Delta U_i^{**} \cdot v$$

$$\Delta U_i \cdot v = \Delta U_i^* \cdot v + \left(-\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i}\right) \cdot v.$$

Approximating v results in

$$\int_\Omega \Delta U_i \cdot N^a \widetilde{v}_i^a = \int_\Omega \Delta U_i^* \cdot N^a \widetilde{v}_i^a + \int_\Omega \left(-\Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} + \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i} \right) \cdot N^a \widetilde{v}_i^a .$$

Now, interpolation of ΔU_i and ΔU_i^* yields:

$$\int_\Omega N^b \Delta \widetilde{U}_i^b \cdot N^a \widetilde{v}_i^a = \int_\Omega N^b \Delta \widetilde{U}_i^{b*} \cdot N^a \widetilde{v}_i^a - \int_\Omega \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} N^a \widetilde{v}_i^a + \int_\Omega \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i} N^a \widetilde{v}_i^a$$

We can relate both terms for velocity, the auxiliary term ΔU^*, and the change of velocity ΔU, and then take out nodal expressions from the integrals:

$$\int_\Omega N^b \Delta \widetilde{U}_i^b \cdot N^a \widetilde{v}_i^a - N^b \Delta \widetilde{U}_i^{b*} \cdot N^a \widetilde{v}_i^a = - \int_\Omega \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} N^a \widetilde{v}_i^a + \int_\Omega \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i} N^a \widetilde{v}_i^a$$

$$\int_\Omega \left(\Delta \widetilde{U}_i^b - \Delta \widetilde{U}_i^{b*} \right) N^a N^b \widetilde{v}_i^a = - \int_\Omega \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} N^a \widetilde{v}_i^a + \int_\Omega \frac{\Delta t^2}{2} u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i} N^a \widetilde{v}_i^a$$

$$\left(\Delta \widetilde{U}_i^b - \Delta \widetilde{U}_i^{b*} \right) \widetilde{v}_i^a \int_\Omega N^a N^b = -\Delta t \int_\Omega \frac{\partial p^{n+\theta_2}}{\partial x_i} N^a \widetilde{v}_i^a + \frac{\Delta t^2}{2} \int_\Omega u_k \frac{\partial^2 p^n}{\partial x_k \partial x_i} N^a \widetilde{v}_i^a .$$

It is time to interpolate the pressure terms and operate them in order to obtain a matricial form:

$$\begin{aligned} \left(\Delta \widetilde{U}_i^b - \Delta \widetilde{U}_i^{b*} \right) \widetilde{v}_i^a \int_\Omega N^a N^b = & - \Delta t \int_\Omega \left(\frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \right) N^a \widetilde{v}_i^a \\ & + \frac{\Delta t^2}{2} \int_\Omega u_k \frac{\partial}{\partial x_k} \left(\frac{\partial p^n}{\partial x_i} \right) N^a \widetilde{v}_i^a . \end{aligned}$$

Interpolating pressure terms

$$\left(\Delta\widetilde{U}_i^b - \Delta\widetilde{U}_i^{b*}\right)\widetilde{v}_i^a \int_\Omega N^a N^b = -\Delta t \int_\Omega \left(\frac{\partial(N_p^b \widetilde{p}^n)}{\partial x_i} + \theta_2 \frac{\partial(N_p^b \widetilde{\Delta p})}{\partial x_i}\right) N^a \widetilde{v}_i^a + \frac{\Delta t^2}{2} \int_\Omega u_k \frac{\partial N^a}{\partial x_k} \left(\frac{\partial(N_p^b \widetilde{p}^n)}{\partial x_i}\right) \widetilde{v}_i^a$$

and expanding

$$\left(\Delta\widetilde{U}_i^b - \Delta\widetilde{U}_i^{b*}\right)\widetilde{v}_i^a \int_\Omega N^a N^b = -\Delta t \widetilde{v}_i^a \widetilde{p}^n \int_\Omega N^a \frac{\partial N_p^b}{\partial x_i} - \Delta t \widetilde{v}_i^a \theta_2 \widetilde{\Delta p} \int_\Omega N^a \frac{\partial N_p^b}{\partial x_i} + \frac{\Delta t^2}{2} \widetilde{v}_i^a \widetilde{p}^n \int_\Omega u_k \frac{\partial N^a}{\partial x_k} \left(\frac{\partial N_p^b}{\partial x_i}\right).$$

Notice that some terms are equal to those achieved in Step 2

$$\left(\Delta\widetilde{U}_i^b - \Delta\widetilde{U}_i^{b*}\right)\widetilde{v}_i^a \int_\Omega \underbrace{N^a N^b}_{M^{ab}} = -\Delta t \widetilde{v}_i^a \widetilde{p}^n \int_\Omega \underbrace{N^a \frac{\partial N_p^b}{\partial x_i}}_{G^{ab}} - \Delta t \widetilde{v}_i^a \theta_2 \widetilde{\Delta p} \int_\Omega \underbrace{N^a \frac{\partial N_p^b}{\partial x_i}}_{G^{ab}} + \frac{\Delta t^2}{2} \widetilde{v}_i^a \widetilde{p}^n \int_\Omega \underbrace{\frac{\partial(u_k N^a)}{\partial x_k} \left(\frac{\partial N_p^b}{\partial x_i}\right)}_{P^{ab}}.$$

Rewriting by using the matrices in curl brackets, and factoring:

$$\left(\Delta\widetilde{U}_i^b - \Delta\widetilde{U}_i^{b*}\right)\widetilde{v}_i^a M^{ab} = -\Delta t \widetilde{v}_i^a \left[G^{ab}\left(\widetilde{p}^n + \theta_2 \widetilde{\Delta p}\right) + \frac{\Delta t}{2} P^{ab} \widetilde{p}^n\right].$$

As in the other steps, the arbitrary function v can be removed:

$$\left(\Delta\widetilde{U}_i^b - \Delta\widetilde{U}_i^{b*}\right) M^{ab} = -\Delta t \left[G^{ab}\left(\widetilde{p}^n + \theta_2 \widetilde{\Delta p}\right) + \frac{\Delta t}{2} P^{ab} \widetilde{p}^n\right]$$

$$\Delta\widetilde{U}_i^b = \Delta\widetilde{U}_i^{b*} - \frac{1}{M^{ab}} \Delta t \left[G^{ab}\left(\widetilde{p}^n + \theta_2 \widetilde{\Delta p}\right) + \frac{\Delta t}{2} P^{ab} \widetilde{p}^n\right].$$

In summary, the three matricial equations to solve are:

1. Find ΔU_i^*

$$\Delta \widetilde{U}_i^{b*} = -\frac{1}{M^{ab}} \Delta t \left[C^{ab} \widetilde{U}_i^b + K_\tau^{ab} \widetilde{u}_i^b - f - \frac{\Delta t}{2} \left(K_s^{ab} \widetilde{U}_i^b + f_s^a \right) \right].$$

2. Find Δp

$$\left[M^{ab} + \Delta t^2 \theta_1 \theta_2 H^{ab} \right] \widetilde{\Delta p} = \Delta t \left[G^{ab} \widetilde{U}_i^b + \theta_1 G^{ab} \Delta \widetilde{U}_i^{b*} - \Delta t \theta_1 H^{ab} \widetilde{p}^n - f_p \right].$$

3. Find ΔU_i for time t^{n+1}

$$\Delta \widetilde{U}_i^b = \Delta \widetilde{U}_i^{b*} - \frac{1}{M^{ab}} \Delta t \left[G^{ab} \left(\widetilde{p}^n + \theta_2 \widetilde{\Delta p} \right) + \frac{\Delta t}{2} P^{ab} \widetilde{p}^n \right].$$

2.3 Fixed Grid implementation

Regarding the discretization process, it is necessary to describe the way the main variables in the analysis, specifically those who appear after proposing the weak formulation, are interpolated. The analysis is based on four-node, squared elements proposed in the physical coordinate space, this means that it is not required for the element to be transformed into other coordinate system. Because of the features of the proposed method, the mesh is fixed and the integration can be done analytically over one element. The interpolation functions that are used here belong to the family of Lagrange elements based on *n-th* order Lagrange polynomials. For the case of study the mesh will remain fixed in time, interpolation functions are given by Reddy (2006). The interpolation functions are described based on Figure 2.1, this configuration makes possible the use of rectangular or square elements.

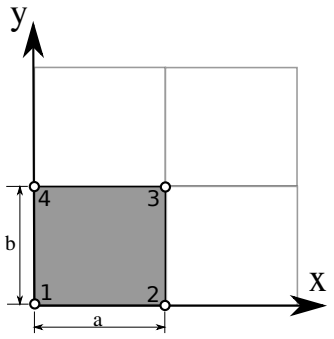


Figure 2.1: Node numbers and connectivity order for a square element in a cartesian grid

The functions are

$$\begin{aligned} N_1 &= \left(1-\frac{x}{a}\right)\left(1-\frac{y}{b}\right) \\ N_2 &= \frac{x}{a}\left(1-\frac{y}{b}\right) \\ N_3 &= \frac{xy}{ab} \\ N_4 &= \left(1-\frac{x}{a}\right)\frac{y}{b}, \end{aligned} \tag{2.31}$$

where a and b are the dimensions of the element in x and y directions, respectively.

As the changes in density are negligible when treating incompressible flow regimes, the time term of conservation equation can be dismissed. When using explicit time stepping schemes, the transient density term can be changed by an equivalent pressure term, both terms are linked by a compressibility parameter, the speed of sound c. When the speed of sound is very high (approaches infinity) a stern limitation of the time step appears affecting the explicit feature of the solution. Because of this consideration, the use of an artificial compressibility parameter β instead of the speed of sound, eliminates the restriction of the time step at the second step of the split. The selection of the artificial compressibility parameter is based on the convective and diffusive velocities of the problem, and calculated locally. In the present work, as the mesh is fixed, this parameter is calculated once. β is taken as the maximum value between both convective and diffusive velocities, and a constant parameter ϵ.

$$\boldsymbol{u}_{conv} = \sqrt{\boldsymbol{u}_i \boldsymbol{u}_i} \tag{2.32}$$

$$\boldsymbol{u}_{diff} = \frac{2}{hRe} \tag{2.33}$$

In these equations, h is the local element size, unique for this work, and Re is the Reynolds number. Now, after calculating both velocities, the timestep is obtained from choosing the minimum value of the following relation:

$$\triangle t = min(\triangle t_{conv}, \triangle t_{diff}) \tag{2.34}$$

where:

$$\triangle t_{conv} = \frac{h}{\boldsymbol{u}_{conv} + \beta} \quad ; \quad \triangle t_{diff} = \frac{h^2 Re}{2} \tag{2.35}$$

In order to accelerate convergence of the method, Nithiarasu (2003) proposes the use of a safety factor varying from 0.5 to 2.0 depending on the problem. This factor can be seen multiplying the calculated $\triangle t$ in the work of de Carvalho et al. (2009).

As reported in Nithiarasu (2003) and Manzari (1999), the results of the lid driven cavity problem (see section 3.2) for low Reynolds numbers present some difficulties when using several artificial compressibility schemes. The results reported by Nithiarasu (2003) evindenced good behavior at low Reynolds numbers for coarse structured and unstructured meshes, but showed that for high Reynolds numbers the mesh must be refined near the boundaries of the cavity in order to capture vortices and to achieve convergence of the method. The work from Zienkiewicz et al. (1999) showed an excellent behavior for the lid driven problem as well as other benchmarks. Nevertheless, for Reynolds numbers of 5000 and above the mesh is needed to be adapted in order to capture vorticity effects and stable solutions. Other results reported by Wang et al. (2011) showed that results are improved when the timestep value is decreased and it is also accurate and stable for small mesh density (30x30) The Characteristic Based Split method is adequate for simulating any condition of Newtonian flows but it has some trouble with the imposition of boundary conditions, and the setup of Artificial Compressibility parameters.

2.4 Discussion

The method was implemented using a fully explicit scheme as proposed by Zienkiewicz et al. (2005) but presented some difficulties when choosing the timestep and the selection of the parameters. Explicit methods are conditioned by the selection of appropriate timesteps, therefore it may be necessary to choose small timesteps in order to capture the behavior of convection. Despite of using a Crank Nicolson scheme for the time discretization, the resulting matricial system was sparse and the solution became expensive for the aim of the project. Furthermore, when using the Artificial Compressibility Method, it was tough to find decent values for the timestep, so a freely chosen timestep of $1e-6$ was used, which reduced even more the efficiency of calculations. The discretization process of the CBS scheme is not very clear and despite the advantages for the treatment of convection, the method itself presented some parameters for which the setup was somehow tricky. Because of these issues, the solution of the Navier Stokes Equation under a Lagrange-Galerkin Method is presented in chapter 3. The Fixed Grid approach is considered for the new proposal of solution, but for this case, an additional node is placed at the center of the element. The advantage of using this type of element is that it is still linear and satsfies the LBB condition. The results of using this type of element are similar to the obtained with quadratic elements with the asset of using fewer degrees of freedom (DOF). Further description of the element is presented in chapter 3, section 3.2.

Chapter 3

Solution of the NS Equation using the Lagrange-Galerkin method

3.1 Lagrange Galerkin Method

The Lagrange-Galerkin method is used to find the solution to a problem involving convection. The method is suitable to find the solution to any hyperbolic equation like the Navier Stokes set. The solution of the convective part is done in a lagrangian fashion, following the trajectory of each material point in the domain, which is fixed in space. The advantage of performing such solution is to get rid of remeshing thus avoiding for example the formulation of ALE methods. The use of large timesteps makes this method more efficient than Eulerian schemes Giraldo (1998), Ferretti (2013). The procedure is intended to solve the equations for incompressible flow only. Let's recall the Navier-Stokes Equations,

$$\begin{aligned} \rho\left(\frac{\partial u}{\partial t}+\left(u\cdot\nabla\right)u\right)-\mu\triangle u+\nabla p &= \rho f \\ div\left(u\right) &= 0 \end{aligned} \tag{3.1}$$

Where u is the velocity field, p is the pressure, μ is the dynamic viscosity and ρ is the density. The convective acceleration can be ignored if the fluid velocity is low compared to viscous forces (Stokes flow) but in case of high speed flows this is the dominant term of the equation. Dividing both sides of the equation to the density, constant for the case

of incompressible flow, an alternate formulation of the problem can be achieved.

$$\begin{aligned} \left(\frac{\partial u}{\partial t} + (u \cdot \nabla) u\right) - \nu \triangle u + \frac{1}{\rho} \nabla p &= f \\ div\,(u) &= 0 \end{aligned} \tag{3.2}$$

To complete the set of equations, boundary and initial conditions are needed. Initial condition $u\,(x, t = 0) = u_0\,(x)$ must be divergence-free. Boundary conditions are of Dirichlet and Neumann type. In general, the Dirichlet boundary condition is defined as a function $g\,(x, t = t)$ at the boundary $\partial\Omega$,whereas the homogeneous Dirichlet condition used to define the No slip condition is set as $u\,(x, t = t) = g\,(x, t = t)$ in $\partial\Omega$, $(u_\Gamma = 0)$ The mixed condition, derived from the variational formulation is defined as $\mu \frac{\partial u}{\partial n} - pn = g$ in Γ_N.

3.1.1 Variational formulation and spatial discretization

The equation is transformed from its strong form to a discrete system of matricial equations. The process starts with the multiplication of each term of the equation with a test function defined over an adequate vector space, and integration using Green's formula. Functional spaces are defined as $V = H_0^1\,(\Omega)$ for the velocity and $Q = L_0^2\,(\Omega) = \left\{q \in L^2\,(\Omega)\,; \int_\Omega q\,dx = 0\right\}$ for the pressure.

In this way we get,

$$\begin{aligned} \int_\Omega \frac{\partial u}{\partial t} \cdot v\,dx + \int_\Omega \left[(u \cdot \nabla) u\right] \cdot v\,dx - \int_\Omega \nu \triangle u \cdot v\,dx + \frac{1}{\rho} \int_\Omega \nabla p \cdot v\,dx &= \int_\Omega f \cdot v\,dx \\ \int_\Omega div\,(u) \cdot q\,dx &= 0 \end{aligned} \tag{3.3}$$

After multiplying the test function, the diffusive and pressure terms are integrated by parts and expanded.

$$\underbrace{\int_\Omega \frac{\partial u}{\partial t} \cdot v\,dx + \int_\Omega \left[(u \cdot \nabla) u\right] \cdot v\,dx}_{Total\,accel.} + \nu \underbrace{\int_\Omega \nabla u : \nabla v\,dx}_{A} -$$

$$\underbrace{\int p \cdot div(v)\, dx}_{B^T} - \nu \int_{\partial\Omega} \frac{\partial u}{\partial n} v\, ds + \int_{\partial\Omega} pv \cdot n\, ds = \int_{\Omega} f \cdot v\, dx$$

$$\underbrace{\int_{\Omega} div\,(u) \cdot q\, dx}_{B} = 0 \tag{3.4}$$

It can be observed that after rearranging terms, the discrete formulation can be lead into a matricial system involving a matrix A for the laplacian operator, a matrix B involving the pressure gradient as well as its transpose which has into account the divergence of the velocity field in the mass conservation equation. According to Elman et al. (2006), the matricial system is

$$\begin{bmatrix} A & 0 & B_x^T \\ 0 & A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \\ p \end{Bmatrix} = \begin{Bmatrix} F_x \\ F_y \\ 0 \end{Bmatrix} \tag{3.5}$$

where,

$$\begin{aligned} A &= B_x^T \int_{\Omega} \nabla u : \nabla v; \qquad B_x^T = -\int_{\Omega} p \cdot div(v_x); \qquad B_y^T = -\int_{\Omega} p \cdot div(v_y) \\ B_x &= -\int_{\Omega} q \cdot div(u_x); \qquad B_y = -\int_{\Omega} q \cdot div(u_y) \end{aligned} \tag{3.6}$$

The mixed formulation of the matricial system suggests that the discrete spaces to approximate velocity and pressure fields, must be compatible, accomplishing the inf-sup condition. Matrix A is positive definite, implying that exists a unique solution regarding velocity, nevertheless, it may exist more than one solution for the pressure, generating the spurious pressure modes Elman et al. (2006). Although simple approximations for the mixed formulation exist, they may be unstable and, in some cases, they can be conditioned to mesh size. One way to avoid the indetermination of the pressure solution is to define the pressure in one point of the domain or to define an average of zero pressure over the domain. One procedure to remove the spurious pressure modes and be able to achieve a unique solution is by using stabilization techniques that modify the incompressibility condition (Quarteroni and Valli (2008), Gunzburger (1989)). In this case, a penalization method, in which a perturbation coefficient $\epsilon > 0$ is introduced in the mass conservation

equation.

$$\begin{aligned} \int_\Omega \frac{\partial u}{\partial t} \cdot v \, dx + \int_\Omega \left[(u \cdot \nabla) \, u \right] \cdot v \, dx - \int_\Omega \nu \triangle u \cdot v \, dx + \frac{1}{\rho} \int_\Omega \nabla p \cdot v \, dx &= \int_\Omega f \cdot v \, dx \\ \int_\Omega div \, (u) \cdot q \, dx - \epsilon \int_\Omega p \cdot q dx &= 0 \end{aligned} \tag{3.7}$$

After the approximation it is possible to introduce a mass matrix that multiplies the perturbation coefficient. This matrix must be inverted and may become inefficient when using small values of ϵ because the resulting system may be ill conditioned and present trouble when solving via iterative methods Segal (2011). In this case a value of $\epsilon = 1e-4$ is used. A way to penalize the mixed system and eliminate the pressure term is by lumping of the mass matrix and its multiplication with the perturbation coefficient. However, the solution is equivalent to replace the mass matrix for the identity matrix so additional computation is avoided Gunzburger (1989)

The system to solve now becomes,

$$\begin{bmatrix} A & 0 & B_x^T \\ 0 & A & B_y^T \\ B_x & B_y & -\epsilon I \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \\ p \end{Bmatrix} = \begin{Bmatrix} F_x \\ F_y \\ 0 \end{Bmatrix} \tag{3.8}$$

3.1.2 Temporal discretization

The set of local and convective acceleration define the concept of total acceleration or material derivative in the Navier Stokes equation. The material derivative is the starting point to perform the temporal discretization process in characteristic curves. According to Pironneau (1982) and Donea and Huerta (2003) , with reference to a Lagrangian point of view, along the characteristic curve, the material derivative is reduced to a time derivative which can be discretized by an implicit method in a finite difference method scheme.

$$\frac{Du}{Dt} = \left(\frac{\partial u}{\partial t} + (u \cdot \nabla) \, u \right) = \frac{u^{n+1} - u_*^n}{dt} \tag{3.9}$$

In this way the Navier Stokes Equation becomes,

$$\begin{aligned} \frac{u^{n+1}-u_*^n}{dt} - \nu\triangle u^{n+1} + \frac{1}{\rho}\nabla p^{n+1} &= f^{n+1} \\ div\left(u^{n+1}\right) &= 0 \end{aligned} \tag{3.10}$$

The u_*^n term is obtained after solving the characteristic curve equation for a material point of the mesh. According to Pironneau (1982) and Bermudez et al. (2006), for a given point $(x, y) \in \Omega$, the characteristic curve that passes through that point is the function $X(x, t)$ that solves the Initial Value Problem,

$$\begin{aligned} \frac{\partial X}{\partial \tau} &= u\left(X, \tau\right) \\ X\left(t\right) &= x \end{aligned} \tag{3.11}$$

The idea of the method is to follow the trajectory of a material point in the domain through its characteristic curve, finding the position where the point was an instant of time before, performing there the calculation of the scalar quantity being transported. For the case of the Navier Stokes equation, the transported variable is the velocity u_*^n. After solving this velocity variable, it is replaced on equation 3.10 and the system is solved for time t^{n+1}. This procedure is done iteratively until the convergence of the method is reached ($1e-5$ for velocity and pressure residuals)

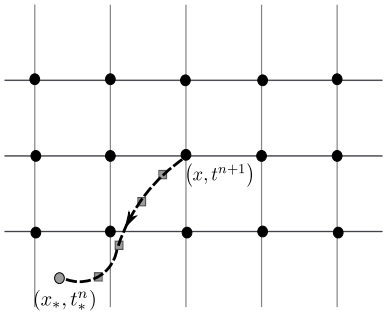


Figure 3.1: Trajectory of a material point through its characteristic curve

Figure 3.1 shows the trajectory of a material point of the domain, travelling from a position labeled (t^{n+1}) to the position where it was before at time t^n. Gray squares in the figure represent the divisions of the characteristic curve for the time integration

process,done by using a Backward Euler scheme (see equation 3.12)

$$x_* = x - u(x)dt \tag{3.12}$$

where x_* is the position where the particle was an instant before, x is the position on the grid, and $u(x)$ the value of the velocity evaluated on the grid.

The general algorithm for the Navier Stokes solver is presented in 1. The basic features of a CFD simulation are defined on this algorithm.

Algorithm 1 General algorithm for the Navier Stokes solver

Preprocessing
Define physical properties
Detect boundaries in the domain
Assign boundary and initial conditions
Create local matrices
Processing
Assembly global matrices
Solve the transient term (Characteristics)
Solve the Stokes system
Post processing
Plot results

Regarding the temporal part of the equation, the detailed algorithm for the treatment of the characteristics is:

Algorithm 2 Computation of the characteristics

```
Preprocessing
for t = 0 to t = t_final with increments dt do
    Create internal node array
    for Each internal node do
        Find position (x^{n+1}, y^{n+1})
        Find velocity at (x^{n+1}, y^{n+1})
        for τ = 0 to dt do
            Call Backward Euler function
            Find element number
            Find connectivity
            Find node coordinates
            Call the interpolation function to obtain u*^n
            Update velocity Update position
        end for
        Solve the Stokes system
        Store results
    end for
    Update variables
end for
Plot results
```

The inner cycle is used to calculate the position where the material point was at time t^n. It is important to mention that the inner cycle uses an internal timestep value denoted by τ

The implementation was done in C++ using SuiteSparse, a suite of sparse matrix packages, for the solution of the matricial system.

3.2 Case study: Lid Driven Cavity Flow

The problem of the lid driven cavity flow is widely known for being a benchmark to validate new numerical methods in fluid mechanics. The case is a square cavity with a no slip velocity condition ($u = 0, v = 0$) in all the walls excepting the top one, in which water is flowing with a dimensionless velocity of $u = 1, v = 0$, see figure 3.2. The behavior of the fluid is different depending which Reynolds number is used.

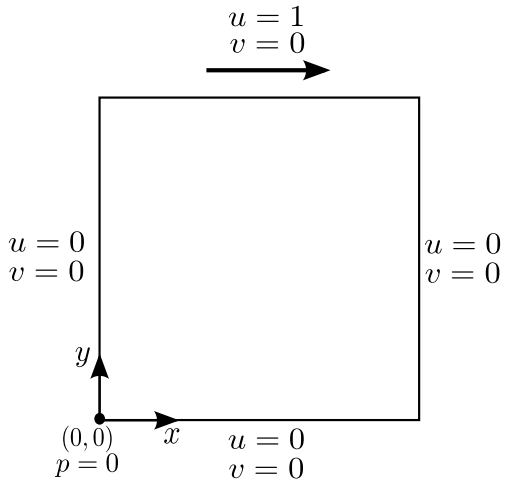


Figure 3.2: Lid driven cavity flow

The domain is discretized with a structured fixed grid, with square $Q1b/Q1$ elements, which is the smallest subspace that guarantees stability as it satisfies the LBB condition Fourestey and Moubachir (2002). This configuration presents a square element with velocity degrees of freedom in each corner and center nodes, and pressure degrees of freedom only at the corners of the element. In contrast to the work reviewed in the literature, the development of the method of characteristics is done using a structured square mesh, providing another perspective of the solution. The use of a structured grid for this type of problems has not been seen in the literature yet. This type of element is closely related to the $P1b/P1$ triangular element which is a simplification of the widely known Hood-Taylor element. The element used in the implementation can be seen in figure 3.3.

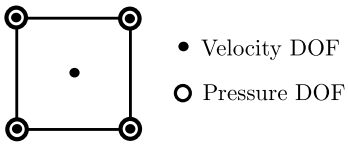


Figure 3.3: Q1b/Q1 element

The integration is done analytically over one element with origin at the lower left corner node. The process of integration and resulting matrices are defined on appendix A. The fixed grid concept belongs to the field of meshless methods, being used for the

treatment of problems in which the geometry of the domain changes in time. The method has been used for phase change problems, elasticity problems, and it might be suitable for problems involving fluid structure interaction. The advantage of using this approach is because there is no need for remeshing, ideal for the characteristic curve implementation. Regarding to the implementation, the points of the mesh are not stored but calulated according the maximum dimensions of the domain. In this way, the mesh data (points and connectivity) is only needed for post processing the results. In figure 3.4 a fixed mesh of 32 x 32 elements can be seen. Observe that the node at the center of each element is not plotted.

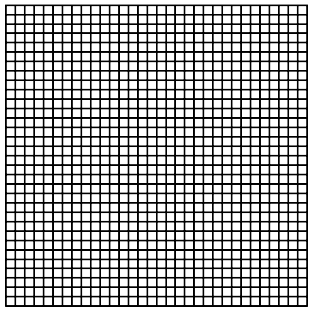
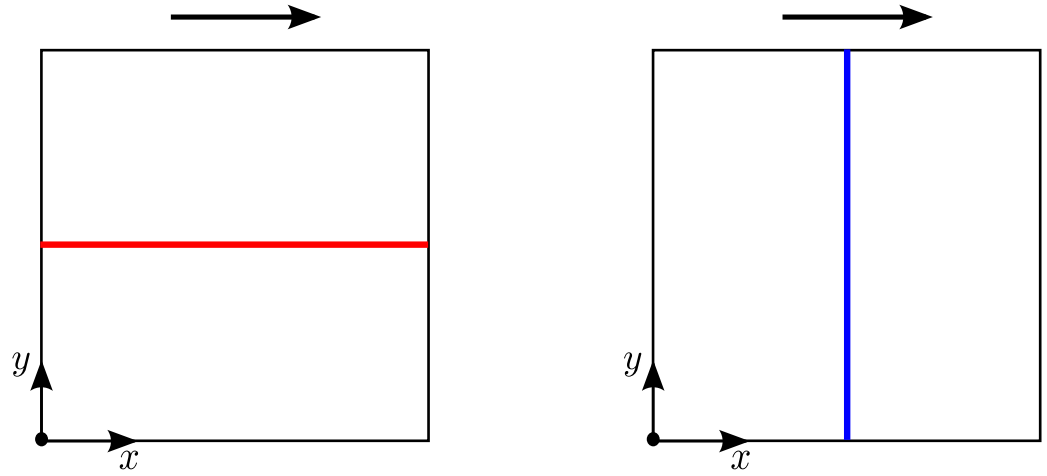


Figure 3.4: Finite Element Fixed Grid (32 x 32) elements

3.3 Results

The case study was solved using several values for the Reynolds number (Re=100,400,1000,3200) and it is named as charNS. The results show good behavior of the flow when compared to other experimental works done by Ghia et al. (1982) and openFOAM. The way to validate the method is comparing the velocity profiles from a vertical and a horizontal line that cross the geometric center of the cavity.



(a) Horizontal line for measurement (b) Vertical line for measurement

Figure 3.5: Horizontal and Vertical lines for velocity measurement

In order to compare the velocity values, seven points were distributed on each line, according the documentation of Ghia et al. (1982). A fixed mesh of 129 x 129 square elements was used. In the following tables the results for several Reynolds number are compared,

For $Re = 100$,

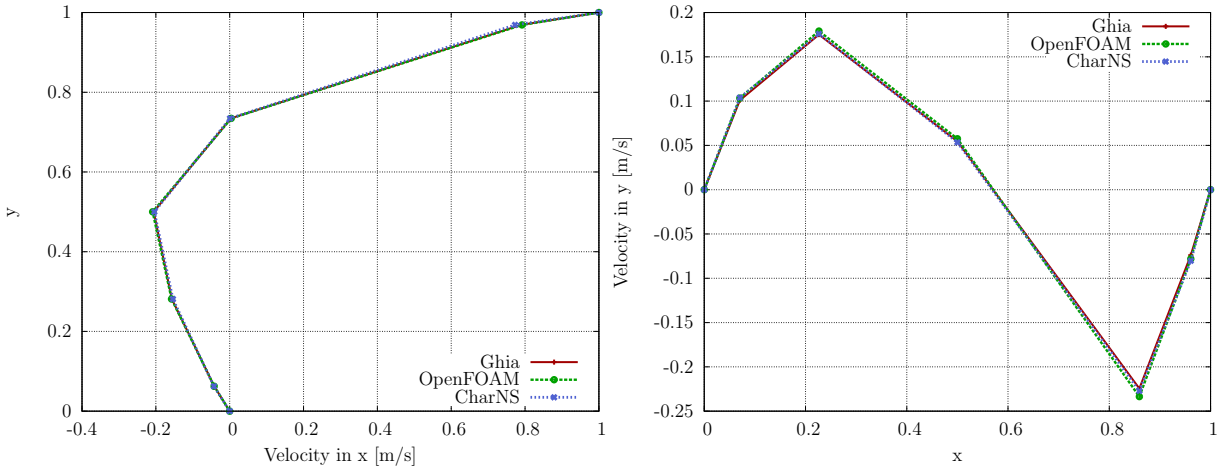
	Re 100		
Position	u_x		
y	Ghia et al.	OpenFOAM	CharNS
1.0	1.0	1.0	1.0
0.9688	0.78871	0.79176	0.7730
0.7344	0.00332	0.004072	0.00084
0.5	-0.20581	-0.2087	-0.20417
0.2813	-0.15662	-0.15746	-0.15394
0.0625	-0.04192	-0.04197	-0.04266
0.0	0.0	0.0	0.0

Table 3.1: (u_x) Velocities through the vertical line of the cavity for $Re = 100$

	Re 100		
Position	u_y		
x	Ghia et al.	OpenFOAM	CharNS
1.0	0.0	0.0	0.0
0.9609	-0.07391	-0.07805	-0.08019
0.8594	-0.22445	-0.23354	-0.22714
0.5	0.05454	0.05751	0.05342
0.2266	0.17507	0.17901	0.1759
0.0703	0.10091	0.10336	0.10389
0.0	0.0	0.0	0.0

Table 3.2: (u_y) Velocities through the horizontal line of the cavity for $Re = 100$

Results for the velocity in horizontal direction (u_x) through the vertical line, and the velocity in vertical direction (u_y) through the horizontal line for $Re = 100$ are shown in figure 3.6



(a) u_x,$Re = 100$ (b) u_y,$Re = 100$

Figure 3.6: Velocities for $Re = 100$

The results for $Re = 400$ are shown below.

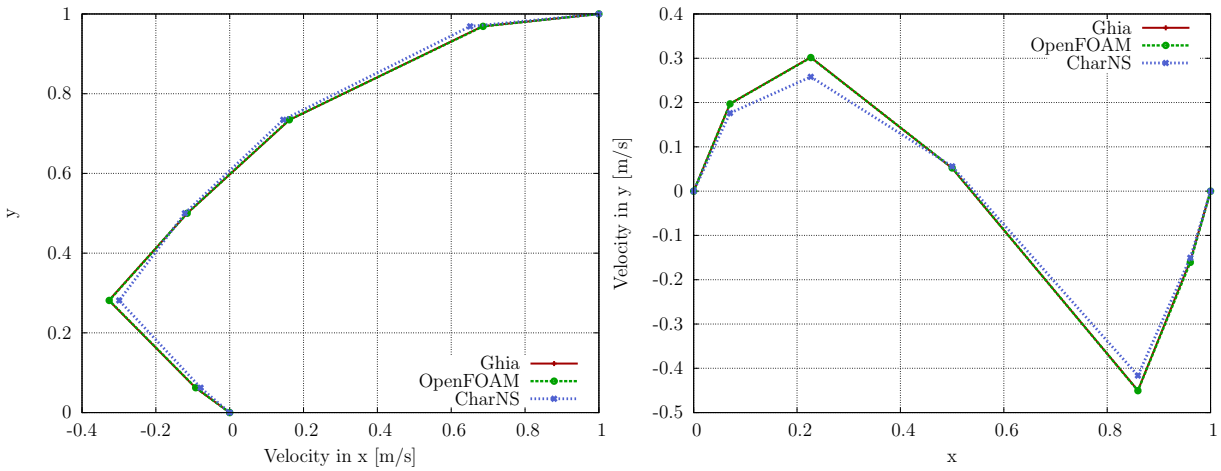
	Re 400		
Position	u_y		
x	Ghia et al.	OpenFOAM	CharNS
1.0	0.0	0.0	0.0
0.9609	-0.15663	-0.16089	-0.1504
0.8594	-0.44993	-0.45056	-0.4163
0.5	0.05186	0.05238	0.05601
0.2266	0.30203	0.30159	0.25803
0.0703	0.19713	0.19698	0.17620
0.0	0.0	0.0	0.0

Table 3.4: (u_y) Velocities through the horizontal line of the cavity for $Re = 400$

	Re 400		
Position	u_x		
y	Ghia et al.	OpenFOAM	CharNS
1.0	1.0	1.0	1.0
0.9688	0.68439	0.6863	0.6513
0.7344	0.16256	0.16149	0.1451
0.5	-0.11477	-0.11498	-0.1213
0.2813	-0.32726	-0.32625	-0.2993
0.0625	-0.09266	-0.09216	-0.0788
0.0	0.0	0.0	0.0

Table 3.3: (u_x) Velocities through the vertical line of the cavity for $Re = 400$

Plotting the results for $Re = 400$ in figure 3.7



(a) u_x,$Re = 400$ (b) u_y $Re = 400$

Figure 3.7: Velocities for $Re = 400$

Now,the tables with the results for a Reynolds number of 1000 are shown below. Some values slightly differ from the references.

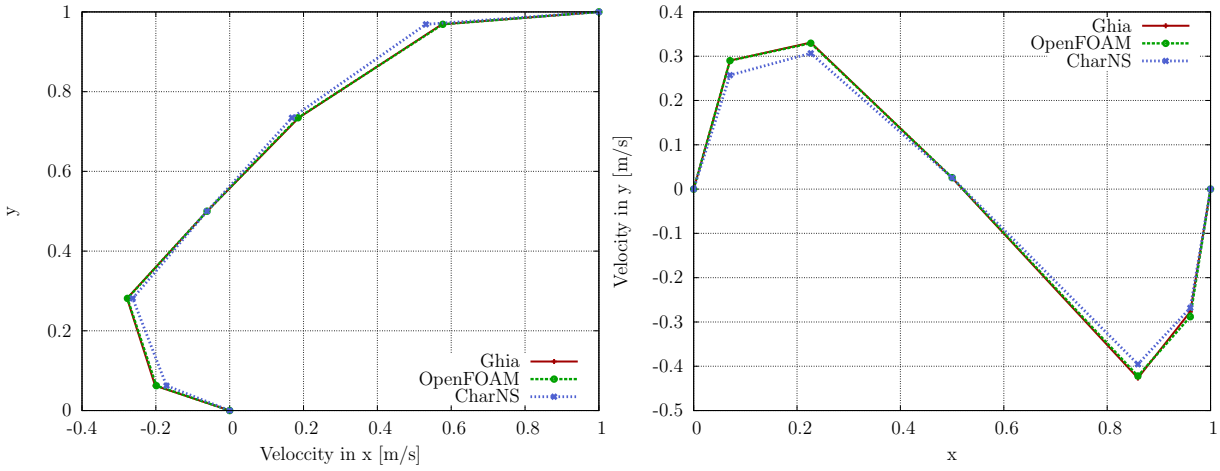
	Re 1000		
Position	u_x		
y	Ghia et al.	OpenFOAM	CharNS
1.0	1.0	1.0	1.0
0.9688	0.57492	0.57721	0.5315
0.7344	0.18719	0.18524	0.16848
0.5	-0.06080	-0.06126	-0.0613
0.2813	-0.27805	-0.27702	-0.26271
0.0625	-0.20196	-0.19874	-0.1711
0.0	0.0	0.0	0.0

Table 3.5: (u_x) Velocities through the vertical line of the cavity for $Re = 1000$

	Re 1000		
Position	u_y		
x	Ghia et al.	OpenFOAM	CharNS
1.0	0.0	0.0	0.0
0.9609	-0.27669	-0.28819	-0.2681
0.8594	-0.42665	-0.42197	-0.39546
0.5	0.02526	0.02575	0.02555
0.2266	0.33075	0.32940	0.30655
0.0703	0.29012	0.29008	0.25719
0.0	0.0	0.0	0.0

Table 3.6: (u_y) Velocities through the horizontal line of the cavity for $Re = 1000$

Plotting the results for $Re = 1000$ in figure 3.8



(a) u_x,$Re = 1000$ (b) u_y,$Re = 1000$

Figure 3.8: Velocities for $Re = 1000$

The last simulated case was $Re = 3200$, which means a flow in the transition regime, almost turbulent. It can be seen that vortices are found without the inclusion of turbulence models.

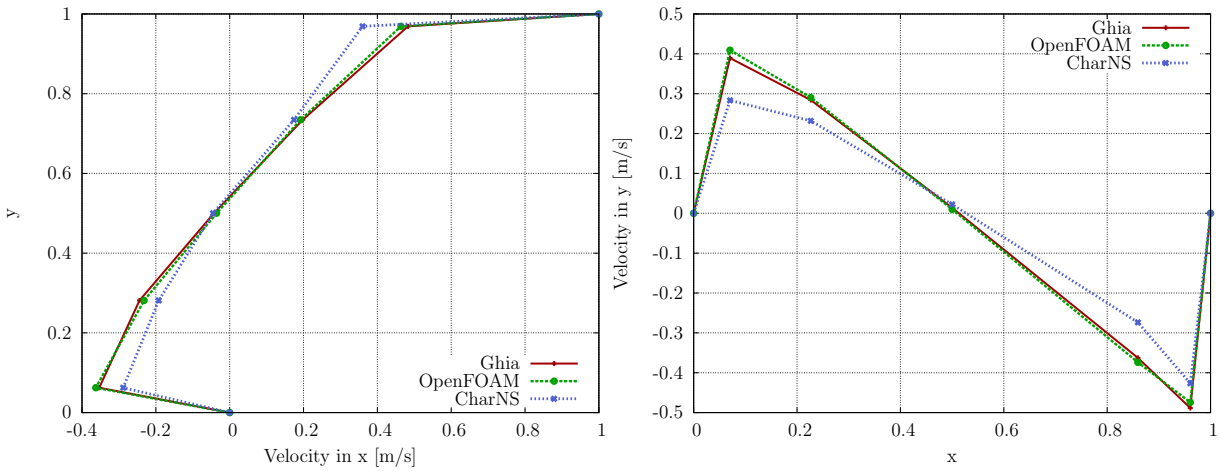
	Re 3200		
Position	u_x		
y	Ghia et al.	OpenFOAM	CharNS
1.0	1.0	1.0	1.0
0.9688	0.48296	0.4640	0.35982
0.7344	0.19791	0.1919	0.17418
0.5	-0.04272	-0.03561	-0.04511
0.2813	-0.24427	-0.231938	-0.1923
0.0625	-0.35344	-0.36277	-0.2885
0.0	0.0	0.0	0.0

Table 3.7: (u_x) Velocities through the vertical line of the cavity for $Re = 3200$

	Re 3200		
Position	u_y		
x	Ghia et al.	OpenFOAM	CharNS
1.0	0.0	0.0	0.0
0.9609	-0.47425	-0.4885	-0.42656
0.8594	-0.37401	-0.3628	-0.2737
0.5	0.00999	0.01372	0.0221
0.2266	0.29030	0.28412	0.23192
0.0703	0.40917	0.38948	0.28326
0.0	0.0	0.0	0.0

Table 3.8: (u_y) Velocities through the horizontal line of the cavity for $Re = 3200$

Plotting the results for $Re = 3200$ in figure 3.9



(a) u_x,$Re = 3200$ (b) u_y,$Re = 3200$

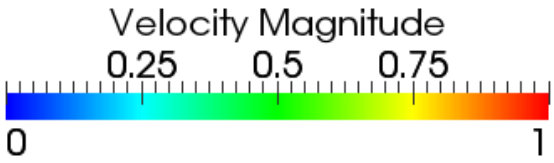
Figure 3.9: Velocities for $Re = 3200$

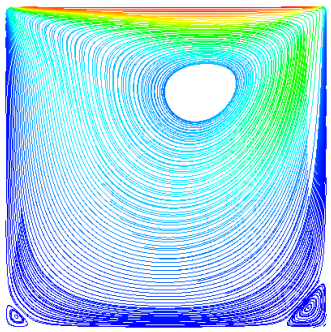
Comparing the results from figures(3.6, 3.7, 3.8, 3.9) it can be observed that horizontal velocity values (u_x) found with the implementation of the characteristics method are similar to the references, in which a finite volume method was used. There is only one of the points in which the velocity differs about 10% according to reference Ghia et al. (1982). Analyzing the graphs for the velocity in vertical direction (u_y), when the Reynolds number is increased, velocities show little discrepancy when compared with the references. This difference might be due to the presence of numerical diffusion, proper of a first order upwinding method, therefore affecting the stability of the solution Fourestey and Piperno (2004).

Nevertheless, with the mesh used, vortices can be seen at $Re = 1000$ and $Re = 3200$, see figure 3.10. Vortex formation is generated according the behavior of the flow as stated by Erturk et al. (2005). The main vortex can be seen, located near the center of the cavity, while as the Reynols number is increased, other vortices appear (secondary and tertiary vortices). The location of the primary vortex center is very close to the value reported by Ghia et al. (1982) (see table 3.9). It can be seen in figure 3.10c

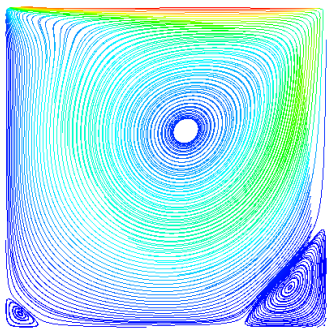
Coordinate	Ghia et al.	CharNS
x	0.5313	0.534884
y	0.5625	0.565891

Table 3.9: Location of the center of the primary vortex for $Re = 1000$

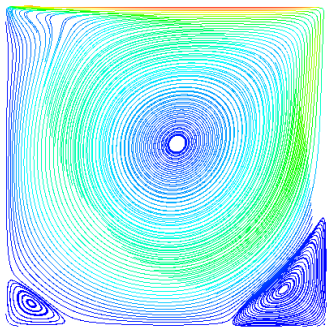




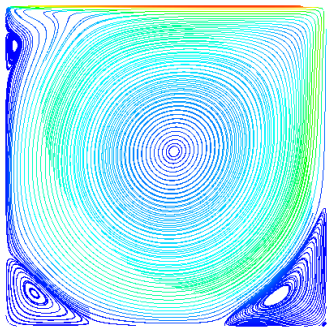
(a) $Re = 100$



(b) $Re = 400$



(c) $Re = 1000$



(d) $Re = 3200$

Figure 3.10: Streamlines for different Reynolds numbers

3.4 Conclusion

The development of the Lagrange-Galerkin method is able to solve the Navier Stokes equation for incompressible flow combining advantages from the Eulerian and Lagrangian schemes, as it can be implemented on a fixed mesh avoiding the process of remeshing. When using this method, transient problems can be solved efficiently using large timesteps whenever the characteristic curve is sufficiently smooth, in other words, if it can be split in small sections. Compared to the CBS scheme, the discretization process is much clearer as it derives from a variational formulation of the Stokes problem and the material derivative is treated as a simple time derivative which can be discretized in a finite difference manner. Moreover, the imposition of boundary conditions is also clearer because of the appearance of these in the variational formulation in a natural way.

Regarding to the results, despite the existence of concordance with the calculated values and those reported on the references, when the Reynolds number is increased, the values begin to differ, being necessary the change of the characteristic curve approximation to a second order scheme in order to lessen numerical dissipation. Another way to improve the solution is by doing a mesh refinement on zones where vortices tend to appear. The main advantages of the method are the ability to handle the convective term in a Lagrangian fashion, the clear variational formulation and the use of large timestepping.

Appendix A

Stokes flow: calculation of local matrices

|STOKES FLOW
Calculation of Matrices
Jose David Villegas

BASE FUNCTIONS AND GRADIENT

```
> restart:
  with(LinearAlgebra):
  with(CodeGeneration):



  #An Introduction to the FEM 3rd Ed. J.N Reddy pag. 535
  # Base Functions for a four node element with origin at lower left corner
  (1)
  #           4---------3
  #           |         |
  #         h |    5    |
  #           |         |
  #           1---------2
  #                h




  N1:=-(3/2)*x/h-(3/2)*y/h+x*y/h^2+1+(1/2)*(x^2+y^2)/h^2:
  N2:=(1/2)*x/h-(1/2)*y/h-x*y/h^2+(1/2)*(x^2+y^2)/h^2:
  N3:=-(1/2)*x/h-(1/2)*y/h+x*y/h^2+(1/2)*(x^2+y^2)/h^2:
  N4:=-(1/2)*x/h+(1/2)*y/h-x*y/h^2+(1/2)*(x^2+y^2)/h^2:
  N5:=2*x/h+2*y/h-2*(x^2+y^2)/h^2:
```

```
> # Base functions Vector N = Nu = Np

  N := Vector([N1,N2,N3,N4,N5]):
  Np:= Vector([N1,N2,N3,N4]):
```

```
> # Calculation of the gradient for the functions

  dN:=Matrix(5,2):
  for i from 1 to 5 do
    dN[i,1]:= simplify(diff(N[i],x));
    dN[i,2]:= simplify(diff(N[i],y));
  od:
  dN:
```

MATRICES CALCULATION

Mass Matrix (*Velocity*)

$$M = \int_{\Omega} N^a N^b$$

```
> Mlocal:=Matrix(5,5):
```

```
Mlump:=Matrix(5,5):
Stab:=Matrix(4,4):
Ident:= IdentityMatrix(4):

for r from 1 to 5 do
  for s from 1 to 5 do

    Mlocal[r,s] := simplify((int(int(N[r]*N[s],y=0..h),x=0..h))):
    Mlump[r,r] :=Mlump[r,r] + Mlocal[r,s]:

  od:
od:

for r from 1 to 4 do
 for s from 1 to 4 do
  Stab[r,s]:=-Ident[r,s]*epsilon:#*Mlump[r,s]:
 od:
od:
```

A matrix (Equal to CBS H matrix)

$$A = \int_{\Omega} \nabla N^a \cdot \nabla N^b$$

```
> A:=Matrix(5,5):
  for i from 1 to 5 do
    for j from 1 to 5 do
     A[i,j] := simplify(kinvisc*int(int(dN[i,1]*dN[j,1]+
                                        dN[i,2]*dN[j,2],y=0.. h),x=0..h)):
    od:
  od:
```

Body force **in** 1 **and** 2 *directions*

$$f_i = \int_{\Omega} \rho g_i N^a$$

```
> f1 := Vector(5):
  for r from 1 to 5 do
    f1[r] := simplify(int(int(dt*(1/rho)*rho*g1*N[r],y=0..h),x=0..h)):
  od:


  f2 := Vector(5):
  for r from 1 to 5 do
    f2[r] := simplify(int(int(dt*(1/rho)*rho*g2*N[r],y=0..h),x=0..h)):
  od:
```

$$B = \int_{\Omega} N_p^a \nabla N^b$$

```
> Bx:=Matrix(4,5):
  By:=Matrix(4,5):
  Bxt:=Matrix(5,4):
```

```
Byt:=Matrix(5,4):

for r from 1 to 4 do
 for s from 1 to 5 do
  Bx[r,s]:=simplify((-1/rho)*int(int(Np[r]*dN[s,1],x=0..h),y=0..h)):
 od:
od:

for r from 1 to 4 do
 for s from 1 to 5 do
  By[r,s]:=simplify((-1/rho)*int(int(Np[r]*dN[s,2],x=0..h),y=0..h)):
 od:
od:

Bx:
By:

Bxt:=rho*Transpose(Bx):
Byt:=rho*Transpose(By):
```

K matrix

$$\begin{bmatrix} A & 0 & B_x^T \\ 0 & A & B_y^T \\ B_x & B_y & -\epsilon I \end{bmatrix}$$

```
> K:=Matrix(14,14):

  #A1

  K[1,1]:=dt*A[1,1]+Mlocal[1,1]:
  K[1,2]:=dt*A[1,2]+Mlocal[1,2]:
  K[1,3]:=dt*A[1,3]+Mlocal[1,3]:
  K[1,4]:=dt*A[1,4]+Mlocal[1,4]:
  K[1,5]:=dt*A[1,5]+Mlocal[1,5]:

  K[2,1]:=dt*A[2,1]+Mlocal[2,1]:
  K[2,2]:=dt*A[2,2]+Mlocal[2,2]:
  K[2,3]:=dt*A[2,3]+Mlocal[2,3]:
  K[2,4]:=dt*A[2,4]+Mlocal[2,4]:
  K[2,5]:=dt*A[2,5]+Mlocal[2,5]:

  K[3,1]:=dt*A[3,1]+Mlocal[3,1]:
  K[3,2]:=dt*A[3,2]+Mlocal[3,2]:
  K[3,3]:=dt*A[3,3]+Mlocal[3,3]:
  K[3,4]:=dt*A[3,4]+Mlocal[3,4]:
  K[3,5]:=dt*A[3,5]+Mlocal[3,5]:

  K[4,1]:=dt*A[4,1]+Mlocal[4,1]:
  K[4,2]:=dt*A[4,2]+Mlocal[4,2]:
  K[4,3]:=dt*A[4,3]+Mlocal[4,3]:
  K[4,4]:=dt*A[4,4]+Mlocal[4,4]:
  K[4,5]:=dt*A[4,5]+Mlocal[4,5]:

  K[5,1]:=dt*A[5,1]+Mlocal[5,1]:
  K[5,2]:=dt*A[5,2]+Mlocal[5,2]:
  K[5,3]:=dt*A[5,3]+Mlocal[5,3]:
  K[5,4]:=dt*A[5,4]+Mlocal[5,4]:
  K[5,5]:=dt*A[5,5]+Mlocal[5,5]:
```

```
#A2

K[6,6]:=dt*A[1,1]+Mlocal[1,1]:
K[6,7]:=dt*A[1,2]+Mlocal[1,2]:
K[6,8]:=dt*A[1,3]+Mlocal[1,3]:
K[6,9]:=dt*A[1,4]+Mlocal[1,4]:
K[6,10]:=dt*A[1,5]+Mlocal[1,5]:

K[7,6]:=dt*A[2,1]+Mlocal[2,1]:
K[7,7]:=dt*A[2,2]+Mlocal[2,2]:
K[7,8]:=dt*A[2,3]+Mlocal[2,3]:
K[7,9]:=dt*A[2,4]+Mlocal[2,4]:
K[7,10]:=dt*A[2,5]+Mlocal[2,5]:

K[8,6]:=dt*A[3,1]+Mlocal[3,1]:
K[8,7]:=dt*A[3,2]+Mlocal[3,2]:
K[8,8]:=dt*A[3,3]+Mlocal[3,3]:
K[8,9]:=dt*A[3,4]+Mlocal[3,4]:
K[8,10]:=dt*A[3,5]+Mlocal[3,5]:

K[9,6]:=dt*A[4,1]+Mlocal[4,1]:
K[9,7]:=dt*A[4,2]+Mlocal[4,2]:
K[9,8]:=dt*A[4,3]+Mlocal[4,3]:
K[9,9]:=dt*A[4,4]+Mlocal[4,4]:
K[9,10]:=dt*A[4,5]+Mlocal[4,5]:

K[10,6]:=dt*A[5,1]+Mlocal[5,1]:
K[10,7]:=dt*A[5,2]+Mlocal[5,2]:
K[10,8]:=dt*A[5,3]+Mlocal[5,3]:
K[10,9]:=dt*A[5,4]+Mlocal[5,4]:
K[10,10]:=dt*A[5,5]+Mlocal[5,5]:

#Bxt

K[1,11]:=dt*Bxt[1,1]:
K[1,12]:=dt*Bxt[1,2]:
K[1,13]:=dt*Bxt[1,3]:
K[1,14]:=dt*Bxt[1,4]:

K[2,11]:=dt*Bxt[2,1]:
K[2,12]:=dt*Bxt[2,2]:
K[2,13]:=dt*Bxt[2,3]:
K[2,14]:=dt*Bxt[2,4]:

K[3,11]:=dt*Bxt[3,1]:
K[3,12]:=dt*Bxt[3,2]:
K[3,13]:=dt*Bxt[3,3]:
K[3,14]:=dt*Bxt[3,4]:

K[4,11]:=dt*Bxt[4,1]:
K[4,12]:=dt*Bxt[4,2]:
K[4,13]:=dt*Bxt[4,3]:
K[4,14]:=dt*Bxt[4,4]:

K[5,11]:=dt*Bxt[5,1]:
K[5,12]:=dt*Bxt[5,2]:
K[5,13]:=dt*Bxt[5,3]:
K[5,14]:=dt*Bxt[5,4]:

#Byt

K[6,11]:=dt*Byt[1,1]:
```

```
K[6,12]:=dt*Byt[1,2]:
K[6,13]:=dt*Byt[1,3]:
K[6,14]:=dt*Byt[1,4]:

K[7,11]:=dt*Byt[2,1]:
K[7,12]:=dt*Byt[2,2]:
K[7,13]:=dt*Byt[2,3]:
K[7,14]:=dt*Byt[2,4]:

K[8,11]:=dt*Byt[3,1]:
K[8,12]:=dt*Byt[3,2]:
K[8,13]:=dt*Byt[3,3]:
K[8,14]:=dt*Byt[3,4]:

K[9,11]:=dt*Byt[4,1]:
K[9,12]:=dt*Byt[4,2]:
K[9,13]:=dt*Byt[4,3]:
K[9,14]:=dt*Byt[4,4]:

K[10,11]:=dt*Byt[5,1]:
K[10,12]:=dt*Byt[5,2]:
K[10,13]:=dt*Byt[5,3]:
K[10,14]:=dt*Byt[5,4]:


#Bx
K[11,1]:=dt*Bx[1,1]:
K[11,2]:=dt*Bx[1,2]:
K[11,3]:=dt*Bx[1,3]:
K[11,4]:=dt*Bx[1,4]:
K[11,5]:=dt*Bx[1,5]:

K[12,1]:=dt*Bx[2,1]:
K[12,2]:=dt*Bx[2,2]:
K[12,3]:=dt*Bx[2,3]:
K[12,4]:=dt*Bx[2,4]:
K[12,5]:=dt*Bx[2,5]:

K[13,1]:=dt*Bx[3,1]:
K[13,2]:=dt*Bx[3,2]:
K[13,3]:=dt*Bx[3,3]:
K[13,4]:=dt*Bx[3,4]:
K[13,5]:=dt*Bx[3,5]:

K[14,1]:=dt*Bx[4,1]:
K[14,2]:=dt*Bx[4,2]:
K[14,3]:=dt*Bx[4,3]:
K[14,4]:=dt*Bx[4,4]:
K[14,5]:=dt*Bx[4,5]:

#By
K[11,6]:=dt*By[1,1]:
K[11,7]:=dt*By[1,2]:
K[11,8]:=dt*By[1,3]:
K[11,9]:=dt*By[1,4]:
K[11,10]:=dt*By[1,5]:

K[12,6]:=dt*By[2,1]:
K[12,7]:=dt*By[2,2]:
K[12,8]:=dt*By[2,3]:
K[12,9]:=dt*By[2,4]:
K[12,10]:=dt*By[2,5]:
```

```
    K[13,6]:=dt*By[3,1]:
    K[13,7]:=dt*By[3,2]:
    K[13,8]:=dt*By[3,3]:
    K[13,9]:=dt*By[3,4]:
    K[13,10]:=dt*By[3,5]:

    K[14,6]:=dt*By[4,1]:
    K[14,7]:=dt*By[4,2]:
    K[14,8]:=dt*By[4,3]:
    K[14,9]:=dt*By[4,4]:
    K[14,10]:=dt*By[4,5]:

    #S
    K[11,11]:=dt*Stab[1,1]:
    K[12,12]:=dt*Stab[2,2]:
    K[13,13]:=dt*Stab[3,3]:
    K[14,14]:=dt*Stab[4,4]:
```

Bibliography

Y. Achdou and J.-L. Guermond. Convergence analysis of a finite element projection/lagrange-galerkin method for the incompressible navier-stokes equations. *SIAM Journal of Numerical Analysis*, 37:799–826, 2000.

S.N Atluri and T. Zhu. A new meshless local petrov galerkin (mlpg) approach in computational mechanics. *Computational Mechanics*, 22:117–127, 1998.

T. Belytschko, Y.Y. Lu, and L. Gu. Element free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

J.P. Benque, B. Ibler, and G. Labadie. A finite element method for navier-stokes equations. In *Proceedings from the Third International Conference on Finite Elements in Flow Problems. Alberta, Canada*, 1980.

A. Bermudez, M. Nogueiras, and C. Vasquez. Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements. part i: Time discretization. *SIAM Journal on Numerical Analysis*, 44(5):1829–1853, 2006.

K. Boukir, B. Maday, B. Métivet, and E. Razafindrakoto. A high order characteristics/finite element method for the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 25:1421–1454, 1997.

A. Chorin. Numerical solution of navier-stokes equations. *Mathematics of Computation*, 22(104):748–762, 1968.

M. P. de Carvalho, V. L. Scalon, and A. Padilha. Analysis of cbs numerical algorithm execution to flow simulation using the finite element method. *Ingeniare Revista chilena de Ingeniería*, 17(2):166–174, 2009. doi: 10.4067/S0718-33052009000200005.

J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley, third edition, 2003.

H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2006.

E. Erturk, T.C. Corke, and Gökçöl. Numerical solutions of 2-d steady incompressible driven cavity flow at high reynolds numbers. *International Journal for Numerical Methods in Fluids*, 48:747–774, 2005.

R. Ferretti. On the relationship between semi-lagrangian and lagrange-galerkin schemes. *Numer*, 124:31–56, 2013.

G. Fourestey and M. Moubachir. Optimal control of navier-stokes equations usinglagrange-galerkin methods. Technical report, Unité de recherche INRIA Sophia Antipolis, 2002.

G. Fourestey and S. Piperno. A second-order time-accurate ale lagrange-galerkin method applied to wind engineering and control of bridge profiles. *Computer me*, 193:4117–4137, 2004.

U. Ghia, K.N Ghia, and T. Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387–411, 1982.

R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181: 375–389, 1977.

F. Giraldo. The lagrange-galerkin spectral element method on unstructured quadrilateral grids. *Journal of computational physics*, 147:114–146, 1998.

Max Gunzburger. *Finite Element Methods for Viscous Incompressible Flows. A Guide to Theory,Practice, and Algorithms*. Academic Press Inc., 1989.

S. Koshizuka, A. Nobe, and Y. Oka. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, 26:751–769, 1998.

M.T. Manzari. An explicit finite element algorithm for convection heat transfer problems. *International Journal of Numerical Methods for Heat & Fluid Flow*, 9:860–877, 1999.

P. Nithiarasu. On boundary conditions of the characteristic based split (cbs) algorithm for fluid dynamics. *International Journal for Numerical Methods in Engineering*, 54: 523–536, 2002. doi: 10.1002/nme.434.

P. Nithiarasu. An efficient artificial compressibility (ac) scheme based on the characteristic based split (cbs) method for incompressible flows. *International Journal for Numerical Methods in Engineering*, 56:1815–1845, 2003. doi: 10.1002/nme.712.

O. Pironneau. On the transport-diffusion algorithm and its applications to the navier-stokes equations. *Numerische Mathematik*, 38:309–332, 1982.

A. Quarteroni and A. Valli. *Numerical Approximation of Partial DIfferential Equations*. Springer, 2008.

J.N Reddy. *An Introduction to the Finite Element Method*. McGraw-Hill, third edition, 2006.

I. Segal. Finite element methods for the incompressible navier-stokes equations. Technical report, Delft University of Technology, 2011.

Roger Temam. *Navier-Stokes Equations*. North-Holland, 1977.

D. Wang, H. Wang, J. Xiong, and Tham L G. Characteristic-based operator-splitting finite element method for navier-stokes equations. *Science China Technological Sciences*, 54: 2157–2166, 2011.

W.X Wang and Y. Takao. Isoparametric finite point method in computational mechanics. *Computational Mechanics*, 33:481–490, 2004.

O.C Zienkiewicz, P. Nithiarasu, R. Codina, M. Vásquez, and P. Ortiz. The characteristic-based-split procedure: an efficient and accurate algorithm for fluid problems. *International Journal for Numerical Methods in Fluids*, 31:359–392, 1999. doi: 10.1002/(SICI)1097-0363(19990915)31:1⟨359::AID-FLD984⟩3.0.CO;2-7.

O.C Zienkiewicz, R.L Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics*. Elsevier, sixth edition, 2005.