Thesis Report

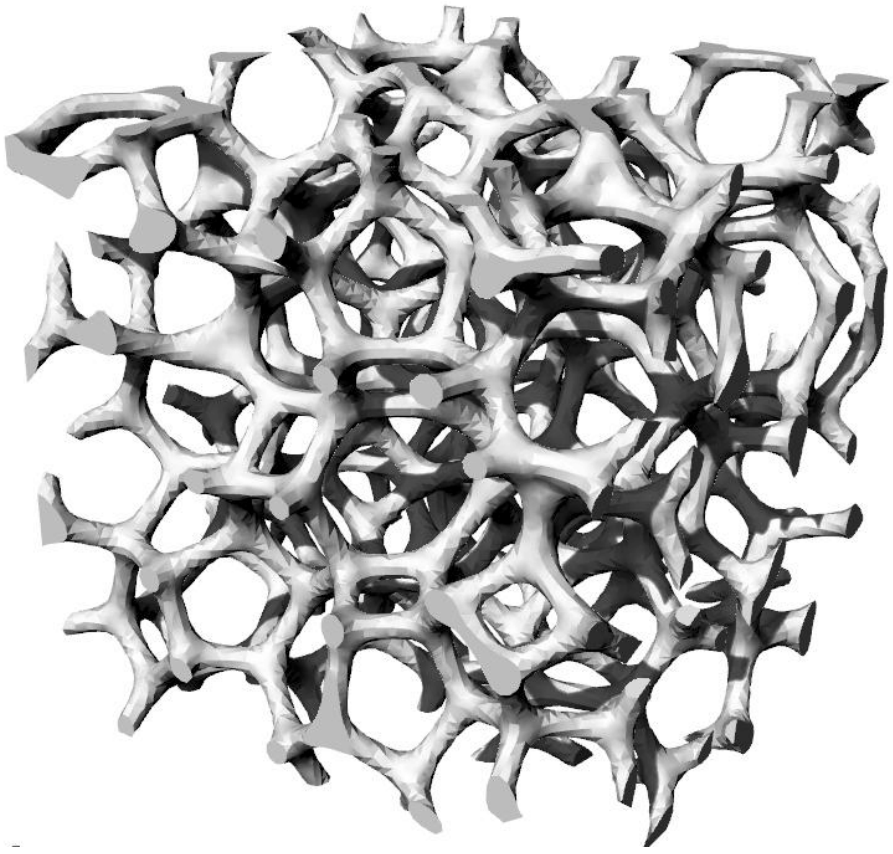
—

Delft University of Technology

Physics-Compatible Numerical Methods for Flow through Porous Media

Author:
Julian Johnston (4581075)

Supervisors:
Oriol Colomés Gené (CiTG)
Deepesh Toshniwal (EEMCS)



TU Delft

Delft, January 2024

Abstract

In this thesis report, we investigate the difference in using physics-compatible elements and standard elements in the context of fluid flow through porous media. For this, we compare the $H(\mathrm{div})$-conforming Raviart-Thomas elements to Lagrange elements. Our model involves a channel flow over a porous media, resulting in a fluid and a porous subdomain. We distinguish between three approaches, as based on a research paper. The first is the one-domain approach called PE, which sets one equation on the entire domain. The other two approaches are two-domain approaches, which use different equations in the two subdomains, with interface conditions to couple them. The first of these is NSD, which couples Navier-Stokes with the Darcy equation. The second is NSF, which does the same but also incorporates the Forchheimer term, which models inertial effects.

In general, we were not able to make a completely successful model, as the PE case did not match the paper, and the results found when using Raviart-Thomas elements had some unexplained anomalies. We were also unable to carry out an accurate error analysis. We were able to see that Raviart-Thomas produced an exactly divergence-free solution, as expected from the theory, at the cost of more computing power and time needed.

Contents

Part I

Theoretical Background

1

Introduction

The modelling of fluid flow through porous media knows many applications, such as groundwater flow and biomedical applications. The equations that govern this type of flow have a certain special property that can be taken advantage of when modelling such a scenario. In this thesis, we will compare two numerical techniques. One of these incorporates this advantage, while the other doesn't. We will make this comparison in the context of a channel flow over a porous medium.

The structure of this report is as follows. We first explain what porous media are and what the governing equations are in section 2. Next, we explain the Finite Element Method and the theory behind physics-compatible elements in section 3. After this, we will go to our implementation. We first introduce the working model in section 4. Then, we discuss the code in section 5. We give the results in section 6. Lastly, we give a conclusion in section 7 and a discussion of possible future research in section 8.

2

Fluid Flow through Porous Media

In this section, we give a description of porous media and the equations that describe fluid flow through these media.

2.1 Description of Porous Media

Porous media are — as the name implies — media with a certain *porosity*. Porosity is measured as the ratio of empty space to solid material: the more empty space a material inhibits, the higher its porosity. At the microscopic scale, porous media have a structure consisting of a complex network of channels, cavities, and pores. An example of what this might look like is given in Figure 2.1.

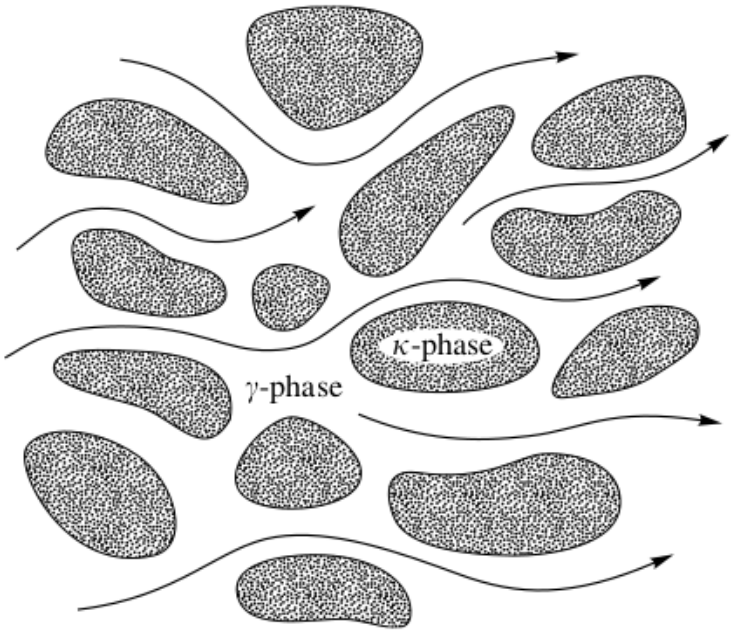


Figure 2.1: Schematic depiction of flow through a porous medium at the microscopic scale (image taken from [1])

This network allows fluid to pass through the medium. We call the flow of fluids through a porous medium *porous flow*. One example of a field of study in which porous media play an important role is groundwater flow. It involves the flow of water through sand, soil, and permeable rock. Another example involving man-made materials is filter design.

Since porous media inhibit such a complex structure, attempting to model fluid flow through such a medium explicitly would be very challenging. One way to model this situation more efficiently is to view the medium at a larger scale, averaging the velocity over the domain. This averaged velocity is called the *seepage velocity*. Because we ignore the inherent structure of the medium, the modelling simplifies significantly.

2.2 Governing Equations

For a fully porous flow, the standard equation to use is *Darcy's law*, also known as the *Darcy equation*. It is named after Henry Darcy, who is often attributed as the first to describe porous flow in an analytical way. As former Chief Director for Water and Pavements in Paris, he experimented with water flowing through a cylinder filled with sand [2]. In this experiment, he found that

$$Q \propto \frac{A\Delta h}{L}, \tag{2.1}$$

where Q is the volumetric flow rate [$\mathrm{m}^3 \cdot \mathrm{s}^{-1}$], A is the cross-sectional area of the cylinder [m^2], L is the length of the cylinder [m], and Δh the height difference between the inlet and outlet [m]. Instead of the height difference, we may also use the pressure difference Δp [Pa] (= [$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-2}$]), as these are proportional through Stevin's law [3] by

$$\Delta p = \rho g \Delta h, \tag{2.2}$$

where ρ is the density [$\mathrm{kg} \cdot \mathrm{m}^{-3}$] and g is the gravitational constant [$\mathrm{m} \cdot \mathrm{s}^{-2}$]. Furthermore, we may define $u = Q/A$ as the seepage velocity [$\mathrm{m} \cdot \mathrm{s}^{-1}$], the averaged velocity through the porous medium. Additionally, we set $\nabla p = \Delta p / L$, the pressure gradient [$\mathrm{kg} \cdot \mathrm{m}^{-2} \cdot \mathrm{s}^{-2}$]. This means that the pressure gradient is proportional to the seepage velocity:

$$\nabla p \propto u. \tag{2.3}$$

Later, the proportionality constant was found [4], so the Darcy equation may be written as

$$-\nabla p = \frac{\mu}{k} u, \tag{2.4}$$

where μ is the dynamic viscosity of the fluid [$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$], and k is the permeability of the porous medium [m^2]. This makes sense intuitively, as a more viscous fluid will flow more slowly than a less viscous fluid in the same circumstances. Similarly, a higher permeability will allow the fluid to pass through more easily.

The permeability of a porous medium is hard to define, but it relates to its porosity and inherent structure. According to [5], k takes the form

$$k = \frac{\phi^3 \delta^2}{a(1-\phi)^2}, \tag{2.5}$$

where ϕ is the porosity of the medium, δ is the characteristic pore size, and a is a constant that models the inherent structure of the porous medium. When $a = 180$, Equation 2.4 may be referred to as the Kozeny-Carman equation, which is a special case for fluid flow through a packed bed of spheres [6, 7, 8]. Though this is a special case, a is often taken as 180, regardless.

The Darcy equation works well for flows with a low *Reynolds number*. This is a dimensionless number that is often used in the context of fluid mechanics, as it works as a gauge for how turbulent the flow is. It is defined by

$$Re = \frac{\rho U L}{\mu}, \tag{2.6}$$

where U is the characteristic velocity [$\mathrm{m} \cdot \mathrm{s}^{-1}$], and L is the characteristic length [m]. In the context of porous flows, L may be replaced by a characteristic pore size δ.

For situations where the flow has a higher Reynolds number, an additional term is introduced to account for inertial forces. This term may be called the Forchheimer term. When this term is added to the Darcy equation (2.4), it is called the *Forchheimer equation* [9], which reads

$$-\nabla p = \frac{\mu}{k} u + \frac{\beta}{\sqrt{k}} \rho |u| u, \tag{2.7}$$

where ρ is the density of the fluid [$\mathrm{kg} \cdot \mathrm{m}^{-3}$], and β is the inertial resistance coefficient [m^{-1}]. The value of β has to be found through experimental analysis.

When considering a non-porous fluid flow, the Stokes equations are useful:

$$\begin{cases} \nabla p = \mu \Delta u, & \text{(2.8a)} \\ \nabla \cdot u = 0, & \text{(2.8b)} \end{cases}$$

where (2.8a) is the momentum equation and (2.8b) the continuity equation. When these equations are juxtaposed to the Darcy equation, so as to model a flow through a clear domain alongside a porous domain, a problem occurs: the orders of the equations are not compatible. In this case, a diffusion term can be added to the Darcy equation, whence we get the *Brinkman equation* [10]:

$$-\nabla p + \mu' \Delta u = \frac{\mu}{k} u, \tag{2.9}$$

where μ' is the effective viscosity. What this coefficient entails exactly is highly debated, but it has been stated that it is expected to be related to the *tortuosity* of the porous medium. This is a rather vague term, but it measures the "windiness" of the channels within the medium. The more a path deviates from a straight line, the higher the tortuosity. However, since an exact definition is not agreed upon, many authors simply take $\mu' = \mu$, and we will do this as well.

An important thing to note about the Brinkman equation (2.9) is that it can be seen as a blend between the Darcy equation (2.4) and the Stokes momentum equation (2.8a). When k grows large, the term $\frac{\mu}{k}u$ goes to zero, and the Stokes momentum equation is recovered. On the other hand, when the value of k is very small, the diffusion term $\mu\Delta u$ can be neglected, so that we get the Darcy equation. For a channel flow over a porous medium, the Brinkman equation could thus be utilised, with a variable k, depending on the y-coordinate.

By the same logic as before, we could also add the Forchheimer term $\frac{\beta}{\sqrt{k}}\rho|u|u$, in case of a flow where inertial effects are important. Other terms to potentially incorporate are a transient and convective term, as if from the Navier-Stokes equations. Putting everything together yields the *transient convective Brinkman-Forchheimer equation*:

$$\rho\left(\frac{\partial u}{\partial t} + (u \cdot \nabla)u\right) - \mu\Delta u + \nabla p = \frac{\mu}{k}u + \frac{\beta}{\sqrt{k}}\rho|u|u, \tag{2.10}$$

where the density ρ is added for dimensional consistency.

2.3 One-Domain vs. Two-Domain

In the case that there is both a porous and a non-porous (also called *free*, *clear*, or *fluid*) subdomain in the domain, there are two approaches possible for modelling: the one-domain approach, and the two-domain approach.

The two-domain approach imposes separate equations on each of the subdomains. For example, one might have the Darcy Equation in the porous domain, and Stokes in the fluid domain. The two domains are then coupled using *interface conditions*. An often used interface condition is the following:

$$\frac{\partial u_f}{\partial y} = \frac{\alpha}{\sqrt{k}}(u_f - u_\infty), \tag{2.11}$$

introduced by Beavers and Joseph [11]. It was found experimentally, but later verified and simplified by Saffman [12]:

$$\frac{\partial u_f}{\partial y} = \frac{\alpha}{\sqrt{k}}u_f + \mathcal{O}(\sqrt{k}). \tag{2.12}$$

In the above, u_f denotes the velocity in the fluid domain, α is some proportionality constant, and u_∞ represents the velocity away from the boundary. This is now called the Beavers-Joseph-Saffman (BJS) condition. Other conditions were summarized by Alazmi and Vafai [13]:

$$(1): \qquad \left.\frac{du}{dy}\right|_+ = \left.\frac{du}{dy}\right|_-, \tag{2.13a}$$

$$(2): \qquad \mu'\left.\frac{du}{dy}\right|_+ = \mu\left.\frac{du}{dy}\right|_-, \tag{2.13b}$$

$$(3): \qquad \frac{\mu}{\phi}\left.\frac{du}{dy}\right|_+ - \mu\left.\frac{du}{dy}\right|_- = \beta\frac{\mu}{\sqrt{k}}u, \tag{2.13c}$$

$$(4): \qquad \frac{\mu}{\phi}\left.\frac{du}{dy}\right|_+ - \mu\left.\frac{du}{dy}\right|_- = \beta\frac{\mu}{\sqrt{k}}u + \beta_2\rho u^2. \tag{2.13d}$$

All these conditions have the additional constraint that $u_+ = u_-$. Here, the subscript + denotes the limit from the fluid side, and − from the porous side of the interface. Conditions (1) and (2) are based on the work of Neale and Nader [14], who considered the BJS-condition to be non-physical, as it imposed a discontinuity into the velocity profile. Conditions (3) and (4) are based on Ochoa-Tapia and Whitaker [15, 16, 17]. The parameters β and β_2 depend on the porous medium at hand and are to be found experimentally.

The one-domain approach views the entire domain as one domain, and thus imposes one equation over its entirety. We already saw an example of this in the Brinkman equation (2.9), with a variable k. Often, a buffer region is implemented around the interface, where any parameters that vary between the two domains undergo a sharp, but smooth, transition. This approach has the advantage that it is usually easier to implement than the two-domain approach, but it generally gives a less accurate representation of the actual situation. An example of this is given in Figure 2.2, where the two-domain Darcy-Stokes approach is compared with the one-domain Brinkman approach:

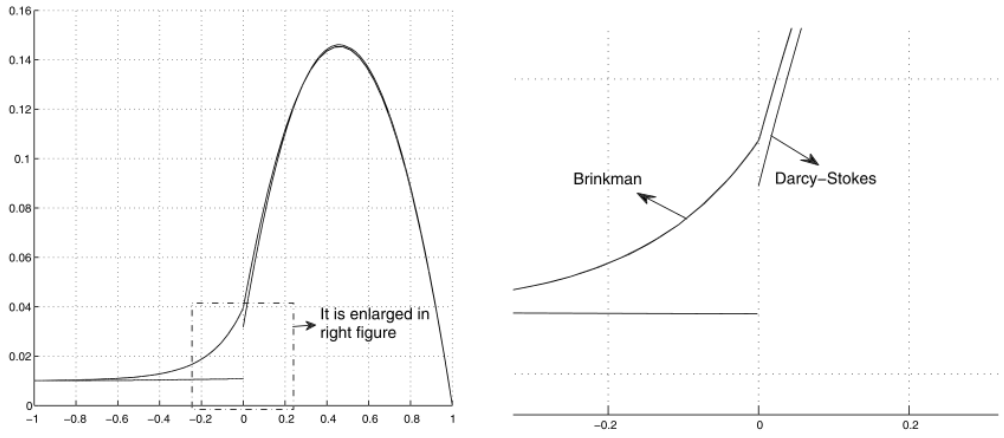


Figure 2.2: Comparison of the flow profile using the one- and two-domain approaches (image taken from [18])

3

Physics-Compatible Finite Elements

The Finite Element Method is an often used method for discretising differential equations in the context of a certain domain. It has the advantage of being easily adaptable to irregularly shaped domains, which simpler methods as the Finite Difference Method lack. In this chapter, we will first give a general overview of the Finite Element Method. After this, we will look at some theoretical analysis regarding well-posedness and convergence theory, specifically in the case of problems with two unknowns. Then, we will use this theory to motivate the use of a certain finite element for tackling the equations we have seen in the previous chapter. We will assume a two-dimensional setting for the entire chapter.

3.1 The Finite Element Method

Let us give a quick walk-through of the Finite Element Method. We will use the Poisson equation in two dimensions as the working example. We assume homogeneous Dirichlet boundary conditions.

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{3.1}$$

where Ω is the domain, which we assume to have a regular boundary $\partial\Omega$. Furthermore, f is the source function, and u is the unknown, which we assume to be in $H^1(\Omega)$. The spaces H^k are called Sobolev spaces, and are defined as

$$H^k = \{u : \partial^\alpha u \in L^2, \forall |\alpha| \leq k\}. \tag{3.2}$$

Here, α is a multiindex and $|\alpha|$ denotes the sum of the elements of α. In our case, α will be two-dimensional, and thus

$$H^1(\Omega) = \{u : u, \partial_x u, \partial_y u \in L^2(\Omega)\}. \tag{3.3}$$

The associated norm is:

$$\|u\|^2_{H^k(\Omega)} = \sum_{|\alpha| \leq k} \|\partial^\alpha u\|^2_{L^2}, \tag{3.4}$$

which, in our example will be

$$\|u\|^2_{H^1(\Omega)} = \|u\|^2_{L^2} + \|\partial_x u\|^2_{L^2} + \|\partial_y u\|^2_{L^2} \tag{3.5}$$

or, equivalently,

$$\|u\|^2_{H^1(\Omega)} = \|u\|^2_{L^2} + \|\nabla u\|^2_{L^2}. \tag{3.6}$$

The first step in the Finite Element Method is to rework the equation into its *weak formulation*. This is done by multiplying the equation by a *test function* and integrating over the domain. We will denote the test function with v and assume that $v \in H^1_0(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$, i.e., the subspace of functions in the search space that vanish at the boundary. Then we get

$$\int_\Omega -\Delta u v \; d\Omega = \int_\Omega f v \; d\Omega. \tag{3.7}$$

We can then use integration by parts on the left hand side integral to reduce the order of the derivative:

$$\begin{aligned} \int_\Omega -\Delta u v \; d\Omega &= \int_\Omega \nabla u \cdot \nabla v \; d\Omega - \int_{\partial\Omega} (\nabla u \cdot \mathbf{n}) v \; d\Omega \\ &= \int_\Omega \nabla u \cdot \nabla v \; d\Omega, \end{aligned} \tag{3.8}$$

by the boundary conditions imposed on v. In the above, $\mathbf{n}$ is the outward pointing normal vector. The weak formulation is then:

Find $u \in H^1(\Omega)$, *such that*

$$\int_\Omega \nabla u \cdot \nabla v \; d\Omega = \int_\Omega f v \; d\Omega \tag{3.9}$$

for all $v \in H_0^1(\Omega)$.

The following step is to discretise the domain. We subdivide the domain into subregions, usually triangles or quadrilaterals, which we call elements. We will assign certain properties to these elements, but since these properties will be the same for each element, we will assign them to a *reference element*, from which the properties can be mapped to the elements in the domain. This idea is what makes the Finite Element Method so suitable for irregularly shaped domains, as we only need to define the properties with regards to the reference element, and the mapping from this element to the elements in the domain. This reference element is usually $\hat{T}_\triangle = ((0,0),(1,0),(0,1))$ for triangles, and $\hat{T}_\square = ((0,0),(1,0),(1,1),(0,1))$ for quadrilaterals.

By Ciarlet [19], a finite element is defined as a triplet $(T, \mathcal{V}, \mathcal{L})$, where

- T is a closed and bounded subset of $\mathbb{R}^2$, with non-empty interior, and piecewise smooth boundary;
- $\mathcal{V}$ is an n-dimensional function space on T;
- $\mathcal{L}$ is the set $\{l_i\}_{i=1}^n$ of degrees of freedom, which form a basis of $\mathcal{V}'$, the dual space of $\mathcal{V}$.

Associated with $\mathcal{L}$ is the set of *basis functions* $\{\phi_i\}_{i=1}^n$ (also known as *shape functions*). It forms a basis for $\mathcal{V}$ and satisfies

$$l_i(\phi_j) = \delta_{ij}, \tag{3.10}$$

the Kronecker delta function. This means that each degree of freedom has an associated basis function which is 1 on the given degree of freedom, and 0 on the others.

In general, we use these basis functions to form an approximation of the unknown function, as follows:

$$u_h = \sum_{i=1}^n u_i \phi_i, \tag{3.11}$$

where the u_i are coefficients to be determined. We substitute this approximation into the weak form that we found before, and for each degree of freedom in the domain, we form an equation where we use the associated basis function as the test function. In our example this will read:

$$\int_\Omega \nabla \left(\sum_{i=1}^n u_i \phi_i \right) \cdot \nabla \phi_j \; d\Omega = \int_\Omega f \phi_j \; d\Omega, \quad \text{for } j = 1, \ldots, n. \tag{3.12}$$

Since the u_i are just constant coefficients, we can rewrite the left hand side integral to get

$$\sum_{i=1}^n u_i \int_\Omega \nabla \phi_i \cdot \nabla \phi_j \; d\Omega = \int_\Omega f \phi_j \; d\Omega, \quad \text{for } j = 1, \ldots, n. \tag{3.13}$$

Since the basis functions and f are known, we can use these equations to construct a matrix and a right-hand side vector. Note that, by the way the basis functions are defined, most entries will be zero, whereby the matrix will be sparse. The resulting matrix-vector equation can be solved by any preferred method. The solution is then calculated from (3.11).

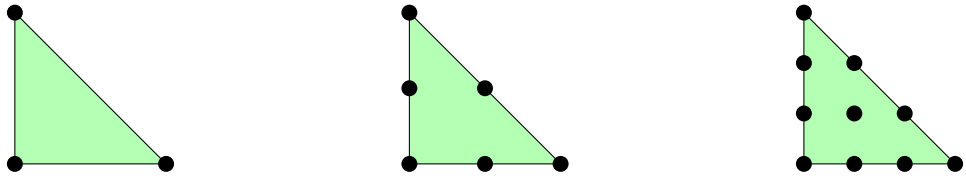
3.2 Lagrange Elements

Let us look at a specific example of an element to better understand the general definition of a finite element as given above. The simplest elements are called Lagrange elements. For these elements, the function space $\mathcal{V}$ is given by $\mathcal{P}_r(T)$, which is the polynomial space up to the chosen order r. For example, for $r = 1$, the space is

$$\mathcal{V} = \text{span}\{1, x, y\}. \tag{3.14}$$

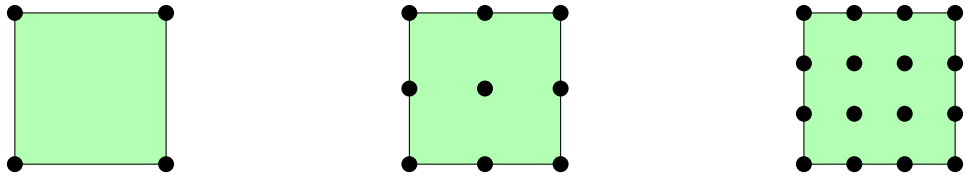
We will denote the Lagrange function space of degree r with P_r. Thereby, we will sometimes use the name "P_r elements" for Lagrange elements of order r. Furthermore, in a mixed setting, we say we use, for example, $P_r - P_{r-1}$ elements, when we use Lagrange elements of order r for the velocity, and of order $r-1$ for the pressure.

The degrees of freedom, and thereby the basis functions, depend on the order, and on whether we use triangles or quadrilaterals. For $r = 1$, they lie on the vertices of the element. For higher r, the progression is made clear in Figure 3.1 and 3.2.



(a) $r = 1$ (b) $r = 2$ (c) $r = 3$

Figure 3.1: Triangular Lagrange Elements of orders $r = 1, 2, 3$.



(a) $r = 1$ (b) $r = 2$ (c) $r = 3$

Figure 3.2: Quadrilateral Lagrange Elements of orders $r = 1, 2, 3$.

Let us now take $r = 1$, to illustrate the basis functions. Recall that the basis functions need to satisfy the given condition $l_i(\phi_i) = \delta_{ij}$. This means that each basis function will be 1 on its associated degree of freedom (vertex of the element), and zero on the others. This is illustrated below in Figure 3.3 and 3.4.

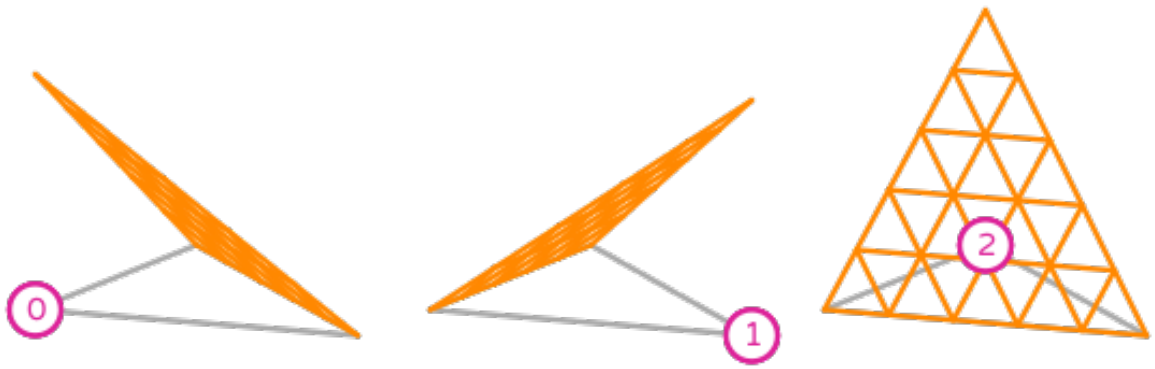


Figure 3.3: Triangular Lagrange basis functions (images taken from [20])

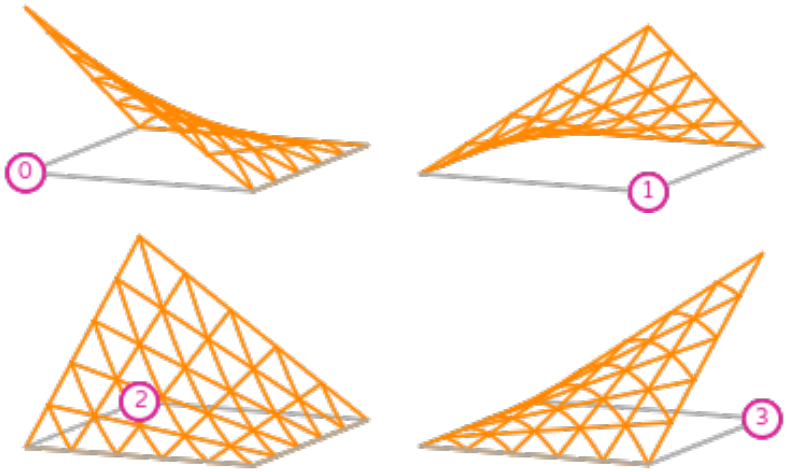


Figure 3.4: Quadrilateral Lagrange basis functions (images taken from [20])

Explicitly, the basis functions of order 1 on $T_\triangle$ are given by:

$$\phi_0 = 1 - x - y, \tag{3.15a}$$
$$\phi_1 = x, \tag{3.15b}$$
$$\phi_2 = y. \tag{3.15c}$$

The basis functions on $T_\square$ are given by:

$$\phi_0 = (1-x)(1-y), \tag{3.16a}$$
$$\phi_1 = x(1-y), \tag{3.16b}$$
$$\phi_2 = (1-x)y, \tag{3.16c}$$
$$\phi_3 = xy. \tag{3.16d}$$

3.3 Lax-Milgram for Mixed Variational Problems

Something that we conveniently left out in the derivation of the weak form is that it is not guaranteed to have a unique solution, just because the strong form does. The Lax-Milgram theorem, a fundamental theorem of variational calculus, aids us with this problem. It states: given a Hilbert space V and a bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ that satisfies

$$\textbf{continuity}: \quad \exists C < \infty: \quad |a(u,v)| \leq C\|u\|_V\|v\|_V \quad \forall u, v \in V, \tag{3.17a}$$
$$\textbf{coercivity}: \quad \exists \alpha > 0: \quad a(v,v) \geq \alpha\|v\|_V^2, \quad \forall v \in V, \tag{3.17b}$$

there exists a unique solution u to the following variational problem:

Find $u \in V$ such that

$$a(u,v) = F(v), \quad \forall v \in V, \tag{3.18}$$

where F is continuous in V'.

To illustrate this theorem, we can use the variational form of the Poisson equation (3.9) that we derived before. Take

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \; d\Omega \tag{3.19}$$

and

$$F(v) = \int_\Omega fv \; d\Omega. \tag{3.20}$$

Continuity is easily shown through the Cauchy-Schwarz inequality (A.1):

$$|a(u,v)| \leq \|u\|_{L^2}\|v\|_{L^2} \leq \|u\|_{H^1}\|v\|_{H^1}. \tag{3.21}$$

Coercivity can be shown using Poincaré's inequality (A.2), since we are looking at a bounded domain:

$$\|u\|_{H^1}^2 = \|u\|_{L^2}^2 + \|\nabla u\|_{L^2}^2 \leq (C_P^2 + 1)\|\nabla u\|_{L^2}^2 = (C_P^2 + 1)a(u,u), \tag{3.22}$$

where C_P is the Poincaré constant. Thus

$$a(u,u) \geq \frac{1}{C_P^2 + 1}\|u\|_{H^1}^2 = \alpha\|u\|_{H^1}^2. \tag{3.23}$$

Hereby, the variational Poisson equation has a unique solution.

The Lax-Milgram theorem can be extended to variational forms involving multiple spaces, which we call *mixed* variational forms. In particular, we look at the following form, discussed by Brenner and Scott [21]:

Find $u \in V$ and $p \in \Pi$ *such that*

$$\begin{cases} a(u,v) + b(v,p) = F(v), & \forall v \in V, \\ b(u,q) = 0, & \forall q \in \Pi, \end{cases} \tag{3.24}$$

with F continuous in V'. Note that the variational form of the Darcy equation (2.4) equipped with the continuity equation $\nabla \cdot \mathbf{u} = 0$ can be written in this form, with

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \frac{k}{\mu}\mathbf{u} \cdot \mathbf{v} \; d\Omega, \tag{3.25}$$

$$b(\mathbf{v}, p) = -\int_\Omega p(\nabla \cdot \mathbf{v}) \; d\Omega, \tag{3.26}$$

and $F(\mathbf{v}) = 0$.

The spaces V and Π can be linked with an operator $\mathcal{D} : V \to \Pi$, that we assume to be continuous:

$$\|\mathcal{D}v\|_\Pi \leq C\|v\|_V, \tag{3.27}$$

and we say that

$$b(v,p) = (\mathcal{D}v, p)_\Pi. \tag{3.28}$$

Let us assume that $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ and $b(\cdot,\cdot) : V \times \Pi \to \mathbb{R}$ are continuous:

$$\begin{aligned} a(u,v) &\leq C\|u\|_V\|v\|_V, \quad \forall u, v \in V, \\ b(v,p) &\leq C\|v\|_V\|p\|_\Pi, \quad \forall v \in V, p \in \Pi. \end{aligned} \tag{3.29}$$

If we define the space

$$Z = \{v \in V : b(v,q) = 0, \forall q \in \Pi\}, \tag{3.30}$$

then u can be found through the following variational form:

Find $u \in Z$, *such that*

$$a(u,v) = F(v), \quad \forall v \in Z. \tag{3.31}$$

If $a(\cdot,\cdot)$ is then coercive on Z, then the conditions for Lax-Milgram are satisfied, and thus u is a unique solution. Having found u, we can find p by the following:

Find $p \in \Pi$ *such that*

$$b(v,p) = -a(u,v) + F(v), \quad \forall v \in V. \tag{3.32}$$

Since $b(\cdot,\cdot)$ is defined on two different Hilbert spaces, the definition of coercivity we gave above cannot apply. Note, however, that the coercivity condition can be rewritten as

$$\alpha\|v\|_V \leq \sup_{w \in V} \frac{a(w,v)}{\|w\|_V}, \quad \forall v \in V, \tag{3.33}$$

by taking $w = v$. This idea can be used to define a type of coercivity on $b(\cdot,\cdot)$:

$$\beta\|p\|_\Pi \leq \sup_{w \in V} \frac{b(w,p)}{\|w\|_V}, \quad \forall p \in \Pi. \tag{3.34}$$

This condition is known as the Ladyzhenka-Babuška-Brezzi condition, or the inf-sup condition. With this condition in place, one can prove that there exists a unique solution p for the variational form (3.32).

So, to summarize: if we are given two Hilbert spaces V and Π, the bilinear forms $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ and $b(\cdot,\cdot) : V \times \Pi \to \mathbb{R}$ are continuous, $a(\cdot,\cdot)$ is coercive on Z, and $b(\cdot,\cdot)$ satisfies condition (3.34), then there exist unique solutions u and p for the mixed variational form (3.24).

3.4 Well-Posedness in The Discrete Setting

Let us now discretise this abstract setting. We take $V_h \subseteq V$ and $\Pi_h \subseteq \Pi$, and we define

$$Z_h = \{v \in V_h : b(v, q) = 0, \forall q \in \Pi_h\}. \tag{3.35}$$

The discrete variational form then reads:

Find $u_h \in V_h$ and $p_h \in \Pi_h$ *such that*

$$\begin{cases} a(u_h, v) + b(v, p_h) = F(v), & \forall v \in V_h, \\ b(u_h, q) = 0, & \forall q \in \Pi_h, \end{cases} \tag{3.36}$$

or equivalently:

Find $u_h \in Z_h$ *such that*

$$a(u_h, v) = F(v), \quad \forall v \in Z_h, \tag{3.37}$$

and then find $p_h \in \Pi_h$ *such that*

$$b(v, p_h) = -a(u_h, v) + F(v), \quad \forall v \in V_h. \tag{3.38}$$

Because we will not encounter a case where it is not so, we will assume that $Z_h \subseteq Z$. Then, if $a(\cdot, \cdot)$ is coercive on Z, we can use Céa's lemma (A.3) to see that

$$\|u - u_h\|_V \leq \frac{C}{\alpha} \inf_{v \in Z_h} \|u - v\|_V. \tag{3.39}$$

This means that the error $u - u_h$ depends only on how well Z_h approximates Z, and is stable as long as $a(\cdot, \cdot)$ is continuous and coercive.

Now, in order for p_h to be a unique solution of (3.38), the discrete inf-sup condition needs to be satisfied:

$$0 < \beta := \inf_{q \in \Pi_h} \sup_{v \in V_h} \frac{|b(v, q)|}{\|v\|_V \|q\|_\Pi}. \tag{3.40}$$

Given that u, p, and u_h are determined, this p_h can be proven to satisfy

$$\|p - p_h\|_\Pi \leq \frac{C}{\beta} \|u - u_h\|_V + \left(1 + \frac{C}{\beta}\right) \inf_{q \in \Pi_h} \|p - q\|_\Pi, \tag{3.41}$$

so the error $p - p_h$ depends on the approximability of both spaces V and Π.

We note that the theory discussed here relies on linearity of the problem. The inclusion of the Forchheimer term and convective term as in (2.10) introduces non-linearity to the problem. We will simply assume that the theory above also holds in case of non-linearity, because the actual analysis is beyond the scope of this thesis report.

3.5 Raviart-Thomas Elements

Let us now recall the Stokes equations from before:

$$\begin{cases} \nabla p = \mu \Delta \mathbf{u}, & \text{(3.42a)} \\ \nabla \cdot \mathbf{u} = 0. & \text{(3.42b)} \end{cases}$$

For compatibility reasons, we assume the boundary condition

$$p\mathbf{n} - \mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = \mathbf{0}, \qquad \text{on } \partial\Omega, \tag{3.43}$$

with $\mathbf{n}$ the outward-pointing normal. We will use these equations to illustrate the so-called Raviart-Thomas elements. These equations have a certain property that Raviart-Thomas elements, and the class of elements they belong to, were specifically designed to deal with.

For the two unknowns, we assume that $p \in L^2(\Omega)$ and that $\mathbf{u}$ is in the special space

$$H(\mathrm{div};\Omega) = \{\mathbf{v} : \mathbf{v} \in (L^2(\Omega))^2, \nabla \cdot \mathbf{v} \in L^2(\Omega)\}. \tag{3.44}$$

In words, this space encompasses the two-dimensional vector functions of which each component is square-integrable, and the divergence of the entire vector is as well. The norm associated with this space is given by

$$\|\mathbf{u}\|^2_{H(\mathrm{div})} = \|\mathbf{u}\|^2_{L^2} + \|\nabla \cdot \mathbf{u}\|^2_{L^2} \tag{3.45}$$

The weak form of the above equations is given by:

Find $p \in L^2(\Omega)$ *and* $\mathbf{u} \in H(\mathrm{div};\Omega)$ *such that*

$$\begin{aligned} \int_\Omega \mu \nabla \mathbf{u} : \nabla \mathbf{v} \; d\Omega - \int_\Omega p(\nabla \cdot \mathbf{v}) \; d\Omega = 0, \qquad & \forall \mathbf{v} \in H_0(\mathrm{div};\Omega), \\ \int_\Omega q(\nabla \cdot \mathbf{u}) \; d\Omega = 0, \qquad & \forall q \in L^2(\Omega), \end{aligned} \tag{3.46}$$

where

$$H_0(\mathrm{div};\Omega) = \{\mathbf{v} : \mathbf{v} \in H(\mathrm{div};\Omega), \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}. \tag{3.47}$$

This fits the abstract variational form (3.24), with

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \mu \nabla \mathbf{u} : \nabla \mathbf{v} \; d\Omega \tag{3.48}$$

and

$$b(\mathbf{v}, p) = -\int_\Omega p(\nabla \cdot \mathbf{v}) \; d\Omega. \tag{3.49}$$

The kernel of $b(\cdot,\cdot)$, as derived from the definition in (3.30), is here given by

$$Z = \{\mathbf{v} \in H(\mathrm{div};\Omega) : \nabla \cdot \mathbf{v} = 0\}, \tag{3.50}$$

whereby $\|\mathbf{v}\|_{H(\mathrm{div})} = \|\mathbf{v}\|_{L^2}$, for all $\mathbf{v} \in Z$. By this fact, it is easy to see that $a(\cdot,\cdot)$ is coercive on Z.

Furthermore, from the bilinear form $b(\cdot,\cdot)$ we see that the operator $\mathcal{D}$ from (3.28) between the velocity space and the pressure space is given by

$$\mathcal{D}v = -\nabla \cdot v. \tag{3.51}$$

By construction of the space $H(\mathrm{div})$, this mapping $\mathcal{D} : H(\mathrm{div}) \to L^2$ is *surjective*, which means that for every $q \in L^2$, there exists a $\mathbf{v} \in H(\mathrm{div})$ such that $\nabla \cdot \mathbf{v} = q$. Hereby, the inf-sup condition (3.34) is satisfied. Continuity is also easily shown, so the problem is well-posed.

The class of elements which have a function space that discretises the space $H(\mathrm{div})$ is called the class of $H(\mathrm{div})$-conforming elements, which lies in the more general class of physics-compatible elements. The Raviart-Thomas elements are the most well-known finite elements that lie in this class.

Firstly, the function space for the Raviart-Thomas elements is given by the set of all vector functions where each component is a polynomial of degree at most r in the corresponding coordinate. Denoting the function space of degree r by RT_r, we could write this as

$$RT_r = (\mathcal{P}_{r-1}(T))^2 + \mathbf{x} \circ (\mathcal{P}_{r-1}(T))^2, \tag{3.52}$$

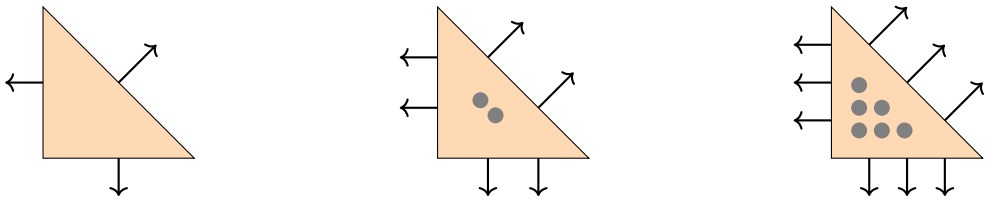
where $\circ$ denotes element-wise multiplication and $\mathbf{x}$ is the vector $(x, y)^\top$. It is also often written as

$$RT_r = \mathcal{P}_{r,r-1}(T) \times \mathcal{P}_{r-1,r}(T), \tag{3.53}$$

where $\mathcal{P}_{p,q}(T)$ denotes the space of polynomial functions on T of degree at most p in x and at most q in y. For $r = 1$,

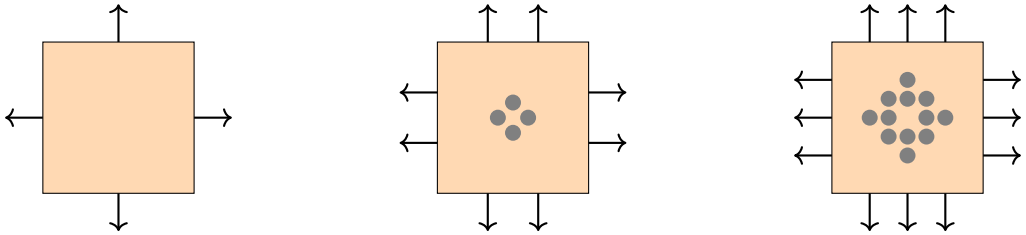
$$RT_1 = \mathrm{span}\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix} \right\}. \tag{3.54}$$

Secondly, these elements have degrees of freedom in the normal component on the edges of the element, as illustrated in Figure 3.5 and 3.6. This ensures inter-elemental continuity in the normal component. The grey discs, present for higher orders, represent the internal moments.



(a) $r = 1$ (b) $r = 2$ (c) $r = 3$

Figure 3.5: Triangular Raviart-Thomas Elements of orders $r = 1, 2, 3$.



(a) $r = 1$ (b) $r = 2$ (c) $r = 3$

Figure 3.6: Quadrilateral Raviart-Thomas Elements of orders $r = 1, 2, 3$.

The basis functions for order 1 are then given as in Figure 3.7 and 3.8.

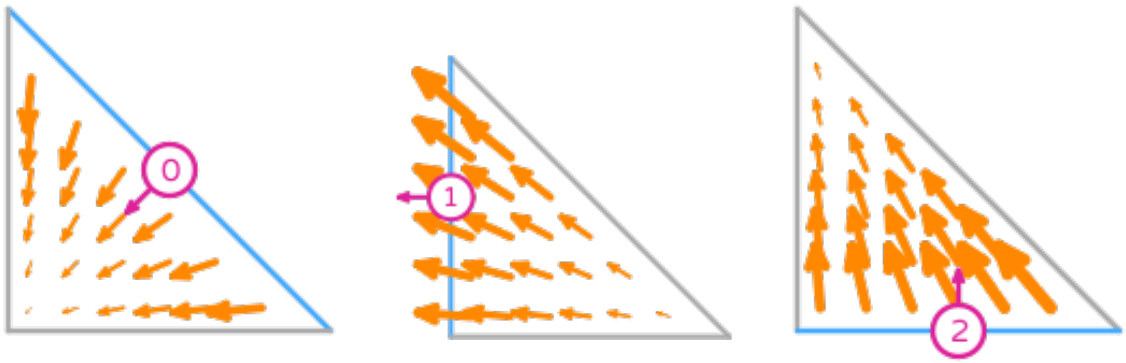


Figure 3.7: Triangular Raviart-Thomas basis functions (images taken from [20])

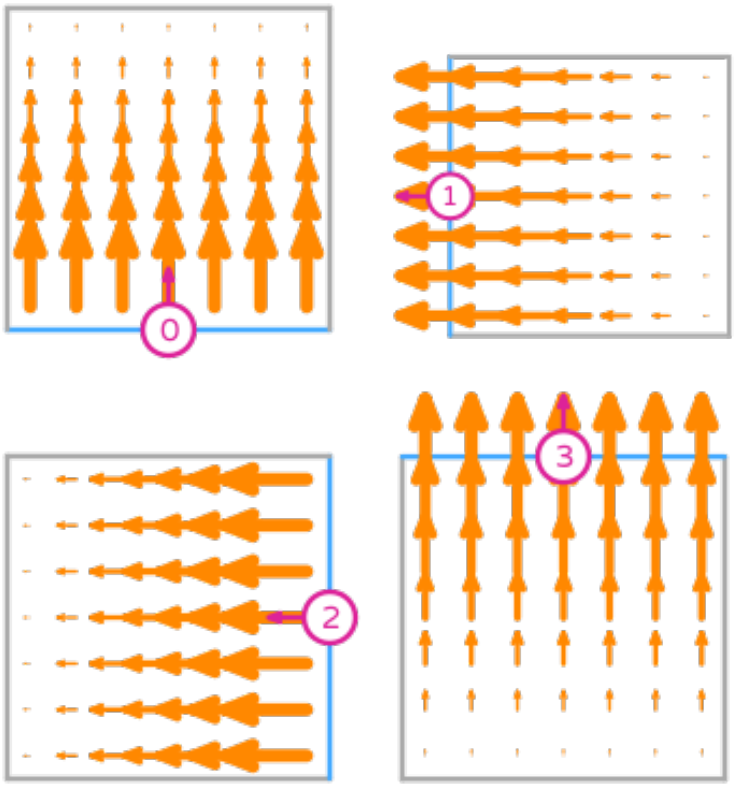


Figure 3.8: Quadrilateral Raviart-Thomas basis functions (images taken from [20])

Explicitly, on $T_\triangle$, the basis functions of order 1 are given by:

$$\phi_0 = \begin{pmatrix} -x \\ -y \end{pmatrix}, \tag{3.55a}$$

$$\phi_1 = \begin{pmatrix} x-1 \\ y \end{pmatrix}, \tag{3.55b}$$

$$\phi_2 = \begin{pmatrix} -x \\ 1-y \end{pmatrix}. \tag{3.55c}$$

On $T_\square$, they are

$$\phi_0 = \begin{pmatrix} 0 \\ 1-y \end{pmatrix}, \tag{3.56a}$$

$$\phi_1 = \begin{pmatrix} x-1 \\ 0 \end{pmatrix}, \tag{3.56b}$$

$$\phi_2 = \begin{pmatrix} -x \\ 0 \end{pmatrix}, \tag{3.56c}$$

$$\phi_3 = \begin{pmatrix} 0 \\ y \end{pmatrix}. \tag{3.56d}$$

The important thing now is that the mapping between the discrete spaces $\mathcal{D} = -\nabla\cdot : RT_r \to P_{r-1}$ is surjective, just as between the continuous spaces that they discretise. This means that the discrete inf-sup condition (3.40) is satisfied. Therefore, if we use RT_r for the discrete velocity space, and P_{r-1} for the discrete pressure space, the discrete variational problem is well-posed.

Part II

Implementation and Results

4 Working Model

In this chapter we will define the situation that we are modelling. We base our implementation on a paper by Cimolin and Discacciati [22], in which three approaches are compared for modelling a channel flow over a porous domain. We will use their results as a comparison of our own results.

4.1 Model definition

The domain Ω consists of a channel Ω_f with fluid flowing over a porous domain Ω_p. The different parts of the boundary and the interface are named as visualised in Figure 4.1 below:

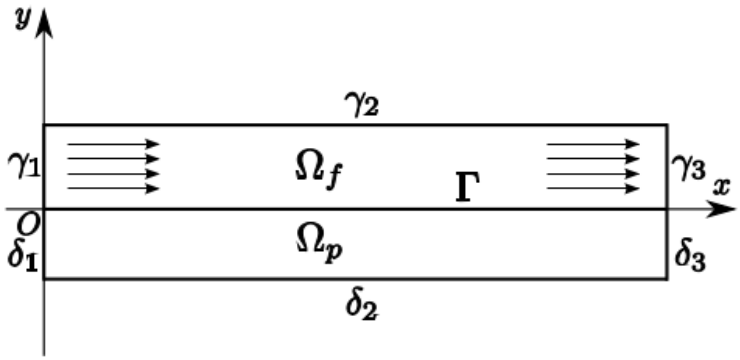


Figure 4.1: Domain used in this test-case (picture taken from [22])

If we need to unite multiple parts of the same boundary, we will use the simple notation as in the following example:

$$\gamma_{1,2,3} := \gamma_1 \cup \gamma_2 \cup \gamma_3. \tag{4.1}$$

We also use the following straightforward notations for the boundaries:

$$\partial\Omega := \gamma_{1,2,3} \cup \delta_{1,2,3}, \tag{4.2a}$$
$$\partial\Omega_f := \gamma_{1,2,3} \cup \Gamma, \tag{4.2b}$$
$$\partial\Omega_p := \delta_{1,2,3} \cup \Gamma. \tag{4.2c}$$

4.1.1 PE

The first way of modelling is a one-domain approach termed by the authors as the Penalisation method, abbreviated by PE. It reads:

$$\begin{cases} \dfrac{\partial u}{\partial t} + (u \cdot \nabla)u - \dfrac{1}{Re}\Delta u + \nabla p + Gr_v u + Gr_i |u| u \chi_{\Omega_p} = 0, & \text{(4.3a)} \\ \nabla \cdot u = 0. & \text{(4.3b)} \end{cases}$$

We note that all quantities have been made dimensionless by utilising U and L, the characteristic values for the velocity and the length, respectively. We have two dimensionless constants:

$$Gr_v = \frac{\mu L}{\rho U k}\chi_{\Omega_p}, \tag{4.4a}$$

$$Gr_i = \frac{C_F L}{\sqrt{k}}\chi_{\Omega_p}, \tag{4.4b}$$

with χ_{Ω_p} the characteristic function which is defined to be 1 in Ω_p and 0 elsewhere. Thereby, the Darcy and Forchheimer terms are only "active" in the porous domain. Furthermore, we have the Reynolds number, defined by

$$Re = \frac{\rho U L}{\mu}. \tag{4.5}$$

The boundary conditions for the PE case are given by

$$\begin{cases} u = u_{\text{pois}} & \text{on } \gamma_1, & (4.6a)\\ u = 0 & \text{on } \gamma_2, & (4.6b)\\ pn - \dfrac{1}{Re}\dfrac{\partial u}{\partial n} = 0 & \text{on } \gamma_3, & (4.6c)\\ u \cdot n = 0 & \text{on } \delta_{1,2}, & (4.6d)\\ \left(\dfrac{\partial u}{\partial n}\right)_\tau = 0 & \text{on } \delta_{1,2}, & (4.6e)\\ pn - \dfrac{1}{Re}\dfrac{\partial u}{\partial n} = 0 & \text{on } \delta_3. & (4.6f) \end{cases}$$

In (4.6a), u_{pois} is the inlet velocity, which is defined as a Poiseuille flow, given by

$$u_{\text{pois}}(x, y) = (y(h_f - y), 0), \tag{4.7}$$

with h_f the height of the fluid domain. The subscript τ in (4.6e) indicates the tangential component of the vector which is formally defined by

$$(v)_\tau = v - (v \cdot n)n. \tag{4.8}$$

Note that $(v \cdot n)n$ is the normal part of the vector.

4.1.2 NSD

The second way of modelling the flow is with the two-domain approach of coupling Navier-Stokes with Darcy, abbreviated as NSD, reading:

$$\begin{cases} \dfrac{\partial u_f}{\partial t} + (u_f \cdot \nabla)u_f - \dfrac{1}{Re_f}\Delta u_f + \nabla p_f = 0 & \text{in } \Omega_f, & (4.9a)\\ \nabla \cdot u_f = 0 & \text{in } \Omega_f, & (4.9b)\\ -\nabla \cdot (Gr_n \nabla p_p) = 0 & \text{in } \Omega_p. & (4.9c) \end{cases}$$

Here, the subscripts f and p denote quantities relevant to the fluid domain and porous domain, respectively. Re_f is again the Reynolds number, but specific to the fluid domain. It is also possible to define a porous Reynolds number by

$$Re_p = \frac{\rho U \delta}{\mu}, \tag{4.10}$$

but we will not use this. Gr_n is another dimensionless constant, defined by

$$Gr_n = \frac{\rho K U}{\mu L}. \tag{4.11}$$

Note that u_p is not present in (4.9). The subequation (4.9c) inherently encompasses the (dimensionless) Darcy equation equipped with the continuity equation:

$$\begin{cases} -Gr_n \nabla p_p = u_p, & (4.12a)\\ \nabla \cdot u_p = 0. & (4.12b) \end{cases}$$

In the paper, the velocity u_p is recovered from p_p using Darcy's law (4.12a) as a postprocessing step.

Since this is a two-domain approach, we need interface conditions. They are given by

$$\begin{cases} u_f \cdot n_f = Gr_n \dfrac{\partial p_p}{\partial n_p} & \text{on } \Gamma, \quad (4.13a) \\ p_f - \dfrac{1}{Re_f} \dfrac{\partial u_f}{\partial n_f} \cdot n_f = p_p & \text{on } \Gamma, \quad (4.13b) \\ -\left(\dfrac{\partial u_f}{\partial n_f} \right)_\tau = Gr_c (u_f)_\tau & \text{on } \Gamma. \quad (4.13c) \end{cases}$$

Here, n_f and n_p are the normals on the interface on the fluid and porous side, respectively. Furthermore,

$$Gr_c = \frac{\alpha_{BJ} L}{\sqrt{k}}, \tag{4.14}$$

where α_{BJ} is the parameter from the Beavers-Joseph interface condition.

The first interface condition (4.13a) represents the continuity of the normal component of the velocity, rewritten by Darcy's law (4.12a). The second condition (4.13b) represents the continuity of the normal stresses on the interface. Finally, (4.13c) is the BJS-condition.

Besides interface conditions, we also need boundary conditions. They are given by

$$\begin{cases} u_f = u_{\text{pois}} & \text{on } \gamma_1, \quad (4.15a) \\ u_f = 0 & \text{on } \gamma_2, \quad (4.15b) \\ p_f n_f - \dfrac{1}{Re_f} \dfrac{\partial u_f}{\partial n_f} = 0 & \text{on } \gamma_3, \quad (4.15c) \\ \dfrac{\partial p_p}{\partial n_p} = 0 & \text{on } \delta_{1,2}, \quad (4.15d) \\ p_p = 0 & \text{on } \delta_3. \quad (4.15e) \end{cases}$$

4.1.3 NSF

The final way of modelling the flow is with the inclusion of the Forchheimer term, abbreviated as NSF, reading:

$$\begin{cases} \dfrac{\partial u_f}{\partial t} + (u_f \cdot \nabla) u_f - \dfrac{1}{Re_f} \Delta u_f + \nabla p_f = 0 & \text{in } \Omega_f, \quad (4.16a) \\ \nabla \cdot u_f = 0 & \text{in } \Omega_f, \quad (4.16b) \\ u_p + Gr_f u_p |u_p| = -Gr_n \nabla p_p = 0 & \text{in } \Omega_p, \quad (4.16c) \\ \nabla \cdot u_p = 0 & \text{in } \Omega_p, \quad (4.16d) \end{cases}$$

with

$$Gr_f = \frac{\rho C_F U \sqrt{k}}{\mu}, \tag{4.17}$$

where C_F is the inertial resistance coefficient from the Forchheimer equation (2.7). The interface conditions are the same as for the NSD case, except that (4.13a) has been replaced with

$$u_f \cdot n_f = -u_p \cdot n_p. \tag{4.18}$$

The boundary conditions are also the same, except that (4.15d) is replaced with

$$u_p \cdot n_p = 0, \qquad \text{on } \delta_{1,2}. \tag{4.19}$$

4.1.4 Parameter specification

In the specific situation that is considered, the domain is 50 mm long in the x-direction, and 7 mm in the y-direction, where the porous domain is 3 mm high and the fluid domain 4 mm. The parameters are chosen as follows: $\rho = 1.184 \text{ kg} \cdot \text{m}^{-3}, \mu = 1.855 \cdot 10^{-5} \text{ Pa} \cdot \text{s}, k = 3.71 \cdot 10^{-7} \text{ m}^2, \alpha_{BJ} = 1.0, C_F = 0.5$. The characteristic values are $L = 10^{-3}$ m (height of the fluid domain) and $U = 10^{-1} \text{ m} \cdot \text{s}^{-1}$ (maximal inlet velocity). Hereby, the dimensionless values are $Re_f = 6.38, Gr_n = 2.37, Gr_f = 1.94, Gr_c = 1.64, Gr_\nu = 0.42$, and $Gr_i = 0.82$.

4.2 Weak forms

Let us now derive the weak forms for the given formulations. They are found by multiplying the equations by test functions and integrating over the domain. The interface conditions and boundary conditions are also incorporated.

4.2.1 PE

First, let's look at the PE equations (4.3). We define the trial and test spaces as follows:

$$X = (H^1(\Omega))^2, \tag{4.20a}$$
$$V = \{v \in X : v = 0 \text{ on } \gamma_{1,2}, v \cdot n = 0 \text{ on } \delta_{1,2}\}, \tag{4.20b}$$
$$Q = L^2(\Omega). \tag{4.20c}$$

Note that Q is both the test and trial space for the pressure. We multiply the equations with $v \in V$ and $q \in Q$, respectively, and integrate over the domain:

$$\begin{cases} \displaystyle\int_\Omega \frac{\partial u}{\partial t} \cdot v + ((u \cdot \nabla)u) \cdot v - \frac{1}{Re}\Delta u \cdot v + \nabla p \cdot v + Gr_\nu u \cdot v + Gr_i |u| u \cdot v \; d\Omega = 0, & \text{(4.21a)} \\ \displaystyle\int_\Omega (\nabla \cdot u) q \; d\Omega = 0, & \text{(4.21b)} \end{cases}$$

We integrate the third term of (4.21a) by parts:

$$\int_\Omega -\frac{1}{Re}\Delta u \cdot v \; d\Omega = \int_\Omega \frac{1}{Re}\nabla u \cdot \nabla v \; d\Omega - \int_{\partial\Omega} \frac{1}{Re} v \cdot \frac{\partial u}{\partial n} \; dS. \tag{4.22}$$

We do the same for the fourth term:

$$\int_\Omega \nabla p \cdot v \; d\Omega = -\int_\Omega p(\nabla \cdot v) \; d\Omega + \int_{\partial\Omega} p(v \cdot n) \; dS. \tag{4.23}$$

We combine the boundary integrals into one and rearrange a bit:

$$\int_{\partial\Omega} p(v \cdot n) - \frac{1}{Re} v \cdot \frac{\partial u}{\partial n} \; dS = \int_{\partial\Omega} v \cdot \left(pn - \frac{1}{Re}\frac{\partial u}{\partial n} \right) dS. \tag{4.24}$$

Now, we want to eliminate this boundary integral from the equation. As each part of the boundary has separate boundary conditions, we have to look at each part separately. For γ_3 and δ_3, it is easy to see that we can use the boundary conditions (4.6c) and (4.6f). Next, in the space V we have enforced that $v = 0$ on $\gamma_{1,2}$, whereby the integral vanishes there also. We have also enforced in V that $v \cdot n = 0$ on $\delta_{1,2}$. In order to use this we have to be a bit clever. We return to the left-hand side of (4.24) and note that the $p(v \cdot n)$ term vanishes immediately. What we are left with, we can decompose into the tangential and normal component, by (4.8):

$$\int_{\delta_{1,2}} -\frac{1}{Re} v \cdot \frac{\partial u}{\partial n} \; dS = \int_{\delta_{1,2}} -\frac{1}{Re} v \cdot \left(\left(\frac{\partial u}{\partial n} \right)_\tau + \left(\frac{\partial u}{\partial n} \cdot n \right) n \right) dS \tag{4.25}$$
$$= \int_{\delta_{1,2}} -\frac{1}{Re} \left(v \cdot \left(\frac{\partial u}{\partial n} \right)_\tau + (v \cdot n) \left(\frac{\partial u}{\partial n} \cdot n \right) \right) dS \tag{4.26}$$

Now we can use the boundary condition (4.6e) and the enforcement of $v \cdot n = 0$ to eliminate the boundary integral. The boundary conditions that we did not incorporate implicitly will have to be enforced strongly.

In order to discretise the resulting weak form, we define the following spaces:

$$X_h = \{v \in C^0(\bar{\Omega}) : v|_T \in (\mathcal{P}_2(T))^2 \; \forall T \in \mathcal{T}_h\}, \tag{4.27a}$$
$$V_h = \{v \in X_h : v = 0 \text{ on } \gamma_{1,2}, v \cdot n = 0 \text{ on } \delta_{1,2}\}, \tag{4.27b}$$
$$Q_h = \{q \in C^0(\bar{\Omega}) : q|_T \in \mathcal{P}_1(T) \; \forall T \in \mathcal{T}_h\} \tag{4.27c}$$

Here, $\mathcal{T}_h$ represents the discretisation of Ω. The discretised weak form of the PE case then reads as below:

PE weak form

Find $u_h \in X_h$ ***and*** $p_h \in Q_h$ ***such that***

$$\int_\Omega \frac{\partial u_h}{\partial t} \cdot v_h + ((u_h \cdot \nabla)u_h) \cdot v_h + \frac{1}{Re}\nabla u_h \cdot \nabla v_h - p_h \nabla \cdot v_h + Gr_\nu u_h \cdot v_h + Gr_i |u_h| u_h \cdot v_h \; d\Omega = 0, \tag{4.28a}$$

$$\int_\Omega q_h \nabla \cdot u_h \; d\Omega = 0, \tag{4.28b}$$

for all $v_h \in V_h$***, and*** $q_h \in Q_h$***, with strong boundary conditions***

$$u_h = u_{\text{pois}} \quad \textbf{on } \gamma_1, \tag{4.29a}$$

$$u_h = 0 \quad \textbf{on } \gamma_2, \tag{4.29b}$$

$$u_h \cdot n = 0 \quad \textbf{on } \delta_{1,2}. \tag{4.29c}$$

4.2.2 NSD

Now, we tackle the NSD equations. The spaces we use are defined as follows:

$$X_f = (H^1(\Omega_f))^2, \tag{4.30a}$$

$$V_f = \{v \in X_f : v = 0 \text{ on } \gamma_{1,2}\}, \tag{4.30b}$$

$$Q_f = L^2(\Omega_f), \tag{4.30c}$$

$$W_p = \{q \in H^1(\Omega_p) : q = 0 \text{ on } \delta_3\}. \tag{4.30d}$$

Again, we first multiply by the test functions $v_f \in V_f$, $q_f \in Q_f$, $q_p \in W_p$, and integrate over the appropriate subdomains:

$$\begin{cases} \displaystyle\int_{\Omega_f} \frac{\partial u_f}{\partial t} \cdot v_f + ((u_f \cdot \nabla)u_f) \cdot v_f - \frac{1}{Re_f}\Delta u_f \cdot v_f + \nabla p_f \cdot v_f + (\nabla \cdot u_f) q_f \; d\Omega_f = 0, & \text{(4.31a)} \\ \displaystyle\int_{\Omega_p} -\nabla \cdot (Gr_n \nabla p_p) q_p \; d\Omega_p = 0. & \text{(4.31b)} \end{cases}$$

By similar steps as with PE, we can partially integrate the third and fourth term of (4.31a) to get

$$\int_{\Omega_f} -\frac{1}{Re_f}\Delta u_f \cdot v_f + \nabla p_f \cdot v_f \; d\Omega_f = \int_{\Omega_f} \frac{1}{Re_f}\nabla u_f \cdot \nabla v_f - (\nabla \cdot v_f) p_f \; d\Omega_f + \int_{\partial\Omega_f} v_f \cdot \left(p_f n_f - \frac{1}{Re_f}\frac{\partial u_f}{\partial n_f} \right) dS. \tag{4.32}$$

We can eliminate the boundary integral on the outer boundary by boundary condition (4.15c) and the enforcement of $v_f = 0$ on $\gamma_{1,2}$.

Next, we partially integrate (4.31b) to get

$$\int_{\Omega_p} -\nabla \cdot (Gr_n \nabla p_p) q_p \; d\Omega_p = \int_{\Omega_p} Gr_n \nabla p_p \cdot \nabla q_p \; d\Omega_p - \int_{\partial\Omega_p} Gr_n q_p \frac{\partial p_p}{\partial n_p} dS. \tag{4.33}$$

This boundary integral can also easily be eliminated by boundary condition (4.15d) and the enforcement that $q_p = 0$ on δ_3.

Note that $\partial\Omega_f$ and $\partial\Omega_p$ overlap on the interface Γ, and we did not account for this part yet. We will use the interface conditions to rewrite this remaining interface integral. The part resulting from the porous boundary integral is easily rewritten using (4.13a):

$$\int_\Gamma -Gr_n q_p \frac{\partial p_p}{\partial n_p} \; dS = \int_\Gamma -(u_f \cdot n_f) q_p \; dS. \tag{4.34}$$

The fluid part is a bit more cumbersome:

$$\begin{aligned}
\int_\Gamma v_f \cdot \left(p_f n_f - \frac{1}{Re_f}\frac{\partial u_f}{\partial n_f}\right) dS &= \int_\Gamma v_f \cdot \left(p_f n_f - \frac{1}{Re_f}\left(\left(\frac{\partial u_f}{\partial n_f}\right)_\tau + \left(\frac{\partial u_f}{\partial n_f}\cdot n_f\right) n_f\right)\right) dS \\
&= \int_\Gamma v_f \cdot \left(\left(p_f - \frac{1}{Re_f}\frac{\partial u_f}{\partial n_f}\cdot n_f\right) n_f - \frac{1}{Re_f}\left(\frac{\partial u_f}{\partial n_f}\right)_\tau\right) dS \\
&= \int_\Gamma v_f \cdot \left(p_p n_f - \frac{1}{Re_f}\left(\frac{\partial u_f}{\partial n_f}\right)_\tau\right) dS \qquad \text{(by (4.13b))} \\
&= \int_\Gamma p_p(v_f \cdot n_f) - \frac{1}{Re_f} v_f \cdot \left(\frac{\partial u_f}{\partial n_f}\right)_\tau dS \\
&= \int_\Gamma p_p(v_f \cdot n_f) + \frac{Gr_c}{Re_f} v_f \cdot (u_f)_\tau \, dS \qquad \text{(by (4.13c))} \\
&= \int_\Gamma p_p(v_f \cdot n_f) + \frac{Gr_c}{Re_f} (v_f)_\tau \cdot (u_f)_\tau \, dS.
\end{aligned} \tag{4.35}$$

The last step follows from the fact that

$$((v_f \cdot n_f) n_f) \cdot (u_f)_\tau = 0, \tag{4.36}$$

as the vectors are orthogonal.

As said above, this weak form does not incorporate u_p explicitly, as the paper recovers it from the Darcy equation (4.12a) as a post-processing step. However, we can immediately do this within the weak form by adding the Darcy equation directly. For this, we assume that $u_p, v_p \in X_p = (H^1(\Omega_p))^2$, and then multiply and integrate to get

$$\int_{\Omega_p} u_p \cdot v_p + Gr_n v_p \cdot \nabla p_p \, d\Omega_p. \tag{4.37}$$

We can then add this to the porous integral we derived before.

The discrete spaces are:

$$X_{fh} = \{v \in C^0(\bar{\Omega}_f) : v|_T \in (\mathcal{P}_2(T))^2 \ \forall T \in \mathcal{T}_{fh}\}, \tag{4.38a}$$
$$X_{ph} = \{v \in C^0(\bar{\Omega}_p) : v|_T \in (\mathcal{P}_2(T))^2 \ \forall T \in \mathcal{T}_{ph}\}, \tag{4.38b}$$
$$V_{fh} = \{v \in X_{fh} : v = 0 \text{ on } \gamma_{1,2}\}, \tag{4.38c}$$
$$Q_{fh} = \{q \in C^0(\bar{\Omega}_f) : q|_T \in \mathcal{P}_1(T) \ \forall T \in \mathcal{T}_{fh}\}, \tag{4.38d}$$
$$W_{ph} = \{q \in C^0(\bar{\Omega}_p) : q|_T \in \mathcal{P}_2(T) \ \forall T \in \mathcal{T}_{ph}, q = 0 \text{ on } \delta_3\}. \tag{4.38e}$$

Here, $\mathcal{T}_{fh}$ and $\mathcal{T}_{ph}$ represent the discretisations of Ω_f and Ω_p, respectively. Then the discretised weak form of the NSD case is given as below:

NSD weak form

Find $u_{fh} \in X_{fh}$, $u_{ph} \in X_{ph}$, $p_{fh} \in Q_{fh}$, ***and*** $p_{ph} \in W_{ph}$ ***such that***

$$\begin{aligned}
\int_{\Omega_f} \frac{\partial u_{fh}}{\partial t} \cdot v_{fh} + ((u_{fh} \cdot \nabla) u_{fh}) \cdot v_{fh} + \frac{1}{Re_f} \nabla u_{fh} \cdot \nabla v_{fh} - p_{fh} \nabla \cdot v_{fh} \\
+ q_{fh} \nabla \cdot u_{fh} \, d\Omega_f = 0,
\end{aligned} \tag{4.39a}$$

$$\int_{\Omega_p} Gr_n \nabla p_{ph} \cdot \nabla q_{ph} \, d\Omega_p + \int_{\Omega_p} u_{ph} \cdot v_{ph} + Gr_n v_{ph} \cdot \nabla p_{ph} \, d\Omega_p = 0, \tag{4.39b}$$

$$\int_\Gamma p_{ph}(v_{fh} \cdot n_{fh}) + \frac{Gr_c}{Re_f}(u_{fh})_\tau \cdot (v_{fh})_\tau - (u_{fh} \cdot n_{fh}) q_{ph} \, d\Gamma, \tag{4.39c}$$

for all $v_{fh} \in V_{fh}$, $v_{ph} \in X_{ph}$, $q_{fh} \in Q_{fh}$, ***and*** $q_{ph} \in W_{ph}$, ***with strong boundary conditions***

$$u_{fh} = u_{\text{pois}} \qquad \textbf{on } \gamma_1, \tag{4.40a}$$
$$u_{fh} = 0 \qquad \textbf{on } \gamma_2, \tag{4.40b}$$
$$p_{ph} = 0 \qquad \textbf{on } \delta_3. \tag{4.40c}$$

4.2.3 NSF

The paper does not give a (discretized) weak form for the NSF case, as they only solve for p_p in Ω_p. We can derive it ourself, similarly to the NSD case. Using the same spaces as NSD, let us first multiply by the test functions and integrate over the domains:

$$\begin{cases} \displaystyle\int_{\Omega_f} \frac{\partial u_f}{\partial t} \cdot v_f + ((u_f \cdot \nabla)u_f) \cdot v_f - \frac{1}{Re_f}\Delta u_f \cdot v_f + \nabla p_f \cdot v_f + (\nabla \cdot u_f) q_f \; d\Omega_f = 0, & \text{(4.41a)} \\ \displaystyle\int_{\Omega_p} u_p \cdot v_p + Gr_f |u_p| u_p \cdot v_p + Gr_n \nabla p_p \cdot v_p + (\nabla \cdot u_p) q_p \; d\Omega_p = 0. & \text{(4.41b)} \end{cases}$$

By the same steps as before, we can take the third and fourth terms of (4.41a) to get the same result as (4.32). Since we have the same boundary conditions and use the same spaces on Ω_f, we can eliminate the boundary integral on $\gamma_{1,2,3}$ in the same way as before.

For the porous domain, we take the final term of (4.41b) and integrate by parts:

$$\int_{\partial\Omega_p} (\nabla \cdot u_p) q_p \; d\Omega_p = -\int_{\Omega_p} \nabla q_p \cdot u_p \; d\Omega_p + \int_{\partial\Omega_p} q_p (u_p \cdot n_p) \; dS. \tag{4.42}$$

The boundary integral can easily be eliminated on $\delta_{1,2,3}$ by boundary condition (4.19) and the enforcement of $q_p = 0$ on δ_3.

On Γ, we have the following integral remaining:

$$\int_\Gamma v_f \cdot \left(p_f n_f - \frac{1}{Re_f} \frac{\partial u_f}{\partial n_f} \right) + q_p (u_p \cdot n_p) dS. \tag{4.43}$$

The first part follows the same steps as before, so that we again get

$$\int_\Gamma v_f \cdot \left(p_f n_f - \frac{1}{Re_f} \frac{\partial u_f}{\partial n_f} \right) dS = \int_\Gamma p_p (v_f \cdot n_f) + \frac{Gr_c}{Re_f} (u_f)_\tau \cdot (v_f)_\tau \; dS. \tag{4.44}$$

The second term is rewritten using (4.18):

$$\int_\Gamma q_p (u_p \cdot n_p) \; dS = -\int_\Gamma q_p (u_f \cdot n_f) \; dS. \tag{4.45}$$

The resulting discretised weak form is then given by

NSF weak form

Find $u_{fh} \in X_{fh}$, $u_{ph} \in X_{ph}$, $p_{fh} \in Q_{fh}$, ***and*** $p_{ph} \in W_{ph}$ ***such that***

$$\begin{aligned} &\int_{\Omega_f} \frac{\partial u_{fh}}{\partial t} \cdot v_{fh} + ((u_h \cdot \nabla)u_{fh}) \cdot v_{fh} + \frac{1}{Re_f} \nabla u_{fh} \cdot \nabla v_{fh} - p_{fh} \nabla \cdot v_{fh} \\ &\qquad + q_{fh} \nabla \cdot u_{fh} \; d\Omega_f = 0, \end{aligned} \tag{4.46a}$$

$$\int_{\Omega_p} u_{ph} \cdot v_{ph} + Gr_n \nabla p_{ph} \cdot v_{ph} + Gr_f |u_{ph}| u_{ph} \cdot v_p - \nabla q_{ph} \cdot u_{ph} \; d\Omega_p = 0, \tag{4.46b}$$

$$\int_\Gamma p_{ph} (v_{fh} \cdot n_{fh}) + \frac{Gr_c}{Re_f} (u_{fh})_\tau \cdot (v_{fh})_\tau - (u_{fh} \cdot n_{fh}) q_{ph} \; d\Gamma, \tag{4.46c}$$

for all $v_{fh} \in V_{fh}$, $v_{ph} \in X_{ph}$, $q_{fh} \in Q_{fh}$, ***and*** $q_{ph} \in W_{ph}$, ***with strong boundary conditions***

$$u_{fh} = u_{\text{pois}} \qquad \textbf{on } \gamma_1, \tag{4.47a}$$

$$u_{fh} = 0 \qquad \textbf{on } \gamma_2, \tag{4.47b}$$

$$p_{ph} = 0 \qquad \textbf{on } \delta_3. \tag{4.47c}$$

5 Implementation

In this section, we will walk through the implementation of the case described above. The programming language used for the implementation is Julia. Julia is often acclaimed for its resemblance to Python, making it simple to use, while also being more efficient. Furthermore, Julia allows the user to make extensive use of Unicode symbols — which can be entered through LaTeX-syntax — such that the code resembles mathematical notation more closely. The package used for incorporating the Finite Element Method is Gridap. It is a fairly recent development, whereby its documentation is often lacklustre, but it has a lot of built-in functionality with regards to the FEM.

The complete codes are available at `https://github.com/jumaropa/julian-msc/tree/main/thesis-codes`. The contents of `functions.jl` will be explained here.

5.1 Channel Flow with Lagrange Elements

First, we model the channel flow described in the previous section with the simple Lagrange elements. This is officially done in order to verify that the results match up with the paper's, but unofficially to get an understanding of the modelling software. We will use the same formulations and parameters as the paper. For each case, We discretise the domain into quadrilaterals.

5.1.1 Setting up the domain

Since this part is essentially the same for all approaches, we will describe the process of setting up the domain and its discretisation once. First, we define the entire domain and discretise it. Below, we set 50 elements in the x-direction, and 7 in the y-direction:

```
domain = (0,L_x,-L_y_p,L_y_f)
n_x = 50
n_y = 7
partition = (n_x,n_y)
model_Ω = CartesianDiscreteModel(domain,partition)
```

In the above, `domain` is a rectangle defined by the coordinates of two opposite corners, in this case $(0, \texttt{L\_x})$ and $(-\texttt{L\_y\_p}, \texttt{L\_y\_f})$. The constants `L_x`, `L_y_p`, and `L_y_f` correspond respectively to the width of the domain, the height of the porous domain, and the height of the fluid domain. By defining the domain in this way, the interface sits at $y = 0$. Next, we do

```
labels_Ω = get_face_labeling(model_Ω)
```

to get labels associated with the domain's features. By default, a numbered tag is given for the four corners, the four boundary edges, and the interior; see Figure 5.1. A tag named `"boundary"` is also present for all points in the boundary. The tags apply to the vertices of the elements, their edges, and their faces. These tags can be used to set boundary conditions, or to define subdomains, for example.

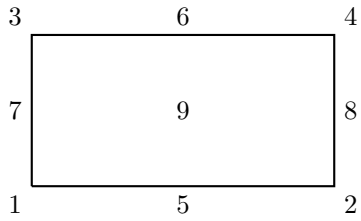


Figure 5.1: Default tags of a domain

Since we want to distinguish the porous domain from the fluid domain, we will need extra tags. We loop through the elements and determine for their vertices, edges, and faces, whether they sit in the porous or fluid domain. If any feature lies exactly on the interface, we assume it to be in the porous domain. For the edges and vertices on the left and right boundaries within the porous domain, we update the tags, as well as for all the faces. The code is below:

```
for d in 0:2
    face_coords = get_cell_coordinates(Grid(ReferenceFE{d}, model_Ω))
    left_p_boundary  = findall(is_p_boundary(face_coords, 0.0))
    for i in left_p_boundary
        labels_Ω.d_to_dface_to_entity[d+1][i] = entity_tag_left_p
    end
    right_p_boundary  = findall(is_p_boundary(face_coords, L_x))
    for i in right_p_boundary
        labels_Ω.d_to_dface_to_entity[d+1][i] = entity_tag_right_p
    end

    p_region = findall([lazy_map(is_p, face_coords)[i] for i in 1:length(face_coords)])
    for i in p_region
        if labels_Ω.d_to_dface_to_entity[d+1][i] == 9
            labels_Ω.d_to_dface_to_entity[d+1][i] = entity_tag_p
        end
    end
end
```

In the above, `is_p_boundary()` is a self-written function that determines whether a given feature lies on the given porous boundary (left or right). Furthermore, `entity_tag_left_p`, `entity_tag_right_p`, and `entity_tag_p` are given respectively as 10, 11, and 12. In order to make it easier to use these tags, we also give them descriptive names:

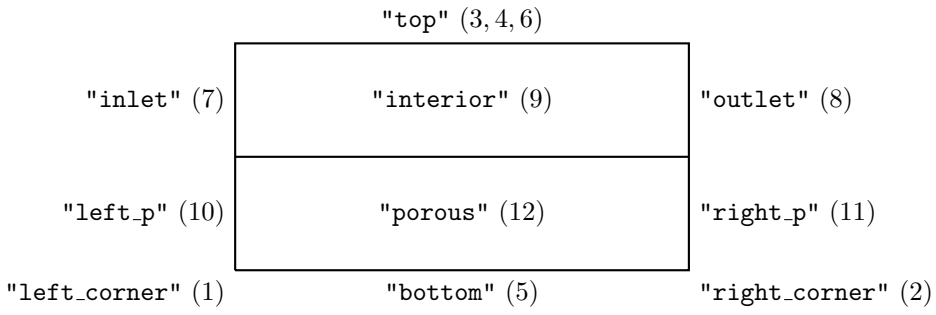


Figure 5.2: Our given tags for the channel flow domain

We also define `"fluid_b"` and `"porous_b"` as respectively the fluid and porous interiors, together with their adjacent boundaries. We use these tags to define the relevant subdomains:

```
degree = 2*order
Ω_h = Triangulation(model_Ω)
dΩ = Measure(Ω_h,degree)
```

```
Ω_p = Triangulation(model_Ω, tags="porous_b")
dΩ_p = Measure(Ω_p,degree)
Ω_f = Triangulation(model_Ω,tags="fluid_b")
dΩ_f = Measure(Ω_f,degree)
```

5.1.2 Penalisation Method

For the first approach, we choose PE, the one-domain approach, since we will not have to deal with the interface condition yet. We will first set up the elements and the spaces. First we define it for the velocity:

```
order = 2
reffe_u = ReferenceFE(lagrangian,VectorValue{2,Float64},order)
```

So we have a (for example) second-order two-dimensional Lagrangian element. Next, we set the test and trial spaces:

```
V = TestFESpace(model_Ω,
                reffe_u,
                conformity=:H1,
                dirichlet_tags=["inlet", "top", "left_p", "bottom", "left_corner"],
                dirichlet_masks=[(true, true), (true, true), (true, false),
                                 (false, true), (true, true)])
u_pois((x,y)) = VectorValue(y*(L_y_f-y),0)
u0= VectorValue(0,0)
U = TrialFESpace(V, [u_pois, u0, u0, u0, u0])
```

The `dirichlet_tags` parameter is used to state on which parts of the boundary there will be Dirichlet type boundary conditions. The `dirichlet_masks` parameter assigns boolean tuples to these parts that signify in which direction the boundary condition applies. For example, for the `"bottom"` tag, the tuple `(false, true)` is given, which means there is a condition in the y-direction, but not in the x-direction. This corresponds to the condition $u_h \cdot n = 0$ that we set strongly on δ_2. Note that we apply a separate boundary condition on the lower left corner — labelled — `left_corner`. This is because both boundary conditions for δ_1 and δ_2 should be satisfied in this corner.

We follow a similar procedure for the pressure:

```
reffe_p = ReferenceFE(lagrangian,Float64,order-1)
Q = TestFESpace(model_Ω,
                reffe_p,
                conformity=:L2)
P = TrialFESpace(Q)
```

Note that we don't have to set any boundary conditions in this case, because we don't have any strong boundary conditions pertaining to the pressure. The spaces we defined are then combined into a singular entity:

```
X = MultiFieldFESpace([U,P])
Y = MultiFieldFESpace([V,Q])
```

Next, we enter the weak form. We will leave out the transient term, as the authors of the paper describe a steady-state in any case. In Gridap, we set up the weak form as follows:

```
a((u,p),(v,q)) = ∫(1/Re * ∇(u)⊙∇(v) - p*(∇⋅v) + q*(∇⋅u))dΩ +
                 ∫( v⊙(Gr_v*u) )dΩ_p
b(u,v) =         ∫( v⊙(Gr_i*((u⋅u).^(1/2))*u) )dΩ_p
db(u,du,v) = ∫(v⊙(Gr_i*((u⋅u).^(1/2))*du))dΩ_p
           + ∫(v⊙(Gr_i*((u⋅du)/(unorm(u))*u)))dΩ_p
c(u,v) = ∫( v⊙(conv∘(u,∇(u))) )dΩ
dc(u,du,v) = ∫( v⊙(dconv∘(du,∇(du),u,∇(u))) )dΩ
```

The function `a` includes the Stokes momentum equation, along with the Darcy term. The function `b` adds the Forchheimer term, and `c` adds the convective term. Above, we use a few helper functions, defined as

```
conv(u,∇u) = (∇u')⋅u
dconv(du,∇du,u,∇u) = conv(u,∇du)+conv(du,∇u)
unorm(u) = (u⋅u).^(1/2) + 1e-12
```

The function `unorm` includes a small perturbation included to avoid division by zero, as the function is divided by.

Note the inclusion of the functions `db` and `dc`. These are needed for the Jacobian of the residual, which we construct as follows:

```
res((u,p),(v,q)) = a((u,p),(v,q)) + b(u,v) + c(u,v)
jac((u,p),(du,dp),(v,q)) = a((du,dp),(v,q)) + db(u,du,v) + dc(u,du,v)
```

We use these to construct the operator:

```
op = FEOperator(res,jac,X,Y)
```

Then we solve the equation using a non-linear solver:

```
nls = NLSolver(show_trace=true, method=:newton, iterations=50)
uh, ph = solve(nls, op)
```

With `method:=newton`, we specify that we are using the Newton-Raphson method. This works by taking an initial guess u_0, and iteratively solving

$$J(u_n)(u_{n+1} - u_n) = -F(u_n), \tag{5.1}$$

with J the Jacobian matrix and F the residual matrix, until a certain convergence criterion is met. The default convergence criterion is set to 10^{-8}.

5.1.3 Navier-Stokes/Darcy

The implementation for the NSD case is similar, but there are some key differences. We have to define elements and spaces for the fluid and porous domains seperately. First, we define the elements:

```
reffe_uf = ReferenceFE(lagrangian,VectorValue{2,Float64},order)
reffe_pf = ReferenceFE(lagrangian,Float64,order-1)
reffe_up = ReferenceFE(lagrangian,VectorValue{2,Float64},order)
reffe_pp = ReferenceFE(lagrangian,Float64,order)

# fluid velocity
Vf = TestFESpace(
  Ω_f,
  reffe_uf,
  conformity=:H1,
  dirichlet_tags=["left_p", "inlet", "top"],
  dirichlet_masks=[(true, false), (true, true), (true, true)])

# porous velocity
Vp = TestFESpace(
  Ω_p,
  reffe_up,
  conformity=:H1)

# fluid pressure
Qf = TestFESpace(
  Ω_f,
  reffe_pf,
  conformity=:L2)

# porous pressure
Qp = TestFESpace(
  Ω_p,
  reffe_pf,
```

```
  conformity=:H1,
  dirichlet_tags=["right_p", "right_corner"])

##trial spaces
Uf = TrialFESpace(Vf, [u_0, u_pois, u_0])
Up = TrialFESpace(Vp)
Pf = TrialFESpace(Qf)
Pp = TrialFESpace(Qp, [0.0, 0.0])

Y = MultiFieldFESpace([Vf,Qf,Vp,Qp]) #test
X = MultiFieldFESpace([Uf,Pf,Up,Pp]) #trial
```

We also have to define the interface in order to apply the interface conditions correctly:

```
Γ_fp = InterfaceTriangulation(Ω_f,Ω_p)
dΓ_fp = Measure(Γ_fp,2*order)
```

For this, we also need the normal vector on the interface, which we obtain through

```
n_Γfp = get_normal_vector(Γ_fp)
```

We then define the weak form as follows:

```
#fluid
af((uf,pf),(vf,qf)) = ∫( 1/Re*∇(uf)⊙∇(vf) - pf*(∇⋅vf) + qf*(∇⋅uf) )dΩ_f
c((uf,pf),(vf,qf)) = ∫( vf⊙(conv∘(uf,∇(uf))) )dΩ_f
dc(uf,duf,vf) = ∫( vf⊙(dconv∘(duf,∇(duf),uf,∇(uf))) )dΩ_f

#porous
ap((up,pp),(vp,qp)) = ∫( Gr_n*∇(pp)⋅∇(qp) )dΩ_p +
                      ∫( up⋅vp + Gr_n*∇(pp)⋅vp )dΩ_p

#interface
apf((uf,pf,up,pp),(vf,qf,vp,qp)) = ∫( Gr_c/Re*(uf.plus-(uf.plus⋅n_Γfp.plus)*n_Γfp.plus)
                    ⋅(vf.plus-(vf.plus⋅n_Γfp.plus)*n_Γfp.plus) )dΓ_fp -
                    ∫( (uf.plus⋅n_Γfp.plus)*qp.minus )dΓ_fp +
                    ∫( (vf.plus⋅n_Γfp.plus)*pp.minus )dΓ_fp

l((vf,qf)) = ∫(0*qf)dΩ_f

a((uf,pf,up,pp),(vf,qf,vp,qp)) = af((uf,pf),(vf,qf)) + ap((up,pp),(vp,qp))
                                 + apf((uf,pf,up,pp),(vf,qf,vp,qp))

res((uf,pf,up,pp),(vf,qf,vp,qp)) = a((uf,pf,up,pp),(vf,qf,vp,qp)) + c((uf,pf),(vf,qf))
jac((uf,pf,up,pp),(duf,dpf,dup,dpp),(vf,qf,vp,qp)) = a((duf,dpf,dup,dpp),(vf,qf,vp,qp))
                                                      + dc(uf,duf,vf)
```

This is solved in the same way as the PE case.

5.1.4 Navier-Stokes/Darcy-Forchheimer

For NSF, the process is the exact same as for NSD, expect for the porous integral, which consists of two parts:

```
ap((up,pp),(vp,qp)) = ∫( up⋅vp + Gr_n*∇(pp)⋅vp - ∇(qp)⋅up )dΩ_p
apb((up,pp),(vp,qp)) = ∫( vp⊙(Gr_f*((up⋅up).^(1/2))*up) )dΩ_p
dapb(up,dup,vp) = ∫(vp⊙(Gr_f*((up⋅up).^(1/2))*dup))dΩ_p
                + ∫(vp⊙(Gr_f*((up⋅dup)/(unorm(up))*up)))dΩ_p
```

`apb` and `dapb` are to be added to the residual and Jacobian, respectively.

5.2 Channel Flow with Raviart-Thomas Elements

We now turn to the implementation of the same cases, but using the Raviart-Thomas elements. The overall implementation is the same as for the Lagrange case, but there are a few key differences.

5.2.1 Penalisation Method

Firstly, we have to define the Raviart-Thomas element for the velocity space:

```
order = 1
reffe_u = ReferenceFE(raviart_thomas,Float64,order)
```

Note that we don't need to set the type as a `VectorValue` anymore, since Raviart-Thomas elements are inherently two-dimensional. Gridap defines the order differently than we have done in section 3.5. It defines it as the order of the Lagrange space that the divergence of the functions in the RT-basis belongs to, i.e.

$$RT_r = \mathcal{P}_{r+1,r}(T) \times \mathcal{P}_{r,r+1}(T). \tag{5.2}$$

Next, we define the test space:

```
V = TestFESpace(model_Ω,
                reffe_u,
                labels=labels_Ω,
                conformity=:Hdiv,
                dirichlet_tags=["inlet", "top", "left_p", "bottom", "left_corner"])
```

See how the conformity is set to `Hdiv`. Notice also that we don't use the `dirichlet_masks` parameter anymore. That is because it doesn't really have meaning in case of Raviart-Thomas elements. Gridap, by default, only takes the normal boundary conditions. The tangential boundary conditions have to be enforced weakly. We do this through Nitsche's method (see e.g. [23]). This involves adding

$$\int_{\partial\Omega} \frac{\gamma}{h} v \cdot (u - g) - (\nabla u \cdot n) \cdot v - (\nabla v \cdot n) \cdot (u - g)\, dS \tag{5.3}$$

to the left hand side of the weak form. Here, γ is the penalisation coefficient, n is the normal vector on the boundary, and g is the boundary condition to enforce. The tangential boundary conditions on the porous domain are actually already implicitly contained within the derived weak form, so we only need to look at the fluid domain. In Gridap, we can use the boundary tags we created to enforce this weak boundary condition only on certain parts of the domain, in our case the inlet and the top of the domain:

```
Γ = BoundaryTriangulation(model_Ω, tags=["inlet", "top"])
n_Γ = get_normal_vector(Γ)
dΓ = Measure(Γ,degree)
```

Then we can simply set

```
g(x) = u_pois(x)
```

to correctly enforce the boundary conditions:

```
∫( (γd/h)*v⋅(u-g) - (∇(u)⋅n_Γ)⋅v - (∇(v)⋅n_Γ)⋅(u-g) )dΓ
```

The Raviart-Thomas basis functions are only continuous in the normal component, so we need to ensure full continuity by adding

$$\sum_{e \in \mathcal{E}_h} \int_e \frac{\gamma}{h} [\![v]\!][\![u]\!] - (\{\!\{\nabla u\}\!\} \cdot n)[\![v]\!] - (\{\!\{\nabla v\}\!\} \cdot n)[\![u]\!]\, dS. \tag{5.4}$$

Here, $\mathcal{E}_h$ is the set of edges of the discretisation. Furthermore, for a vector $u = (u_1, u_2)$, the jump and mean operation are respectively defined as

$$[\![u]\!] = u_1 - u_2, \tag{5.5}$$

$$\{\!\{u\}\!\} = \tfrac{1}{2}(u_1 + u_2). \tag{5.6}$$

In Gridap, we add

```
∫( (γd/h)*jump(v)⋅jump(u) - (mean(∇(u))⋅n_Λ.⁺)⋅jump(v)
                          - (mean(∇(v))⋅n_Λ.⁺)⋅jump(u) )dΛ
```

with

```
Λ = Skeleton(model_Ω)
n_Λ = get_normal_vector(Λ)
dΛ = Measure(Λ,degree)
```

5.2.2 Navier-Stokes/Darcy and Navier-Stoker/Forchheimer

For the two-domain approaches, we define four different elements to use:

```
reffe_uf = ReferenceFE(raviart_thomas,Float64,order-1)
reffe_pf = ReferenceFE(lagrangian,Float64,order-1)
reffe_up = ReferenceFE(lagrangian,VectorValue{2,Float64},order)
reffe_pp = ReferenceFE(lagrangian,Float64,order)
```

Other than this, the implementation is the same as before, with the added Nitsche terms being the same.

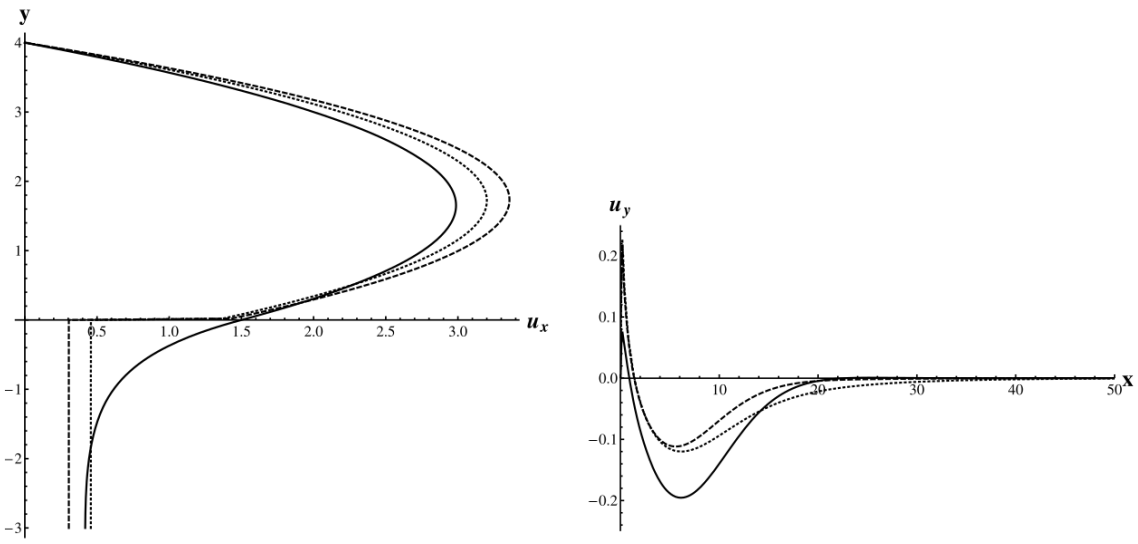
6

Results

In this section, we will give the results obtained from the implementation above. We will first check whether our results line up with the paper's when using their specifications. After that, we will see how using Raviart-Thomas elements compares with Lagrange for the three approaches we have discussed.

6.1 Verification of the Lagrange Case

First, let us look at the results from the paper that we are comparing with. The paper gives many results from a thorough parameter analysis, but we are only interested whether our Lagrange plots line up with the paper's, so as to have a reference point. In Figure 6.1, we show the normal velocities at the outlet ($x = 50$) and the interface ($y = 0$) for the three approaches, as given in the paper. We clearly see the difference between the one-domain and two-domain approaches, as the first varies continuously near the interface, while the latter has a discontinuity. These plots will be used to determine whether our Lagrange plots are accurate.



(a) u_x at the outlet

(b) u_y at the interface

Figure 6.1: Normal velocities at the outlet and interface, as taken from [22] (NSD dotted line, NSF dashed line, PE solid line)

The paper uses $P_2 - P_1$ elements, so we will too. The combination of P_r elements for the velocity and P_{r-1} elements for the pressure is known as the Taylor-Hood mixed finite element, which is well-known to satisfy the inf-sup condition (3.40) and is thus a stable choice. Next, we use a gridsize of 150×21, as the papers does. It is worth mentioning that the paper uses triangular elements, whereas we will use quadrilateral elements. First, we show the overall magnitude of the velocity in the domain in Figure 6.2, taking the PE case. We see the overall shape of velocity field is what we expect. There is a clear flow through the fluid domain with some filtration into the porous domain.

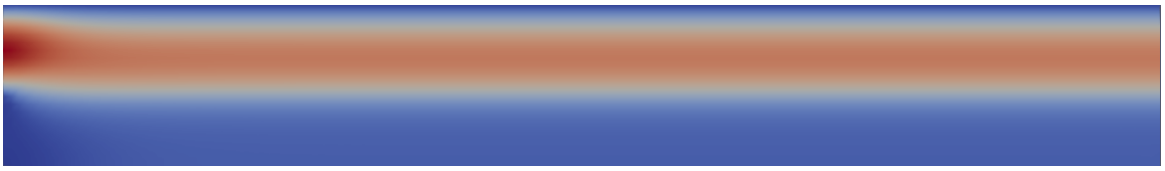
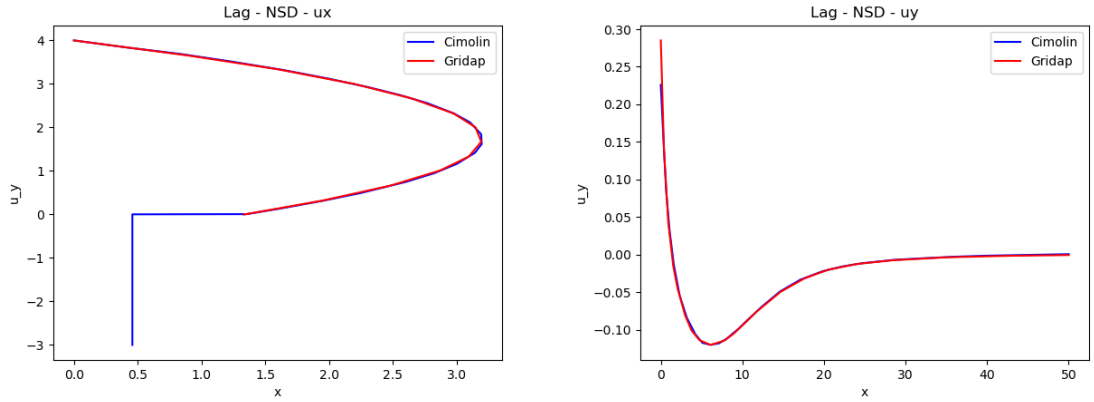


Figure 6.2: PE: Magnitude of the velocity with P_2 elements

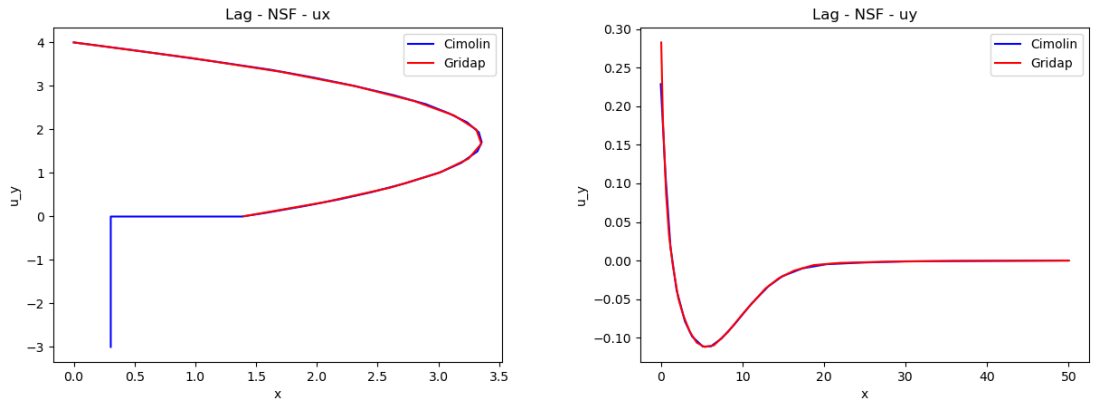
When we compare the outlet and interface velocities using our implementations of NSD and NSF using Lagrange elements, we get the results as seen in Figure 6.3 and 6.4.



(a) u_x at the outlet

(b) u_y at the interface

Figure 6.3: Comparison of our and Cimolin's implementation of NSD

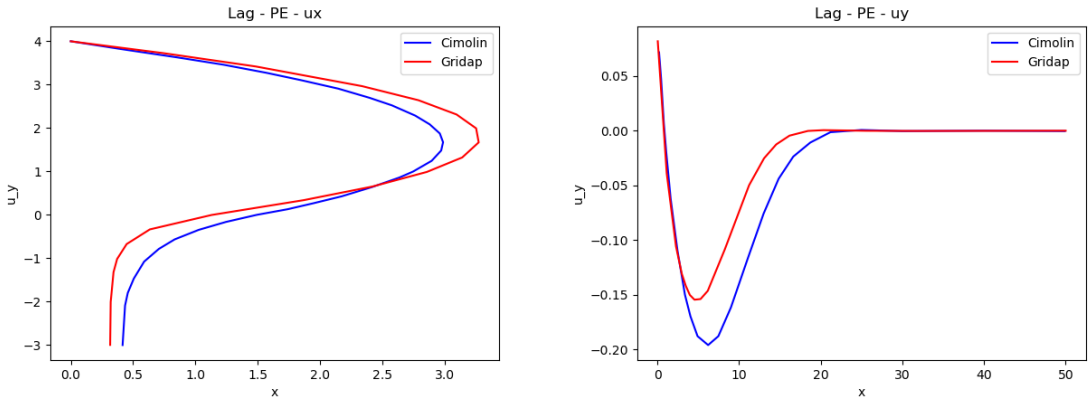


(a) u_x at the outlet

(b) u_y at the interface

Figure 6.4: Comparison of our and Cimolin's implementation of NSF

We see excellent agreement. Only the interface velocities have a small discrepancy at $x = 0$. However, when we look at the PE comparison (see Figure 6.5), the results are quite different.



(a) u_x at the outlet

(b) u_y at the interface

Figure 6.5: Comparison of our and Cimolin's implementation of PE

This suggests a programming error on our part, but we could not find any errors in our implementation, so we will assume our results as in Figure 6.5 are correct, and take these as a future reference point.

Now we will use Raviart-Thomas elements for the velocity, and see how this compares to using Lagrange elements. We are particularly interested in the divergence of the velocity, as Raviart-Thomas elements are $H(\text{div})$-conforming.

6.2 PE with Raviart-Thomas Elements

First, we look at the PE case. We will compare RT to Lagrange for orders $r = 1, 2, 3$, using different gridsizes. In order to keep things tidy, we will define some notations for the different mixed element types. We will use $PE_{Lag,r}$ for the Lagrange case, which indicates that we use P_r elements for the velocity, and P_{r-1} elements for the pressure. For Raviart-Thomas, we use $PE_{RT,r}$, which uses RT_r elements for the velocity, and P_{r-1} elements for the pressure.

6.2.1 Divergence

First, we compare the divergence and the iteration number. We use the default tolerance of 10^{-8}. The divergence is calculated using Gridap's `divergence()` function, i.e.

```
div_uhf = divergence(uhf)
```

We then take the L^2 norm, that is

$$\left(\int_\Omega (\nabla \cdot u_h)^2 \, d\Omega\right)^{\frac{1}{2}}. \tag{6.1}$$

In the code, this reads as below:

```
div_uh = divergence(uh)
l2_norm = sqrt(sum(∫( div_uh*div_uh )dΩ))
```

We first look at $PE_{Lag,1}$ versus $PE_{RT,1}$. The results are given below in Table 6.1.

	$PE_{Lag,1}$		$PE_{RT,1}$	
gridsize	#iterations	divergence	#iterations	divergence
50×7	6	0.308107	4	2.134031e-15
100×14	5	0.215788	4	4.681290e-15
150×21	5	0.166292	4	9.895104e-15
200×28	5	0.135324	4	9.213859e-15

Table 6.1: Divergence L^2-norm and iteration number for different gridsizes, for the PE case using $PE_{Lag,1}$ and $PE_{RT,1}$ elements

We already see how RT trumps Lagrange in terms of divergence. Our theoretical assumption that RT provides an exactly divergence-free solution for this problem is verified. However, looking at the magnitude of the velocity in Figure 6.6, we see that it is not what we expect.

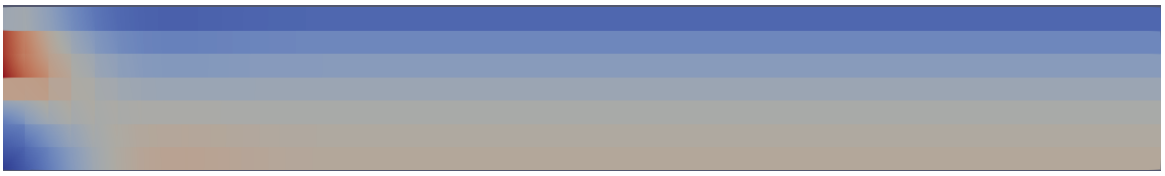


Figure 6.6: PE: Magnitude of the velocity with $PE_{RT,1}$ elements (50×7)

The most likely explanation for this is that we are penalising the tangential component of the flux along the element interface, but at the lowest order we don't have enough degrees of freedom to accommodate this, whereby it is over-constrained. Within an element, the horizontal component is constant along the vertical axis, and vice versa. If the tangential components of two adjoining elements are then made to be (approximately) the same along their shared edge, this affects the opposite edges as well, causing a chain reaction. Papers with similar approaches, e.g. [24], also do not take Raviart-Thomas elements of the lowest order.

Now we look at $PE_{Lag,2}$ versus $PE_{RT,r}$. The results for the divergence are given below in Table 6.2.

	$PE_{Lag,2}$		$PE_{RT,2}$	
gridsize	#iterations	divergence	#iterations	divergence
50×7	5	0.148994	7	3.959796e-14
100×14	5	0.081081	7	7.677469e-14
150×21	5	0.053126	7	1.246596e-13
200×28	5	0.039063	6	1.622965-13

Table 6.2: Divergence L^2-norm and iteration number for different gridsizes, for the PE case using $PE_{Lag,2}$ and $PE_{RT,2}$ elements

We see a similar overall picture to before. This time, RT takes a few more iterations, and the divergence is slightly higher, but still within machine precision. This is likely the result of having to do more calculations, whereby the round-off errors add up to a bigger number. This time the overall magnitude of the velocity looks as we expect it to, as seen in Figure 6.7.

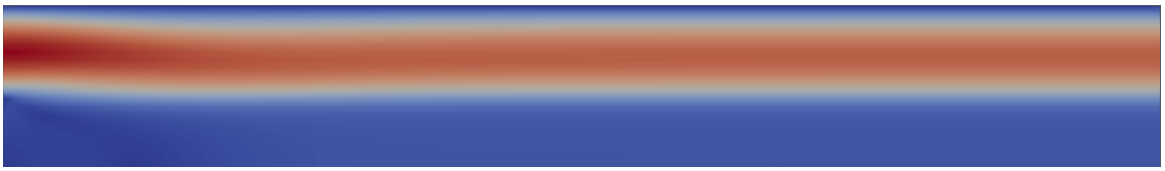


Figure 6.7: PE: Magnitude of the velocity using $PE_{RT,2}$ elements (150×21)

In order to see where the filtration into the porous domain happens, we also show just the y-component of the velocity in Figure 6.8 for Lagrange and RT. We see that most of the filtration happens very near the inlet. The two pictures are very similar but we see some differences. Firstly, Lagrange has a greater range of values it attains, meaning the filtration is slightly more pronounced. Also, for Lagrange, the red splodge indicating fluid circulating back up is directly at the left boundary, while for RT it is slightly away from it. Closer investigation shows that it is exactly one element's width away.

(a) using $PE_{Lag,2}$ elements

(b) using $PE_{RT,2}$ elements

Figure 6.8: PE: y-component of the velocity using $PE_{Lag,2}$ and $PE_{RT,2}$ elements (150×21)

We also show the pressure plots for Lagrange and RT in Figure 6.9. The Lagrange picture looks as expected, with the pressure relatively high at the inlet, gradually reducing to zero as the fluid reaches the outlet. The RT picture has an unfortunate spike in the upper left corner, which dominates the picture. This spike actually contains values oscillating between negative and positive values, as seen from the colour bar. Apart from this spike, the picture actually looks quite similar to the Lagrange case. This is shown in Figure 6.9c, where only the values attained by Lagrange are shown. This also makes it apparent that a similar issue is present at the origin, although less severe.

(a) using $PE_{Lag,2}$ elements

(b) using $PE_{RT,2}$ elements

(c) using $PE_{RT,2}$ elements - rescaled

Figure 6.9: PE: pressure using $PE_{Lag,2}$ and $PE_{RT,2}$ elements (150×21)

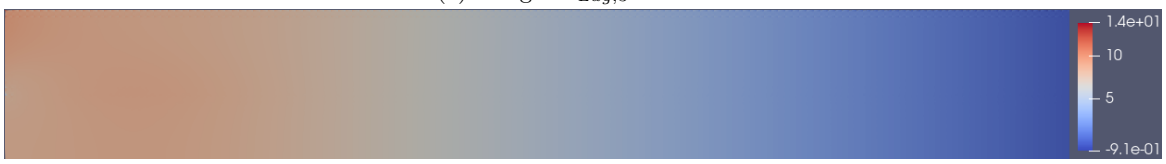
Lastly, we look at $PE_{Lag,3}$ versus $PE_{RT,3}$ elements. For this order of RT elements, we found that the default tolerance of 10^{-8} was too sharp. The coarsest gridsize did not converge for this case, instead fluctuating around values in the order of 10^{-7}. Finer gridsizes fluctuated around increasingly higher values. Therefore, we set the tolerance at 10^{-4}. The results we obtained herewith are found below in Table 6.3.

	$PE_{Lag,3}$		$PE_{RT,3}$	
gridsize	#iterations	divergence	#iterations	divergence
50×7	5	0.084315	7	1.120651e-11
100×14	5	0.03923	6	2.330963e-11
150×21	5	0.023918	6	3.498368e-11
200×28	5	0.017296	6	4.749699e-11

Table 6.3: Divergence L^2-norm and iteration number for different gridsizes, for the PE case using $PE_{Lag,3}$ and $PE_{RT,3}$ elements

In Figure 6.10, we show the pressure again. We see that the problem we encountered with the RT_2 is still causes some issues, albeit way less severe. we see slight oscillations along the entire top of the domain.

(a) using $PE_{Lag,3}$ elements



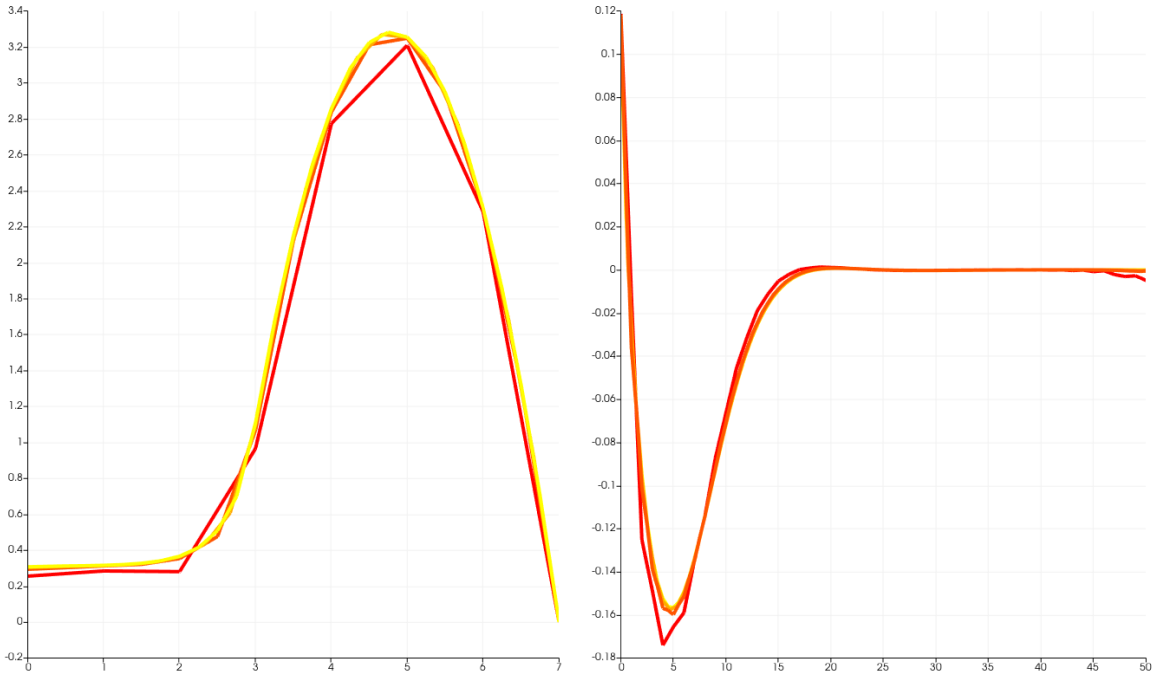
(b) using $PE_{RT,3}$ elements

Figure 6.10: PE: pressure using $PE_{Lag,3}$ and $PE_{RT,3}$ elements (150×21)

6.2.2 Velocity Profiles

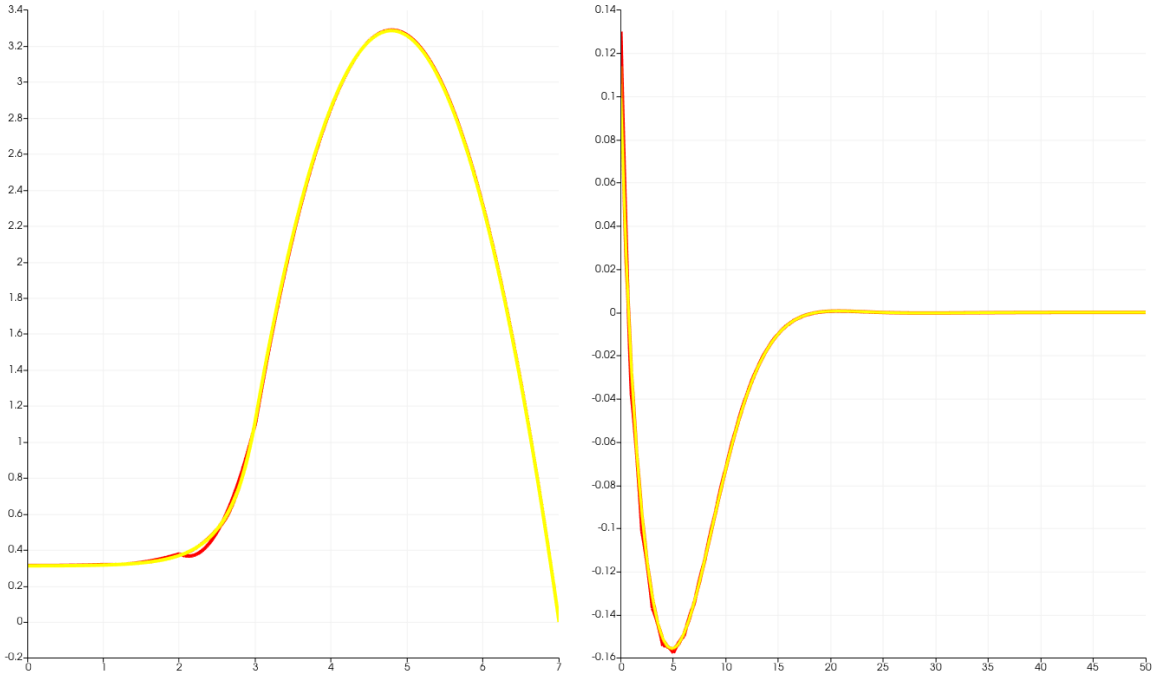
We have verified that the RT elements give an exactly divergence-free solution, matching the theory. Next, we look at how the outlet and interface velocities compare between RT and Lagrange. Because of the way RT-elements are defined, it actually makes little sense to compare the same order in this context, as the actual polynomial orders will be different. That is because we are looking at the u_x velocity plotted over the vertical axis, and the u_y velocity over the horizontal axis, thus essentially showing the lower order for the RT-elements. Therefore, we compare $PE_{RT,r}$ elements with $PE_{Lag,r-1}$ elements in this regard.

First, we will plot the velocities using the same element for various gridsizes in the same figure. Figure 6.11 and 6.12 show the results for $PE_{Lag,1}$ and $PE_{Lag,2}$ elements, respectively. Figure 6.13 and 6.14 show the results for $PE_{RT,2}$ and $PE_{RT,3}$ elements, respectively. Note that the outlet velocities are given with the y-axis horizontally.



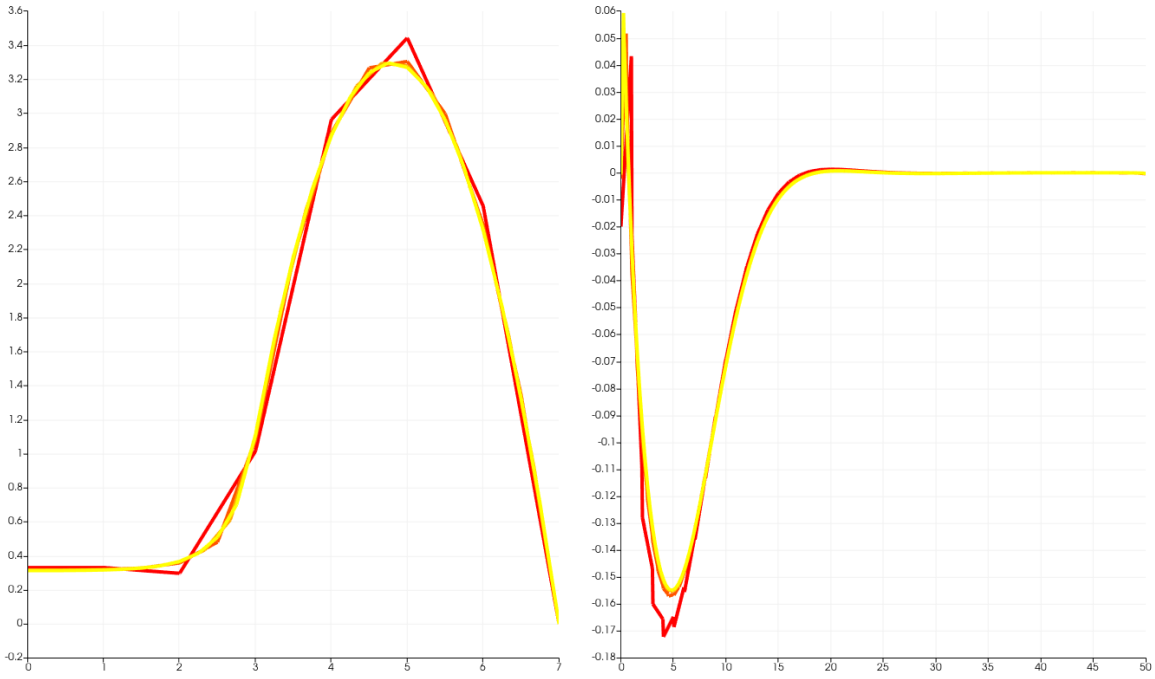
(a) u_x at the outlet (b) u_y at the interface

Figure 6.11: PE: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $PE_{Lag,1}$ elements



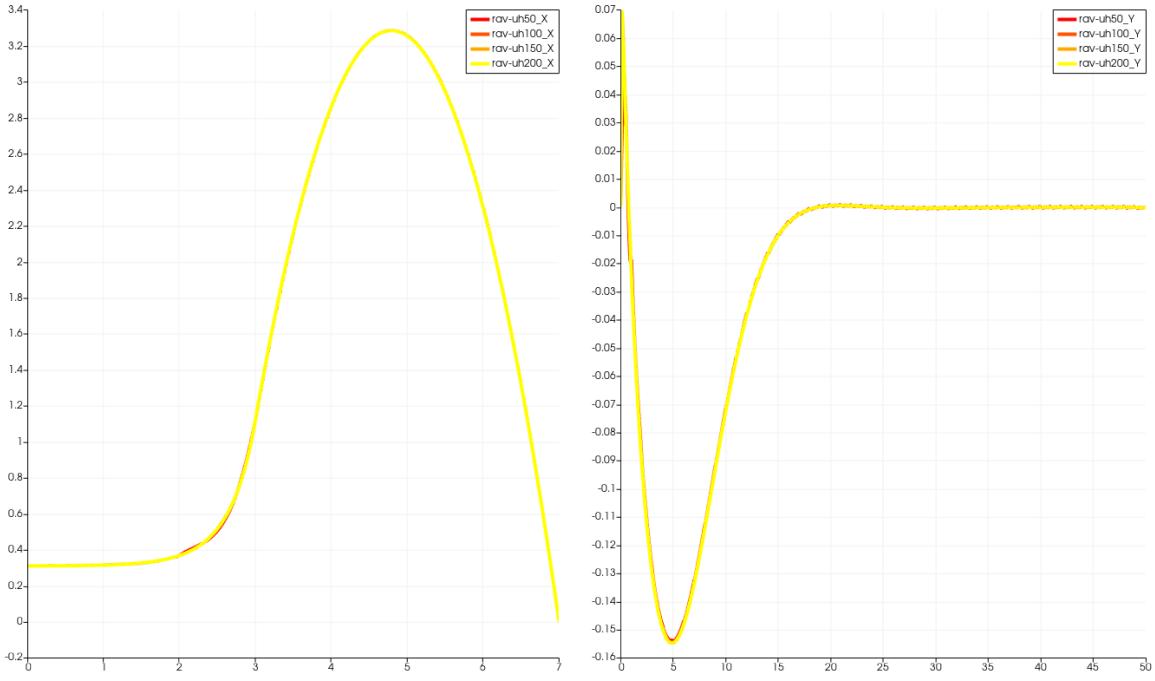
(a) u_x at the outlet (b) u_y at the interface

Figure 6.12: PE: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $PE_{Lag,2}$ elements



(a) u_x at the outlet (b) u_y at the interface

Figure 6.13: PE: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $PE_{RT,2}$ elements



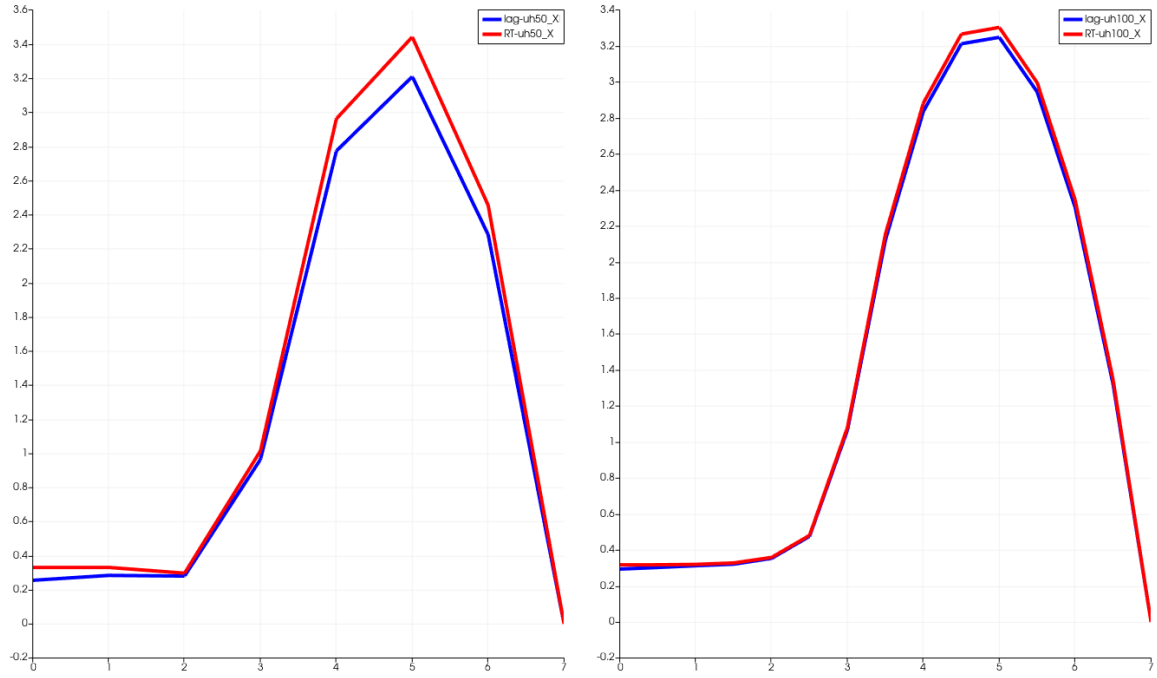
(a) u_x at the outlet (b) u_y at the interface

Figure 6.14: PE: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) — using $PE_{RT,3}$ elements

An interesting observation can be made when comparing Figure 6.11 and 6.13. For the Lagrange case, we see clearly for the outlet velocity that the solution is approximated in the gridnodes, as that is where the degrees of freedom lie. This is not the case for RT, where the solution is more approximated over the whole element.

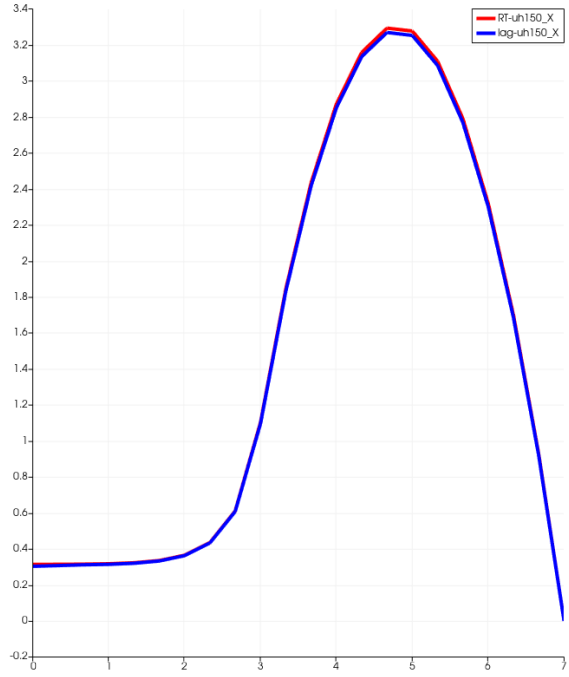
Another observation that we can make is that the interface velocity is always zero at $x = 0$ for the RT case. For Lagrange, we had to choose whether the interface is part of the fluid or the porous domain, as that is actually ambiguous from the given model. In the paper, it is clearly part of the porous domain, as the vertical velocity in the origin is free, as we see in Figure 6.1b. This would not correspond to the boundary condition enforced on γ_1. However, for the RT case, this distinction actually makes no difference, because there is no degree of freedom corresponding to the origin.

In order to get a clearer comparison, we also compare the Lagrange and RT elements directly for each gridsize, and each order. First, we compare $PE_{RT,2}$ elements with $PE_{Lag,1}$ elements. See Figure 6.15 and 6.16.

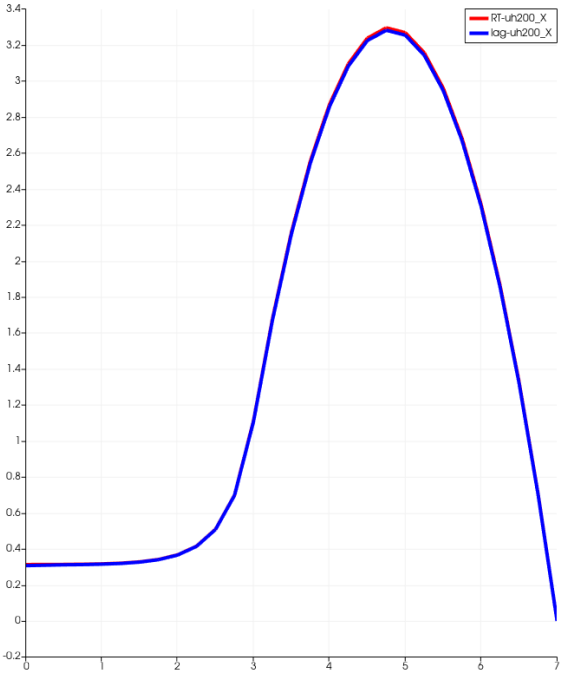


(a) 50×7

(b) 100×14

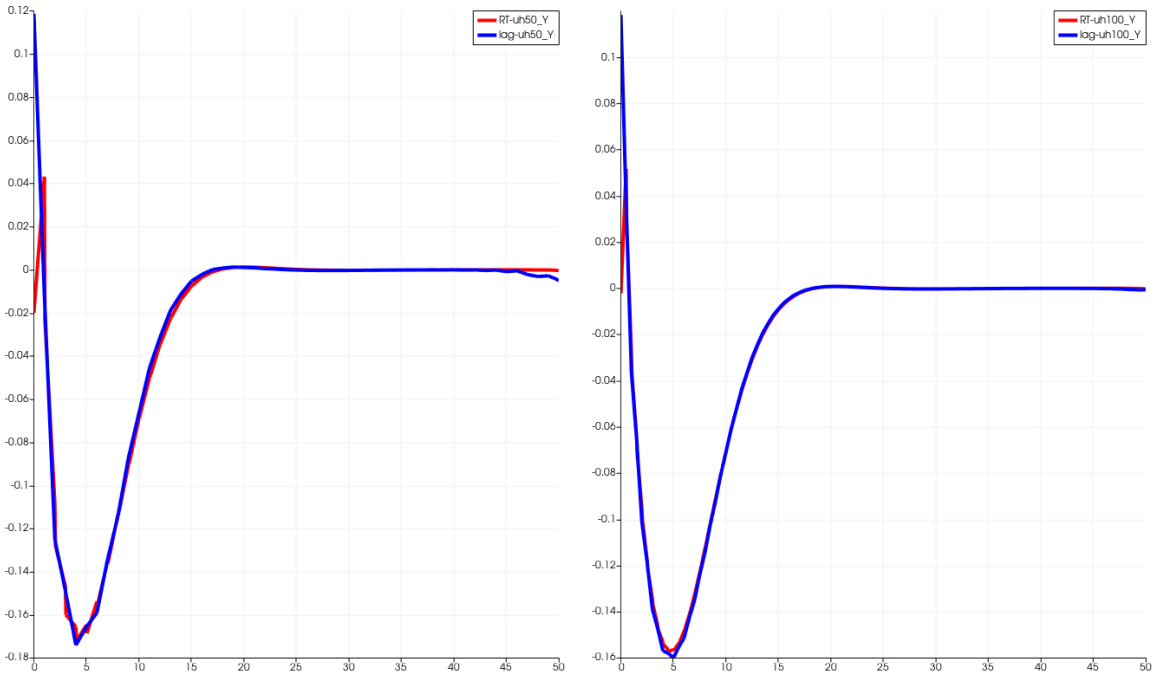


(c) 150×21



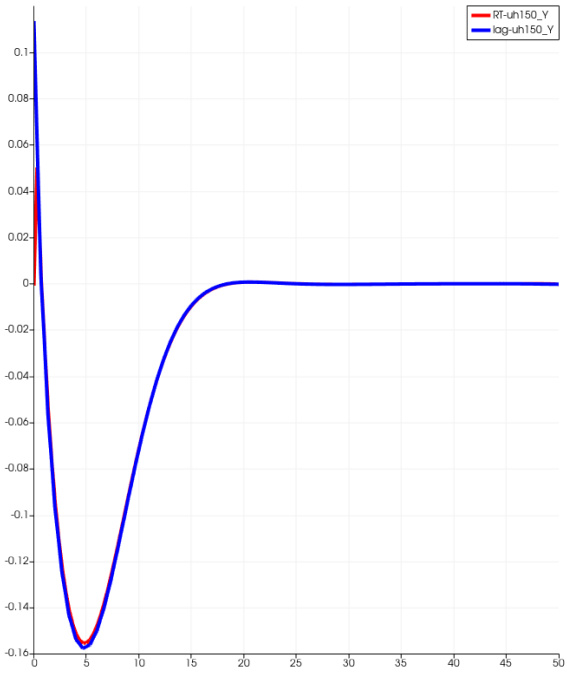
(d) 200×28

Figure 6.15: PE: outlet velocity for various gridsizes, comparing $PE_{RT,2}$ (red) with $PE_{Lag,1}$ (blue) elements

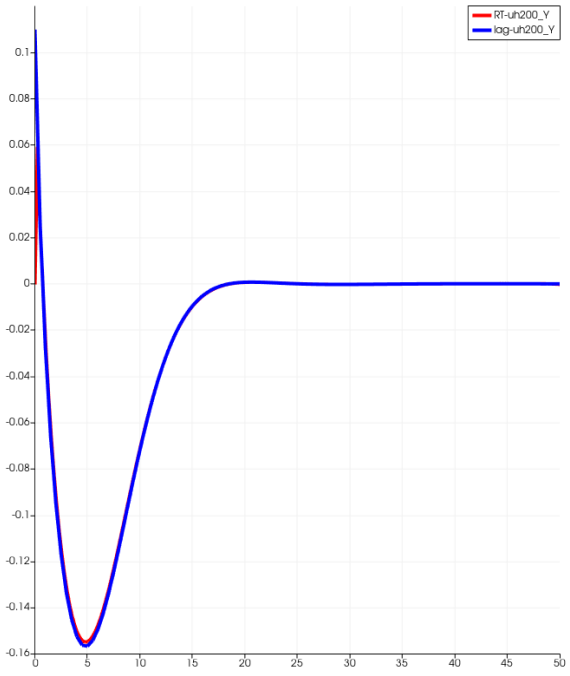


(a) 50×7

(b) 100×14



(c) 150×21

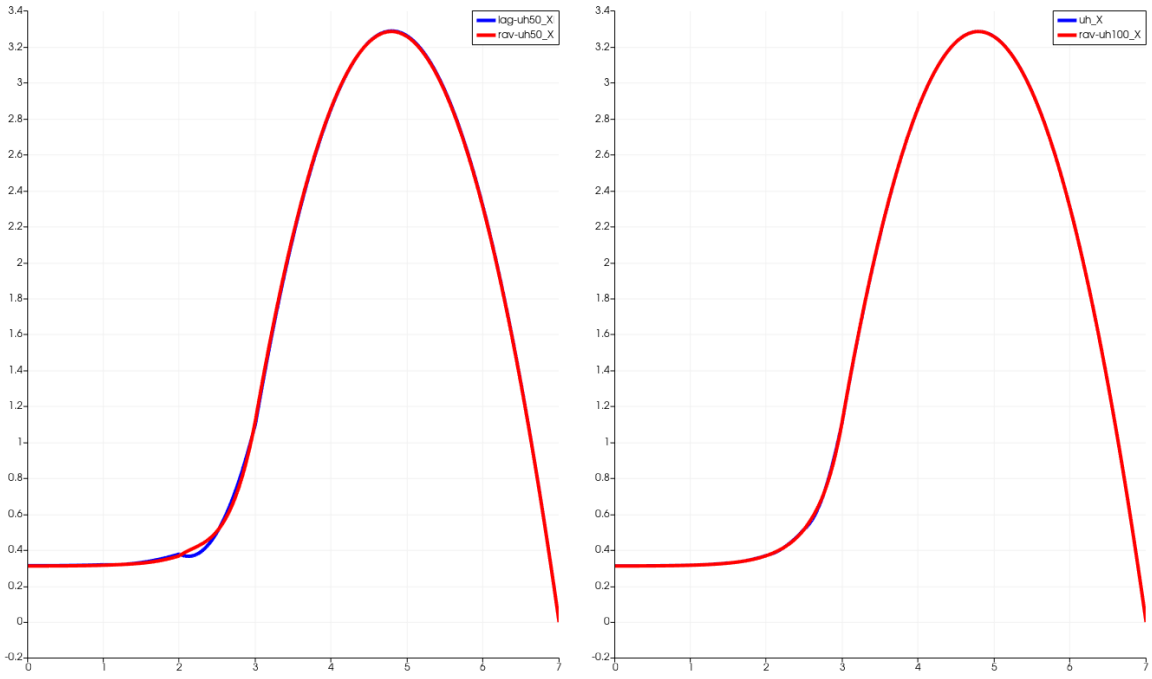


(d) 200×28

Figure 6.16: PE: interface velocity for various gridsizes, comparing $PE_{RT,2}$ (red) with $PE_{Lag,1}$ (blue) elements

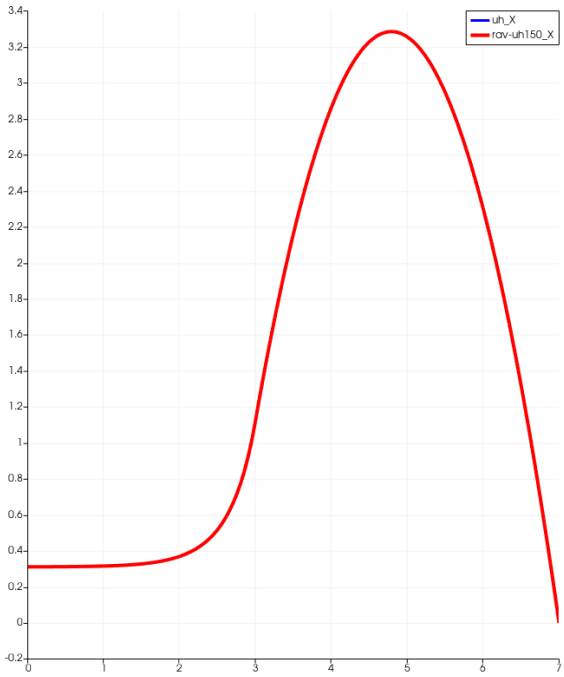
We now clearly see the effect that we discussed above. The two solutions have the same overall shape, but the RT solution is — in a sense — already "closer" to the exact solution than Lagrange. This is most notable for the outlet velocity. For the interface velocity, we don't see much difference between the two approaches, except the zero velocity at the origin we already mentioned, and a slight discrepancy for Lagrange near the end of the channel in Figure 6.16a.

Next, we compare $PE_{RT,3}$ with $PE_{Lag,2}$ elements. See Figure 6.17 and 6.18.

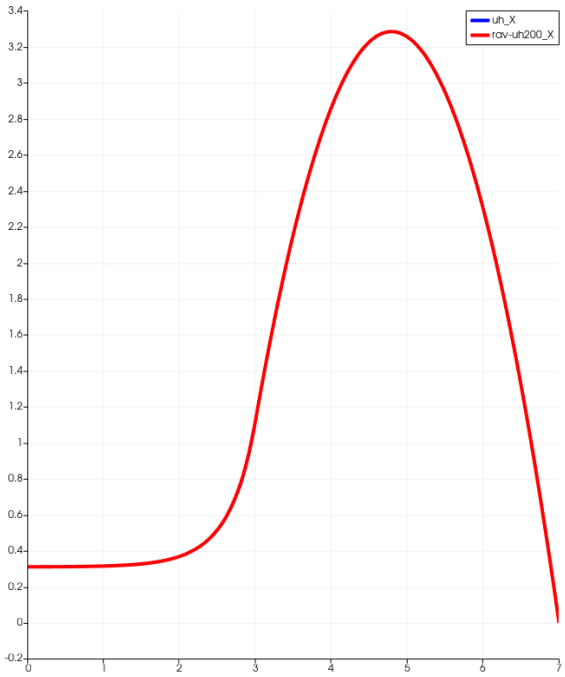


(a) 50×7

(b) 100×14

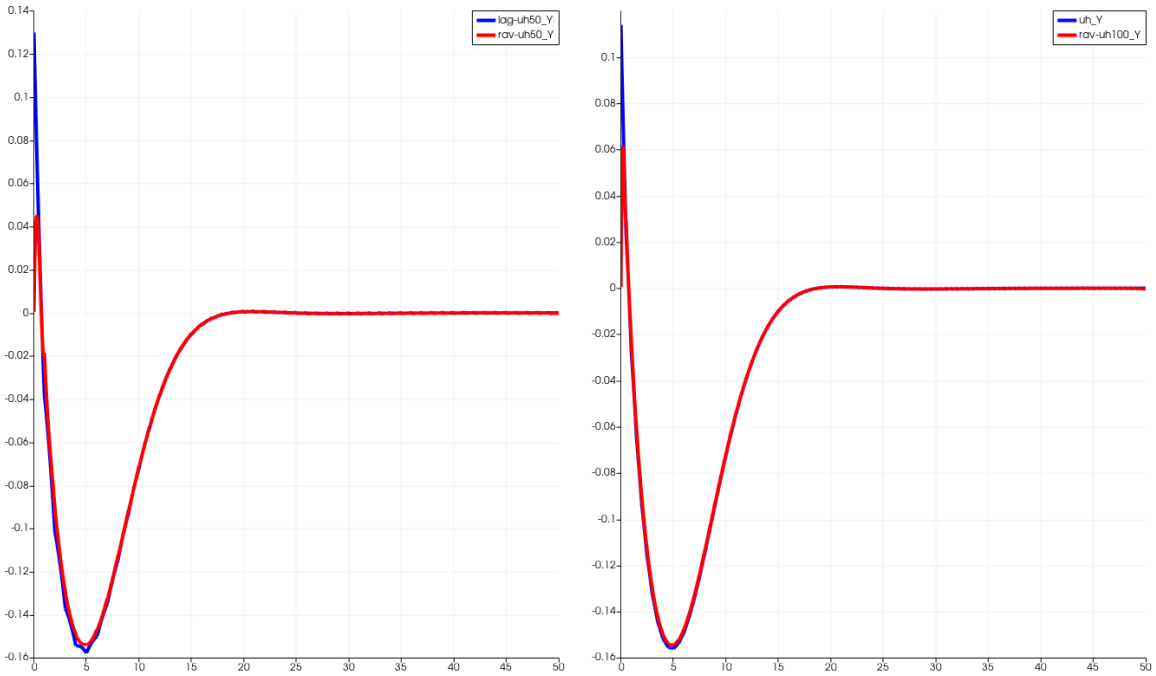


(c) 150×21



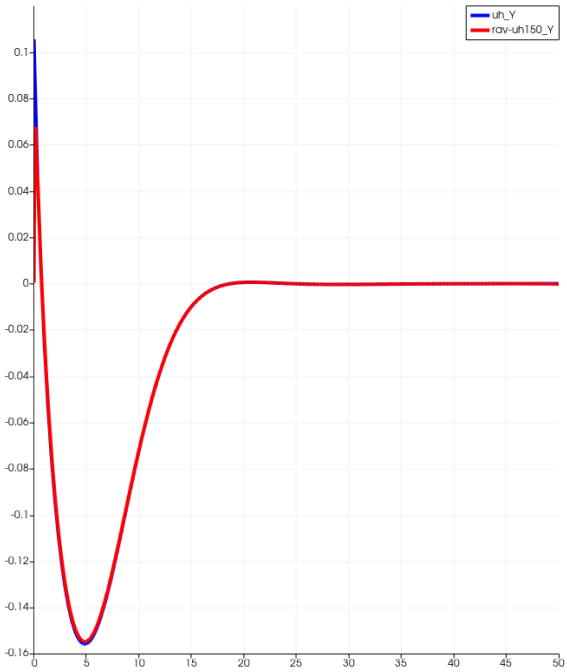
(d) 200×28

Figure 6.17: PE: outlet velocity for various gridsizes, comparing $PE_{RT,3}$ (red) with $PE_{Lag,2}$ (blue) elements

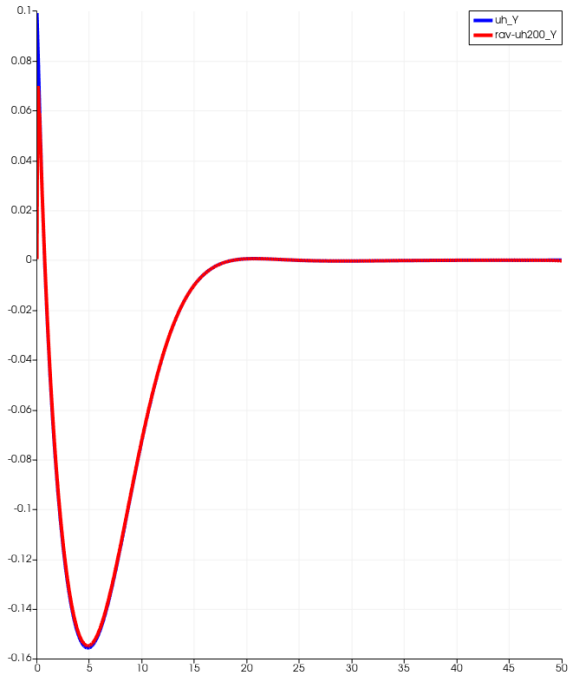


(a) 50×7

(b) 100×14



(c) 150×21



(d) 200×28

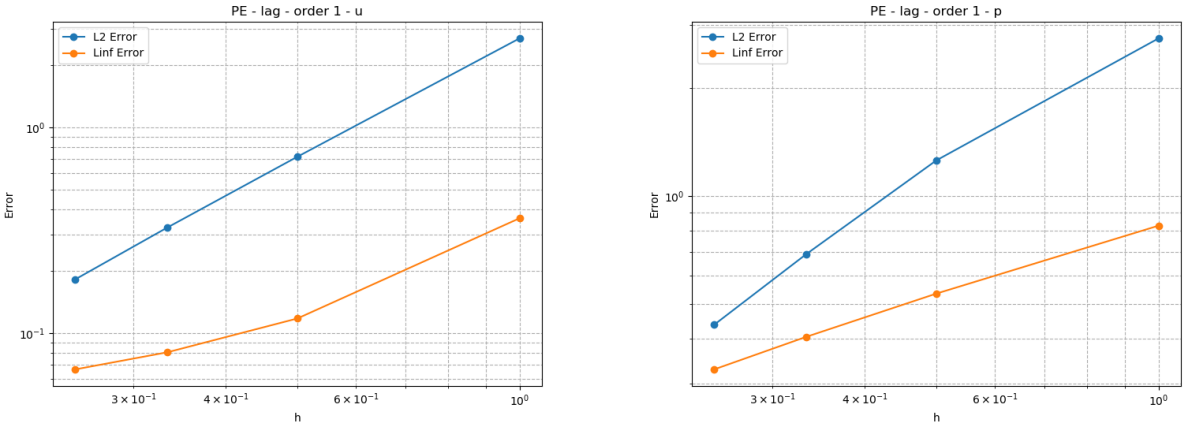
Figure 6.18: PE: interface velocity for various gridsizes, comparing $PE_{RT,3}$ (red) with $PE_{Lag,2}$ (blue) elements

In these figures, we see again, that RT approximates the solution faster, than Lagrange. However, Lagrange catches up more quickly than in the lower order case.

6.2.3 Error Analysis

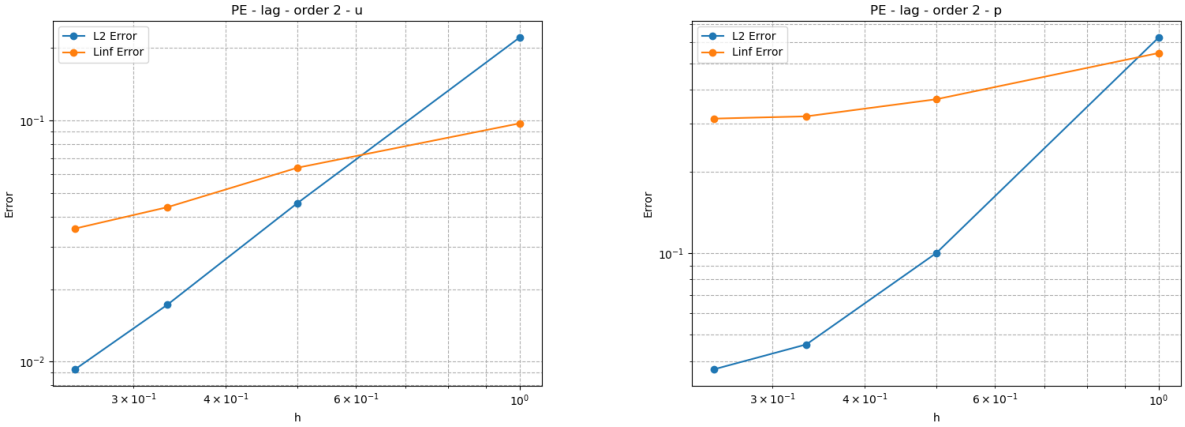
Now, these were just comparisons that visually confirmed that the two methods are close. In order to get a more rigorous proof of this fact, we calculate the errors. With these, we will estimate the convergence rate of the errors. For this, we ran the code on the Delftblue supercomputer with the very fine grid partition in order to obtain a reference solution. This was 1000×140 for the lowest order, and 800×112 for the higher orders.

We then ran the code for the four gridsizes we discussed above, and calculated the L^2 and L^∞ errors for the velocity and the pressure with respect to the reference solution. First we show the errors for $PE_{Lag,r}$, $r = 1, 2, 3$ in Figure 6.19, 6.20, and 6.21.



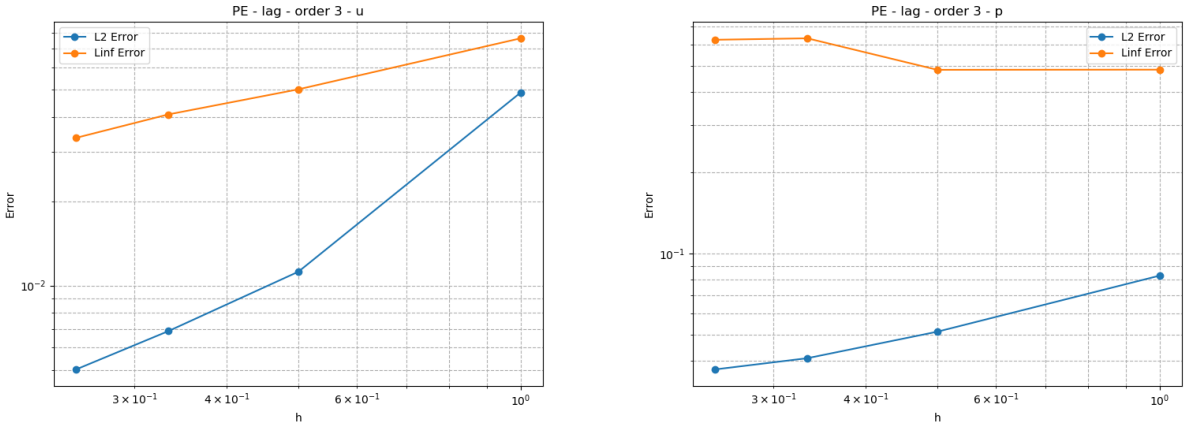
(a) u (b) p

Figure 6.19: $PE_{lag,1}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u (b) p

Figure 6.20: $PE_{Lag,2}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u (b) p

Figure 6.21: $PE_{Lag,3}$ - L^2 and L^∞ errors in u and p for various gridsizes

In order to estimate the rate of convergence, we calculate the slopes of each segment using

$$k = \frac{\log(e_i/e_{i+1})}{\log(h_i/h_{i+1})}, \tag{6.2}$$

where $h_i = 1/i$, $i = 1, 2, 3, 4$, corresponding to the gridsizes we have been using, and e_i is the (L^2 or L^∞) error at h_i. The results for Lagrange are shown in Table 6.4 and 6.5. In the tables, segment i corresponds to the segment between h_i and h_{i+1}.

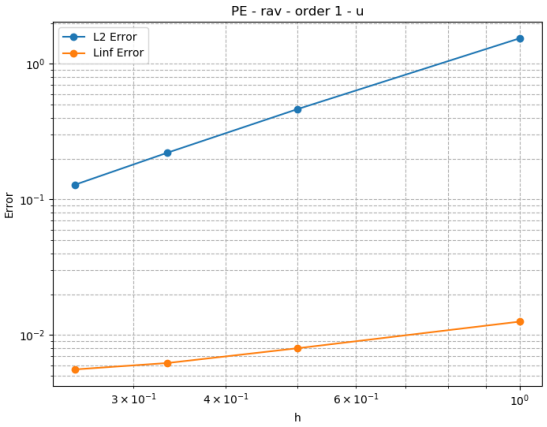
	u			p		
segment	$PE_{lag,1}$	$PE_{lag,2}$	$PE_{lag,3}$	$PE_{lag,1}$	$PE_{lag,2}$	$PE_{lag,3}$
1	1.911	2.283	2.126	1.130	2.641	0.692
2	1.952	2.393	1.205	1.483	1.915	0.560
3	2.019	2.143	1.095	1.577	0.733	0.334

Table 6.4: PE, Lagrange - Estimated rates of convergence of the L^2 error

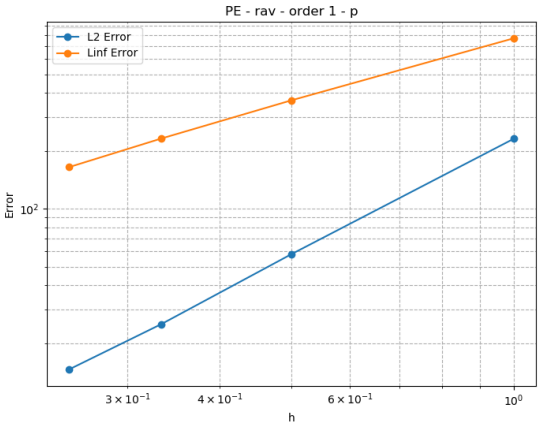
	u			p		
segment	$PE_{lag,1}$	$PE_{lag,2}$	$PE_{lag,3}$	$PE_{lag,1}$	$PE_{lag,2}$	$PE_{lag,3}$
1	1.618	0.612	0.605	0.629	0.566	0.002
2	0.933	0.930	0.509	0.684	0.358	-0.665
3	0.666	0.708	0.669	0.728	0.067	0.043

Table 6.5: PE, Lagrange - Estimated rates of convergence of the L^∞ error

Now we do the same for RT. See Figure 6.22, 6.23, 6.24 for the plots, and Table 6.6, 6.7 for the calculated slopes.

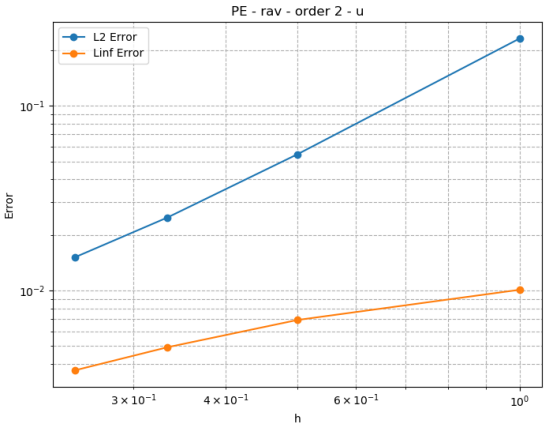


(a) u

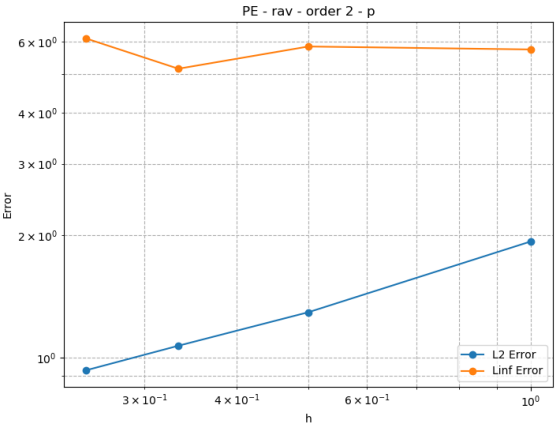


(b) p

Figure 6.22: $PE_{rav,1}$ - L^2 and L^∞ errors in u and p for various gridsizes

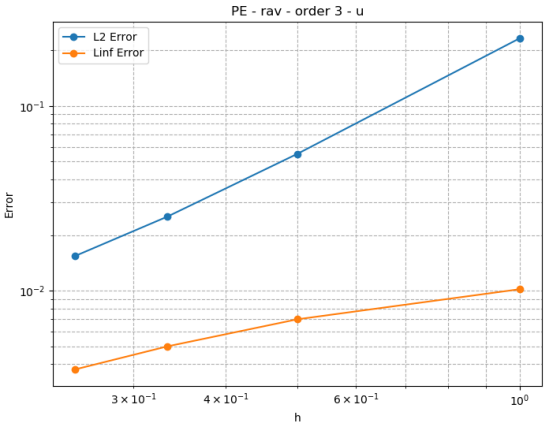


(a) u

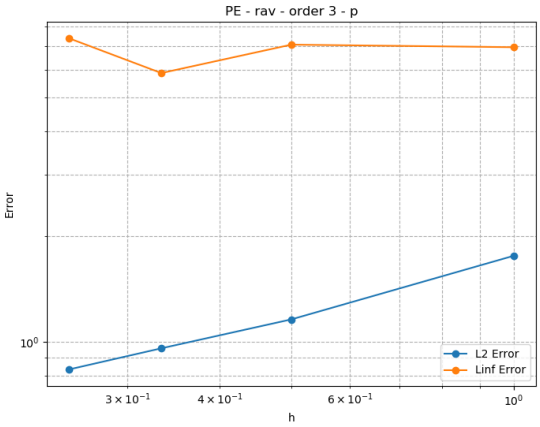


(b) p

Figure 6.23: $PE_{rav,2}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u



(b) p

Figure 6.24: $PE_{rav,3}$ - L^2 and L^∞ errors in u and p for various gridsizes

	u			p		
segment	$PE_{rav,1}$	$PE_{rav,2}$	$PE_{rav,3}$	$PE_{rav,1}$	$PE_{rav,2}$	$PE_{rav,3}$
1	1.735	2.083	2.077	1.994	0.579	0.604
2	1.825	1.945	1.929	2.065	0.469	0.470
3	1.886	1.719	1.705	1.887	0.483	0.484

Table 6.6: PE, Raviart-Thomas - Estimated rates of convergence of the L^2 error

	u			p		
segment	$PE_{rav,1}$	$PE_{rav,2}$	$PE_{rav,3}$	$PE_{rav,1}$	$PE_{rav,2}$	$PE_{rav,3}$
1	0.653	0.544	0.541	1.072	-0.024	-0.024
2	0.618	0.839	0.832	1.128	0.311	0.462
3	0.377	0.997	0.998	1.191	-0.596	-0.796

Table 6.7: PE, Raviart-Thomas - Estimated rates of convergence of the L^∞ error

Sadly, it looks like we can't draw any meaningful conclusion from these results. We would expect the rate of convergence to be dependent on the order of the chosen element. In particular, the L^2 error convergence rate is expected to be $r+1$ for u and r for p using elements of order r. We see that this is roughly true for the first order, but after that there is seemingly no correlation. Especially the L^∞ error behaves rather strangely, even growing for higher orders as the gridsize gets finer. Our best guess as to why this is happening is twofold: firstly, the reference solution is just not computed on a fine enough grid. The reference grid is 4 to 5 times finer than the finest calculation grid, which is not a huge factor. Unfortunately, we could not go any higher due to memory limits, but also convergence limits. Secondly, we saw some inaccuracies in the visual plots, which may have contributed to the convergence rates going astray. Since the reference solution was calculated in the same way as the other solutions, it would have to include the same inaccuracies. We note that we only saw these anomalies in the case of Raviart-Thomas, but they may nonetheless be present for Lagrange as well.

In order to get a bit more insight, we also generated visualisations of the error. The error of the magnitude is plotted for each order for Lagrange in Figure 6.25 and for RT in Figure 6.26 (we were not able to generate the error plots for $PE_{RT,3}$). We see that the errors are mostly along the interface, but even more present at the origin. We also plot the pressure errors in Figure 6.27 and 6.28. Here, we see a stark difference between the two element types, where Lagrange has it biggest error spike at the origin, while RT has it in the upper left corner. This is not very surprising, considering the pressure spike we saw there for RT elements before. We found that the error in that specific point did decrease as the gridsize was refined, indicating that the spike does converge to zero eventually. However, in the reference solution itself, the spike is still present with quite a high value.

(a) order 1

(b) order 2

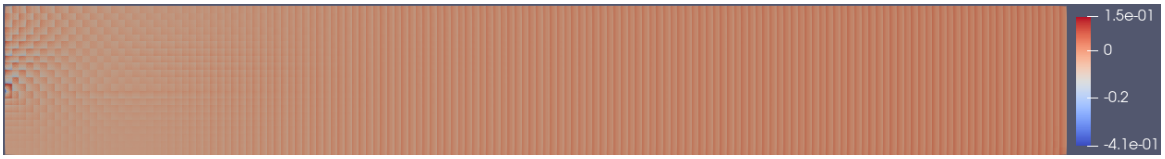
(c) order 3

Figure 6.25: PE: errors of the magnitude of the velocity using Lagrange elements (150×21)

(a) order 1

(b) order 2

Figure 6.26: PE: errors of the magnitude of the velocity using RT elements (150×21)



(a) order 1

(b) order 2

(c) order 3

Figure 6.27: PE: errors of the pressure using Lagrange elements (150×21)

(a) order 1

(b) order 2

Figure 6.28: PE: errors of the pressure using RT elements (150×21)

6.3 NSD with Raviart-Thomas Elements

Now we look at the NSD case. For Lagrange, we define $NSD_{Lag,r}$ as using $P_r - P_{r-1}$ elements in Ω_f and $P_r - P_r$ elements in Ω_p. For RT, we use $RT_r - P_{r-1}$ elements in Ω_f and $P_r - P_r$ elements in Ω_p, indicated with $NSD_{RT,r}$.

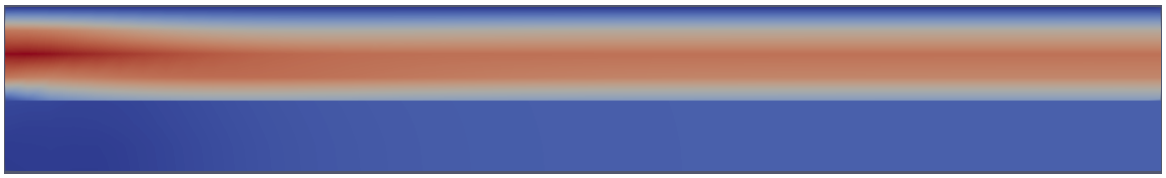
6.3.1 Divergence and overall plots

Let us first look at $NSD_{Lag,1}$ and $NSD_{RT,1}$. See Table 6.8 for the iteration and divergence results. We split the divergence over the fluid and porous domains, as we only use RT elements in Ω_f. We see that again, the divergence is zero, there where RT elements are used.

	$NSD_{Lag,1}$			$NSD_{RT,1}$		
gridsize	#iterations	divergence		#iterations	divergence	
		Ω_f	Ω_p		Ω_f	Ω_p
50×7	6	0.283094	0.093065	4	7.229132e-16	2.566399
100×14	6	0.215909	0.070973	4	1.506323e-15	2.377602
150×21	6	0.177210	0.064050	4	3.725336e-15	2.329507
200×28	6	0.151928	0.058725	5	2.570934e-15	2.303233

Table 6.8: Divergence L^2-norm and iteration number for different gridsizes, for the NSD case using $NSD_{Lag,1}$ and $NSD_{RT,1}$ elements

Let us look at the magnitude of the velocity in Figure 6.29. For Lagrange, we see the discontinuity along the interface quite clearly. For RT, we see the same problem as with the $PE_{RT,1}$ elements, where they are over-constrained. Of course, this only occurs in the fluid domain, as we use Lagrange elements in the porous domain, but there seems to be an issue there as well, looking at the bottom right corner.



(a) using $NSD_{Lag,1}$ elements

(b) using $NSD_{RT,1}$ elements

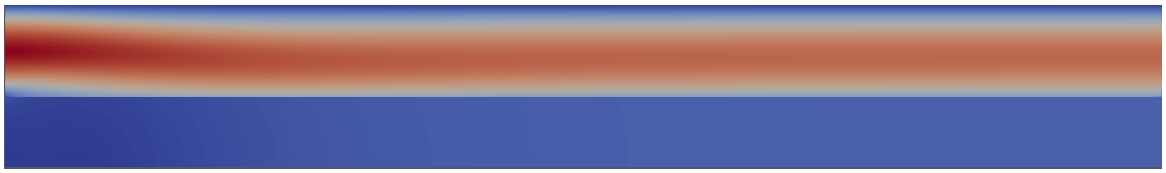
Figure 6.29: NSD: magnitude of the velocity using $NSD_{Lag,1}$ and $NSD_{RT,1}$ elements (50×7)

Next, we look at $NSD_{Lag,2}$ and $NSD_{RT,2}$ elements. The divergence results are given in Table 6.9. We notice, as was the case with PE, that the iteration number is slightly higher for RT. We also note that while the divergence is zero in Ω_f, it is higher in Ω_p for RT than it is for Lagrange.

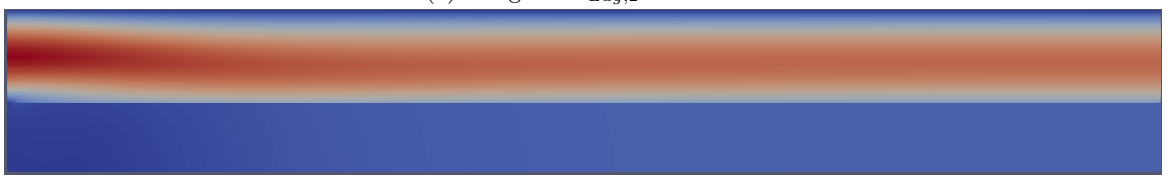
gridsize	$NSD_{Lag,2}$			$NSD_{RT,2}$		
	#iterations	divergence		#iterations	divergence	
		Ω_f	Ω_p		Ω_f	Ω_p
50×7	5	0.149904	0.058556	7	7.746669e-14	0.267217
100×14	5	0.089188	0.045583	7	1.564696e-13	0.260634
150×21	5	0.064256	0.034759	6	2.530011e-13	0.259501
200×28	5	0.053251	0.026435	6	3.145580e-13	0.258924

Table 6.9: Divergence L^2-norm and iteration number for different gridsizes, for the NSD case using $NSD_{Lag,2}$ and $NSD_{RT,2}$ elements

In Figure 6.30, we show the magnitude of the velocity using both Lagrange and RT elements, and we see no real difference between them. Although it is not visible in the figure, there is still a small disturbance in the lower right corner. This disturbance is also the cause of the higher divergence.



(a) using $NSD_{Lag,2}$ elements

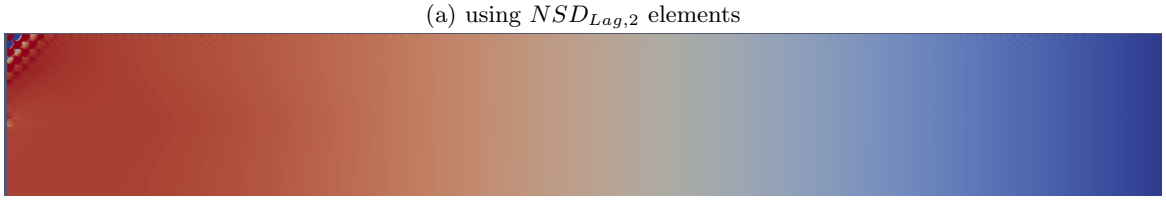


(b) using $NSD_{RT,2}$ elements

Figure 6.30: NSD: magnitude of the velocity using $NSD_{Lag,2}$ and $NSD_{RT,2}$ elements (150×21)

Next, we look at the pressure in Figure 6.31. Again, we rescaled the RT picture to match the values of the Lagrange picture. Thence we can see that the overall picture is the same, save for the same pressure peak in the upper left corner, together with a small disturbance at the origin.

(a) using $NSD_{Lag,2}$ elements



(b) using $NSD_{RT,2}$ elements

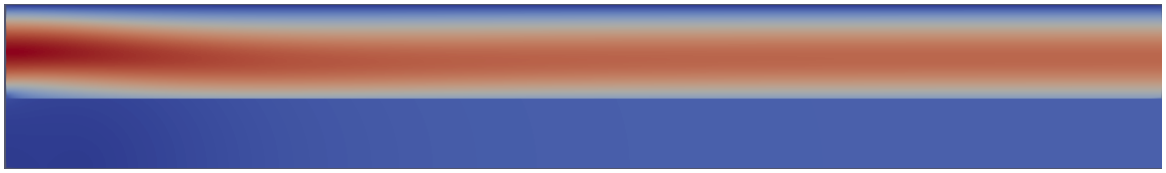
Figure 6.31: NSD: pressure using $NSD_{Lag,2}$ and $NSD_{RT,2}$ elements (150×21)

For $NSD_{RT,3}$ elements, the divergence is even higher than with the second order, though the iteration number has decreased, as is shown in Table 6.10.

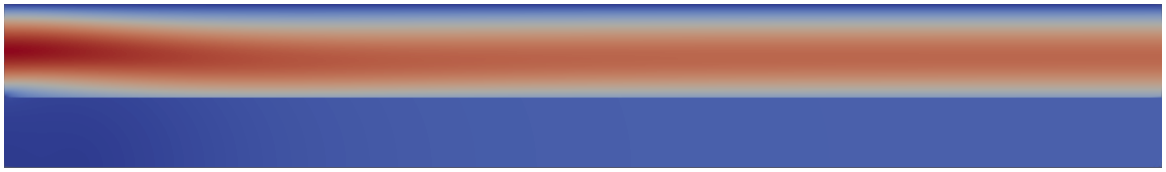
gridsize	$NSD_{Lag,3}$			$NSD_{RT,3}$		
	#iterations	divergence		#iterations	divergence	
		Ω_f	Ω_p		Ω_f	Ω_p
50×7	5	0.087096	0.027439	6	1.787382e-11	0.395749
100×14	5	0.042375	0.012559	5	3.613782e-11	0.395178
150×21	5	0.032708	0.005536	5	5.451205e-11	0.394481
200×28	5	0.030268	0.004054	5	7.253150e-11	0.393838

Table 6.10: Divergence L^2-norm and iteration number for different gridsizes, for the NSD case using $NSD_{Lag,3}$ and $NSD_{RT,3}$ elements

We show the velocity magnitude plots in Figure 6.32. They again look indiscernible.



(a) using $NSD_{Lag,3}$ elements

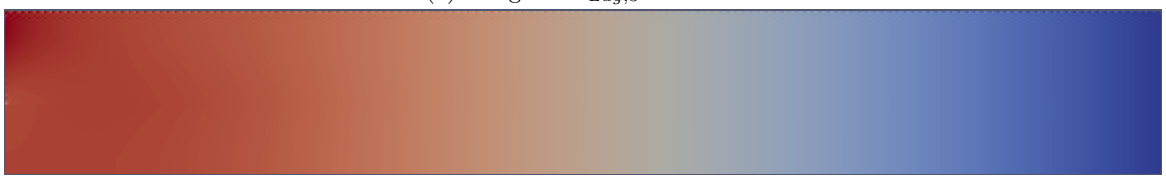


(b) using $NSD_{RT,3}$ elements

Figure 6.32: NSD: magnitude of the velocity using $NSD_{Lag,3}$ and $NSD_{RT,3}$ elements (150×21)

We show the pressure plots in Figure 6.33. The peak has subsided, but there is still some oscillation at the top of the domain.

(a) using $NSD_{Lag,3}$ elements

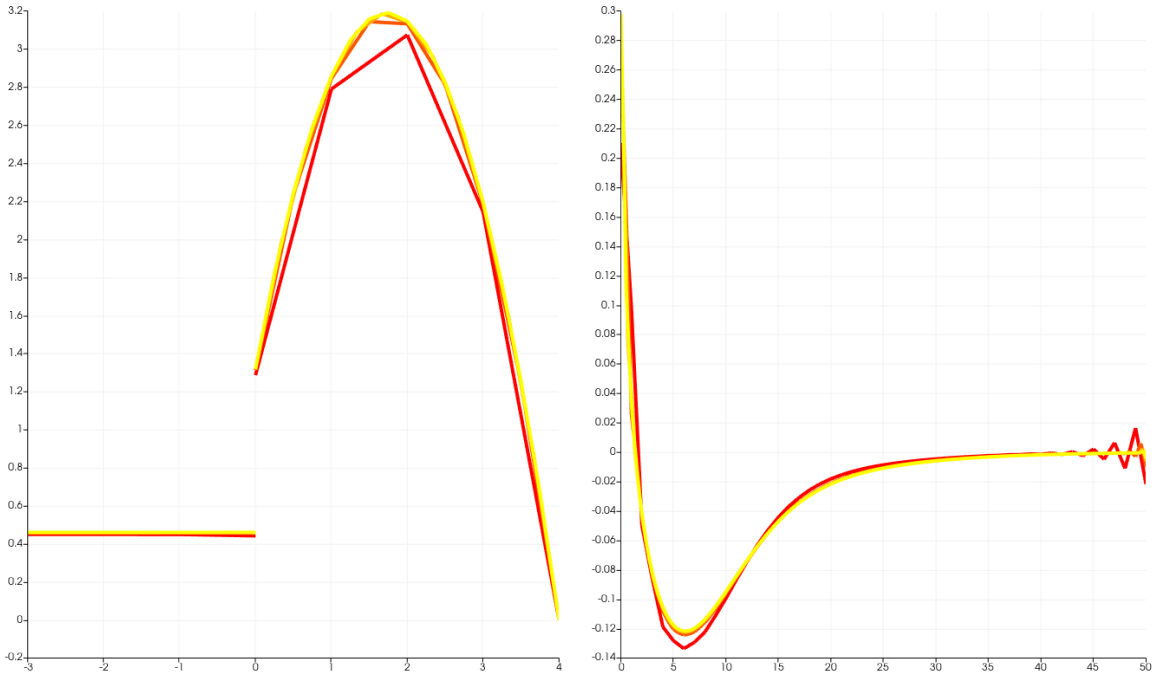


(b) using $NSD_{RT,3}$ elements

Figure 6.33: NSD: pressure using $NSD_{Lag,3}$ and $NSD_{RT,3}$ elements (150×21)

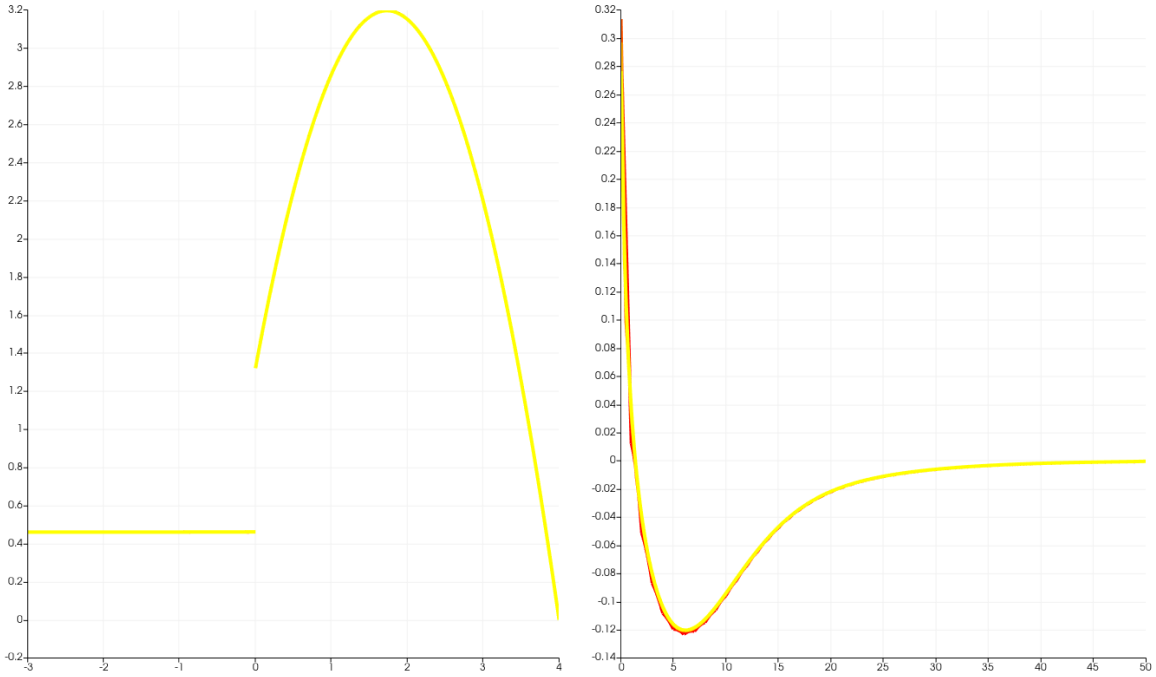
6.3.2 Velocity profiles

Next, we compare the velocity profiles for NSD. The Lagrange cases for orders 1 and 2 are given in Figure 6.34 and 6.35, respectively. For RT, orders 2 and 3 are shown in Figure 6.36 and 6.37, respectively.



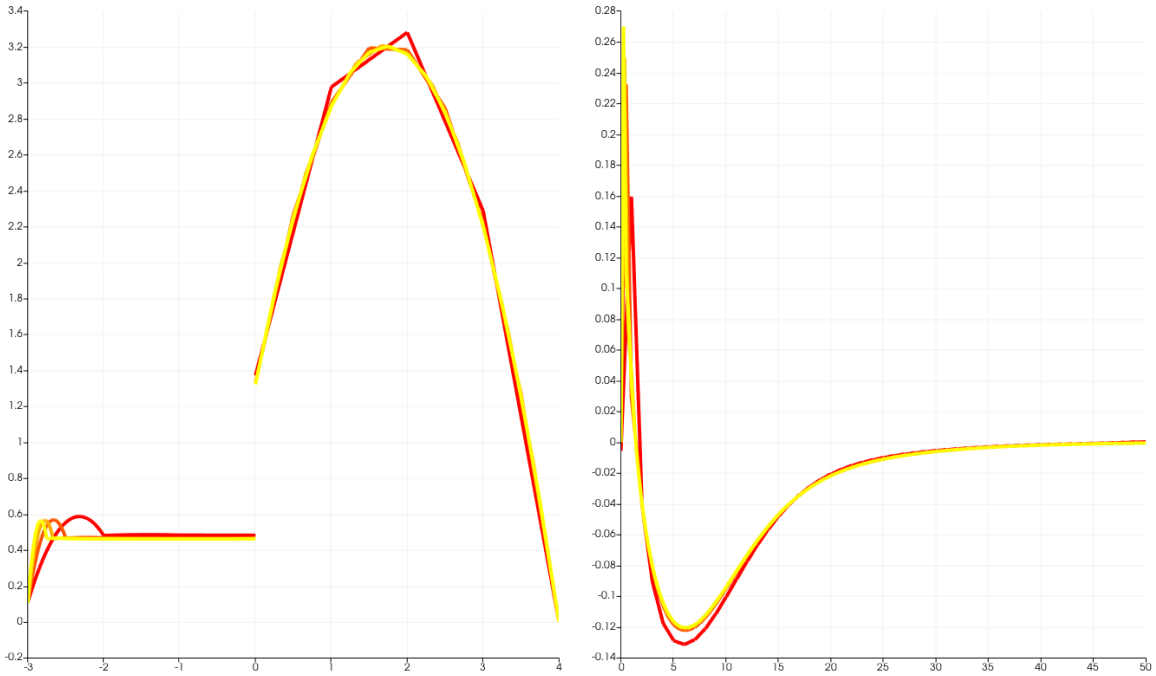
(a) u_x at the outlet (b) u_y at the interface

Figure 6.34: NSD: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSD_{Lag,1}$ elements



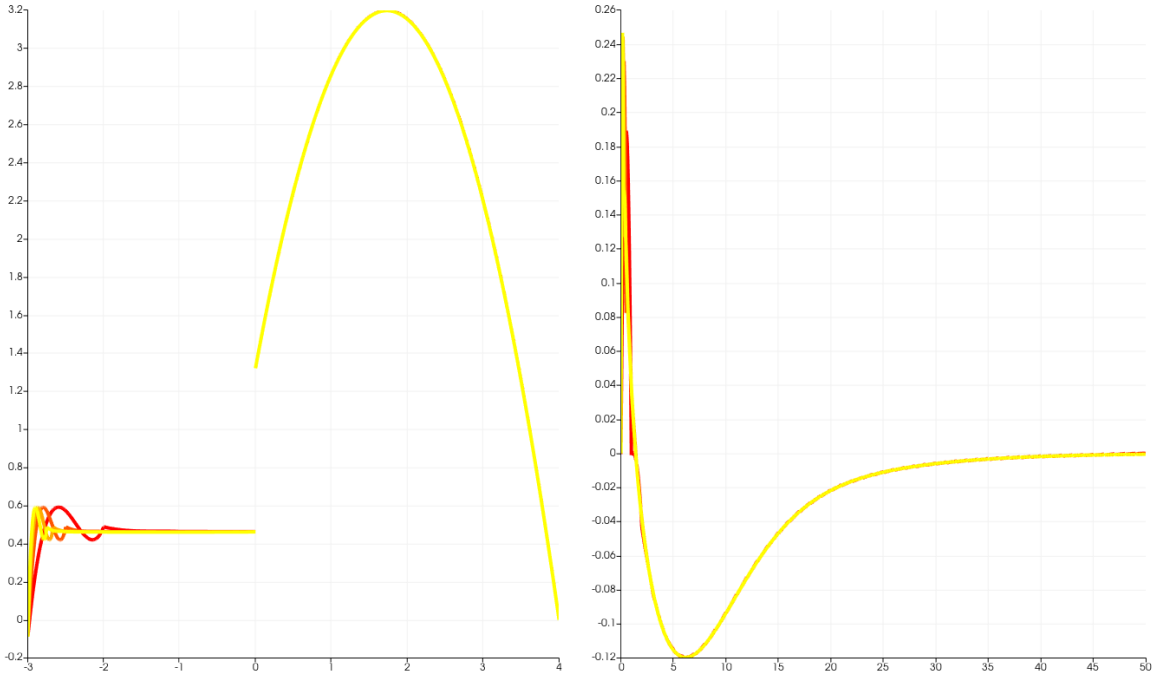
(a) u_x at the outlet (b) u_y at the interface

Figure 6.35: NSD: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSD_{Lag,2}$ elements



(a) u_x at the outlet (b) u_y at the interface

Figure 6.36: NSD: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSD_{RT,2}$ elements



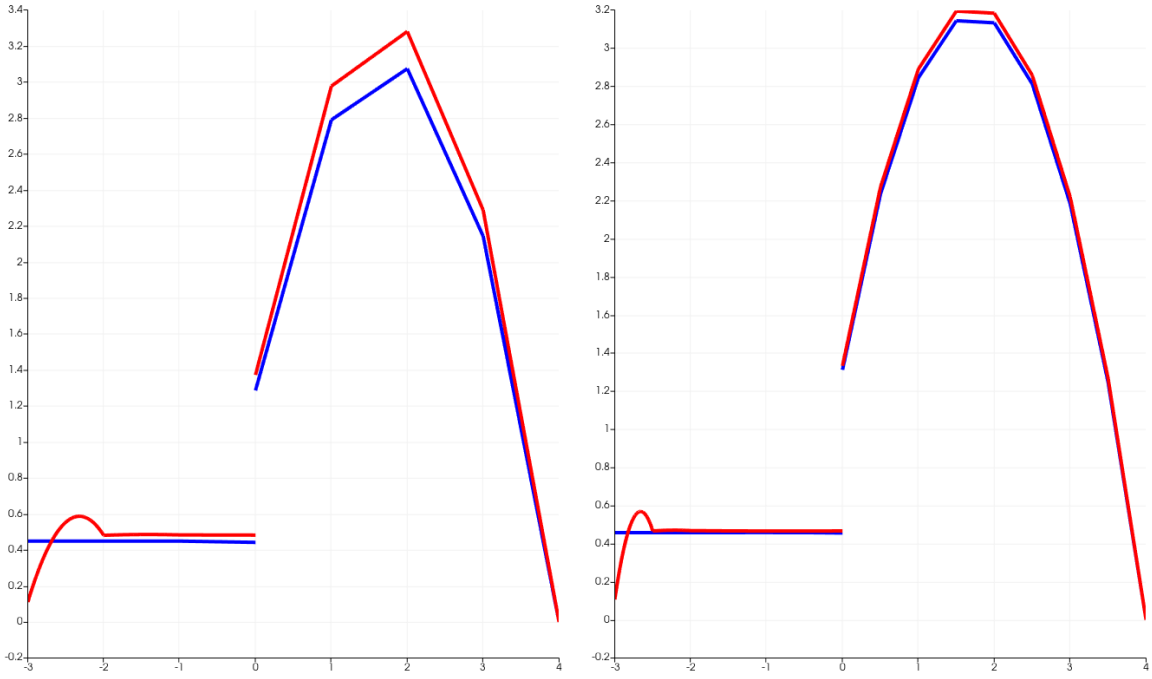
(a) u_x at the outlet (b) u_y at the interface

Figure 6.37: NSD: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSD_{RT,3}$ elements

We see a similar trend to PE, where Raviart-Thomas seemingly ends up at the correct solution sooner than Lagrange, because of how the degrees of freedom are defined. We also see a little oscillation quirk for Lagrange order 1 at the outlet, which isn't present for RT. We do, however, see strange behaviour at the bottom of the

outlet velocity profile. This matches with the disturbance we saw in the overall velocity plot before. It seems that the velocity wants to go to zero there. This is probably the same issue as the value at the origin being zero, as we saw (and see here also) in the interface velocity profiles. This is supported by the fact that the inconsistency only happens in the element adjacent to the bottom boundary.

Next, we compare $NSD_{RT,r}$ with $NSD_{Lag,r-1}$ directly as before. See Figure 6.38 and 6.39 for $NSD_{RT,2}$ versus $NSD_{Lag,1}$, and Figure 6.40 and 6.41 for $NSD_{RT,3}$ versus $NSD_{Lag,2}$. There is not much to say about these figures. We see more clearly the oscillations we saw above for $NSD_{Lag,1}$. For the rest, the results are basically the same as for PE.



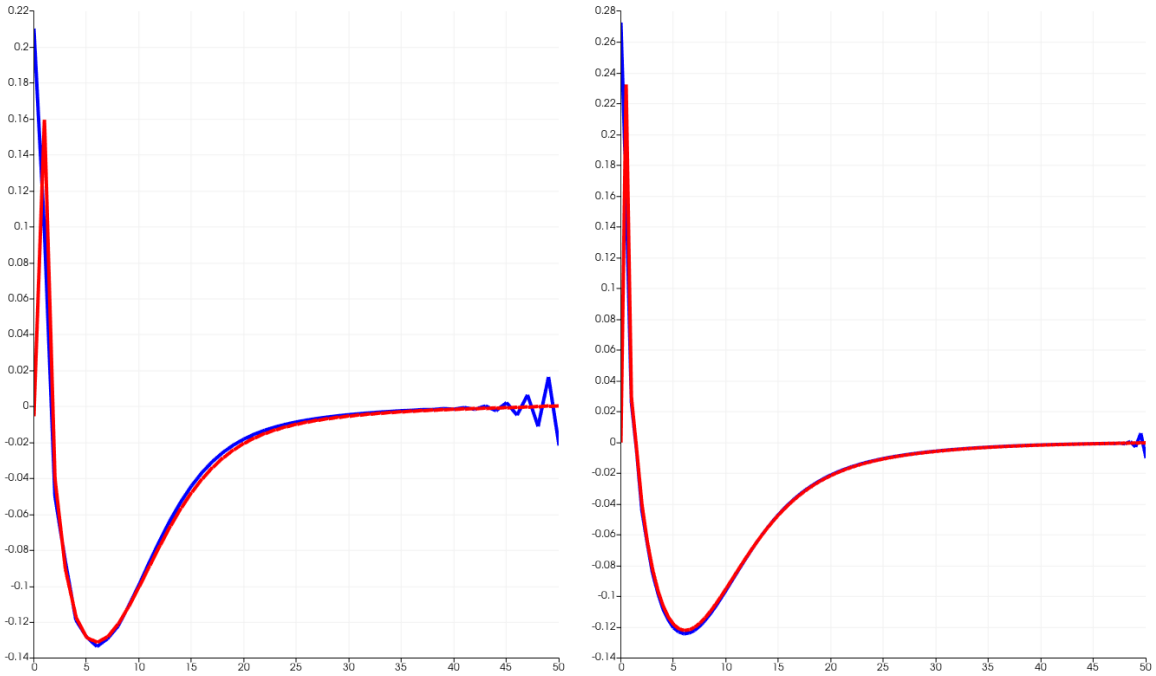
(a) 50×7

(b) 100×14

(c) 150×21

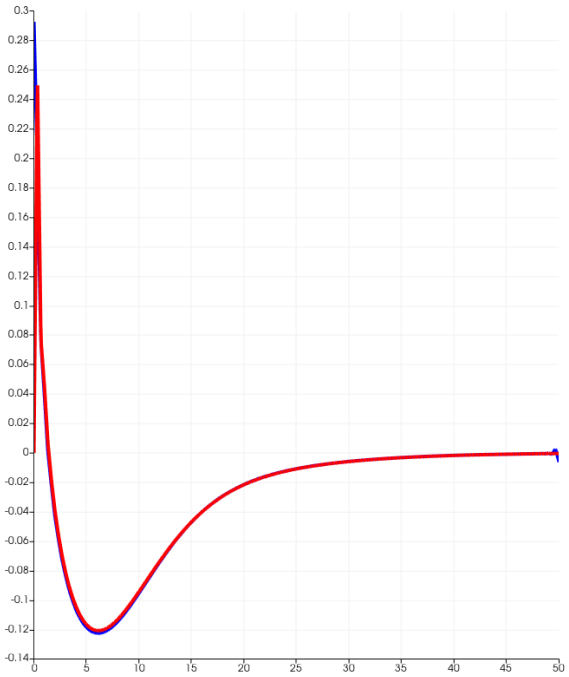
(d) 200×28

Figure 6.38: NSD: outlet velocity for various gridsizes, comparing $NSD_{RT,2}$ (red) with $NSD_{Lag,1}$ (blue) elements



(a) 50×7

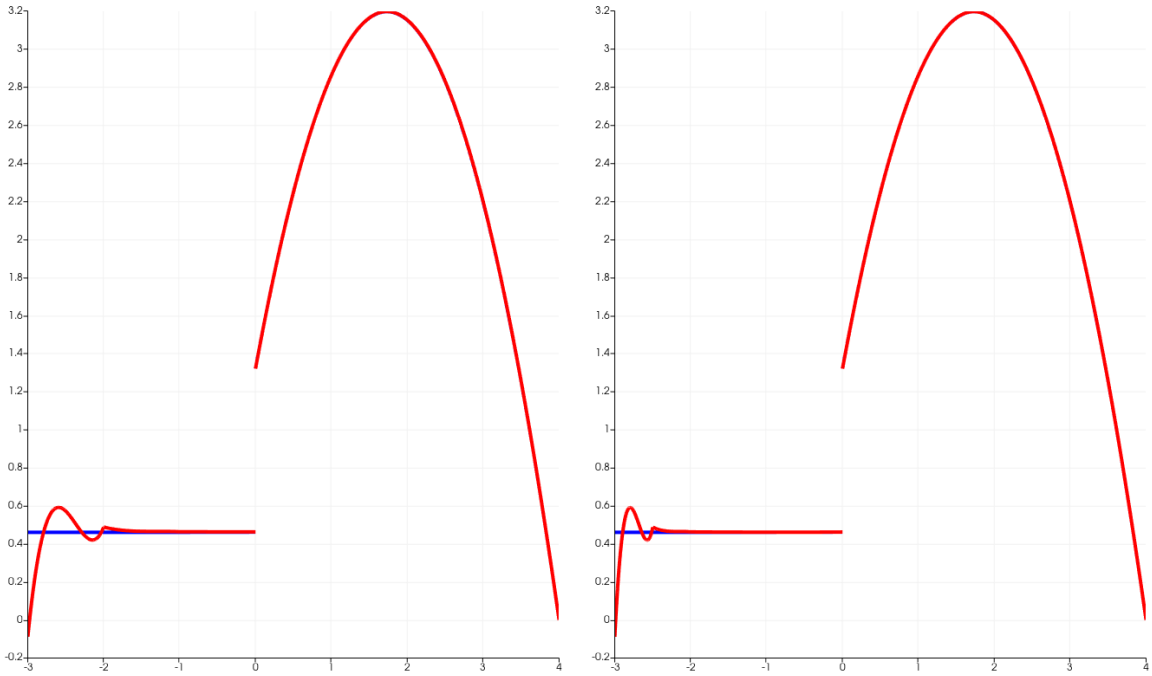
(b) 100×14



(c) 150×21

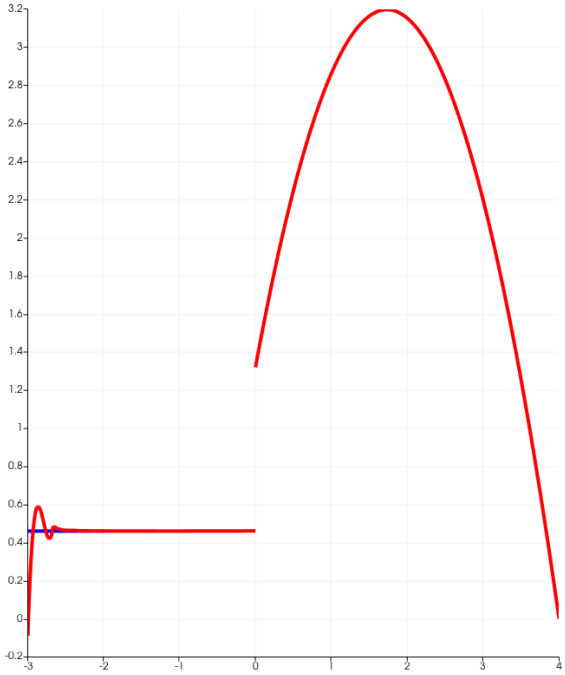
(d) 200×28

Figure 6.39: NSD: interface velocity for various gridsizes, comparing $NSD_{RT,2}$ (red) with $NSD_{Lag,1}$ (blue) elements

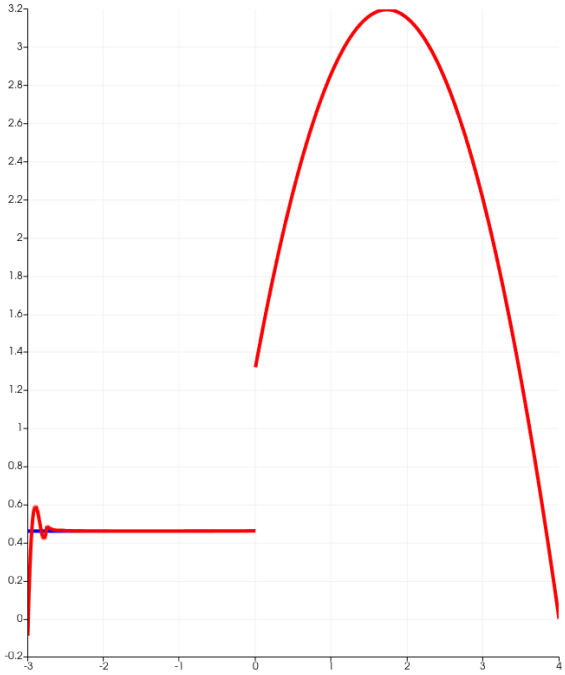


(a) 50×7

(b) 100×14

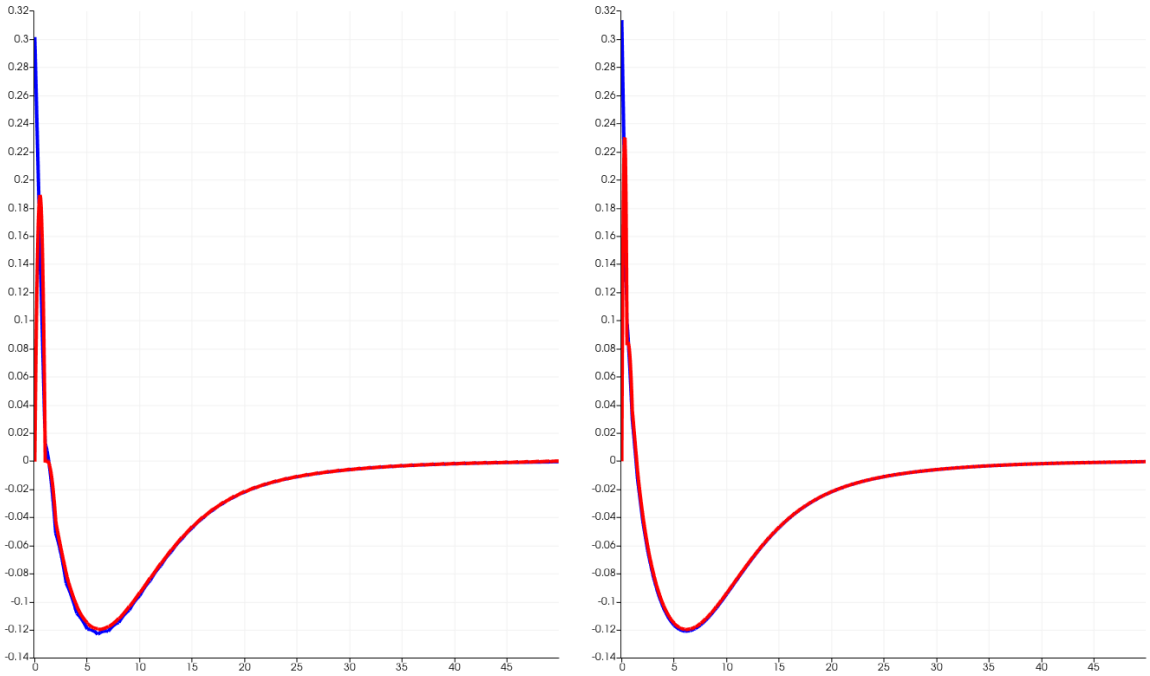


(c) 150×21



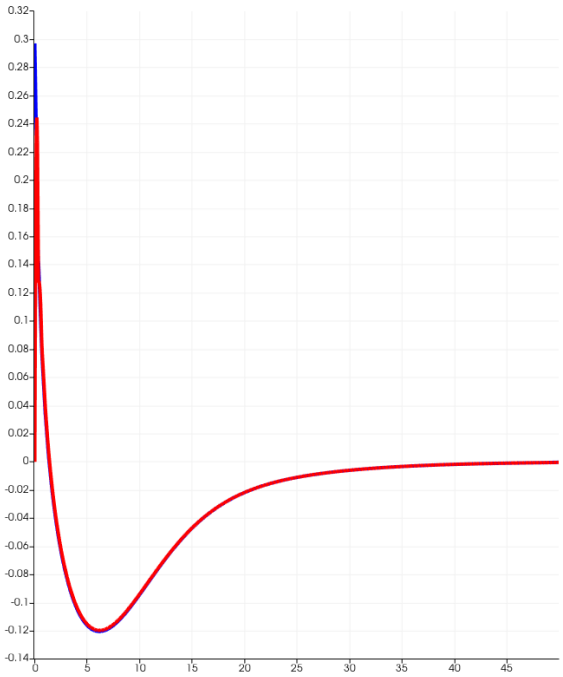
(d) 200×28

Figure 6.40: NSD: outlet velocity for various gridsizes, comparing $NSD_{RT,3}$ (red) with $NSD_{Lag,2}$ (blue) elements

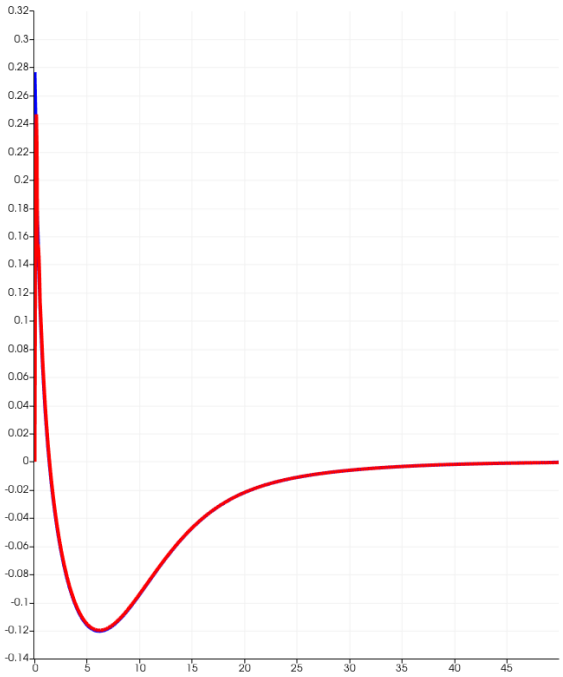


(a) 50×7

(b) 100×14



(c) 150×21

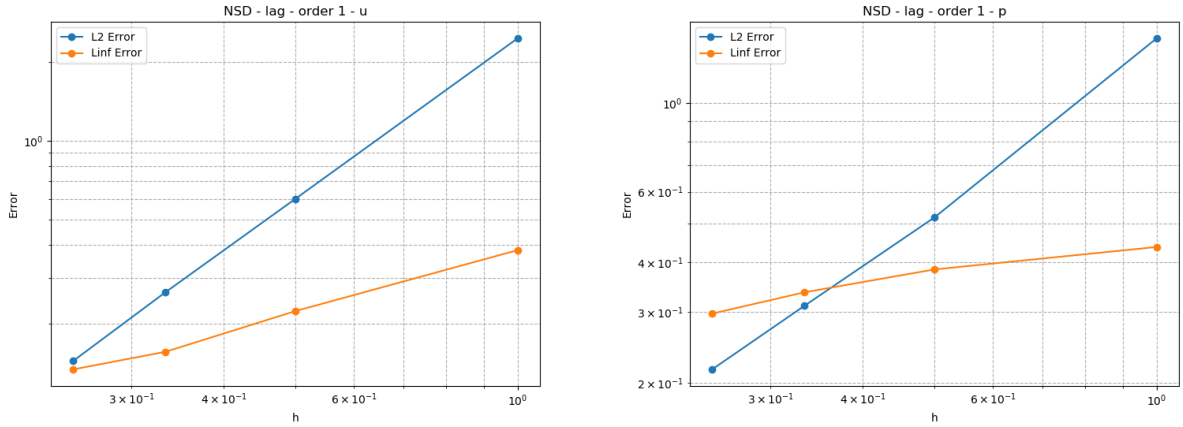


(d) 200×28

Figure 6.41: NSD: outlet velocity for various gridsizes, comparing $NSD_{RT,3}$ (red) with $NSD_{Lag,2}$ (blue) elements

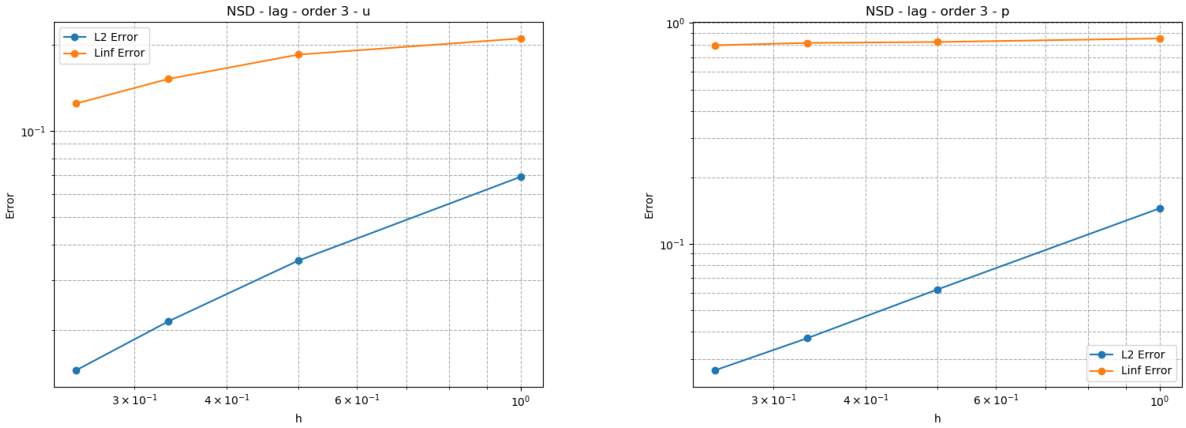
6.3.3 Error analysis

Now we present the error plots, generated as with PE. This time, the gridsize is 800×112 for all reference solution. Unfortunately, the reference solution for $NSD_{lag,2}$ did not converge, so it is left out in the figures. See Figure 6.42 and 6.43 for the results of the Lagrange case.



(a) u (b) p

Figure 6.42: $NSD_{lag,1}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u (b) p

Figure 6.43: $NSD_{Lag,3}$ - L^2 and L^∞ errors in u and p for various gridsizes

We calculate the slopes as with PE. See Table 6.11 and 6.12.

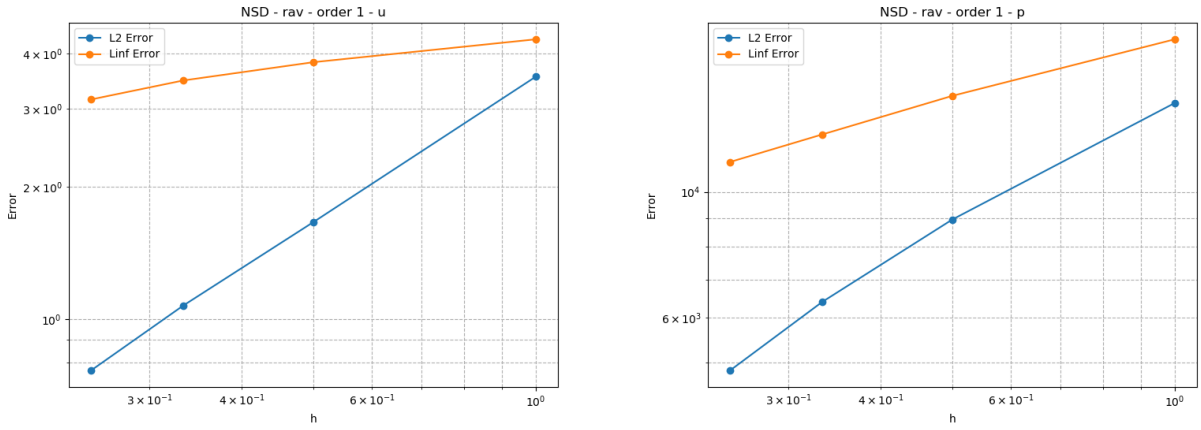
	u			p		
segment	$NSD_{lag,1}$	$NSD_{lag,2}$	$NSD_{lag,3}$	$NSD_{lag,1}$	$NSD_{lag,2}$	$NSD_{lag,3}$
1	2.034	-	0.975	1.488	-	1.222
2	2.025	-	1.210	1.255	-	1.253
3	2.097	-	1.372	1.271	-	1.164

Table 6.11: NSD, Lagrange - Estimated rates of convergence of the L^2 error

	u			p		
segment	$NSD_{lag,1}$	$NSD_{lag,2}$	$NSD_{lag,3}$	$NSD_{lag,1}$	$NSD_{lag,2}$	$NSD_{lag,3}$
1	0.772	-	0.188	0.187	-	0.053
2	0.882	-	0.484	0.324	-	0.023
3	0.537	-	0.690	0.427	-	0.086

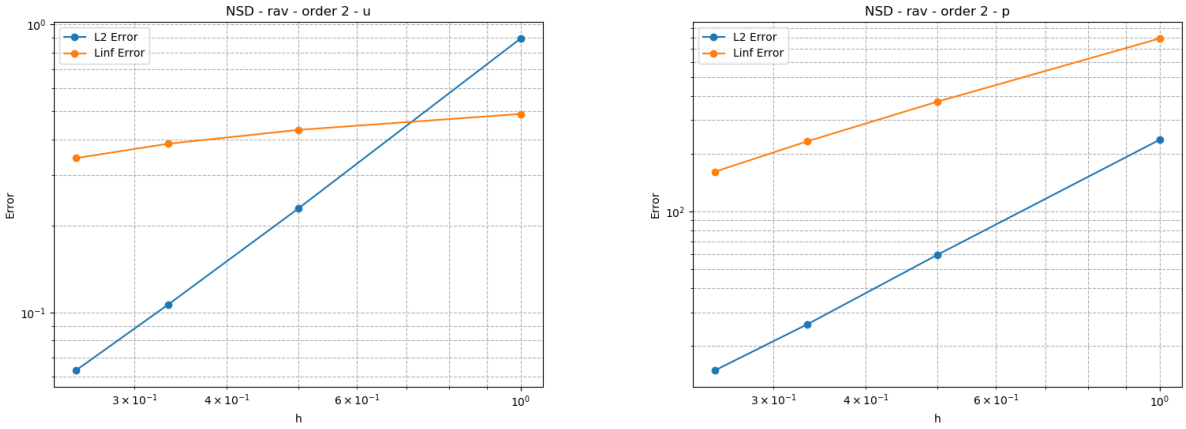
Table 6.12: NSD, Lagrange - Estimated rates of convergence of the L^∞ error

And we do the same for RT in Figure 6.44, 6.45, 6.46, and in Table 6.13 and 6.14.



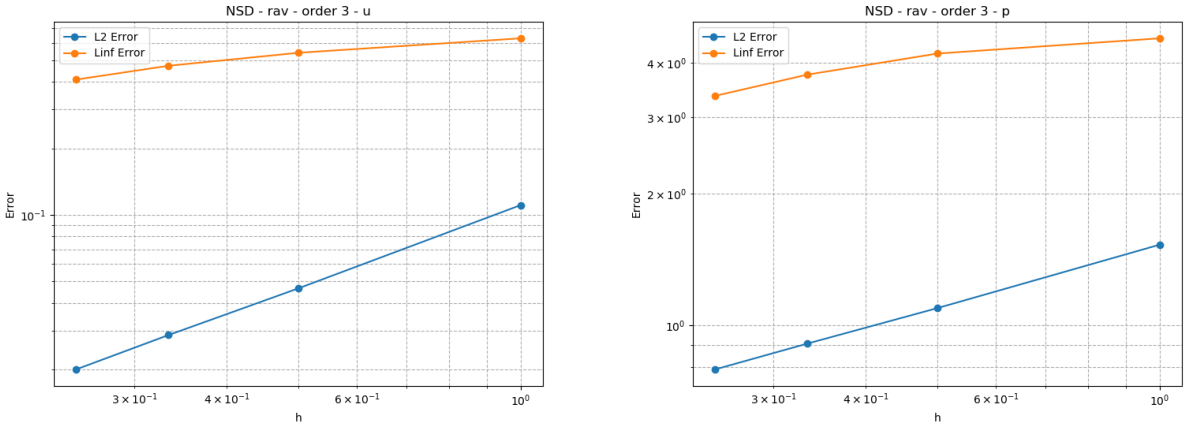
(a) u (b) p

Figure 6.44: $NSD_{rav,1}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u (b) p

Figure 6.45: $NSD_{rav,2}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u (b) p

Figure 6.46: $NSD_{rav,3}$ - L^2 and L^∞ errors in u and p for various gridsizes

	u			p		
segment	$NSD_{RT,1}$	$NSD_{RT,2}$	$NSD_{RT,3}$	$NSD_{RT,1}$	$NSD_{RT,2}$	$NSD_{RT,3}$
1	1.094	1.960	1.250	0.681	1.988	0.484
2	1.073	1.891	1.200	0.824	2.051	0.463
3	1.177	1.830	1.242	0.967	1.914	0.474

Table 6.13: NSD, Raviart-Thomas - Estimated rates of convergence of the L^2 error

	u			p		
segment	$NSD_{RT,1}$	$NSD_{RT,2}$	$NSD_{RT,3}$	$NSD_{RT,1}$	$NSD_{RT,2}$	$NSD_{RT,3}$
1	0.172	0.184	0.218	0.331	1.092	0.116
2	0.235	0.273	0.334	0.386	1.171	0.275
3	0.345	0.398	0.498	0.388	1.261	0.390

Table 6.14: NSD, Raviart-Thomas - Estimated rates of convergence of the L^∞ error

Again, there are no meaningful conclusions to be taken from these results. As with PE, the first order results very roughly line up with expectation, but the rest goes off the rails completely. We do see a slight improvement over PE, in that the L^∞ never grows; it is always converging, albeit slowly. Since our methods of generating these results were the same, and there were anomalies in this case as well, we cannot confirm our speculations from before; we can only infer that they also apply here.

6.4 NSF with Raviart-Thomas elements

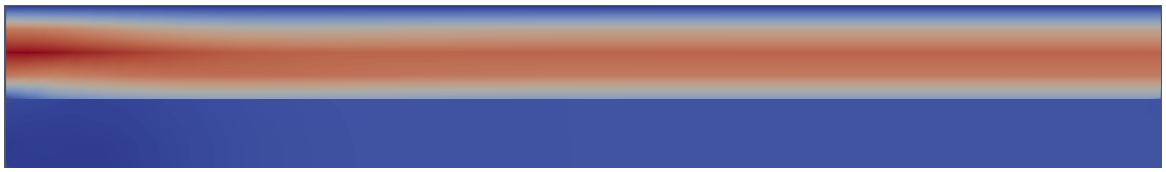
Now we look at NSF. We use the same element arrangement as with NSD, but notate them with $NSF_{Lag,r}$ and $NSF_{RT,r}$ for consistency's sake.

6.4.1 Divergence and overall plots

For $NSF_{Lag,1}$ and $NSF_{RT,1}$, the results for the divergence are shown in Table 6.15, and overall velocity magnitude plots are given in Figure 6.47. The results are very similar to those of NSD.

gridsize	$NSF_{Lag,1}$			$NSF_{RT,1}$		
	#iterations	divergence		#iterations	divergence	
		Ω_f	Ω_p		Ω_f	Ω_p
50×7	5	0.278751	0.094216	6	7.184466e-16	3.491497
100×14	5	0.214343	0.058023	6	1.204612e-15	3.169146
150×21	5	0.176215	0.056096	6	3.671431e-15	3.135605
200×28	5	0.151127	0.044511	6	3.050750e-15	3.098366

Table 6.15: Divergence L^2-norm and iteration number for different gridsizes, for the NSF case using $NSF_{Lag,1}$ and $NSF_{RT,1}$ elements



(a) using $NSF_{Lag,1}$ elements

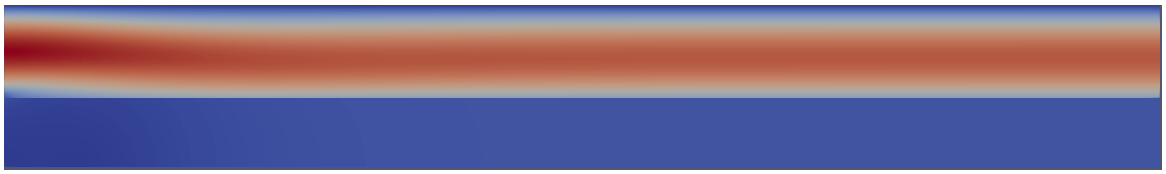
(b) using $NSF_{RT,1}$ elements

Figure 6.47: NSF: magnitude of the velocity using $NSD_{Lag,1}$ and $NSD_{RT,1}$ elements (50×7)

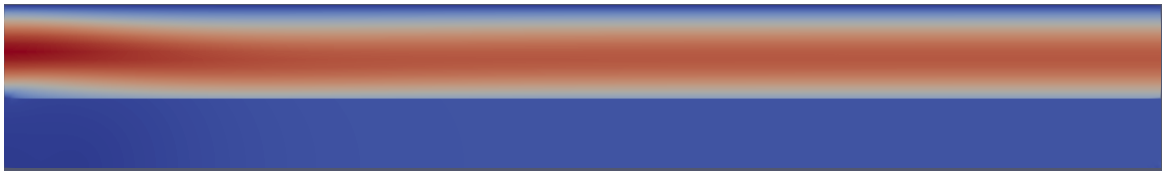
For order 2, the divergence results are given in Table 6.16. The velocity magnitude plots are in Figure 6.48. The pressure plots are in Figure 6.49. Again, the results are comparable to NSD: the divergence is zero in Ω_f for RT, but higher in Ω_p when compared to Lagrange. The velocity plots look similar, and the pressure has spikes in the same locations.

gridsize	$NSF_{Lag,2}$			$NSF_{RT,2}$		
	#iterations	divergence		#iterations	divergence	
		Ω_f	Ω_p		Ω_f	Ω_p
50×7	5	0.149633	0.052865	6	7.907837e-14	0.203865
100×14	5	0.089218	0.037765	6	1.614568e-13	0.200441
150×21	5	0.064166	0.026656	6	2.598224-13	0.199852
200×28	5	0.053106	0.018997	6	3.226754-13	0.1992515

Table 6.16: Divergence L^2-norm and iteration number for different gridsizes, for the NSF case using $NSF_{Lag,2}$ and $NSF_{RT,2}$ elements



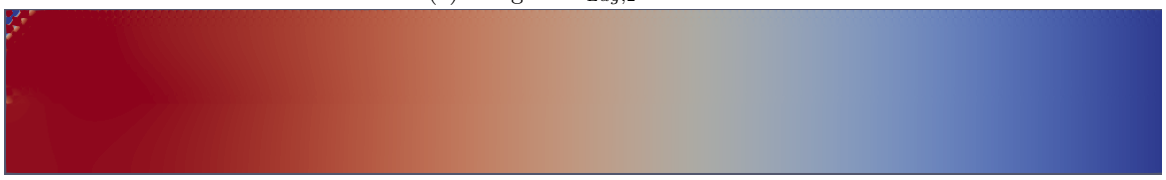
(a) using $NSF_{Lag,2}$ elements



(b) using $NSF_{RT,2}$ elements

Figure 6.48: NSF: magnitude of the velocity using $NSD_{Lag,2}$ and $NSD_{RT,2}$ elements (150×21)

(a) using $NSF_{Lag,2}$ elements



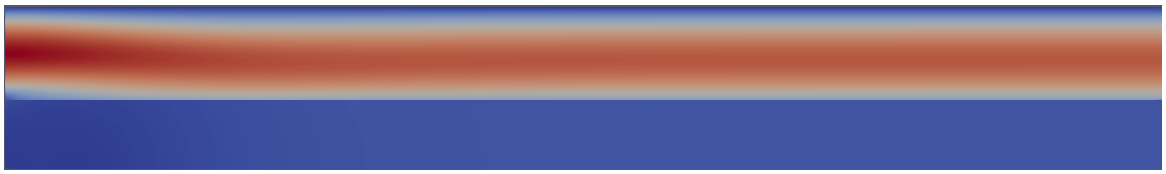
(b) using $NSF_{RT,2}$ elements

Figure 6.49: NSF: magnitude of the velocity using $NSD_{Lag,2}$ and $NSD_{RT,2}$ elements (150×21)

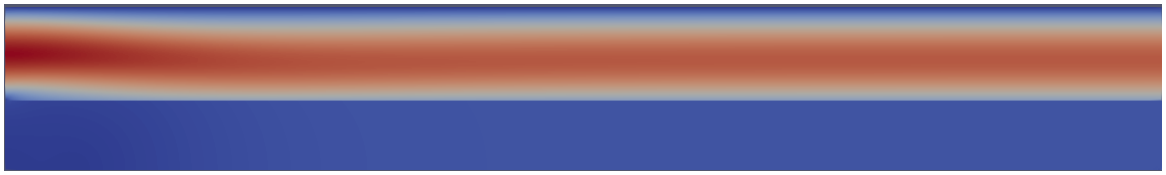
The results for order 3 are given in Table 6.17, Figure 6.50, and Figure 6.51. They are very similar to before.

	$NSF_{Lag,3}$			$NSF_{RT,3}$		
gridsize	#iterations	divergence		#iterations	divergence	
		Ω_f	Ω_p		Ω_f	Ω_p
50×7	5	0.087322	0.034826	6	1.836999e-11	0.288362
100×14	5	0.042370	0.0155026	6	3.715196e-11	0.288362
150×21	5	0.032552	0.005954	5	5.603433e-11	0.287255
200×28	5	0.030197	0.004897	5	7.453326e-11	0.286725

Table 6.17: Divergence L^2-norm and iteration number for different gridsizes, for the NSF case using $NSF_{Lag,3}$ and $NSF_{RT,3}$ elements



(a) using $NSF_{Lag,3}$ elements



(b) using $NSF_{RT,3}$ elements

Figure 6.50: NSF: magnitude of the velocity using $NSD_{Lag,3}$ and $NSD_{RT,3}$ elements (150×21)

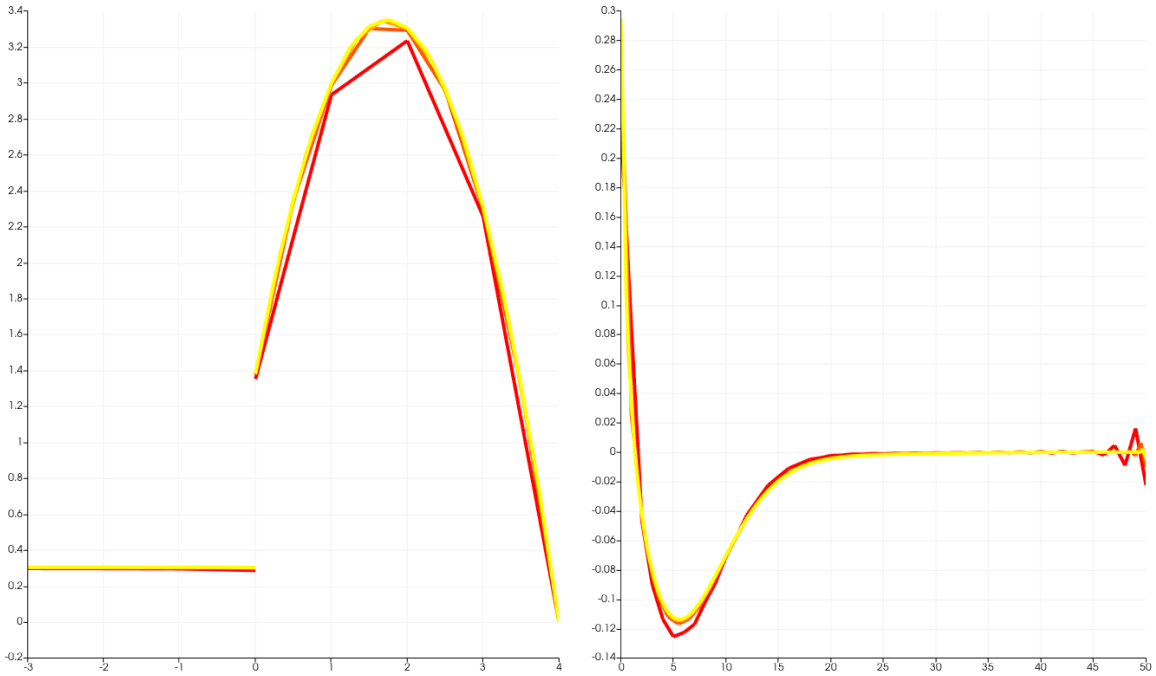
(a) using $NSF_{Lag,3}$ elements

(b) using $NSF_{RT,3}$ elements

Figure 6.51: NSF: magnitude of the velocity using $NSD_{Lag,3}$ and $NSD_{RT,3}$ elements (150×21)

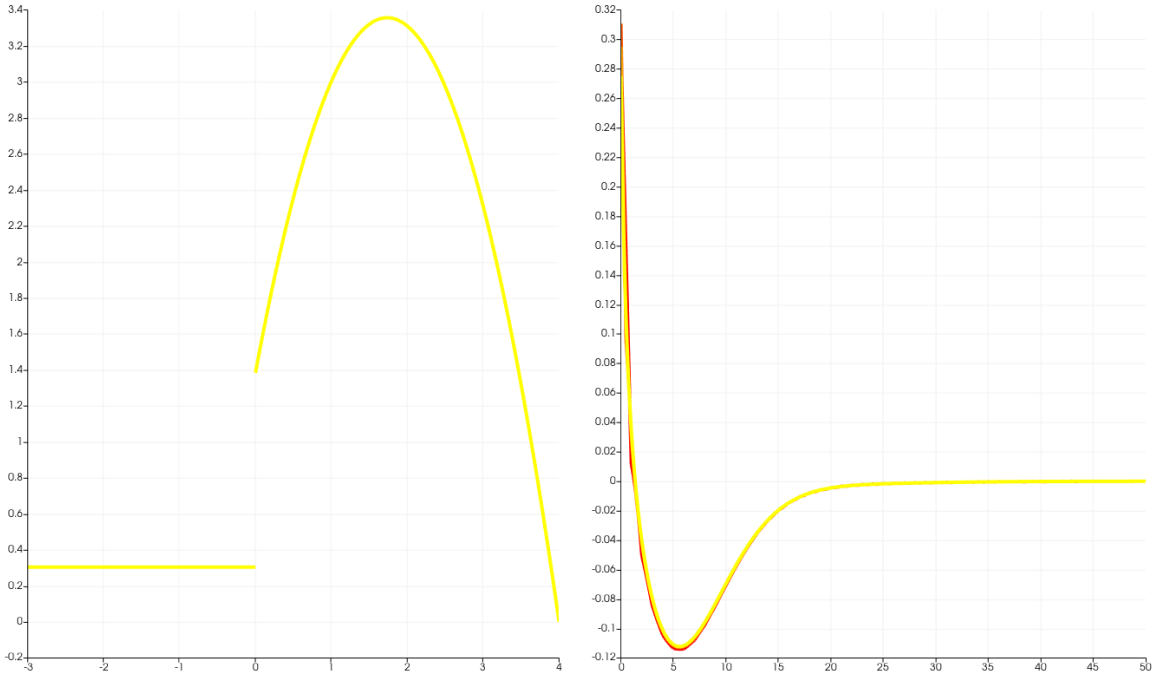
6.4.2 Velocity profiles

The plots for the outlet and interface velocity profiles per element type are given in Figure 6.52, 6.53, 6.54, and 6.55. They are very similar to NSD.



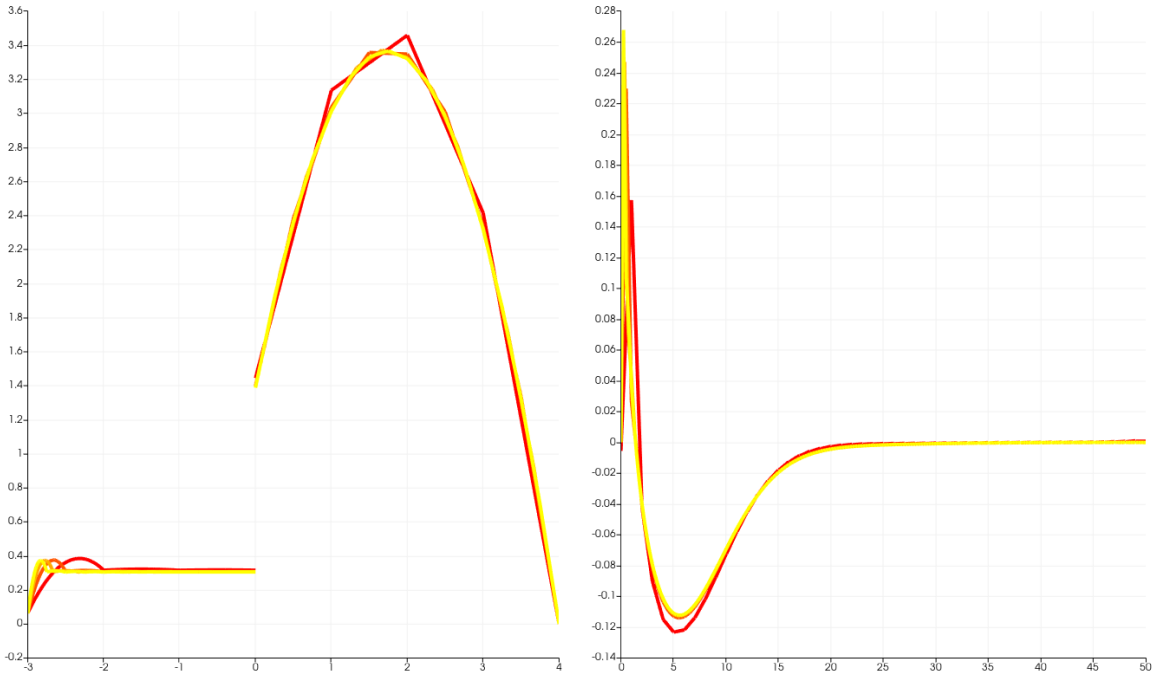
(a) u_x at the outlet (b) u_y at the interface

Figure 6.52: NSF: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSF_{Lag,1}$ elements



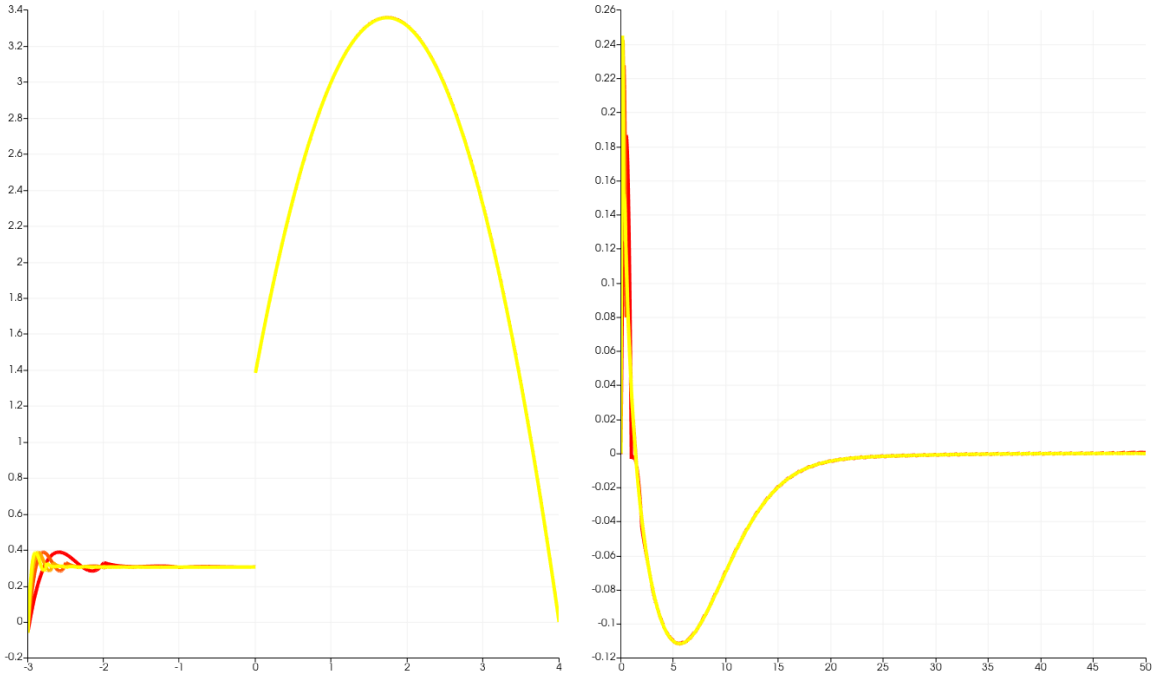
(a) u_x at the outlet (b) u_y at the interface

Figure 6.53: NSF: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSF_{Lag,2}$ elements



(a) u_x at the outlet (b) u_y at the interface

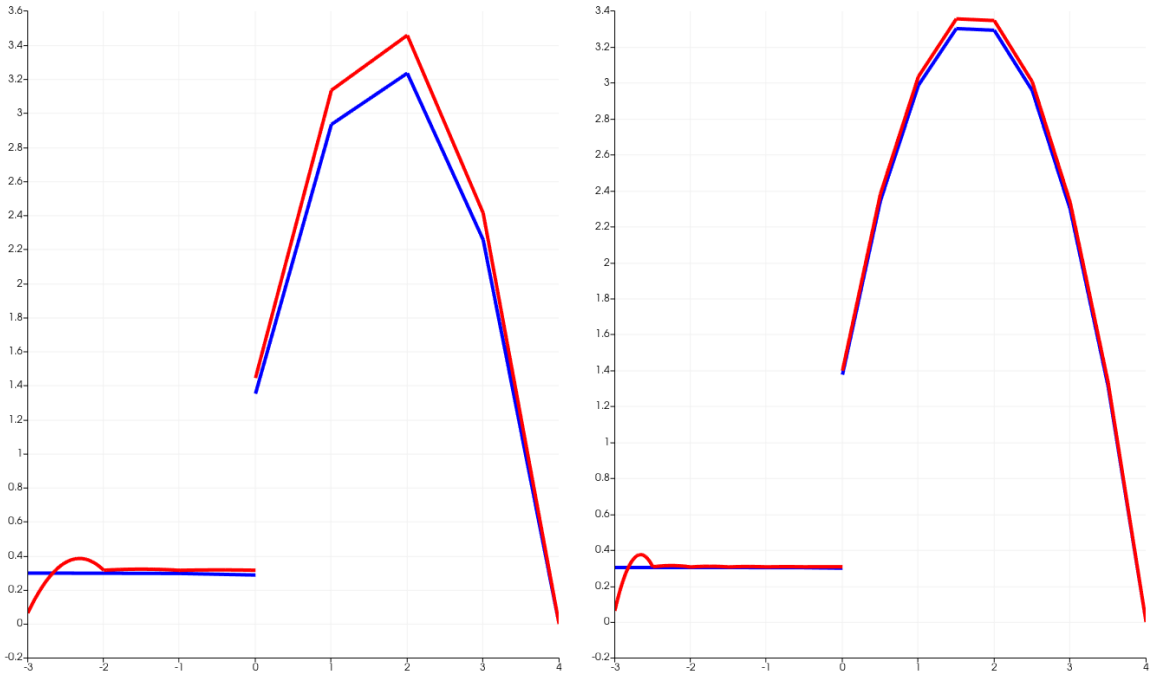
Figure 6.54: NSF: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSF_{RT,2}$ elements



(a) u_x at the outlet (b) u_y at the interface

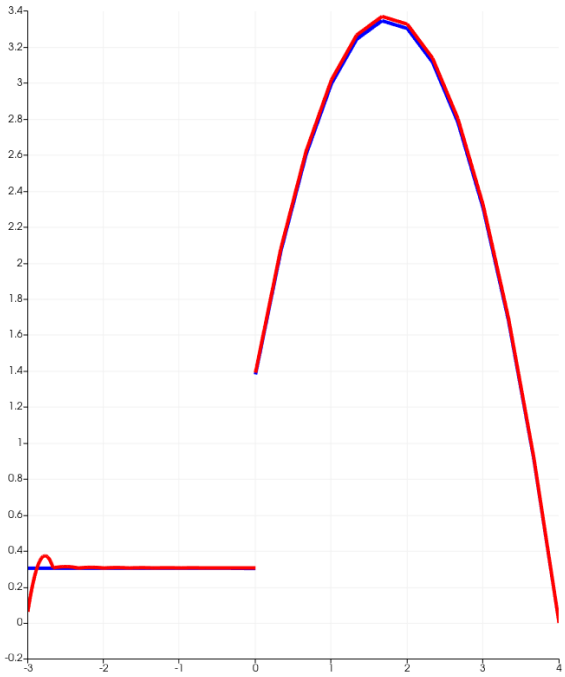
Figure 6.55: NSF: outlet and interface velocities for various gridsizes — 50×7 (red), 100×14, 150×21, 200×28 (yellow) —using $NSF_{RT,3}$ elements

The comparison plots of $NSF_{RT,r}$ versus $NSF_{Lag,r-1}$ are given in Figure 6.56, 6.57, 6.58, and 6.59. Again, they are very similar to NSD.

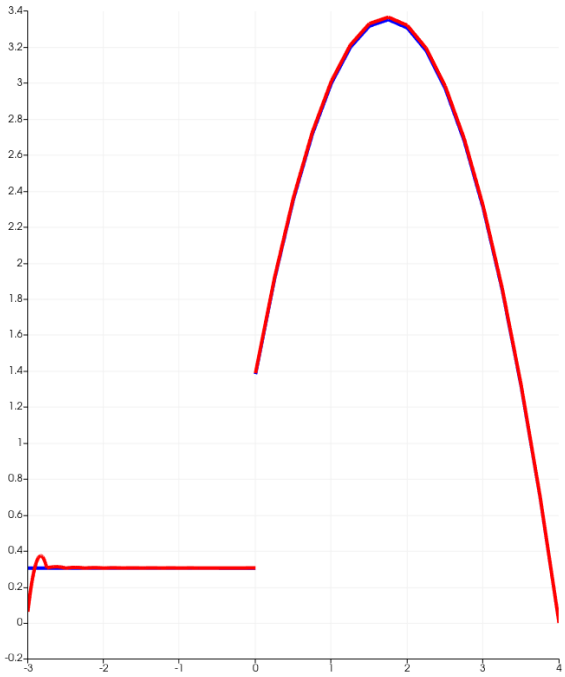


(a) 50×7

(b) 100×14

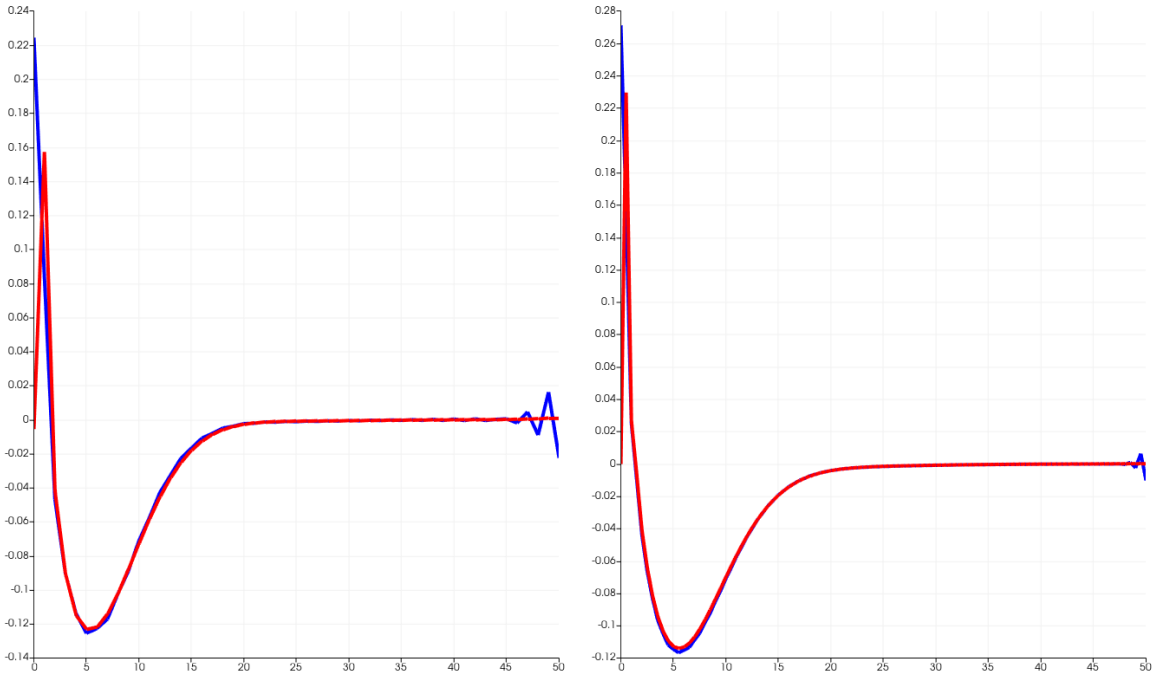


(c) 150×21



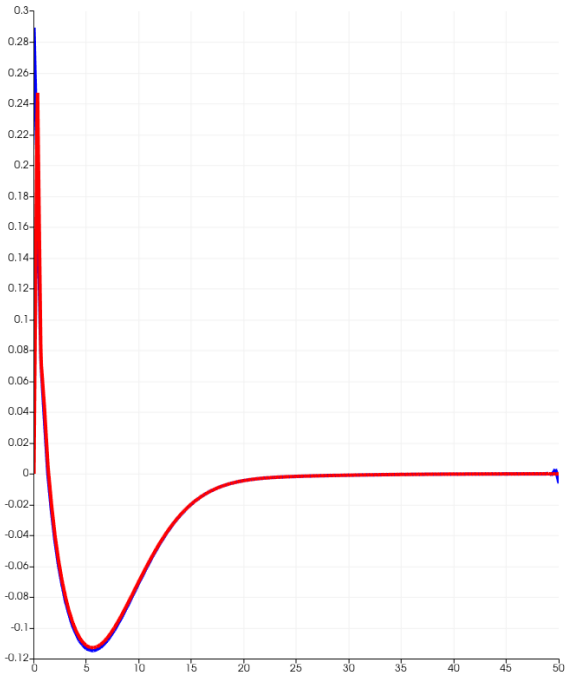
(d) 200×28

Figure 6.56: NSF: outlet velocity for various gridsizes, comparing $NSF_{RT,2}$ (red) with $NSF_{Lag,1}$ (blue) elements

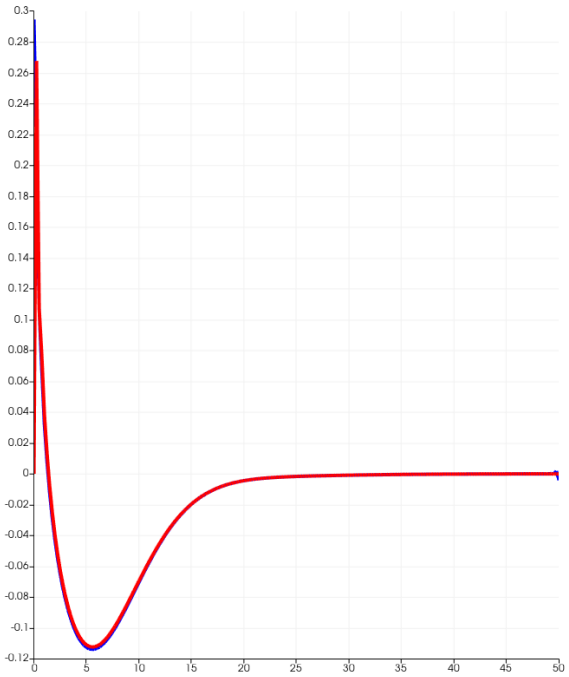


(a) 50×7

(b) 100×14

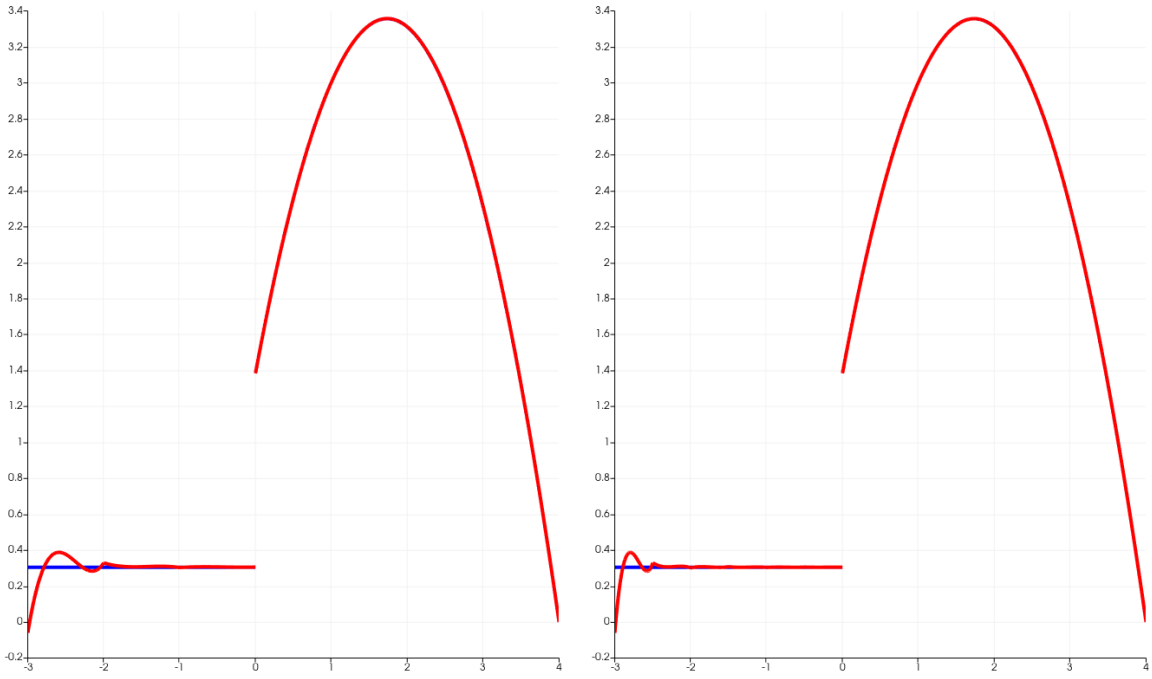


(c) 150×21



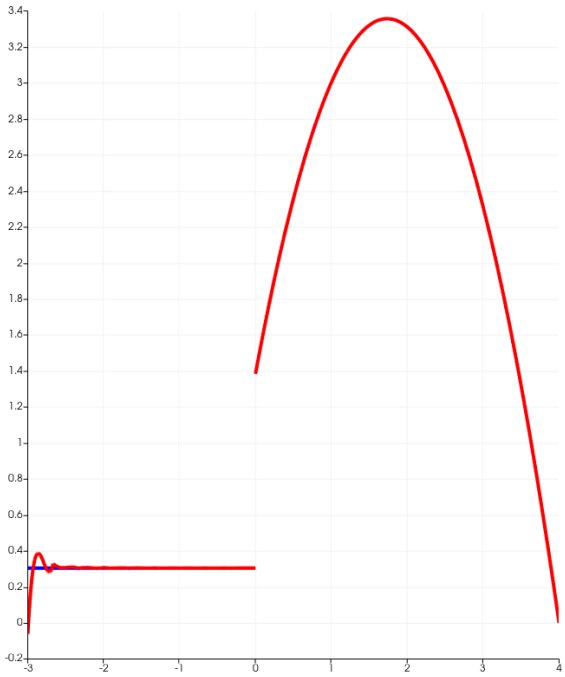
(d) 200×28

Figure 6.57: NSF: interface velocity for various gridsizes, comparing $NSF_{RT,2}$ (red) with $NSF_{Lag,1}$ (blue) elements

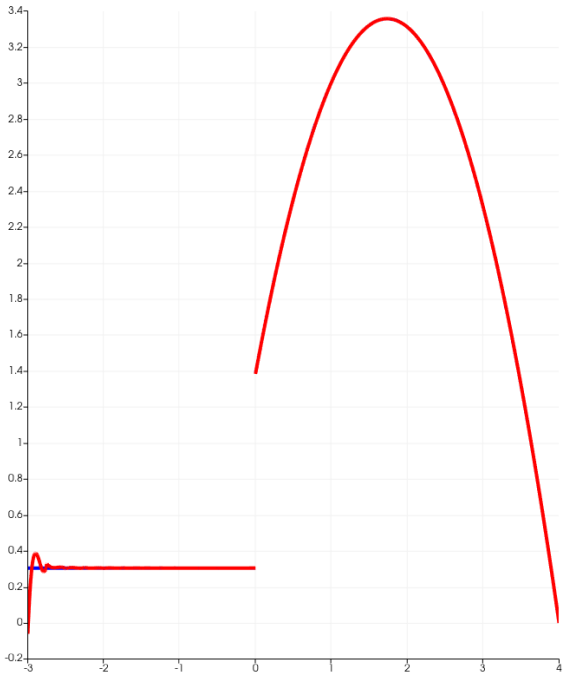


(a) 50×7

(b) 100×14

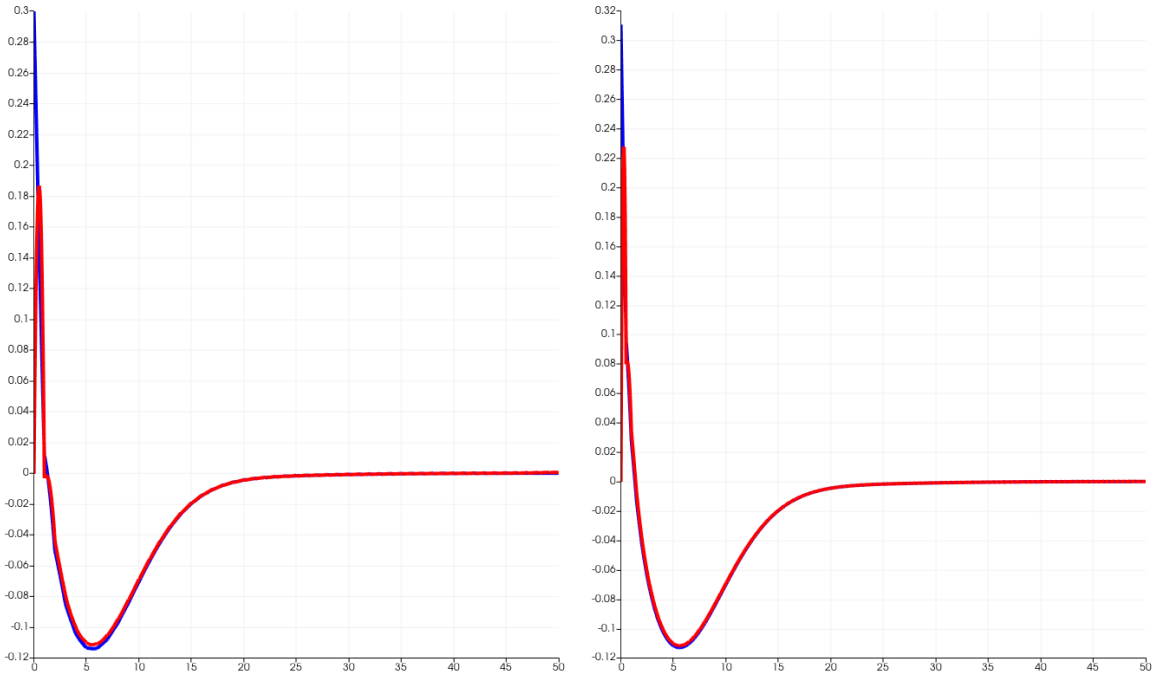


(c) 150×21



(d) 200×28

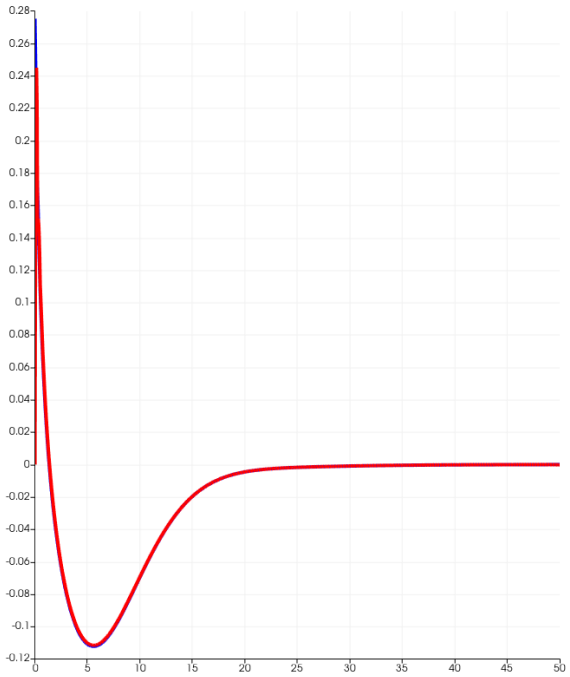
Figure 6.58: NSF: outlet velocity for various gridsizes, comparing $NSF_{RT,3}$ (red) with $NSF_{Lag,2}$ (blue) elements



(a) 50×7

(b) 100×14

(c) 150×21

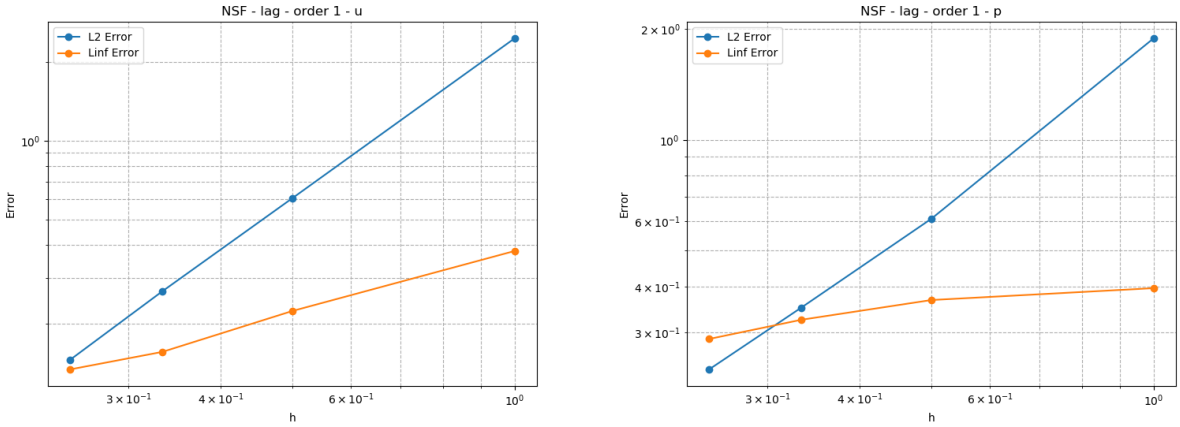


(d) 200×28

Figure 6.59: NSF: interface velocity for various gridsizes, comparing $NSF_{RT,3}$ (red) with $NSF_{Lag,2}$ (blue) elements

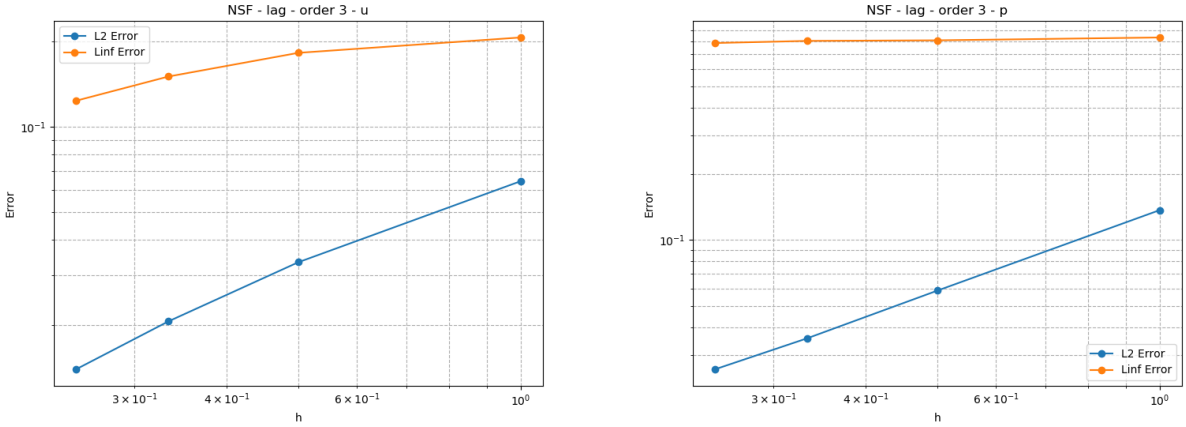
6.4.3 Error analysis

The error plots and slopes are in Figure 6.60, Figure 6.61, Table 6.18, and Table 6.19. Again, the reference solution for order 2 did not converge.



(a) u (b) p

Figure 6.60: $NSF_{lag,1}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u (b) p

Figure 6.61: $NSF_{Lag,3}$ - L^2 and L^∞ errors in u and p for various gridsizes

	u			p		
segment	$NSF_{lag,1}$	$NSF_{lag,2}$	$NSF_{lag,3}$	$NSF_{lag,1}$	$NSF_{lag,2}$	$NSF_{lag,3}$
1	2.025	-	0.951	1.627	-	1.218
2	2.016	-	1.188	1.269	-	1.232
3	2.092	-	1.354	1.343	-	1.131

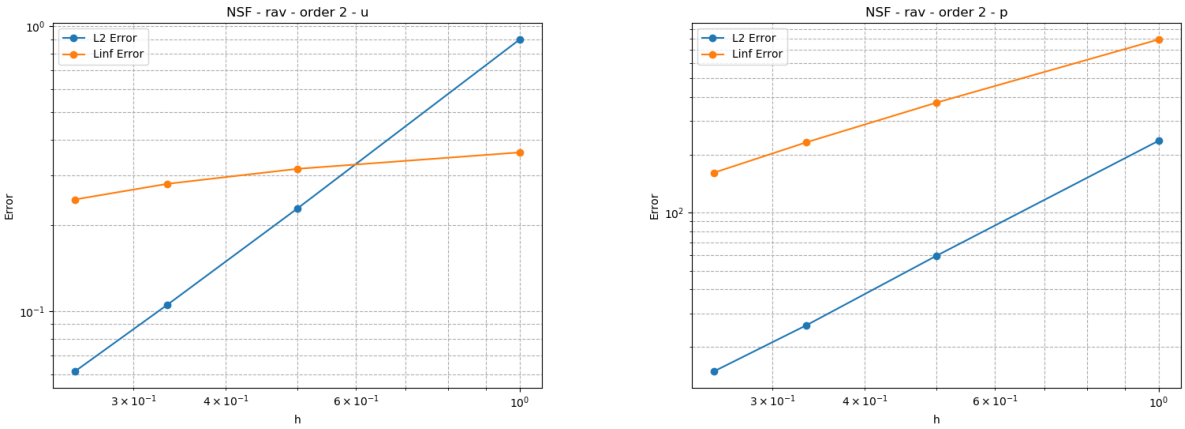
Table 6.18: NSF, Lagrange - Estimated rates of convergence of the L^2 error

	u			p		
segment	$NSF_{lag,1}$	$NSF_{lag,2}$	$NSF_{lag,3}$	$NSF_{lag,1}$	$NSF_{lag,2}$	$NSF_{lag,3}$
1	0.760	-	0.180	0.107	-	0.043
2	0.886	-	0.476	0.304	-	0.014
3	0.535	-	0.684	0.417	-	0.079

Table 6.19: NSF, Lagrange - Estimated rates of convergence of the L^∞ error

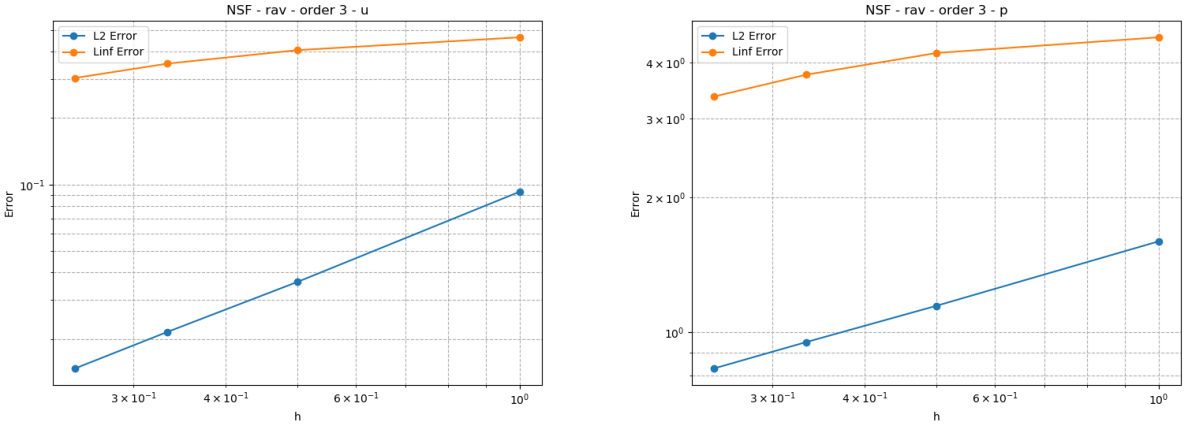
For RT, the first order reference solution did not converge. The other results are given in Figure 6.62, Figure 6.63,

Table 6.20, and Table 6.21. The results are very similar to NSD.



(a) u (b) p

Figure 6.62: $NSF_{RT,2}$ - L^2 and L^∞ errors in u and p for various gridsizes



(a) u (b) p

Figure 6.63: $NSF_{RT,3}$ - L^2 and L^∞ errors in u and p for various gridsizes

	u			p		
segment	$NSF_{RT,1}$	$NSF_{RT,2}$	$NSF_{RT,3}$	$NSF_{RT,1}$	$NSF_{RT,2}$	$NSF_{RT,3}$
1	-	1.971	1.355	-	1.988	0.478
2	-	1.918	1.284	-	2.051	0.459
3	-	1.869	1.320	-	1.914	0.471

Table 6.20: NSF, Raviart-Thomas - Estimated rates of convergence of the L^2 error

	u			p		
segment	$NSF_{RT,1}$	$NSF_{RT,2}$	$NSF_{RT,3}$	$NSF_{RT,1}$	$NSF_{RT,2}$	$NSF_{RT,3}$
1	-	0.191	0.193	-	1.092	0.116
2	-	0.299	0.347	-	1.171	0.275
3	-	0.439	0.518	-	1.261	0.390

Table 6.21: NSF, Raviart-Thomas - Estimated rates of convergence of the L^∞ error

7

Conclusion

In this thesis report, we have discussed the use of physics-compatible finite elements for modelling fluid flow through porous media. We discussed the governing equations of (Navier-)Stokes, Darcy, and Forchheimer, and the different ways to combine them into a single model. We also gave a quick rundown of the Finite Element Method, and the theory behind the $H(\text{div})$-compatible Raviart-Thomas element. After this, we took to implementing a model, based on a paper, which compared the use of Raviart-Thomas elements with Lagrange elements in the one-domain and two-domain approaches.

In our literature review, we identified a number of research gaps. We found research on this topic involving physics-compatible elements was mostly limited to the two-domain approaches. We found no literature comparing the use of physics-compatible elements with standard elements in the context of both the one-domain and two-domain approaches. We also found that the literature often omitted the transient term, and little research was done on the influence of the Reynolds number. These latter two points have not been addressed in this report, but will be discussed in the Discussion section.

In order to fill the first research gap, we made our own implementation incorporating both Lagrange and Raviart-Thomas elements. We based our implementation on a paper that already compared the one-domain and two-domain approaches, so that we had a good reference point. The paper called their one-domain approach PE, and had two different two-domain approaches: NSD and NSF. Unfortunately, when trying to replicate the paper using Lagrange elements, we were not able to make the one-domain approach overlap exactly with the paper's results. However, we still had a good reference point with which we could compare the use of Raviart-Thomas elements. We were not able to create a completely flawless model with the Raviart-Thomas elements. In all approaches, we found a big pressure spike in the upper left corner. There was also some disturbance in the origin. Furthermore, we were not able to perform an accurate error analysis, for we did not have an exact reference solution. We attempted to create our own by running our model on as fine a grid as possible, but this turned out to not be good enough. Also, because of the previous problems, the reference solution also had inaccuracies in the case of Raviart-Thomas.

Besides all this negativity, we were able to implement Raviart-Thomas elements accurately, in the sense that they provided an exactly divergence-free solution, as expected from the theory. Also, these problems we discussed are present in a few isolated points on the boundary, suggesting that the issue lies in the implementation of the boundary conditions. Indeed, the Raviart-Thomas elements involved a more elaborate implementation in this regard, as we had to enforce the tangential boundary conditions weakly through Nitsche's method, so a bug could have easily found its way in. Therefore, while the implementation involves some issues, it should be a good starting point for another to pick up the research.

Another comparison that we can draw a conclusion on is the difference in runtime and memory usage regarding Lagrange and Raviart-Thomas elements. In general, our implementation of Raviart-Thomas took longer to compile and run than Lagrange. While the number of iterations was similar for both element types, the time that each individual iteration took was a lot longer. This is probably because the Raviart-Thomas weak forms involved a lot more terms because of the penalisations. Also, the number of degrees of freedom increases faster with the order for Raviart-Thomas than for Lagrange. We do believe that this is an acceptable price to pay for the exactly divergence-free solution that you get.

8

Discussion

As with most research, the answers you get lead to more questions. This research is no exception to that idea. In this section, we give some suggestions for future research topics building upon this report.

Firstly, as discussed in the conclusion, there seem to be a few faults in our codes which would need to be investigated. We found in particular that our results for the PE case did not match the paper's. We did not find any discrepancy in our weak form or space definitions, so something else is going wrong. Another point of attention is the pressure spike in the upper left corner of the domain which occurs when using Raviart-Thomas elements, as well as the disturbance in the lower right corner.

Secondly, in order to have a more reliable error analysis, we suggest that future research makes use of manufactured solutions, so that an exact solution is known. Some examples of papers using manufactured solutions in this research topic are [25] and [26].

Next, we focus on the points from the literature review that we were not able to address in this report. Firstly, we did not consider the transient term of the weak form. While the paper we compared to did include it, it did not really investigate its influence either, as the authors only looked at the solution once it had reached steady-state. Though it is not expected that this term has much influence, it could be an avenue worth exploring. Furthermore, the paper embarked on an elaborate parameter analysis, something that we also did not address. We are particularly interested in the influence of the Reynolds number, as that more or less determines the influence of the Forchheimer term, and thus highlights the difference between the NSD and NSF approaches. It would be interesting to see if physics-compatible elements react differently to these parameter changes.

This brings us to our next and final point. We only talked about Raviart-Thomas elements in this report, but there are many more physics-compatible elements out there. Another well-known $H(\mathrm{div})$-conforming element type is the Brezzi-Douglas-Marini (BDM) element. The website DefElement [20] lists some more. Each element has its usage cases, so it would be interesting to investigate the applicability of these elements in the context of this research topic. We note that not all these element types are available in the Julia package Gridap that we have utilised, so for this investigation, other implementation methods would have to be employed.

— Appendix —

- A

Useful Theorems

Cauchy-Schwarz inequality

Given the inner product space $(V, (\cdot, \cdot))$, we have for all $u, v \in V$:

$$|(u, v)| \leq (u, u)^{1/2}(v, v)^{1/2} = \|u\|_V \|v\|_V \tag{A.1}$$

Poincaré inequality

Given a bounded domain Ω and $1 \leq p < \infty$, there exists a constant C, such that for all $u \in W_0^{1,p}(\Omega)$:

$$\|u\|_{L^2} \leq C\|\nabla u\|_{L^2}. \tag{A.2}$$

Céa's lemma

Given V, a closed subspace of a Hilbert space, and $a(\cdot, \cdot)$, a continuous and coercive bilinear form on V, we state the following variational form:

Find $u \in V$ such that

$$a(u, v) = F(v), \quad \forall v \in V,$$

with $F \in V'$. Then, given $V_h \subseteq V$ and $F \in V'$, the approximation problem is given by

Find $u_h \in V_h$ such that

$$a(u_h, v) = F(v), \quad \forall v \in V_h.$$

Then we have that

$$\|u - u_h\|_V \leq \frac{C}{\alpha} \min_{v \in V_h} \|u - v\|_V, \tag{A.3}$$

with C the continuity constant and α the coercivity constant.

References

[1] Michel Quintard and Stephen Whitaker. Coupled, nonlinear mass transfer and heterogeneous reaction in porous media. *Handbook of Porous Media*, page 1, 2005.

[2] Henry Darcy. *Les fontaines publiques de la ville de Dijon: exposition et application des principes à suivre et des formules à employer dans les questions de distribution d'eau*, volume 1. Victor dalmont, Paris, 1856.

[3] Roberto Mauri. *Transport phenomena in multiphase flows*. Springer, 2015.

[4] M. Muskat. *The flow of homogeneous fluids through porous media*. J. W. Edwards, Inc., Ann Arbor, Michigan, 1st edition, 1946.

[5] Chin-Tsau Hsu. Dynamic modeling of convective heat transfer in porous media. *Handbook of porous media*, 2:39–80, 2005.

[6] Josef Kozeny. Über kapillare Leitung des Wassers im Boden (Aufstieg, Versickerung und Anwedung auf die Bewässerung). *Sitzungsberichte der Akademie der Wissenschaften mathematisch-naturwissenschaftliche Klasse*, 136(2a):271–306, 1927.

[7] P. C. Carman. Fluid flow through granular beds. *Transactions, Institution of Chemical Engineers*, 15:150–166, 1937.

[8] P. C. Carman. *Flow of gases through porous media*. Butterworths, London, 1956.

[9] P. Forchheimer. Wasserbewegung durch Boden. *Z. Ver. Deutsch. Ing.*, 45:1782–1788, 1901.

[10] Hendrik C. Brinkman. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Flow, Turbulence and Combustion*, 1(1):27–34, 1949.

[11] Gordon S. Beavers and Daniel D. Joseph. Boundary conditions at a naturally permeable wall. *Journal of fluid mechanics*, 30(1):197–207, 1967.

[12] Philip Geoffrey Saffman. On the boundary condition at the surface of a porous medium. *Studies in applied mathematics*, 50(2):93–101, 1971.

[13] B. Alazmi and K. Vafai. Analysis of fluid flow and heat transfer interfacial conditions between a porous medium and a fluid layer. *International Journal of Heat and Mass Transfer*, 44(9):1735–1749, 2001.

[14] Graham Neale and Walter Nader. Practical significance of Brinkman's extension of Darcy's law: Coupled parallel flows within a channel and a bounding porous medium. *The Canadian Journal of Chemical Engineering*, 52(4):475–478, 1974.

[15] J. Alberto Ochoa-Tapia and Stephen Whitaker. Momentum transfer at the boundary between a porous medium and a homogeneous fluid—I. theoretical development. *International Journal of Heat and Mass Transfer*, 38(14):2635–2646, 1995.

[16] J. Alberto Ochoa-Tapia and Stephen Whitaker. Momentum transfer at the boundary between a porous medium and a homogeneous fluid—II. comparison with experiment. *International Journal of Heat and Mass Transfer*, 38(14):2647–2655, 1995.

[17] J. Alberto Ochoa-Tapia and Stephen Whitaker. Momentum jump condition at the boundary between a porous medium and a homogeneous fluid: inertial effects. *J. Porous Media*, 1(3):201–217, 1998.

[18] Huangxin Chen and Xiao-Ping Wang. A one-domain approach for modeling and simulation of free fluid over a porous medium. *Journal of Computational Physics*, 259:650–671, 2014.

[19] Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.

[20] Matthew W. Scroggs, Pablo D. Brubeck, Joseph P. Dean, Jørgen S. Dokken, India Marsden, et al. DefElement: an encyclopedia of finite element definitions. `https://defelement.com`, 2024. [Online; accessed 23-October-2024].

[21] Susanne C. Brenner and L. Ridgway Scott. *The mathematical theory of finite element methods*. Springer, 2008.

[22] F. Cimolin and M. Discacciati. Navier–Stokes/Forchheimer models for filtration through porous media. *Applied Numerical Mathematics*, 72:205–224, 2013.

[23] Joseph Benzaken, John A Evans, and Rasmus Tamstorf. Constructing Nitsche's method for variational problems. *Archives of Computational Methods in Engineering*, 31(4):1867–1896, 2024.

[24] Junping Wang, Yanqiu Wang, and Xiu Ye. A robust numerical method for stokes equations based on divergence-free H (div) finite element methods. *SIAM Journal on Scientific Computing*, 31(4):2784–2802, 2009.

[25] Sergio Caucao and Marco Discacciati. A mixed FEM for the coupled Brinkman–Forchheimer/Darcy problem. *Applied Numerical Mathematics*, 190:138–154, 2023.

[26] Pablo GS Carvalho, Philippe RB Devloo, and Sônia M Gomes. On the use of divergence balanced H (div)-L2 pair of approximation spaces for divergence-free and robust simulations of Stokes, coupled Stokes–Darcy and Brinkman problems. *Mathematics and computers in simulation*, 170:51–78, 2020.

[27] MACIEJ Sniechowski, JAKUB Kaminski, S Wronski, et al. Heterogeneous materials based on aperiodic structures for bone tissue substitutes. In *VI International Conference on Computational Bioengineering*, 2015.

Front page image taken from [27]