

# Chapter 4

## DISCRETE FLUID FLOW EQUATIONS

### 4.1 Introduction

We began the process of analysing our problem by discretizing the flow region and the boundary conditions, that is, making them describable in terms of the values of a few parameters. But while we have discretized our problem, the *answers* to that problem, that is, the solutions of the continuous Navier Stokes equations, discussed in Chapter 3, are still general functions  $u(x, y)$ ,  $v(x, y)$  and  $p(x, y)$ . Except for very special cases, these answers are essentially only describable by giving their values at every point in the flow region, and this is not something that is computable.

We now consider how we discretize our flow equations and their solutions, so that an answer is describable in terms of the values of a finite number of coefficients, and these coefficients can be determined by solving a finite number of equations. Of course, we need to do this in such a way that the answers to the discretized equations can be expected to approximate the answers to the continuous equations.

The closeness of the approximation is dependent on a *discretization parameter*  $h$ . We con-

sider constraints on the choice of a class of approximating finite element spaces which will guarantee that as  $h \rightarrow 0$ , the approximate solutions converge to the continuous solution. We then choose the approximating spaces associated with the *Taylor Hood* element.

We then consider the actual form of the finite element equations that replace the original continuous state equations. These will compose a nonlinear algebraic system of considerable dimension. We solve these nonlinear equations using Newton's method, or a hybrid approach involving Picard iteration followed by Newton's method.

Finally, we quote results about the approximation error between the solutions of the continuous and discrete problems. These results assure us that, for small enough  $h$ , the results of the discrete problem give good estimates of the behavior of the continuous problem.

## 4.2 Derivation of the Finite Element Equations

In general, there are no methods of producing the exact solution functions  $u$ ,  $v$  and  $p$  for the continuous Navier Stokes problem. The Poiseuille flow problem was a very special case that allowed us to get an exact solution. However, even that simple problem would become insoluble if the form of the inflow were changed, the walls were made irregular in any way, or a small obstacle were placed in the flow.

We want to look at a wide number of Navier Stokes problems, and we need a way of solving them all. Since we can't solve the continuous problem, we now consider the construction of a discrete version of the continuous problem, using the finite element method. We expect that the solution of the discrete problem will approximate the solution of the continuous problem. Given sufficient computational resources, this approximation can be improved as much as desired. The discrete problem can be solved in a straightforward way for the variety of region shapes, boundary conditions, and Reynolds numbers we are interested in.

### 4.3 Representation of the Solution

We begin the derivation of the finite element equations by making an assumption about the form of the solution functions. For clarity, where a solution to the original, continuous problem is symbolized as  $(u, v, p)$ , we will symbolize a solution to the approximating problem as  $(u^h, v^h, p^h)$ . The solution functions for the original continuous problem might have a very elaborate form, but for the approximate problem, we will immediately assume that solutions are constructed using simple linear combinations of a finite set of basis functions to be selected beforehand.

Since  $u^h$  and  $v^h$  are really coordinate projections of a two-dimensional velocity vector, there are good physical reasons to use the same set of basis functions for both. We represent a typical velocity basis function as  $w_i(x, y)$ . For technical reasons the pressure cannot be represented with the same set of basis functions, but must use a different set, a typical element of which we will write as  $q_i(x, y)$ . The pressure basis functions must be chosen in a way that is compatible with the velocity basis functions. We will not explicitly choose the basis functions yet. If we assume that there are  $N_w$  velocity basis functions, and  $N_q$  pressure basis functions, we may now write out the *finite element representations* of the discretized solutions:

$$u^h(x, y) = \sum_{i=1}^{N_w} u_i^h w_i(x, y) \quad (4.1)$$

$$v^h(x, y) = \sum_{i=1}^{N_w} v_i^h w_i(x, y) \quad (4.2)$$

$$p^h(x, y) = \sum_{i=1}^{N_q} p_i^h q_i(x, y) \quad (4.3)$$

The symbol  $u^h$  is used here in two ways: as the name of the solution function,  $u^h(x, y)$ , and as the name of an entry  $u_i^h$  of the coefficient vector. No confusion should arise from this ambiguity, if it is understood that the function and the coefficient vector represent the same

information.

The assumption that we can represent solutions of the discretized problem by a linear combination of basis functions makes the discretized problem much simpler than the continuous one. To define the solutions  $(u^h, v^h, p^h)$ , we simply have to determine the  $2N_w + N_q$  coefficients of the basis functions, rather than determining the value at every point  $(x, y)$  of three almost arbitrary functions  $(u, v, p)$ .

## 4.4 Discretizing the Geometry

We must now produce a set of appropriate basis functions for the problem. To do so, we will first produce a set of *nodes*, that is, special points in the region  $\Omega$  and on the boundary  $\Gamma$ . We will actually have two sets of nodes, *pressure nodes* and *velocity nodes*. However, for the method we will use, the pressure nodes will simply be a subset of the velocity nodes.

Much thought can go into the choice of the number and location of the nodes. We will make some very simple choices. We will march along the horizontal axis of the channel, taking equally spaced steps. At each step, including the inflow and outflow boundaries, we will place a column of nodes. The nodes in a particular column will be equally spaced in the vertical direction. We require the first node in a column to lie on the bottom wall or bump, and the last node to lie on the upper wall. Each column will have the same number of nodes, but if the column is above the bump, then the vertical spacing between the nodes will be changed from the spacing used in the rectangular portion of the channel. With an eye on future needs, we will require that there be an *odd* number of rows and columns in both directions. For convenience, we will also assume that there is some numbering of the nodes, which we will call the *global node ordering*.

The maximum spacing between nodes will have a certain effect on the accuracy of the

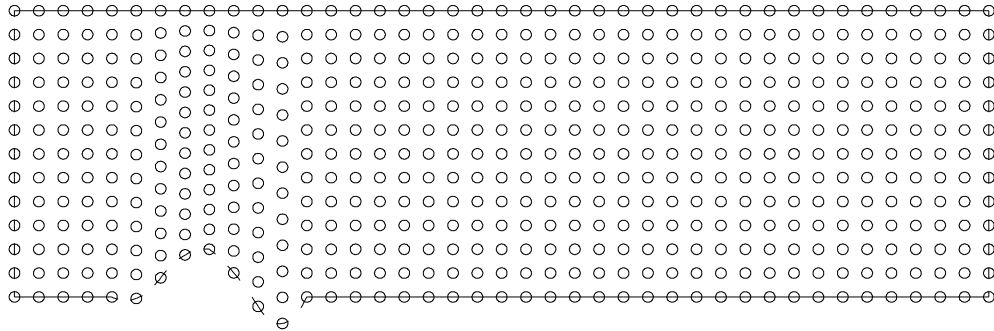


Figure 4.1: Decomposition of a flow region into nodes.

computational results. The value of the discretization parameter  $h$  can be taken to be the larger of the two spacings in the rectangular region of the channel, that is, the horizontal and vertical node spacings.

A typical decomposition into nodes is shown in Figure 4.1.

## 4.5 Choosing a Master Element

In order to generate the basis functions, we are now going to use the nodes to decompose the flow region into a set of separate patches of area, called *elements*. Each element will have the same general structure, and will be defined by listing the nodes that make it up.

There are many choices for elements, but for a particular physical problem, only some elements may be appropriate. Essentially because of the way the the Navier Stokes equations involve two types of state variables, the velocities and pressures, the finite element formulation results in a problem of *mixed type*. This in turn implies that the usual finite element convergence results will not apply unless the basis functions (or equivalently, the approximating spaces) for velocities and pressures are chosen in a suitable, compatible way. The main constraint on this choice is known as the *inf-sup* or *Ladyzhenskaya-Babuska-Brezzi*

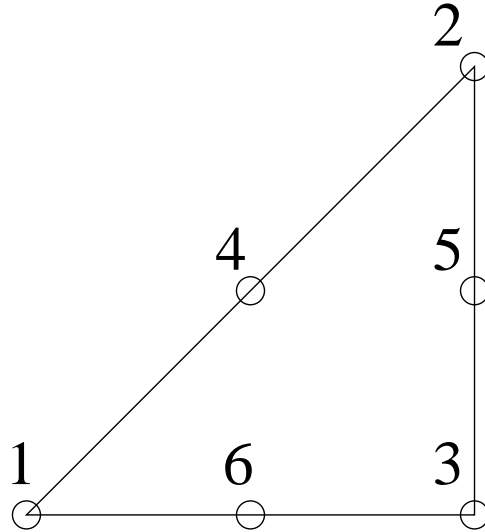


Figure 4.2: A Taylor Hood element.  
 All nodes have an associated velocity. Nodes 1, 2 and 3  
 have an associated pressure as well.

condition. Verification that a particular set of basis functions satisfy this condition is not trivial. We will use the *Taylor Hood* element, whose approximating spaces for velocities and pressure have been shown to satisfy the inf-sup condition. For details, refer to Girault and Raviart [13].

Geometrically, the Taylor Hood element  $TH$  has the shape of a triangle, outlined by three *corner nodes* and three *midside nodes*. We will use the special coordinate system  $(\xi, \eta)$  when referring to points in this element. The  $(\xi, \eta)$  coordinates of the corner nodes 1, 2 and 3 are then  $(0,0)$ ,  $(1,1)$  and  $(1,0)$ . All six nodes will be associated with velocity, but only the corner nodes will be associated with pressure.

Details about the construction of the associated basis functions will be discussed in the next section. For the moment, we confine ourselves to certain geometrical issues.

It's easy to see that we can completely cover the rectangular portion of the flow region with

copies of the Taylor Hood element, with no overlap. The area over the bump, however, might seem to present us with a problem. On the lower boundary of this area we have the curved surface of the bump. Above the bump, the rectilinear spacing of nodes is disrupted, so that the three nodes needed to form the horizontal side of a Taylor Hood element will not generally lie on a straight line.

We may deal with this more challenging geometry by using the standard technique of *isoparametric* mapping, which allows us to produce smoothly curved images of the Taylor Hood element that fit the local geometry. A simple mapping technique allows us to make the appropriate adjustments for computing basis functions and spatial derivatives on such an isoparametric element.

We are now prepared to produce a *triangulation*, that is, a decomposition of the original flow region into elements with disjoint interiors. We consider each element in the triangulation to be the image of the master element. If we call the  $i$ -th element in the triangulation  $\Omega_i$ , then we can actually construct a differentiable mapping

$$\phi_i : TH \rightarrow \Omega_i, \tag{4.4}$$

which maps any point  $(\xi, \eta) \in TH$  to a unique point  $(x, y) \in \Omega_i$ , and so that every point of  $\Omega_i$  is the image of some point  $(x, y)$ . Hence, there is also an inverse map:

$$\phi_i^{-1} : \Omega_i \rightarrow TH. \tag{4.5}$$

This mapping is completely determined by specifying the correspondence between the 6 nodes in the Taylor Hood element and the 6 nodes in the image. Each node in an element is the image of a node in  $TH$ ; the *local node index* of a node in  $\Omega_i$  is the label for the node's preimage in  $TH$ . A typical triangulation of a flow region, including isoparametric elements above the bump, is displayed in Figure 4.3.

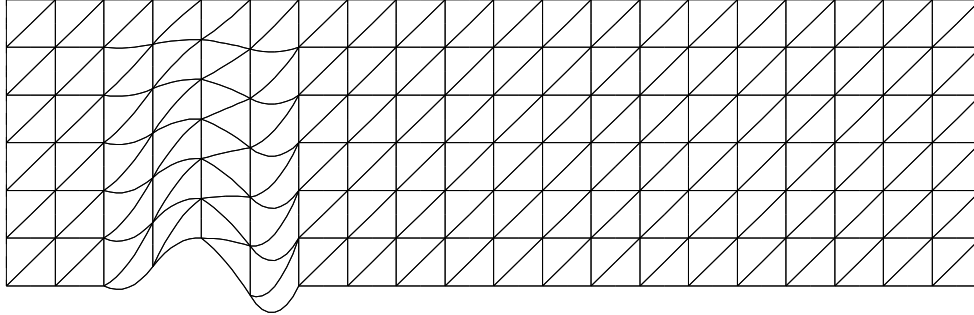


Figure 4.3: Decomposition of a flow region into elements.  
 These elements are built from the nodes shown in Figure 4.1.

The triangulation of the region in Figure 4.3 is built out of multiple copies of the Taylor Hood element, though some of them may have been curved or stretched a bit by an isoparametric mapping. Because we have decomposed the large, complicated geometry into multiple copies of a single, simple element, and because we know how these simple shapes build the flow region, we can proceed to construct the basis functions for the single master element, and our work will then naturally apply to all the images of that element.

## 4.6 Constructing Basis Functions

We are now prepared to construct the basis functions that will be used to represent the solution functions. The basis functions will be divided into two families, with  $w_i(x, y)$  used for velocities and  $q_i(x, y)$  for pressures. We will first consider the velocity basis functions.

Each velocity basis function will be associated with a particular velocity node. These nodes will have several special properties. First, at any velocity node, the associated velocity basis function will have the value 1, while all other velocity basis functions will be zero there. Secondly, if some function  $g^h(x, y)$  is a linear combination of the velocity basis functions, then we can determine the coefficient  $g_i^h$  associated with the velocity node  $i$  by evaluating

$g^h$  at that node. In other words, the finite element coefficients for  $g^h$  are its values at the associated nodes.

Let  $i$  be the global number of any velocity node, and  $w_i(x, y)$  the associated velocity basis function. At any point  $(x, y)$  outside of the flow region  $\Omega$ , we let  $w_i(x, y) = 0$ . Any point  $(x, y)$  within the flow region must lie in some element  $\Omega_j$ . If node  $i$  does not lie on the boundary of element  $\Omega_j$ , then again  $w_i(x, y)$  is zero. We are left with the case where node  $i$  lies on the the boundary of  $\Omega_j$ . In this case, we refer to the local node number  $l(i)$  of node  $i$  in  $\Omega_j$ . We then consider the unique quadratic polynomial  $\hat{w}_i(\xi, \eta)$  defined on the Taylor Hood element  $TH$ , which has the value 1 at node  $l(i)$  and 0 at the five other velocity nodes. We then define

$$w_i(x, y) = \hat{w}_i(\phi^{-1}(x, y)). \quad (4.6)$$

This slightly cumbersome definition allows us to consider both nonisoparametric and isoparametric cases together. The “preimage” of the basis function,  $\hat{w}_i(\xi, \eta)$ , is always a quadratic polynomial in  $\xi$  and  $\eta$ . For elements with straight sides, it is also true that  $w_i(x, y)$  is quadratic in  $x$  and  $y$ , basically because  $\phi^{-1}$  is linear for such elements. But in an isoparametric element, the basis function  $w_i(x, y)$  will not, in general have the simple form of a quadratic polynomial when written in terms of  $x$  and  $y$ .

It should be clear from the definition that  $w_i(x, y)$  is always 1 at node  $i$  and zero at all other velocity nodes, as desired. It should also be clear that the function  $w_i(x, y)$  is well defined, even when a point  $(x, y)$  lies on the boundary of two or more elements, and that  $w_i$  is a continuous function everywhere.

A little thought will reveal the fact that, if we choose an arbitrary vector of values  $u_i^h$ , then the function defined by

$$u^h(x, y) = \sum_{i=1}^{N_w} u_i^h w_i(x, y) \quad (4.7)$$

will have the value  $u_i^h$  at node  $i$ .

The pressure basis functions are defined similarly, keeping in mind that in any element there are only three pressure nodes. The major resulting difference for pressure nodes is that we consider the unique *linear* polynomial  $\hat{q}_i(\xi, \eta)$  defined on the Taylor Hood element  $TH$ , which has the value 1 at node  $l(i)$  and 0 at the two other pressure nodes.

In terms of basis functions, a shorthand (and for isoparametric elements, slightly inaccurate) way of defining the Taylor Hood element is, therefore, the use of piecewise quadratic velocities and piecewise linear pressures on a 6 node triangular element.

The use of the finite element basis functions has a number of advantages. In particular, although a discrete set of data represents the solution functions, the solution functions themselves are continuous functions, defined everywhere in the flow region. Moreover, it is obvious how to take any continuously defined function and construct the finite element representation for it, simply by sampling it at the appropriate nodes. (Other versions of the finite element method, such as the least squares approach, do not share this particular feature). Finally, operations such as differentiation of a solution function can be reduced to the same operations on the underlying basis functions, whose properties are easy to derive.

## 4.7 Discretizing the Navier Stokes Equations

We now have a way of representing functions over the flow region. An arbitrary set of values assigned to the nodes in the proper way will produce a set of functions  $(u, v, p)$ . However, we don't want an arbitrary set of functions; we want functions that come close to the solution functions for the continuous Navier Stokes equations. How can we do this?

The accepted finite element technique poses the equivalent question of how we determine each of the unknown coefficients. To do so, we go back to the three equations that define the

continuous Navier Stokes problem. In particular, let us start with the horizontal momentum equation. For each unknown coefficient  $u_i^h$  corresponding to a horizontal velocity, we will multiply the horizontal momentum equation by the corresponding basis function  $w_i(x, y)$ , and integrate over the region  $\Omega$ , to arrive at the following equation:

$$\int_{\Omega} \left( -\frac{\partial^2 u^h}{\partial x^2} - \frac{\partial^2 u^h}{\partial y^2} + Re \left( u^h \frac{\partial u^h}{\partial x} + v^h \frac{\partial u^h}{\partial y} + \frac{\partial p^h}{\partial x} \right) \right) w_i \, d\Omega = 0. \quad (4.8)$$

For technical reasons, including the desire to allow the use of basis functions of lower differentiability, we will use integration by parts to lower the order of differentiation on certain terms involving the solution variables. To do so, we apply a version of *Green's theorem*:

**Theorem 4.1 (Green's Theorem in the Plane)** *Suppose  $\Omega$  is a bounded, open, connected subset of  $R^2$  with a piecewise differentiable boundary  $\Gamma$ . Suppose that  $u$  and  $w$  are functions defined on  $\bar{\Omega}$ , the closure of  $\Omega$ , and continuously differentiable there. Then*

$$\int_{\Omega} \frac{\partial u(x, y)}{\partial x} w(x, y) \, dx \, dy = - \int_{\Omega} u(x, y) \frac{\partial w(x, y)}{\partial x} \, dx \, dy + \int_{\Gamma} u(x, y) w(x, y) \hat{n}_x(x, y) \, ds \quad (4.9)$$

where  $\hat{n}_x(x, y)$  is the  $x$  component of the unit outward normal vector to  $\Gamma$ . A similar formula holds for differentiation with respect to  $y$ .

For the horizontal momentum equation, we wish to apply Theorem 4.1 to the terms  $-\frac{\partial^2 u^h}{\partial x^2} w_i$  and  $-\frac{\partial^2 u^h}{\partial y^2} w_i$ , reducing the highest order of differentiation applied to a velocity basis function from 2 to 1. We could also apply Green's theorem to the term  $Re \frac{\partial p^h}{\partial x} w_i$ , eliminating all differentiation of pressures, at the cost of some unwieldy boundary terms.

If we carry out this same set of operations on the other two governing equations, we arrive at the *discretized finite element equations*:

$$\int_{\Omega} \left( \frac{\partial u^h}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial u^h}{\partial y} \frac{\partial w_i}{\partial y} + Re \left( u^h \frac{\partial u^h}{\partial x} + v^h \frac{\partial u^h}{\partial y} + \frac{\partial p^h}{\partial x} \right) w_i \right) \, d\Omega$$

$$\begin{aligned}
&= \int_{\Gamma} \frac{\partial u^h}{\partial n} w_i \, ds & (4.10) \\
\int_{\Omega} \left( \frac{\partial v^h}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial v^h}{\partial y} \frac{\partial w_i}{\partial y} + \operatorname{Re} \left( u^h \frac{\partial v^h}{\partial x} + v^h \frac{\partial v^h}{\partial y} + \frac{\partial p^h}{\partial y} \right) w_i \right) d\Omega
\end{aligned}$$

$$= \int_{\Gamma} \frac{\partial v^h}{\partial n} w_i \, ds \quad (4.11)$$

$$\int_{\Omega} \left( \frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} \right) q_i \, d\Omega = 0 \quad (4.12)$$

In the boundary integrals, we should point out the meaning of the expression  $\frac{\partial u^h}{\partial n}$ , which is sometimes called the *normal derivative*:

$$\frac{\partial u^h}{\partial n} \equiv \frac{\partial u^h}{\partial x} \hat{n}_x + \frac{\partial u^h}{\partial y} \hat{n}_y, \quad (4.13)$$

where, as before, the outward unit normal vector to the flow region  $\Omega$  at  $(x, y)$  is the vector  $(\hat{n}_x(x, y), \hat{n}_y(x, y))$ .

Now the unknowns in this system seem to be the functions  $u^h(x, y)$ ,  $v^h(x, y)$ , and  $p^h(x, y)$ , but since we assume that these functions are represented by linear combinations of the basis functions, the true unknowns are the coefficients, that is, the numbers  $u_i^h$ ,  $v_i^h$ , and  $p_i^h$ . We could make this explicit by replacing all occurrences of the functions  $u^h(x, y)$ ,  $v^h(x, y)$ , and  $p^h(x, y)$  in the finite element equations by the finite sums, but the resulting expressions are much too cumbersome to work with.

By construction, there are exactly as many of these equations as there are unknown coefficients  $u_i^h$ ,  $v_i^h$  and  $p_i^h$ . These equations are *nonlinear*, as are the original Navier Stokes equations, but they are strictly algebraic (in fact, they are actually *quadratic*). That means that they may be treated by a form of Newton's method in which the linearized operator to be inverted is simply a Jacobian matrix.

## 4.8 Discretizing the Boundary Conditions

In order to have a completely posed discretized problem, we must account for the boundary conditions that were applied in the continuous case. We will attempt to do this in such a way that we eliminate redundant equations. In this process, we will disrupt somewhat the original situation, in which, for instance, every corner node always had two associated unknown velocity components and an unknown pressure.

The boundary conditions for the continuous case were:

$$\begin{aligned} v(x, y) &= 0 \text{ along the boundary;} \\ u(0, y) &= \textit{Inflow}(y, \lambda) \text{ along the inflow;} \\ u(x, y) &= 0 \text{ on the walls and the bump;} \\ p(x_{max}, y_{max}) &= 0. \end{aligned}$$

Note, however, that we do *not* have to impose the boundary condition

$$\frac{\partial u}{\partial x}(x_{max}, y) = 0 \text{ along the outflow,} \quad (4.14)$$

because this is automatically enforced weakly by the finite element method itself.

As a first step towards discretization, we may simply interpret these conditions as being statements about the value of the state solution at the corresponding velocity or pressure nodes. In other words, the discrete boundary conditions may be interpreted as:

$$v^h(x, y) = 0 \text{ at all velocity nodes on } \Gamma; \quad (4.15)$$

$$u^h(0, y) = \textit{Inflow}^h(y, \lambda) \text{ at all inflow velocity nodes;} \quad (4.16)$$

$$u^h(x, y) = 0 \text{ at all velocity nodes on the walls and the bump;} \quad (4.17)$$

$$\frac{\partial u^h}{\partial x}(x_{max}, y) = 0 \text{ at outflow velocity nodes;} \quad (4.18)$$

$$p^h(x_{max}, y_{max}) = 0 \text{ for the upper right pressure node.} \quad (4.19)$$

In the above equations,  $Inflow^h(y, \lambda)$  denotes that discretization has been applied to our original inflow function. In our case, we will use the interpolant, that is, at every velocity node, it will be true that:

$$Inflow^h(y, \lambda) = Inflow(y, \lambda) \quad (4.20)$$

The discretized boundary conditions may be translated into algebraic equations involving the finite element coefficients, and we may append these equations to the discretized state equations. However, most of these boundary conditions are simple enough that from them we can easily determine the values of certain coefficients. It seems wasteful to treat such quantities as unknowns. We will find that, by proper application of the given boundary conditions, we can solve for many of the unknowns along the boundary. For such eliminated unknowns, we can discard the corresponding finite element equations and basis functions from our system.

Let us consider the vertical velocity boundary condition,  $v^h(x, y) = 0$  at all velocity nodes on  $\Gamma$ . At each velocity node  $i$  along the boundary, we know that the value of the associated finite element coefficient  $v_i^h$  is zero; in other words,  $v_i^h$  is *not* an unknown. We may therefore apply the boundary condition by dropping the discrete finite element equation associated with  $v_i^h$ , which would normally be used to solve for the unknown coefficient. In all other equations we replace occurrences of  $v_i^h$  by 0. Node  $i$  will no longer have an associated vertical velocity.

We handle the Dirichlet boundary conditions on the horizontal velocity at the inflow, along the walls and on the bump, and the condition on the pressure in the upper right corner in a similar way.

Since we have already eliminated the velocity unknowns associated with the walls, bump, and inflow, all remaining velocity basis functions  $w_i$  are zero on  $\Gamma$ , except, possibly, along

the outflow boundary. But, as we have pointed out, the finite element method automatically weakly enforces the boundary conditions:

$$\frac{\partial u^h}{\partial x} = \frac{\partial u^h}{\partial n} = 0 = \frac{\partial v^h}{\partial x} = \frac{\partial v^h}{\partial n} = 0 \quad (4.21)$$

which means that in Equations (4.10) and (4.11) the right hand side integrals over  $\Gamma$  drop out, leaving us with the slightly simpler system:

$$\int_{\Omega} \left( \frac{\partial u^h}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial u^h}{\partial y} \frac{\partial w_i}{\partial y} + Re(u^h \frac{\partial u^h}{\partial x} + v^h \frac{\partial u^h}{\partial y} + \frac{\partial p^h}{\partial x}) w_i \right) d\Omega = 0 \quad (4.22)$$

$$\int_{\Omega} \left( \frac{\partial v^h}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial v^h}{\partial y} \frac{\partial w_i}{\partial y} + Re(u^h \frac{\partial v^h}{\partial x} + v^h \frac{\partial v^h}{\partial y} + \frac{\partial p^h}{\partial y}) w_i \right) d\Omega = 0 \quad (4.23)$$

$$\int_{\Omega} \left( \frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} \right) q_i d\Omega = 0 \quad (4.24)$$

Because the inflow velocities are generally nonzero, the treatment of the boundary conditions means that the previously unknown velocity coefficient  $u_i^h$  at an inflow node is replaced by a known constant. In some cases, this will result in the computation of a term which we will move to the right hand side.

## 4.9 Estimating the Solution at Nearby Parameters

Let us suppose, for a moment, that the definition of the discretized Navier Stokes problem we are solving includes a set of parameters  $\beta$ , as in Chapter 2, and that we can consider the corresponding flow solutions to be functions of those parameters. If we wish to keep this relationship in mind, we write the flow solution as  $(u^h(\beta), v^h(\beta), p^h(\beta))$ .

Now our discrete Navier Stokes equations and boundary conditions comprise a system of nonlinear algebraic equations, which we plan to solve using some iterative scheme. The very first time that we try to solve the nonlinear system, say at the parameter values  $\beta_0$ , we'll use

zero as the starting guess for  $(u^h(\beta_0), v^h(\beta_0), p^h(\beta_0))$ . But if we are able to solve that initial system, then if we require the values of  $(u^h, v^h, p^h)$  at a nearby set of parameters, we should surely be able to produce a better starting point for the next iteration than simply the zero guess. One way to get a better estimate might be to use the previous solution:

$$u^h(\beta_0 + \Delta\beta) \approx u^h(\beta_0), \quad (4.25)$$

but we usually have data that approximates the first partial derivatives such as  $\frac{\partial u^h(\beta)}{\partial \beta_i}$ , since we are computing the discretized sensitivities or finite difference estimates of the sensitivities.

Hence, we can make an even better estimate by using the first order *Euler prediction*:

$$u^h(\beta_0 + \Delta\beta) \approx u^h(\beta_0) + \sum_i \frac{\partial u^h(\beta_0)}{\partial \beta_i} \Delta\beta_i. \quad (4.26)$$

## 4.10 Picard Iteration Applied to the Discrete Problem

In some cases, the starting point produced by the Euler prediction will be close enough to the correct solution that Newton's method can converge. But this may not be so, particularly for higher Reynolds numbers. Thus, in some cases, we must work harder to produce a satisfactory starting point.

In such situations, we will use the Euler estimate as the starting point for a Picard iteration scheme. We will use the following decomposition  $H(X, X_{old})$  of the discrete state equations  $F(X)$ :

$$\int_{\Omega} \left( \frac{\partial u^h}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial u^h}{\partial y} \frac{\partial w_i}{\partial y} + Re(u_{old}^h \frac{\partial u^h}{\partial x} + v_{old}^h \frac{\partial u^h}{\partial y} + \frac{\partial p^h}{\partial x}) w_i \right) d\Omega = 0 \quad (4.27)$$

$$\int_{\Omega} \left( \frac{\partial v^h}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial v^h}{\partial y} \frac{\partial w_i}{\partial y} + Re(u_{old}^h \frac{\partial v^h}{\partial x} + v_{old}^h \frac{\partial v^h}{\partial y} + \frac{\partial p^h}{\partial y}) w_i \right) d\Omega = 0 \quad (4.28)$$

$$\int_{\Omega} \left( \frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} \right) q_i d\Omega = 0 \quad (4.29)$$

along with the boundary conditions (4.15) through (4.19). Given the value of  $X_{old} =$

$(u_{old}^h, v_{old}^h, p_{old}^h)$  the equations  $H(X, X_{old}) = 0$  are a linear system solvable for the new coefficients  $X = (u^h, v^h, p^h)$  that define the next approximation to the flow solution.

The following theorem from Rheinboldt and Ortega [24] suggests how our Picard iteration could be analyzed and modified to guarantee global convergence:

**Theorem 4.2 (Global Convergence of Picard Iteration)** *Let  $A \in L(\mathbb{R}^n)$  be a symmetric and positive definite matrix, assume that  $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^n$  is continuously differentiable, that  $\phi'(x)$  is symmetric and positive semidefinite for all  $x \in \mathbb{R}^n$ , and that there is a constant  $\beta < +\infty$  for which*

$$\|\phi'(x)\|_2 \leq \beta, \forall x \in \mathbb{R}^n. \quad (4.30)$$

*Then the equation  $Ax + \phi(x) = 0$  has a unique solution  $x^*$ , and for any  $x_0 \in \mathbb{R}^n$ , the sequence  $\{x_k\}$  defined by the Picard iteration:*

$$(A + (\beta/2)I)x_{k+1} = (\beta/2)x_k - \phi(x_k) \quad (4.31)$$

*converges to  $x^*$ .*

The version of Picard iteration we are currently employing does not attempt to estimate the value of the constant  $\beta$ , and thus has the form

$$Ax_{k+1} = -\phi(x_k) \quad (4.32)$$

which is not guaranteed to converge. Indeed, we have seen cases where our Picard formulation fails in practice. In the future, we intend to investigate the reformulation of our iteration along the lines suggested by the theorem, so that we may guarantee global convergence.

Picard iteration can be particularly useful at higher Reynolds numbers, where the radius of convergence for Newton's method is generally much decreased. However, at least when the hypotheses of Theorem 4.2 apply, the guaranteed convergence is only of linear order. Thus, a

reasonable use of Picard iteration is simply to reduce the residual of the initial point enough that the Newton method, with its superior quadratic convergence, may be applied to bring the iteration to a rapid conclusion.

With our current Picard iteration, which does not satisfy the hypotheses of Theorem 4.2, we cannot guarantee global convergence. Thus, our iteration may fail. It is sometimes possible to recover from such a failure; for instance, if we are computing a solution at a high Reynolds number, we can retry the computation by finding a solution at a low Reynolds number, and then using the Euler prediction method discussed in the previous section to produce a better starting estimate at the Reynolds number of interest. In other cases, we may be able to recover by reducing the mesh parameter  $h$ , although this is a costly solution.

## 4.11 The Newton Method Applied to the Discrete Problem

Assuming we have a starting point  $X_0$  that is sufficiently close to a solution of  $F(X) = 0$ , Newton's method produces a sequence of iterates by computing a correction  $\Delta X$  to the current iterate. The heart of the iteration is the pair of calculations:

$$\Delta X_i \leftarrow -FP(X_i)^{-1} F(X_i) \tag{4.33}$$

$$X_{i+1} \leftarrow X_i + \Delta X_i \tag{4.34}$$

The expensive part of the calculation is the computation of the Jacobian matrix  $FP$  and its factorization. Many attempts to improve the efficiency of the Newton method begin by trying to reduce the storage used for this matrix or the frequency with which it is calculated.

As soon as we have the new iterate  $X_{i+1}$ , we can compute the residual  $F(X_{i+1})$ . The norm of this quantity is an indication of how far away we are from a solution. Thus, if we monitor

this quantity, its behavior will allow us to decide whether to take another iterative step, or to accept  $X_{i+1}$  as our approximation to the solution, or to determine that the iteration has failed to converge, or is actually diverging.

The local behavior of Newton's method is laid out in the following theorem (Dennis and Schnabel, [10]):

**Theorem 4.3 (Local Convergence of Newton's Method in  $R^n$ )** *Suppose that the function  $F : R^n \rightarrow R^n$  is continuously differentiable in an open convex set  $\mathcal{D}$ , and that there is a point  $x^* \in \mathcal{D}$  at which  $F(x^*) = 0$ . Suppose that the Jacobian matrix  $FP(x^*)$  is invertible, with  $\|FP^{-1}(x^*)\| = \beta$ . Suppose, finally, that there is an open ball  $B_r(x^*)$  of radius  $r$  centered at  $x^*$ , with  $B_r(x^*) \subset \mathcal{D}$ , so that for every  $x \in B_r(x^*)$ ,  $FP(x)$  is Lipschitz continuous, with Lipschitz constant  $\gamma$  independent of  $x$ .*

*Then there is an  $\epsilon > 0$  with the property that, for any starting point  $x_0 \in B_\epsilon(x^*)$ , the sequence of Newton iterates  $\{x_k\}$  is well defined, converges to  $x^*$ , and for each  $k$  satisfies*

$$\|x_{k+1} - x^*\| \leq \beta\gamma\|x_k - x^*\|^2. \quad (4.35)$$

This theorem informs us that if we are close enough to the correct solution, Newton's method will converge quadratically. Conversely, practical experience shows that if the starting point is too far away, Newton's method will diverge. There is no general way to compute this critical distance beforehand; therefore, we must expect from time to time that we must deal with a Newton failure. In some cases, it may be possible to recover by producing a better starting guess, or trying to solve an easier problem (one whose parameters are closer to the parameters of a problem we've already solved). However, some failures of Newton's method will cause the entire optimization to fail.

To implement Newton's method, we must differentiate each finite element equation with

respect to the unknown coefficients, a tedious computation. For the state equations (4.22)-(4.24) associated with basis functions  $w_i$  or  $q_i$ , we list the entries in the Jacobian  $FP$  with respect to a typical coefficient  $u_j$ ,  $v_j$  or  $p_j$ . These entries are evaluated at the “old” solution point,  $(u_{old}^h, v_{old}^h, p_{old}^h)$ , and are used to solve for the change in the coefficients that will give us the next Newton approximant.

The derivative of the horizontal momentum equation associated with basis function  $w_i$  with respect to coefficient  $u_j^h$  is:

$$\int_{\Omega} \left( \frac{\partial w_j}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial w_j}{\partial y} \frac{\partial w_i}{\partial y} + Re(w_j \frac{\partial u_{old}^h}{\partial x} + u_{old}^h \frac{\partial w_j}{\partial x} + v_{old}^h \frac{\partial w_j}{\partial y}) w_i \right) d\Omega; \quad (4.36)$$

the derivative of the horizontal momentum equation associated with basis function  $w_i$  with respect to coefficient  $v_j^h$  is:

$$\int_{\Omega} Re w_j \frac{\partial u_{old}^h}{\partial y} w_i d\Omega; \quad (4.37)$$

the derivative of the horizontal momentum equation associated with basis function  $w_i$  with respect to coefficient  $p_j^h$  is:

$$\int_{\Omega} Re \frac{\partial q_j}{\partial x} w_i d\Omega; \quad (4.38)$$

the derivative of the vertical momentum equation associated with basis function  $w_i$  with respect to coefficient  $u_j^h$  is:

$$\int_{\Omega} Re w_j \frac{\partial v_{old}^h}{\partial x} w_i d\Omega; \quad (4.39)$$

the derivative of the vertical momentum equation associated with basis function  $w_i$  with respect to coefficient  $v_j^h$  is:

$$\int_{\Omega} \left( \frac{\partial w_j}{\partial x} \frac{\partial w_i}{\partial x} + \frac{\partial w_j}{\partial y} \frac{\partial w_i}{\partial y} + Re(u_{old}^h \frac{\partial w_j}{\partial x} + w_j \frac{\partial v_{old}^h}{\partial y} + v_{old}^h \frac{\partial w_j}{\partial y}) w_i \right) d\Omega; \quad (4.40)$$

the derivative of the vertical momentum equation associated with basis function  $w_i$  with respect to coefficient  $p_j^h$  is:

$$\int_{\Omega} Re \frac{\partial q_j}{\partial y} w_i d\Omega; \quad (4.41)$$

the derivative of the continuity equation associated with basis function  $q_i$  with respect to coefficient  $u_j^h$  is:

$$\int_{\Omega} \frac{\partial w_j}{\partial x} q_i \, d\Omega; \quad (4.42)$$

and the derivative of the horizontal momentum equation associated with basis function  $q_i$  with respect to coefficient  $v_j^h$  is:

$$\int_{\Omega} \frac{\partial w_j}{\partial y} q_i \, d\Omega. \quad (4.43)$$

We note that the Jacobian that defines the Newton iteration is *not* the same as the system matrix used for Picard iteration, although they have the same structure, and share most of the same entries. This means that it will not be possible to reuse a Picard iteration matrix during the Newton iteration, for instance, a fact that will reduce the options for efficient execution later on.

## 4.12 The Approximation of the Continuous Problem

We have shown how the discrete problem is derived from the continuous problem. While it is not clear how to solve the continuous problem, we have outlined an algorithm for producing the solution of the discrete problem. We would like, therefore, to determine in what sense the solution of the discrete problem is a good approximation to the solution of the continuous problem.

It is shown by, for instance, Boland and Nicolaides [1], that as long as the solution of the continuous problem  $(\mathbf{u}, p)$  has a particular smoothness, then the error in the approximation can be bounded in terms of the mesh parameter  $h$ .

Before we can present their results, we must develop the appropriate notation. First, we

define the following spaces of functions:

$$L^2(\Omega) \equiv \{f : \Omega \rightarrow R : \int_{\Omega} f^2 d\Omega < \infty\}. \quad (4.44)$$

We then define the following related spaces:

$$L_0^2(\Omega) \equiv \{f \in L^2(\Omega) : \int_{\Omega} f d\Omega = 0\}, \quad (4.45)$$

and the *Sobolev space*:

$$H^k(\Omega) \equiv \{f \in L^2(\Omega) : D^s f \in L^2(\Omega), s=1, \dots, k\} \quad (4.46)$$

where  $D^s f$  denotes any derivative of order  $s$ .

We define a space of functions that is zero on the boundary of  $\Omega$ :

$$H_0^k(\Omega) \equiv \{f \in H^k(\Omega) : f = 0 \text{ on } \Gamma\}. \quad (4.47)$$

We also define the following norm on the space  $L^2$  (and, of course, on all its subspaces):

$$\|f\|_0 \equiv \left(\int_{\Omega} f^2 d\Omega\right)^{\frac{1}{2}}, \quad (4.48)$$

as well as the following seminorm on functions in  $H_0^1(\Omega)$ :

$$|f|_1 \equiv \left(\sum_{i=1}^n \left\|\frac{\partial f}{\partial x_i}\right\|_0^2\right)^{\frac{1}{2}}. \quad (4.49)$$

We have written  $\mathbf{f}$  when a function takes values in  $R^n$ , and in such cases, we define the corresponding spaces  $\mathbf{H}_0^k(\Omega)$  and so on.

We can now paraphrase the approximation theorem, which states that, for  $m = 2$  or  $3$ , if the solutions  $\mathbf{u}$  and  $p$  to the continuous Navier Stokes problem satisfy:

$$\mathbf{u} \in \mathbf{H}^m(\Omega) \cap \mathbf{H}_0^1(\Omega), \quad (4.50)$$

$$p \in H^{m-1}(\Omega) \cap L_0^2(\Omega), \quad (4.51)$$

then the following error estimates, which compare the solutions of the continuous problem to the solutions  $\mathbf{u}^h$  and  $p^h$  of the discretized problem, hold uniformly in  $h$ :

$$|\mathbf{u} - \mathbf{u}^h|_1 = O(h^{m-1}) \quad (4.52)$$

$$\|\mathbf{u} - \mathbf{u}^h\|_0 = O(h^m) \quad (4.53)$$

$$\|p - p^h\|_0 = O(h^{m-1}) \quad (4.54)$$

For our problem, the exact velocities  $\mathbf{u}$  can not, in general, be supposed to lie in  $\mathbf{H}^3(\Omega)$ , because our region is not convex. The actual smoothness of the solution depends strongly on the amount of deviation from convexity, most prominently in the angles made where the bump surface joins the bottom of the channel. For the class of problems we have chosen, we can expect that the exact velocities will be elements of  $\mathbf{H}^2(\Omega)$ . This means that, if we can satisfy the other hypotheses of the theorem, that we can expect  $O(h^2)$  accurate approximation of velocity, and  $O(h)$  accurate approximation of pressure.

On the face of it, the solutions  $\mathbf{u}$  and  $p$  of the continuous problem do not satisfy the other conditions necessary to apply the results. This is because we do not have zero boundary conditions on  $\mathbf{u}$ , nor do we require that  $p$  have a zero integral over the region. However, these difficulties are easily overcome.

The discrepancy in the pressure conditions is truly minor. If we replace the original pressure condition by

$$\int_{\Omega} p(x, y) \, d\Omega = 0, \quad (4.55)$$

we will recompute the original result, plus or minus a constant. The choice between the two boundary conditions is strictly a matter of convenience.

For the velocity boundary conditions, we appeal to the discussion in Gunzburger [15], where it is shown that the error estimate given for the homogeneous boundary condition still holds if

a nonhomogeneous condition is given, assuming the discretized problem uses the interpolant of the original boundary condition function.

These results tell us that the solution to the finite element problem is indeed close to the solution to the original continuous problem, and that the error of approximation goes to zero with  $h$ .

Spatial derivatives of  $\mathbf{u}$  and  $p$ , which are also produced by the finite element method, are less accurately computed. In general, each degree of spatial differentiation reduces the order of accuracy of approximation by one power of  $h$ . We will find later that, in order to produce certain quantities called geometric sensitivities, we will want accurate approximation of the derivatives of the continuous solution such as  $u_y \equiv \frac{\partial u}{\partial y}$ . We can see that, given the smoothness assumptions about the solution, we can expect a lower order approximation of  $u_x$  and  $u_y$ :

$$(\|u_x - u_x^h\|_0^2 + \|u_y - u_y^h\|_0^2)^{\frac{1}{2}} = O(h), \quad (4.56)$$

which we will take to mean that we have  $O(h)$  error in either approximate derivative  $u_x^h$  or  $u_y^h$ . This fact will have ramifications on the accuracy of the discretized sensitivity computations.