# Numerical Methods for Accurate Computation of Design Sensitivities

Dawn L. Stewart

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

John Burns, Chair
Jeff Borggaard
Gene Cliff
Terry Herdman
John Rossi

June 10, 1998
Blacksburg, Virginia

Keywords: Fluid Flow, Finite Element Method, Navier-Stokes, Partial Differential
Equations, Local and Global Projections, Sensitivity Analysis

# Numerical Methods for Accurate Computation of Design Sensitivities

Dawn L. Stewart

(ABSTRACT)

This work is concerned with the development of computational methods for approximating sensitivities of solutions to boundary value problems. We focus on the continuous sensitivity equation method and investigate the application of adaptive meshing and smoothing projection techniques to enhance the basic scheme. The fundamental ideas are first developed for a one dimensional problem and then extended to 2-D flow problems governed by the incompressible Navier-Stokes equations. Numerical experiments are conducted to test the algorithms and to investigate the benefits of adaptivity and smoothing.

# Acknowledgments

It is a pleasure for me to take this page to thank the organizations and people that made my study possible. First, I would like to thank the personnel committe at the United State Air Force Academy Mathematics department for selecting me to participate in the PhD progam. They opened the door which would enable me to fulfill a lifelong dream. In addition, I would like to thank the Interdisiplinary Center for Applied Mathematics (ICAM) for sponsoring my travel to numerous conferences under AFOSR grant F49620-96-1-0329. The opportunity to interact with other researchers and to present my results at professional conferences was invaluable. Lastly, many, many thanks to Melissa Chase, the person who keeps things running smoothly at ICAM. There were many times her quick thinking kept me out of trouble and her smile at the door was always a welcome sight.

Special thanks to my advisor, Dr John Burns, for his patience and support, especially in dealing with me! Without his never-ending encouragement and guidance, this dissertation would not exist. Also, the numerous suggestions from the rest of my committee throughout the years as well as their comments on my dissertation greatly enhanced the quality of this work and made my study much more enjoyable.

I would like to thank my children, Katie and David, for putting up with my travel and enduring the many times I was lost in thought about some mathematical idea instead of listening to what they were saying. The many *"you can do it"* 's from my parents kept me going when I got frustrated.

Lastly, I would like to thank three very special people without whom I would not have had the emotional strength to get through these past few years. They are my fellow graduate students Lesa Beverly, Kevin Hulsing, and Lisa Stanley. They were there to rejoice with me in my successes and encourage me in my failures. They truly were the glue that kept this jigsaw puzzle from falling apart.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

During the past decade, considerable effort has been devoted to the development of computational tools for the analysis, design, control and optimization of complex physical systems. Sensitivity analysis plays an important role in all aspects of this effort. Many simulation tools can be enhanced by using local information provided by state sensitivities to obtain "near-by" solutions. For example Godfrey ([20]) used flow sensitivities to reduce computational time in simulations of multi-species, chemically reacting flows. Sensitivity methods have long been used to design and optimize structures (see [23], [18] ). Moreover, sensitivity reduction is often employed in the design robust feedback controllers (see [15], [26], [16], [27], [28]). In recent years, applications of gradient based optimization algorithms to multi-disciplinary design optimization (MDO) problems have lead to renewed interest in state sensitivities as one method to compute cost function gradients (see [4], [7]).

In order to make effective use of sensitivities in analysis, design, and control, accurate and efficient computational algorithms are essential. In addition, the natural trade-off between accuracy and efficiency can often be exploited to produce the best overall computational tool for a specific application. For example, when used to enhance a simulation tool, accuracy may be more important than speed. However, when used for gradient computations in an optimization loop, speed may be more crucial than high accuracy.

Two early numerical techniques employed for sensitivity calculations were the straightforward finite difference calculations and the "discretize-then-differentiate" approach. The "discretize-then-differentiate" scheme approximates sensitivities by first employing some discretization scheme to approximate the solution to a PDE and then implicitly differentiating this result to to obtain a sensitivity approximation scheme. Recent advances in this area include the development of automatic differentiation packages (see [32], [21]), like AD-

IFOR, which given source code for state calculations, generates source code for sensitivity calculations.

There are a number of significant disadvantages to each of the above approaches. One disadvantage of finite difference techniques is that they are computationally intensive since they require two flow solves at different parameter values to form a difference quotient. In large aerospace flow problems, this is sometimes an extremely severe requirement because mesh generation itself can take weeks to months. In addition, it is not generally known a-priori what is a sufficiently small step size to ensure accurate approximations, making it difficult to incorporate these ideas into optimizations schemes. A procedure for estimating an optimal step-size is described in [30]. The optimal step-size attempts to balance truncation error and round-off error. Unfortunately, since the discretization is generally non-uniform, the truncation error varies from point to point so that the results must be used in combination with some other decision algorithm.

A disadvantage of the "discretize-then-differentiate" technique is that in cases where the mesh is parameter dependent, as is the case in shape optimization problems, then differentiation of the discrete PDE leads to mesh sensitivities on the right hand side. Although there has been much work done in recent years to get a handle on these quantities (see [38]), calculating mesh derivatives is still not well understood, particularly in cases where the meshes are prescribed adaptively.

In recent years several new approaches have been developed in an attempt to alleviate some of the disadvantages of the two approaches mentioned above. The idea is to employ a "differentiate-then-discretize" scheme or the so-called continuous sensitivity equation method (SEM). The SEM consists of implicitly differentiating the PDE to obtain a sensitivity equation (SE). Then both the PDE and the SE are discretized to obtain finite dimensional equations for numerical approximations to the PDE and the SE.

This technique reduces some of the problems mentioned above. The SEM eliminates the need for a second flow solve saving costly computer time. Also, since the differentiation is done before the discretization, it eliminates the need to calculate "mesh" derivatives, an exceedingly difficult task especially in the case of complicated 2-D and 3-D flow problems. The method has another advantage. The sensitivity equations are always linear and, in principal, it is simple to modify existing flow solvers so that one additional Newton step at the end of a flow solve is sufficient to approximate sensitivities. This feature makes the SEM an easy approach to implement.

The SEM produces an approximation for the solution to the continuous sensitivity equation. In some cases, especially in the case of optimization, what one really needs is the sensitivity of the discrete solution. These solutions are not necessarily equivalent. Because of this

discrepancy, it is not obvious that the SEM would lead to convergence of the optimization algorithm. However, recent theoretical and numerical studies have shown that this issue can often be addressed.

In [11], Burkardt applied this sensitivity approximation scheme to a shape optimization problem involving incompressible fluids. He found that in many cases the SEM gave sufficiently good approximations to allow the optimization algorithm to converge. However, Burkardt also provided examples where the differences between the discrete sensitivities and the numerical solution of the continuous sensitivity equation were large enough to destroy the convergence of the optimizer. This was true even in cases where the finite element method had done a good job approximating the solution of the continuous problem. He also found some problems at higher Reynolds number flows. One of the issues which may have been a factor is the accuracy of the flow derivative information used in the sensitivity calculation. We will show in this paper that improved derivative information can drastically improve the sensitivity approximations and, in turn, have dramatic effects on the performance of an optimization scheme.

In [3], Borggaard provided a partial analysis of this problem. He proved that if the discrete approximation to the continuous sensitivity equation was "close enough" to the sensitivity of the discrete solution, then an appropriately chosen optimization algorithm would converge. He also showed that using the SEM, rather than the finite difference approach, could reduce CPU times by 50 percent or more. This is an extremely important savings for high-cost flow problems.

The influence of flow discontinuities on sensitivity approximations was investigated by Appel in [2]. Appel compared all three sensitivity approximation techniques on Euler flows as well as the 1-D Riemann problem, a problem for which an exact solution is known. He found that the shocks in the solution led to errors in the sensitivity approximations for all of the techniques.

We are interested in applications that lend themselves to physics based modeling. Specifically, we shall concentrate on problems where the governing equations are differential equations. We use a simple boundary value problem in one spatial dimension (1-D) to introduce the ideas and illustrate the various methods. This model problem is described by an ordinary differential equation. Once the basic ideas and techniques have been tested on this 1-D problem, we turn to more substantial problems from fluid dynamics. In particular, we consider 2-D fluid flows governed by the Navier-Stokes equations.

# Chapter 2

# A 1-D Model Problem

In this chapter, we consider a simple 1-D non-linear boundary value problem. We formulate an optimization problem and provide a detailed presentation of the SEM. We use finite elements to construct approximations and compare the numerical results to exact solutions. The goal of this chapter is to use the model problem to describe the basic ideas, identify the fundamental issues, and set the stage for more complex problems to come.

## 2.1   Model Problem

We concentrate on the boundary value problem defined by the non-linear differential equation

$$\frac{d^2}{dx^2}w(x) + \frac{1}{8}\left[\frac{d}{dx}w(x)\right]^3 = 0 \quad \text{for} \ \ 0 < x < q, \tag{2.1}$$

with boundary conditions

$$w(0) = 0 \quad \text{and} \quad w(q) = 4. \tag{2.2}$$

For each $q > 1$, the exact solution to this boundary value problem is given by

$$w(x) = w(x, q) = 4\sqrt{x + \frac{(q-1)^2}{4}} - 2(q-1). \tag{2.3}$$

Let $0 < \hat{x}_1 < \hat{x}_2 < ... < \hat{x}_p < 1$ be fixed locations and assume that $\hat{w}_j$ is data representing

values of $w(x)$ at $\hat{x}_j$. Consider the inverse design problem: Find $q^\star > 1$ such that

$$J_p(q^\star) \leq J_p(q) \triangleq \frac{1}{2} \sum_{j=1}^{p} (w(\hat{x}_j, q) - \hat{w}_j))^2, \tag{2.4}$$

where $w(x, q)$ is the solution of (2.1) - (2.2) and the integer $p$ represents the number of data points. In gradient-based optimization one needs the derivative

$$\nabla_q J_p(q) = \sum_{j=1}^{p} (w(\hat{x}_j, q) - \hat{w}_j) \frac{\partial}{\partial q} w(\hat{x}_j, q). \tag{2.5}$$

One approach to the evaluation of this gradient at $\bar{q}$ is to "compute" the state, $w(\hat{x}_j, \bar{q})$, the sensitivity, $\frac{\partial}{\partial q} w(\hat{x}_j, \bar{q})$, and form the computation (2.5). This involves first solving (2.1) - (2.2) for $w(x, q)$ and then computing the sensitivity $\frac{\partial}{\partial q} w(x, q)$.

## 2.2   The Sensitivity Equation

One benefit of using the model problem is that we can calculate the sensitivity $\frac{\partial}{\partial q} w(x, q)$ by direct differentiation of (2.3). In particular,

$$\frac{\partial}{\partial q} w(x, q) = \frac{q - 1}{\sqrt{x + \frac{(q-1)^2}{4}}} - 2. \tag{2.6}$$

On the other hand, we can implicitly "differentiate" the boundary value problem (2.1) - (2.2) and obtain a boundary value problem for the sensitivity $\frac{\partial}{\partial q} w(x, q) \triangleq s(x, q) = s(x)$. It follows that $s(x)$ satisfies the linear differential equation

$$\frac{d^2}{dx^2} s(x) + \frac{3}{8} \left[ \frac{d}{dx} w(x) \right]^2 \frac{d}{dx} s(x) = 0, \tag{2.7}$$

with boundary conditions

$$s(0) = 0 \qquad \text{and} \qquad s(q) = -\frac{\partial}{\partial x} w(q). \tag{2.8}$$

It is important to be cautious when using (2.2) to derive the boundary conditions for the sensitivity equation. The first boundary condition,

$$s(0) = \frac{\partial}{\partial q} w(0) = 0,$$

is rather obvious. However, deriving the second boundary condition in (2.8) can be tricky (especially in 2-D and 3-D domain optimization problems). Using the boundary condition (2.2), the chain rule leads to the correct boundary condition

$$0 = \frac{D}{Dq} w(q, q) = \left\{ \frac{\partial}{\partial x} w(x, q) + \frac{\partial}{\partial q} w(x, q) \right\} \Bigg|_{(x,q)=(q,q)}.$$

Since this is a model for a more difficult applied problem for which we would not know the state, $w$, and the sensitivity, $s$, exactly, we are interested in constructing good methods for obtaining numerical approximations of these quantities.

## 2.3    Domain Mapping

For computational purposes, a series of transformations are used to first map the problem to a fixed "computational domain" and second to transform the non-homogeneous boundary conditions to homogeneous Dirichlet conditions. This is standard in many CFD (computational fluid dynamics) algorithms. What is not typical is that we apply a similar transformation to the sensitivity equation. Let $T : [0, q] \rightarrow [0, 1]$ be defined by

$$T(x, q) = \frac{x}{q} = \xi. \tag{2.9}$$

To avoid confusion, we use $x$ for the independent variable on $[0, q]$ and $\xi$ for the independent variable on $[0, 1]$. Note that $T = T(x, q)$ depends explicitly on the "shape parameter" q. For each $q > 1$, the inverse transformation

$$[T(x, q)]^{-1} \triangleq M(\xi, q) : [0, 1] \rightarrow [0, q]$$

is defined by

$$M(\xi, q) = q\xi = x. \tag{2.10}$$

Let $\hat{w}(\xi) = \hat{w}(\xi, q) = w(M(\xi, q), q)$ and define $z(\xi)$ by

$$z(\xi) = z(\xi, q) = \hat{w}(\xi) - 4\xi. \tag{2.11}$$

Applying this transformation, we obtain the Dirichlet problem defined on the computational domain $[0, 1]$ by the differential equation

$$\frac{d^2}{d\xi^2} z(\xi) + \frac{1}{8q} \left[ \frac{d}{d\xi} z(\xi) + 4 \right]^3 = 0 \tag{2.12}$$

with Dirichlet boundary conditions

$$z(0) = 0 \qquad \text{and} \qquad z(1) = 0. \tag{2.13}$$

In a similar manner, we transform the sensitivity equation (2.7) - (2.8) to the computational domain $[0, 1]$. In particular, we let $\hat{s}(\xi) = \hat{s}(\xi, q) = s(M(\xi, q), q)$ and define $p(\xi, q)$ by

$$p(\xi) = p(\xi, q) = \hat{s}(\xi) + \frac{\frac{d}{d\xi}\hat{w}(1)}{q}\xi. \tag{2.14}$$

It follows that the transformed sensitivity $p(\xi)$ satisfies the equation

$$\frac{d^2}{d\xi^2}p(\xi) + \frac{3}{8q}\left[\frac{d}{d\xi}z(\xi) + 4\right]^2 \left(\frac{d}{d\xi}p(\xi) - \frac{\frac{d}{d\xi}z(1) + 4}{q}\right) = 0 \tag{2.15}$$

with Dirichlet boundary conditions

$$p(0) = 0 \qquad \text{and} \qquad p(1) = 0. \tag{2.16}$$

In practice, one must use some numerical scheme to solve the boundary value problem (2.1) - (2.2), and the computation of $\frac{\partial}{\partial q}w(x, q)$ must be accomplished by using this approximate solution. We approach this problem by solving the corresponding sensitivity equation. As shown below, there are many "natural" numerical schemes that one can employ in this approach. Although we discuss several schemes, we will concentrate on a projection method approach in later chapters. The basic idea can be extended to complex aerodynamic flow problems. However, many theoretical and technical issues are not yet settled.

**Comment:** It is important to note that the construction of the transformed state equation (2.12) - (2.13) and the transformed sensitivity equation (2.15) - (2.16) requires the derivative (in space) of the transformation $M(\xi, q)$. In particular, one needs $\frac{\partial}{\partial\xi}M(\xi, q)$ or else a numerical approximation of $\frac{\partial}{\partial\xi}M(\xi, q)$. This issue is addressed in many CFD codes and there are good methods for dealing with this problem (e.g. see [38]). However, there is no need to compute the derivative $\frac{\partial}{\partial q}M(\xi, q)$ with this approach. On the other hand, if one transforms the state equations and then derives the sensitivity equation for the transformed state equation, then the chain rule requires the calculation of $\frac{\partial}{\partial q}M(\xi, q)$. In particular, if

$\hat{d}(\xi, q) = \frac{\partial}{\partial q} z(\xi, q)$, then

$$
\begin{aligned}
\hat{d}(\xi, q) &= \frac{\partial}{\partial q} z(\xi, q) \\
&= \frac{\partial}{\partial q} [\hat{w}(\xi, q) - 4\xi] \\
&= \frac{\partial}{\partial q} w(M(\xi, q), q) \\
&= \frac{\partial}{\partial q} w(M(\xi, q), q) + \frac{\partial}{\partial x} w(M(\xi, q), q) \frac{\partial}{\partial q} M(\xi, q) \\
&= \hat{s}(\xi, q) + \frac{\partial}{\partial x} w(M(\xi, q), q) \frac{\partial}{\partial q} M(\xi, q).
\end{aligned}
$$

Consequently, the difference between the transformed sensitivity $\hat{s}(\xi)$ and the sensitivity of the transformed state $\hat{d}(\xi)$ is given by the difference

$$
\hat{d}(\xi, q) - \hat{s}(\xi, q) = \frac{\partial}{\partial x} w(M(\xi, q), q) \frac{\partial}{\partial q} M(\xi, q). \tag{2.17}
$$

The right hand side of (2.17) is the continuous version of the "mesh" gradient and the source of considerable computational complexity. Consequently, one advantage of mapping both the state and the sensitivity equation is that the computation of this gradient can be eliminated.

Finally, we note that once $z(\xi, q)$ and $p(\xi, q)$ are computed on $[0, 1]$, the state $w(x, q)$ and sensitivity $s(x, q)$ can be recovered on $[0, q]$ by

$$
w(x, q) = \hat{w}(T(x, q), q) = z(T(x, q), q) + 4T(x, q), \tag{2.18}
$$

and

$$
s(x, q) = \hat{s}(T(x, q), q) = p(T(x, q), q) - \left[ \frac{\frac{d}{d\xi} z(1, q) + 4}{q} \right] T(x, q), \tag{2.19}
$$

respectively. When applying the inverse transforms to the numerical solutions, there is the possibility that numerical errors can be induced. In particular, the map $T(x, q)$ is often constructed numerically for practical CFD problems. Moreover, in equation (2.19) the presence of the derivative $\frac{\partial}{\partial \xi} z(1, q)$ at the boundary can introduce additional errors. These are practical issues that are important to address in more complex problems. However, observe once again that it is not necessary to compute a mesh gradient to transform back to the physical domain.

## 2.4 Computational Algorithms

We turn now to the issue of approximating the coupled system defined by the state equation

$$\frac{d^2}{d\xi^2} z(\xi) + \frac{1}{8q} \left[ \frac{d}{d\xi} z(\xi) + 4 \right]^3 = 0 \tag{2.20}$$

and sensitivity equation

$$\frac{d^2}{d\xi^2} p(\xi) + \frac{3}{8q} \left[ \frac{d}{d\xi} z(\xi) + 4 \right]^2 \left( \frac{d}{d\xi} p(\xi) - \frac{\frac{d}{d\xi} z(1) + 4}{q} \right) = 0 \tag{2.21}$$

with boundary conditions

$$z(0) = z(1) = 0, \tag{2.22}$$

and

$$p(0) = p(1) = 0, \tag{2.23}$$

respectively. It is important to observe that (2.20)-(2.21) is weakly coupled in the sense that (2.21) does not feed back into (2.20). We take advantage of this structure to develop a family of numerical schemes for computing the sensitivity $p(\xi)$.

Note that for each $\tilde{q} > 1$, $z(\xi, \tilde{q})$ and $p(\xi, \tilde{q})$ exist and belong to $H_0^1(0,1) \bigcap H^2(0,1)$. Let $V = H_0^1(0,1) \times H_0^1(0,1)$ and observe that for each $(\phi(\cdot), \psi(\cdot)) \in V$ the solution pair $(z(\cdot), p(\cdot))$ satisfies the weak system

$$-\left\langle z'(\cdot), \phi'(\cdot) \right\rangle + \frac{1}{8q} \left\langle \left[ z'(\cdot) + 4 \right]^3, \phi(\cdot) \right\rangle = 0 \tag{2.24}$$

$$-\left\langle p'(\cdot), \psi'(\cdot) \right\rangle + \frac{3}{8q} \left\langle \left[ z'(\cdot) + 4 \right]^2 \left( p'(\cdot) - \left[ \frac{z'(1) + 4}{q} \right] \right), \psi(\cdot) \right\rangle = 0, \tag{2.25}$$

where $\langle \cdot, \cdot \rangle$ denotes the $L^2$ inner-product.

## 2.5 A Finite Element Scheme

Although there are several possible choices for finite element spaces, we shall limit our discussion to the simplest (convergent) scheme. We note that in more complex problems

one must choose these spaces with care to ensure the algorithm satisfies the appropriate convergence criteria (inf-sup conditions, etc.). Let $h = \left(\frac{1}{N+1}, \frac{1}{M+1}\right)$ and denote by $V^h$ the product space

$$V^h \triangleq \mathcal{S}_0^N(0,1) \times \mathcal{S}_0^M(0,1) \subseteq V, \tag{2.26}$$

where $\mathcal{S}_0^K(0,1)$ denotes the subspace of $H_0^1(0,1)$ consisting of continuous piecewise linear functions with nodes at $\xi_i = \frac{i}{K+1}, i = 1, 2, \ldots, K$. Let $h_i^N(\cdot)$ and $h_j^M(\cdot)$ denote the standard "hat" functions with nodes $\xi_i$ and $\xi_j$, respectively. Also let

$$z^N(\xi) = \sum_{i=1}^{N} z_i h_i^N(\xi) \tag{2.27}$$

and

$$p^M(\xi) = \sum_{j=1}^{M} p_j h_j^M(\xi) \tag{2.28}$$

be Galerkin approximations of the pair $(z(\cdot), p(\cdot))$ in $V^h$. Consider the Galerkin approximations

$$-\left\langle \frac{d}{d\xi} z^N(\cdot), \frac{d}{d\xi} h_i^N(\cdot) \right\rangle + \frac{1}{8q} \left\langle \left[\frac{d}{d\xi} z^N(\cdot) + 4\right]^3, h_i^N(\cdot) \right\rangle = 0, \tag{2.29}$$

$$-\left\langle \frac{d}{d\xi} p^M(\cdot), \frac{d}{d\xi} h_j^M(\cdot) \right\rangle$$
$$+ \frac{3}{8q} \left\langle \left[\frac{d}{d\xi} z^N(\cdot) + 4\right]^2 \left(\frac{d}{d\xi} p^M(\cdot) - \left[\frac{\frac{d}{d\xi} z^N(1) + 4}{q}\right]\right), h_j^M(\cdot) \right\rangle = 0 \tag{2.30}$$

for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$.

Note that $p^M(\cdot)$ depends on $z^N(\cdot)$ *and* its spatial derivative $\frac{d}{d\xi} z^N(\cdot)$ on $[0,1]$. To emphasize the dependence we let $p^{N,M}(\cdot)$ denote the solution of (2.30), given that $z^N(\cdot)$ obtained from (2.29) is used in (2.30), and let

$$v^h(\xi, q) = \left(z^N(\xi, q), p^{N,M}(\xi, q)\right) \tag{2.31}$$

denote the solution pair. At this point, there are two important observations that play a key role in the construction of accurate numerical sensitivities.

- The freedom to choose separate finite element spaces for the state $z(\xi, q)$ and the sensitivity $p(\xi, q)$ allows for the development of schemes that simultaneously converge to the state and sensitivity. In addition, both h-refinement (mesh refinement) and p-refinement (the selection of higher order elements) can be combined to construct numerical solutions of $v^h(\xi, q) = \left(z^N(\xi, q), p^{N,M}(\xi, q)\right)$ so that the error in $p^{N,M}(\xi, q)$ is sufficiently small to ensure convergence of optimal design algorithms based on the SEM (see [3], [7], [5]).

- The solution $p^{N,M}(\xi)$ depends not only on $z^N(\xi)$ but also its derivative $\frac{d}{d\xi}z^N(\xi)$. Moreover, since $z^N(\xi)$ is piecewise linear, $\frac{d}{d\xi}z^N(\xi)$ is a piecewise constant (discontinuous) function. However, the actual transformed sensitivity $p(\xi)$ is smooth, and one might expect to lose at least one order of accuracy in $p^{N,M}(\xi)$. In fact, things can be much worse unless special care is exercised.

There are two obvious "fixes" to address these issues. One could use higher order splines for the sensitivity variable $p(\xi)$. However, this method will be more expensive, and it is not reasonable to expect great improvements unless higher order schemes are also used for the state equation. The other obvious fix is to use mesh refinement in M (assuming accuracy in N). There is a third approach that makes use of "smoothing projections." The idea is similar to the method used to obtain a-posteriori error estimators for adaptive mesh generation (see [8], [24], [41], [42]). This approach will be outlined in Chapter 4 and applied to the model problem and to a 2-D fluid flow problem.

## 2.6    Numerical Results

We use the FEM scheme outlined above to construct approximations to the state and the state sensitivities. We also evaluate the use of the state sensitivities in an optimization algorithm to solve the inverse design problem outlined in Section 2.1.

### 2.6.1    Convergence of Solutions for the Boundary Value Problem and Sensitivity Equation

In this section, we compare the finite element approximations of the solutions to the state and sensitivity equations with their exact solutions. First, we note that the finite element scheme converges to the exact solution of the nonlinear problem (for each $q > 1$). Figure (2.1) shows the finite element approximations to the solution of the boundary value problem
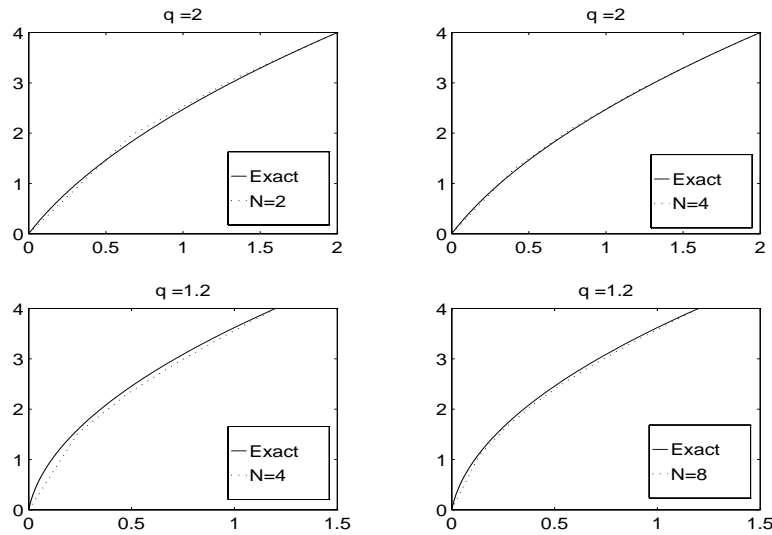
Figure 2.1: Numerical Approximations to the Solution of the Boundary Value Problem at $q = 2$ and $q = 1.2$

at two parameter values: $q = 2$ and $q = 1.2$. Notice that at $q = 2$, the $N = 4$ finite element model provides an excellent match to the exact solution. However, when $q = 1.2$ one sees that a finer mesh ($N = 8$) is required to obtain the same order of accuracy. This convergence is expected because the gradient of the solution becomes singular as $q \to 1^+$ and hence the problem becomes stiff in this parameter region. This is also the case for the sensitivity equation.

Consider the corresponding finite element approximations of the sensitivity equation. Recall that $N$ and $M$ define the meshes for the state and sensitivity equations, respectively. Figure (2.2) shows the finite element approximations for the sensitivity with $q = 2$ and $N = M$ ranging from 2 to 16. Observe in Figure (2.3) that although the finite element scheme produces excellent solutions to the state equation when $N = 4$, the error in the corresponding sensitivity does not diminish until $N = M = 16$.

As mentioned before, one obvious "fix" is to use mesh refinement in M. Figure (2.4) shows the results for this technique when $N = 2$ and $M$ ranges from 2 to 16. Note that improvements in the accuracy of the sensitivity approximation are limited by the accuracy of the state approximation. When we increased $N$ to 4, the sensitivity errors were decreased (see Figure (2.5)).

The stiffness of the problem near $q = 1$ increases the difficulty of getting good sensitivity

Figure 2.2: Numerical Approximations to the Solution of the Sensitivity Equation at $q = 2$ using piecewise constant (PWC) derivatives



Figure 2.3: $L^2$ Error of the Solution and Sensitivity Approximations at $q = 2$
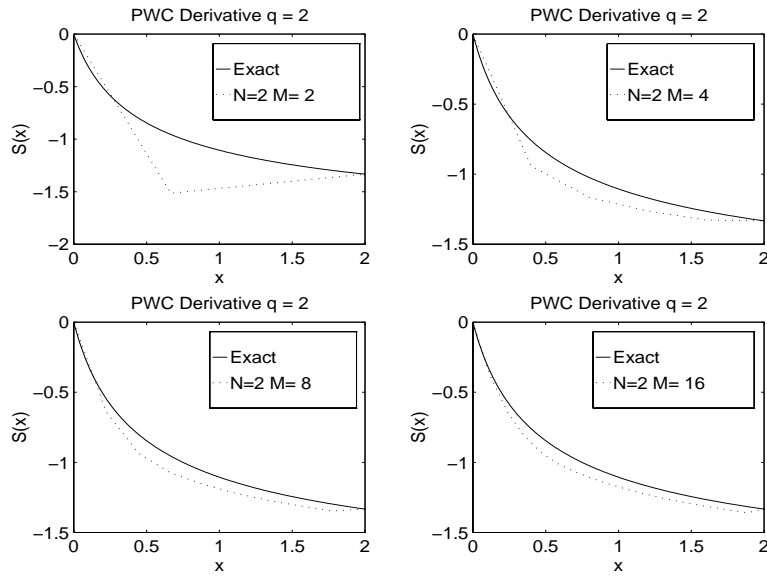
Figure 2.4: Numerical Approximations to the Solution of the Sensitivity Equation at $q = 2$ using PWC derivatives with N = 2 and mesh refinement in M

approximations. Figures (2.6) and (2.7) show that the sensitivity approximations become unreliable as $q \to 1^+$ even though the state approximations are still fairly good. Figure (2.8) displays the difference between the $L^2$ error in the state approximation and the $L^2$ error in the sensitivity approximation. As $N, M$ increases we obtain convergence of the scheme, but the approximations for smaller $N, M$ contain large errors and the convergence of the finite element approximation to the analytical solution is not at all monotone. Figure (2.9) is a graph of the $L^2$ error of the sensitivity approximations for various values of q. We shall observe similar behavior in the next section, where we approximate solutions to 2-D flow problems.
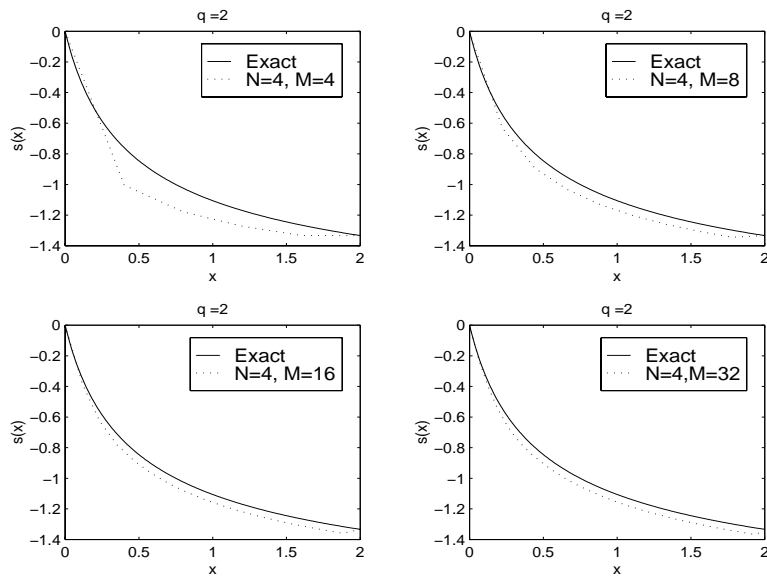
Figure 2.5: Numerical Approximations to the Solution of the Sensitivity Equation at $q = 2$ using PWC derivatives with N = 4 and mesh refinement in M
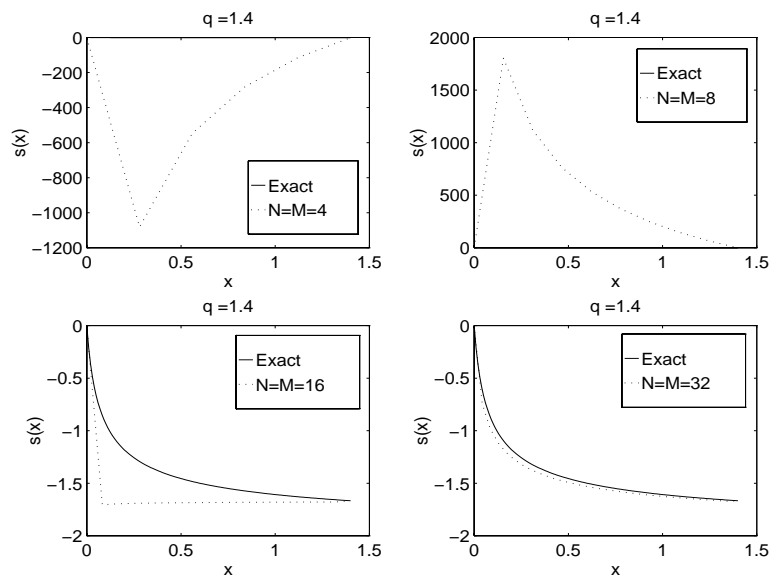


Figure 2.6: Numerical Approximations to the Solution of the Sensitivity Equation at $q = 1.4$ using piecewise constant (PWC) derivatives
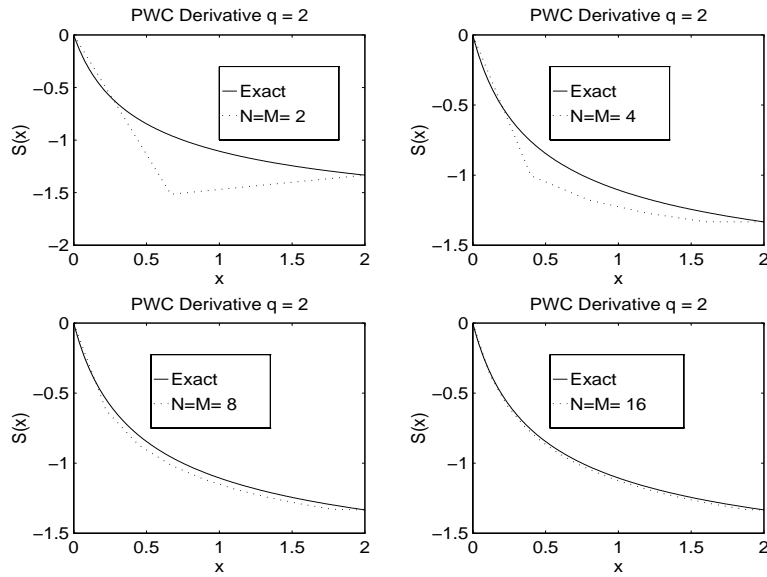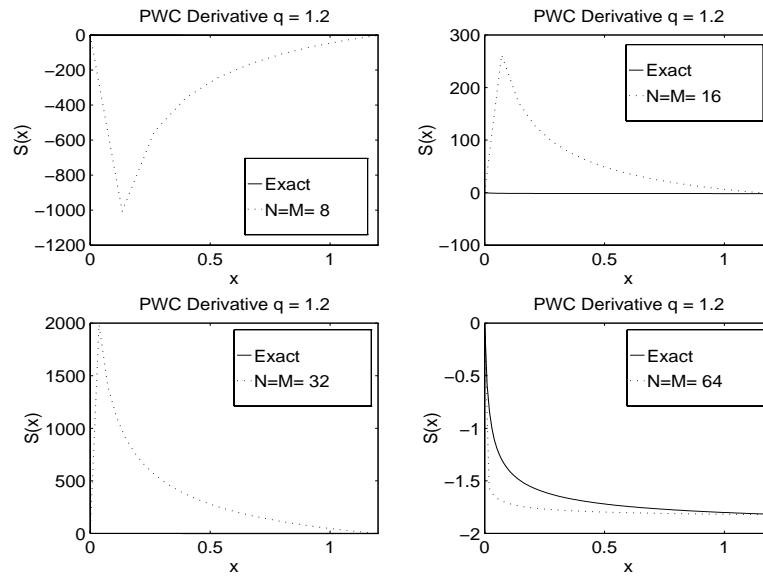
Figure 2.7: Numerical Approximations to the Solution of the Sensitivity Equation at $q = 1.2$ using piecewise constant (PWC) derivatives
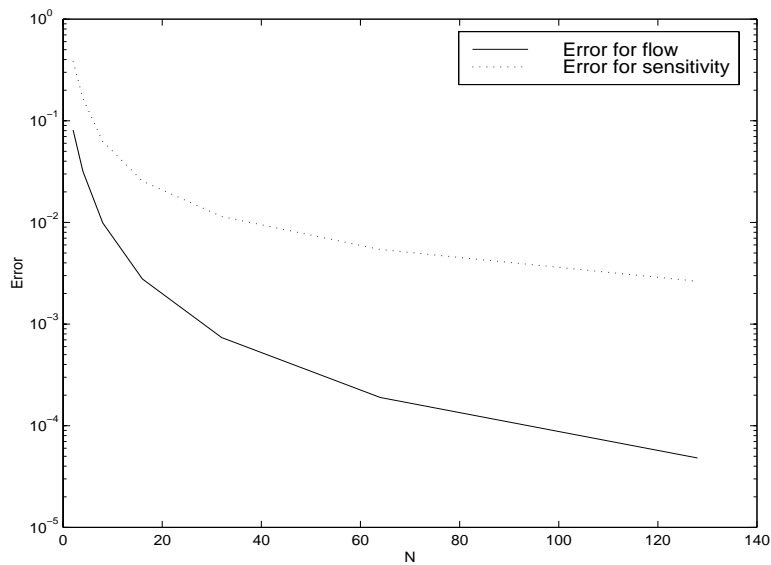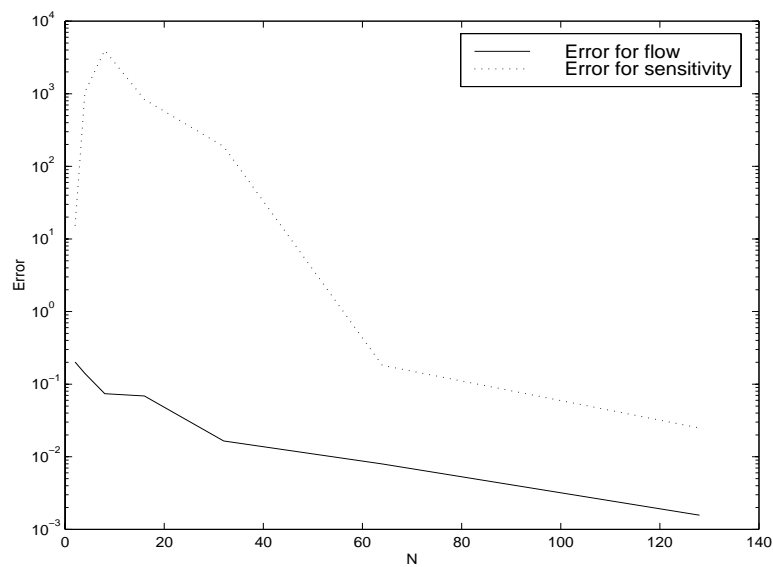


Figure 2.8: $L^2$ Error of the Solution and Sensitivity Approximations at $q = 1.2$
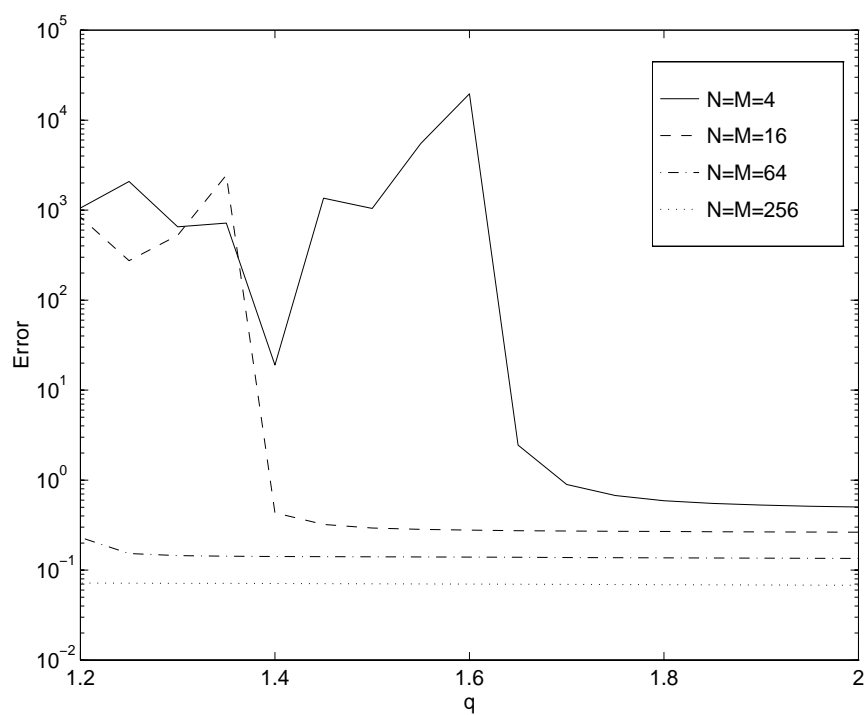
Figure 2.9: $L^2$ Error of Sensitivity Approximations using Piecewise Constant (PWC) Derivatives

## 2.6.2   Optimization Results

We now return to the inverse design problem presented in §2.1 and evaluate the convergence properties of an optimization scheme using the sensitivity equation method. After discretization, the infinite dimensional inverse problem (2.4) becomes: Find $q^* > 1$ such that

$$J_p^N(q^\star) \leq J_p^N(q) \overset{\triangle}{=} \frac{1}{2} \sum_{j=1}^{p} (w^N(\hat{x}_j, q) - \hat{w}_j)^2, \tag{2.32}$$

where $w^N(x, q)$ is obtained using (2.18). Notice that the gradient has the form

$$\nabla_q J_p^N(q) = \sum_{j=1}^{p} (w^N(\hat{x}_j, q) - \hat{w}_j) \frac{\partial}{\partial q} w^N(\hat{x}_j, q). \tag{2.33}$$

The sensitivity equation method applied to the optimization problem replaces $\frac{\partial}{\partial q} w^N(\hat{x}_j, q)$, the sensitivity of the discrete solution, with an approximation to $s(\hat{x}_j, q)$, $s^{N,M}(\hat{x}_j, q)$, from (2.19).

The standard Gauss-Newton algorithm is used to approximate $q^*$ . The algorithm solves a least squares problem at each iteration and proceeds as described in Table 2.1:

Table 2.1: Gauss-Newton Algorithm

Given $q_0$ and $\hat{w}_i$
Set iteration counter: i = 0
Compute $w_i^N = w^N(q_i)$
Compute $s_i^{N,M} = s^{N,M}(q_i)$
While ($i \leq$ max iterations AND $\|\nabla_q J^N(q_i)\| \geq$ tolerance ) Do
      Calculate step $\delta q$
      Set $q_{i+1} = q_i + \delta q$
      Compute $w_{i+1}^N = w^N(q_{i+1})$
      Compute $s_{i+1}^{N,M} = s^{N,M}(q_{i+1})$
      Increment Counter
EndWhile

The "data" to be matched, denoted by $\hat{w}_j$ in (2.4), is indicated by pluses in Figures (2.10) - (2.11). This data set was generated by randomly perturbing the value of $w(\hat{x}_i)$ in (2.3) using $q = 2$ and $q = 1.4$ with $p = 4, 16$ data points. Table 2.2 shows the numerical values of the data for comparison purposes.

Table 2.2: Matching Data for the Optimization

| p=4 | | | | | | |
|---|---|---|---|---|---|---|
| | q=2 | | | q=1.4 | | |
| $\hat{x}_j$ | $w(\hat{x}_j)$ | $\hat{w}_j$ | perturbation | $w(\hat{x}_j)$ | $\hat{w}_j$ | perturbation |
| 0.1250 | 0.4495 | 0.4516 | 0.0021 | 0.8248 | 0.7431 | -0.0817 |
| 0.2500 | 0.8284 | 0.7722 | -0.0563 | 1.3541 | 1.3875 | 0.0335 |
| 0.5000 | 1.4641 | 1.4080 | -0.0561 | 2.1394 | 1.9848 | -0.1546 |
| 0.7500 | 2.0000 | 1.8338 | -0.1662 | 2.7553 | 2.5758 | -0.1795 |
| p=16 | | | | | | |
| | q=2 | | | q=1.4 | | |
| $\hat{x}_j$ | $w(\hat{x}_j)$ | $\hat{w}_j$ | perturbation | $w(\hat{x}_j)$ | $\hat{w}_j$ | perturbation |
| 0.0312 | 0.1213 | 0.1126 | -0.0088 | 0.2677 | 0.2690 | 0.0013 |
| 0.0625 | 0.2361 | 0.2207 | -0.0154 | 0.4806 | 0.4480 | -0.0326 |
| 0.0938 | 0.3452 | 0.3670 | 0.0218 | 0.6629 | 0.6375 | -0.0254 |
| 0.1250 | 0.4495 | 0.4372 | -0.0123 | 0.8248 | 0.7563 | -0.0685 |
| 0.1562 | 0.5495 | 0.5916 | 0.0421 | 0.9720 | 0.9772 | 0.0052 |
| 0.1875 | 0.6458 | 0.6611 | 0.0154 | 1.1079 | 1.0335 | -0.0744 |
| 0.2188 | 0.7386 | 0.7651 | 0.0265 | 1.2347 | 1.2820 | 0.0473 |
| 0.2500 | 0.8284 | 0.8687 | 0.0403 | 1.3541 | 1.4106 | 0.0565 |
| 0.3125 | 1.0000 | 0.9091 | -0.0909 | 1.5749 | 1.4821 | -0.0928 |
| 0.3750 | 1.1623 | 1.2674 | 0.1051 | 1.7768 | 1.6265 | -0.1503 |
| 0.4375 | 1.3166 | 1.2486 | -0.0680 | 1.9641 | 1.9460 | -0.0181 |
| 0.5000 | 1.4641 | 1.6086 | 0.1445 | 2.1394 | 2.0504 | -0.0890 |
| 0.5625 | 1.6056 | 1.6483 | 0.0427 | 2.3048 | 2.0950 | -0.2098 |
| 0.6250 | 1.7417 | 1.9067 | 0.1651 | 2.4619 | 2.3973 | -0.0646 |
| 0.6875 | 1.8730 | 1.9668 | 0.0938 | 2.6117 | 2.8040 | 0.1922 |
| 0.7500 | 2.0000 | 2.0554 | 0.0554 | 2.7553 | 2.9296 | 0.1743 |
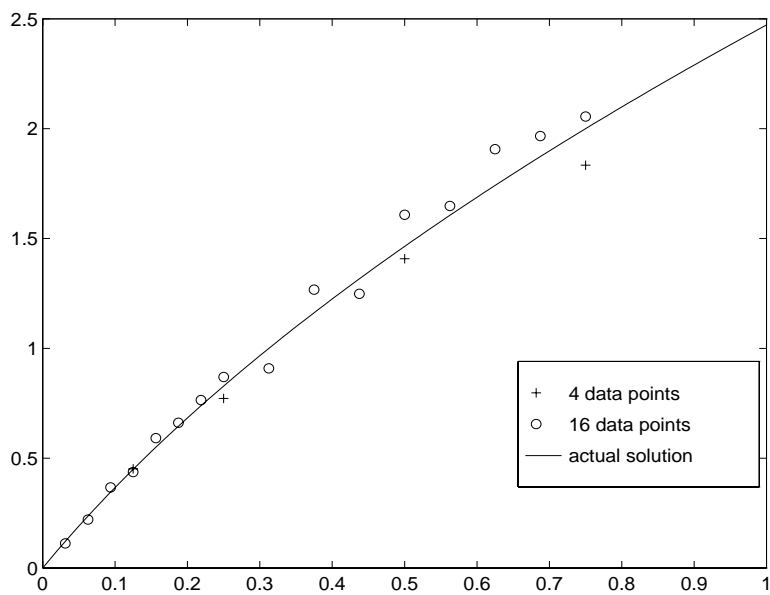
Figure 2.10: Data generated at $q = 2$
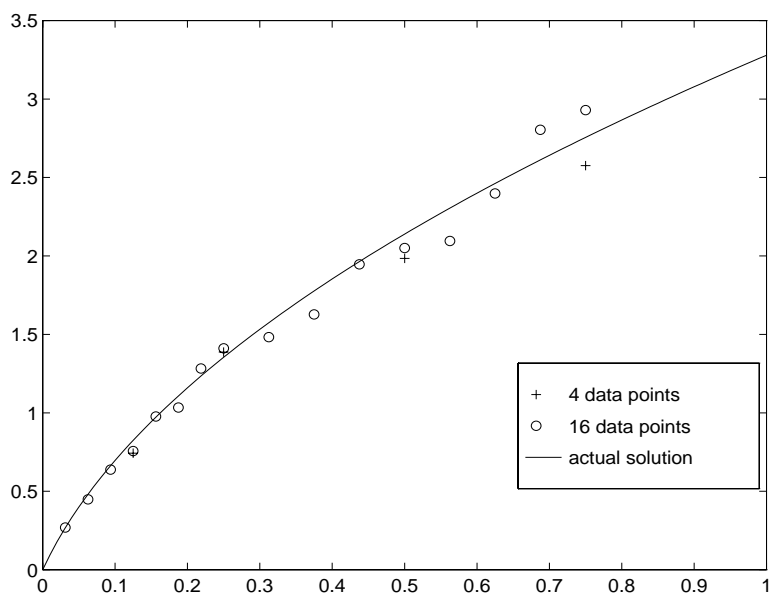


Figure 2.11: Data generated at $q = 1.4$
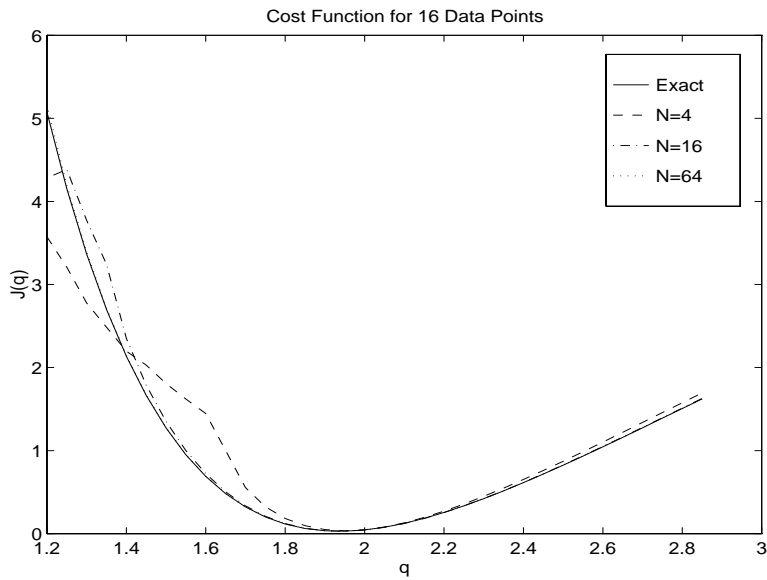
Figure 2.12: The Cost Function and Its Approximations for $p = 16$ and $q^* \sim 2$.
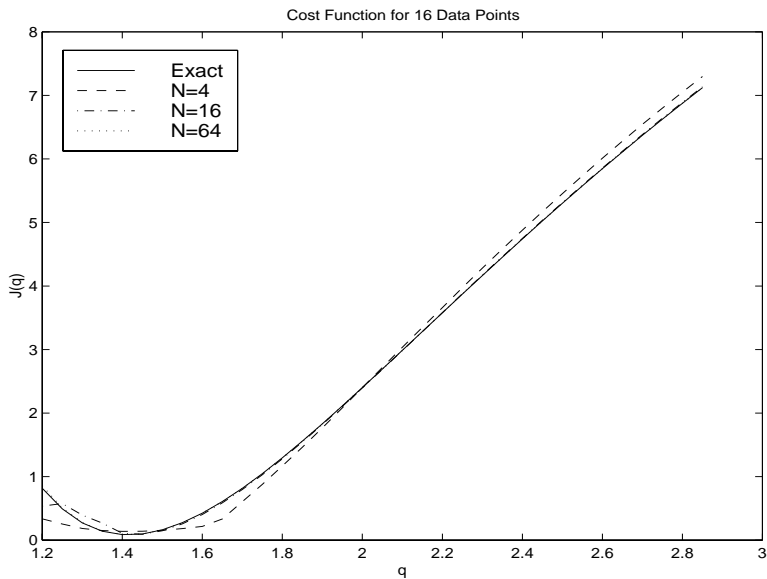


Figure 2.13: The Cost Function and Its Approximations for $p = 16$ and $q^* \sim 1.4$.

The exact cost functional, $J(q)$, and several approximations to it, $J^N(q)$ are plotted in Figures (2.12)-(2.13) for the case where $p = 16$ and $q^* \sim 2$ and $q^* \sim 1.4$, respectively.

Although we made several runs with various data sets, we present the results of two runs.

- Case 1: The solution was calculated for $q = 2.0$ and the noise vector in Table (2.2) was added to obtain data for optimization. Here, the optimal q is approximately $q^* \sim 2$. The optimization algorithm was started at $q_{init} = 1.2$.

- Case 2: In this case, the solution was calculated for $q = 1.4$ and the noise vector in Table (2.2) was again added to obtain data for optimization. Here, the optimal q is approximately $q^* \sim 1.4$. Here, the optimization algorithm was started at $q_{init} = 2.0$.

The scheme was considered converged when the norm of the gradient of the cost functional was less than $10^{-7}$. Notice that the simulations were performed using sensitivities calculated using the natural piecewise constant finite element gradient approximations. Tables (2.3) - (2.4) show the results of these simulations as $N$, $M$ ranged from 2 to 128. The time for the runs was measured in seconds and the runs were performed on a Silicon Graphics Onyx2. Notice, the effect of the bad sensitivities on the convergence of the optimization scheme for Case 1 when $N = M$. As expected, larger values of $N$, $M$ were required for convergence of the optimization algorithm for Case 2.

In Chapter 4, we return to this problem and use smoothing projections to enhance sensitivity computations and convergence. We turn now to the 2-D Navier-Stokes equations and discuss two specific flow problems.

Table 2.3: Optimization Results for $p = 16$, $q_{opt} \sim 2$, $q_{init} = 1.2$, PWC Derivatives

| N | M | CONV/DNC | ITERATIONS | TIME | $\|J^N\|$ | q |
|---|---|---|---|---|---|---|
| 2 | 2 | DNC | 20 | 31.4 | 2.4088 | 1.21993 |
| 2 | 5 | CONV | 17 | 3.1 | 0.32610 | 1.94862 |
| 4 | 4 | DNC | 20 | 68.3 | 2.6252 | 1.15545 |
| 4 | 9 | CONV | 15 | 6.6 | 0.26469 | 1.94795 |
| 8 | 8 | DNC | 20 | 143.1 | 3.1942 | 1.08641 |
| 8 | 17 | CONV | 11 | 7.4 | 0.24332 | 1.94328 |
| 16 | 16 | DNC | 20 | 334.7 | 3.0308 | 1.16909 |
| 16 | 33 | CONV | 12 | 16.0 | 0.24235 | 1.93902 |
| 32 | 32 | DNC | 20 | 916.6 | 3.1796 | 1.18975 |
| 32 | 65 | CONV | 20 | 42.1 | 0.24278 | 1.93771 |
| 64 | 64 | DNC | 20 | 4323.1 | 0.24276 | 1.93477 |
| 64 | 129 | CONV | 12 | 191.9 | 0.242783 | 1.93747 |
| 128 | 128 | CONV | 10 | 252.2 | 0.24285 | 1.93584 |

Table 2.4: Optimization Results for $p = 16$, $q_{opt} \sim 1.4$, $q_{init} = 2.0$, PWC Derivatives

| N | M | CONV/DNC | ITERATIONS | TIME | $\|J^N\|$ | q |
|---|---|---|---|---|---|---|
| 2 | 2 | DNC | 20 | 0.6 | 0.86393 | 1.83807 |
| 2 | 5 | DNC | 20 | 32.5 | 0.72436 | 1.33907 |
| 4 | 4 | DNC | 20 | 63.3 | 0.54447 | 1.48191 |
| 4 | 9 | DNC | 20 | 68.4 | 0.52754 | 1.4419 |
| 8 | 8 | DNC | 20 | 134.9 | 0.60045 | 1.38030 |
| 8 | 17 | DNC | 20 | 61.3 | 0.50601 | 1.47419 |
| 16 | 16 | DNC | 20 | 293.2 | 0.78813 | 1.33031 |
| 16 | 33 | CONV | 18 | 18.8 | 0.41994 | 1.43638 |
| 32 | 32 | CONV | 11 | 6.8 | 0.41001 | 1.41986 |
| 32 | 65 | CONV | 16 | 13.2 | 0.41046 | 1.42342 |
| 64 | 64 | CONV | 13 | 33.7 | 0.40820 | 1.41681 |
| 64 | 129 | CONV | 17 | 58.0 | 0.40858 | 1.42048 |
| 128 | 128 | CONV | 12 | 192.7 | 0.40774 | 1.41605 |

# Chapter 3

# 2-D Flow Problems

We now turn to problems in fluid dynamics. We present the flow equations and derive the sensitivity equations as for the 1-D model problem. An adaptive finite element technique is described and used to solve the state and sensitivity equations. The numerical approximations are presented to investigate the convergence of the adaptive grids.

## 3.1   The Navier-Stokes Equations

We consider steady-state 2-D flow of an incompressible, viscous fluid in a bounded domain $\Omega \subseteq I\!\!R^2$ with a Lipschitz-continuous boundary $\Gamma$. Our discussion and presentation will follow the notation in [19] with the exception that, in most cases, vector notation will be used for ease of presentation. Let $\mathbf{u} = [u(x,y), v(x,y)]^T$ represent the velocity field, and define the stress tensor, $\tau$, by

$$\tau(\mathbf{u}) = \mu \left[ \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right] \tag{3.1}$$

where $\mu$ denotes the viscosity. Kundu, in [31], gives the differential form of the principle of conservation of mass as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{3.2}$$

where $\rho$ is the fluid density. A fluid is incompressible if its density does not change with pressure. The steady incompressible form the continuity equation then becomes

$$\nabla \cdot \mathbf{u} = 0. \tag{3.3}$$

The steady-state statement of conservation of momentum is the Navier-Stokes equation:

$$-\nabla \cdot \tau(\mathbf{u}) + \nabla P + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) = \rho\mathbf{f}, \tag{3.4}$$

where $P$ denotes the ambient pressure and $\mathbf{f}$ represents a density of body forces per unit mass (e.g. gravity). In general, the viscosity $\mu$ is a function of the temperature. Since we neglect temperature differences for the problems we consider herein, $\mu$ is assumed to be constant and can be taken outside the derivative. This, along with the fact that the flow is incompressible, gives the following identity:

$$\nabla \cdot \tau(\mathbf{u}) = \mu \Delta \mathbf{u}. \tag{3.5}$$

As commonly done, we set

$$p = \frac{P}{\rho}, \quad \nu = \frac{\mu}{\rho}, \tag{3.6}$$

where $p$ is the kinematic pressure and $\nu$ is the kinematic viscosity. Then, using (3.5) the system (3.3)-(3.4) becomes

$$\left. \begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ -\nu \Delta \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} &= \mathbf{f} \end{aligned} \right\} \quad \text{in} \quad \Omega. \tag{3.7}$$

It is useful to consider a non-dimensionalized form of the Navier-Stokes equations. To do this, define a length scale $L$, a velocity scale $U$, and a reference pressure $p_0$, for the flow. The dimensionless variables for the flow can be defined as follows:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{L}, \quad \tilde{\mathbf{u}} = \frac{\mathbf{u}}{U}, \text{ and } \quad \tilde{p} = \frac{p - p_0}{\rho U^2}.$$

The non-dimensional Navier-Stokes equations can be written as

$$\left. \begin{aligned} \widetilde{\nabla} \cdot \tilde{\mathbf{u}} &= 0 \\ -\frac{\mu}{\rho U L} \widetilde{\Delta} \tilde{\mathbf{u}} + \widetilde{\nabla} \tilde{p} + \tilde{\mathbf{u}} \cdot \widetilde{\nabla} \tilde{\mathbf{u}} &= \tilde{\mathbf{f}}, \end{aligned} \right\} \quad \text{in} \quad \Omega. \tag{3.8}$$

where $\tilde{\mathbf{f}}$ is the non-dimensionalized $\mathbf{f}$. Dropping the tilde's and defining the Reynolds number, $Re \triangleq \frac{\rho U L}{\mu}$, we have

$$\left. \begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ -\frac{1}{Re} \Delta \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} &= \mathbf{f} \end{aligned} \right\} \quad \text{in} \quad \Omega. \tag{3.9}$$

For ease of notation and to maintain consistency with [19], replace $\frac{1}{Re}$ with $\nu$ to obtain the working version of the Navier-Stokes equation.

$$\left. \begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ -\nu \Delta \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} &= \mathbf{f}. \end{aligned} \right\} \quad \text{in} \quad \Omega. \tag{3.10}$$

## 3.2   The Homogeneous Dirichlet Problem

We begin by considering the case of the homogeneous Dirichlet boundary condition

$$\mathbf{u} = 0 \qquad \text{on} \quad \Gamma. \tag{3.11}$$

In order to discuss existence and uniqueness and to introduce the variational form of the problem, we introduce some standard function spaces as well as some necessary bilinear and trilinear forms.

### 3.2.1   Function Spaces and Notation

As usual, we let $L^2(\Omega)$ denote the space of square integrable functions and $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ denote the $L^2$ inner product and norm, respectively. Let

$$L_0^2(\Omega) = \{q \in L^2(\Omega)\colon \int_\Omega q \, d\Omega = 0\}$$

The Sobolev spaces, $H^1$ and $H_0^1$, are

$$H^1(\Omega) = \{v \in L^2(\Omega)\colon \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in L^2(\Omega)\}, \text{ and}$$

$$H_0^1(\Omega) = \{v \in H^1\colon v|_\Gamma = 0\},$$

with inner product, $\langle \cdot, \cdot \rangle_1$, norm, $\| \cdot \|_1$, and seminorm, $|\cdot|_1$, defined by

$$\langle u, v \rangle_1 = \langle u, v \rangle + \langle \frac{\partial u}{\partial x}, \frac{\partial v}{\partial x} \rangle + \langle \frac{\partial u}{\partial y}, \frac{\partial v}{\partial y} \rangle$$

$$\|u\|_1 = \langle u, u \rangle_1^{1/2}, \text{ and}$$

$$|u|_1 = \left( \langle \frac{\partial u}{\partial x}, \frac{\partial u}{\partial x} \rangle + \langle \frac{\partial u}{\partial y}, \frac{\partial u}{\partial y} \rangle \right)^{1/2},$$

respectively.

Also, denote by $H^{-1}(\Omega)$ the dual space consisting of bounded linear functionals on $H_0^1(\Omega)$. The norm for $H^{-1}(\Omega)$ is given by

$$\|\phi\|_{-1} = \sup_{0 \neq v \in H_0^1(\Omega)} \frac{(\phi, v)}{|v|_1}.$$

Recall the trace theorem (see [10]) which proves the existence of a trace operator $\gamma$ as follows:

**Theorem 1.** *Let $\Omega$ be bounded, and suppose $\Omega$ has a piecewise smooth boundary. In addition, suppose $\Omega$ satisfies the cone condition. Then there exists a bounded linear mapping*

$$\gamma : H^1(\Omega) \to L^2(\Gamma), \qquad \|\gamma(v)\| \leq \|v\|_1, \tag{3.12}$$

*such that $\gamma v = v|_\Gamma$ for all $v \in \mathcal{C}^1(\Omega)$.*

As noted in [39], the range of the trace operator is $H^{1/2}(\Gamma)$. The norm for functions $g$ belonging to $H^{1/2}(\Gamma)$ can be defined by

$$\|g\|_{1/2} = \inf_{\substack{v \in H^1(\Omega), \\ v = g \text{ on } \Gamma}} \|v\|_1.$$

The vector-valued counterparts of these spaces in $I\!\!R^2$ will be denoted by bold-face symbols, i.e.,

$$\mathbf{H}^1(\Omega) \equiv (H^1(\Omega))^2 = \left\{\mathbf{v} : v_i \in H^1(\Omega) \quad \text{for } i = 1, 2\right\},$$

$$\mathbf{H}^{-1}(\Omega) \equiv (H^{-1}(\Omega))^2 = \left\{\mathbf{v} : v_i \in H^{-1}(\Omega) \quad \text{for } i = 1, 2\right\},$$

$$\mathbf{H}^{1/2}(\Gamma) \equiv (H^{1/2}(\Gamma))^2 = \left\{\mathbf{v} : v_i \in H^{1/2}(\Gamma) \quad \text{for } i = 1, 2\right\}, \text{ etc.}$$

The norm for $\mathbf{H}^1(\Omega)$ is defined by

$$\|\mathbf{v}\|_1 = \left(\sum_{i=1}^{2} \|v_i\|_1^2\right)^{1/2}.$$

The divergence free subspace of $\mathbf{H}_0^1(\Omega)$, $\mathbf{Z}_0$, is given by

$$\mathbf{Z}_0 = \{\mathbf{v} \in \mathbf{H}_0^1(\Omega) | \nabla \cdot \mathbf{v} = 0\}.$$

We define the following bilinear form:

$$a_0(\mathbf{u}, \mathbf{v}) = \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} \, d\Omega \qquad \forall \, \mathbf{u}, \mathbf{v} \in \mathbf{H}^1(\Omega)$$

where

$$\nabla \mathbf{u} : \nabla \mathbf{v} = \sum_{i,j=1}^{2} \left(\frac{\partial u_i}{\partial x_j}, \frac{\partial v_i}{\partial x_j}\right).$$

Also, let

$$b(\mathbf{u}, q) = -\int_\Omega q \nabla \cdot \mathbf{u} \, d\Omega \qquad \forall \, \mathbf{u} \in \mathbf{H}^1(\Omega), \, \forall \, q \in L^2(\Omega),$$

and

$$a_1(\mathbf{w}; \mathbf{u}, \mathbf{v}) = \int_\Omega (\mathbf{w} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} \, d\Omega \qquad \forall \, \mathbf{u}, \, \mathbf{v}, \, \mathbf{w} \in \mathbf{H}^1(\Omega).$$

## 3.2.2    Existence and Uniqueness of Solutions to the Variational Form

We present some basic results regarding the existence and uniqueness of solutions to the variational form of the homogeneous Navier-Stokes problem as presented in [19]. To begin, we note that the homogeneous partial differential equation (3.10)-(3.11) can be written in variational form as follows.

**Variational Problem 1.** *Given* $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$, *find a pair* $(\mathbf{u}, p) \in \mathbf{Z}_0 \times L_0^2(\Omega)$ *such that:*

$$a_0(\mathbf{u}, \mathbf{v}) + a_1(\mathbf{u}; \mathbf{u}, \mathbf{v}) + b(\mathbf{u}, p) = \langle \mathbf{f}, \, \mathbf{v} \rangle \qquad \forall \, \mathbf{v} \in \mathbf{H}_0^1(\Omega). \tag{3.13}$$

Recall that the trilinear form, $a_1(\cdot; \cdot, \cdot)$ has some nice properties described in the following lemma.

**Lemma 1.** *Let* $\mathbf{u}, \mathbf{v} \in \mathbf{H}^1(\Omega)$ *and let* $\mathbf{w} \in \mathbf{H}^1(\Omega)$ *with* $\nabla \cdot \mathbf{w} = 0$ *and* $\mathbf{w} \cdot \mathbf{n}|_\Gamma = 0$. *Then, the trilinear form* $a_1(\cdot; \cdot, \cdot)$ *is continuous on* $(H^1(\Omega))^3$ *and satisfies:*

$$a_1(\mathbf{w}; \mathbf{u}, \mathbf{v}) + a_1(\mathbf{w}; \mathbf{v}, \mathbf{u}) = 0, \tag{3.14}$$

$$a_1(\mathbf{w}; \mathbf{v}, \mathbf{v}) = 0. \tag{3.15}$$

Now, the existence result from [19] follows:

**Theorem 2.** *For* $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$, *there exists at least one pair* $(\mathbf{u}, p) \in \mathbf{Z}_0 \times L_0^2(\Omega)$ *which satisfies (3.13).*

In order to discuss the uniqueness of the solutions $(\mathbf{u}, p)$ to (3.13), we introduce the norm of the trilinear form $a_1(\cdot; \cdot, \cdot)$, denoted $\mathcal{N}$, and defined by

$$\mathcal{N} = \sup_{\substack{\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{Z}_0 \\ \mathbf{u}, \mathbf{v}, \mathbf{w} \neq 0}} \frac{a_1(\mathbf{w}; \mathbf{u}, \mathbf{v})}{|\mathbf{u}|_1 \, |\mathbf{v}|_1 \, |\mathbf{w}|_1}. \tag{3.16}$$
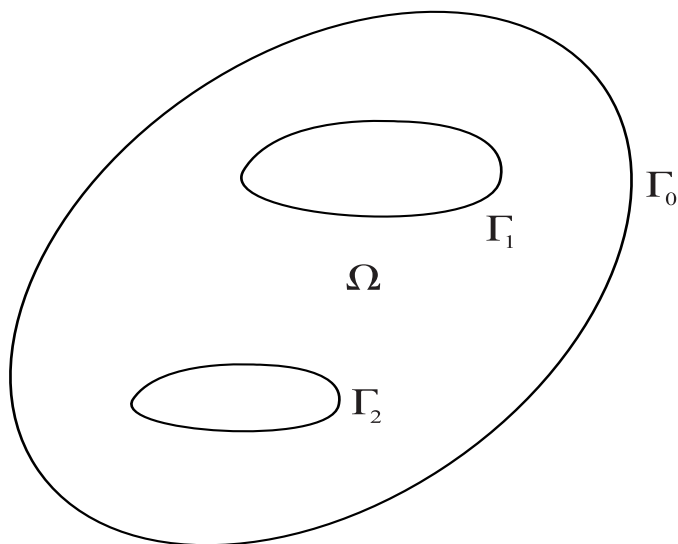
Figure 3.1: Sample Domain with Boundaries

We also set:

$$\|\mathbf{f}\|_{\mathbf{Z}'_0} = \sup_{\substack{\mathbf{v} \in \mathbf{Z}_0 \\ \mathbf{v} \neq 0}} \frac{\langle \mathbf{f}, \mathbf{v} \rangle}{|\mathbf{v}|_1}. \tag{3.17}$$

**Theorem 3.** *If $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$ and*

$$\left( \mathcal{N}/\nu^2 \right) \|\mathbf{f}\|_{\mathbf{Z}'_0} < 1, \tag{3.18}$$

*then the Variational Problem 1 has a unique solution $(\mathbf{u}, p) \in \mathbf{Z}_0 \times L_0^2(\Omega)$.*

## 3.3    The Nonhomogeneous Dirichlet Problem

Now consider the more general case of a nonhomogeneous Dirichlet boundary condition

$$\mathbf{u} = \mathbf{g} \qquad \text{on} \quad \Gamma. \tag{3.19}$$

Denote by $\Gamma_i$, $0 \leq i \leq p$, the connected components of the boundary $\Gamma$ as depicted in Figure (3.1). We shall henceforth assume that

$$\int_{\Gamma_i} \mathbf{g} \cdot \mathbf{n} \, ds = 0, \qquad 0 \leq i \leq p. \tag{3.20}$$

The variational form of the nonhomogeneous partial differential equation (3.10)- (3.19) is obtained using standard weak formulations. In particular, we consider the problem:

**Variational Problem 2.** *Given* $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$, *find a pair* $(\mathbf{u}, p) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$ *such that:*

$$\left. \begin{array}{rll} a_0(\mathbf{u}, \mathbf{v}) + a_1(\mathbf{u}; \mathbf{u}, \mathbf{v}) + b(\mathbf{u}, p) & = \langle \mathbf{f}, \mathbf{v} \rangle & \forall \ \mathbf{v} \in \mathbf{H}_0^1(\Omega) \\ \nabla \cdot \mathbf{u} & = 0 & in \quad \Omega \\ \mathbf{u} & = \mathbf{g} & on \quad \Gamma. \end{array} \right\} \tag{3.21}$$

In order to prove existence for the nonhomogeneous problem, we need the following technical result due to Hopf (see Lemma 2.3, page 287 in [19]).

**Lemma 2.** *Let* $\mathbf{g} \in \mathbf{H}^{\frac{1}{2}}(\Gamma)$ *satisfy (3.20). For any* $\varepsilon > 0$, *there exists a function* $\mathbf{u}_0 = \mathbf{u}_0(\varepsilon) \in \mathbf{H}^1(\Omega)$ *such that*

$$\nabla \cdot \mathbf{u}_0 = 0, \qquad \mathbf{u}_0|_\Gamma = \mathbf{g}, \tag{3.22}$$

$$|a_1(\mathbf{v}; \mathbf{u}_0, \mathbf{v})| \le \varepsilon \, |\mathbf{v}|_1^2 \qquad \forall \ \mathbf{v} \in \mathbf{Z}_0 \tag{3.23}$$

The following existence theorem may be found in [19].

**Theorem 4.** *Let* $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$ *and* $\mathbf{g} \in \mathbf{H}^{\frac{1}{2}}(\Gamma)$ *satisfying (3.20). There exists at least one pair* $(\mathbf{u}, p) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$ *which is a solution of (3.21).*

Before stating the uniqueness result again, we make a few definitions. For any function $\mathbf{u}_0 \in \mathbf{H}^1(\Omega)$, define

$$\rho(\mathbf{u}_0) = \sup_{\substack{\mathbf{v} \in \mathbf{Z}_0 \\ \mathbf{v} \neq 0}} \frac{a_1(\mathbf{v}; \mathbf{u}_0, \mathbf{v})}{|\mathbf{v}|_1^2}, \tag{3.24}$$

and

$$\|l(\mathbf{f}; \mathbf{u}_0)\|_{\mathbf{Z}_0'} = \sup_{\substack{\mathbf{v} \in \mathbf{Z}_0 \\ \mathbf{v} \neq 0}} \frac{\langle l, \mathbf{v} \rangle}{|\mathbf{v}|_1}, \tag{3.25}$$

where

$$\langle l, \mathbf{v} \rangle = \langle \mathbf{f}, \mathbf{v} \rangle - a_0(\mathbf{u}_0, \mathbf{v}) - a_1(\mathbf{u}_0; \mathbf{u}_0, \mathbf{v}). \tag{3.26}$$

Define $\nu_0 = \nu_0(\Omega, \mathbf{f}, \mathbf{g})$ as in [19] by

$$\nu_0 = \inf \left\{ \rho(\mathbf{u}_0) + \left( \mathcal{N} \|l(\mathbf{f}; \mathbf{u}_0)\|_{\mathbf{Z}_0'} \right)^{\frac{1}{2}}; \mathbf{u}_0 \in \mathbf{H}^1(\Omega) \text{ satisfies } (3.22) \right\}. \tag{3.27}$$

The basic uniqueness result for (3.21) is found in [19]. We state it below for convenience.

**Theorem 5.** *Assume the hypothesis of Theorem 4. If $\nu > \nu_0(\Omega, \mathbf{f}, \mathbf{g})$, then the Variational Problem 2 has a unique solution $(\mathbf{u}, p) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$.*

**REMARK:** Although the results above provide existence and uniqueness for the basic Navier-Stokes problems, they do not address the continuity and differentiability of these solutions with respect to the parameter $\mathbf{q}$. For example, in the problems considered below we have $\Omega = \Omega(\mathbf{q})$, $\mathbf{f} = \mathbf{f}(\mathbf{q})$, and/or $\mathbf{g} = \mathbf{g}(\mathbf{q})$ so that $\nu_0 = \nu_0(\Omega, \mathbf{f}, \mathbf{g}) = \nu_0(\Omega, \mathbf{f}, \mathbf{g}, \mathbf{q})$, where $\mathbf{q} \in I\!\!R^n$ is some parameter defining the flow. We need to establish the smoothness of these mappings in order to address the existence and uniqueness of solutions to the sensitivity equations. This is the subject of the following sections.

## 3.4    An Abstract Framework for Navier-Stokes

In this section, we present an abstract framework for analyzing the dependence on $\mathbf{q}$ of solutions to the nonhomogeneous Dirichlet problem for the Navier-Stokes equations. We will show the continuity of solutions with respect to parameters for two specific cases and we will conclude with results about the differentiability of those solutions. We extend the framework in [19] to certain parameter dependent flows.

### 3.4.1    The Framework

Let $X$ and $\mathcal{X}$ be two Banach spaces and $\mathcal{Q} \subseteq \Lambda \subseteq I\!\!R^n$, where $\mathcal{Q}$ is open and $\Lambda$ is compact. Given a $\mathcal{C}^p$-mapping $(p \geq 1)$

$$F : (q, u) \in \Lambda \times X \to F(q, u) \in \mathcal{X}, \tag{3.28}$$

we are interested in solutions to the state equation

$$F(q, u) = 0. \tag{3.29}$$

Let $\{(q, u(q)); q \in \Lambda\}$ be a branch of solutions of equation (3.29). This means that

$$q \to u(q) \text{ is a continuous function from } \Lambda \text{ into } X, and \tag{3.30}$$

$$F(q, u(q)) = 0. \tag{3.31}$$

Moreover, we suppose that these solutions are *nonsingular* in the sense that:

$$D_u F(q, u(q)) \text{ is an isomorphism from } X \text{ onto } \mathcal{X}, \ \forall q \in \Lambda. \tag{3.32}$$

As an immediate consequence of (3.32), it follows from the Implicit Function Theorem (see e.g. [40]) that $q \to u(q)$ is a $\mathcal{C}^p$-function from $\Lambda$ into $X$.

## 3.4.2   Using the Framework

We now show that the parameter-dependent Dirichlet problem for the Navier-Stokes equations in the velocity-pressure formulation (3.10) - (3.19) fits into this abstract framework. We assume that any/all of the following hold:

$$\mathbf{g} = \mathbf{g}(\mathbf{q}), \qquad \text{and/or} \tag{3.33}$$
$$\mathbf{f} = \mathbf{f}(\mathbf{q}) \tag{3.34}$$

where $\mathbf{q} \in \mathcal{Q} \subseteq \Lambda \subseteq I\!\!R^n$ is a design parameter for the flow. Define

$$X = \mathcal{X} = \mathbf{H}^1(\Omega) \times L_0^2(\Omega), \tag{3.35}$$

and the intermediate space

$$Y = \mathbf{H}^{-1}(\Omega) \times \mathcal{V}, \tag{3.36}$$

where $\mathcal{V} = \left\{ \mathbf{g} \in \mathbf{H}^{1/2}(\Gamma); \int_{\Gamma_i} \mathbf{g} \cdot \mathbf{n} \, ds = 0, 0 \leq i \leq p \right\}$. Next we define a linear operator $T$ as follows: given $(\mathbf{f}_*, \mathbf{g}_*) \in Y$, we denote by $(\mathbf{u}_*, p_*) = T(\mathbf{f}_*, \mathbf{g}_*) \in X$ the solution of the Dirichlet problem for the Stokes equations:

$$\left. \begin{array}{rcll} \nabla \cdot \mathbf{u}_* & = 0 & \text{in} & \Omega \\ -\Delta \mathbf{u}_* + \nabla p_* & = \mathbf{f}_* & \text{in} & \Omega \\ \mathbf{u}_* & = \mathbf{g}_* & \text{on} & \Gamma \end{array} \right\}. \tag{3.37}$$

In addition, let $P : \mathcal{Q} \to Y$ be defined by

$$P(\hat{\mathbf{q}}) = \left( \frac{1}{\nu} \mathbf{f}(\hat{\mathbf{q}}), \mathbf{g}(\hat{\mathbf{q}}) \right), \tag{3.38}$$

and the non-linear operator $\mathcal{NL} : X \to Y$ be given by

$$\mathcal{NL}(v = (\mathbf{v}, p)) = \left( \frac{1}{\nu} \left( v_1 \frac{\partial \mathbf{v}}{\partial x} + v_2 \frac{\partial \mathbf{v}}{\partial y} \right), 0 \right), \tag{3.39}$$

where $\nu$ is the constant $\frac{1}{Re}$ as before.

Now finally, with the data $(\mathbf{f}, \mathbf{g}) \in Y$ we associate a $\mathcal{C}^1$-mapping $G$ from $\mathcal{Q} \times X$ into $Y$ defined by

$$G : (\mathbf{q}, v) \to G(\mathbf{q}, v) = \mathcal{NL}(v) - P(\mathbf{q}) \qquad (3.40)$$

and we set

$$F(\mathbf{q}, v) = v + TG(\mathbf{q}, v). \qquad (3.41)$$

The following result follows directly from Lemma 3.1 in [19].

**Lemma 3.** *The pair* $(\mathbf{u}(\mathbf{q}), p(\mathbf{q})) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$ *is a solution of (3.10) - (3.19) if and only if* $(\mathbf{q}, u(\mathbf{q}))$ *is a solution of (3.29) where* $u(\mathbf{q}) = (\mathbf{u}(\mathbf{q}), p(\mathbf{q})/\nu)$ *and where the spaces* $X$ *and* $\mathcal{X}$ *are defined by (3.35) and the compound mapping, $F$, is defined by (3.41).*

### 3.4.3 Continuity of Solutions with Respect to Data

We now address the continuity of solutions $(\mathbf{u}(\mathbf{q}), p(\mathbf{q}))$ to (3.10)-(3.19) with respect to changes in the parameter $\mathbf{q}$. We assume the map $\mathbf{q} \in \mathcal{Q} \to (\mathbf{g}(\mathbf{q}), \mathbf{f}(\mathbf{q})) \in \mathcal{V} \times \mathbf{H}^{-1}(\Omega)$ is continuously Fréchet differentiable. Note, that this is certainly true for the cylinder problem presented in §3.7.1.

To begin, we need Lemma I.3.2 from [19].

**Lemma 4.** *There exists a continuous linear function* $\mathcal{D} : \mathcal{V} \to \mathbf{H}^1(\Omega)$ *such that for each* $\mathbf{g} \in \mathcal{V}$, *we have* $\mathbf{w} = \mathcal{D}(\mathbf{g})$ *satisfies*

$$\left. \begin{array}{c} \nabla \cdot \mathbf{w} = 0, \text{ and } \ \mathbf{w}|_\Gamma = \mathbf{g} \\ \|\mathbf{w}\|_1 \leq C\|\mathbf{g}\|_{1/2} \end{array} \right\}. \qquad (3.42)$$

The following corollary is a direct consequence of Lemma 4.

**Corollary 1.** *The map from* $\mathbf{q} \in \mathcal{Q} \to \mathcal{D}(\mathbf{g}(\mathbf{q})) \in \mathbf{H}^1(\Omega)$ *is Fréchet differentiable.*

In order to analyze the parameter-dependent solution to the weak form of the nonhomogeneous Navier-Stokes problem, we need a result analogous to Lemma 2 for parameter-dependent boundary functions. The following result may be established by a straight forward extension of Lemma 2.3 in [19].

**Lemma 5.** *There exists a continuous linear function $\mathcal{F} : \Lambda \times I\!\!R^+ \to \mathbf{H}^1(\Omega)$ such that for each $\mathbf{q} \in \Lambda,\ \varepsilon > 0$*

$$\mathcal{F}(\mathbf{q}, \varepsilon) = \mathbf{u}_0(\mathbf{q}) \tag{3.43}$$

*where $\mathbf{u}_0(\mathbf{q})$ is defined by Lemma 2 and satisfies*

$$\nabla \cdot \mathbf{u}_0(\mathbf{q}) = 0, \qquad \mathbf{u}_0(\mathbf{q})|_\Gamma = \mathbf{g}(\mathbf{q}), \tag{3.44}$$

*and*

$$|a_1(\mathbf{v}; \mathbf{u}_0(\mathbf{q}), \mathbf{v})| \leq \varepsilon |\mathbf{v}|_1^2 \tag{3.45}$$

*for each $\mathbf{v} \in \mathbf{Z}_0$.*

We now consider continuity with respect to the right hand side function $\mathbf{f}$. We again consider an abstract framework for the Navier-Stokes equations. We define a map $\mathcal{G} : \overline{\mathcal{X}} \times \overline{\mathcal{Y}} \to \overline{\mathcal{Z}}$. Here, $\overline{\mathcal{X}}$ is the set of forcing functions, $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$, and $\overline{\mathcal{Y}}$ is defined as follows:

$$\overline{\mathcal{Y}} = \left\{ (\mathbf{u}, p) : \mathbf{u} \in \mathbf{H}_0^1(\Omega), \text{ and } p \in H^1(\Omega)/I\!\!R \right\}. \tag{3.46}$$

The range of $\mathcal{G}$ is contained in

$$\overline{\mathcal{Z}} = \mathbf{H}^{-1}(\Omega) \times L_0^2(\Omega) \tag{3.47}$$

and $\mathcal{G}$ is defined by

$$\mathcal{G}(x = \mathbf{f}, y = (\mathbf{u}, p)) = \left( -\nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathbf{f}, \nabla \cdot \mathbf{u} \right). \tag{3.48}$$

Note that $\mathcal{G} \in \mathcal{C}^1$ and the Fréchet derivative at a point $(x_0 = \mathbf{f}_0, y_0 = (\mathbf{u}_0, p_0))$ is given by

$$[D\mathcal{G}(x_0, y_0)](x, y) = \left[ \begin{array}{c} -\mathbf{f} + (\mathbf{u}_0 \cdot \nabla)\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}_0 + \nabla p - \nu \Delta \mathbf{u} \\ \nabla \cdot \mathbf{u} \end{array} \right]. \tag{3.49}$$

It is also clear that $(\mathbf{u}, p)$ satisfies the homogeneous Navier-Stokes equations, (3.10) - (3.11), with right hand side $\mathbf{f}$ if and only if $\mathcal{G}(x = \mathbf{f}, y = (\mathbf{u}, p)) = 0$.

The Implicit Function Theorem implies that $(\mathbf{u}, p)$ is a continuously differentiable function of $\mathbf{f}$ if the linear map $[D_y \mathcal{G}(x_0, y_0)]$ is an isomorphism. But this is equivalent to the condition that the homogeneous sensitivity equation has a unique solution in $\overline{\mathcal{Y}}$ for each $\mathbf{f} \in H^{-1}(\Omega)$. We show in the following that the sensitivity equation does indeed have a unique solution in $\overline{\mathcal{Y}}$, so that we have $(\mathbf{u}, p)$ is a $\mathcal{C}^1$ function of $\mathbf{f}$. We will then return to the smoothness of solutions to the parameter-dependent Navier-Stokes equations.

## 3.5    Analysis of the Sensitivity Equations

We begin by stating a general form of the sensitivity equations for the parameter-dependent Navier-Stokes equations. We show that if we have a unique solution $(\mathbf{u}, p)$ of the Navier-Stokes problem, then we have a unique solution $(\mathbf{s}, r)$ of our sensitivity equation. Lastly, we will use an abstract formulation of the Navier-Stokes problem and the implicit function theorem to show that the solution $(\mathbf{u}, p)$ is in fact a nonsingular solution of the Navier-Stokes problem.

### 3.5.1    A General Formulation of the Sensitivity Equations

Let $\hat{q} \in I\!\!R^+$ represent some fixed sensitivity parameter $q_i, (1 \leq i \leq N)$. Now denote the sensitivity forcing function, $\frac{\partial \mathbf{f}}{\partial q_i}(\hat{q})$, by $\mathbf{f}_q^*$ and the boundary function, $\frac{\partial \mathbf{g}}{\partial q_i}(\hat{q})$, by $\mathbf{g}_q^*$. Lastly, recall that in some cases $\Omega = \Omega(\hat{q})$. Then as in the case of the 1-D model problem, the sensitivity equations are obtained by implicitly differentiating the flow equations and their associated boundary conditions. The sensitivity equations fit into the following general form. Given $\mathbf{f}_q^* \in \mathbf{H}^{-1}(\Omega)$ and $\mathbf{g}_q^* \in \mathbf{H}^{\frac{1}{2}}(\Gamma)$ satisfying (3.20) and $(\mathbf{u}, p)$ a solution of (3.10)-(3.19), find a pair $(\mathbf{s}, r) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$ such that $(\mathbf{s}, r)$ satisfies

$$\left.\begin{array}{rcll} \nabla \cdot \mathbf{s} & = 0 & \text{in} & \Omega \\ -\nu \Delta \mathbf{s} + \nabla r + \mathbf{u} \cdot \nabla \mathbf{s} + \mathbf{s} \cdot \nabla \mathbf{u} & = \mathbf{f}_q^* & \text{in} & \Omega \\ \mathbf{s} & = \mathbf{g}_q^* & \text{on} & \Gamma \end{array}\right\}. \qquad (3.50)$$

We can write the variational form of (3.50) as:

**Variational Problem 3.** *Given* $\mathbf{f}_q^* \in \mathbf{H}^{-1}(\Omega)$ *and* $\mathbf{g}_q^* \in \mathbf{H}^{\frac{1}{2}}(\Gamma)$ *satisfying (3.20) and* $(\mathbf{u}, p)$ *a solution (3.10)-(3.19), find a pair* $(\mathbf{s}, r) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$ *such that:*

$$\left.\begin{array}{rcll} a_0(\mathbf{s}, \mathbf{z}) + a_1(\mathbf{u}; \mathbf{s}, \mathbf{z}) + a_1(\mathbf{s}; \mathbf{u}, \mathbf{z}) + b(\mathbf{z}, r) & = \langle \mathbf{f}_q^*, \mathbf{z} \rangle & & \\ \nabla \cdot \mathbf{s} & = 0 & in & \Omega \\ \mathbf{s} & = \mathbf{g}_q^* & on & \Gamma \end{array}\right\}, \qquad (3.51)$$

$\forall \mathbf{z} \in \mathbf{H}_0^1(\Omega)$.

### 3.5.2    Existence and Uniqueness of Solutions to the Sensitivity Equations

We state and prove the following existence and uniqueness result following [19].

**Theorem 6.** *Assume the hypotheses of Theorem 5. If $(\mathbf{u}, p)$ is a unique solution of the Variational Problem 2, then there exists a unique solution, $(\mathbf{s}, r)$, to Variational Problem 3.*

*Proof.* In order to check that the equations (3.51) have a unique solution, it is sufficient to prove that the bilinear form

$$(\mathbf{s}, \mathbf{z}) \rightarrow c(\mathbf{s}, \mathbf{z}) = a_0(\mathbf{s}, \mathbf{z}) + a_1(\mathbf{u}; \mathbf{s}, \mathbf{z}) + a_1(\mathbf{s}; \mathbf{u}, \mathbf{z}) \tag{3.52}$$

is V-elliptic. But it follows from (3.15) that

$$c(\mathbf{s}, \mathbf{s}) = \nu \, |\mathbf{s}|_1^2 + a_1(\mathbf{s}; \mathbf{u}, \mathbf{s}) \qquad \forall \mathbf{s} \in \mathbf{Z}_0. \tag{3.53}$$

Now, assume that $\nu > \nu_0$ where $\nu_0$ is defined by (3.27). Then, there exists a function $\mathbf{u}_0 \in \mathbf{H}^1(\Omega)$ such that

$$\nabla \cdot \mathbf{u}_0 = 0, \qquad \mathbf{u}_0|_\Gamma = \mathbf{g}^*$$

$$\nu > \rho(\mathbf{u}_0) + \left( \mathcal{N} \| l(\mathbf{f}^*; \mathbf{u}_0) \|_{\mathbf{Z}_0'} \right)^{\frac{1}{2}}.$$

Next, setting $\mathbf{u} = \mathbf{u}_0 + \mathbf{w}$, we have by (3.16) and (3.24)

$$\begin{aligned}
|a_1(\mathbf{s}; \mathbf{u}, \mathbf{s})| &\leq |a_1(\mathbf{s}; \mathbf{u}_0, \mathbf{s})| + |a_1(\mathbf{s}; \mathbf{w}, \mathbf{s})| \\
&\leq (\rho(\mathbf{u}_0) + \mathcal{N} \, |\mathbf{w}|_1) \, |\mathbf{s}|_1^2 .
\end{aligned}$$

Since

$$|\mathbf{w}|_1 \leq \frac{\| l(\mathbf{f}^*; \mathbf{u}_0) \|_{\mathbf{Z}_0'}}{\nu - \rho(\mathbf{u}_0)}, \tag{3.54}$$

we obtain

$$\begin{aligned}
c(\mathbf{s}, \mathbf{s}) &\geq \left( \nu - \rho(\mathbf{u}_0) - \frac{\mathcal{N} \| l(\mathbf{f}^*; \mathbf{u}_0) \|_{\mathbf{Z}_0'}}{\nu - \rho(\mathbf{u}_0)} \right) |\mathbf{s}|_1^2 \\
&= (\nu - \rho(\mathbf{u}_0)) \left( 1 - \frac{\mathcal{N} \| l(\mathbf{f}^*; \mathbf{u}_0) \|_{\mathbf{Z}_0'}}{(\nu - \rho(\mathbf{u}_0))^2} \right) |\mathbf{s}|_1^2
\end{aligned}$$

so that the ellipticity property holds.

## 3.6 Differentiability of Solutions with Respect to $q$

We return now to the smoothness of solutions to the parameter-dependent Navier-Stokes equations.

Note that the parameter $\nu_0 = \nu_0(\mathbf{q})$ is a continuous function of $\mathbf{q}$ in this case since the map from $\mathbf{q} \to \mathbf{u}_0$ is continuous and the map from $\mathbf{u}_0 \to \rho$ is continuous. With this, the continuity of the map $(\mathbf{f}, \mathbf{g}) \to (\mathbf{u}, p)$, and the fact that $\mathbf{q} \to (\mathbf{f}(\mathbf{q}), \mathbf{g}(\mathbf{q}))$ is $\mathcal{C}^1$, we can now establish the following result.

**Theorem 7.** *Assume that $\hat{\mathbf{q}} \in \mathcal{Q}$ is such that $\nu > \nu_0(\hat{\mathbf{q}})$. There exists a neighborhood, $\widehat{\mathcal{Q}}$ of $\hat{\mathbf{q}}$, $\hat{\mathbf{q}} \in \widehat{\mathcal{Q}} \subseteq \mathcal{Q} \subseteq \Lambda$ such that for all $\mathbf{q} \in \widehat{\mathcal{Q}}$ the solution of the variational, parameter-dependent Navier-Stokes equations (3.21), with $\mathbf{f} = \mathbf{f}(\mathbf{q})$ and $\mathbf{g} = \mathbf{g}(\mathbf{q})$ exists and is unique. Moreover, the solution is a $\mathcal{C}^1$ function of $\mathbf{q}$.*

**Proof:** We consider the map $F$ defined by (3.41). We have shown that there exists $\hat{\mathcal{Q}} \subset I\!\!R^n$ so that we have a branch of solutions $\left\{ (\mathbf{q}, u(\mathbf{q})); \mathbf{q} \in \hat{\mathcal{Q}} \right\}$, i.e. that both (3.30) and (3.31) hold for $\mathbf{q} \in \hat{\mathcal{Q}}$, for both cases presented in §3.4.3 above.

We now turn our attention to the Fréchet derivative of $F$, $D_u F$. We have

$$[D_u F(\overline{q}, \overline{u})](q, u) = u + T [D_u G(\overline{q}, \overline{u})](q, u) \tag{3.55}$$

$$= u + T [D_u \mathcal{NL}(\overline{q}, \overline{u})](u) \tag{3.56}$$

$$= u + T \left( \frac{1}{\nu} \left( \overline{u}_1 \frac{\partial \mathbf{u}}{\partial x} + \overline{u}_2 \frac{\partial \mathbf{u}}{\partial y} + u_1 \frac{\partial \overline{u}}{\partial x} + u_2 \frac{\partial \overline{u}}{\partial y} \right), 0 \right). \tag{3.57}$$

Again, the fact that $D_u F$ is an isomorphism can be shown to be equivalent to the fact that the homogeneous, sensitivity equation has a unique solution for all $\mathbf{q} \in \hat{\mathcal{Q}}$. We have shown in §3.5.2 that the sensitivity equation does indeed have a unique solution. Then, by the Implicit Function Theorem, we have that $\mathbf{q} \to (\mathbf{u}(\mathbf{q}), p(\mathbf{q}))$ is a $\mathcal{C}^1$ function from $\mathcal{Q}$ into $X$.

## 3.7  Two Specific Problems and Their Sensitivity Equations

We now return to analyzing design sensitivities and focus on two specific parameter-dependent flow problems along with their sensitivity equations.

### 3.7.1  Flow around a Cylinder

We begin by considering the standard problem of 2-D flow around a cylinder. This problem, a non-homogeneous Dirichlet problem on a bounded domain, is modeled on the prob-
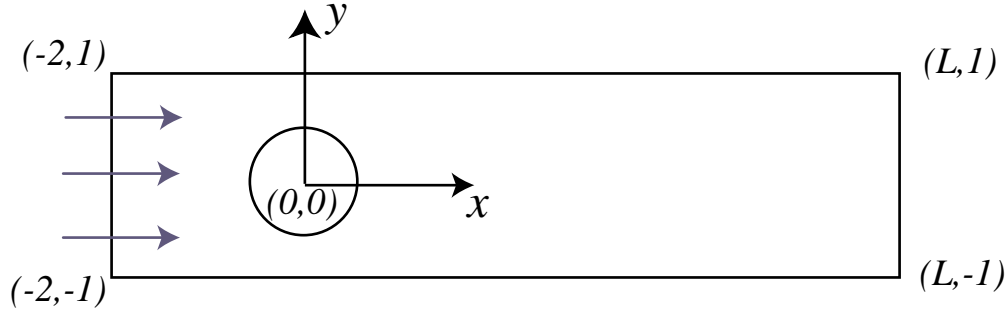
Figure 3.2: Geometry for 2-D Flow around a Cylinder

lem where the boundary is an infinite strip. We assume parabolic inflow into a channel containing a cylindrical obstruction whose geometry is shown in Figure (3.2). Here, $\Omega = [-2, L] \times [-1, 1]$, where $L \in I\!\!R^+$ is a fixed number. We assume that $L$ is large enough for the outflow to have returned to the same parabolic velocity profile present at the inflow. The governing equations are the 2-D incompressible Navier-Stokes equations presented earlier (see (3.10)) with $\mathbf{f} = 0$. The boundary conditions at the inflow and outflow are given by

$$\mathbf{u}(-2, y; q) = \mathbf{u}(L, y; q) = \begin{bmatrix} (1 + q)(1 - y^2) \\ 0 \end{bmatrix} \tag{3.58}$$

for $-1 \leq y \leq 1$, where $q \in I\!\!R^+$ is a parameter describing the strength of the inflow. No-penetration and no-slip conditions are applied on the top, bottom, and cylinder sidewalls (i.e. $\mathbf{u} = 0$).

We are interested in calculating sensitivities with respect to the inflow parameter $q$. As before with the case of the 1-D model problem, define the sensitivity as follows:

$$\left( \frac{\partial}{\partial q} u(x, y; q), \frac{\partial}{\partial q} v(x, y; q) \right) \triangleq \mathbf{s}(x, y; q) = \mathbf{s}(x, y). \tag{3.59}$$

The sensitivity equations are obtained by implicitly differentiating the system (3.10) and its associated boundary conditions with respect to the parameter $q$. Assuming the order of differentiation can be interchanged, we obtain the following:

$$\left. \begin{aligned} \nabla \cdot \mathbf{s} &= 0 \\ -\nu \Delta \mathbf{s} + \nabla \left( \frac{\partial}{\partial q} p \right) + \mathbf{u} \cdot \nabla \mathbf{s} + \mathbf{s} \cdot \nabla \mathbf{u} &= 0 \end{aligned} \right\} \quad \text{in} \quad \Omega. \tag{3.60}$$

Observe that this sensitivity equation does not assume a fully developed flow.
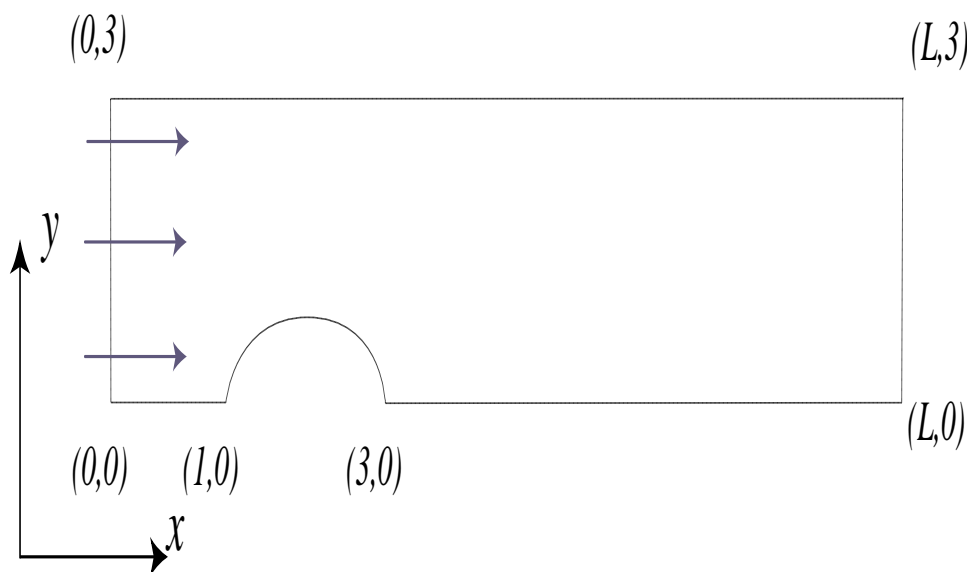
Figure 3.3: Geometry for Flow over a Bump

The sensitivity boundary conditions at the inflow and outflow are obtained by differentiating (3.58) producing

$$\mathbf{s}(-2, y; q) = \mathbf{s}(L, y; q) = \left[ \begin{array}{c} (1 - y^2) \\ 0 \end{array} \right], \qquad -1 \leq y \leq 1. \tag{3.61}$$

After differentiation, the no-penetration, no-slip conditions for $\mathbf{u}$ imply no-penetration, no-slip conditions for $\mathbf{s}$. In §3.5 , we present the weak form of these sensitivity equations and prove an existence result.

## 3.7.2   Flow over a Bump

We also consider a problem examined by Burkardt in [11], In particular, the problem is 2-D incompressible flow over a bump in a channel. The geometry of the channel is indicated in Figure (3.3) with $\Omega = [0, L] \times [0, 3]$, where $L \gg 0$ is again a fixed number representing the length of the channel. As before, the governing equations for this problem are the 2-D incompressible Navier-Stokes equations presented earlier (see (3.10)) with $\mathbf{f} = 0$. The boundary conditions for the flow are as follows:

$$\mathbf{u}(0, y) = \mathbf{u}(L, y) = \left[ \begin{array}{c} \frac{\lambda}{1.5^2} y(3 - y) \\ 0 \end{array} \right] \tag{3.62}$$

for $0 \leq y \leq 3$, where $\lambda = 0.5$ was a constant parameter describing the strength of the inflow. Again, no-penetration and no-slip conditions are applied on the top and bottom channel sidewalls as well as on the bump.

For this application, we examine a shape sensitivity. Here, the shape of the bump is a cubic spline determined by a parameter, $\mathbf{q}$. We examine two cases: $\mathbf{q} = q_1 \in I\!\!R^1$ and $\mathbf{q} = (q_1, q_2, q_3)^T \in I\!\!R^3$. We seek to find the sensitivity of the flow in the channel to changes in $\mathbf{q}$, which we denote $\mathbf{s}_i \stackrel{\triangle}{=} (\frac{\partial u}{\partial q_i}, \frac{\partial v}{\partial q_i}) \equiv (u_{q_i}, v_{q_i})$.

We have to solve a separate set of linear sensitivity equations for each sensitivity. The sensitivity equations are:

$$\left. \begin{array}{rl} \nabla \cdot \mathbf{s}_i &= 0 \\ -\nu \Delta \mathbf{s}_i + \nabla \left( \frac{\partial}{\partial q} p \right) + \mathbf{u} \cdot \nabla \mathbf{s}_i + \mathbf{s}_i \cdot \nabla \mathbf{u} &= 0 \end{array} \right\} \quad \text{in} \quad \Omega. \tag{3.63}$$

In this case, the sensitivity boundary conditions are given by

$$\begin{align} u_{q_i}(x, y) &= 0 \text{ at the inflow, outflow, top, and bottom walls;} \tag{3.64} \\ v_{q_i}(x, y) &= 0 \text{ at the inflow, outflow, top, and bottom walls;} \tag{3.65} \\ u_{q_i}(x, y) &= -\frac{\partial u}{\partial y} \frac{\partial h(x, \mathbf{q})}{\partial q_i} \text{ on the bump;} \tag{3.66} \\ v_{q_i}(x, y) &= -\frac{\partial v}{\partial y} \frac{\partial h(x, \mathbf{q})}{\partial q_i} \text{ on the bump;} \tag{3.67} \end{align}$$

where $h(x, \mathbf{q})$ denotes the $y$-coordinate of the boundary of $\Omega$ for $1 \leq x \leq 3$. We generate $h(x, \mathbf{q})$ using a cubic spline with free boundary conditions. If $\mathbf{q} \in I\!\!R^1$, $q_1$ specifies the height of the spline at $x = 2.0$. In this case, $h(x, \mathbf{q})$ satisfies the following:

$$\left. \begin{array}{c} h(x, \mathbf{q}) \text{ is piecewise cubic,} \\ h(1, \mathbf{q}) = h(3, \mathbf{q}) = 0, \\ h(2, \mathbf{q}) = q_1 \\ h'(1, \mathbf{q}) = h'(2, \mathbf{q}) = h'(3, \mathbf{q}) = 0 \end{array} \right\}. \tag{3.68}$$

For $\mathbf{q} \in I\!\!R^3$, $q_1$, $q_2$, and $q_3$ specify the height of the spline at $x = 1.5, 2.0,$ and $2.5$, respectively. Here, $h(x, \mathbf{q})$ satisfies

$$\left. \begin{array}{c} h(x, \mathbf{q}) \text{ is piecewise cubic,} \\ h(1, \mathbf{q}) = h(3, \mathbf{q}) = 0, \\ h(1.5, \mathbf{q}) = q_1 \\ h(2.0, \mathbf{q}) = q_2 \\ h(2.5, \mathbf{q}) = q_3 \\ h'(1, \mathbf{q}) = h'(3, \mathbf{q}) = 0 \\ h'(x, \mathbf{q}) \text{ is continuous at } x = 1.5, 2.0, \text{ and } 2.5 \end{array} \right\}. \tag{3.69}$$

## 3.8    A Finite Element Formulation

Since exact solutions to the Navier-Stokes equations are unavailable except for the simplest of problems, we now turn our attention to finding good methods for computing approximations to the solutions of the flow equations. We will use a finite element approach to solve our variational problems and, later, our weak sensitivity equations as well.

### 3.8.1    The Discrete Variational Problem and Finite Element Spaces

To begin, we choose a family of finite element spaces, $\mathbf{V}_0^h(\Omega)$ and $S_0^h(\Omega)$, for approximating velocity and pressure, respectively. Here $h$ is some measure of the size of the grid used to divide $\Omega$ into finite elements. Then the finite dimensional weak form of the homogeneous Dirichlet problem becomes:

**Variational Problem 4.** *Given* $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$, *find a pair* $\left(\mathbf{u}^h, p^h\right) \in \mathbf{V}_0^h(\Omega) \times S_0^h(\Omega)$ *such that:*

$$\left.\begin{array}{r} a_0(\mathbf{u}^h, \mathbf{v}^h) + a_1(\mathbf{u}^h; \mathbf{u}^h, \mathbf{v}^h) + b(\mathbf{v}^h, p^h) \;\; = \langle \mathbf{f},\, \mathbf{v}^h \rangle \qquad \forall\, \mathbf{v}^h \in \mathbf{V}_0^h(\Omega) \\ b(\mathbf{u}^h, q^h) = 0 \quad \forall\, q^h \in S_0^h(\Omega) \end{array}\right\} \tag{3.70}$$

When solving Navier-Stokes, the inclusions of our finite element spaces, $\mathbf{V}_0^h \subset \mathbf{H}_0^1(\Omega)$ and $S_0^h \subset L_0^2(\Omega)$ in our infinite dimensional function spaces are not by themselves sufficient to produce a stable, meaningful approximation. For stability, the finite element spaces must also satisfy the additional *div-stability* or *inf-sup* condition. Three equivalent forms of this condition from [22] are given below.

A finite element space is said to satisfy the *div-stability* or *inf-sup* condition if and only if one of the following equivalent conditions holds:

- Given any $q^h \in S_0^h$,

$$\sup_{0 \neq \mathbf{v}^h \in \mathbf{V}_0^h} \left( \frac{b(\mathbf{v}^h, q^h)}{|\mathbf{v}^h|_1} \right) \geq \gamma \|q^h\|_0, \tag{3.71}$$

  where the constant $\gamma > 0$ may be chosen independent of $h$ and of the particular choice of $q^h \in S_0^h$.

- Given any $q^h \in S_0^h$ there exists $\mathbf{v}^h \in \mathbf{V}_0^h$ such that

$$b(\mathbf{v}^h, q^h) \geq \gamma \|q^h\|_0 \left|\mathbf{v}^h\right|_1, \tag{3.72}$$

  where the constant $\gamma > 0$ may be chosen independent of $h$ and of the particular choice of $q^h \in S_0^h$.

- There exists a $\gamma > 0$, independent of $h$, such that

$$\inf_{0 \neq q^h \in S_0^h} \sup_{0 \neq \mathbf{v}^h \in \mathbf{V}_0^h} \left( \frac{b(\mathbf{v}^h, q^h)}{\|q^h\|_0 \left|\mathbf{v}^h\right|_1} \right) \geq \gamma. \tag{3.73}$$

As noted in [22], the *div-stability* condition (3.71) - (3.73) loosely ensures that as $h \to 0$ discretely divergence free functions tend to divergence free functions.

The discrete Weak Formulation 4 is solved in the primitive variables, $\mathbf{u}^h$ and $p^h$. The weak form is discretized using the 7 node Crouzier-Raviart triangular element (see Figure (3.4)) which is a type of "bubble" element described in [14]. This element uses a so-called enriched quadratic velocity interpolant and a discontinuous linear pressure. Then, if $\mathcal{T}_h$ is some triangulation of our domain, $\Omega$, the finite element space, $S_0^h$ can be formally defined as follows:

$$S_0^h = \left\{ q : q \in S_1(\Delta), \Delta \in \mathcal{T}_h; \qquad \int_\Omega q \, d\Omega = 0 \right\}, \text{ where} \tag{3.74}$$

where $S_1(\Delta)$ is the space of linear polynomials on the triangle, $\Delta$. The velocity finite element space is the space of continuous piecewise quadratic polynomials augmented in each triangle, $\Delta \in \mathcal{T}_h$, by the cubic bubble function that vanishes on the three edges of $\Delta$. Thus,

$$\mathbf{V}_0^h = \left\{ \mathbf{v} : \mathbf{v} \in \left[ \mathbf{S}_2(\Delta) \oplus \mathbf{B}_2(\Delta) \right], \Delta \in \mathcal{T}_h; \qquad \mathbf{v} \in \mathbf{C}_0(\Omega); \qquad \mathbf{v} = 0 \text{ on } \Gamma \right\}, \tag{3.75}$$

where, for any $\Delta \in \mathcal{T}_h$,

$$\mathbf{B}_2(\Delta) = \left\{ \mathbf{v} \in \mathbf{S}_2(\Delta) : \mathbf{v} = \lambda_1 \lambda_2 \lambda_3 \mathbf{u}, \qquad \mathbf{u} \in \mathbf{S}_0(\Delta) \right\}. \tag{3.76}$$

Here, $\lambda_i(\mathbf{x})$, $i = 1, 2, 3$, denote the barycentric coordinates of the point $\mathbf{x} \in \Delta$ with respect to the vertices of $\Delta$, see [14].

## 3.8.2 The Penalty Method

The pressure degrees of freedom are treated by the penalty method as in [17]. The penalty method allows for the uncoupling of the momentum and continuity equations through the
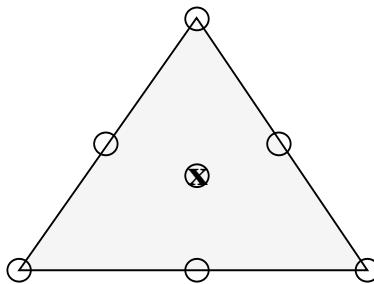
**X**   Pressure Node
    (1 nodal point + 2 derivatives)

O   Velocity Node
    (7 nodal points)

Figure 3.4: Crouzier-Raviart element

use of a small perturbation parameter, $\varepsilon$. This provides large computational advantages as it reduces the size of the system of equations being solved and eliminates the need for partial pivoting when solving the resultant linear systems.

The idea of the penalty method is to perturb the continuity equation with a small term containing the pressure:

$$\varepsilon p + \nabla \cdot \mathbf{u} = 0. \tag{3.77}$$

One can consider this perturbation as the introduction of a slight artificial compressibility. With this perturbation, the pressure p is expressed as

$$p = -\beta \nabla \cdot \mathbf{u}, \tag{3.78}$$

so that once a solution for $\mathbf{u}$ is known, $p$ is determined by the relation (3.78). Here $\beta = 1/\varepsilon$ is called the penalty parameter. With (3.78), we can eliminate pressure from the discrete form of the momentum equation. It can be shown that a solution $(\mathbf{u}^\beta, p^\beta)$ of the system,

$$\left. \begin{array}{r} -\beta \nabla \cdot \mathbf{u} = p \\ -\nu \Delta \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{f} \end{array} \right\} \quad \text{in} \quad \Omega. \tag{3.79}$$

approaches the solution of the unperturbed system (3.10) as $\beta \to \infty$ (see [37] for details for the homogeneous and nonhomogeneous cases).

### 3.8.3 An Implementation Method

Let $\phi_j(\mathbf{x})$, $j = 1, 2, \ldots, N$ and $\psi_l(\mathbf{x})$, $l = 1, 2, \ldots M$ be basis functions for the finite element spaces $\mathbf{V}_0^h$ and $S_0^h$, respectively. The Galerkin approximations $(\mathbf{u}^h, p^h)$ of the penalized variational form can be written as

$$\mathbf{u}^h(\mathbf{x}) \;=\; \sum_{j=1}^{N} \mathbf{u}_j \phi_j(\mathbf{x}), \text{ or, alternately } \left( \begin{array}{c} u^h(\mathbf{x}) \\ v^h(\mathbf{x}) \end{array} \right) = \left( \begin{array}{c} \sum_{j=1}^{N} u_j \phi_j(\mathbf{x}) \\ \sum_{j=1}^{N} v_j \phi_j(\mathbf{x}) \end{array} \right) \quad (3.80)$$

$$p^h(\mathbf{x}) \;=\; \sum_{l=1}^{M} p_l \psi_l(\mathbf{x}). \tag{3.81}$$

Substituting these approximations into the penalized, discrete weak form, we obtain the following system of Galerkin equations:

$$a_0(\mathbf{u}^h, \phi_j) + a_1(\mathbf{u}^h; \mathbf{u}^h, \phi_j) + b(\phi_j, p^h) \;=\; \langle \mathbf{f}, \phi_j \rangle \quad j = 1, 2, \ldots, N \text{ and} \tag{3.82}$$

$$\beta b(\mathbf{u}^h, \psi_l) \;=\; \langle p^h, \psi_l \rangle \quad l = 1, 2, \ldots, M. \tag{3.83}$$

Using the definitions of $a_0$, $a_1$, and $b$ and separating the momentum equations into $u$, $v$ components for clarity, we can write (3.82) - (3.83) in integral form as:

$$\int_{\Omega} \left[ \nu \nabla u^h \cdot \nabla \phi_i + \left( u^h \frac{\partial u^h}{\partial x} + v^h \frac{\partial u^h}{\partial y} \right) \phi_i - p^h \frac{\partial \phi_i}{\partial x} \right] d\Omega \;=\; \int_{\Omega} f_1 \phi_i d\Omega, \; i = 1, 2, \ldots, N \quad (3.84)$$

$$\int_{\Omega} \left[ \nu \nabla v^h \cdot \nabla \phi_i + \left( u^h \frac{\partial v^h}{\partial x} + v^h \frac{\partial v^h}{\partial y} \right) \phi_i - p^h \frac{\partial \phi_i}{\partial y} \right] d\Omega \;=\; \int_{\Omega} f_2 \phi_i d\Omega, \; i = 1, 2, \ldots, N \quad (3.85)$$

$$-\int_{\Omega} \beta \left[ \frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} \right] \psi_l d\Omega \;=\; \int_{\Omega} p^h \psi_l d\Omega, \; l = 1, 2, \ldots, M. \quad (3.86)$$

Equation (3.86) can be written more succinctly in matrix-vector notation as

$$-\beta L^T \mathbf{u}^h = D \mathbf{p}^h \tag{3.87}$$

where,

$$\mathbf{p}^h \in I\!\!R^M, \qquad \mathbf{p}^h = [p_1, p_2, \ldots, p_M]^T, \tag{3.88}$$

$$\mathbf{u}^h \in I\!\!R^{2N}, \qquad \mathbf{u}^h = [u_1, u_2, \ldots, u_N, v_1, v_2, \ldots, v_N]^T, \tag{3.89}$$

and $D \in I\!\!R^{M \times M}$ is the pressure mass matrix

$$D_{ij} = \int_{\Omega} \psi_i \psi_j \, d\Omega \qquad i, j = 1, 2, \ldots, M, \tag{3.90}$$

and $L \in I\!\!R^{M \times 2N}$ is the continuity matrix defined as follows:

$$L_{ij} = \begin{cases} \int_{\Omega} \psi_i \frac{\partial \phi_j}{\partial x} \, d\Omega & i = 1, 2, \ldots, M, \ j = 1, 2, \ldots, N \\ \int_{\Omega} \psi_i \frac{\partial \phi_{j-N}}{\partial y} \, d\Omega & i = 1, 2, \ldots, M, \ j = N+1, N+2, \ldots, 2N \end{cases} . \tag{3.91}$$

We can solve (3.87) for $\mathbf{p}^h$ so that

$$\mathbf{p}^h = -\beta D^{-1} L^T \mathbf{u}^h. \tag{3.92}$$

To linearize (3.84) - (3.85), we consider an iterative method where we write the solution at the new level, $n$, as the sum of the preceding level, $n-1$, and a correction:

$$\mathbf{u}^{h,n} = \mathbf{u}^{h,n-1} + \delta\mathbf{u}^{h,n-1} \tag{3.93}$$

Substituting (3.93) into the convective terms of (3.84) - (3.85) and neglecting the quadratic terms in $\delta\mathbf{u}$, the linearized form of the convective terms of the discrete momentum equations become

$$\int_{\Omega} \left[ u^{h,n} \frac{\partial u^{h,n-1}}{\partial x} + u^{h,n-1} \frac{\partial u^{h,n}}{\partial x} - u^{h,n-1} \frac{\partial u^{h,n-1}}{\partial x} + \right.$$
$$\left. v^{h,n} \frac{\partial u^{h,n-1}}{\partial y} + v^{h,n-1} \frac{\partial u^{h,n}}{\partial y} - v^{h,n-1} \frac{\partial u^{h,n-1}}{\partial y} \right] \phi_i d\Omega \tag{3.94}$$

$$\int_{\Omega} \left[ u^{h,n} \frac{\partial v^{h,n-1}}{\partial x} + u^{h,n-1} \frac{\partial v^{h,n}}{\partial x} - u^{h,n-1} \frac{\partial v^{h,n-1}}{\partial x} + \right.$$
$$\left. v^{h,n} \frac{\partial v^{h,n-1}}{\partial y} + v^{h,n-1} \frac{\partial v^{h,n}}{\partial y} - v^{h,n-1} \frac{\partial v^{h,n-1}}{\partial y} \right] \phi_i d\Omega \tag{3.95}$$

for $i = 1, 2, \ldots, N$.

In matrix-vector form, the linearized convective terms can be written as

$$N(\mathbf{u}^{h,n-1})\mathbf{u}^{h,n} - \mathbf{g}, \tag{3.96}$$

where $N(\mathbf{u}^{h,n-1}) \in I\!\!R^{2N \times 2N}$. The matrix $N$ is defined as

$$N = \begin{bmatrix} N_{uu} & \vdots & N_{uv} \\ \cdots\cdots\cdots\cdots \\ N_{vu} & \vdots & N_{vv} \end{bmatrix} \tag{3.97}$$

with the submatrices defined by the following:

$$[N_{uu}]_{i,j} = \int_\Omega \left( \phi_j \frac{\partial u^{h,n-1}}{\partial x} + u^{h,n-1} \frac{\partial \phi_j}{\partial x} + v^{h,n-1} \frac{\partial \phi_j}{\partial y} \right) \phi_i \, d\Omega, \tag{3.98}$$

$$[N_{vv}]_{i,j} = \int_\Omega \left( \phi_j \frac{\partial v^{h,n-1}}{\partial y} + u^{h,n-1} \frac{\partial \phi_j}{\partial x} + v^{h,n-1} \frac{\partial \phi_j}{\partial y} \right) \phi_i \, d\Omega, \tag{3.99}$$

$$[N_{uv}]_{i,j} = \int_\Omega \phi_j \frac{\partial u^{h,n-1}}{\partial y} \phi_i \, d\Omega, \text{ and } [N_{vu}]_{i,j} = \int_\Omega \phi_j \frac{\partial v^{h,n-1}}{\partial x} \phi_i \, d\Omega, \tag{3.100}$$

for $i, j = 1, 2, \ldots, N$. The vector $\mathbf{g} \in \mathbb{R}^{2N}$ is given by

$$\mathbf{g} = \begin{bmatrix} g_1 \\ \ldots \\ g_2 \end{bmatrix} \text{ with } \begin{cases} g_1 = \int_\Omega \left( u^{h,n-1} \frac{\partial u^{h,n-1}}{\partial x} + v^{h,n-1} \frac{\partial u^{h,n-1}}{\partial y} \right) \phi_i \, d\Omega \ i = 1, 2, \ldots, N \\ g_2 = \int_\Omega \left( u^{h,n-1} \frac{\partial v^{h,n-1}}{\partial x} + v^{h,n-1} \frac{\partial v^{h,n-1}}{\partial y} \right) \phi_i \, d\Omega \ i = 1, 2, \ldots, N. \end{cases} \tag{3.101}$$

With these definitions, the complete, linearized momentum equations (3.84) - (3.85) can be written in matrix-vector notation as

$$S\mathbf{u}^{h,n} + N(\mathbf{u}^{h,n-1})\mathbf{u}^{h,n} + \tau L D^{-1} L^T \mathbf{u}^{h,n} = \mathbf{h} + \mathbf{g}, \tag{3.102}$$

where $S \in \mathbb{R}^{2N \times 2N}$ is the diffusion matrix with

$$S_{i,j} = \begin{cases} \nu \int_\Omega \nabla\phi_i \cdot \nabla\phi_j & i, j = 1, 2, \ldots, N \\ \nu \int_\Omega \nabla\phi_{i-N} \cdot \nabla\phi_{j-N} & i, j = N+1, N+2, \ldots, 2N \\ 0 & \text{otherwise.} \end{cases} \tag{3.103}$$

and the right-hand side vector $\mathbf{h} \in \mathbb{R}^{2N}$ defined by

$$\mathbf{h} = \begin{bmatrix} h_1 \\ \ldots \\ h_2 \end{bmatrix} \text{ with } \begin{cases} h_1 = \int_\Omega f_1 \phi_i \, d\Omega & i = 1, 2, \ldots, N \\ h_2 = \int_\Omega f_2 \phi_i \, d\Omega & i = 1, 2, \ldots, N. \end{cases} \tag{3.104}$$

We finish this section with a few notes about the structure of the matrices $L, S,$ and $D$ for the Crouzier-Raviart element described in § 3.8.1 above. First, note that with the use of the discretized continuity and momentum equations and Gauss' theorem that the velocity, and in turn, the pressure derivatives at the centroid of the element can be written in terms of the other nodal quantities. This results in a velocity approximation, $\mathbf{u}^h$, determined by six nodal points and a pressure approximation, $p^h$, determined by one nodal point. The numerical advantage of this element then is that the number of equations *and* unknowns in each element is reduced by 3, without affecting the accuracy of the approximation. In addition, the matrix $D$ can be written as a diagonal matrix with entries $D_{i,i}$ representing the area of element $e_i$. For further details on the implementation of the penalty function method with the Crouzier-Raviart element, the reader is referred to [17].

### 3.8.4   Adaptive Methodology

The basic idea of adaptive gridding is to use some error estimation technique to evaluate the quality of your finite element approximation and to strategically modify the grid based on that evaluation. The grid modification scheme allows the user control over element size and grading. This process has been shown to be successful in resolving shear, stagnation points, jets, and wakes (see [34], [33], [24], [29], [25]). The two main elements of the adaptive process are error estimation and grid generation. We discuss each of these below.

**Error Estimation**

The error estimation is performed using an approach introduced by Zhu and Zienkiewicz (see [41], [42], [1]) and involves the postprocessing of stresses and strains. Recall that the energy norm of $\mathbf{u}$ is

$$\|\mathbf{u}\|_E = \sqrt{\int_\Omega \tau(\mathbf{u}) : \tau(\mathbf{u})\, d\Omega}, \tag{3.105}$$

or, given in Cartesian coordinates,

$$\|\mathbf{u}\|_E = \sqrt{\int_\Omega \mu^2 \left[ 4\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + 4\left(\frac{\partial v}{\partial y}\right)^2 \right] d\Omega}. \tag{3.106}$$

Note that the energy norm has a very similar form to the $H^1$ seminorm. In fact, it can be shown that they are equivalent norms. Using the energy norm for the velocity, define the so-called Stokes norm of the solution as

$$\|(\mathbf{u}, p)\|_S = \sqrt{\|\mathbf{u}\|_E^2 + \|p\|^2}. \tag{3.107}$$

As pointed out in [35] the use of the energy norm over the $H^1$ seminorm offers some advantages especially to the engineering community. Note that both the velocity and pressure norms are expressed in terms of surface forces which are the quantities of prime interest in engineering fluid mechanics. Secondly, errors computed in these norms can be interpreted as errors in the stresses which can then easily be related to errors in global quantities such as lift and drag.

The Zhu and Zienkiewicz approach uses the Stokes norm to measure the error, $\mathbf{e}(\mathbf{u}, p) = (\mathbf{u}_{ex} - \mathbf{u}^h, p_{ex} - p^h)$, where $(\mathbf{u}_{ex}, p_{ex})$ is the exact solution of the flow problem and $(\mathbf{u}^h, p^h)$

is the finite element approximation. Then the norm of the error is

$$\|\mathbf{e}(\mathbf{u}, p)\|_S = \sqrt{\|\mathbf{u}_{ex} - \mathbf{u}^h\|_E^2 + \|p_{ex} - p^h\|^2}. \tag{3.108}$$

We concentrate on forming an approximation to $\|\mathbf{u}_{ex} - \mathbf{u}^h\|_E$. Since the exact solution, and more particularly the gradients of the exact solution are not known, the approach is to use the finite element approximation to construct approximations to these gradients. Note that the finite element approximations to the gradients are discontinuous across element faces while the exact gradients are, in most cases, continuous across the domain. Thus, the first goal in error estimation is to obtain continuous approximations to the discontinuous finite element gradients. Two methods have been evaluated for this process: global projections and local least squares projections. Global projections are done over the entire domain, $\Omega$, and involve finding the best approximation to the discontinuous finite element gradients in the original continuous finite element space. For example, if a piecewise linear approximation of the flow solution is calculated, then the finite element gradient is piecewise constant and discontinuous across elements. The global projection would replace the piecewise constant gradient approximation with a piecewise linear approximation calculated by projecting the piecewise constant function onto the original finite element basis functions. The local least squares projections, however, are done node by node and only consider gradient information from subdomains of $\Omega$ which contain the current node. Thus, a series of smaller projections are done in combination with some averaging techniques to obtain a continuous gradient projection for the entire domain, $\Omega$. The details of each of these projection techniques will be described further in Chapter 4.

The $\|p_{ex} - p^h\|$ is similarly approximated except that we construct a continuous, quadratic approximation of $p_{ex}$ using local projections of the discontinuous linear finite element approximations. Once this is done the $L^2$ norm can be calculated as usual.

We now return to the issue of adaptive gridding to briefly describe the remeshing strategy.

**Remeshing Strategy**

Once error estimates are obtained for each element, say $e_i$, a new mesh density (or element size), $d$, is calculated which requires equidistribution of the element errors across all the elements. For example if we wish to reduce the error in each element by a factor of $\gamma$, then the target error, $e_T$, for an element in the new mesh can be given by

$$e_T = \frac{\gamma e}{\sqrt{N}} \tag{3.109}$$

where $e$ is the error over the entire domain and N is the number of elements. If one assumes that the finite element method is of order k, then it is reasonable to write

$$e_i \ = \ ch^k \tag{3.110}$$

$$e_T \ = \ cd^k, \tag{3.111}$$

where $h$ is the current element size and $d$ is the "ideal" element size we seek. Clearly, we have assumed that we are in the asymptotic range of the finite element method and that the convergence constant, $c$, is the same for both meshes. This may or may not be the case. Nevertheless, the system (3.110) - (3.111) can be solved for the new mesh density, $d$, obtaining

$$d = \left( \frac{\gamma e}{e_i \sqrt{N}} \right)^{1/k} h. \tag{3.112}$$

The new element size computation is done for each of the dependent variables in the problem, e.g. velocity, pressure, and an ideal mesh density is obtained for each. The one used for the generation of the new mesh is the minimum of each of these.

The details of the actual redefinition of the mesh are omitted here. We refer the interested reader to [24] and [25].

Despite the rather major assumptions made above, this adaptive remeshing strategy works remarkably well. The strategy has been verified in a series of numerical experiments for a variety of flow types (see [34], [33], [24], [29], [25]). We use this adaptive strategy in the problems investigated below.

## 3.9   Some Numerical Results

In this section, we apply the computational techniques outlined above to the two problems presented in §3.7 above and investigate convergence of the approximate solution as the mesh is refined.

### 3.9.1   Flow around a Cylinder

Consider first the cylinder problem presented in §3.7.1 . We again assume parabolic inflow into the channel as follows:

$$\mathbf{u}(-2, y; q) = \begin{bmatrix} (1+q)(1-y^2) \\ 0 \end{bmatrix} \qquad -1 \le y \le 1. \tag{3.113}$$

The outflow boundary condition is modified, however, since applying the Dirichlet boundary condition at the outflow causes numerical instability (see [17]). A free boundary condition is used which requires

$$
\left. \begin{array}{rcl}
-p(L, y; q)\hat{\mathbf{n}} + \tau(\mathbf{u}(L, y; q)) \cdot \hat{\mathbf{n}} &=& 0, \\
v(L, y; q) &=& 0
\end{array} \right\} \qquad -1 < y < 1 \qquad (3.114)
$$

where $\hat{\mathbf{n}}$ is the outward normal at the end of the channel. No-flow and no-slip conditions are applied on the top, bottom, and cylinder sidewalls.

As noted earlier, we wish to approximate the sensitivity of the solution with respect to the inflow parameter $q$. The sensitivity boundary conditions at the inflow are obtained as before, with

$$
\mathbf{s}(-2, y; q) = \left[ \begin{array}{c} (1 - y^2) \\ 0 \end{array} \right], \qquad -1 \le y \le 1. \qquad (3.115)
$$

The computational outflow boundary conditions for sensitivity become

$$
\left. \begin{array}{rcl}
-r(L, y; q)\hat{\mathbf{n}} + \tau(\mathbf{s}(L, y; q)) \cdot \hat{\mathbf{n}} &=& 0, \\
\mathbf{s}_u(L, y; q) &=& 0
\end{array} \right\} \qquad -1 < y < 1. \qquad (3.116)
$$

Numerical results for this flow problem were generated using the approximation techniques outlined in §3.8. The Reynolds number for the calculations was $Re = 100$, the length of the channel was 8 ($L = 6$), and the sensitivity parameter was $q = 0$. Contour plots of the $u, v$-velocity fields as well as the $u, v$-velocity sensitivities are given in Figures (3.6) - (3.9). In Figure (3.5), the initial and adapted grids are shown. It is clear that the mesh is refined around the cylinder and in areas of large velocity gradients. This gives improved approximations of the velocity field as can be seen in Figures (3.6) and (3.7). Since the mesh refines on the velocity field, it is convenient that, for this problem, the sensitivity flow field is similar to the velocity field, so that as the mesh refines we obtain improved sensitivity approximations as well (see Figures (3.8)-(3.9)). It is important to note that this is not always the case as is shown in [8]. The code has been modified by Jeff Borggaard to adapt on the sensitivity field as well and some results for this will be shown later.
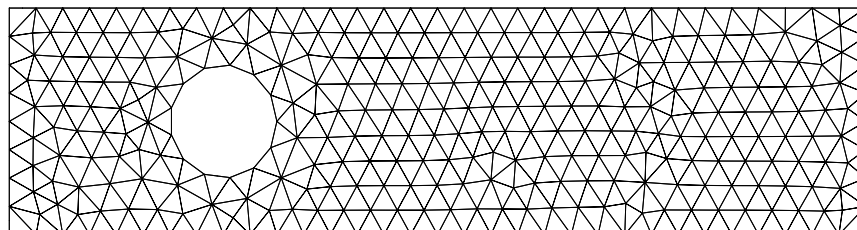
a) Initial Mesh (1219 nodes/563 elements)
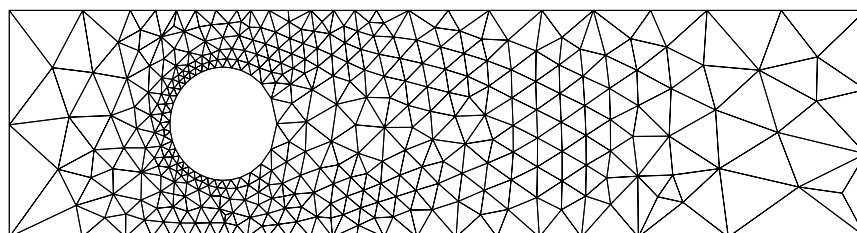
b) Second Adapted Mesh (1439 nodes/677 elements)
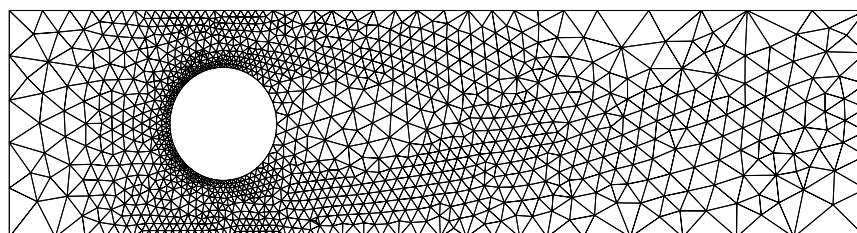
c) Fourth Adapted Mesh (19600 nodes/10818 elements)

Figure 3.5: Initial and Adapted Meshes for Cylinder Problem
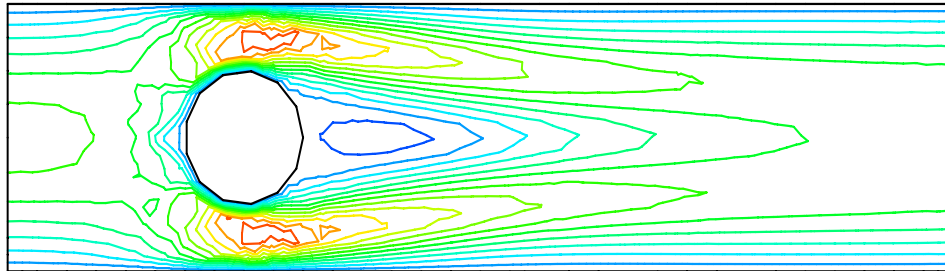
a) Initial Mesh
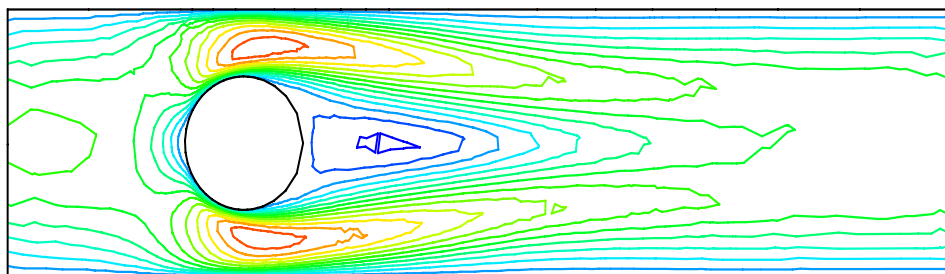
b) Second Adapted Mesh
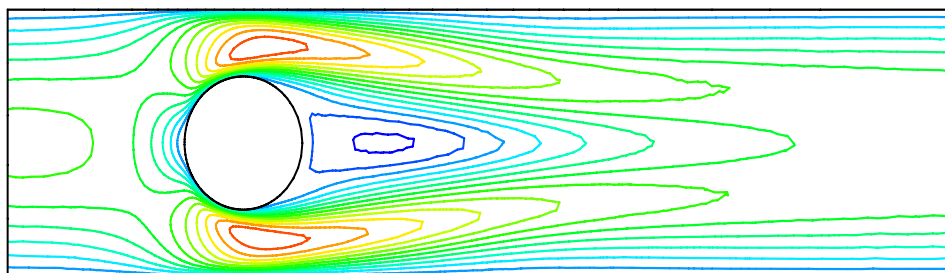
c) Fourth Adapted Mesh
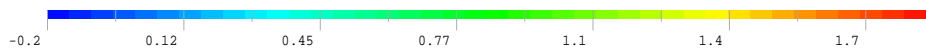
Figure 3.6: u-Velocity Contours for Flow around a Cylinder

a) Initial Mesh



b) Second Adapted Mesh



c) Fourth Adapted Mesh

-1 -0.83 -0.67 -0.5 -0.33 -0.17  0  0.17 0.33  0.5 0.67 0.83  1
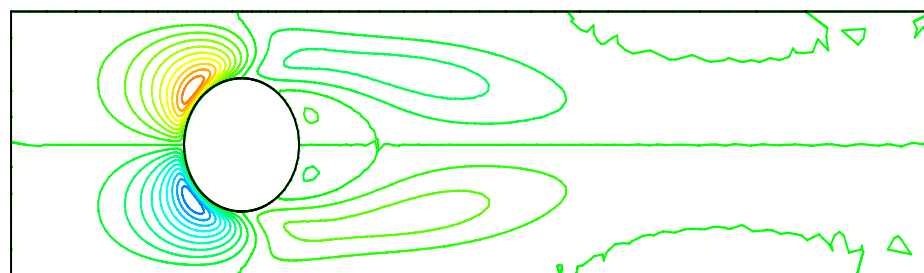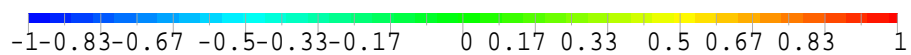
Figure 3.7: v-Velocity Contours for Flow around a Cylinder

a) Initial Mesh

b) Second Adapted Mesh

c) Fourth Adapted Mesh

−1.1        −0.67        −0.18        0.3        0.79        1.3        1.8
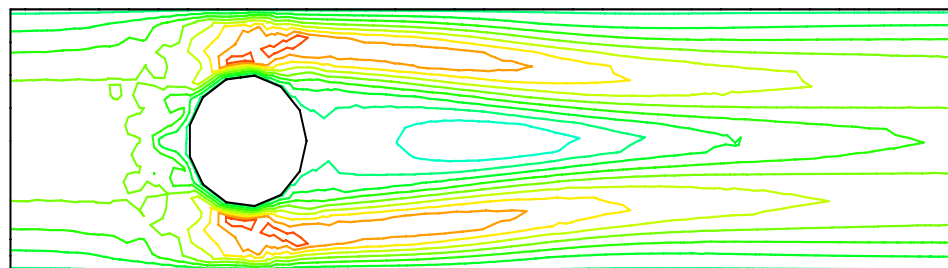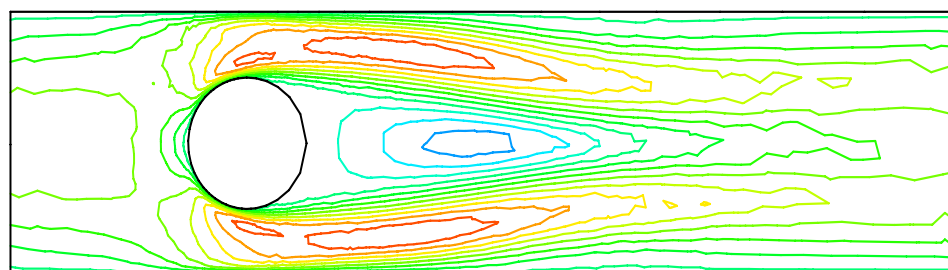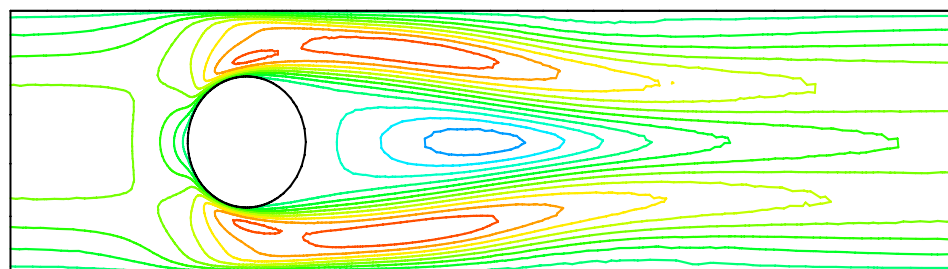
Figure 3.8: u-Velocity Sensitivity Contours for Flow around a Cylinder

a) Initial Mesh



b) Second Adapted Mesh
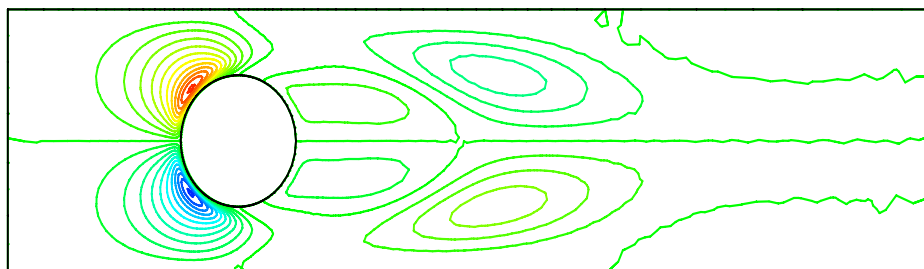


c) Fourth Adapted Mesh



Figure 3.9: v-Velocity Sensitivity Contours for Flow around a Cylinder

a) Initial Mesh (625 Nodes/276 Elements)



b) Second Adapted Mesh (1094 Nodes/505 Elements)



c) Fourth Adapted Mesh (3962 Nodes/1901 Elements)
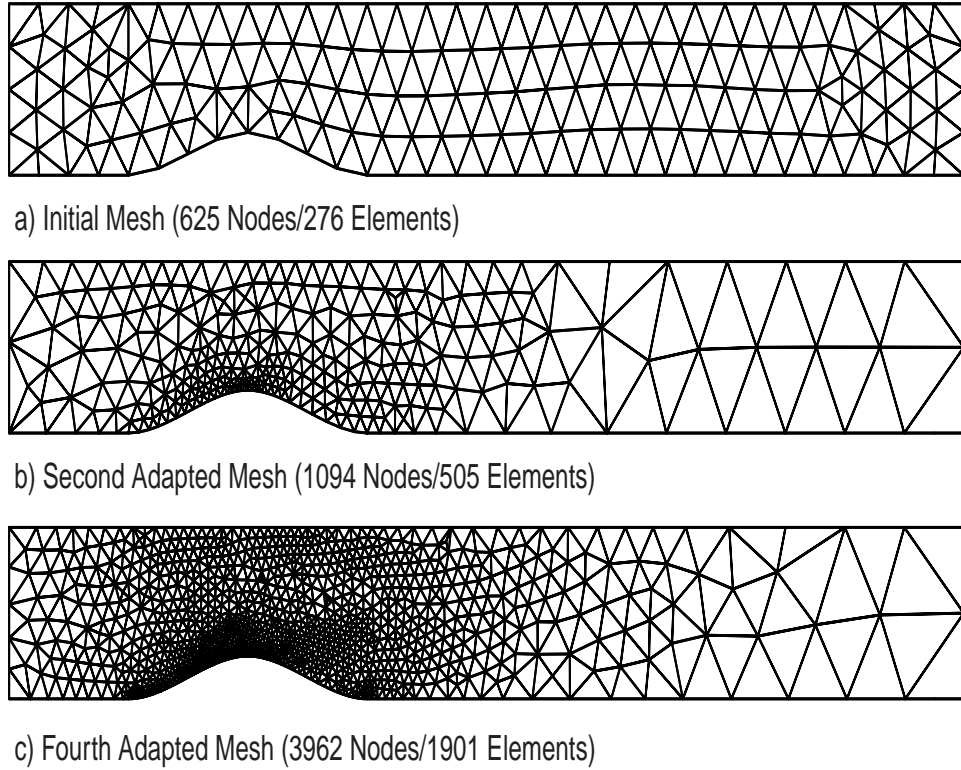
Figure 3.10: Initial and Adapted Meshes for Bump Problem

## 3.9.2   Flow over a Bump

We next consider the flow over a bump problem presented in §3.7.2. The outflow boundary condition for the computations is again taken to be a free boundary condition as follows:

$$\left.\begin{array}{rl} -p(L, y; q)\hat{\mathbf{n}} + \tau(\mathbf{u}(L, y; q)) \cdot \hat{\mathbf{n}} = & 0, \\ v(L, y; q) = & 0 \end{array}\right\} \qquad 0 < y < 3. \qquad (3.117)$$

The Reynolds number for the calculations was $Re = 100$ and the length of the channel was $L = 8$. The initial and adapted grids are shown in Figure (3.10). Contour plots for $u, v$ velocities and sensitivities are displayed in Figures (3.11) - (3.12). Note that the mesh refines in the area of the bump and the elements are allowed to become larger downstream in the channel where the flow is again quadratic. The bump problem is similar to the cylinder problem in that the sensitivities are largest in the same areas where the mesh refines. Thus we obtain improved approximations for sensitivities as we refine the mesh for the flow.

So far we have developed sensitivity equations for both 1-D and 2-D parameter dependent differential equations. We have obtained good numerical approximations for the continuous sensitivity equations using an adaptive finite element technique. In the 1-D problem, there were parameter ranges for which it became difficult to obtain good sensitivity approximations. For the 2-D flow problems, we observed that significant refinement was needed to obtain "good" sensitivity approximations. In the next chapter, we look at a projection technique for obtaining better derivative approximations for the state. We will show that these derivative approximations improve the accuracy of the sensitivity calculations and, in addition, stabilize those calculations, over larger parameter ranges.
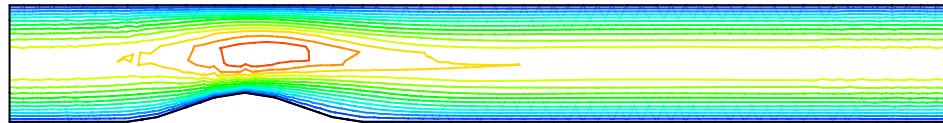
a) u-Velocity Contours - Initial Mesh
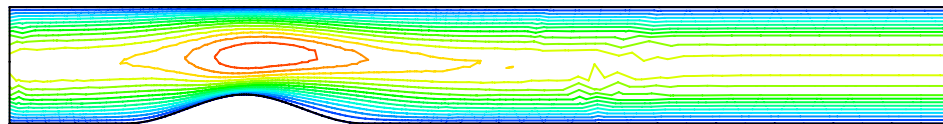
b) u-Velocity Contours - Second Adapted Mesh

c) u-Velocity Contours - Fourth Adapted Mesh
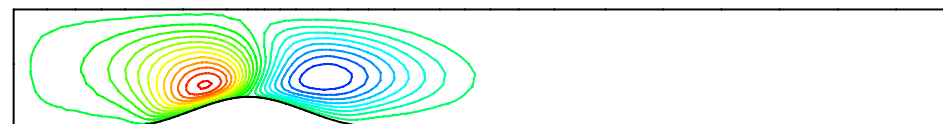
| 0 | 0.091 | 0.18 | 0.27 | 0.37 | 0.46 | 0.55 | 0.64 |

d) v-Velocity Contours - Initial Mesh

e) v-Velocity Contours - Second Adapted Mesh

f) v-Velocity Contours - Fourth Adapted Mesh

| -0.07 | -0.054 | -0.038 | -0.022 | -0.006 | 0.01 | 0.026 | 0.042 | 0.058 | 0.074 | 0.09 |
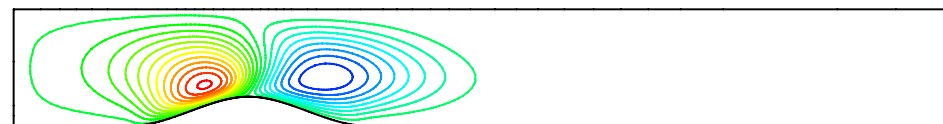
Figure 3.11: u,v-Velocity Contours for Flow over a Bump

a) u-Velocity Sensitivity Contours - Initial Mesh
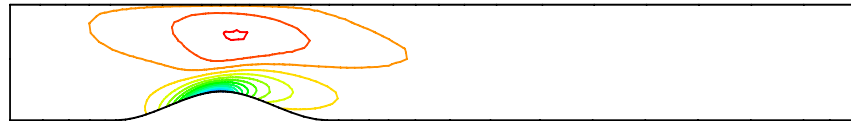
b) u-Velocity Sensitivity Contours - Second Adapted Mesh
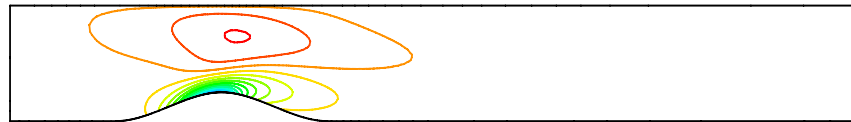
c) u-Velocity Sensitivity Contours - Fourth Adapted Mesh

−5.5        −4.6        −3.6        −2.7        −1.8        −0.86        0.071        1

d) v-Velocity Sensitivity Contours - Initial Mesh
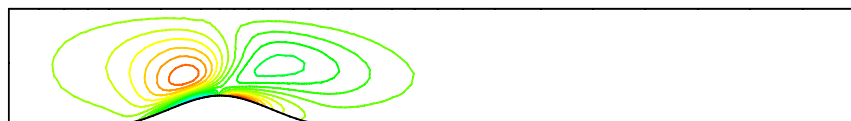
e) v-Velocity Sensitivity Contours - Second Adapted Mesh

f) v-Velocity Sensitivity Contours - Fourth Adapted Mesh

−1.1    −0.93    −0.75    −0.58    −0.4    −0.23    −0.056    0.12    0.29    0.47    0.64
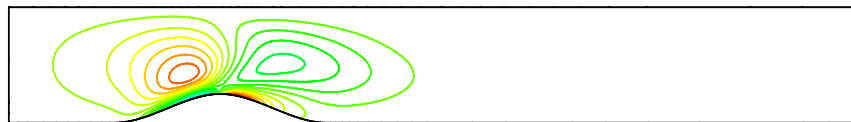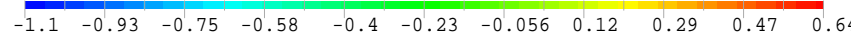
Figure 3.12: u,v-Velocity Sensitivity Contours for Flow over a Bump

# Chapter 4

# Gradient Approximations

## 4.1  Improving the Gradient Approximations

The goal of this chapter is to describe how projection techniques can be used to obtain better sensitivity approximations by obtaining better state gradient approximations. There are a number of ways to do this. In the following chapter, we analyze a global and local projection technique for calculating continuous gradient approximations and evaluate their impact on providing improved sensitivity values.

## 4.2  1-D Model Problem

Recall that the approximate sensitivity equation (2.30) for the 1-D model problem has the form

$$-\left\langle \frac{\partial}{\partial \xi} p^M(\cdot), \frac{\partial}{\partial \xi} h_j^M(\cdot) \right\rangle$$

$$+\frac{3}{8q} \left\langle \left(g^N(\cdot)\right)^2 \left( \frac{\partial}{\partial \xi} p^M(\cdot) - \frac{g^N(1)}{q} \right), h_j^M(\cdot) \right\rangle = 0 \tag{4.1}$$

where $j = 1, 2, \ldots, M$ and

$$g^N(\xi) = \frac{d}{d\xi} z^N(\xi) + 4. \tag{4.2}$$

Here, $g^N(\xi)$ is a piecewise step function providing an approximation of the spatial derivatives needed in (4.1). Even in higher dimensions or when using higher order basis functions, the finite element derivatives will be discontinuous across element faces. As in the error estimation technique described above, we seek to replace $g^N(\xi)$ with a continuous function which we will denote $\overline{g^N(\xi)}$. We will now describe global and local projection schemes for constructing this function.

## 4.2.1  A Global Projection Scheme

In its simplest form, this approach replaces the discontinuous piecewise constant function $g^N(\xi)$ by its projection onto the space of piecewise linear splines on the mesh defined by the nodes $\xi_i$. Thus, we consider the space

$$\mathcal{S}^N = \left\{ g(\xi) \ : \ g(\xi) = \sum_{i=0}^{N+1} g_i h_i^N(\xi) \right\}, \tag{4.3}$$

where $h_i^N(\cdot)$ are the hat functions defined in §2.5. Observe that $\mathcal{S}_0^N \subseteq \mathcal{S}^N$ and $\mathcal{S}^N$ contains functions with non-zero trace on the boundary of $[0,1]$.

Here, we let $\overline{g^N(\xi)}$ be the orthogonal projection of $g^N(\xi)$ onto $\mathcal{S}^N$. In particular,

$$\overline{g^N(\xi)} = \sum_{i=0}^{N+1} \beta_i h_i^N(\xi) \tag{4.4}$$

where $\beta = (\beta_0, \beta_1, \dots, \beta_{N+1})^T$ is the solution of a linear system of the form

$$M_G \beta = F\alpha \tag{4.5}$$

and $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ contains the coefficients defined by the finite element approximation of $z^N(\xi)$. Therefore,

$$g^N(\xi) = \sum_{i=1}^{N} \alpha_i \frac{d}{d\xi} h_i^N(\xi), \tag{4.6}$$

$M_G$ is the $(N+2) \times (N+2)$ global mass matrix

$$M_G = \left[ \langle h_j^N(\cdot), h_k^N(\cdot) \rangle \right]^T, \tag{4.7}$$

and $F$ is the $(N+2) \times N$ matrix

$$F = \left[ \left\langle \frac{d}{d\xi} h_i^N(\cdot), h_j^N(\cdot) \right\rangle \right]^T, \tag{4.8}$$

for $i = 1, 2, \dots, N$ and $j, k = 0, 1, 2, \dots, N+1$.

## 4.2.2   A Local Projection Scheme

In addition to the global projection scheme, we consider a local projection scheme which involves performing a series of local projections on subdomains of $\Omega = [0, 1]$. At each element vertex, $\xi_i$, we define the subdomain $\Omega_i$ to be the union of all elements for which $\xi_i$ is a vertex. In $\Omega_i$, define $\widetilde{g}_i(\xi)$ to be the least squares projection of $g^N(\xi)\big|_{\Omega_i}$ onto the space of linear polynomials spanned by monomial basis functions. Denote these basis functions as $P_1(\xi) = 1$ and $P_2(\xi) = \xi$. On the subdomain $\Omega_i$, we express the projection as

$$\widetilde{g}_i(\xi) = \sum_{i=1}^{2} a_i P_i(\xi), \tag{4.9}$$

where the vector $a = (a_1, a_2)^T$ contains the coefficients of the basis functions. These coefficients are determined by solving the normal equation system

$$M_L a = b. \tag{4.10}$$

The matrix $M_L$ is of the form

$$M_L = \left[ \langle P_i(\cdot), P_j(\cdot) \rangle|_{\Omega_i} \right], \tag{4.11}$$

for $i, j = 1, 2$. The vector on the right side of the equation is

$$b = \left[ \langle P_i(\cdot), g^N(\cdot) \rangle|_{\Omega_i} \right], \tag{4.12}$$

for $i = 1, 2$. Then, on $\Omega$ we define the continuous local projection to be

$$\overline{g^N(\xi)} = \sum_{i=0}^{N+1} \widetilde{g}_i(\xi_i) h_i^N(\xi). \tag{4.13}$$

With higher order finite elements and in higher dimensions, one must resolve the value of $\overline{g^N(\xi)}$ at a non-vertex node. An averaging technique is generally used. The technique will be described in more detail in the next section.

## 4.2.3   Numerical Results

**Derivative Approximations**

Figures (4.1) and (4.2) show the finite element derivative and its local and global projections against the exact spatial derivative for a $N = 4$, $q = 2.0$ and $N = 8$, $q = 1.2$,
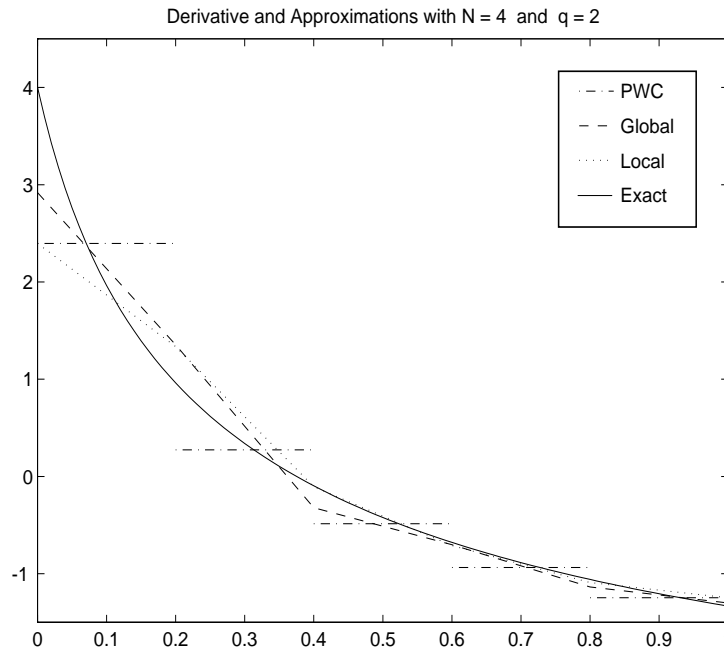
Figure 4.1:  Finite Element Derivatives with Projections at $N = 4$ and $q = 2$
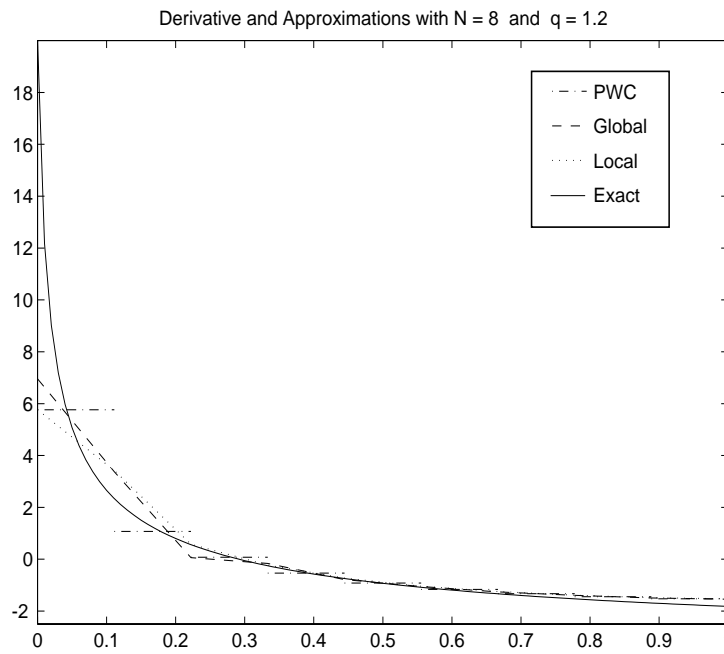


Figure 4.2:  Finite Element Derivatives with Projections at $N = 8$ and $q = 1.2$
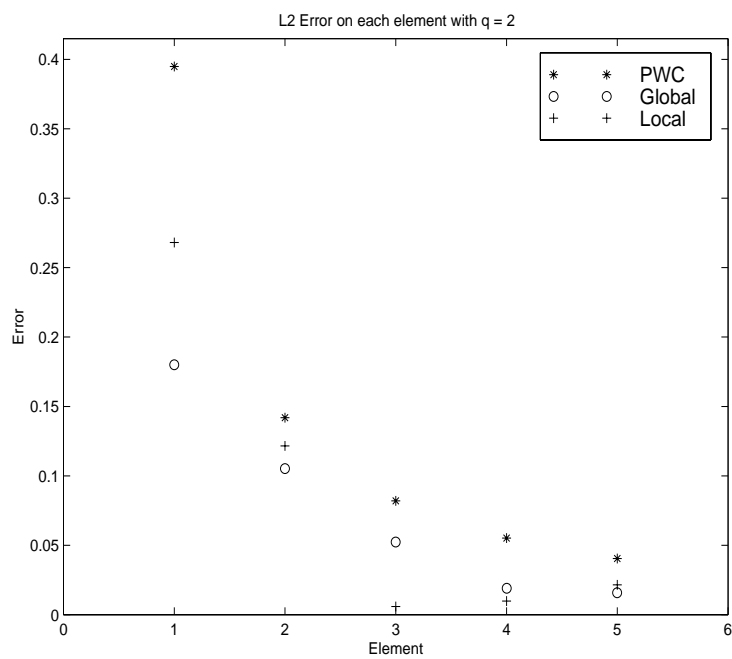
Figure 4.3: $L^2$ Error on each Element for $N = 4$ and $q = 2$
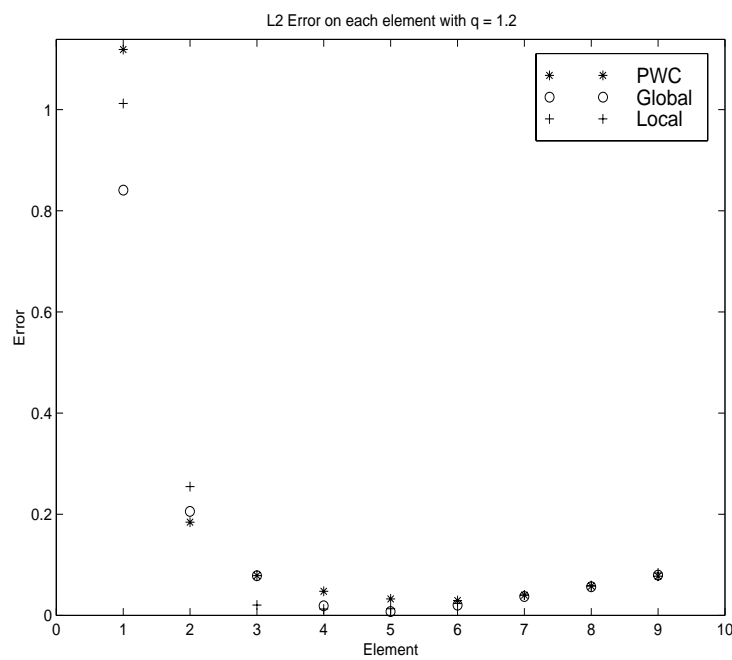


Figure 4.4: $L^2$ Error on each Element for $N = 8$ and $q = 1.2$

Table 4.1: $L^2$ Errors for the Derivative Approximations

| | $L^2$ Errors | |
|---|---|---|
| | $N = 4$ and $q = 2$ | $N = 8$ and $q = 1.2$ |
| PWC | 0.4330 | 1.1432 |
| Global | 0.2165 | 0.8760 |
| Local | 0.2954 | 1.0501 |

respectively. Clearly, the global and local projections give different results. Since we have an exact solution, and thus its spatial derivative, we can calculate element by element $L^2$ errors as well as overall $L^2$ errors for the whole domain for each derivative approximation. Figures (4.3) - (4.4) show the $L^2$ element errors for the two cases, $q = 2.0, /N = 4$ and $q = 1.2, /N = 8$, repectively. The $L^2$ errors over the whole domain for each of these cases is summarized in Table (4.1). Note that the errors for the local projection technique are highest near the boundary, but that away from the boundary the local projection technique actually has less error than the global projection technique for these two cases. We now analyze these techniques in calculating sensitivities for varying discretizations ($N$, $M$) and parameter values ($q$) to understand better how the improved derivative approximations affect our numerical sensitivities.

**Sensitivity Approximations**

Figures (4.5), (4.6), and (4.7) display the sensitivity approximations using the three different derivative approximations against the exact sensitivities for $q = 2.0$, 1.4, and 1.2, respectively. It is clear that the use of a projection technique greatly improves the accuracy of our sensitivity approximations, especially as $q \to 1$. Recall that at $q = 1.4$ and $q = 1.2$ our sensitivity approximations obtained using PWC derivatives were extremely inaccurate so that the approximations do not even show up on the graphs in Figures (4.6) and (4.7) for some values of N. As shown in Figure (4.8), the local projections clearly decrease the $L^2$ error of the sensitivities for a given $N$ and $q$. Moreover, the most promising result is that the projections stabilize the calculations over the parameter range. The local projections give slightly better sensitivity approximations for some values of $q$. Figure (4.9) compares the $L^2$ error of sensitivities calculated using local projections with the error calculated using global projections. Note that as $N$ increases, the global projections do a better job than local projections over a wider range of $q$. This is not surprising since the global projection is the "best" least squares linear approximation (using the finite element basis) of the piecewise continuous finite element derivative in the limit.
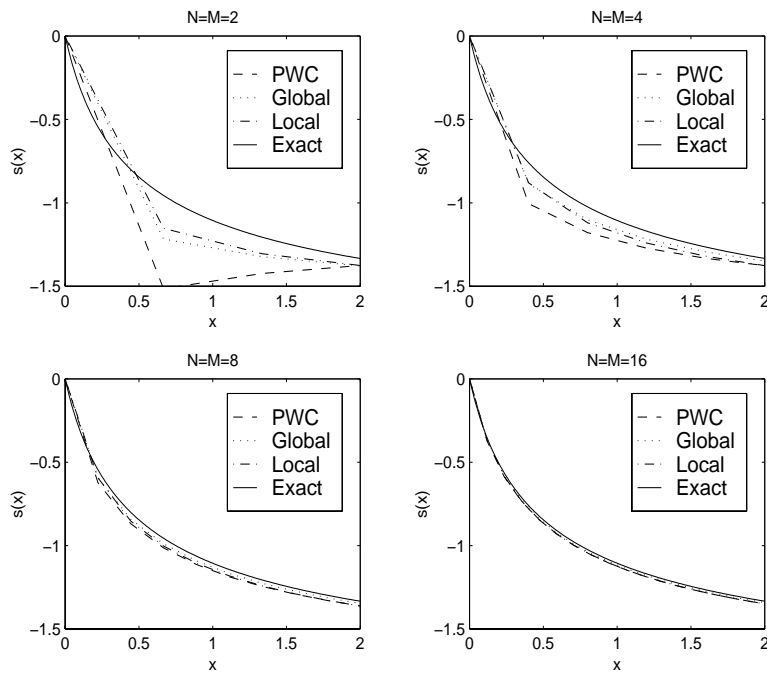
Figure 4.5: Sensitivity Approximations at $q = 2$
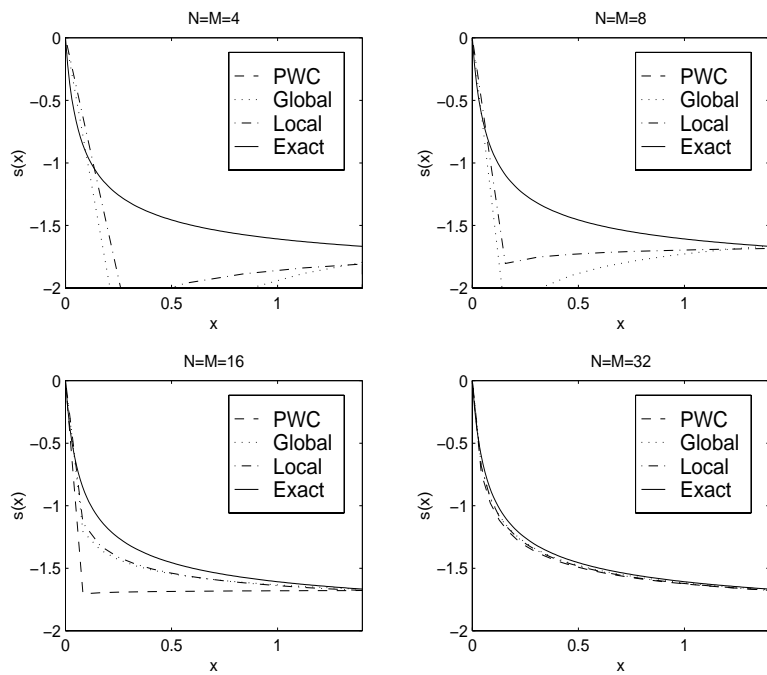


Figure 4.6: Sensitivity Approximations at $q = 1.4$
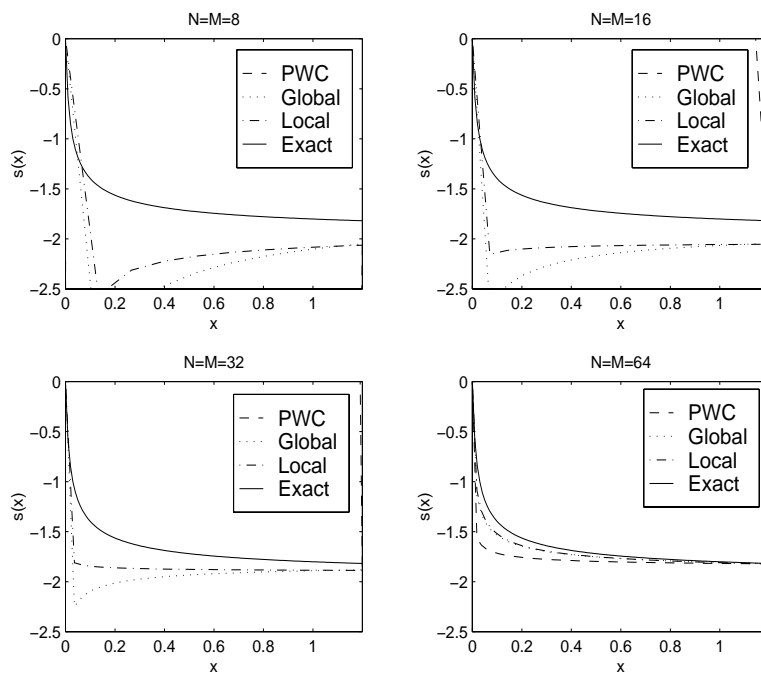
Figure 4.7: Sensitivity Approximations at $q = 1.2$
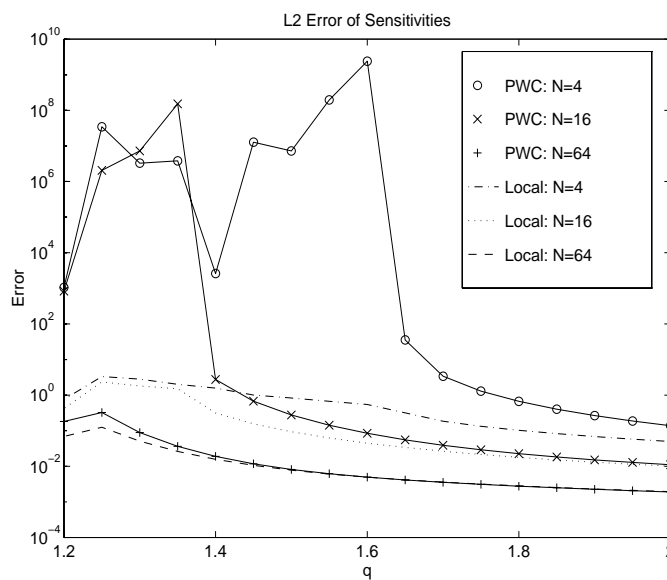


Figure 4.8: Model Problem - $L^2$ Error of Sensitivity Approximations (PWC and Local)
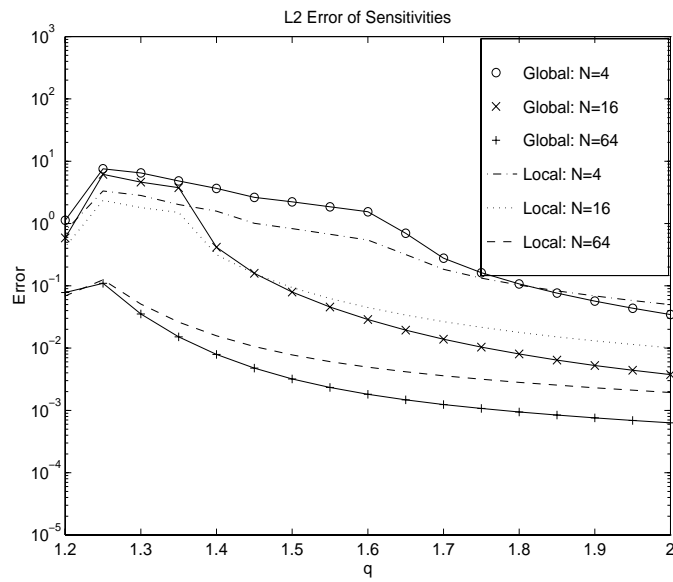
Figure 4.9: Model Problem - $L^2$ Error of Sensitivity Approximations (Global and Local)

**Optimization Results**

We evaluated the same two cases considered in §2.6.2 above: (Case 1) $q_{init} = 1.2$ with $q^* \sim 2$ and (Case 2) $q_{init} = 2.0$ with $q^* \sim 1.4$. This time the simulations were performed using the piecewise linear derivative approximations. Tables (4.2) - (4.5) show the results of these simulations as $N$, $M$ ranged from $2 - 128$. Note that the use of the piecewise linear derivative approximations clearly improved the results of the optimization algorithm for Case 1 and that the global and local projection schemes provided very similar results. The improvement was not as marked for Case 2, however the scheme did converge for the case $N = M = 16$ with the improved gradient approximations and did not with the piecewise constant derivatives. In addition, the local scheme to slightly less time to converge than the global projection scheme (see Tables (4.4) - (4.5)).

Table 4.2: Optimization Results for Case 1, Global Projection Scheme

| N | M | CONV/DNC | ITER | TIME | $\|J^N\|$ | q |
|---|---|---|---|---|---|---|
| 2 | 2 | CONV | 20 | 9.43 | .32586 | 1.9604 |
| 2 | 5 | CONV | 16 | 9.06 | .32548 | 1.9514 |
| 4 | 4 | CONV | 15 | 17.80 | .26463 | 1.9509 |
| 4 | 9 | CONV | 15 | 12.91 | .26468 | 1.9481 |
| 8 | 8 | CONV | 12 | 27.01 | .24327 | 1.9423 |
| 8 | 17 | CONV | 13 | 25.84 | .24326 | 1.9421 |
| 16 | 16 | CONV | 13 | 27.95 | .24235 | 1.9375 |
| 16 | 33 | CONV | 13 | 31.62 | .24235 | 1.9376 |
| 32 | 32 | CONV | 12 | 77.32 | .24272 | 1.9361 |
| 32 | 65 | CONV | 12 | 78.90 | .24273 | 1.9361 |
| 64 | 64 | CONV | 12 | 288.06 | .24274 | 1.9359 |
| 64 | 129 | CONV | 11 | 302.73 | .24274 | 1.9359 |
| 128 | 128 | CONV | 11 | 355.10 | .24285 | 1.9358 |

Table 4.3: Optimization Results for Case 1, Local Projection Scheme

| N | M | CONV/DNC | ITER | TIME | $\|J^N\|$ | q |
|---|---|---|---|---|---|---|
| 2 | 2 | CONV | 20 | 9.43 | .32586 | 1.9604 |
| 2 | 5 | CONV | 16 | 9.06 | .32548 | 1.9514 |
| 4 | 4 | CONV | 15 | 17.80 | .26463 | 1.9509 |
| 4 | 9 | CONV | 15 | 12.91 | .26468 | 1.9481 |
| 8 | 8 | CONV | 12 | 27.01 | .24327 | 1.9423 |
| 8 | 17 | CONV | 13 | 25.84 | .24326 | 1.9421 |
| 16 | 16 | CONV | 13 | 27.95 | .24235 | 1.9375 |
| 16 | 33 | CONV | 13 | 31.62 | .24235 | 1.9376 |
| 32 | 32 | CONV | 12 | 77.32 | .24272 | 1.9361 |
| 32 | 65 | CONV | 12 | 78.90 | .24273 | 1.9361 |
| 64 | 64 | CONV | 12 | 288.06 | .24274 | 1.9359 |
| 64 | 129 | CONV | 11 | 302.73 | .24274 | 1.9359 |
| 128 | 128 | CONV | 11 | 355.10 | .24285 | 1.9358 |

Table 4.4: Optimization Results for Case 2, Global Projection Scheme

| N | M | CONV/DNC | ITER | TIME | $\|J^N\|$ | q |
|---|---|---|---|---|---|---|
| 2 | 2 | DNC | 20 | 2.06 | .93423 | 2.3054 |
| 2 | 5 | DNC | 20 | 64.37 | .72262 | 1.3530 |
| 4 | 4 | DNC | 20 | 133.51 | .53248 | 1.4557 |
| 4 | 9 | DNC | 20 | 132.27 | .52703 | 1.4405 |
| 8 | 8 | DNC | 20 | 127.18 | .48242 | 1.5029 |
| 8 | 17 | DNC | 20 | 143.56 | .50490 | 1.4791 |
| 16 | 16 | CONV | 14 | 28.69 | .41967 | 1.4329 |
| 16 | 33 | CONV | 15 | 33.07 | .41973 | 1.4322 |
| 32 | 32 | CONV | 15 | 13.22 | .41001 | 1.4197 |
| 32 | 65 | CONV | 11 | 12.75 | .41001 | 1.4196 |
| 64 | 64 | CONV | 13 | 45.43 | .40819 | 1.4168 |
| 64 | 129 | CONV | 14 | 48.74 | .40819 | 1.4167 |
| 128 | 128 | CONV | 13 | 258.93 | .40774 | 1.4160 |

Table 4.5: Optimizations Results for Case 2, Local Projection Scheme

| N | M | CONV/DNC | ITER | TIME | $\|J^N\|$ | q |
|---|---|---|---|---|---|---|
| 2 | 2 | DNC | 20 | 60.82 | .73574 | 1.4154 |
| 2 | 5 | DNC | 20 | 64.60 | .72273 | 1.3634 |
| 4 | 4 | DNC | 20 | 130.65 | .53216 | 1.4558 |
| 4 | 9 | DNC | 20 | 131.56 | .52991 | 1.4513 |
| 8 | 8 | DNC | 20 | 133.27 | .50451 | 1.4830 |
| 8 | 17 | DNC | 20 | 133.92 | .50470 | 1.4807 |
| 16 | 16 | CONV | 14 | 30.00 | .41968 | 1.4328 |
| 16 | 33 | CONV | 16 | 32.03 | .41974 | 1.4322 |
| 32 | 32 | CONV | 11 | 11.21 | .41000 | 1.4197 |
| 32 | 65 | CONV | 15 | 14.34 | .41001 | 1.4196 |
| 64 | 64 | CONV | 13 | 44.45 | .40820 | 1.4168 |
| 64 | 129 | CONV | 12 | 45.70 | .40819 | 1.4168 |
| 128 | 128 | CONV | 12 | 253.86 | .40774 | 1.4160 |
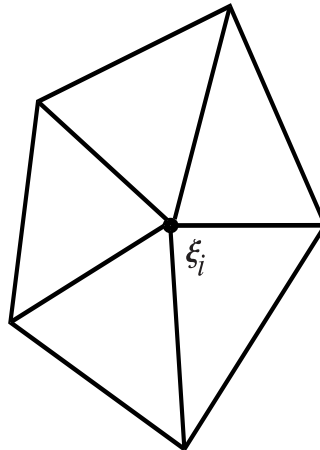
Figure 4.10: Typical Subdomain of an Element Vertex $\boldsymbol{\xi}_i$

## 4.3   A Local Projection for Higher Dimensions

In this section, we describe the local projection scheme employed in the error estimation module of the finite element code described in Chapter 2. This local projection scheme is virtually the same as the local scheme for the 1-D Model Problem discussed in Section 4.2.2 above, yet is presented for a two dimensional problem below since the use of higher order finite elements in higher dimensions adds some complexities which were not previously discussed.

Again, the local projection scheme involves performing a series of projections on subdomains of $\Omega$. The projected gradients are given as polynomial expansions around a given vertex, $\boldsymbol{\xi}_i$, of a finite element mesh. The subdomain, $\Omega_{\xi_i}$, over which the projection consists of all elements having $\boldsymbol{\xi}_i = (x_i, y_i)$ as a vertex. Figure (4.10) illustrates a typical subdomain a finite element mesh of quadratic triangles. In principle, the choice of the degree of the polynomial expansion for the improved gradient approximation is independent of the selection of the finite element basis being used. However, in practice, the degree of the polynomial expansion is chosen to match the degree of the finite element basis employed. This leads to an order of accuracy improvement in the gradient approximations. For all the numerical results presented in Chapter 5 below, quadratic triangular finite elements were used and so the locally projected gradients are written as polynomials of degree two on $\Omega_{\xi_i}$.

At element vertices, $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2$, and $\boldsymbol{\xi}_3$, we define $\mathbf{g}^\star$ to be the local least squares projection of the finite element derivative, $\nabla \mathbf{u}^h$, onto the space of quadratic polynomials. For ease of notation, we denote $\nabla \mathbf{u}^h$ by $\mathbf{g}^h$. Letting $\mathbf{P} = [1, x, y, x^2, xy, y^2]$ denote the basis functions of this space, we can express each component of the gradient projection, $\mathbf{g}_x^\star$ and $\mathbf{g}_y^\star$, as

$$\mathbf{g}_x^\star = \mathbf{P}^T \mathbf{a}_x \quad \text{and,} \quad \mathbf{g}_y^\star = \mathbf{P}^T \mathbf{a}_y \tag{4.14}$$

where the vectors $\mathbf{a}_x, \mathbf{a}_y \in I\!\!R^6$ contain the coefficients of the basis functions. These coefficients are obtained by solving the following least-squares problems:

$$\min \frac{1}{2} \int_{\Omega_{\xi_i}} \left( \mathbf{g}_x^\star - \mathbf{g}_x^h \right) d\Omega \quad \text{and,} \quad \min \frac{1}{2} \int_{\Omega_{\xi_i}} \left( \mathbf{g}_y^\star - \mathbf{g}_y^h \right) d\Omega \tag{4.15}$$

Thus, for each component of $\mathbf{g}^\star$, we solve the following 6 x 6 system of linear equations for $\mathbf{a}_x$ and $\mathbf{a}_y$:

$$\left[ \int_{\Omega_{\xi_i}} \mathbf{P}^T \mathbf{P} d\Omega \right] \{\mathbf{a}_x\} = \left\{ \int_{\Omega_{\xi_i}} \mathbf{P}^T \mathbf{g}_x^h d\Omega \right\} \tag{4.16}$$

$$\left[ \int_{\Omega_{\xi_i}} \mathbf{P}^T \mathbf{P} d\Omega \right] \{\mathbf{a}_y\} = \left\{ \int_{\Omega_{\xi_i}} \mathbf{P}^T \mathbf{g}_y^h d\Omega \right\} . \tag{4.17}$$

The finite element fluxes, $\mathbf{g}_x^h$ and $\mathbf{g}_y^h$, are obtained in the usual manner by differentiating the finite element basis functions. Note that the left hand matrix is independent of the quantity being projected and thus can be viewed as the projection matrix for node, $\boldsymbol{\xi}_i$, and can be used for obtaining locally projected derivative approximations for all of the dependent variables (e.g. $u$, $v$, and $T$) as long as the projection basis, $\mathbf{P}$, is not changed.

Once we are done, we have a quadratic expression for the locally projected derivative, $\mathbf{g}^\star$, at *each* vertex. Let $\mathbf{g}_{\xi_1}^\star$ denote the expression for $\mathbf{g}^\star$ obtained by solving the systems (4.16)-(4.17) where $\Omega_{\xi_1}$ is the subdomain associated with element vertex, $\boldsymbol{\xi}_1$. Define $\mathbf{g}_{\xi_2}^\star$ and $\mathbf{g}_{\xi_3}^\star$ similarly for the remaining element vertices, $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$. We need a *unique* definition of $\mathbf{g}^\star$ for any point, $\boldsymbol{\xi}$ inside the element (see Figure (4.11)). For quadratic elements, there are several ways to do this. I describe the technique employed in the current version of the code.

Unique nodal values of $\mathbf{g}^\star$ at the element vertices, which we denote $\mathbf{g}_1^\star$, $\mathbf{g}_2^\star$, and $\mathbf{g}_3^\star$, are simply defined as follows:

$$\mathbf{g}_i^\star = \mathbf{g}_{\xi_i}^\star(\boldsymbol{\xi_i}), \quad i = 1, 2, 3. \tag{4.18}$$

Nodal values for the midside nodes are obtained by averaging the values of the polynomial expressions for $\mathbf{g}^\star$ at the endpoints of element side. For example,

$$\mathbf{g}^\star(\boldsymbol{\xi}_{12}) = \frac{1}{2} \left( \mathbf{g}_{\xi_1}^\star(\boldsymbol{\xi}_{12}) + \mathbf{g}_{\xi_2}^\star(\boldsymbol{\xi}_{12}) \right) . \tag{4.19}$$
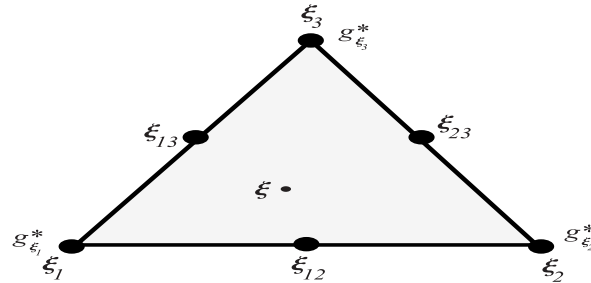
Figure 4.11: Element with Three Quadratic Expressions for $\mathbf{g}^\star$

Then, at any point $\boldsymbol{\xi}$ in the element, the value of the locally projected derivative $\mathbf{g}^\star$ at $\boldsymbol{\xi}$ is

$$\mathbf{g}^\star(\boldsymbol{\xi}) = \sum_{j=1}^{6} N_j \mathbf{g}_j^\star \tag{4.20}$$

where the $N_j$ are the quadratic basis functions for the finite element space and the $\mathbf{g}_j^\star$ are the nodal values of the locally projected derivative obtained as describe above.

In the next chapter, we apply the local projection technique described above in order to obtain improved sensitivity approximations for two flow problems.

# Chapter 5

# Numerical Results for 2-D Problems

We now return to the two specific flow problems described in §3.7 to see if the projection techniques described in §4.3 can be used to obtain improved sensitivity approximations. We begin by considering the cylinder problem discussed in §3.7.1.

## 5.1  Flow Around a Cylinder

Numerical approximations to the state and sensitivities for this problem were calculated over a range of Reynolds numbers. In each case, the results obtained for the initial and first adapted meshes were compared to an approximation which was generated by adapting to a very fine, final mesh and then interpolating the "true" solution from the final mesh onto the initial or first adapted meshes. On this final mesh, both schemes converged to solutions differing by less than $10^{-3}$. For comparison purposes, we chose the unprojected solution on the final mesh as the "true" solution.

We note that the length of the channel was 8 (i.e., $L = 6$) for all of the runs in this section. This length was not sufficient for the flow at the outflow to return to the parabolic inflow especially over the range of Reynolds numbers being considered, however, it was sufficient to meet the computational "free" outflow boundary condition. We will use the $Re = 350$ case to show that the results for the higher Reynolds cases, were not affected by the length of the channel.

In this chapter, we will also use an adaptive technique which refines on sensitivity errors as well as flow errors. This technique is completely analogous to the technique used to refine

on the flow errors (see [9]). In the results which follow, we will be careful to identify meshes which were adapted on approximations of flow errors and meshes which were adapted on approximations of both flow and sensitivity errors.

We present a detailed error analysis of the velocity sensitivities **s** for two Reynolds numbers, $Re = 100$ and $Re = 350$. We begin with the results at $Re = 100$. Figure (5.1) shows the initial mesh, the first adapted meshes, and final mesh for this case. Notice that we present three first adapted meshes: one adapted on the flow only, one adapted on both the flow and the sensitivities calculated with unprojected derivatives, and one adapted on both the flow and sensitivities calculated with projected derivatives. At this Reynolds number, the difference between these three meshes is less dramatic than it is at $Re = 350$, yet we present all of them for consistency and comparison purposes.

As expected, the scheme employing the locally projected derivatives gives better sensitivity results as seen in Figures (5.2) and (5.3). In these figures, recall that the "true" solution is the unprojected solution on the final mesh interpolated onto the initial mesh. Once this interpolation is made, node by node errors can be calculated. These errors are shown in Figure (5.4). From these plots, we can see that the local projection scheme reduces the maximum error by about a factor of 2.

We also look at the sensitivity solutions using schemes on the first adapted mesh. The first adapted mesh (for the flow) is shown in Figure (5.1b). The sensitivity results for the unprojected and locally projected derivative schemes for this mesh are shown in Figures (5.5) and (5.6). In these figures, the "true" solution is now the unprojected solution on the final mesh interpolated onto the first adapted mesh. Note that the difference between the approximations using the projected gradients and those using the "natural" finite element derivatives is no longer as apparent. Again, a node by node comparison was made. These errors are shown in Figures (5.7a,b) - (5.8a,b) and were plotted on the same scales used for the initial mesh for easy comparison. We also plotted nodal errors for the two schemes in the cases where we adapted on the flow *and* the sensitivities (see Figures (5.7c,d) - (5.8c,d)). At this Reynolds number, there does not seem to be a significant advantage gained by adapting on both the flow and sensitivity errors.
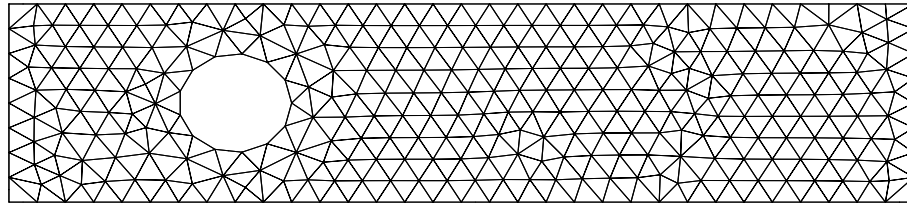
a) Initial Mesh (1219 nodes/563 elements)
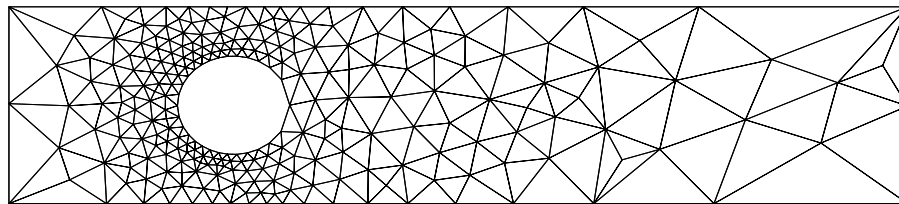


b) First Adapted Mesh - Flow Only (956 nodes/444 elements)
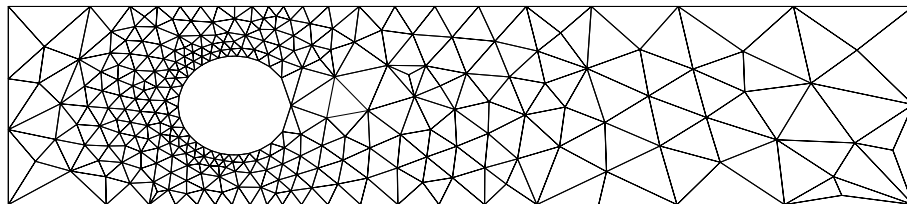


c) First Adapted Mesh - Flow & Unprojected Sens (1082 nodes/504 elements)
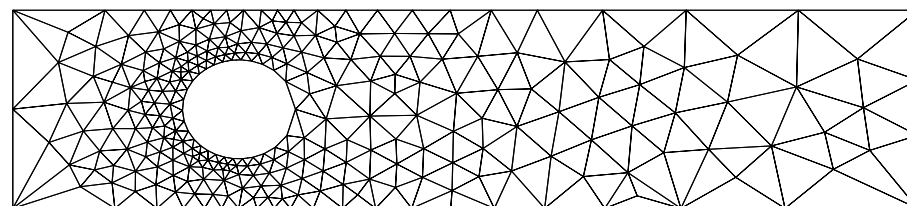


d) First Adapted Mesh - Flow & Projected Sens (1066 nodes/496 elements)
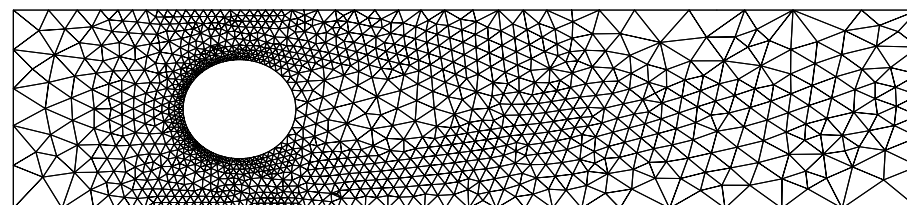


e) Final Mesh (19600 nodes/10818 elements)

Figure 5.1: Meshes for Cylinder Problem at $Re = 100$
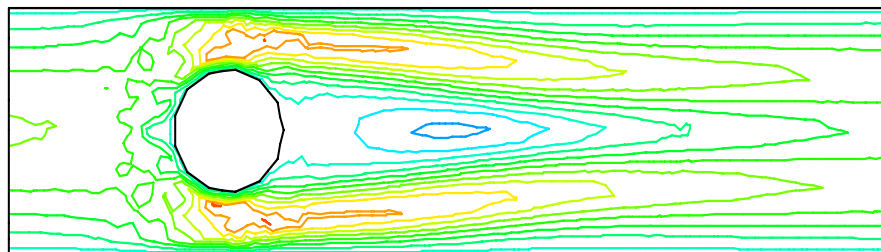
a) Unprojected
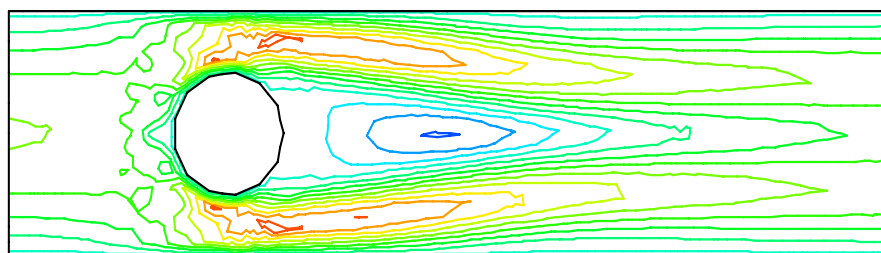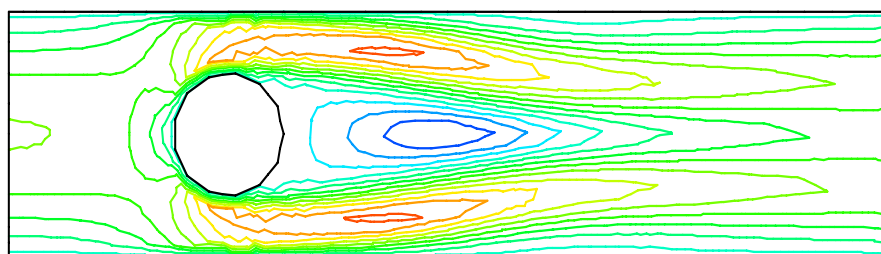
b) Projected
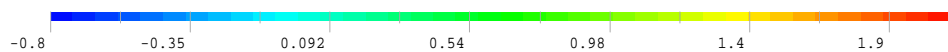
c) "True" Solution

Figure 5.2: u-Velocity Sensitivities on Initial Mesh for $Re = 100$

a) Unprojected

b) Projected
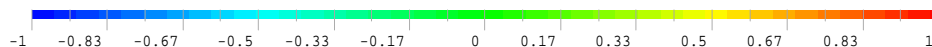
c) "True" Solution

Figure 5.3: v-Velocity Sensitivities on Initial Mesh for $Re = 100$

a) Error in u Sensitivitities - Unprojected Derivatives

0      0.082      0.16      0.24      0.33      0.41      0.49

b) Error in u Sensitivitities - Projected Derivatives

c) Error in v Sensitivitities - Unprojected Derivatives

0      0.085      0.17      0.25      0.34      0.42      0.51

d) Error in v Sensitivitities - Projected Derivatives
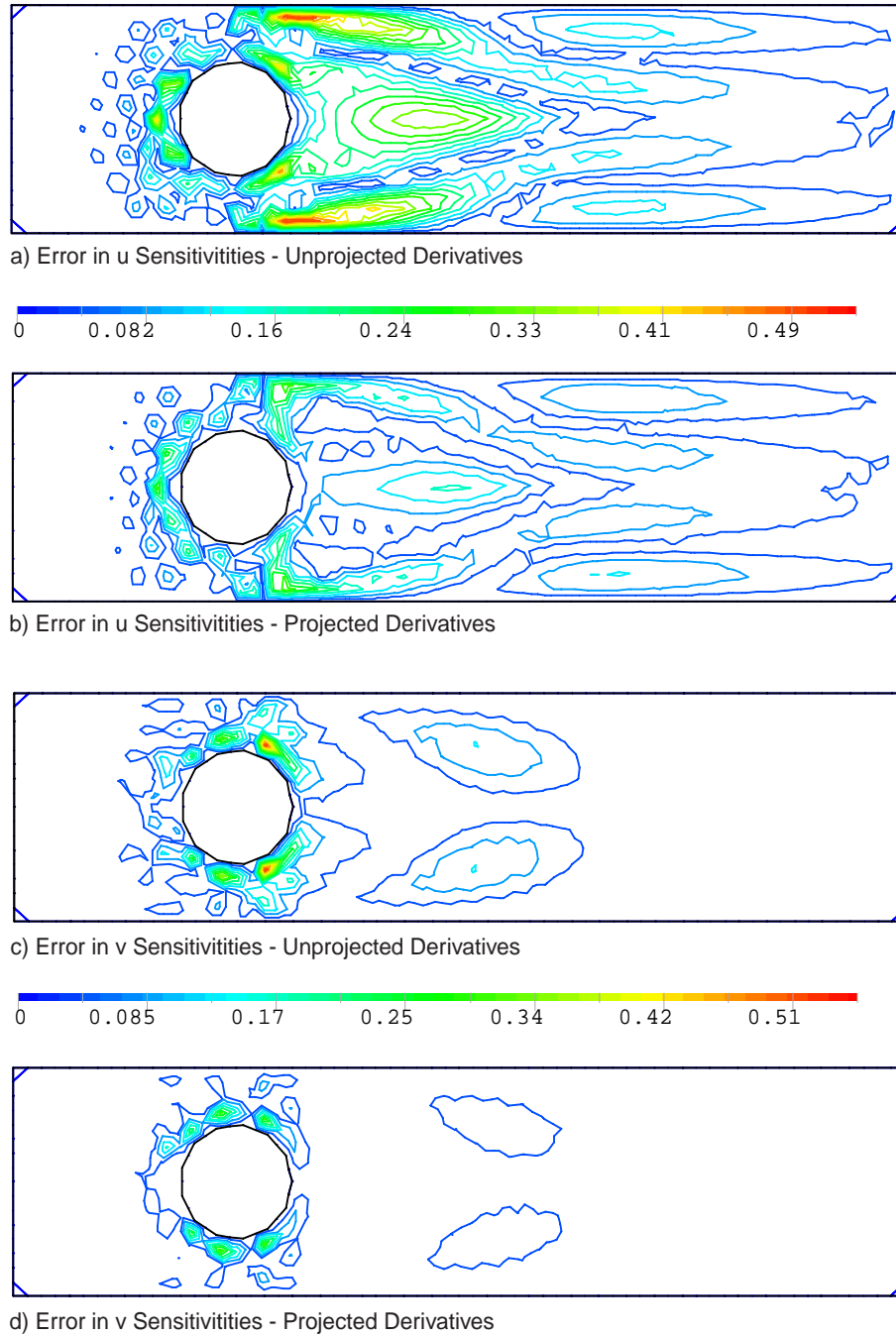
Figure 5.4: Error of Sensitivity Approximations on Initial Mesh for $Re = 100$
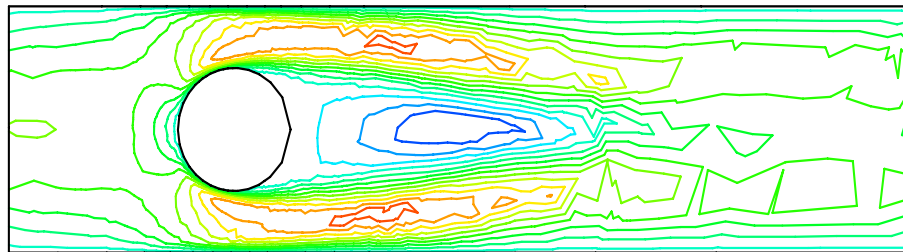
a) Unprojected Derivatives
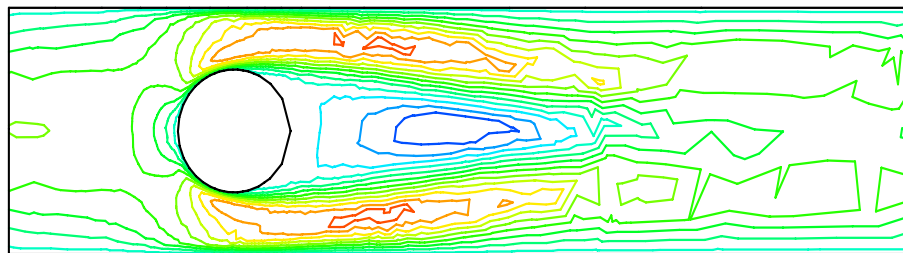


b) Projected Derivatives
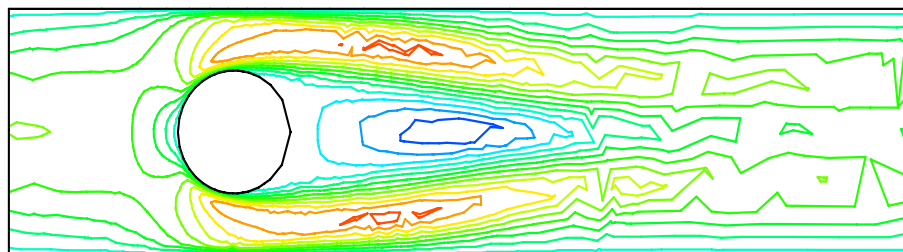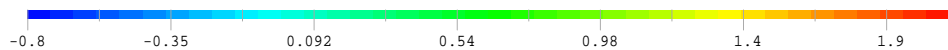


c) "True" Solution



Figure 5.5: u-Velocity Sensitivities on First Adapted Mesh for $Re = 100$

a) Unprojected Derivatives

b) Projected Derivatives

c) "True" Solution

-1    -0.83    -0.67    -0.5    -0.33    -0.17    0    0.17    0.33    0.5    0.67    0.83    1
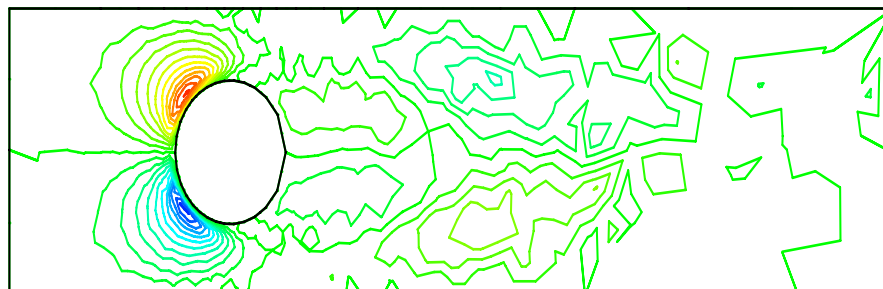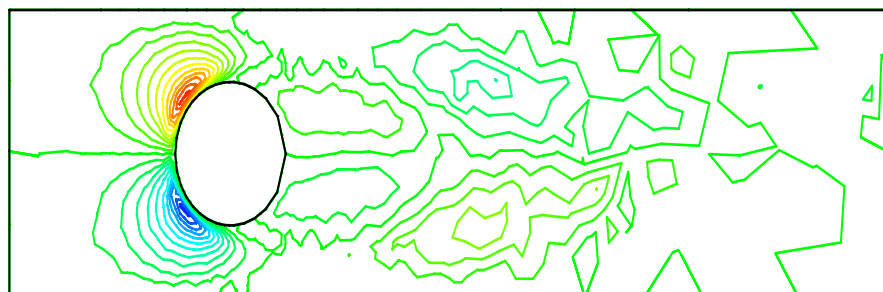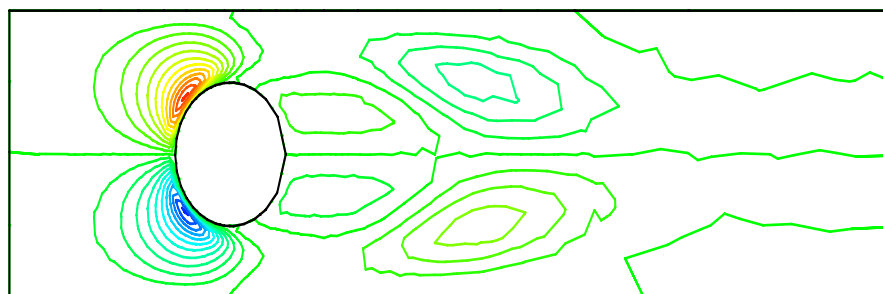
Figure 5.6: v-Velocity Sensitivities on First Adapted Mesh for $Re = 100$
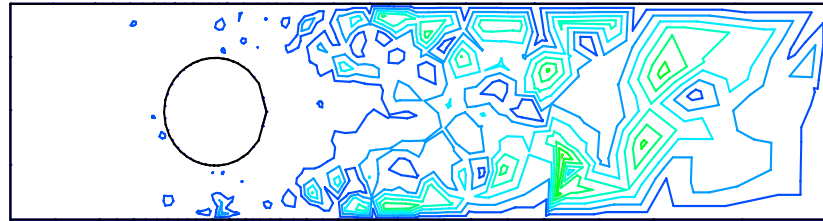
a) Unprojected Derivatives, Adapt on Flow Only

b) Projected Derivatives, Adapt on Flow Only

c) Unprojected Derivatives, Adapt on Flow and Sensitivity

d) Projected Derivatives, Adapt on Flow and Sensitivity

| 0 | 0.082 | 0.16 | 0.24 | 0.33 | 0.41 | 0.49 |
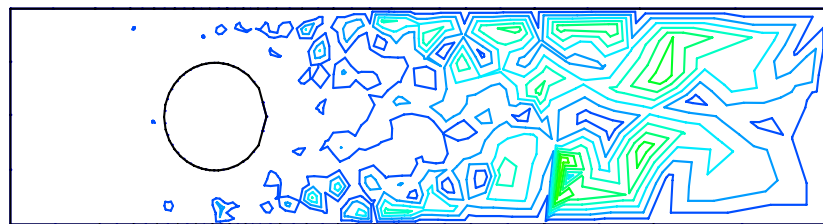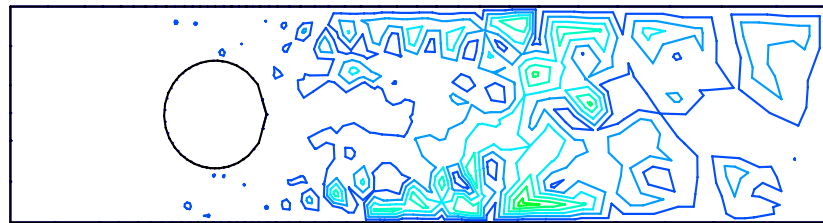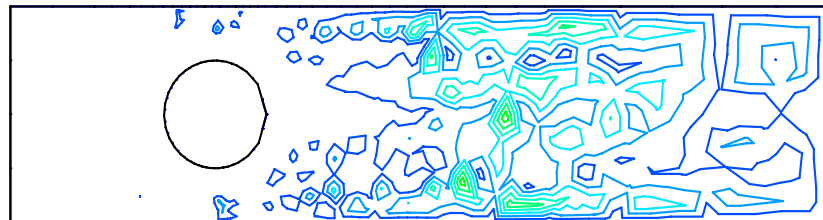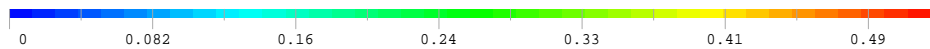
Figure 5.7: Error of u-Velocity Sensitivity Approximations on First Adapted Mesh for $Re = 100$
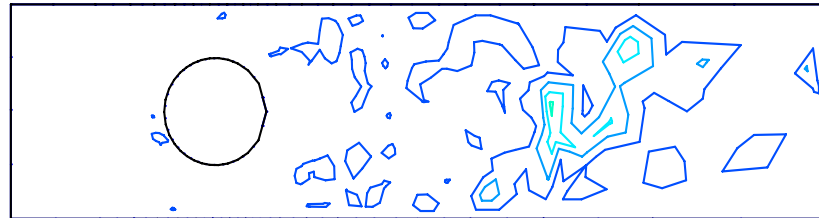
a) Unprojected Derivatives, Adapt on Flow Only

b) Projected Derivatives, Adapt on Flow Only

c) Unprojected Derivatives, Adapt on Flow and Sensitivity

d) Projected Derivatives, Adapt on Flow and Sensitivity

| 0 | 0.085 | 0.17 | 0.25 | 0.34 | 0.42 | 0.51 |
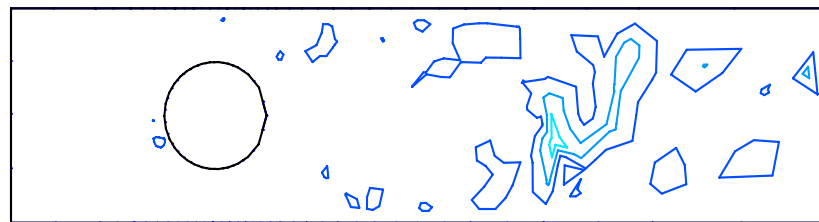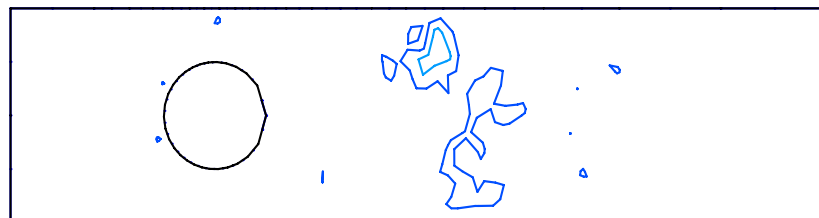
Figure 5.8:  Error of v-Velocity Sensitivity Approximations on First Adapted Mesh for $Re = 100$

Table 5.1: $L^2$ Errors for Flow and Sensitivities at $Re = 100$

| | Initial Mesh | | | |
|---|---|---|---|---|
| | Unproj | Proj | | |
| $u$ | 1.7397E-01 | 1.7397E-01 | | |
| $v$ | 9.0160E-02 | 9.0160E-02 | | |
| $s_u$ | 5.0463E-01 | 2.8227E-01 | | |
| $s_v$ | 2.2453E-01 | 1.3050E-01 | | |
| | First Adapted Meshes | | | |
| | Unproj (Flow) | Proj (Flow) | Unproj (Flow & Sens) | Proj (Flow & Sens) |
| $u$ | 1.0313E-01 | 1.0313E-01 | 7.2637E-02 | 5.3305E-02 |
| $v$ | 4.5997E-02 | 4.5997E-02 | 3.3000E-02 | 2.8115E-02 |
| $s_u$ | 3.4173E-01 | 3.5800E-01 | 2.5988E-01 | 2.7273E-01 |
| $s_v$ | 1.4487E-01 | 1.1039E-01 | 9.1075E-02 | 7.6292E-02 |

In order, to get an overall evaluation of the errors, we used the node by node errors to calculate an $L^2$ error over the entire domain for the flow and the sensitivities (see Table (5.1)). This more clearly shows the 50% reduction of the error for the local projection scheme on the initial mesh. Note that the errors for $u, v$ on the first adapted meshes are the same for the unprojected and projected schemes on the mesh that was refined only on the flow, but are slightly different (since the meshes are slightly different) when the meshes were also refined on the sensitivity. Note that adapting on the flow *and* sensitivity errors not only improved the sensitivity approximations over those obtained by adapting on the flow alone, but also improved the numerical approximations for the flow.

We now turn to the results for $Re = 350$. We begin by comparing the state and sensitivity approximations for $L = 6$ and $L = 15$ to ascertain whether or not the length of the channel affects our results. Figure (5.9) shows the initial meshes for the two channel lengths as well as the $u, v$ velocity contours. It is clear that the flow more nearly returns to the inflow for the longer channel. However, the results for the shorter channel are very similar to those for the longer channel in the areas where they overlap. In fact, Figures (5.10) and (5.11) show that the dramatic difference between the sensitivities obtained with unprojected derivatives and those obtained using locally projected derivatives occur in both the long and the shorter channels. This gives us reasonable confidence that the length of the channel is not affecting our results.

We now return to complete the same error analysis for $Re = 350$ that we did for $Re = 100$. The initial and first adapted meshes are shown in Figure (5.12). Note that the difference between the three first adapted meshes is definitely greater at this Reynolds

number. This is due, at least in part, to the greater discrepancy between the flow error, the unprojected sensitivity error, and the projected sensitivity error. At this Reynolds number, the differences between the sensitivities calculated with the two different derivative schemes is now dramatic as can be seen in Figures (5.13) - (5.14). The node by node error analysis (see Figure (5.15)) shows that locally projected derivatives are definitely better, although at this Reynolds number, both approximations contain fairly larger errors. Using the locally projected derivatives to calculate sensitivities reduces the overall $L^2$ error by about 600% (see Table (5.2)).

The same analysis is done on the first adapted meshes (See Figures (5.16) - (5.19)). Here, adapting on the flow as well as the sensitivity makes a greater difference in the error reduction from the initial mesh to the first adapted mesh. This is easily seen in the overall $L^2$ errors for the approximate flow and sensitivities displayed in Table (5.2). It is interesting to note that the errors for *all* the quantities are less for the Projected (Flow & Sensitivity) than they are for the Unprojected (Flow & Sensitivity) despite the fact that the Unprojected (Flow & Sensitivity) mesh is quite a bit finer.

To conclude, we observe that using the locally projected derivatives clearly stabilizes the calculations over a larger range of Reynolds numbers as it did in §4.2.3 for the 1-D model problem. Plots of the overall $L^2$ sensitivity errors on the initial mesh are shown in Figures (5.20) - (5.21). In addition, the mesh refinement very effectively reduces the errors for the flow and the sensitivities. At most Reynolds numbers, after the mesh is refined once, there is little difference between the sensitivities calculated with the unprojected and locally projected techniques. At higher Reynolds numbers, however, using locally projected derivatives to calculate sensitivities remains advantageous.

We now turn our attention to obtaining numerical approximations to the design sensitivities for the flow over a bump problem.

Table 5.2: $L^2$ Errors for Flow and Sensitivities at $Re = 350$

| | Initial Mesh | | | |
|---|---|---|---|---|
| | Unproj | Proj | | |
| $u$ | 1.4808E-00 | 1.4808E-00 | | |
| $v$ | 4.1024E-01 | 4.1024E-01 | | |
| $s_u$ | 1.9438E+01 | 3.6791E-00 | | |
| $s_v$ | 3.0582E-00 | 5.8348E-01 | | |
| | First Adapted Meshes | | | |
| | Unproj (Flow) | Proj (Flow) | Unproj (Flow & Sens) | Proj (Flow & Sens) |
| $u$ | 9.6580E-01 | 9.6580E-01 | 3.5985E-01 | 2.8674E-01 |
| $v$ | 2.6044E-01 | 2.6044E-01 | 1.0257E-01 | 7.7747E-02 |
| $s_u$ | 3.8080E-00 | 3.1614E-00 | 1.1145E-00 | 1.0423E-00 |
| $s_v$ | 1.5339E-00 | 8.7266E-01 | 3.4686E-01 | 2.9430E-01 |

a) Initial Mesh - L = 6



b) Initial Mesh - L = 15



c) u Velocity Contours - L = 6



d) u Velocity Contours - L = 15



e) v Velocity Contours - L = 6



f) v Velocity Contours - L = 15
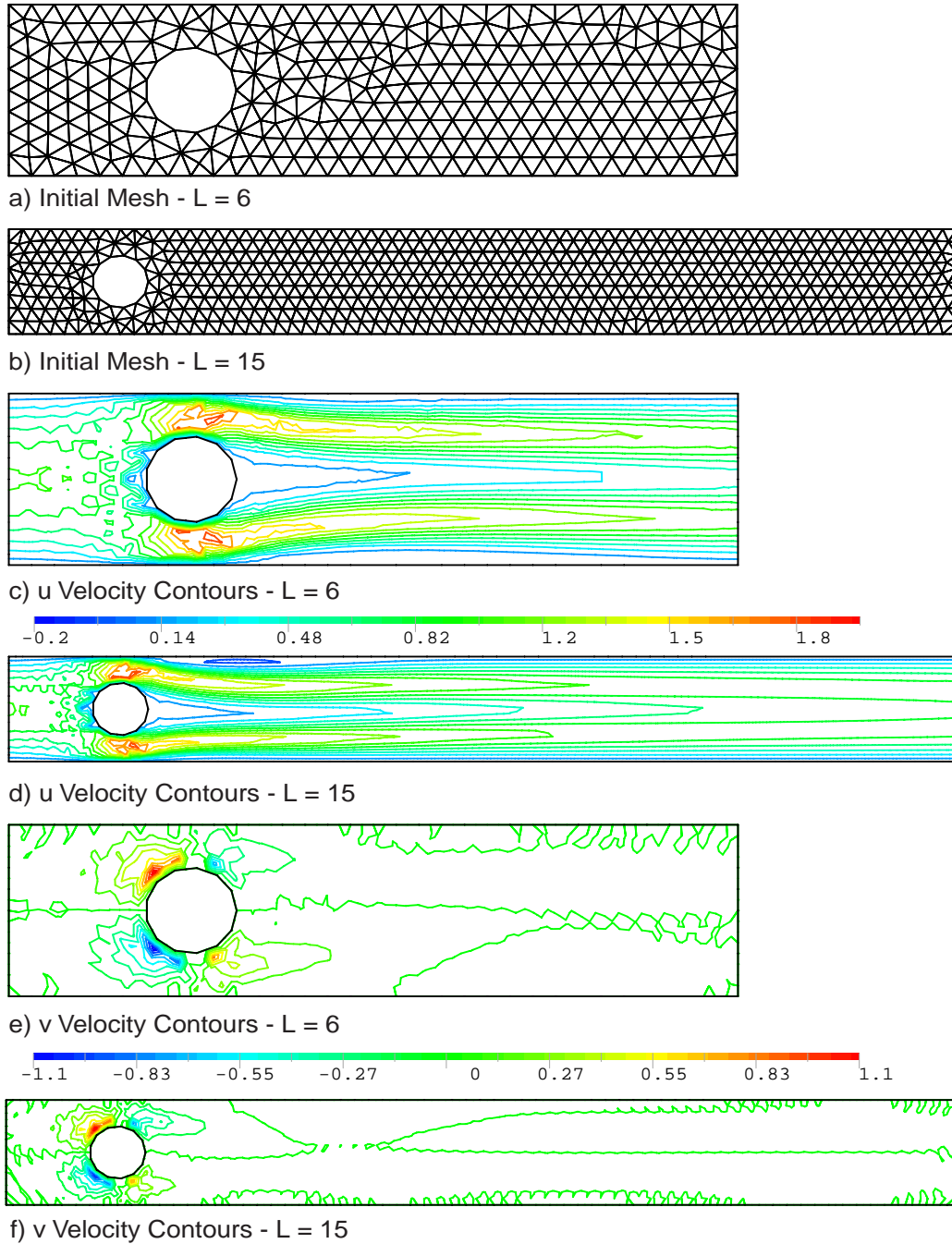
Figure 5.9: Initial Meshes and u,v-Velocity Contours for $L = 6, 15$ and $Re = 350$

a) u Velocity Sensitivity Contours, Unprojected Derivatives: L = 6
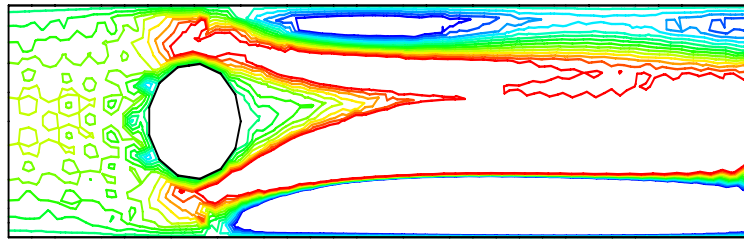
-0.95      -0.53      -0.1      0.32      0.74      1.2      1.6
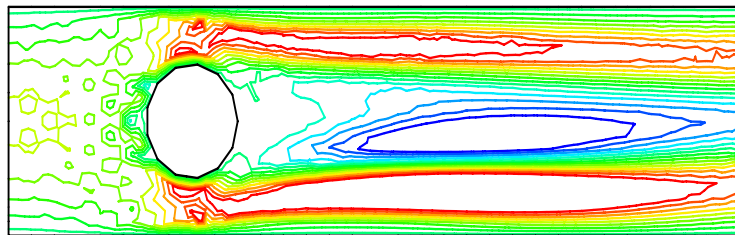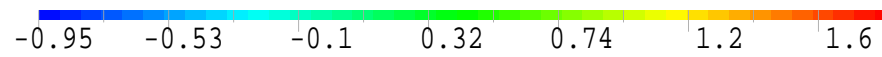


b) u Velocity Sensitivity Contours, Projected Derivatives: L = 6



c) u Velocity Sensitivity Contours, Unprojected Derivatives: L = 15

-0.95      -0.53      -0.1      0.32      0.74      1.2      1.6
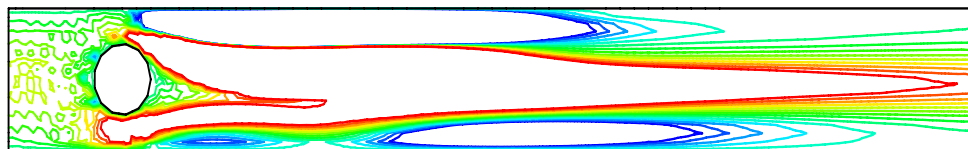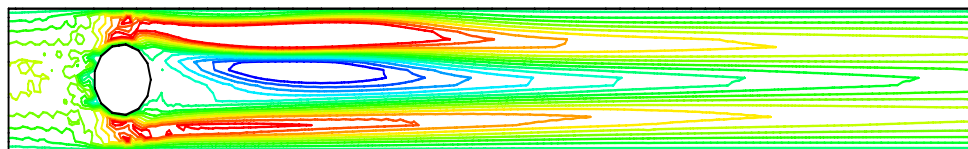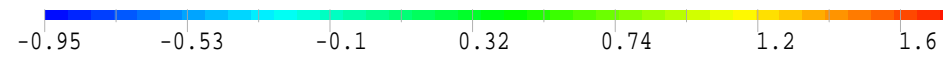


d) u Velocity Sensitivity Contours, Projected Derivatives: L = 15

Figure 5.10: u-Velocity Sensitivity Contours for $L = 6, 15$ and $Re = 350$

a) v Velocity Sensitivity Contours, Unprojected Derivatives: L = 6

-0.8 -0.64 -0.48 -0.32 -0.16     0   0.16   0.32   0.48   0.64   0.8
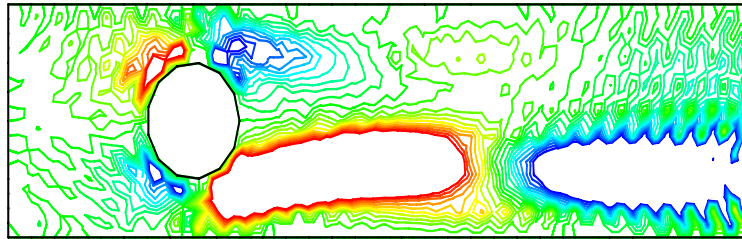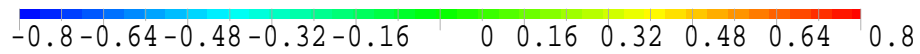
b) v Velocity Sensitivity Contours, Projected Derivatives: L = 6

c) v Velocity Sensitivity Contours, Unprojected Derivatives: L = 15

-0.8   -0.64   -0.48   -0.32   -0.16     0   0.16   0.32   0.48   0.64   0.8
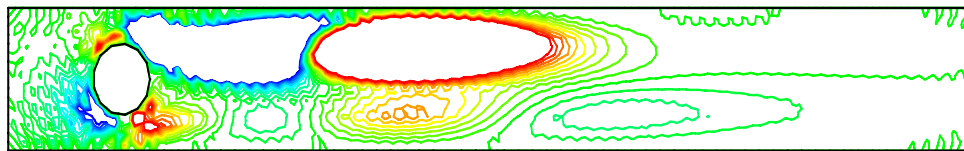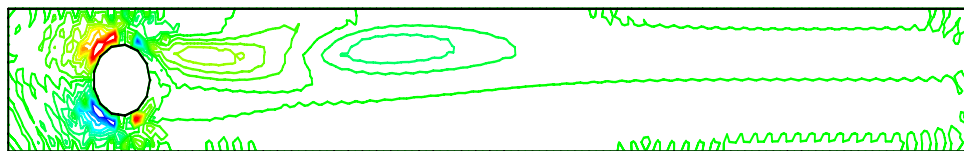
d) v Velocity Sensitivity Contours, Projected Derivatives: L = 15

Figure 5.11: v-Velocity Sensitivity Contours for $L = 6, 15$ and $Re = 350$
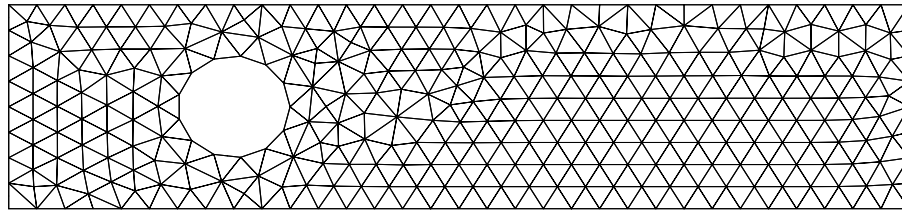
a) Initial Mesh (1219 nodes/563 elements)
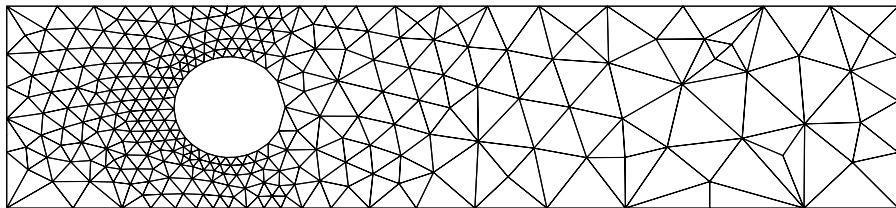


b) First Adapted Mesh - Flow Only (1164 nodes/542 elements)
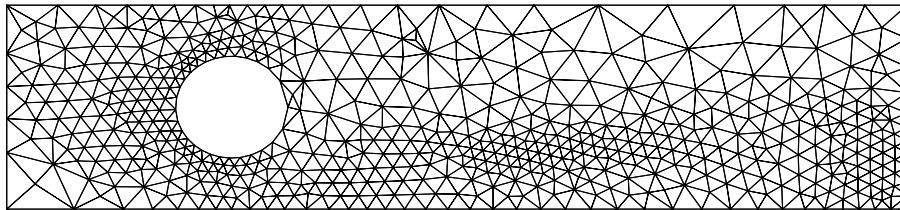


c) First Adapted Mesh - Flow &Unprojected Sens (2249 nodes/1071 elements)
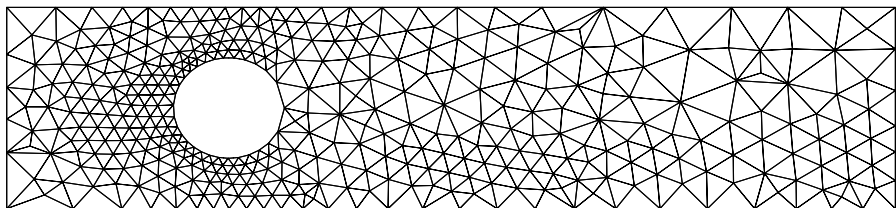


d) First Adapted Mesh - Flow & Projected Sens (1556 nodes/732 elements)
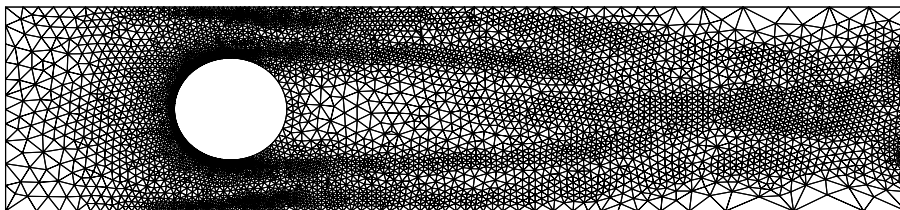


e) Final Mesh -  (21998 nodes/10818 elements)

Figure 5.12: Meshes for Cylinder Problem at $Re = 350$

a) Unprojected Derivatives

b) Projected Derivatives

c) "True" Solution

-0.95    -0.53    -0.1    0.32    0.74    1.2    1.6
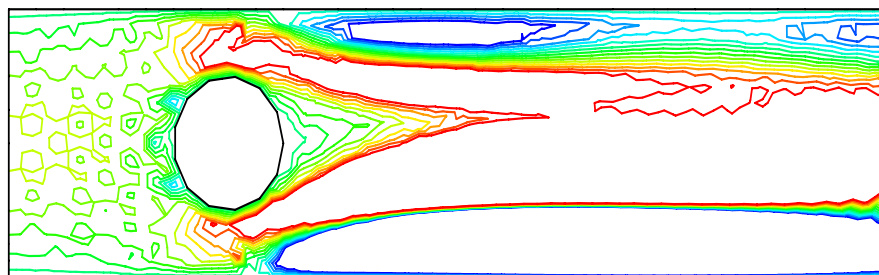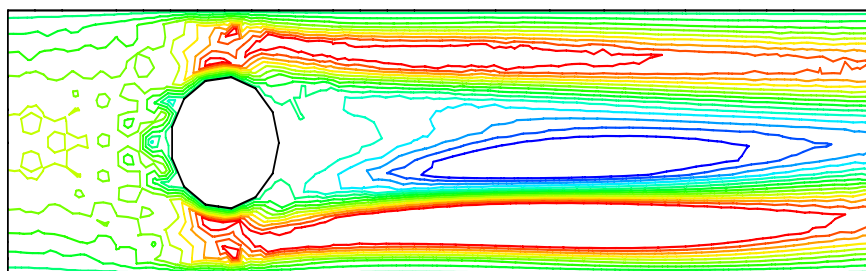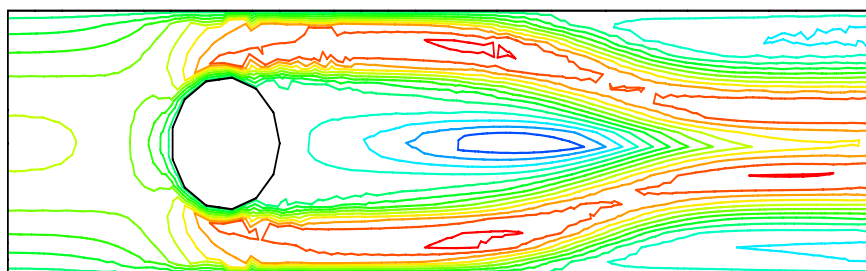
Figure 5.13: u-Velocity Sensitivities on Initial Mesh for $Re = 350$

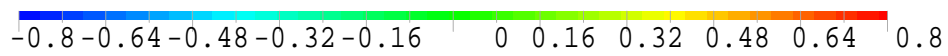a) Unprojected Derivatives

b) Projected Derivatives

c) "True" Solution

−0.8 −0.64 −0.48 −0.32 −0.16   0  0.16  0.32  0.48  0.64   0.8
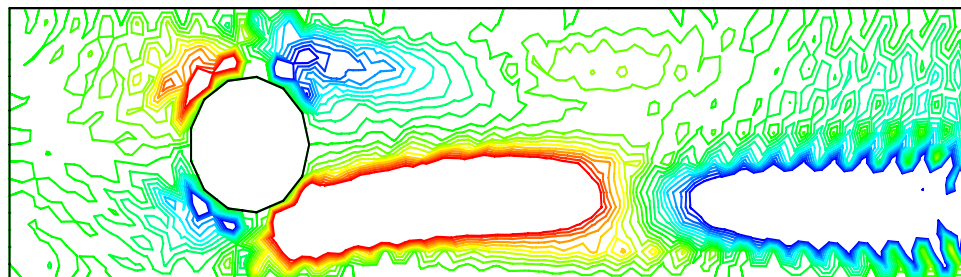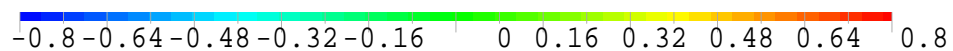
Figure 5.14: v-Velocity Sensitivities on Initial Mesh for $Re = 350$
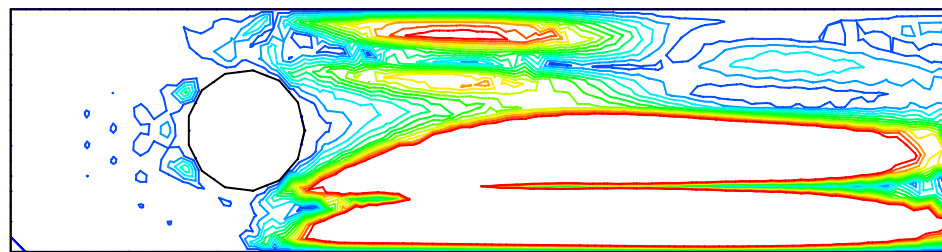
a) Error in u Sensitivities - Unprojected Derivatives

0    0.46    0.92    1.4    1.8    2.3    2.8
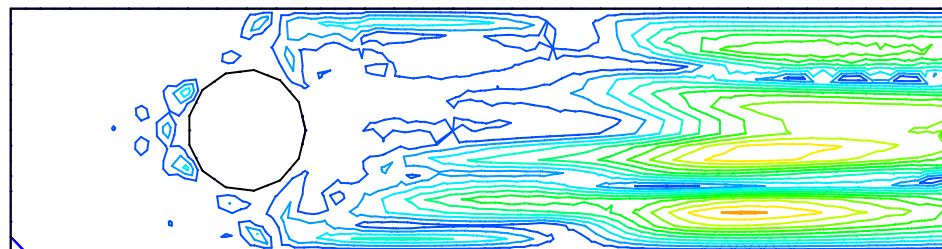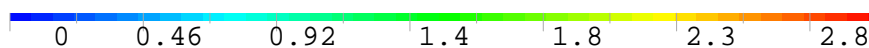
b) Error in u Sensitivities - Projected Derivatives
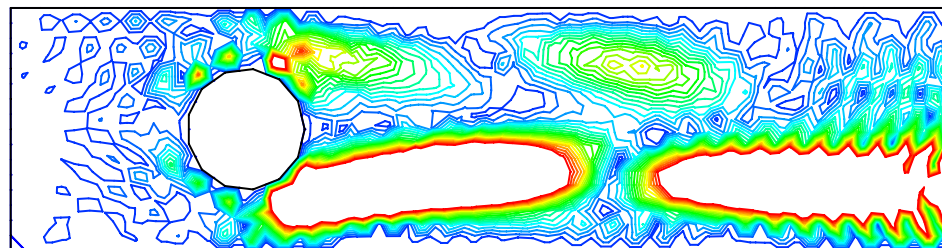
c) Error in v Sensitivities - Unprojected Derivatives

0   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1
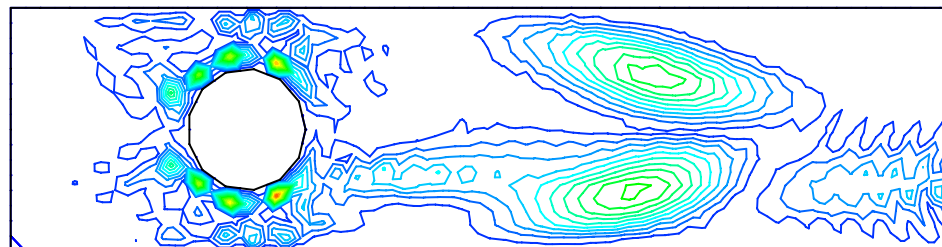
d) Error in v Sensitivities - Projected Derivatives

Figure 5.15: Error of Sensitivity Approximations on Initial Mesh for $Re = 350$

a) Unprojected Derivatives

b) Projected Derivatives

c) "True" Solution

-0.95     -0.53     -0.1     0.32     0.74     1.2     1.6
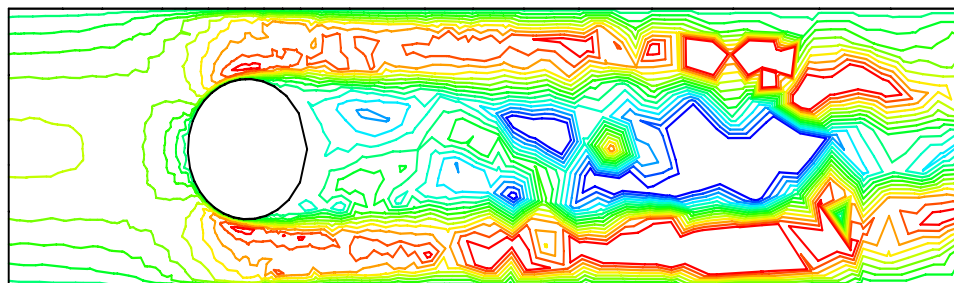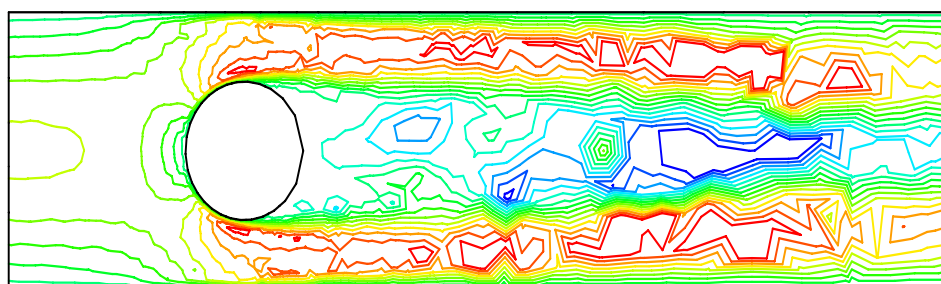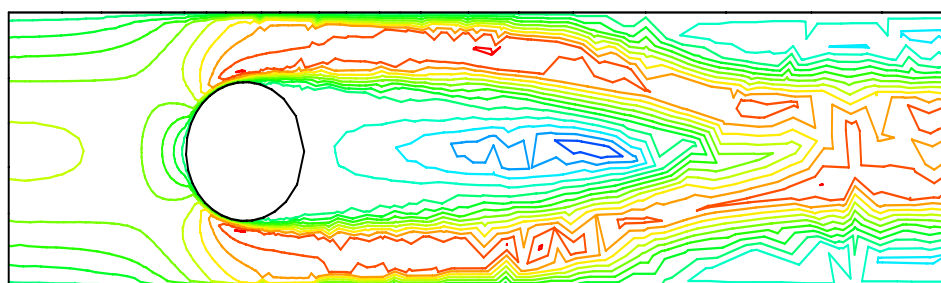
Figure 5.16: u-Velocity Sensitivities on First Adapted Mesh for $Re = 350$

a) Unprojected Derivatives

b) Projected Derivatives

c) "True" Solution

−0.8 −0.64 −0.48 −0.32 −0.16    0  0.16  0.32  0.48  0.64    0.8
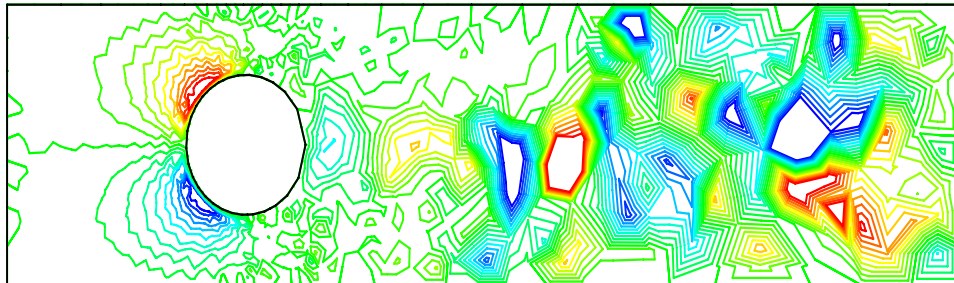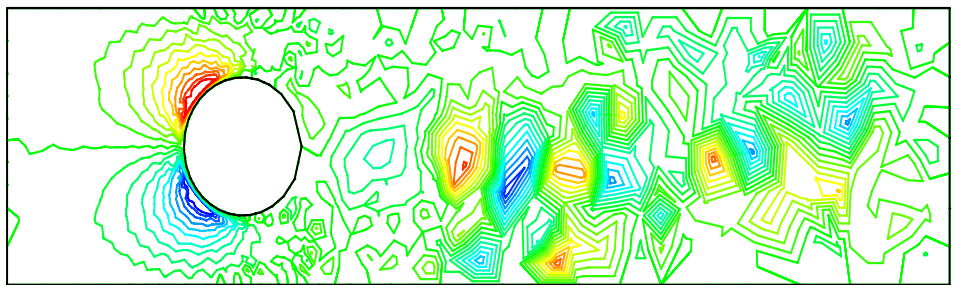
Figure 5.17: v-Velocity Sensitivities on First Adapted Mesh for $Re = 350$

a) Unprojected Derivatives, Adapt on Flow Only

b) Projected Derivatives, Adapt on Flow Only

c) Unprojected Derivatives, Adapt on Flow and Sensitivity

d) Projected Derivatives, Adapt on Flow and Sensitivity

| 0 | 0.46 | 0.92 | 1.4 | 1.8 | 2.3 | 2.8 |
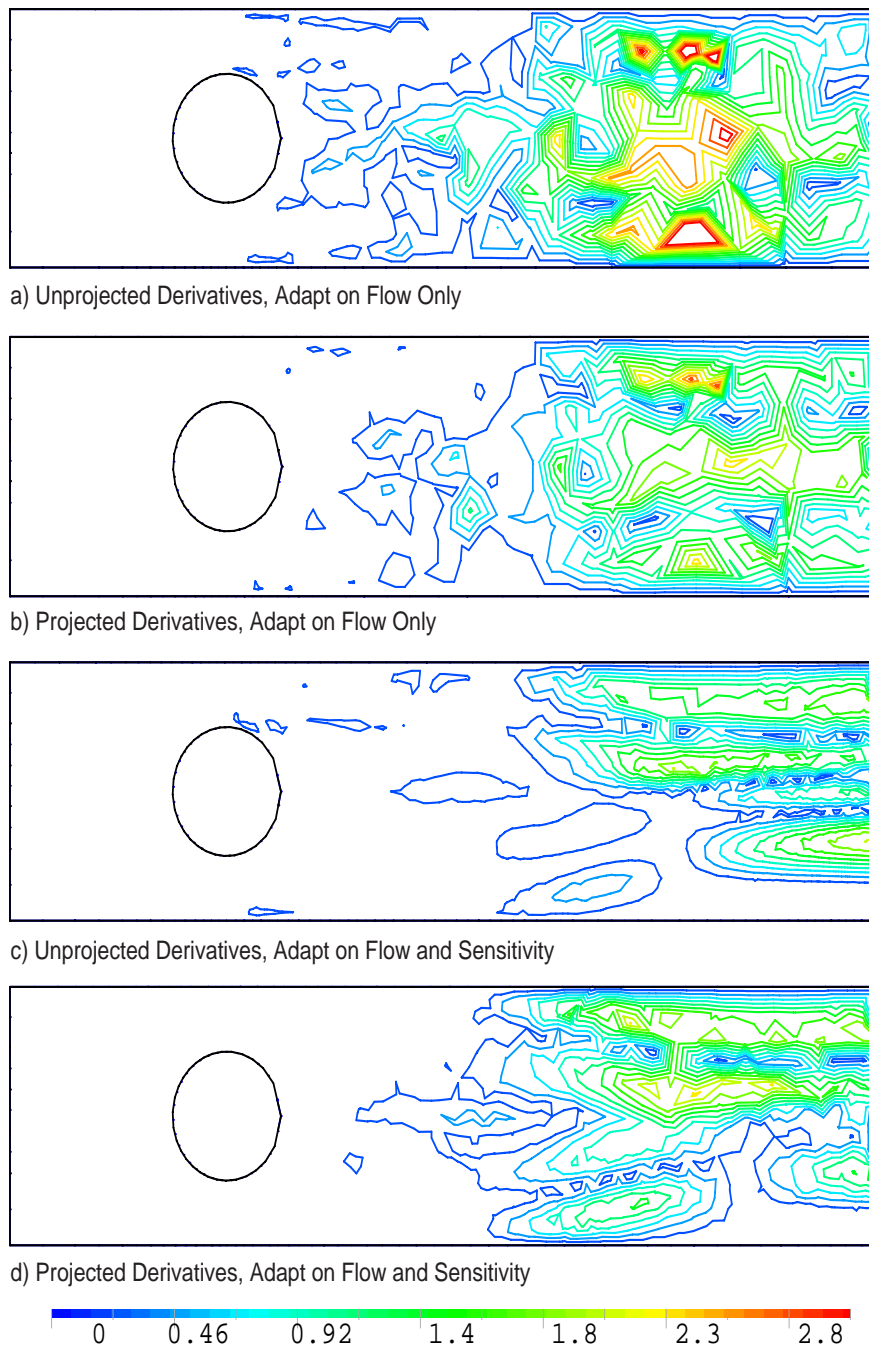
Figure 5.18: Error of u-Velocity Sensitivity Approximations on First Adapted Mesh for $Re = 350$

a) Unprojected Derivatives, Adapt on Flow Only

b) Projected Derivatives, Adapt on Flow Only

c) Unprojected Derivatives, Adapt on Flow and Sensitivity

d) Projected Derivatives, Adapt on Flow and Sensitivity
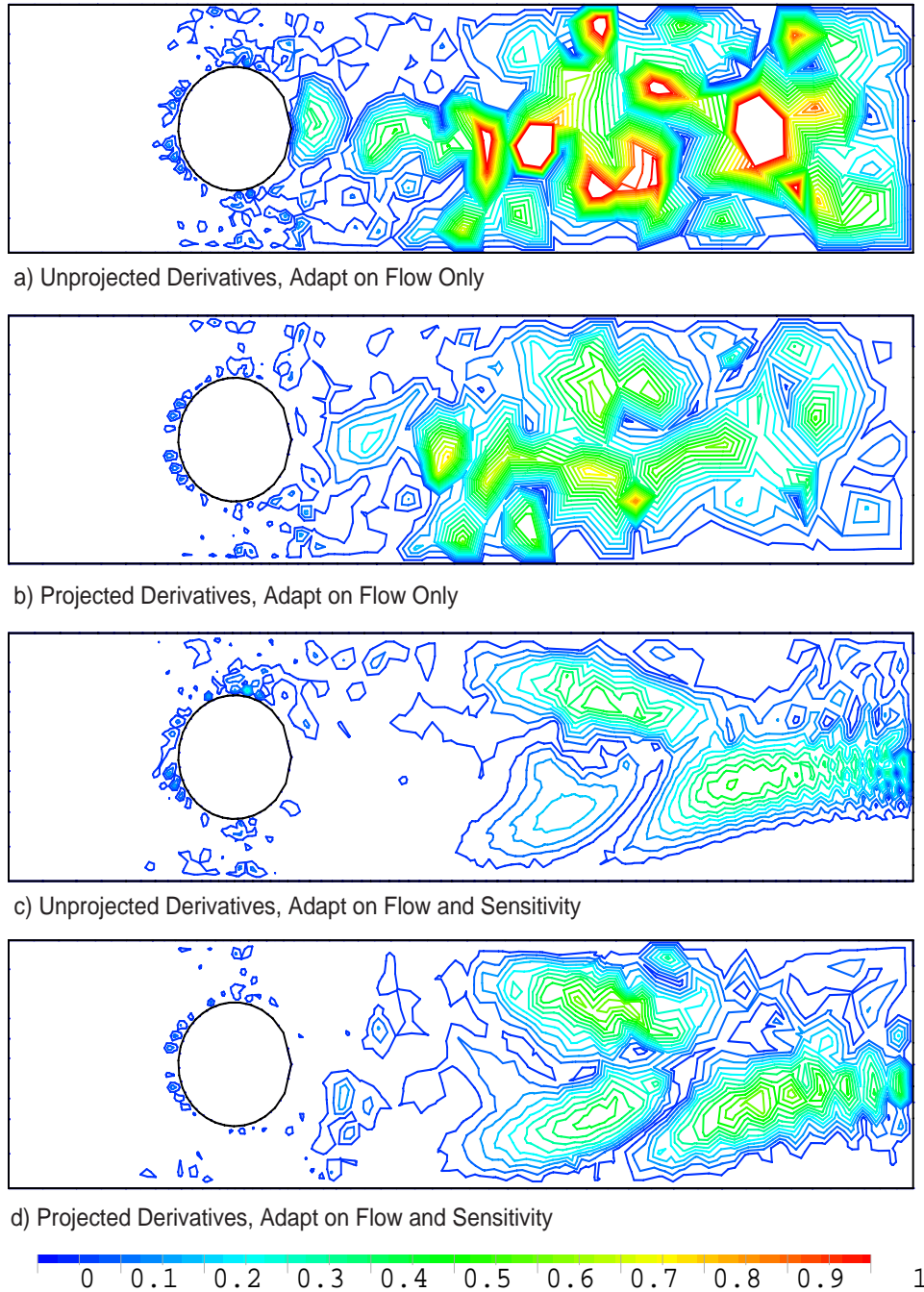
Figure 5.19: Error of v-Velocity Sensitivity Approximations on First Adapted Mesh for $Re = 350$
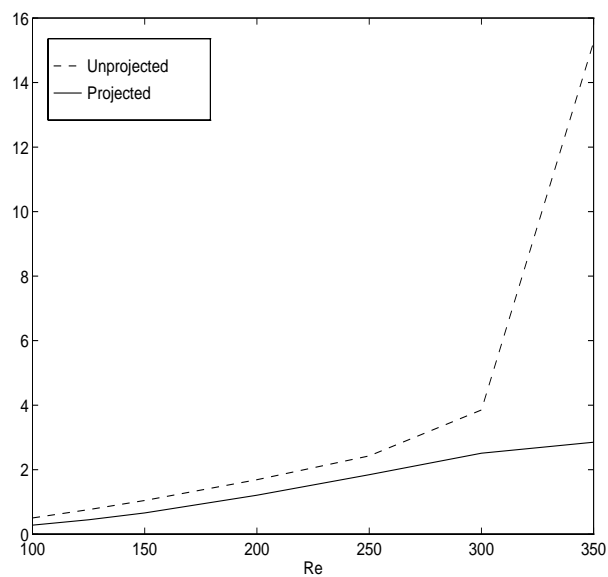
Figure 5.20: Flow About Cylinder - $L^2$ Error of u-Sensitivity Approximations on Initial Mesh
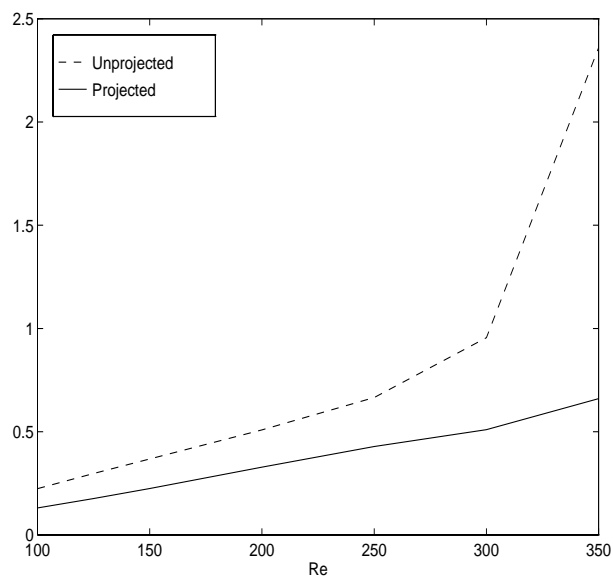


Figure 5.21: Flow About Cylinder - $L^2$ Error of v-Sensitivity Approximations on Initial Mesh

## 5.2  Flow over a Bump

We begin by doing a qualitative comparison of the sensitivity values we obtain, with those presented in [11]. We will point out some of the difficulties of calculating shape sensitivities and show how the process of error estimation and grid refinement is extremely important in obtaining accurate numerical approximations of the sensitivities for these problems.

Figure (5.22) displays vector sensitivity plots for $\boldsymbol{\alpha} = \alpha_1 = 0.5$, $\lambda = 0.5$, and $L = 10$. These plots are qualitatively very comparable to those presented on page 172 in [11]. As Burkardt notes, the shape sensitivities for smaller Reynolds numbers appear as "whirlpools" and are predominately localized to the region above the bump. As the Reynolds number increases, however, the effect of the bump is carried downstream. This is especially apparent in Figure (5.22c).

Another effect of increasing the Reynolds number on flow calculations, is that the task of meeting the outflow boundary conditions becomes more challenging numerically. We clearly see this effect in the $Re = 500$ case shown in Figure (5.23). With a channel length of $L = 10$, the flow was not able reach the parabolic flow profile of the inflow. So for this case, we lengthened the channel to $L = 20$. Notice also that we are getting a large secondary "whirlpool" further downstream from the bump. This phenomenon did not appear in the sensitivities presented in [11]. We will examine this further for the case $Re = 1000$.

We used the case of $Re = 1000$ to examine a number of issues relating to our sensitivity approximations. First, we investigate the secondary "whirlpool" which appears in our sensitivity calculations. We also evaluate the accuracy of the sensitivity approximations near the bump. We do this by using the error estimation and grid refinement process presented in §3.8.4. The mesh was adapted only on flow errors. Figure(5.24) shows the initial mesh, which is similar in density to the one used in [11], and the adapted meshes. Note that the mesh refines where one would expect, in the regions of large velocity gradients around and downstream from the bump. It is also refining at the outflow in an effort to accurately meet the outflow boundary condition.

Figures (5.25) and (5.26) display $u$ and $v$ contours for the flow on the initial and adapted meshes. It is important to note that as the mesh refines, the contours smooth and the accuracy of the gradients in $u$ and $v$ are greatly improved especially in the vicinity of the bump. This lack of accuracy of the velocity gradients on the initial mesh greatly hampers our ability to obtain good sensitivity approximations. This is seen by noting that the sensitivity boundary conditions on the bump, (3.66) - (3.67) require approximations to the velocity gradients on the boundary. If there are large errors in the velocity gradients, there will be large errors in the sensitivity approximations. This effect can be seen in Figures

(5.27) - (5.29). As the gradient approximations are improved in the second and third adapted meshes, the values of the sensitivities become much more accurate, not only in the local area of the bump, but downstream as well. Note also that the secondary "whirlpool" which appeared in the $Re = 500$ case, is seen on the initial mesh in this case also. As the mesh is refined, however, the size of the "whirlpool" is diminished until it is almost gone as can be seen in Figure (5.29c).
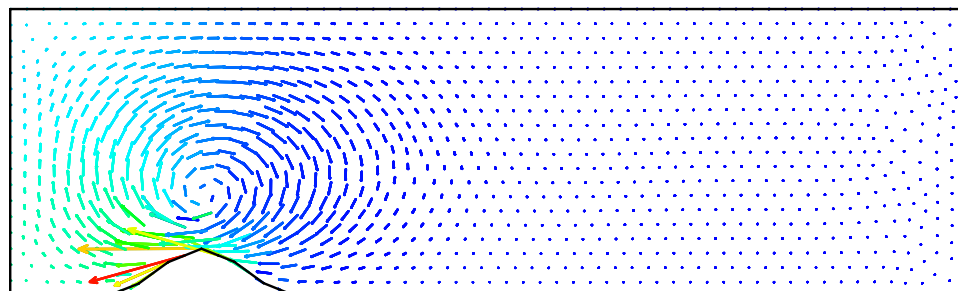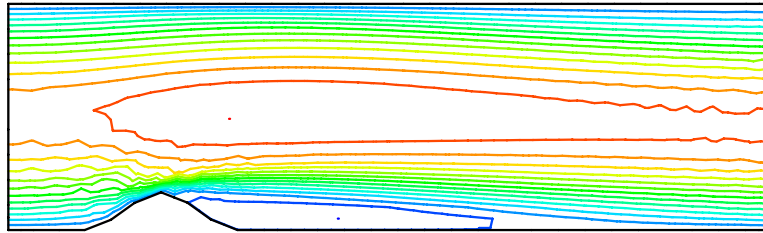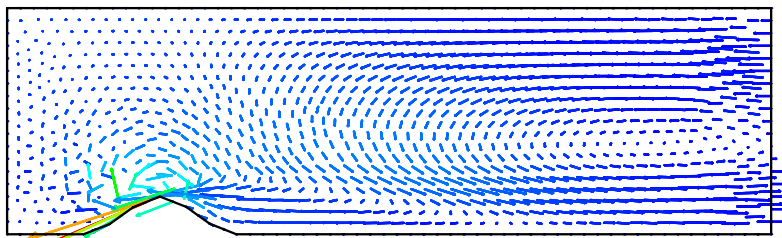
a) Re = 1
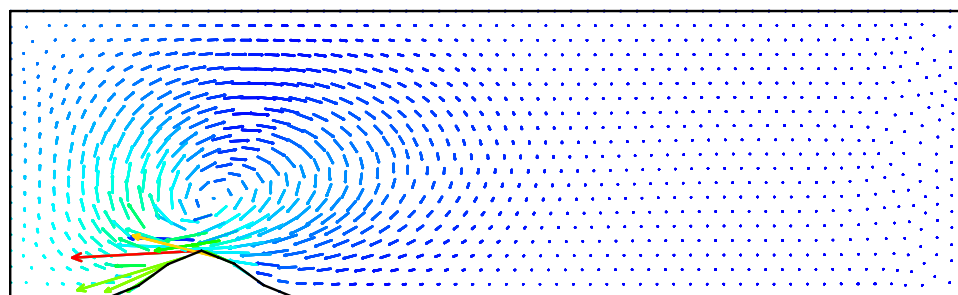
b) Re = 10
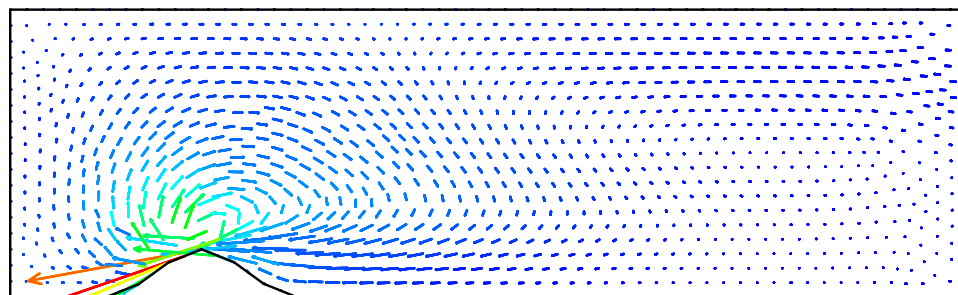
c) Re = 100

Figure 5.22: Sensitivity Vectors, $L = 10$, Flow over a Bump

a) u Contour Plot, Re = 500, L = 10
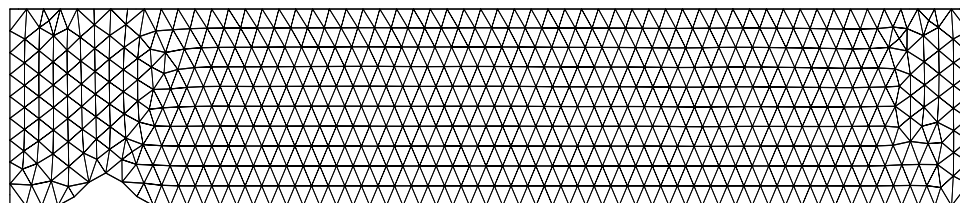


b) Sensitivity Vector Plot, Re = 500, L = 10
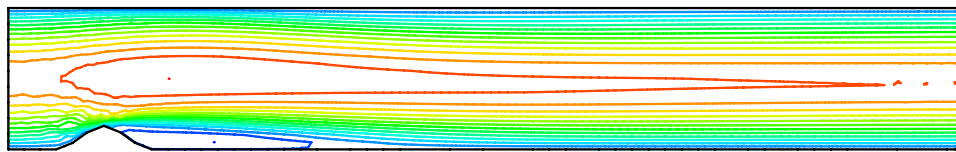


c) u Contour Plot, Re = 500, L = 20
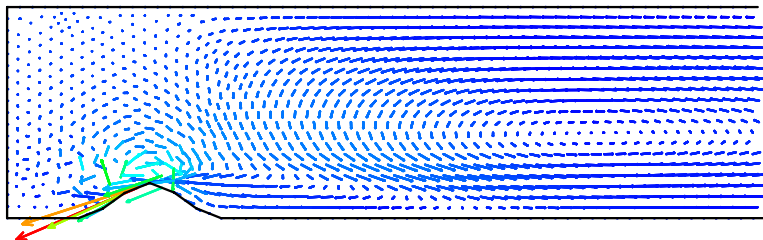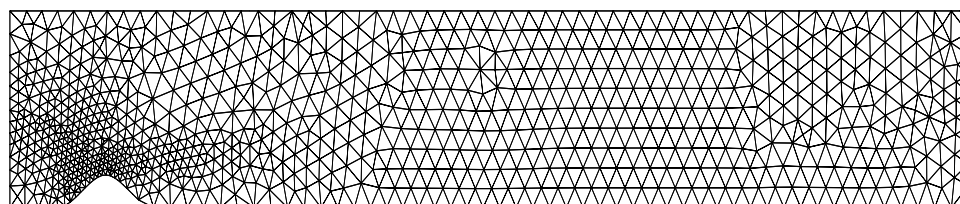


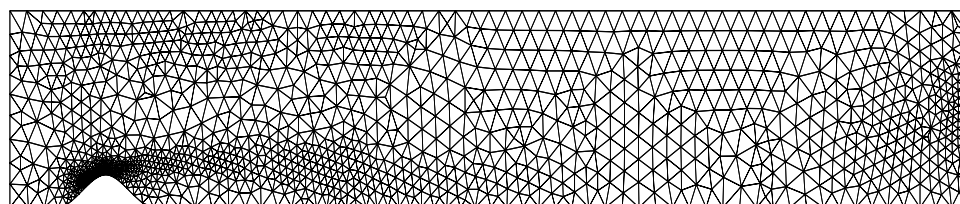d) Sensitivity Vector Plot, Re = 500, L = 20 (Part of Channel)

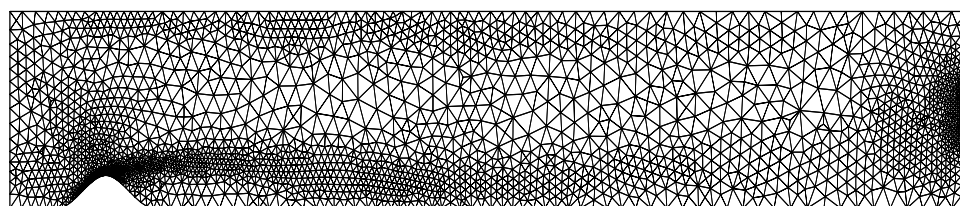Figure 5.23: u-Velocity Contours and Sensitivity Vectors, $Re = 500$, Flow over a Bump
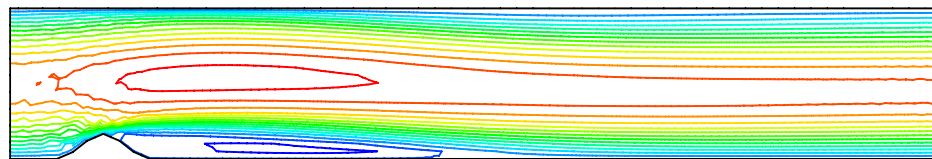
a) Initial Mesh
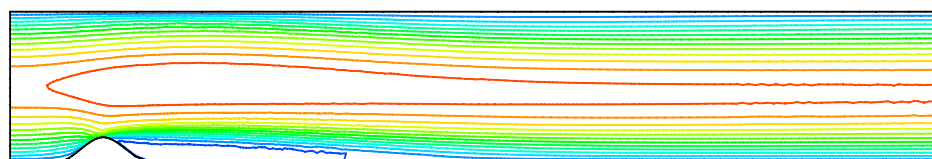


b) First Adapted Mesh
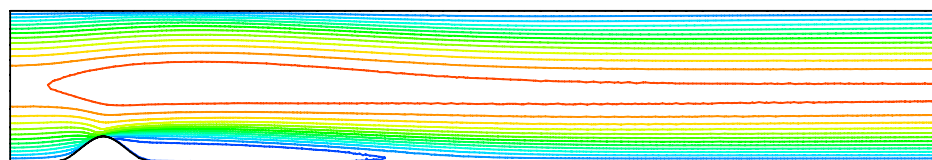


c) Second Adapted Mesh



d) Third Adapted Mesh

Figure 5.24: Flow over a Bump, Initial and Adapted Meshes for $Re = 1000, L = 20$

a) u Contour Plot, Re = 1000, Initial Mesh

b) u Contour Plot, Re = 1000, First Adapted Mesh

c) u Contour Plot, Re = 1000, Second Adapted Mesh

d) u Contour Plot, Re = 1000, Third Adapted Mesh

-0.045          0.039          0.12          0.21          0.29          0.38          0.46          0.55

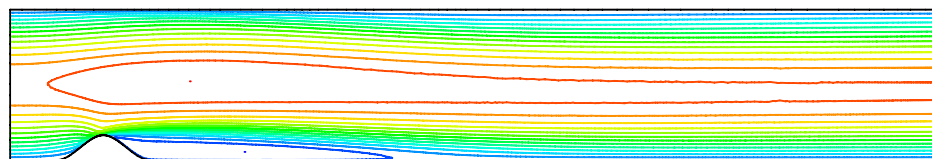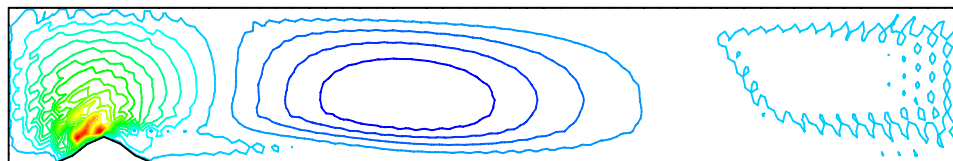Figure 5.25: u-Velocity Contours for Flow over a Bump

a) v Contour Plot, Re = 1000, Initial Mesh

b) v Contour Plot, Re = 1000, First Adapted Mesh

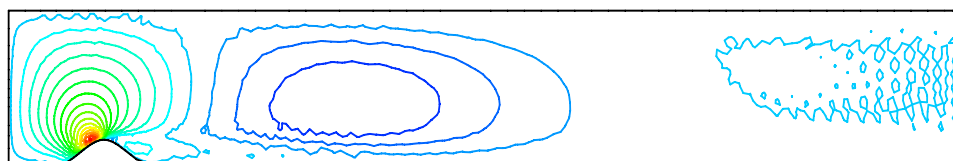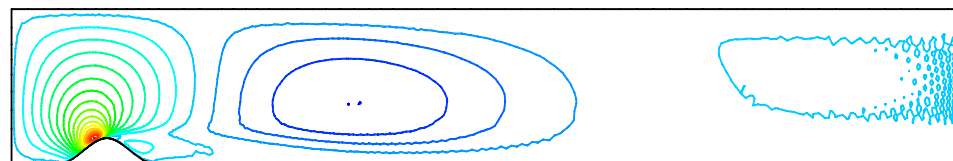c) v Contour Plot, Re = 1000, Second Adapted Mesh

d) v Contour Plot, Re = 1000, Third Adapted Mesh

-0.025    -0.012    0.0015    0.015    0.028    0.041    0.054    0.068    0.081    0.094    0.11
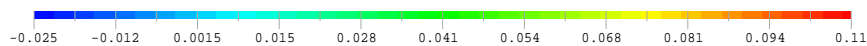
Figure 5.26: v-Velocity Contours for Flow over a Bump

a) u Sensitivity Contour Plot, Re = 1000, Initial Mesh

b) u Sensitivity Contour Plot, Re = 1000, First Adapted Mesh

c) u Sensitivity Contour Plot, Re = 1000, Second Adapted Mesh

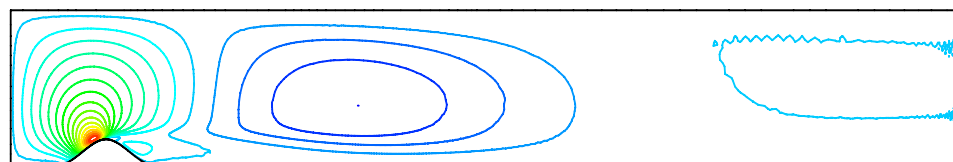d) u Sensitivity Contour Plot, Re = 1000, Third Adapted Mesh

Figure 5.27: u-Velocity Sensitivity Contours for Flow over a Bump

a) v Sensitivity Contour Plot, Re = 1000, Initial Mesh

b) v Sensitivity Contour Plot, Re = 1000, First Adapted Mesh

c) v Sensitivity Contour Plot, Re = 1000, Second Adapted Mesh

d) v Sensitivity Contour Plot, Re = 1000, Third Adapted Mesh

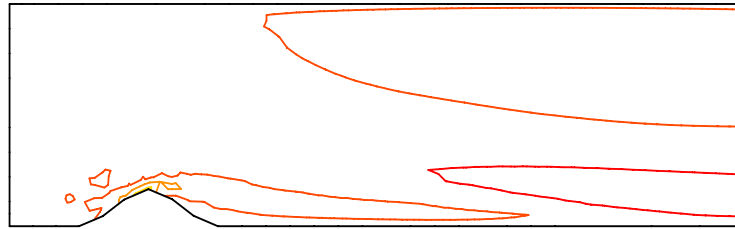-3.1    -2.7    -2.4    -2    -1.6    -1.3    -0.93    -0.57    -0.21    0.14    0.5

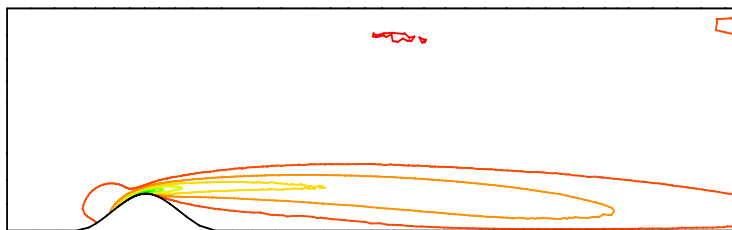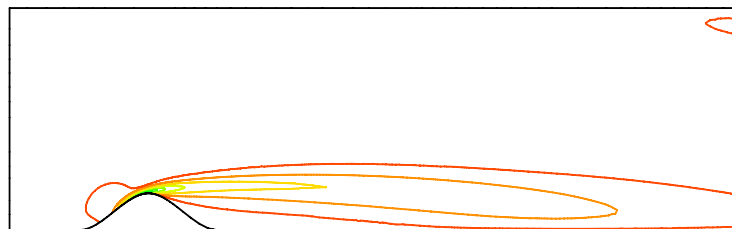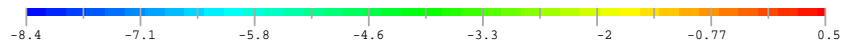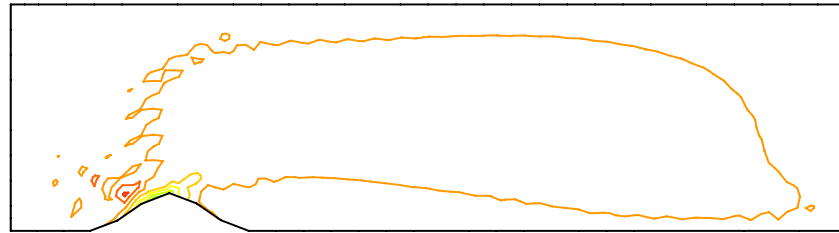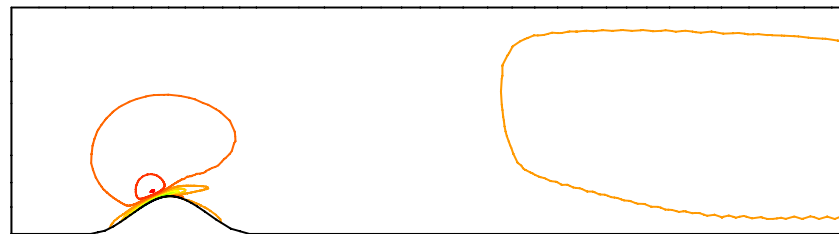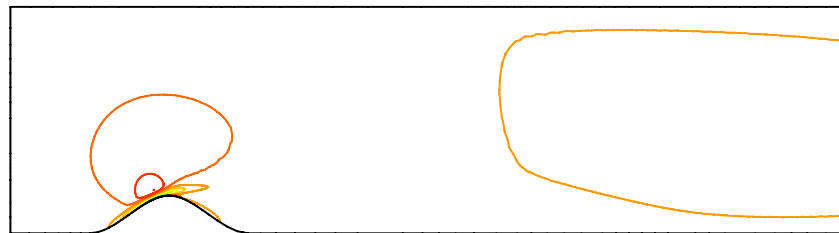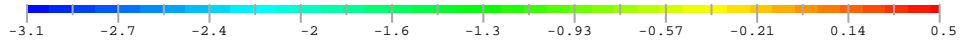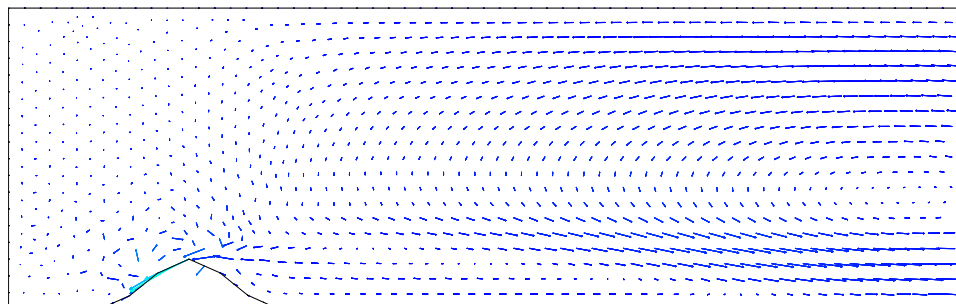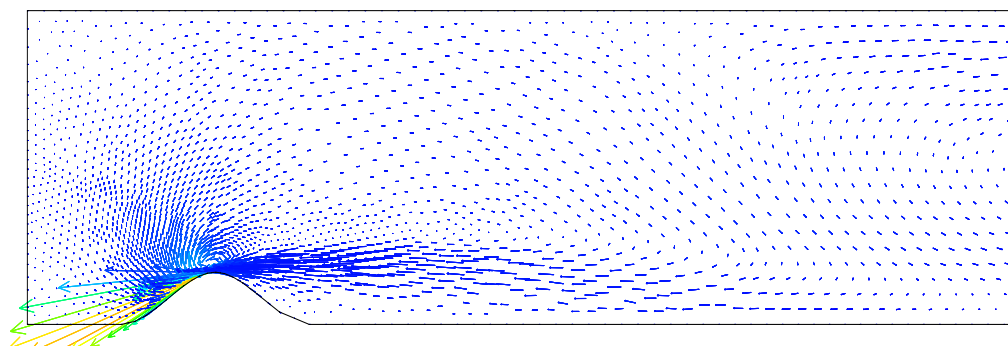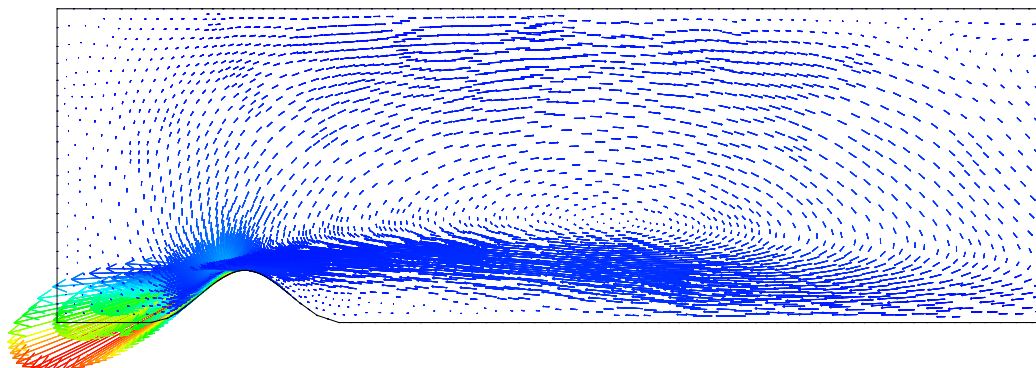Figure 5.28: v-Velocity Sensitivity Contours for Flow over a Bump

a) Sensitivity Vector Plots, Re = 1000, Initial Mesh

b) Sensitivity Vector Plots, Re = 1000, First Adapted Mesh

c) Sensitivity Vector Plots, Re = 1000, Second Adapted Mesh

Figure 5.29: Sensitivity Vector Plots on Initial and Adapted Meshes

**Analysis of Cost Functionals and Gradients - An Optimization Issue**

We now turn to the issue of using numerical approximations of the state and sensitivities to approximate cost functionals and their gradients. We do a comparison with a problem considered in Chapter 11 of [11]. In this chapter, Burkardt presents what he refers to as a discretized sensitivity failure. We will evaluate his results and show that the process of adaptive mesh refinement is key to obtaining good cost function and gradient evaluations in order to prevent inaccurate results from an optimization code.

We look at the numerical experiment carried out in §11.2 of [11]. We evaluate the following cost functional:

$$J^h(u(x,y), \mathbf{q}) = \frac{1}{2} \sum_{i=1}^{P} (u^h(3, y_i) - u^{Targ}(3, y_i))^2. \tag{5.1}$$

Here, $P$ is the number of matching points and for the results we present herein $P$ was fixed at 15. The matching points, $(3, y_i)$, $i = 1, 2, \cdots, P$, were evenly distributed along the line $x = 3, y \in [0, 3]$. Also, the target profile, $u^{Targ}(3, y)$, was obtained by computing a finite element solution with the shape parameter, $\mathbf{q}^{Targ} = (0.375, 0.5, 0.375)^T$, and $\lambda = 0.5$, and then interpolating to obtain the values for $u^T(3, y_i)$, $i = 1, 2, \cdots, P$.

The gradient of the cost function with respect to $\mathbf{q}$ is expressed as

$$\nabla_{\mathbf{q}} J^h = \sum_{i=1}^{P} (u^h(3, y_i) - u^{Targ}(3, y_i)) \frac{\partial u^h}{\partial \mathbf{q}}(3, y_i). \tag{5.2}$$

The SEM uses the discrete approximation of the continuous SE as before to approximate the value of the cost gradient. We will denote this approximations of the cost gradient by $\widetilde{\nabla}_{\mathbf{q}} J^h$, thus we have

$$\nabla_{\mathbf{q}} J^h \approx \widetilde{\nabla}_{\mathbf{q}} J^h = \sum_{i=1}^{P} (u^h(3, y_i) - u^{Targ}(3, y_i)) \mathbf{s}^h(3, y_i). \tag{5.3}$$

In order to nearly duplicate the numerical experiment carried out by Burkardt, we have $L = 10$, $Re = 1$ and we generated the matching profile from the initial mesh for $\mathbf{q}^{Targ}$. We calculated the values of the cost functional $J^h$ and its gradient along a line parameterized by $S$, connecting $\mathbf{q} = (-0.117, 0.419, -0.149)^T$ at $S = 0$ and $\mathbf{q} = (0.375, 0.5, 0.375)^T$ at $S = 25$. This is the same line along which Burkardt explored. His results are presented in Table (11.2) on page 153, and Figures (11.3) and (11.4), page 145 of [11].

Table 5.3: Values of $J^h$ along a Line, $Re = 1$, and $\lambda = 0.5$

| $S$ | $q_1$ | $q_2$ | $q_3$ | $J^h \cdot 10^3$ | $(\nabla_{\mathbf{q}} J^h \cdot \hat{S}) \cdot 10^3$ | $J^h \cdot 10^3$ | $(\nabla_{\mathbf{q}} J^h \cdot \hat{S}) \cdot 10^3$ |
|---|---|---|---|---|---|---|---|
| | | | | Initial Mesh | | 03 Mesh | |
| 0 | -0.117 | 0.419 | -0.149 | 16.9 | -0.99 | 12.1 | -0.56 |
| 1 | -0.097 | 0.422 | -0.128 | 16.3 | -1.03 | 11.7 | -0.57 |
| 2 | -0.077 | 0.425 | -0.107 | 15.5 | -1.00 | 11.3 | -0.58 |
| 3 | -0.057 | 0.428 | -0.086 | 14.6 | -0.94 | 10.8 | -0.60 |
| 4 | -0.038 | 0.432 | -0.065 | 13.7 | -0.91 | 10.3 | -0.61 |
| 5 | -0.018 | 0.435 | -0.044 | 12.9 | -0.92 | 9.95 | -0.62 |
| 6 | 0.001 | 0.438 | -0.023 | 12.2 | -1.04 | 9.49 | -0.63 |
| 7 | 0.020 | 0.441 | -0.002 | 11.4 | -1.00 | 9.02 | -0.65 |
| 8 | 0.040 | 0.444 | 0.018 | 10.6 | -0.96 | 8.56 | -0.67 |
| 9 | 0.060 | 0.448 | 0.039 | 9.72 | -0.99 | 8.00 | -0.68 |
| 10 | 0.079 | 0.451 | 0.060 | 9.02 | -1.15 | 7.48 | -0.70 |
| 11 | 0.099 | 0.454 | 0.081 | 8.35 | -1.07 | 6.95 | -0.72 |
| 12 | 0.119 | 0.457 | 0.102 | 7.46 | -1.08 | 6.39 | -0.74 |
| 13 | 0.138 | 0.461 | 0.123 | 6.21 | -0.77 | 5.76 | -0.75 |
| 14 | 0.158 | 0.464 | 0.144 | 5.62 | -0.78 | 5.17 | -0.76 |
| 15 | 0.178 | 0.467 | 0.165 | 5.00 | -0.79 | 4.56 | -0.77 |
| 16 | 0.197 | 0.470 | 0.186 | 4.38 | -0.78 | 3.94 | -0.76 |
| 17 | 0.217 | 0.474 | 0.207 | 3.70 | -0.76 | 3.27 | -0.75 |
| 18 | 0.237 | 0.477 | 0.228 | 3.08 | -0.74 | 2.66 | -0.72 |
| 19 | 0.256 | 0.480 | 0.249 | 2.47 | -0.70 | 2.07 | -0.68 |
| 20 | 0.276 | 0.483 | 0.270 | 1.89 | -0.64 | 1.51 | -0.62 |
| 21 | 0.296 | 0.487 | 0.291 | 1.33 | -0.56 | 0.99 | -0.53 |
| 22 | 0.316 | 0.490 | 0.312 | 0.72 | -0.50 | 0.57 | -0.42 |
| 23 | 0.335 | 0.493 | 0.333 | 0.32 | -0.35 | 0.25 | -0.29 |
| 24 | 0.355 | 0.496 | 0.354 | 0.11 | -0.21 | 0.05 | -0.13 |
| 25 | 0.375 | 0.500 | 0.375 | 0.00 | 0.00 | 0.01 | 0.05 |
| 26 | 0.394 | 0.503 | 0.396 | 0.06 | 0.16 | 0.16 | 0.28 |
| 27 | 0.414 | 0.506 | 0.417 | 0.31 | 0.38 | 0.53 | 0.53 |
| 28 | 0.434 | 0.509 | 0.438 | 0.81 | 0.63 | 1.16 | 0.81 |
| 29 | 0.453 | 0.513 | 0.459 | 1.63 | 0.92 | 2.08 | 1.13 |
| 30 | 0.473 | 0.516 | 0.480 | 2.72 | 1.22 | 3.30 | 1.48 |

Our results are displayed in Table (5.3) and Figures (5.30) - ( 5.31). Burkardt reported a local maximum at $S = 5$. As seen in Figure (5.30), we do not get the same results on the initial mesh. In order to determine if the difference was due to not having sufficient accuracy on the initial mesh, we also presented the results for meshes which were refined on the flow field. It is possible that the difference between our results and those of Burkardt could be due to our having fixed the value of $\lambda$ at 0.5. It is also possible that the difference is a result of numerical inaccuracies due to discretization differences.

We did one more numerical experiment exactly similar to the one described above with one exception, we now used $Re = 100$. Figure (5.32) - (5.33) show the cost function and gradient along the same line explored above. Note that adapting the mesh is even critical to obtaining accurate, smooth cost and gradient approximations at this Reynolds number.
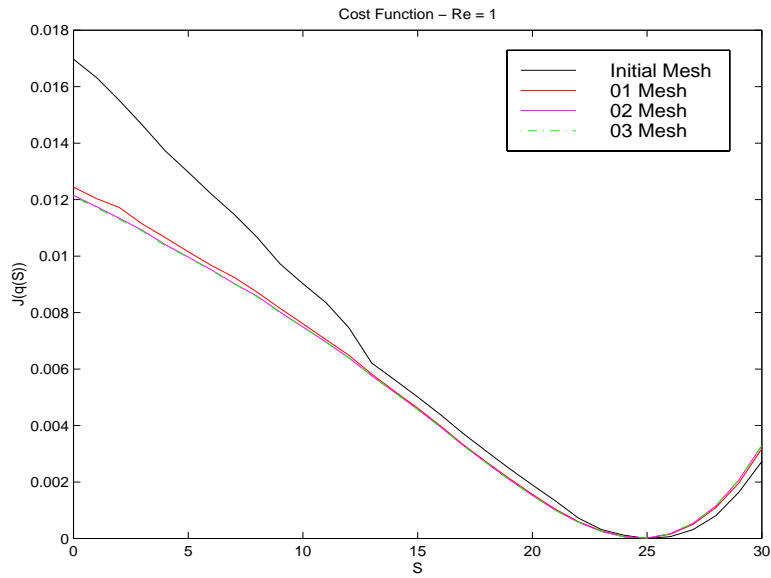
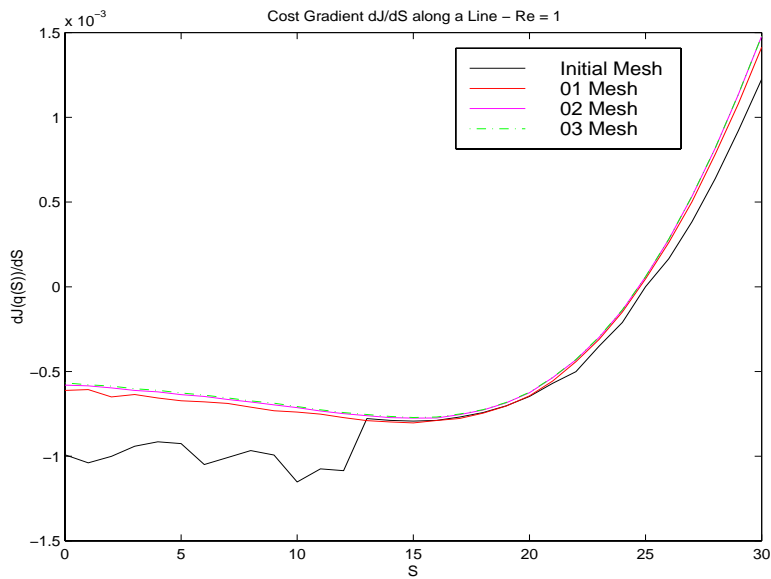Figure 5.30: Values of $J^h(\mathbf{q}(S))$ along a line.



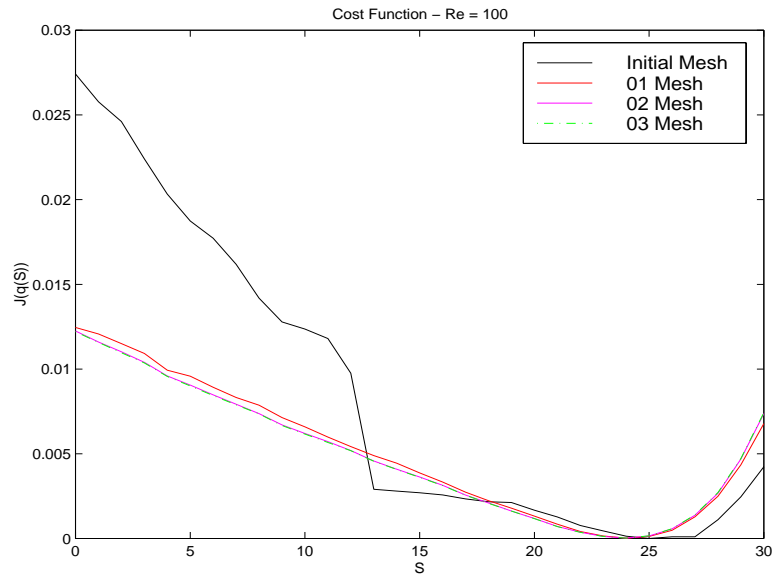Figure 5.31: Values of $\frac{dJ^h(\mathbf{q}(S))}{dS}$ along a line.

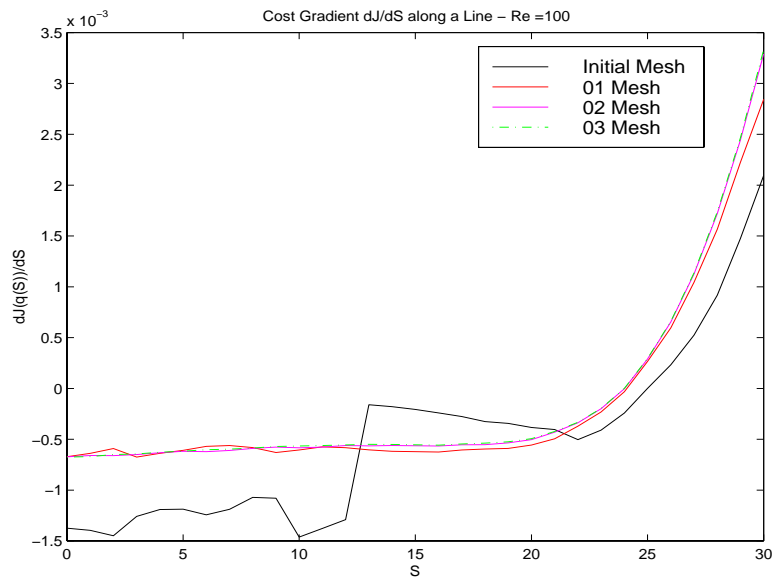Figure 5.32: Values of $J^h(\mathbf{q}(S))$ along a line.



Figure 5.33: Values of $\frac{dJ^h(\mathbf{q}(S))}{dS}$ along a line.

# Chapter 6

# Conclusions

The primary goal of this work was to develop and analyze new algorithms for accurate computation of design sensitivities. We focused on finite element schemes applied to the continuous sensitivity equation. The SEM is more efficient than standard finite difference schemes. However, accuracy has been an issue for flow problems. Our work concentrated on two methods for improving accuracy without compromising speed.

The first method used enhanced spatial derivative approximations in the sensitivity equation. Local and global projection techniques were employed to obtain higher accuracy in the derivative approximations. These projection techniques not only provided better sensitivity approximations at a fixed parameter value, but also stabilized the numerics over greater parameter ranges. For the 1-D model problem, the local projection did better than the global projection at most parameter values when the mesh was coarse. As the mesh refined, however, the use of global projection techniques provided more accurate sensitivities. At this point there are no theoretical results to explain this behavior. This issue is something that needs to be addressed in future work.

The second technique, mesh refinement based on errors in the flow, also dramatically improved the sensitivity approximations. This was especially true in the case of high Reynolds number calculations or in shape sensitivity problems, as in the case of flow over a bump. In addition, we showed that at higher Reynolds numbers adapting on errors in the sensitivities, as well as the flow, improved the accuracy of the sensitivity approximations. The refinement is important in the case of shape problems since accurate gradient approximations of the state are needed to specify the sensitivity boundary conditions.

We discovered that when these two techniques were used for approximating sensitivities for

use in the calculation of cost function gradients, the convergence properties of optimization algorithms was greatly improved.

There are many issues which were not addressed in this dissertation which will be the subject of future research. These include the development of schemes specifically targeted to improve the accuracy of local projections at the boundaries. Also, a study of diagonalization methods for use in optimization algorithms must be carried out to determine how the trade-off between the accuracy of cost function/gradient approximations and speed of convergence. Finally, improved error estimates are needed to prove convergence of these numerical techniques and to evaluate rates of convergence for the different state derivative approximations.

# Bibliography

[1] Ainsworth, M., Zhu, J.Z., Craig, A.W. and Zienkiewicz, O.C. (1989), "Analysis of the Zienkiewicz-Zhu A-Posteriori Error Estimator in the Finite Element Method," *International Journal for Numerical Methods in Engineering, Vol. 28*, 2161-2174.

[2] Appel, J. (1997), "Sensitivity Calculations for Conservation Laws with Application to Discontinuous Fluid Flows," Ph.D. Thesis, Virginia Tech, Blacksburg, VA.

[3] Borggaard, J.T. (1994), "The Sensitivity Equation Method for Optimal Design," Ph.D. Thesis, Virginia Tech, Blacksburg, VA.

[4] Borggaard, J., Burkardt, J., Cliff, E., Gunzburger, M., Kim, H., Lee, J., Peterson, J., Shenoy, A. and Wu, X., (1995), "Algorithms for Flow Control and Optimization," *Optimal Design and Control*, Borggaard, J., Burkardt, J., Gunzburger, M. and Peterson, J., Eds., Birkhauser, 97-116.

[5] Borggaard, J.T. and Burns, J.A., (1997), "A PDE Sensitivity Equation Method for Optimal Aerodynamic Design," *Journal of Computational Physics*.

[6] Borggaard, J.T. and Burns, J.A., (1994), "A Sensitivity Equation Approach to Optimal Design of Nozzles," *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Design, AIAA Paper 94-4274*, 232-241.

[7] Borggaard, J.T. and Burns, J.A., (1995), "A Sensitivity Equation Approach to Shape Optimization in Fluid Flows," *Flow Control, Proceedings of the IMA, Vol. 68*, M. Gunzburger, Ed., Springer-Verlag.

[8] Borggaard, J.T. and Pelletier, D., (1996), "Computing Design Sensitivities using an Adaptive Finite Element Method," *27th AIAA Fluid Dynamics Conference, AIAA Paper 96-1938*.

[9] Borggaard, J.T. and Pelletier, D., (1998), "Observations in Adaptive Refinement Strategies for Optimal Design," *Proceedings of the AFOSR Workshop on Optimal De-*

117

*sign and Control*, J. Borggaard, J. Burns, E. Cliff, amd S. Schreck, Eds., Birkhauser, 59-76.

[10] Braess, D., (1997), *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, Springer-Verlag.

[11] Burkardt, J., (1995), "Sensitivity Analyses and Computational Shape Optimization for Incompressible Flows," Ph.D. Thesis, Virginia Tech, Blacksburg, VA.

[12] Burns, J.A. and Spies, R., (1994), "A Numerical Study of Parameter Sensitivities in Landau-Ginzburg Models of Phase Transitions in Shape Memory Alloys," *Journal of Intelligent Material Systems and Structures, Vol. 5*, 321-332.

[13] Burns, J.A. and Spies, R., (1992), "Sensitivity Analysis for a Dynamic Model of Phase Transition in Materials with Memory," *Recent Advances in Adaptive and Sensory Materials and their Applications*, Rogers, C.A. and Rogers, R.C., Eds., Technomic Publishing Co., Basel, 82-93.

[14] Ciarlet, P., (1978), *The Finite Element Method for Elliptic Problems*, North-Holland.

[15] Cruz, J.B. Jr, Freudenberg, J.S., and Looze, D.P., (1981), "A Relationship between Sensitivity and Stability of Multivariable Feedback Systems," *IEEE Transactions on Automatic Control, Vol. 26, No. 1*, 66-74.

[16] Cruz, J.B. Jr and Perkins, W.R., "A New Approach to the Sensitivity Problem in Multivariable Feedback Design," *IEEE Transactions on Automatic Control, Vol. 9*, 216-223.

[17] Cuvelier, C., Segal, A., and van Steenhoven, A.A., (1986), *Finite Element Methods and Navier-Stokes Equations*, Reidel.

[18] Gallagher, R.H. and Zienkiewicz, O.C., Eds., (1973), *Optimum Structural Design, Theory and Applications*, John Wiley & Sons.

[19] Girault, V. and Raviart, P., (1987), *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*, Springer-Verlag.

[20] Godfrey, A., (to appear), "Using Sensitivities for Flow Analysis," *Computational Methods for Optimal Design and Control, Proceedings of the AFOSR Workshop on Optimal Design and Control*, J. Borggaard, J. Burns, E. Cliff, amd S. Schreck, Eds., Birkhauser, 181-196.

[21] Griewank, A. and Corliss, G., Eds., (1991), *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, SIAM.

[22] Gunzburger, M., (1989), *Finite Element Methods for Viscous Incompressible Flows: A guide to Theory, Practice, and Algorithms*, Academic Press, Inc.

[23] Haug, E., Choi, K., and Komkov, V., (1986), *Design Sensitivity Analysis of Structural Systems*, Academic Press.

[24] Hétu, J.-F. and Pelletier, D., (1992), "Adaptive Remeshing for Viscous Incompressible Flows," *AIAA Journal, Vol. 30, No. 8*, 1986-1992.

[25] Hétu, J.-F. and Pelletier, D, (1992), "Fast Adaptive Remeshing for Viscous Incompressible Flows," *AIAA Journal, Vol. 30, No. 11*, 2677-2682.

[26] Horowitz, I.M., (1963), *Synthesis of Feedback Systems*, Academic Press.

[27] Kreindler, E., (1968), "Closed-loop Sensitivity Reduction of Linear Optimal Control Systems," *IEEE Transactions on Automatic Control, Vol. 13*, 254-262.

[28] Kreindler, E., (1969), "On the Sensitivity of Closed-loop Nonlinear Optimal Control Systems," SIAM Journal on Control, Vol 7, 512-520.

[29] Ilinca F. and Pelletier, D., (1992), "A Unified Approach for Adaptive Solutions of Compressible and Incompressible Flows," *AIAA 35th Aerospace Sciences Meeting and Exhibit, AIAA Paper 97-0330*

[30] Iott, J., Haftka, R., and Adelman, H., (1985), "Selecting Step Sizes in Sensitivity Analysis by Finite Differences," *NASA Technical Memorandum 86282*.

[31] Kundu, P., (1990), *Fluid Mechanics*, Academic Press.

[32] Newman, P., (1994), "Preparation of Advanced CFD Codes for use in Sensitivity Analyses and Multidisiplinary Design Optimization," *Optimal Design and Control*, Borggaard, J., Burkardt, J., Gunzburger, M. and Peterson, J., Eds., Birkhauser, 241-274.

[33] Pelletier, D. and Ilinca, F., (1997), "Adaptive Remeshing for the $k - \varepsilon$ Model of Turbulence," *AIAA Journal, Vol. 35, No. 4*, 640-646.

[34] Pelletier, D., Ilinca, F., and Hétu, J.-F., (1994), "Adaptive Finite Element Method for Turbulent Flow Near a Propeller," *AIAA Journal, Vol. 32, No. 11*, 2186-2198.

[35] Pelletier, D. and Trépanier, J.Y. (1997), "Implementation of Error Analysis and Norms to Computational Fluid Dynamics Applications," Internal Report.

[36] Hou, G.W., Taylor , A.C. III, Mani, S.V., and Newman, P.A., (1993), "Formulation for Simultaneous Aerodynamic Analysis and Design Optimization," Presented at the Second U.S. National Congress on Computational Mechanics, Aug 16-18, Washington, D.C.

[37] Temam, R., (1984), *Navier-Stokes Equations*, North-Holland.

[38] Thompson, J., Warsi, Z.U.A., and Mastin, C.W. (1985), *Numerical Grid Generation: Foundations and Applications*, North-Holland.

[39] Wloka, J., (1987), *Partial Differential Equations*, Cambridge University Press.

[40] Zeidler, E. (1995), *Applied Functional Analysis: Main Principles and Their Applications*, Springer-Verlag.

[41] Zienkiewicz, O.C. and Zhu, J.Z., (1987), "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis," *International Journal for Numerical Methods in Engineering, Vol. 24*, 337-357.

[42] Zienkiewicz, O.C. and Zhu, J.Z., (1992), "A Super Convergent Patch Recovery and a posteriori Error Estimators. Part I: The Recovery Technique," *International Journal for Numerical Methods in Engineering, Vol. 33*, 1331-1364.

# Vita

Dawn Lynn DeVault Stewart was born in Columbus AFB, Mississippi on January 17, 1964. She graduated from Lakeview High School, Sandy Lake, Pennsylvania in June, 1982. With an Air Force Reserve Officer Training Corps (AFROTC) scholarship, she began to pursue a degree in Electrical Engineering at Grove City College, a small, Christian, liberal arts school in Grove City, Pennsylvania. She soon discovered that what she really enjoyed was the study of mathematics. She graduated with Honors in Mathematics and a commission as a Second Lieutenant in the United States Air Force in May, 1986. The Air Force, in need of weather personnel, then sent her to get a degree in Meteorology at Pennsylvania State University in State College, Pennsylvania. After receiving her Bachelor of Science degree in June, 1987, she was assigned to Los Angeles Air Force Station, Los Angeles AFB, California as a staff meteorologist. There she acted as a weather consultant to a number of space programs including the Titan IV Expendable Launch Vehicle program. In May, 1990, she was reassigned as weather satellite coordinator to the island base of Lajes, Azores. While at Lajes, Dawn began the process of convincing the Air Force to allow her to pursue her love of mathematics and her interest in teaching. She applied for a position as instructor at the United States Air Force Academy, was accepted, and in May, 1992 was transferred back to Pennsylvania State University where she obtained a Master of Arts in Mathematics. Upon completion of her degree in December, 1993, Dawn was assigned to the Department of Mathematics at the academy. There she enjoyed teaching a variety of math courses and again requested Air Force sponsorship to further her education. She was accepted into the PhD program and began her studies at Virginia Polytechnic Instutute and State University in June, 1995. Along the way, Dawn was promoted to a Major in the United States Air Force. In addition, she has two children, Katie Leigh, age 11, and David Alexander, age 6. In June, 1998, Dawn will return to the Air Force Academy as an Assistant Professor of Mathematics.