A Posteriori Error Estimation For Finite Volume Simulations of Fluid Flows

Brian Van Straalen

July 8, 1996

Abstract

A technique for numerically estimating the discretization error in upwind based finite volume fluid flow simulation was developed. The technique is based on residual estimation, followed by solving the global error equation over the computational domain. One and two dimensional analysis of the error estimation process was performed for a simple homogeneous advection-diffusion equation. The technique was then extended to encompass the laminar Navier-Stokes equations. The effectiveness of the technique was investigated for flow over a two dimensional backward-facing step. The results show promise for future implementations of effective and practical error estimation.

This report was submitted as a thesis for the Master's degree, University of Waterloo. A large debt of gratitude goes to my two supervisors: Prof. Bruce Simpson and Prof. Gord Stubley, for their lessons in perseverance. This research was jointly funded by Advanced Scientific Computing Ltd. and the Natural Science and Engineering Research Council. Special thanks to Dr. M.J. Raw of Advanced Scientific Computing for his insightful discussions.

Contents

List of Figures

Nomenclature

a	cell Peclet number
$\mathcal{E}$	exact error
E	approximate error
f	scalar source term
h	mesh spacing
i	discrete mesh index
i	control volume index
$\mathcal{L}$	exact linear operator
L	approximate linear operator
M	ratio of error estimate to exact error
$\mathcal{N}$	exact nonlinear operator
N	approximate nonlinear operator (Chapter 4)
N	number of control volumes in mesh
$\vec{n}$	outward normal unit vector
P	ratio of estimated to exact operator residual (Chapter 2)
P	fluid pressure of exact solution
p	fluid pressure of approximate solution
R	exact operator residual
Re	Reynolds number
r	approximate operator residual
S	surface of arbitrary volume
S	mapping function from discrete to continuous
t	time
s, t	local element coordinates
U	X-component of exact fluid velocity
u	X-component of approximate fluid velocity
V	Y-component of exact fluid velocity
v	Y-component of approximate fluid velocity
$\vec{V}$	fluid velocity vector
$\mathcal{V}$	mesh over problem domain
x, y	Cartesian coordinates

α	weighting function for element residual methods
Φ	Exact scalar solution
ϕ	approximate scalar solution
Γ	scalar diffusion coefficient
Γ_D, Γ_N	finite element domain boundary
μ	fluid dynamic viscosity
Ω	problem domain
ρ	fluid density
ψ	arbitrary scalar field

Superscripts

$\star$	outward flux formulation of operator
$full$	fully linearized operator
h	associated with a discrete mesh
$semi$	semi-linearized operator

Subscripts

$\mathcal{E}$	error equation operator
FVM	associated with a finite volume method discretization
j	jump in value across element boundary
k	associated with a particular control volume or element
l	associated with the left boundary
D	Dirichlet boundary condition
N	Neumann boundary condition
p	error in pressure field
r	associated with the right boundary
$spec$	specified as part of the problem definition
u	error in X-component of velocity
v	error in Y-component of velocity
x, y	in the x and y coordinate directions respectively

Chapter 1

Background

Analysts typically derive numerical solutions of partial differential equations in terms of a solution vector corresponding to some finite discretization of the problem domain.

From an engineering standpoint, the first question to ask oneself is "*Is this solution accurate enough ?*". Typical attempts to answer this question in the Finite Volume Method community have been of the form:

- Solve same problem on successively finer grids until the solution values stabilize to within an acceptable tolerance.
- Construct grids which utilize *a priori* information about the numerical scheme being employed (e.g., Richardson's Extrapolation)
- Apply other heuristics based on a physical interpretation of the data (domain conservation laws, etc.)

Each of these methods has problems associated with their successful implementation; excessive computing requirements, excessive user interaction (man hours), difficulty distinguishing modeling error from numerical error, etc.

For the purpose of this thesis, a more desirable type of error analysis would be some form of auxiliary calculation procedure which creates a discrete estimate of the discretization error, E^h, corresponding to the current approximate solution ϕ^h.

1.1 Introduction

This thesis is an attempt to take simple linear operator notions and extend them to construct an auxiliary linear equation set to estimate the error within an approximate solution. This document does not start with a current literature review, as is typical, since the concepts presented here do not require previous knowledge of error estimation to be clearly understood; they are derived from first principles.

The derivation of the error estimator is given in chapter 1.2. After this derivation, a commentary on current related research is given to place this work in context.

A one-dimensional analysis of the error estimation procedure for linear advection-diffusion equations is given in chapter 2. Two-dimensional numerical experimentation is presented in

chapter 3 for the linear advection-diffusion equation. Chapter 4 outlines the extension to two-dimensional Navier-Stokes equations and analyzes the error estimation procedure for two flows over a backward-facing step.

1.2 Error Estimation Principles

In this chapter, the technique of error estimation is introduced and formalized. The derivation draws on the advection-diffusion equation as a reference partial differential equation. After the theory and computational methodology are explicated, a comparison of our technique to contemporary error estimation techniques from other researchers is given.

1.3 The Error Equation

Consider the steady advection-diffusion equation, given in its pointwise form as

$$\vec{V}\nabla\Phi - \nabla(\Gamma\nabla\Phi) = f, \tag{1.1}$$

where Φ represents the concentration of the transported scalar, $\vec{V}$ is the advection field vector, Γ represents the diffusivity and f is a source of Φ per unit mass.

Equation 1.1 can be written in a simplified operator notation as

$$\mathcal{L}(\Phi) = f \tag{1.2}$$

where $\mathcal{L}$ is the partial differential operator.

Let ϕ_{approx} be an approximate solution of (1.2), and $\mathcal{E} \equiv \Phi - \phi_{approx}$ be the corresponding error. From elementary linear operator theory, assuming ϕ_{approx} is in the operator domain of $\mathcal{L}$, we can define the *Exact Operator Residual* as

$$R(\phi_{approx}) \equiv \mathcal{L}(\phi_{approx}) - f \tag{1.3}$$

and note that $\mathcal{E}$ is the solution of the *Error Equation*,

$$\mathcal{L}(\mathcal{E}) = -R(\phi_{approx}). \tag{1.4}$$

The error equation (1.4) utilizes the exact operator residual as a source term, in the same way that the original equation, (1.2), utilizes f as a source term. Evidently, one can think of the residual as an error 'source' term, and the error is transported and modified in the solution domain by the advection and diffusion processes. This description has been given by Ferziger [12].

It is perhaps useful at this point to distinguish between the exact operator residual as defined in (1.3) and the *Approximate Operator Residual*, which also appears in the numerical analysis literature, but which plays no role in this thesis. If L is an discrete approximation of the exact operator, $\mathcal{L}$, then the approximate operator residual, r_L, is usually defined as the residual of the exact solution Φ with respect to L, i.e.:

$$r_L \equiv L(\Phi) - f \tag{1.5}$$

The residual shown in (1.5) is commonly studied in numerical analysis, where it leads to a discussion of *order*, since r_L is typically expressed as a Taylor-series expansion (about the fixed point of the exact solution). From this point onward, the term *residual* will refer to the exact operator residual.

To recapitulate, there are two components identified for error estimation: the Exact Operator Residual, and the Global Error Equation. The next section discusses how to move from an operator formulation, through the finite volume method, to something which one can actually compute.

1.4 Error Estimation

The application of these ideas to finite volume methods (FVMs) is facilitated by the introduction of an integral form of (1.1) on the domain Ω. In this section we review the principles of the FVM that relate to extending the basic ideas of §1.1. See Hirsch [19] for a more comprehensive discussion of the FVM for fluid flows. In Appendix D we provide a detailed description of the FVM for the incompressible Navier-Stokes equations.

Let $\mathcal{V}$ be a mesh of the domain Ω, dividing it into finite volumes, or cells v_k for $k = 1$ to N. Discrete functions based on $\mathcal{V}$ are vectors of length N; we will write them with superscript 'h' such as ϕ^h, with ϕ^h_k being the value associated with the finite volume v_k in $\mathcal{V}$.

If we apply the divergence theorem to each cell, v_k, then equation (1.2) implies

$$\int_{\partial v_k} \mathcal{L}^\star(\Phi)\, ds = \int_{v_k} f\, dv, \tag{1.6}$$

where

$$\mathcal{L}^\star(\psi) \equiv (-\Gamma \nabla \psi + \psi V) \cdot \vec{n}, \tag{1.7}$$

and ds is the cell differential surface area.

Derivations of FVMs are typically based on replacing the flux expression, $\mathcal{L}^\star(\psi(x))$, of a continuous distribution $\psi(x)$ by an approximate expression based on the values of a discrete distribution, ψ^h. We will designate a typical FVM expression for the flux through a face of cell v_k by $L^\star(\psi^h)$. Then, by analogy with (1.6), we can express the cell conservation equations of the FVM as

$$\int_{\partial v_k} L^\star(\phi^h_{FVM}) ds = \int_{v_k} f dv. \tag{1.8}$$

Intuitively, we would like to apply the ideas of §1.2 to a ϕ_{approx} that corresponds to ϕ^h_{FVM}. However, for general FVMs, there is no method specific way to extend ϕ^h_{FVM} of (1.8) to a continuous distribution on Ω. For the standard forms of Ω and finite volume cells, we can use piecewise linear interpolation to extend a discrete distribution, ψ^h, on $\mathcal{V}$ to a continuous one on Ω, which we will designate as $S\psi^h(x)$. This extension has a special relationship with centred differencing FVMs since the discrete flux expressions for centred differencing, $L^\star_{CD}$, satisfy

$$L^\star_{CD}(\psi^h) = \mathcal{L}^\star(S\psi^{(h)}(x)). \tag{1.9}$$

The extension of ψ^h to $S\psi^h(x)$ does not allow us to use the formalism of §1.2 directly, since piecewise linear functions are not smooth enough to belong to the operator domain of

$\mathcal{L}$. Therefore, we turn to identifying a discrete estimate of the residual in (1.3) based on an integral form of the error equation, (4). If ϕ_{approx} is a smooth approximate solution and $\mathcal{E} \equiv \Phi - \phi_{approx}$ then

$$\begin{aligned} \int_{\partial v_k} \mathcal{L}^\star(\mathcal{E}) ds &= \int_{v_k} f dv - \int_{\partial v_k} \mathcal{L}^\star(\phi_{approx}) ds \\ &= -\int_{v_k} R(\phi_{approx}) dv. \end{aligned} \tag{1.10}$$

We now identify the average of $R(\phi_{approx})$ on v_k as our estimate of the exact operator residual for ϕ_{approx} at cell v_k, i.e.,

$$R_k^h(\phi_{approx}) \equiv \frac{1}{|v_k|} \int_{v_k} R(\phi_{approx}) dv \tag{1.11}$$

where $|v_k|$ is the volume of cell v_k.

Equations (1.10) and (1.11) can now be used to *define* $R_k^h(\phi_{approx})$ for any continuous distribution, ϕ_{approx}, in the operator domain of $\mathcal{L}^\star$, and this wider definition admits the piecewise linear extensions of discrete distributions, $S\psi^h(x)$.

In particular, (1.10) and (1.11) provide a definition for a discrete residual estimate for ϕ_{FVM}^h, i.e.,

$$|v_k| R_k^h(S\phi_{FVM}^h) \equiv \int_{v_k} f dv - \int_{\partial v_k} \mathcal{L}^\star(S\phi_{FVM}^h) ds. \tag{1.12}$$

Computationally, this residual estimate is simply the centred difference FVM residual of ϕ_{FVM}^h, as we can recognize using (1.9).

This discrete residual estimate provides the 'error source' term for our estimate of the global error in ϕ_{FVM}^h, which we designate by E^h. E^h is computed as the solution of the original upwind FVM method with source terms that are the discrete residual estimates, i.e.

$$\int_{\partial v_k} L^\star(E^h) ds = -|v_k| R_k^h(S\phi_{FVM}^h) \quad , \quad k = 1 \; to \; N. \tag{1.13}$$

1.5 Related Research

We will refer to the technique described in the preceding sections as the *Global Error Equation Method*. It represents a novel approach to *a posteriori* error estimation compared to other contemporary methods.

At the current time, there are three main approaches used to estimate errors in approximate solutions: *Element Residual Methods*, *Flux Projection Methods*, and, *Extrapolation Methods*. The points of contrast between these methods and our approach are outlined in this section to provide a context for evaluation and illuminate the research motivations.

A more detailed description of these three methods is provided in appendix A. It is in this appendix that comprehensive references to the literature are provided.

The Element Residual Methods most closely resemble the approach described in this thesis. They are reported in a finite element method context universally. Research on this approach by Ainsworth and Oden [1], Babuška and Rheinboldt [4], Bank [7], Bank and

Weiser [6], Kelly *et al.* [20, 21, 22, 23], Oden, Weihan and Ainsworth [26], Strouboulis and Haque [36], Strouboulis and Oden [37] and Verfurth [40] is reviewed in Appendix A. They involve estimation of an exact operator residual, but instead of assembling a global system of equations to solve, the error between adjacent elements is assumed to decouple, thus the error estimate is obtained using an element-by-element algorithm. The different published approaches to the Element Residual Methods differ primarily in how they pose the boundary conditions for the element based error equation.

The Element Residual Methods may be interpreted as a form of *local* transformation of the exact operator residual. As such, the resulting error estimate, E^h, bear little resemblance to the global error, $\mathcal{E}$. Unlike this thesis, the distributed error estimate, E^h, is never reported in the literature. Instead, they invariably quote the *energy of the error*, also called the *energy norm.*

Since the Element Residual Methods represents a local transformation of the residual, it should come as no surprise that they can be used successfully to drive grid refinement algorithms. Recall that the residual acts as a 'source' term in the error equation. If one targets the regions of large magnitude residuals for grid refinement, then one would be most efficiently reducing the *sources* of errors in the approximate solution.

The Flux Projection Methods stem from a simple observation: The exact solution to the PDE problem probably does not have discontinuous derivatives, whereas the approximate solution does. If one performs what is termed a *projection* calculation, then one can create an approximate solution which *does* have continuous derivatives. Assuming that the smooth solution should be more accurate than the non-smooth solution, the difference between them should be an error estimate. Research by Ainsworth *et al.* [3], Babuška and Rodriguez [5], Hétu and Pelletier [16, 17, 18], Zienkiewicz and Zhu [43, 44] is reviewed in Appendix A.

Like the Element Residual Methods, the Flux Projection Methods are invariably reported using the energy norm. They are also most successfully used to drive grid refinement algorithms.

For both the Element Residual Methods and Flux Projection Methods reported in literature authors have assumed, explicitly or implicitly, that a local calculation is the only computationally feasible approach to pursue. The rationale is that a local computation is inexpensive and adequate for the purpose of grid refinement. In this thesis, this assumption is not made, for two main reasons. First, it is clear that the global nature of the error equation is essential to the process of estimating the global distributed error. The local error estimation techniques have not demonstrated any success in achieving our stated goal of determining approximate solution quality, which we feel is a different goal from error estimation for the purpose of grid refinement.

Second, computationally feasible implies that the computational cost of the error estimate should be *small* compared to the cost of generating the approximate solution. Now, for the majority of the literature, linear elliptic PDEs have been studied. For these problems, it is true that global error estimation costs just as much as solving the original problem, which is not acceptable. But, when solving coupled non-linear PDEs (like Navier-Stokes equations) the situation changes. Approximate solutions are typically generated in a process of iterative refinement, involving solving on the order of 5-20 coupled linear system of equations. In this context, a single linear equation set becomes a feasible computation. In addition, since the error equation is of the same class as the original PDE, the optimized solver techniques,

(preconditioners, multigrid methods, etc.), created for solving the approximate solution are at your disposal for solving the error equation.

Extrapolation Methods involve exploiting *a priori* knowledge of the order of the numerical method being employed to extrapolate from a set of approximate solutions to an estimate of the exact solution. With this exact solution, error estimation can be performed. Research on this method from Bradley, Missaghi and Chin [8], Caruso, Ferziger and Oliger [10], Ferziger [12, 13], Kessler, Peric and Scheurer [24], Roache [32] is given in Appendix A.

A commonly used form of extrapolation is Richardson's Extrapolation, which would estimate the error with a formula such as

$$E_2 = \frac{\phi_2 - \phi_1}{\left(\frac{h_1}{h_2}\right)^p - 1} + O(h^{p+1}), \tag{1.14}$$

where ϕ_1 and ϕ_2 are p-order solutions on grids whose grid spacing are characterized by h_1 and h_2 respectively. E_2 would the error estimate for the approximate solution ϕ_2. Like the Global Error Equation Method, this method does produce a distributed error estimate. It is also simple to understand and theoretically should apply to any PDE, and to any functional of the solution.

However, the difficulties with Extrapolation Methods are many. They assume that one can characterize an entire discretization method with a single parameter p, the order. In practice, only very simplistic discretizations have a readily identifiable convergence order. Implementation details such as boundary condition treatment, scheme blending, solution discontinuity treatment, etc. make it difficult to determine a unique single valued order for a given discretization.

The Extrapolation error estimate will only be valid if the $O(h^{p+1})$ term is significantly smaller than the leading term in equation (1.14). This is referred to as being in the *asymptotic range*. With two successive solutions on two different grids, there is no way of knowing if the asymptotic range has been reached. If one assumes that one knows p, and assumes that a single parameter grid set has been used, then it is possible to check if one is in the asymptotic range if one uses *three* solutions on *three* different grids.

Extrapolation also assumes that multiple computational grids can be defined with a single parameter variation, h. Unfortunately, only very simplistic grids lend themselves to such single parameter sets. Great effort must be expended attempting to create a single parameter grid set for a complex geometry. There is also no way to know if one has made the attempt successfully. On a more pragmatic note, the time and computational cost that must be used to generate a complex geometry grid set, and solve multiple approximate solutions, will usually be larger than the cost of the original approximate solution, i.e., it becomes computationally infeasible.

Suffice to say, the Global Error Equation Method does not suffer from these drawbacks.

1.5.1 Current Contributions

The Global Error Equation Method as described in §1.1 and §1.2 more closely resemble the Element Residual Method family than the other two approaches to error estimation outlined above. Both emphasize estimating the exact operator residual, however, since

the application of the element residual method has been primarily targeted towards grid adaption, the emphasis has been on computationally inexpensive local transformations of the residual to global error units, rather than accurate estimates of the distribution of the global error.

1.6 Summary

The Global Error Equation Method has been derived from standard linear operator theory. Beginning with a partial differential equation in operator notation (equation 1.2),

$$\mathcal{L}(\Phi) = f,$$

an approximate solution ϕ^h_{FVM} is generated using the Finite Volume Method (equation 1.8):

$$\int_{\partial v_k} L^\star(\phi^h_{FVM})ds = \int_{v_k} f dv.$$

The error estimation procedure begins with the calculation of the residual estimate (equation 1.12),

$$|v_k| R^h_k(S\phi^h_{FVM}) \equiv \int_{v_k} f dv - \int_{\partial v_k} \mathcal{L}^\star(S\phi^h_{FVM})ds.$$

Then the global error equation is solved (equation 1.13),

$$\int_{\partial v_k} L^\star(E^h)ds = -|v_k| R^h_k(S\phi^h_{FVM}) \quad , \quad k = 1 \; to \; N,$$

to yield an error estimate E^h.

The technique was contrasted, in principle, to contemporary *a posteriori* error estimation methods, in terms of the perceived strengths of this approach. In particular: The use of the full global error equation instead of a local calculation, error given in distributed form, instead of an energy norm, no need for the questionable assumptions imposed by extrapolation methods.

In the next chapter, a one dimensional analytical example of this procedure is given, with emphasis placed on evaluating the validity of the Global Error Equation Method. Chapter 3 follows these steps again, for a battery of two dimensional numerical problems instead of a finite analytical analysis.

Chapter 2

One Dimensional Advection-Diffusion Equation

To assess the validity of the Global Error Equation Method Consider the two steps of the process in a model situation that is analytically tractable. Consider the one-dimensional version of (1.1):

$$\begin{aligned} -\Gamma\frac{\partial^2\Phi}{\partial x^2} + V\frac{\partial\Phi}{\partial x} = 0 \quad V,\Gamma > 0, \\ \Phi = \Phi_l \quad at\ x = 0, \\ \Phi = \Phi_r \quad at\ x = 1. \end{aligned} \tag{2.1}$$

We can carry out an analytical analysis of the various components of the preceding section for discretizations of this problem using a uniform mesh of $0 \leq x \leq 1$ of width $h = 1/N$ centered at $x_k = kh$ [1].

The discrete form of the solution of (2.1) using a simple upwind FVM is [2]

$$\begin{aligned} \phi_k^h = \Phi_l + (\Phi_r - \Phi_l)\frac{(1+a)^k - 1}{(1+a)^N - 1}, \\ 0 \leq k \leq N, \end{aligned} \tag{2.2}$$

where a is the grid Peclet number, defined as $a \equiv Vh/\Gamma$.

2.1 Residual Estimation

In order to appraise the residual estimation process, it would be beneficial to generate a residual from the original partial differential operator and compare this function to the residual estimate.

[1] These computations were done with the MAPLE symbolic computation package

[2] See appendix B for a derivation of a finite difference analytical form

A natural extension of (2.2) to a C^∞ approximate solution of (2.1), ϕ_{approx}, can be obtained by replacing k by x/h, and allowing x to vary continuously from 0 to 1. The residual of ϕ_{approx} can now be computed as

$$R_k = \frac{(\Phi_r - \Phi_l)(1+a)^k}{(1+a)^N - 1} \left[\frac{Vln(1+a)}{h}(1 - ln(1+a)/a)\right]. \tag{2.3}$$

Calculation of a residual estimate for this solution using central differencing yields R^h,

$$R_k^h = \frac{(\Phi_r - \Phi_l)(1+a)^k}{(1+a)^N - 1}\frac{a^2 V}{2(1+a)h}. \tag{2.4}$$

An inspection of equations (2.3) and (2.4) reveals that they have the same dependence on k and differ by a constant factor that depends only on the grid Peclet number. The ratio of R^h to R provides an indication of the quality of the residual estimate, and how it varies with the problem parameters:

$$P \equiv \frac{R^h}{R} = \frac{a^3}{2(1+a)\ln(1+a)(a - \ln(1+a))}. \tag{2.5}$$

In the limit of $a = 0$, P approaches unity. P can be approximated to 90% accuracy in the range $0 \leq a \leq 13$ by

$$P \approx 1 + \frac{a}{6}$$

2.2 Error Estimation

Using the residual estimate R^h from §2.1, we now calculate an error estimate using a simple upwind differencing form of the error equation (as represented in equation (1.13)). The error equation will have Dirichlet boundary conditions (as in the original PDE) $E_l = E_r = 0$. Again, for brevity, the full derivation is given in appendix B .

The error estimate, E^h, then has the analytical form

$$E_k^h = \frac{a^2(\Phi_r - \Phi_l)}{2(1-(1+a)^N)^2}[(1-(1+a)^N)k(1+a)^{k-1} + N(1+a)^{N+k-1} - N(1+a)^{N-1}]. \tag{2.6}$$

Utilizing the approximate FVM solution (2.2), and the exact solution of equation (2.1), the exact error in the approximate solution can then be calculated. The ratio of the error estimate to the exact error can then be calculated as

$$M = \frac{E^h}{\mathcal{E}} = \frac{a^2(1-e^{Na})}{2(1-(1+a)^N)}\left[\frac{(1-(1+a)^N)k(1+a)^k + N(1+a)^{N-1}((1+a)^k - 1)}{(1-(1+a)^N)(e^{ka}-1) + (e^{Na}-1)((1+a)^k - 1)}\right]. \tag{2.7}$$

M is commonly referred to in the literature as a *nodal effectivity index*. The difference here being that the distributed variable representation is compared, instead of the energy norm of the error (see Babuška and Rodriguez [5]).

Unlike the residual estimate effectivity, P, the error estimate effectivity varies spatially over the domain (the dependence on k), as well as the grid density, N, in addition to the Peclet number dependence.

As a first pass at understanding what the quantity M is telling us, we can take the limit of the function as $a \equiv \frac{Vh}{\Gamma} \to 0$.

$$\lim_{a \to 0} M = 1.$$

So, in the limit of grid refinement, or in a diffusion dominated case (large Γ, small V), the error estimate asymptotically approaches the exact error.

More generally, one wants to know how M behaves when a is not zero. To get an impression of this behavior we can study M at a unique location spatially. An obvious choice would be the point where

$$\frac{\partial \mathcal{E}}{\partial k} = 0,$$

that is, the point of maximum error in the approximate solution.

At this point, The function M does seem to collapse to strictly a function of cell Peclet number (except in the near zero region). Figure 2.1 shows the effectivity index M plotted as a function of a for $N = 10, 100, 1000$. Note that all curves converge to a single curve in the region beyond $0 < a < 0.5$.

It can be seen that over a large range of Peclet numbers, and a large variance in grid density, the effectivity index varies by only about 12% from unity.

2.3 Summary

The Global Error Equation Method was applied to a finite analytical solution of the 1D advection-diffusion equation.

The residual estimate was compared to an *exact* form of the residual and shown to be functionally identical, but vary by a multiplicative constant.

The analytical error estimate was compared to the exact error through the use of an effectivity index. The effectivity index was shown to approach unity (error estimate approaches exact error) in the limit of grid cell Peclet number going to zero. Hence the error estimate should always improve from grid refinement.

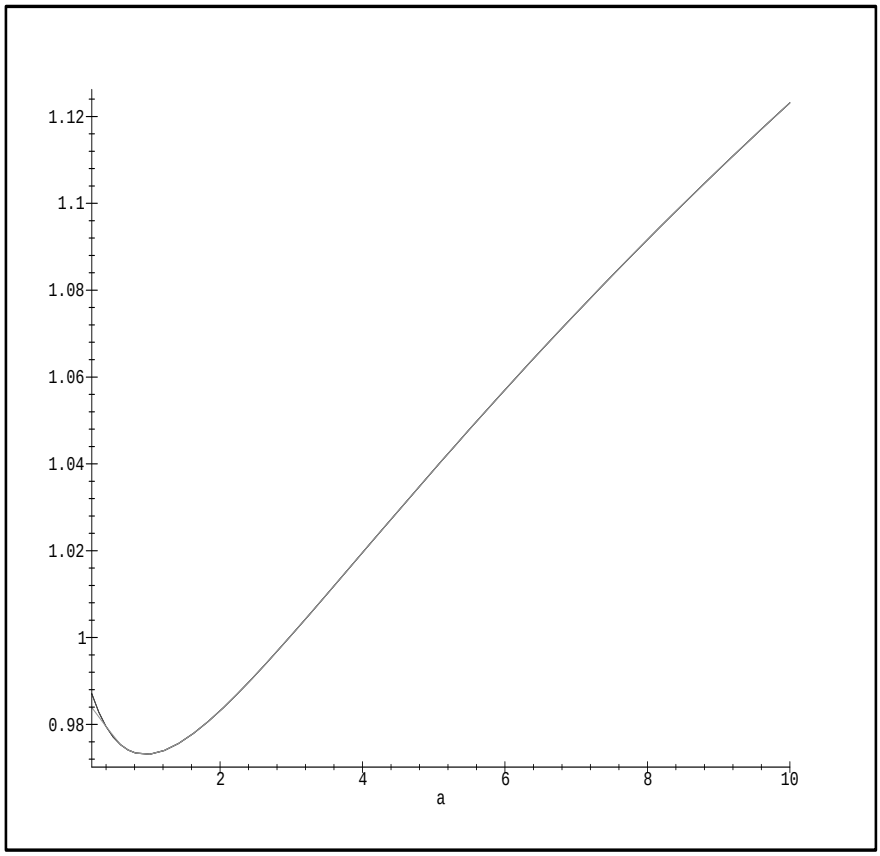


Figure 2.1: Maximum of Effectivity Index vs. Cell Peclet Number: N=10,100,1000

Chapter 3

Two Dimensional Advection-Diffusion Equation

To extend the evaluation of this new FVM global error estimator, a set of 2-dimensional test cases were created.

The extension into two space dimensions makes generation of closed form analytical finite difference solutions (as used in §2) unwieldy at best. For that reason, we limit our work to particular numerical problems, for which we know the analytic exact solution.

The 2D homogeneous scalar advection-diffusion equation is

$$\vec{V}\nabla\Phi - \nabla(\Gamma\nabla\Phi) = 0. \tag{3.1}$$

Solutions for this equation can be generated through use of a coordinate transform and using the applicable Green's function (This process is shown in detail in appendix C and is based on Stakgold [35]).

3.1 Test Suite

There are four different cases considered in this study:

- Point source subjected to grid-aligned flow.
- Point source subjected to grid skewed flow.
- A boundary layer field generated by a discontinuous Dirichlet boundary condition.
- A boundary layer field generated by a discontinuous Neumann boundary condition.

The first two are referred to as the *point source problems*, while the latter two are called the *boundary layer problems.*

For each of these cases, a solution is derived for the flow velocities of 0.15 and 0.45, and the diffusion coefficient set to 0.01 . The far-field conditions are assumed to be zero. The form of the exact solutions for these problems is given in its integral form. The integrals are evaluated numerically until the desired number of significant digits are achieved (7 digits for the purpose of this study).

For all problems, two different approximate solutions are generated. The first is a standard upwind differencing, referred to henceforth as the UDS approximate solution. It is a first-order numerical scheme and is the 2D analogue of the approximation explored in §2.

The second is a linear profile skew upwinding scheme with advection correction, similar to that described by Schneider and Raw [33], modified by an advection correction term described by Raithby [29], henceforth referred to as the LPS+PAC approximate solution. It is a second-order numerical scheme and represents a more realistic, commercially viable, discretization.

Thus, for each problem, the following steps are performed:

1. A grid is defined.
2. An exact analytical solution to the problem is calculated at each grid point (Φ^h).
3. Approximate numerical solutions are generated using UDS and LPS+PAC (ϕ^h).
4. The Global Error Equation Method is employed to produce an error estimate (E^h) for the approximate solution.
5. The error estimate is compared to the exact error ($\Phi^h - \phi^h$).
6. A subjective assessment of the quality of the error estimate is made.

3.2 Point Source Problems

Consider the problem where there is a unit point source at the origin. For this problem the exact solution can be expressed as

$$\Phi(x,y) = e^{\frac{Vx}{2\Gamma}} K_0(\frac{V}{2\Gamma}(x^2+y^2)^{\frac{1}{2}}), \tag{3.2}$$

where K_0 is a zero degree modified Bessel function of the second kind.

Since the exact solution is not defined at the origin for this problem, the computational domain is slightly offset from the origin. For the numerical solutions, all boundary conditions are implemented as Dirichlet conditions utilizing the exact solution

3.2.1 Grid Aligned Flow

For the grid aligned flow condition, two problems are considered: a low speed flow, V=0.15, and a high speed flow, V=0.45. The exact solutions for these problems is shown in figures 3.1 and 3.2 respectively.

For the first cases, the point source problems, a coarse grid is used. The grid is shown in figure 3.3. The results of the numerical experimentation with the UDS approximate solutions are shown in figures 3.4 through 3.7.

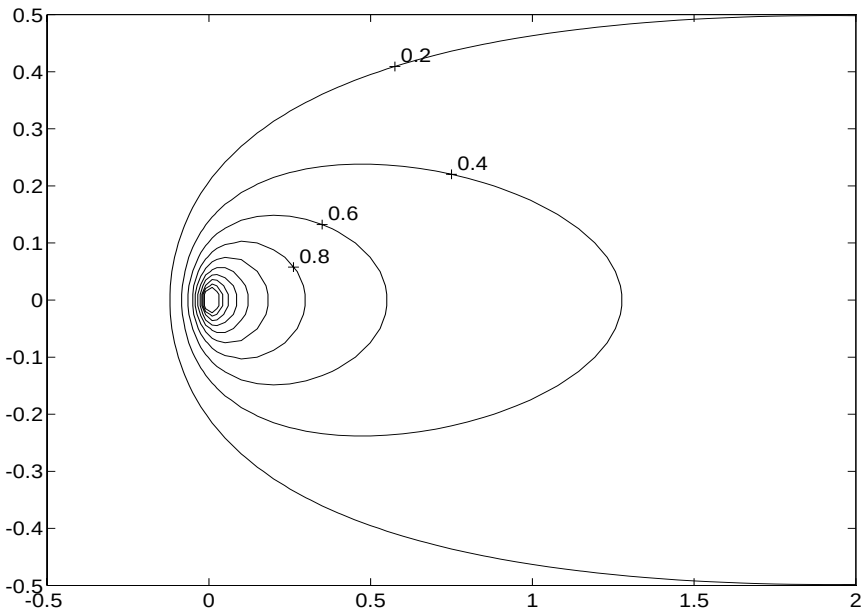


Figure 3.1: Point Source Exact Solution: V=0.15

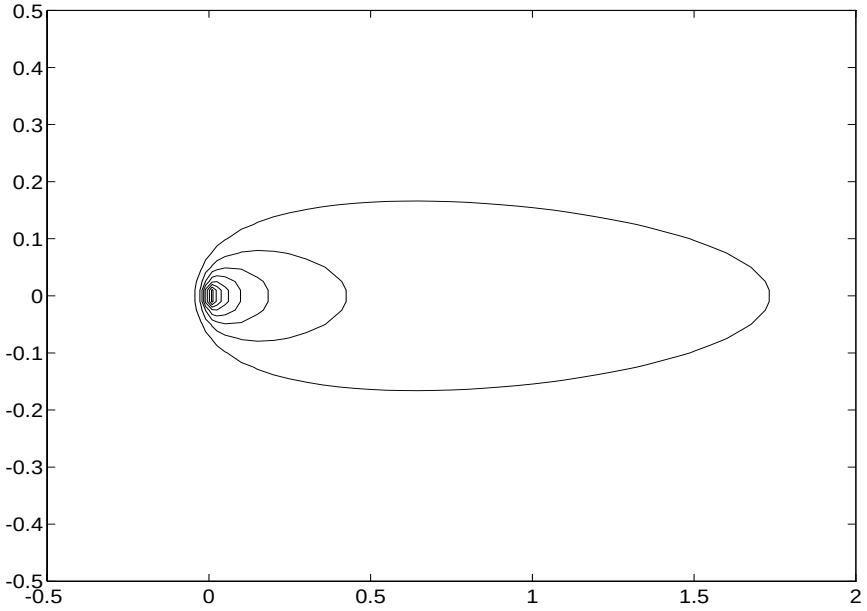


Figure 3.2: Point Source Exact Solution: V=0.45

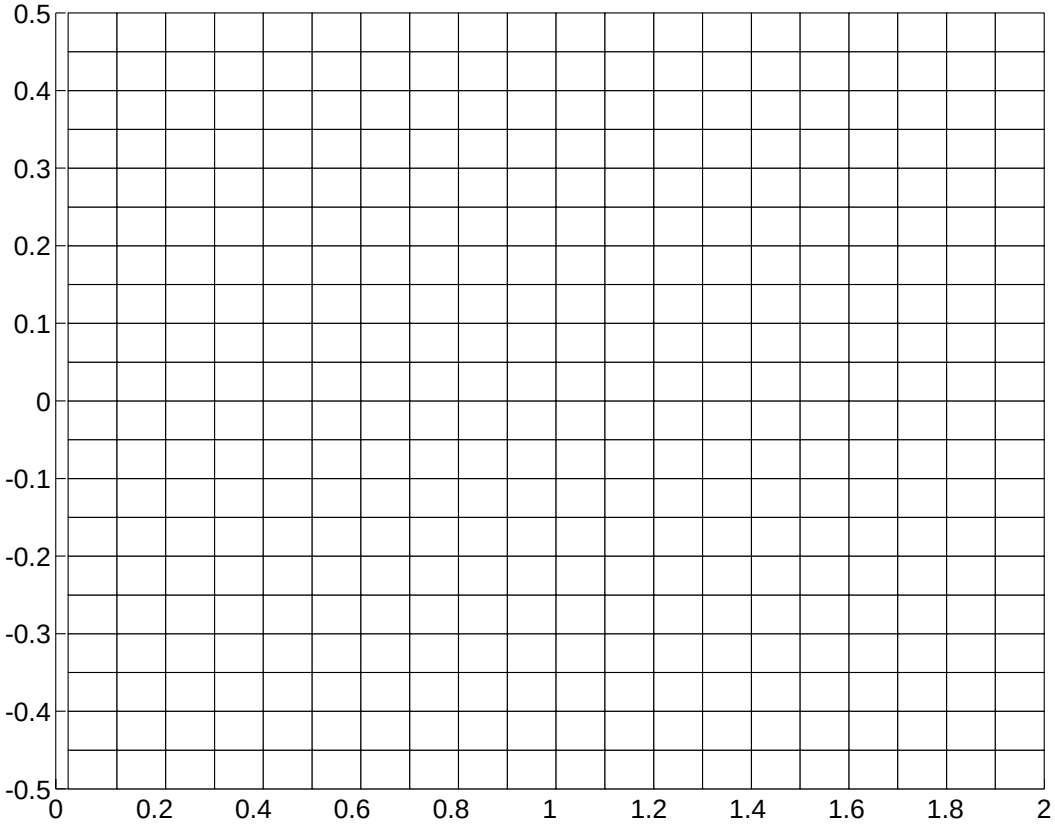


Figure 3.3: Mesh for Grid Aligned Point Source Problem

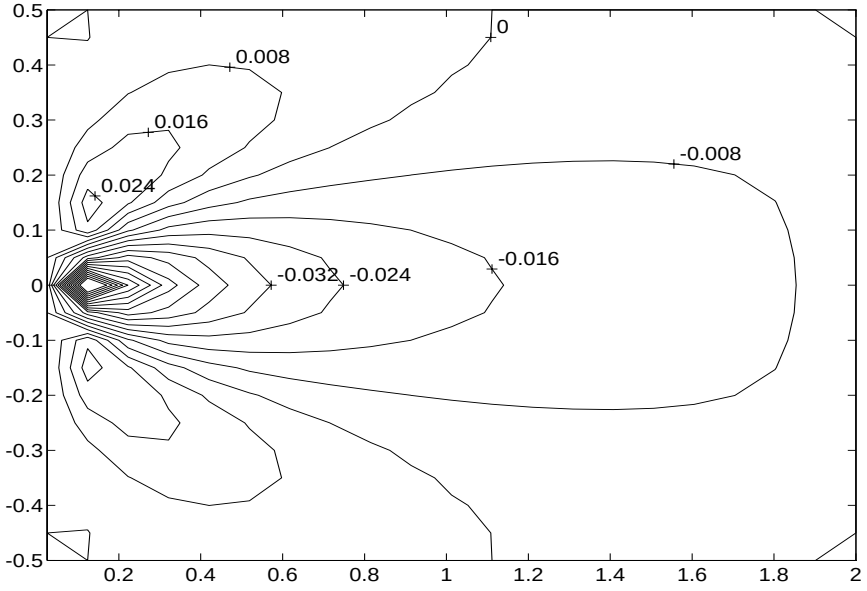


Figure 3.4: Exact error:V=0.15: UDS approximate solution

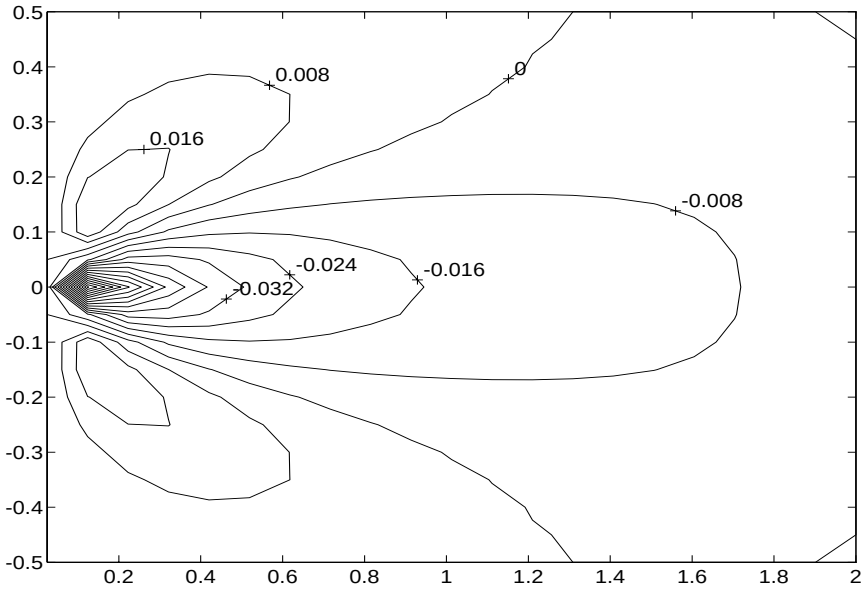


Figure 3.5: Error estimate:V=0.15: UDS approximate solution

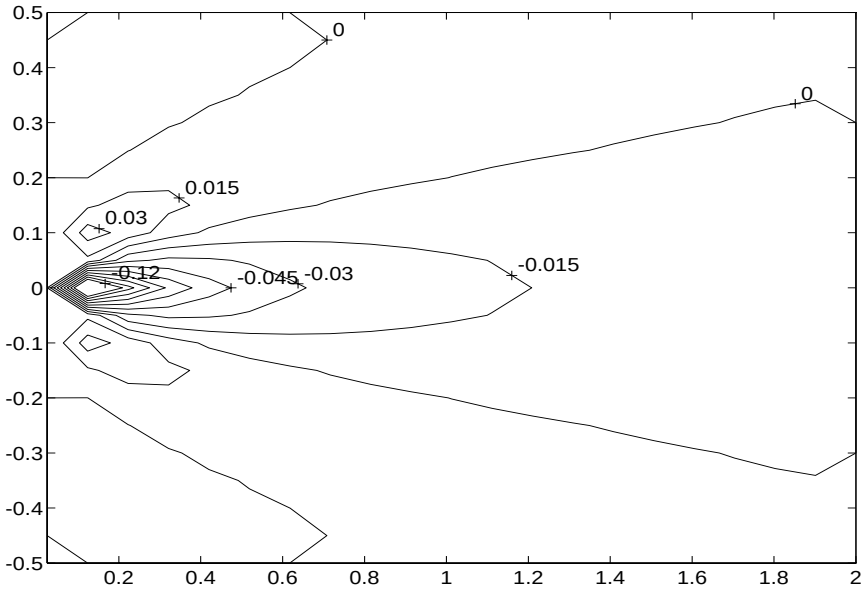


Figure 3.6: Exact error:V=0.45: UDS approximate solution

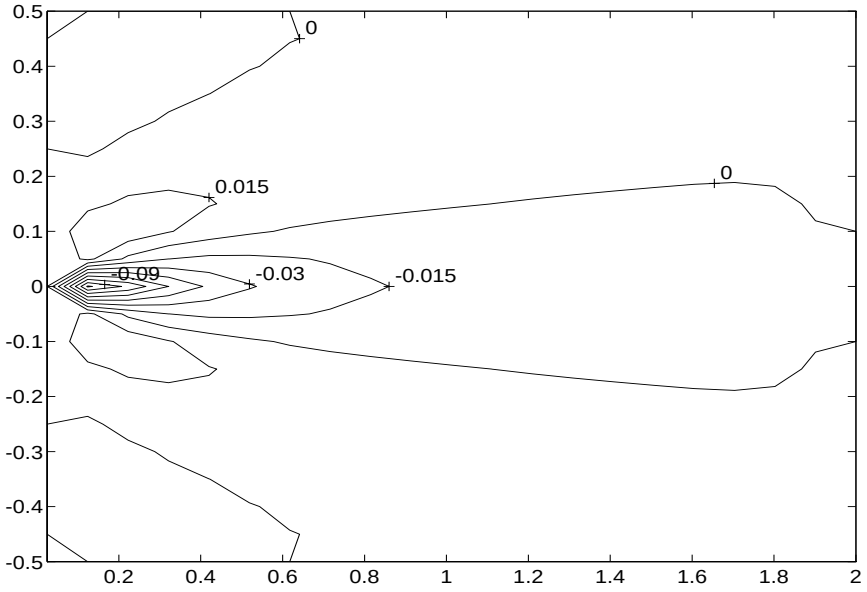


Figure 3.7: Error estimate:V=0.45: UDS approximate solution

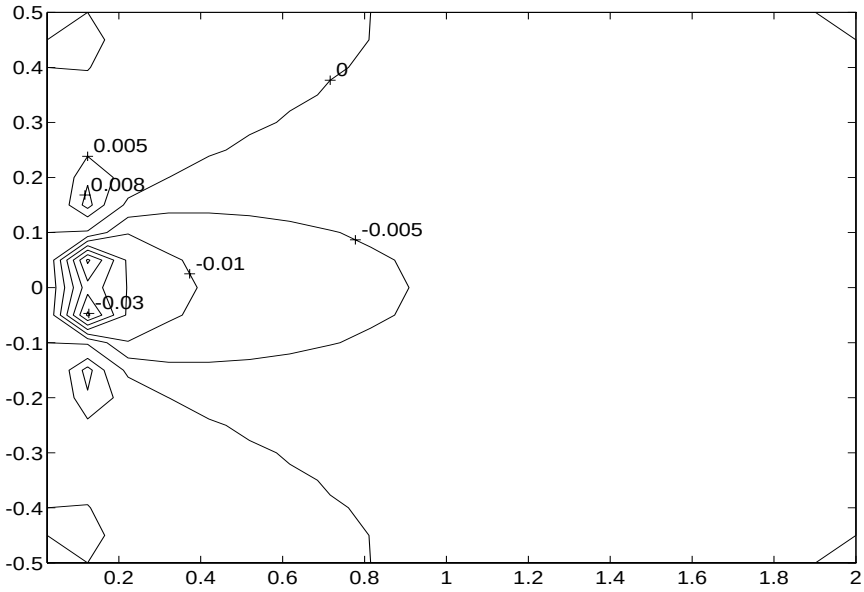


Figure 3.8: Exact error:V=0.15: LPS+PAC approximate solution

Some observation for these plots can be made at this point:

- The level of error ranges between 10-15% relative error, with the maximum occurring one node in from the inlet boundary condition.
- The zero error contours are remarkably congruent between the estimate and exact error fields. This results in the error estimator consistently having the correct sign for the error. The qualitative distribution of the error is quite well predicted for all cases.
- In the regions above and below the high error central core, where the overall level of error is relatively small, the error estimate is within 10% of the actual error (The relative error in the error estimate is less than 10%)
- On a line through the point source along the velocity vector, where the error is the highest, the error estimator under-predicts the magnitude of the error by a quite consistent factor of 2/3.

The same two problems were also run using the LPS+PAC approximate solution. The low speed flow results are shown here. The observations made apply equally well to the high speed flow case. Results are shown in figures 3.8 and 3.9

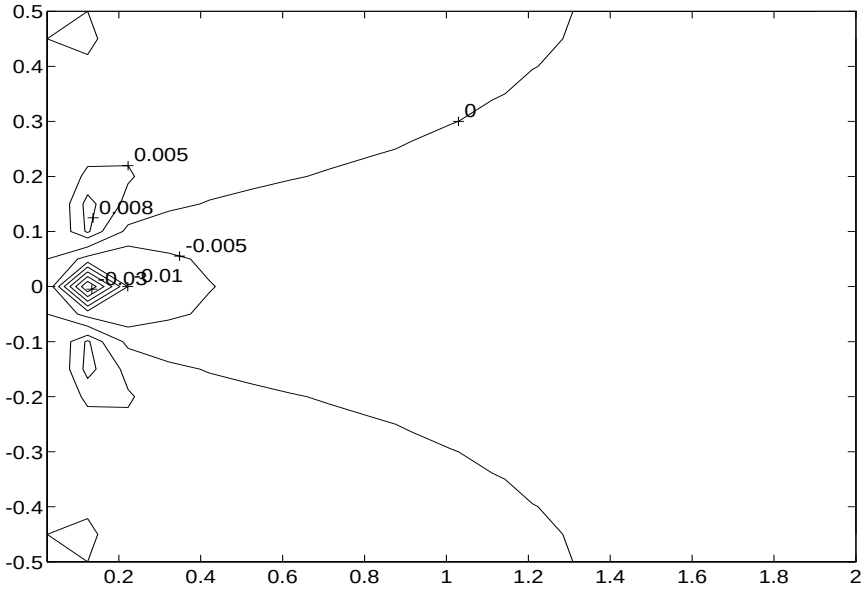


Figure 3.9: Error estimate:V=0.15: LPS+PAC approximate solution

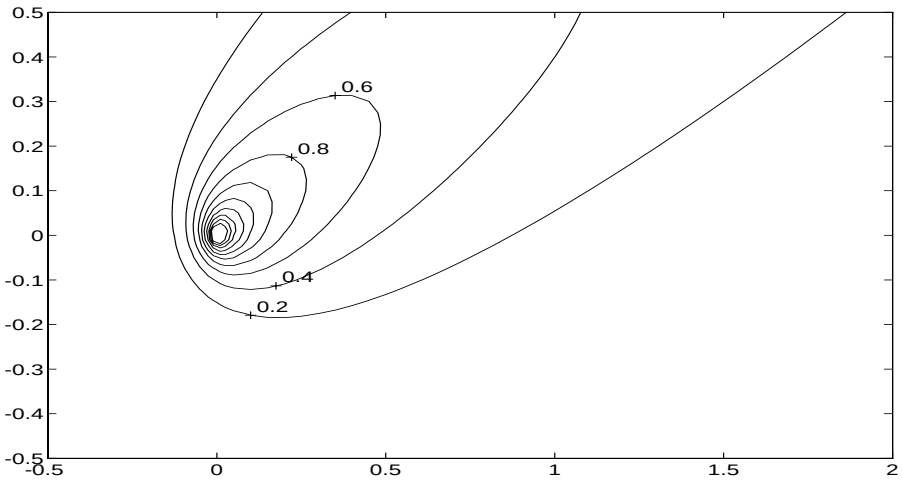


Figure 3.10: Point Source Exact Solution: V=0.15 inclined at 30 degrees

Observations of the numerical results for the LPS+PAC approximate solution indicate:

- The LPS+PAC solutions are, as expected, noticeably better than the UDS solutions, for all problems.
- Qualitatively and quantitatively, the error estimation results have a comparable quality to the UDS approximate solution results.

This last point is important. Until this point, we have only analyzed bilinear residual estimates operating on first-order approximate solutions (UDS solutions). LPS+PAC is the same order of numerical approximation as a full bilinear approximation (second-order). At the start of this research project, there were doubts that a second-order residual estimator would be of any use for these higher order approximations. These results would seem to indicate that the order of the numerical scheme is not the only factor to consider when generating a residual estimator. In this case, a bilinear residual estimate performed admirably.

3.2.2 Grid Skewed Flow

As was done in §3.2.1, a low speed and a high speed flow problem are considered (V=0.15 and V=0.45). The difference now being that the flow vector is skewed by 30 degrees to the X-coordinate direction. The exact solutions for these problems are shown in figures 3.10 and 3.11 respectively.

The same computational mesh is used to generate the approximate solutions, except it is repositioned with respect to the origin to make more effective use of the mesh. The results utilizing the UDS approximate solution are shown in figures 3.12 through 3.15.

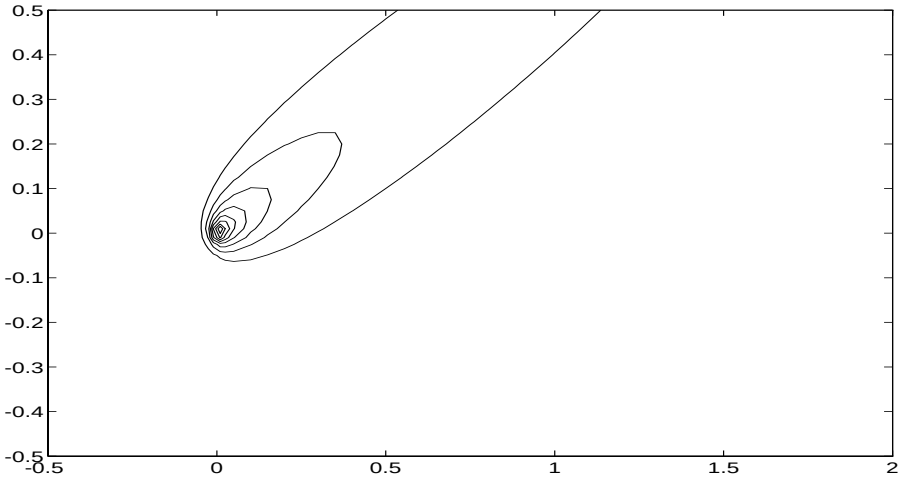


Figure 3.11: Point Source Exact Solution: V=0.45 inclined at 30 degrees

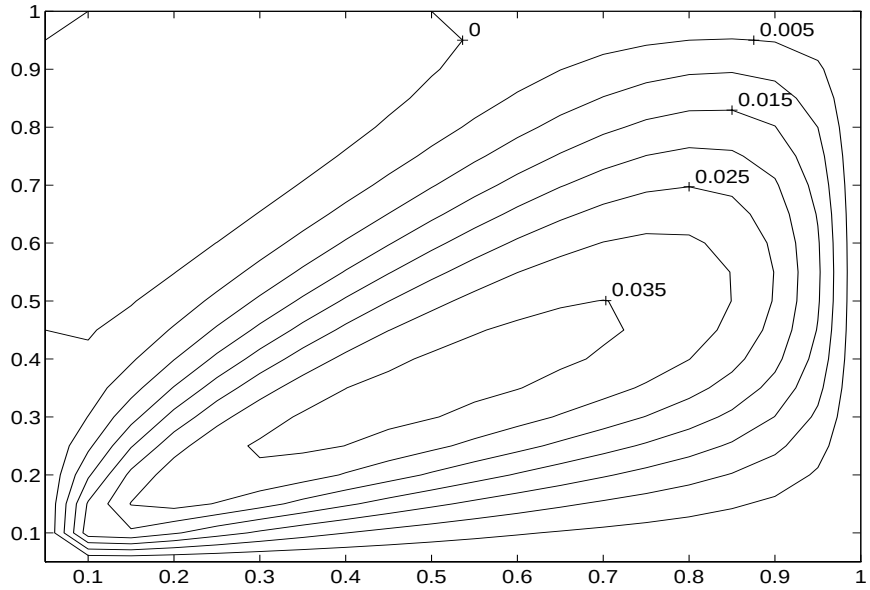


Figure 3.12: Exact error: V=0.15 inclined at 30 degrees: UDS approximate solution

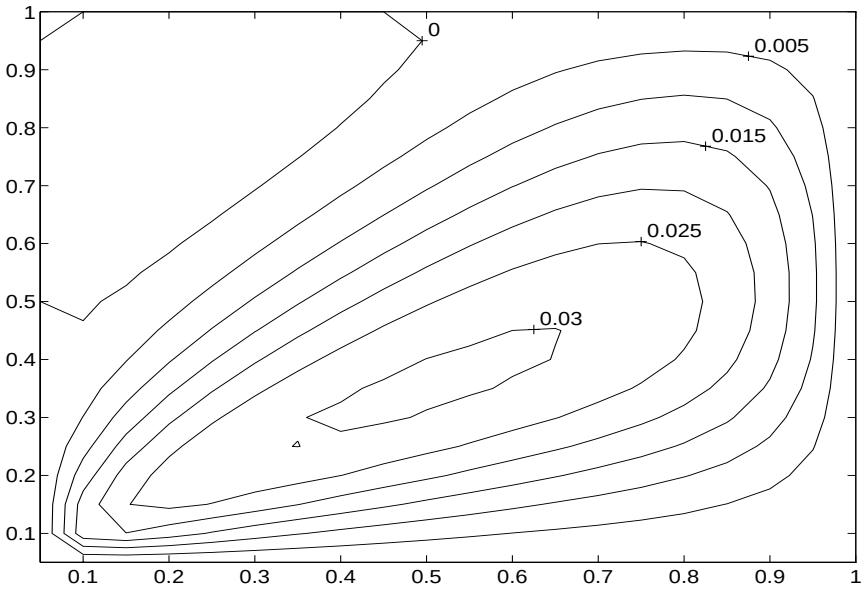


Figure 3.13: Error estimate:V=0.15 inclined at 30 degrees: UDS approximate solution

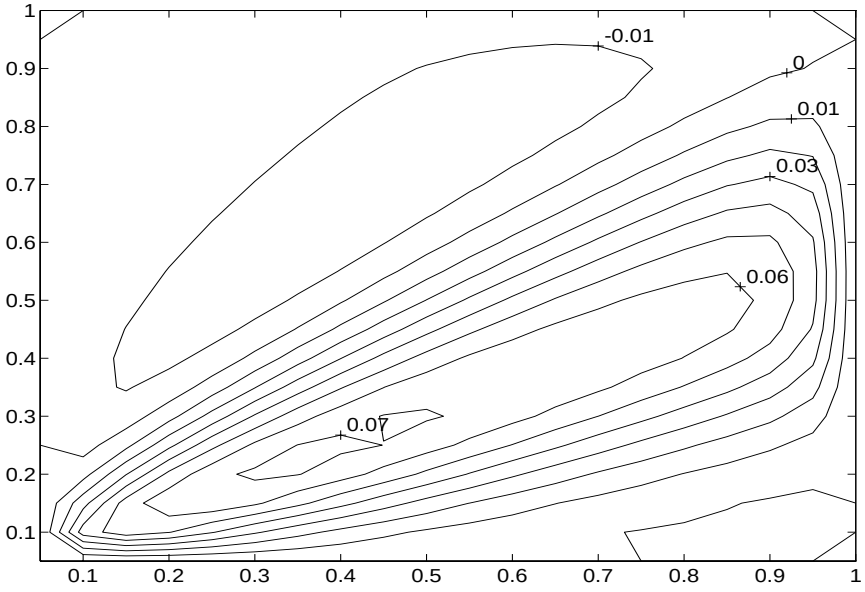


Figure 3.14: Exact error: V=0.45 inclined at 30 degrees: UDS approximate solution

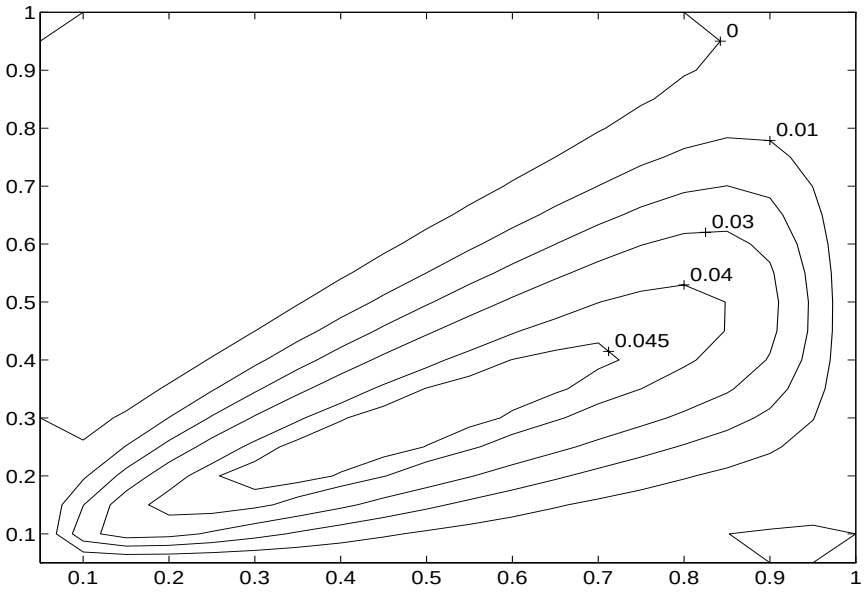


Figure 3.15: Error estimate:V=0.45 inclined at 30 degrees: UDS approximate solution

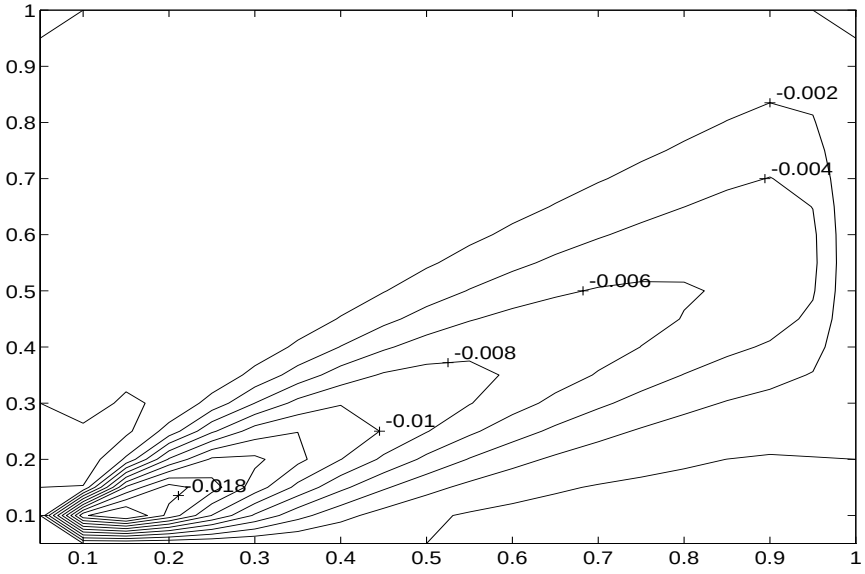


Figure 3.16: Exact error: V=0.45 inclined at 30 degrees: LPS+PAC approximate solution

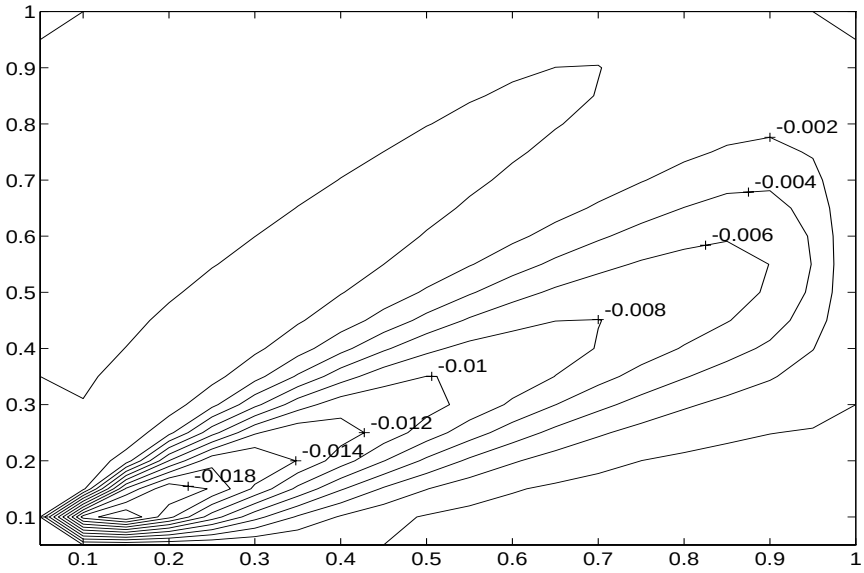


Figure 3.17: Error estimate: V=0.45 inclined at 30 degrees: LPS+PAC approximate solution

Observations are the same as for the grid aligned flow problem. The error estimate performs well for both cases examined.

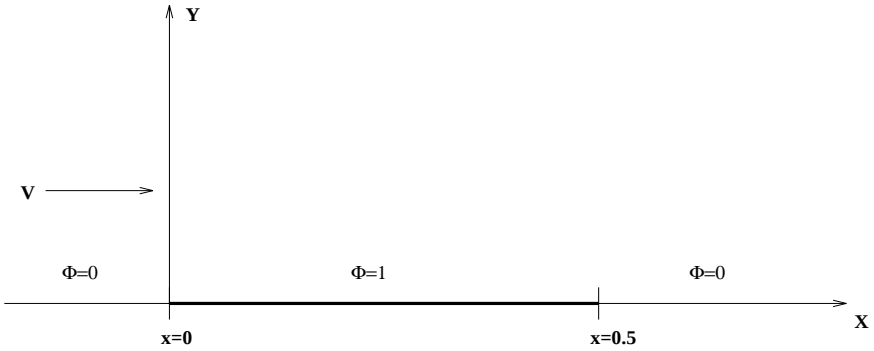
Similar to §3.2.1, the LPS+PAC approximate solution results are similar between the high and low speed problems. An indicative example is shown in figures 3.16 and 3.17

As with the UDS results, the observations from §3.2.1 can be applied equally well to the LPS+PAC results.

3.3 Boundary Layer Problems

3.3.1 Discontinuous Dirichlet Problem

Consider the following positive-Y plane and boundary conditions:



Introducing a new scalar variable,

$$z = \frac{V}{2\Gamma}\sqrt{(x-r)^2 + y^2}, \tag{3.3}$$

the exact solution for this problem can be evaluated by calculating

$$\Phi(x,y) = e^{\frac{Vx}{2\Gamma}} \frac{Vy}{2\Gamma} \int_0^{0.5} \frac{K_1(z)}{z} e^{\frac{-Vr}{2\Gamma}}\, dr. \tag{3.4}$$

The integral in 3.4 becomes difficult to calculate as you approach the singular region ($y = 0, 0 \leq x \leq \frac{1}{2}$). In addition, the exact solution has singularities at $(x,y) = (0,0)$ and $(\frac{1}{2}, 0)$. For these reasons the numerical problem posed has been offset from the x-axis. This results in sharp Dirichlet boundary gradients, but not singularities, while still preserving the analytic boundary layer field. Figures 3.18 and 3.19 show what the exact solutions for to two cases (low speed flow and high speed flow). Each contour represents a 0.1 scalar step. A finer grid spacing was used for this problem compared to the point source problem. The grid is shown in figure 3.20. The implemented Dirichlet boundary condition is shown in figure 3.21.

The results of the numerical experimentation with the UDS approximate solutions are shown in figures 3.22 through 3.25.

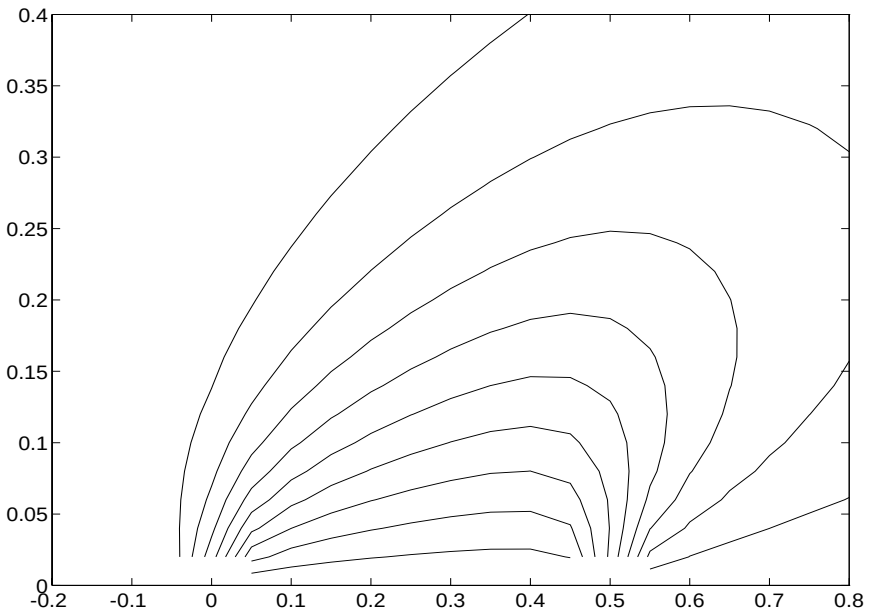


Figure 3.18: Discontinuous Dirichlet Exact Solution: V=0.15

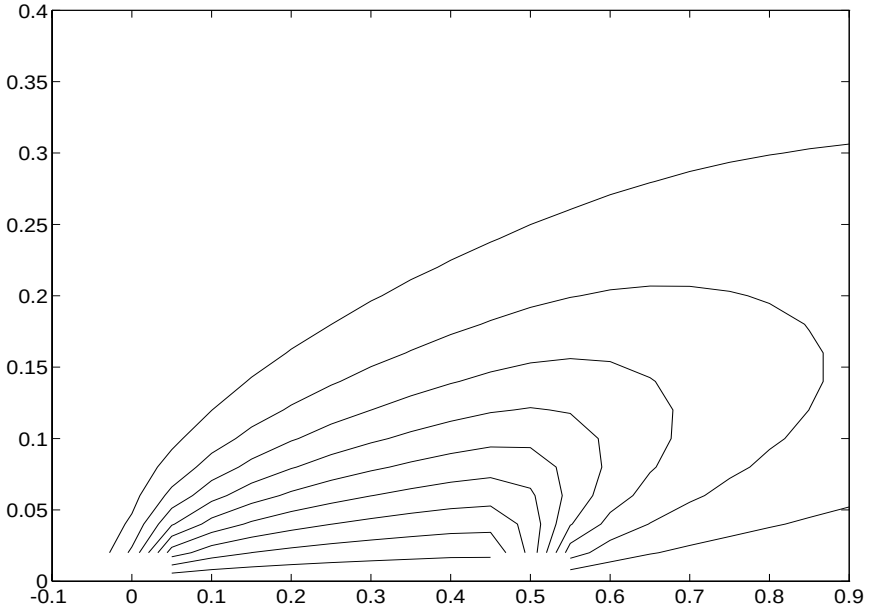


Figure 3.19: Discontinuous Dirichlet Exact Solution: V=0.45

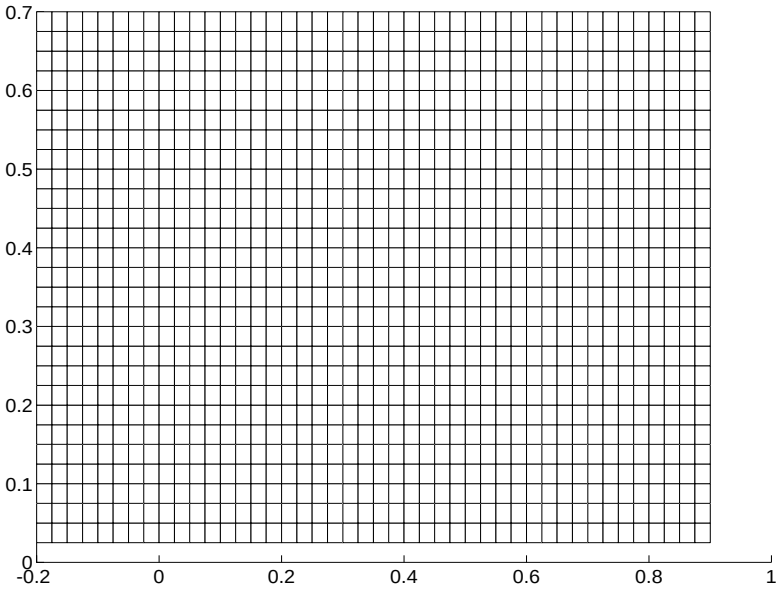


Figure 3.20: Mesh for discontinuous Dirichlet boundary condition problem

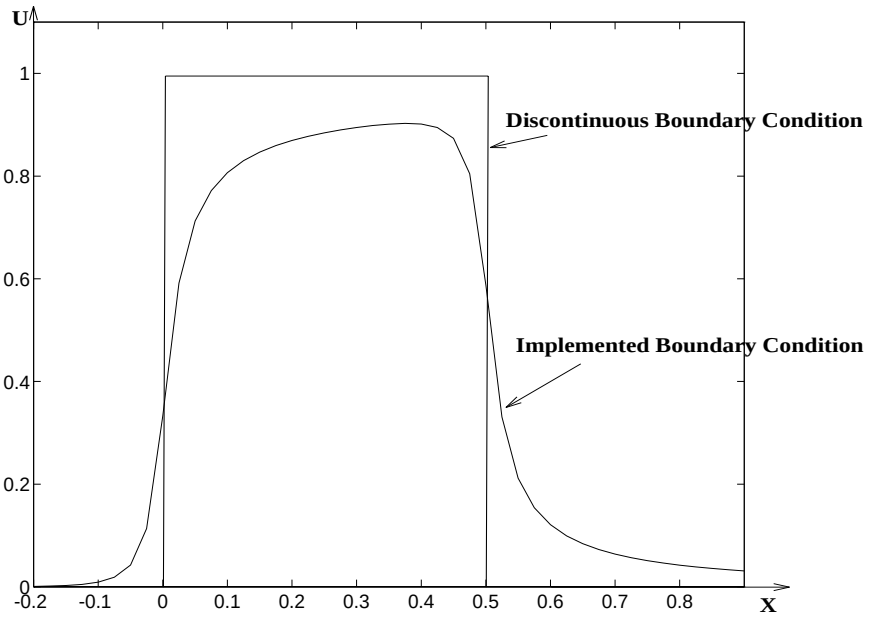


Figure 3.21: Discontinuous Boundary Condition vs. Actual Implementation

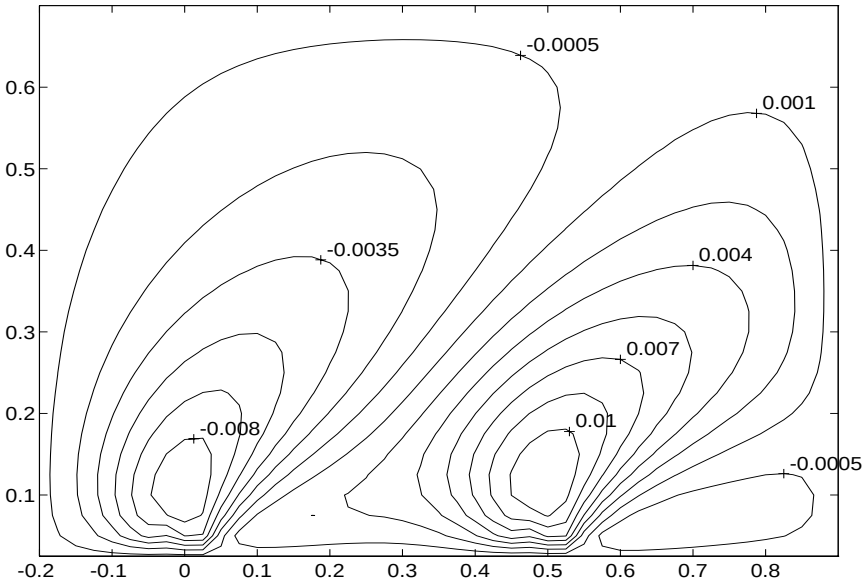


Figure 3.22: Exact error:V=0.15: UDS approximate solution

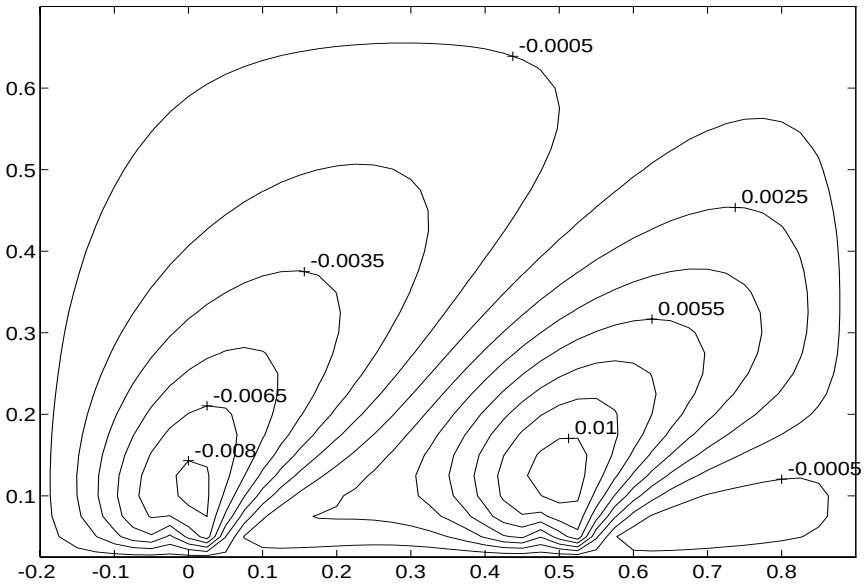


Figure 3.23: Error estimate:V=0.15: UDS approximate solution

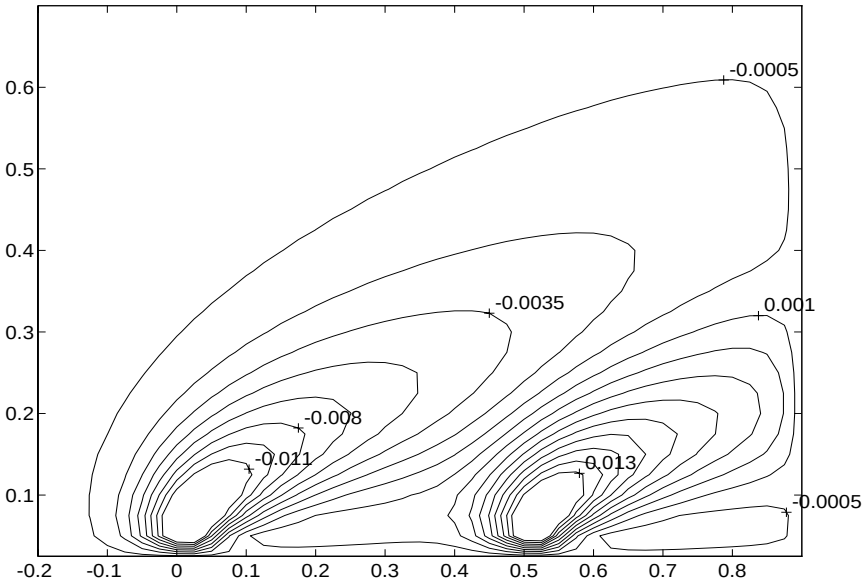


Figure 3.24: Exact error:V=0.45: UDS approximate solution

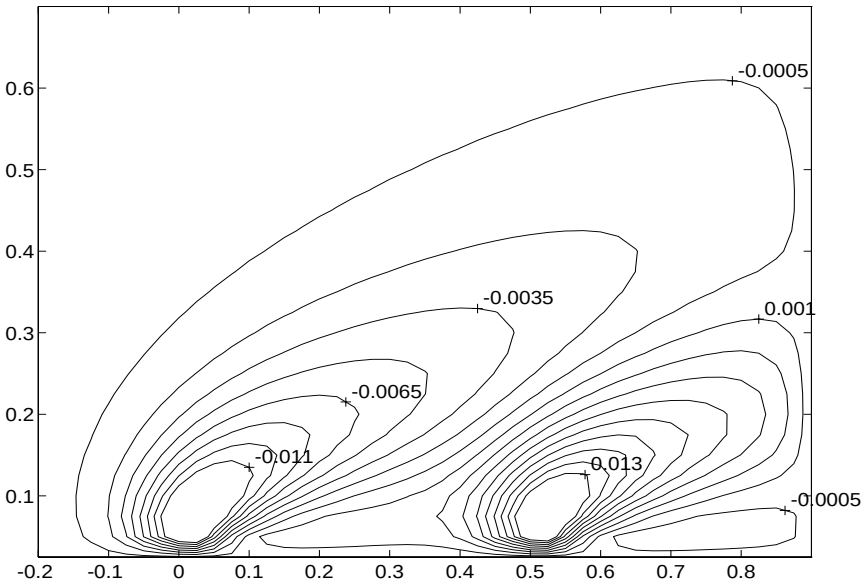


Figure 3.25: Error estimate:V=0.45: UDS approximate solution

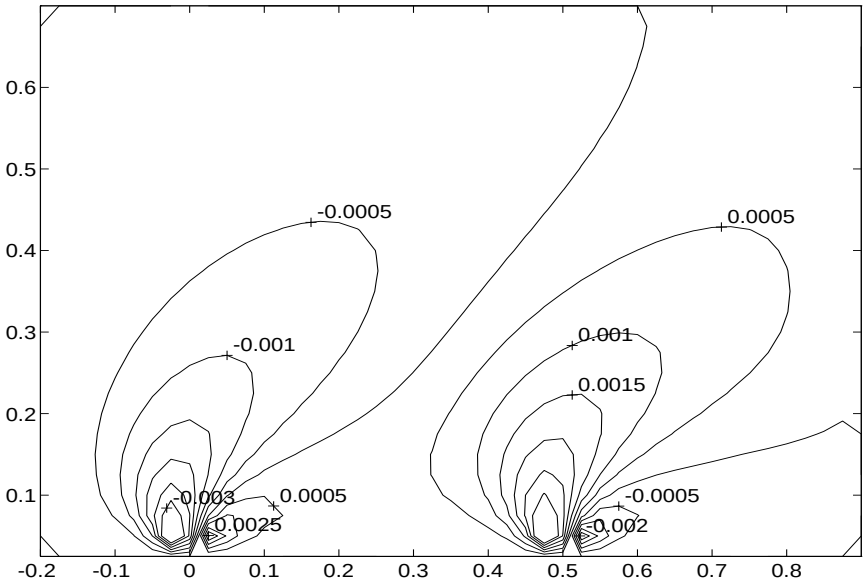


Figure 3.26: Exact error:V=0.15: LPS+PAC approximate solution

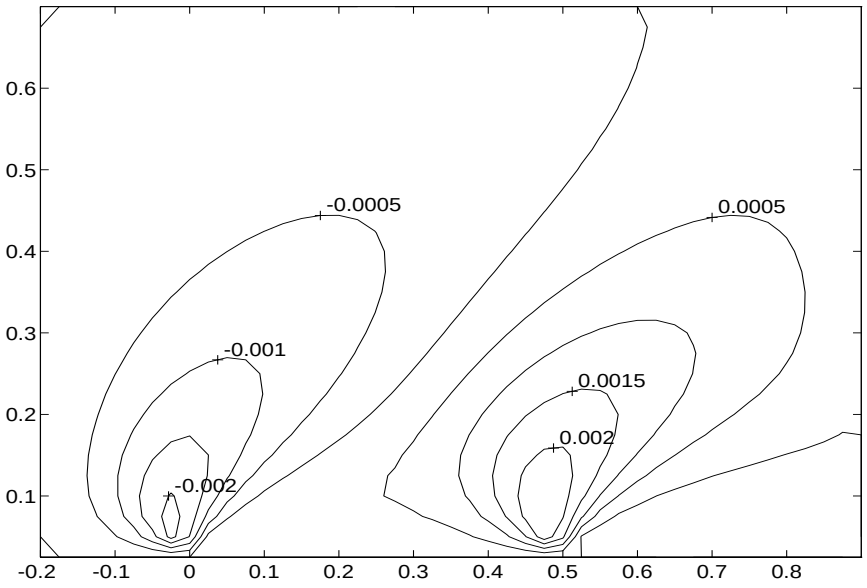


Figure 3.27: Error estimate:V=0.15: LPS+PAC approximate solution

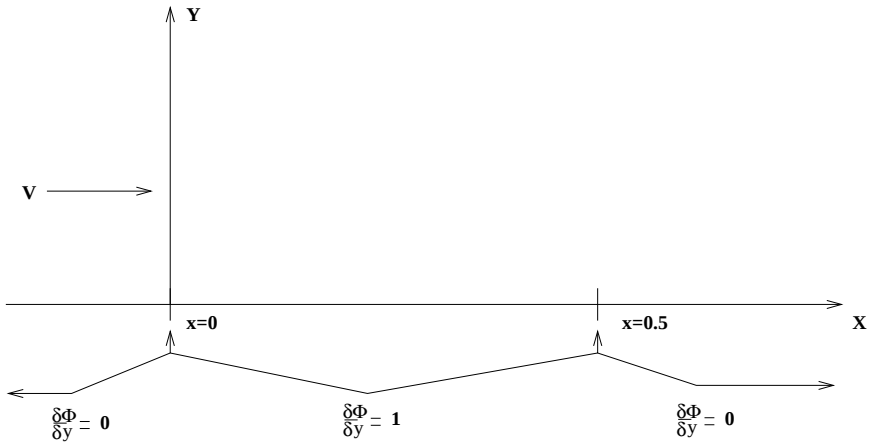
Some observations that can be made:

- With the finer grid, we see that the error now ranges from .05-.2% error for large regions away from the boundaries, up to 1.5-2% error close to the x-axis boundary. The dominant feature for these error fields appears to be large error production in regions of large streamwise solution gradients. UDS performs poorly for this type of flow phenomenon.
- As before, the qualitative distribution of the error is well estimated. Visual inspection of contour plots sometimes cannot distinguish between the exact error and error estimate.
- In regions of low error levels, the relative error in the error estimate runs from 0-2%. In those regions where the error level is higher, this can reach a maximum of roughly 5% error.

As with the point source problems, the numerical results for the LPS+PAC approximate solutions are similar in their observable characteristics. An example is given in figures 3.26 and 3.27. The same observations made for the point source problems can be granted to the Dirichlet boundary problem.

3.3.2 Discontinuous Neumann Problem

Consider the following positive-Y half plane and boundary conditions:



Again, introducing a new scalar variable,

$$z = \frac{V}{2\Gamma}\sqrt{(x-r)^2 + y^2}, \tag{3.5}$$

the exact solution for this problem can be evaluated by calculating

$$\Phi(x,y) = e^{\frac{Vx}{2\Gamma}} \frac{1}{\pi} \int_0^{0.5} K_0(z) e^{\frac{-Vr}{2\Gamma}}\, dr. \tag{3.6}$$

as detailed in Appendix C.

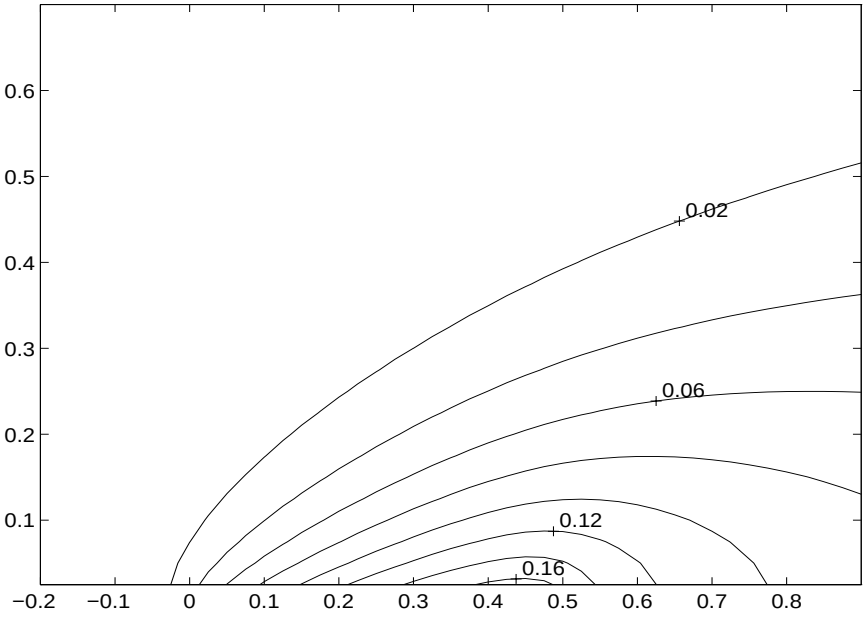


Figure 3.28: Discontinuous Neumann Problem Exact Solution: V=0.15

The exact solutions are shown in figures 3.28 and 3.29 respectively.

The exact and estimated error for the UDS approximate solutions are shown in figures 3.30 through 3.33. As with the Dirichlet boundary condition problems, there is, in general, good agreement between the exact and estimated error.

The LPS+PAC approximate solutions do not demonstrate any significant difference from the UDS approximate solution in terms of the correspondence between the exact and estimated error. The low speed flow results are shown in figures 3.34 and 3.35.

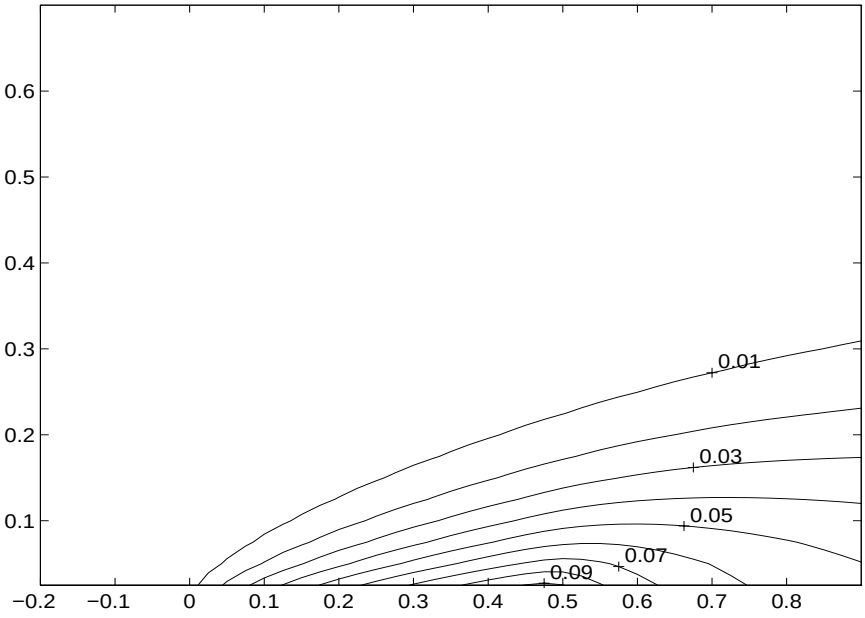


Figure 3.29: Discontinuous Neumann Problem Exact Solution: V=0.45

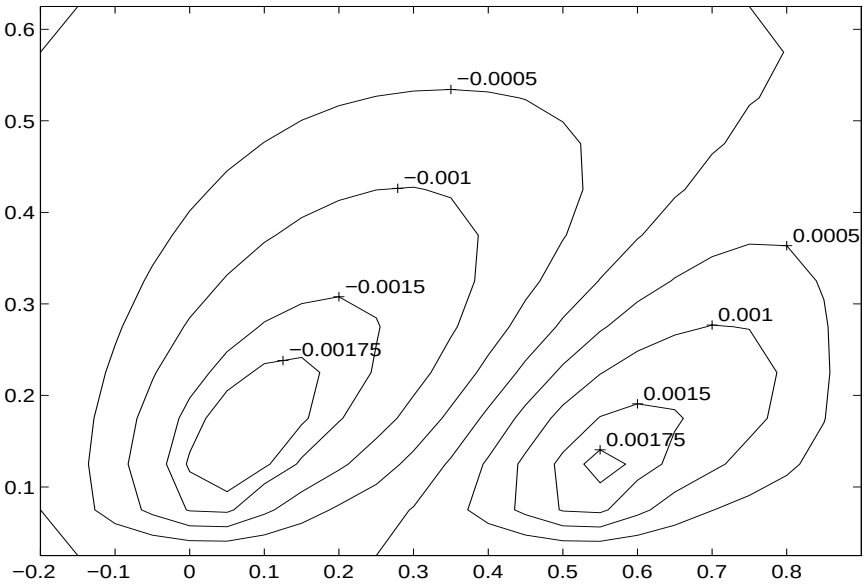


Figure 3.30: Exact error: V=0.15: UDS approximate solution

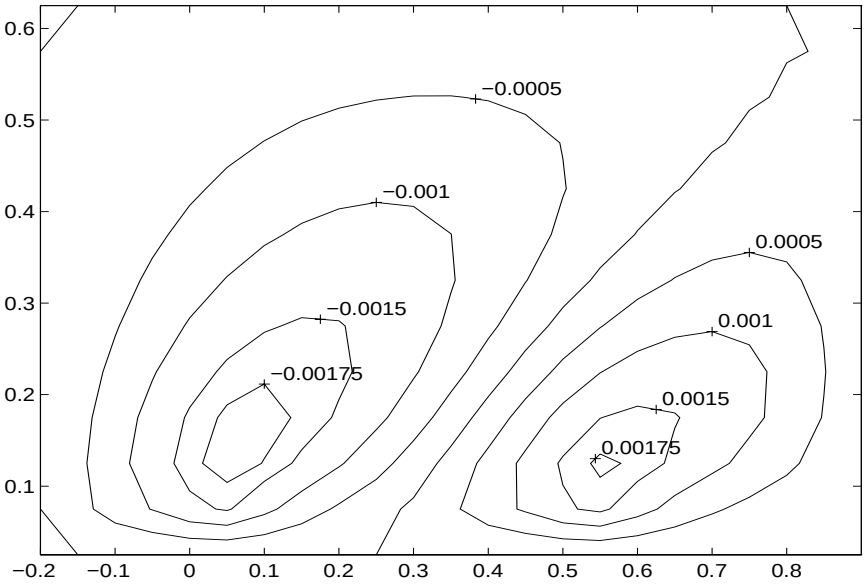


Figure 3.31: Error estimate: V=0.15: UDS approximate solution

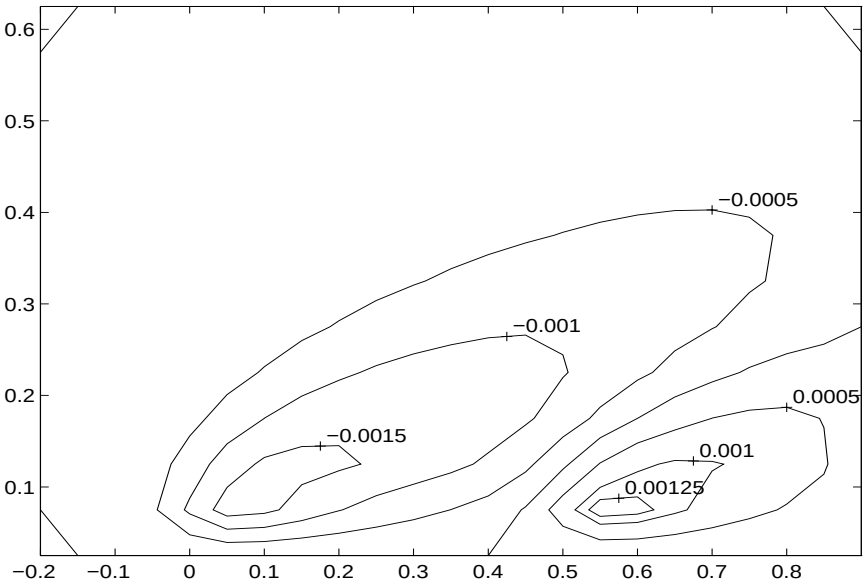


Figure 3.32: Exact error: V=0.45: UDS approximate solution

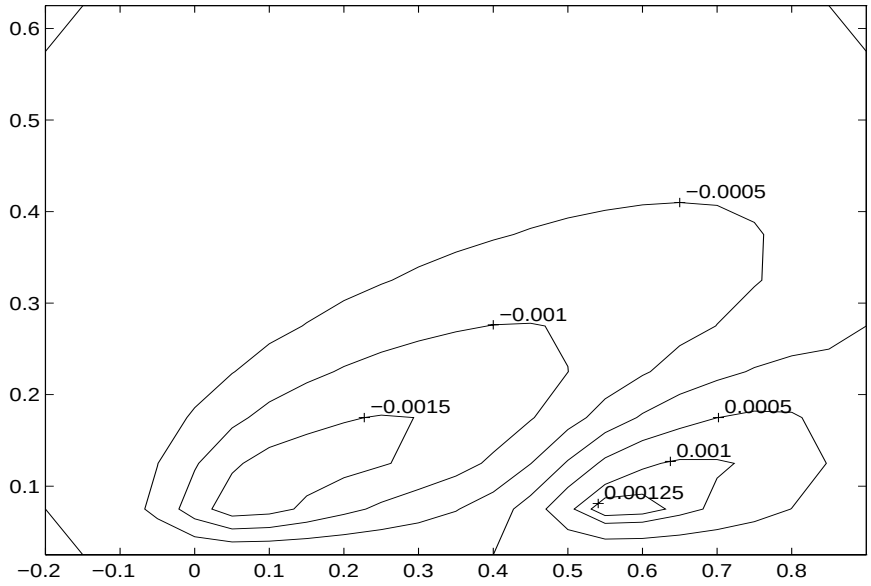


Figure 3.33: Error estimate: V=0.45: UDS approximate solution

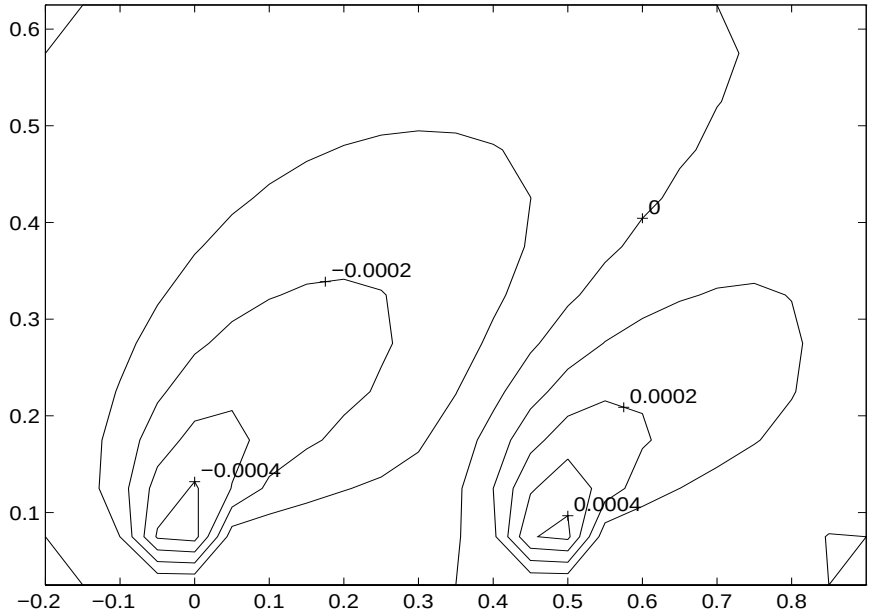


Figure 3.34: Exact error: V=0.15: LPS+PAC approximate solution

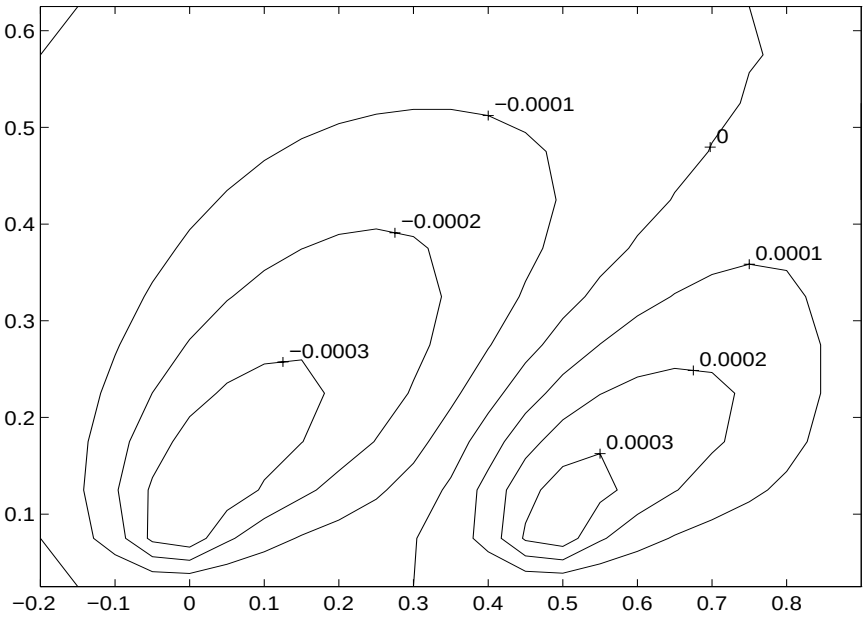


Figure 3.35: Error estimate: V=0.15: LPS+PAC approximate solution

3.4 Summary

The Global Error Equation Method was tested on a spectrum of 2-D scalar advection diffusion problems, for two different types of approximate solutions (UDS and LPS+PAC). All results showed good correspondence between the error estimate and the actual error.

The scalar advection-diffusion is a rather simple fluid transport model. Success with this equation indicates that there is now a need to extend the work to a more realistic fluid dynamic model. In the next chapter we report on work with the Navier-Stokes equations. Although more physically realistic, they require dealing with a non-linear vector equation set. These new elements require additional care.

Chapter 4

Navier-Stokes Equations

In this chapter we move our study from the linear scalar advection-diffusion equation to the Navier-Stokes equations.

Consider the basic Navier-Stokes equations, representing steady incompressible laminar flow in two dimensions with no body forces. The equations representing the conservation of mass, and linear momentum in the x and y coordinate directions are respectively:

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0, \tag{4.1}$$

$$\rho\left[U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y}\right] + \frac{\partial P}{\partial x} - \mu\left[\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right] = 0, \tag{4.2}$$

$$\rho\left[U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y}\right] + \frac{\partial P}{\partial y} - \mu\left[\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}\right] = 0. \tag{4.3}$$

U and V are the fluid velocities in the x and y coordinate directions respectively, P is the fluid pressure, ρ is the fluid density and μ is the fluid dynamic viscosity.

Equations (4.1)to (4.3), along with appropriate boundary conditions can be written in operator notation as

$$\mathcal{N}(U, V, P) = (0, 0, 0). \tag{4.4}$$

The Navier-Stokes equations, as presented, have two major differences from the advection-diffusion equation studied in earlier chapters:

1. Instead of a single equation for the scalar field Φ, the Navier-Stokes equations are a coupled system of equations for the vector field $[U, V, P]$

2. The equations are non-linear

The error estimation principles of chapter 1.2 are reformulated to reflect these differences.

4.1 The Error Equation

The error will now be a three component vector field. Thus if $[u, v, p]$ represents an approximate solution of (4.4), then the error in $[u, v, p]$ is given as

$$\vec{\mathcal{E}} = [\mathcal{E}_u, \mathcal{E}_v, \mathcal{E}_p] \equiv [U, V, P] - [u, v, p]. \tag{4.5}$$

Now, the *Exact Operator Residual* can be expressed as

$$[R_{mass}, R_{x\_mom}, R_{y\_mom}] \equiv \mathcal{N}(u, v, p). \tag{4.6}$$

By taking (4.5) and substituting into (4.4) and separating terms (keeping in mind how the exact operator residual has been defined, (4.6), the non-linear Error Equation, can be expressed as:

$$\frac{\partial \mathcal{E}_u}{\partial x} + \frac{\partial \mathcal{E}_v}{\partial y} = -R_{mass}, \tag{4.7}$$

$$\rho\left[\left(\mathcal{E}_u\frac{\partial \mathcal{E}_u}{\partial x} + \mathcal{E}_v\frac{\partial \mathcal{E}_u}{\partial y}\right)_a + \left(u\frac{\partial \mathcal{E}_u}{\partial x} + v\frac{\partial \mathcal{E}_u}{\partial y}\right)_b + \left(\frac{\partial u}{\partial x}\mathcal{E}_u + \frac{\partial u}{\partial y}\mathcal{E}_v\right)_c\right] + \frac{\partial \mathcal{E}_p}{\partial x} - \mu\left[\frac{\partial^2 \mathcal{E}_u}{\partial x^2} + \frac{\partial^2 \mathcal{E}_u}{\partial y^2}\right] = -R_{x\_mom}, \tag{4.8}$$

$$\rho\left[\left(\mathcal{E}_u\frac{\partial \mathcal{E}_v}{\partial x} + \mathcal{E}_v\frac{\partial \mathcal{E}_v}{\partial y}\right)_a + \left(u\frac{\partial \mathcal{E}_v}{\partial x} + v\frac{\partial \mathcal{E}_v}{\partial y}\right)_b + \left(\frac{\partial v}{\partial x}\mathcal{E}_u + \frac{\partial v}{\partial y}\mathcal{E}_v\right)_c\right] + \frac{\partial \mathcal{E}_p}{\partial y} - \mu\left[\frac{\partial^2 \mathcal{E}_v}{\partial x^2} + \frac{\partial^2 \mathcal{E}_v}{\partial y^2}\right] = -R_{y\_mom}. \tag{4.9}$$

where the subscripts a, b and c are included to aid in our discussion below.

Equations (4.7) to (4.9) can be expressed in operator form,

$$\mathcal{N}_{\mathcal{E}}(\mathcal{E}_u, \mathcal{E}_v, \mathcal{E}_p) = -[R_{mass}, R_{x\_mom}, R_{y_m om}]. \tag{4.10}$$

The resulting equations are similar to the advection-diffusion error equation (1.4) in that the error equation utilizes the exact operator residual as an error 'source' term. The error, then, is transported through the solution domain to generate the error field. The notable difference being that the Navier-Stokes error equations are non-linear.

Full Linearization

To effect a computationally practical error estimation, it is desirable to solve a linearized form of (4.10). To allow for *a posteriori* evaluation the equations are linearized about the current approximate solution $[u, v, p]$. This is equivalent to discarding the terms highlighted in equations (4.8) and (4.9) with the $()_a$ subscript. Let this new system of PDEs be represented by the continuous linear operator $\overline{\mathcal{N}_{\mathcal{E}}}^{full}$,

$$\overline{\mathcal{N}_{\mathcal{E}}}^{full}(\mathcal{E}_u, \mathcal{E}_v, \mathcal{E}_p) = -[R_{mass}, R_{x\_mom}, R_{y\_mom}]. \tag{4.11}$$

The resulting operator, $\overline{\mathcal{N}_{\mathcal{E}}}^{full}$, is equivalent to the Newton-Raphson linearization of the Navier-Stokes equations. Spectral analysis of $\overline{\mathcal{N}_{\mathcal{E}}}^{full}$ is still an active area of research. We will see later that the inversion of $\overline{\mathcal{N}_{\mathcal{E}}}^{full}$ can be problematic. Spectral analysis of $\overline{\mathcal{N}_{\mathcal{E}}}^{full}$ is beyond the scope of this work; we proceed under the assumption that $\overline{\mathcal{N}_{\mathcal{E}}}^{full}$ possesses both left and right inverses within the range of problems studied.

Semi-linearization

As an alternative to the full linearization of (4.10), a *semi-linearization* is also constructed. The semi-linearization ignores terms $()_a$ and $()_c$ of equation (4.10). This remaining equation set, $\overline{\mathcal{N}_{\mathcal{E}}}^{semi}$, is essentially a system of weakly coupled linear advection-diffusion equations. All

solutions of the linear advection-diffusion equation are stable. Through proper discretization (i.e., upwinding etc.), the numerical problem can be guaranteed to be stable. $\overline{\mathcal{N}_{\mathcal{E}}}^{semi}$ is the same partial differential operator that is at the root of a frozen coefficient non-linear iteration commonly used to solve the Navier-Stokes equations.

This leads to another (practical) advantage of the semi-linearized error equation: since the error equation is now equivalent to frozen coefficient non-linear iteration for the Navier-Stokes equation, the mechanism used to assemble the algebraic equations can be used to solve the original approximate solution.

4.2 Error Estimation

In the previous section two linearized partial differential equations were presented as a basis for *a posteriori* error estimation of numerical solutions of Navier-Stokes equations. In this section these equations are expressed in a FVM context. As was done in §1.4, an integral formulation is introduced. A detailed description of the integral formulation and discretization is given in appendix D, an abridged description is given here.

Let $\mathcal{V}$ be a mesh of the domain Ω, divided into finite volumes, or cells v_k for $k = 1$ to N. Equation (4.4) implies

$$\int_{\partial v_k} \mathcal{N}^\star(U, V, P)\, ds = [0, 0, 0], \tag{4.12}$$

where $\mathcal{N}^\star(U, V, P)$ represents the flux of mass, and momentum through a differential surface ds .

To generate an approximate solution of (4.12) the FVM replaces the continuous distribution $[u, v, p]$, by values of a discrete distribution $[u^h, v^h, p^h]$. The discretized form of $\mathcal{N}^\star(u, v, p)$ will be written as $N^\star(u^h, v^h, p^h)_{FVM}$,

$$\int_{\partial v_k} N^\star(u^h, v^h, p^h)_{FVM}\, ds = [0, 0, 0], \tag{4.13}$$

where FVM would refer to the particular discretization employed

Equation (4.13), although finite dimensional, is still non-linear. To compute a solution to (4.13) it is necessary to solve a sequence of linear problems. For this work, a frozen coefficient linearization about the current approximate solution is performed. Denote this linearized operator as $\overline{N^\star}$ for future reference.

The residual estimate for $[u^h, v^h, p^h]$ can be written as

$$|v_k|[R^h_{mass}, R^h_{x\_mom}, R^h_{y\_mom}] \equiv \int_{\partial v_k} N^\star_{CD}(u^h, v^h, p^h) ds \tag{4.14}$$

where, recall, the CD subscript indicates a central differencing discretization.

The full linearization of (4.10) is now discretized, in the same way as was done for the original PDE, via an integral formulation. The only difference being the treatment of the $()_c$ terms. For the purpose of this thesis, the $()_c$ terms are discretized using a lumped mass approximation. The discrete error estimate, $[E^h_u, E^h_v, E^h_p]$, can then be obtained by solving

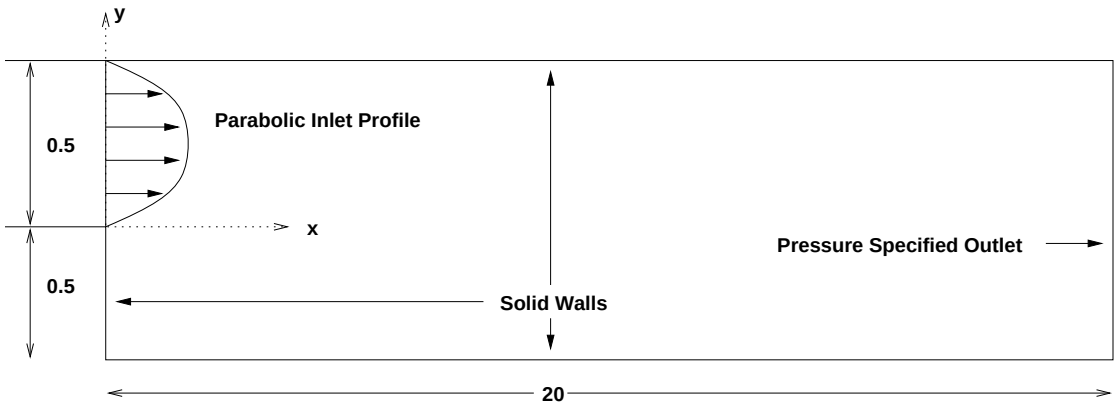


Figure 4.1: Backward Facing Step Problem Definition

$$\int_{\partial v_k} \overline{N\varepsilon}^{\star,full}(E_u^h, E_v^h, E_p^h)ds = -|v_k|[R_{mass}^h, R_{x\_mom}^h, R_{y\_mom}^h] \quad , \quad k = 1 \; to \; N. \tag{4.15}$$

The semi-linearization of (4.10) is treated in the same manner. From a pragmatic standpoint, it is important to note that the discretized semi-linearized error operator is the same as the linearized form of the Navier-Stokes equation used for the original solution which, in our notation would be expressed as

$$\int_{\partial v_k} \overline{N\varepsilon}^{\star,semi}(E_u^h, E_v^h, E_p^h)ds = \int_{\partial v_k} \overline{N}^{\star}(E_u^h, E_v^h, E_p^h)ds = -|v_k|[R_{mass}^h, R_{x\_mom}^h, R_{y\_mom}^h] \; , \quad k = 1 \; to \; N. \tag{4.16}$$

4.3 Computational Experiments

As was done with the 2D advection-diffusion equation, the error estimation technique, as described, was implemented in a computer code and tested on numerical problems for which the exact error in the approximate solution was available. The error estimates were compared to the exact error.

The details of the numerical method are given in appendix D. The Backward Facing Step problem was chosen for the computational experiments. It represents a flow which embodies the essential differences between Navier-Stokes class equations and other simpler flow equations (Stokes equations, Euler equations etc.). It is also a standard benchmark problem for which there are well established numerical solutions.

The setup was the same as given by Gartling [14]. A schematic of the problem definition is given in figure 4.1. The inlet boundary condition specified the velocity profile . The inlet velocity, as in [14], was the parabola $U = 24(0.5 - y)y$. This gave the flow field downstream of the step a maximum velocity of 1.0. The channel expands from a width of 0.5 to 1.0 at $x = 0$. At the outlet, the pressure was set to a constant value and the normal and tangential fluid shear stress was set to zero.

Two Reynolds number flows were solved for, a low Reynolds number flow: Re=100, and a high Reynolds number flow, Re=800 (Reynolds number defined as Re=$(U_{max}h\rho)/\mu$ where h is the channel height). For both cases, five successively finer grids were generated and solved using the higher accuracy LPS+PAC scheme in order to estimate the *exact* solution $[U, V, P]$. The grid dimensions were respectively: 80 X 24, 120 X 36, 344 X 48, 240 X 72, and 320 X 96. The last four grids represent refinements of the first grid by refinement factors of 1.5, 2, 3, and 4. Utilizing an extrapolation analysis recommended by Roache [32], the fine grid results were estimated to contain no more than 0.4% relative error for the low Reynolds number flow, and under 1.5% relative error for the high Reynolds number flow. Furthermore, the high Reynolds number flow was compared at the $x = 7$ and $x = 15$ cross-sections to those reported by Gartling [14]. The fine grid results, for the x-velocity and pressure fields, deviated from the reported results by no more than 2.0%. With this computed "exact solution" it was possible to determine the exact error $\vec{\mathcal{E}}$ with which to compare to the error estimates.

As was done with the advection-diffusion equation, approximate solutions were created using two different types of upwinding procedure: Standard upwind differencing (UDS), and Linear Profile Skew Upwind Differencing with Physical Advection Correction (LPS+PAC). Error estimation was performed on the second grid in the set (120 X 36).

The error field from a Navier-Stokes equation solution, as mentioned previously, is a three component vector. For this work, the x-component of velocity was chosen for plotting purposes. This does not seem to be a limiting choice, since it was witnessed that the level of correspondence between the exact and estimated error was the same for all three components of the error vector.

4.3.1 Low Reynolds Number Flow

The streamlines of the exact solution for this problem are shown in figure 4.2. It is obviously a low Reynolds number flow, containing but one small recirculation zone behind the step.

Using the procedure outlined in §4.2, a full linearization error estimate is computed for the approximate solutions from the UDS and the LPS+PAC numerical schemes. Figure 4.3 shows the comparison between the exact error in the x-component of velocity and the estimated error. Figure 4.4 is the same plot for the LPS+PAC approximate solution.

The degree of correspondence between the exact and estimated error is less than it was for the advection-diffusion equation. For the UDS approximate solution, the main error field features are clearly duplicated, but the error magnitude is an over-estimate by about a factor of two (a similar quality of error estimate is demonstrated for the y-velocity and pressure fields). Nevertheless, the error estimation yields a quantitatively useful measure of the solution quality.

A dramatic degradation of estimator performance is witnessed for the LPS+PAC approximate solution. For the first time it is witnessed that the fundamental characteristics of the error field (locations of maximum and minimum error, zero error contours, etc.) are not properly predicted. The magnitude of the error estimate is similar to the actual error, but it is easy to dismiss this as coincidence. So, what happened ?

There are only two steps to the error estimation process: residual estimation and solving the linearized error equation. Lacking a unique continuous representation of $[u, v, p]$ for an

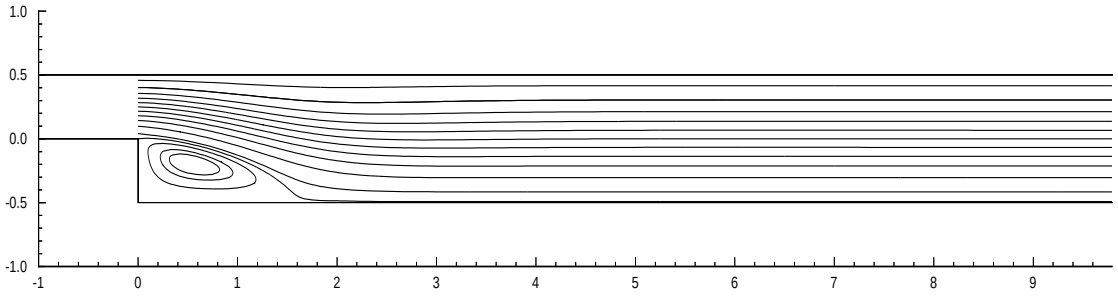


Figure 4.2: Streamlines: Low Reynolds Number Flow

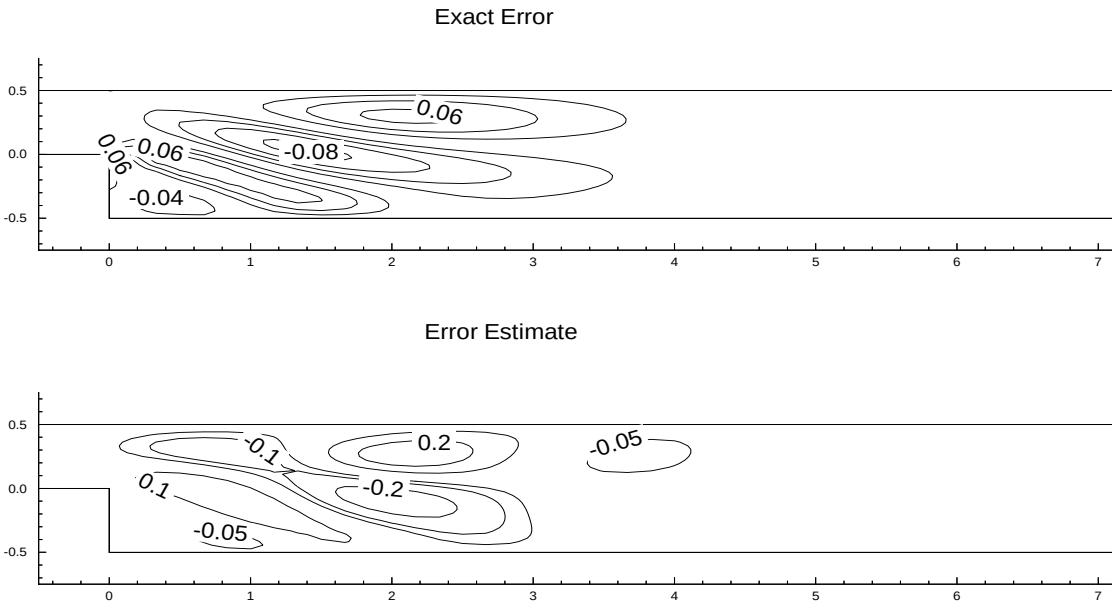


Figure 4.3: Re=100: X-Velocity Error: UDS : Full Linearization Error Equation

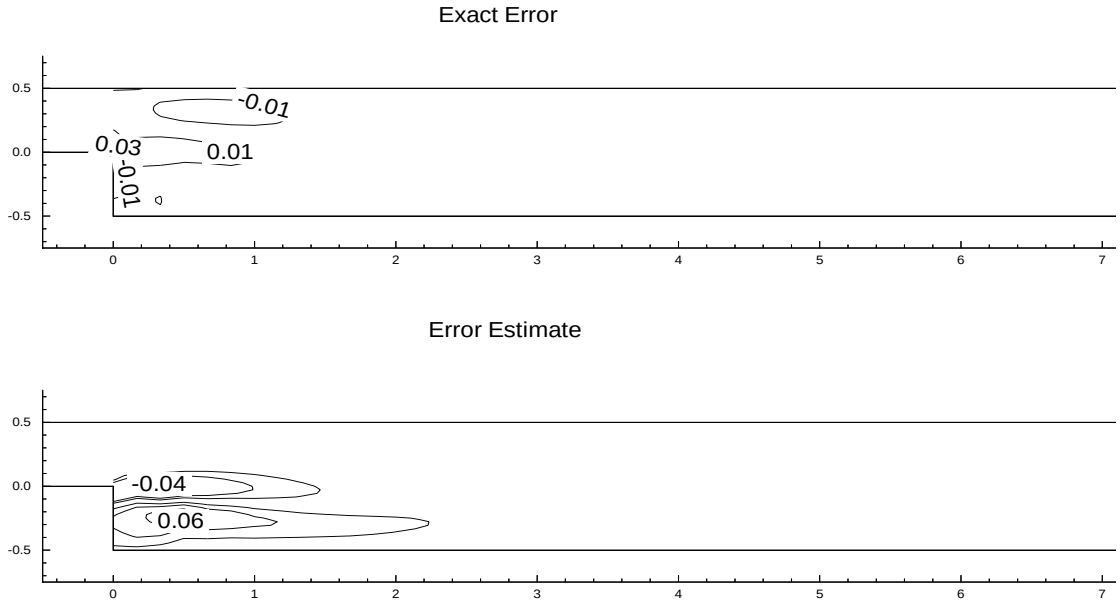


Figure 4.4: Re=100: X-Velocity Error: LPS+PAC : Full Linearization

approximate LPS+PAC solution, it is not possible to determine the exact operator residual for comparison. Nevertheless, it is suspected that the residual estimation is the problematic step. The same linearization has been used for both the UDS and LPS+PAC error estimation. If the fully linearized error equation poorly characterised the non-linear error equation at Re=100, then we would expect the UDS error estimation to also perform poorly, but this is not the case. Furthermore, the difference between a UDS discretization of the error equation, and an LPS+PAC error equation should not be large enough to cause such a dramatic difference in the error estimation, which leaves us to suspect the residual estimation step.

To explore this hypothesis, further numerical experiments were performed. One can take the residual vector field generated from the LPS+PAC solution, and instead of passing them into an error equation linearized about the LPS+PAC approximate solution, one passes them into the error equation used for the UDS error estimation (UDS discretized error equation linearized about the UDS approximate solution). If the error estimation process is sensitive to the linearization, or the method of solving the error equation (UDS or LPS+PAC discretization), then a significantly different error field should be predicted. If instead the error equation is insensitive to such changes, then a similar error field to the LPS+PAC error equation should result.

The results of such a computation are shown in figure 4.5. The striking similarity between the two contour plots supports the hypothesis that the residual estimation process is the underlying difficulty.

The case of the semi-linearized error equation was also examined for the low Reynolds

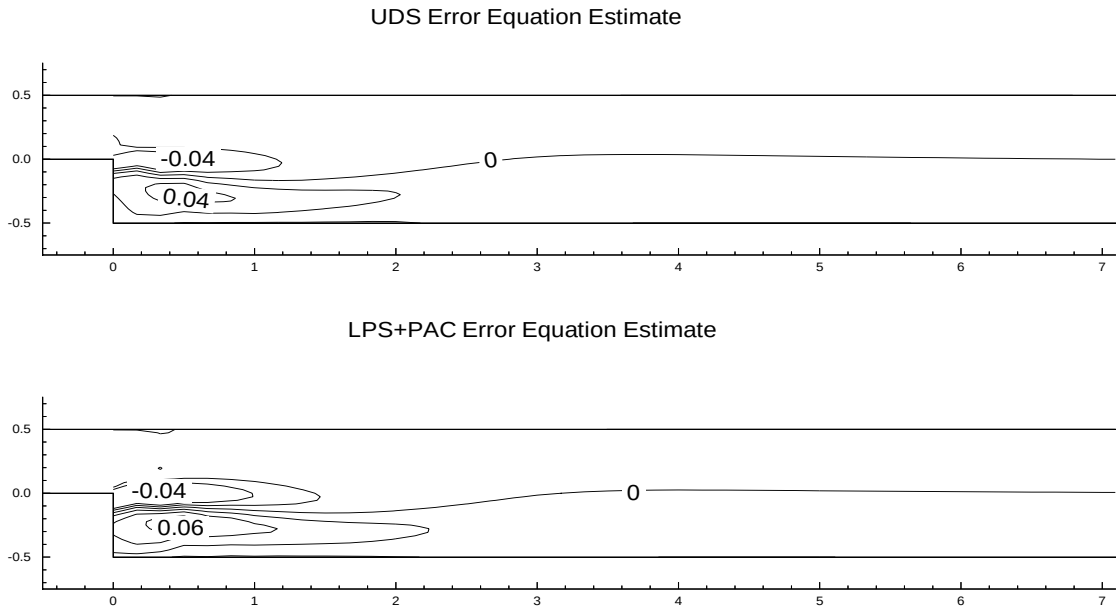


Figure 4.5: Re=100: X-Velocity Error: LPS+PAC : LPS+PAC Error Equation vs. UDS Error Equation: Full Linearization

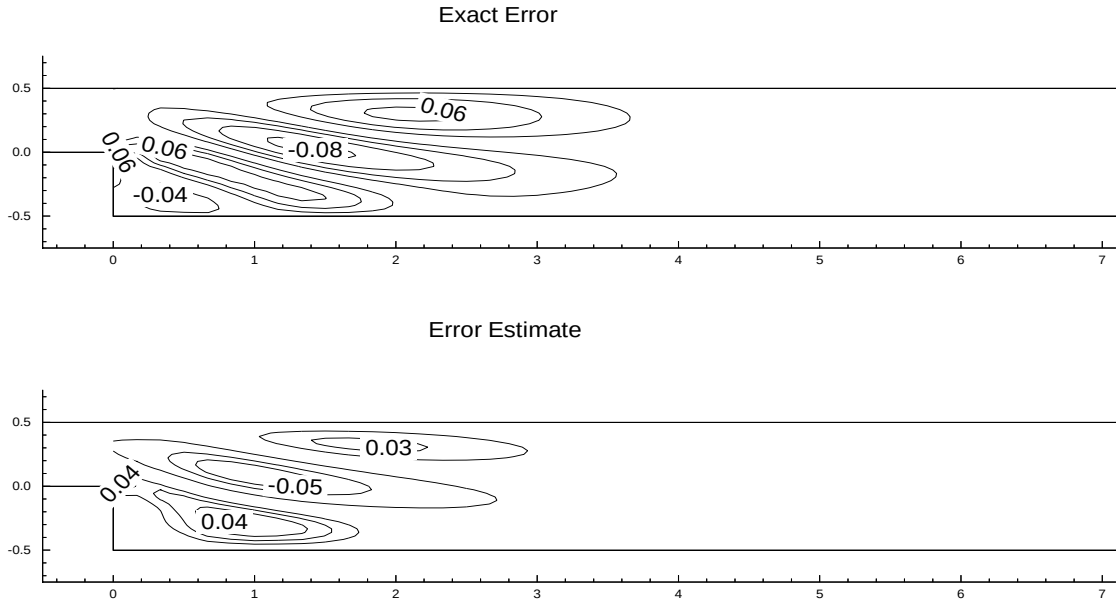


Figure 4.6: Re=100: X-Velocity Error: UDS : Semi-Linearization

number flow. Similar poor results were witnessed for the LPS+PAC error estimation, casting further doubt on the quality of the residual estimation. The UDS error estimation, however, had interesting results as shown in figure 4.6

Although the semi-linearization could be considered a cruder linearization of the non-linear error equation, it seems to produce as good, if not better, error estimation than that produced by the the full linearization, given the same residual field. Why this should be so is not clear at present. What *is* obvious is the large difference between the full and semi-linearization error estimates (figures 4.3 and 4.6).

4.3.2 High Reynolds Number Flow

The High Reynolds flow definition is identical to the Low Reynolds number flow except that μ is decreased by a factor of 8. This has the effect of raising the Reynolds number to 800. The streamlines of the steady-state solution are shown in figure 4.7

The computational experiments did not go as smoothly for the high Reynolds number flow, compared to the low Reynolds number flow. The full linearization error equation proved difficult to solve using a variety of solver techniques (high level factorization preconditioning, over and under relaxation, etc.) It is not clear at this time whether this difficulty is the result of the degree of instability in the governing equations or induced by poor choice of discretization. It has been conjectured that the steady incompressible 2D flow over a backward facing step is unstable at Re=800 (Gresho, Gartling, Cliff, Garret, Spence, Winters

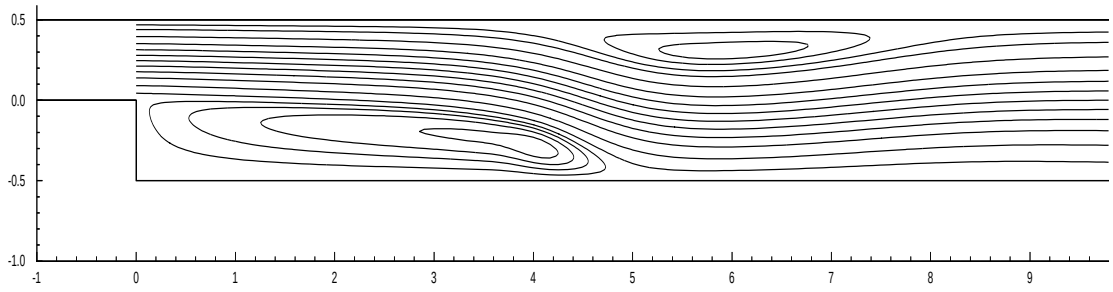


Figure 4.7: Streamlines: High Reynolds Number Flow

and Goodrich [15]). Indeed, the Re=800 flow required an order of magnitude more iterations to solve properly compared to the Re=100 flow; another indicator of a near critical flow field. The increase in iterations required by preconditioned conjugate gradient type methods when solving dynamically unstable linearized equations has been described by Simpson [34]. This has important implications for the process of error estimation, since one would expect the global error to be poorly determined by the residual field for unstable flows.

In contrast to the full linearization, solutions to the semi-linearized error equation could be obtained without difficulty. Results for the semi-linearized error equation are shown in figures 4.8 and 4.9.

As in the low Reynolds number flow, the LPS+PAC error estimation bears little resemblance to the actual error. This is consistent with our earlier hypothesis of an inadequate residual estimate.

The UDS error estimation again captures the significant features of the error field. A further degradation of the estimate quality is experienced; now underestimating the magnitude of the error by a factor of roughly three. It is not known at this time what factors cause a decline in performance with rising Reynolds number. This same effect was witnessed to a lesser degree with the 2D advection-diffusion equation results (in moving from low to high advection velocities). The 1D analytical results showed that the residual estimate effectivity does degrade with increasing cell Peclet number. These are possibly all analogous phenomenon.

4.4 Summary

The Global Error Equation Method was extended to encompass the Navier-Stokes equations. This involved addressing the vector nature of the dependent variables, and drawing the

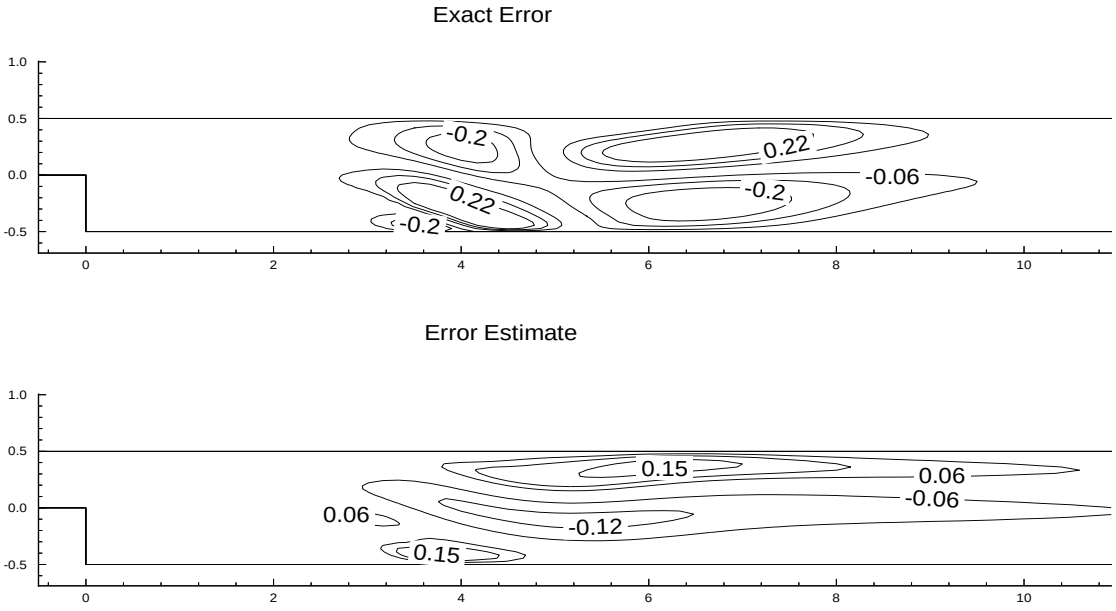


Figure 4.8: Re=800: X-Velocity Error: UDS : Semi-Linearization

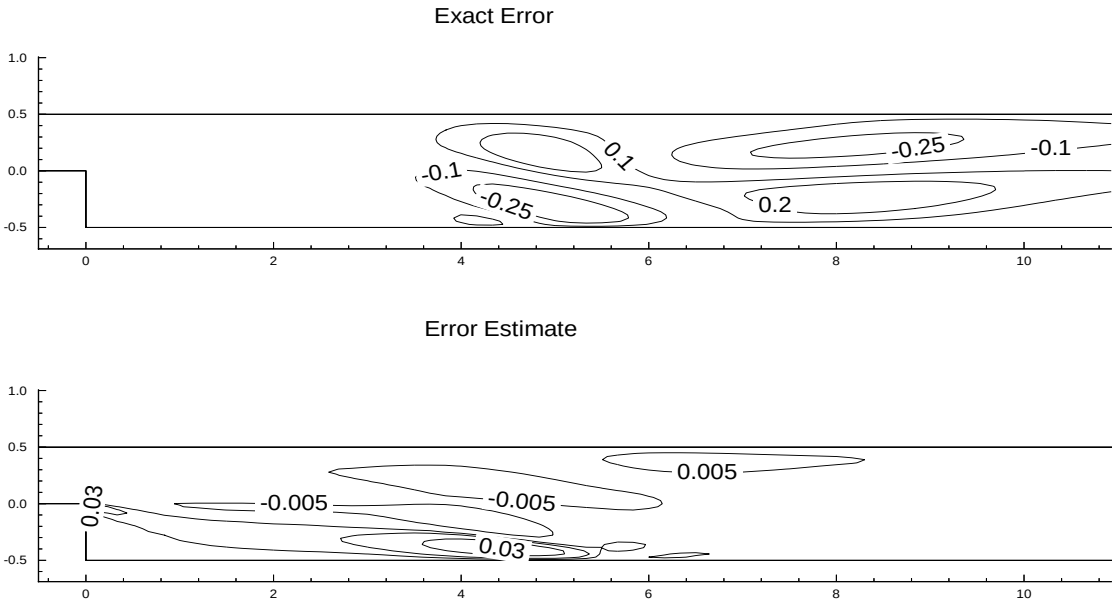


Figure 4.9: Re=800: X-Velocity Error: LPS+PAC : Semi-Linearization

distinction between the full non-linear error equation, and its linearizations.

For this thesis, two types of linearizations were presented and subjected to numerical experimentation: Full and Semi-linearization. We experienced difficulty solving the full linearization form of the error equation for the Re=800 flow.

For the UDS approximate solutions, a successful error estimation was demonstrated for both the low and high Reynolds number flows. The effectivity of the error estimation is not as good as was witnessed for the linear advection-diffusion equation problems. This might be as a result of the extra approximation introduced in linearizing the error equation (a step not required for a linear PDE). In contrast to the scalar advection-diffusion equation, the central difference residual estimate does not seem to reliably produce an effective residual estimate for LPS+PAC approximate solutions of Navier-Stokes equations.

Chapter 5

Closure

5.1 Summary

This thesis presents what we refer to as the Global Error Equation Method of *a posteriori* error estimation for finite volume simulations of fluid transport. The process involves two main steps:

- Estimating the Exact Operator Residual
- Solving the Global Error Equation

The first step has ties to the Element Residual Method although, for this work, residual estimation has been expressed in terms of the familiar centred difference operator. The Global Error Equation is unique to this thesis.

The investigation was made on several fronts. An analytical analysis of the procedure was performed for the 1D linear advection-diffusion equation using an upwind differencing approximate solution. For the 2D linear advection-diffusion equation a suite of numerical problems were posed, under the general headings of *point source problems* and *boundary layer problems*. Each, in turn, were solved with two different finite volume approximations: upwind differencing, and, linear profile skew upwinding with advection correction. For each approximate solution an error estimation was performed and compared against the exact error generated from an analytical solution.

Finally, the 2D Navier-Stokes equations were considered. The error estimation process was re-derived to account for the vector nature of the solutions, and to linearize the nonlinear error equations. The process was experimentally studied by analyzing the error estimator performance for a low and high Reynolds number backward facing step problem.

5.2 Conclusions

For the linear advection-diffusion equation, the error estimation process outlined demonstrated excellent performance. It was also demonstrated that the residual estimation process need not utilize a higher truncation order discretization for the residual estimation process in the linear equation case.

The work with the 2D Navier-Stokes equations demonstrated three important elements involved with the extension of this work:

1. Under certain circumstances, the error estimation process can be successfully extended to a coupled nonlinear PDE system. In this case, the approximate solution is an upwind difference approximation of the Navier-Stokes equations.

2. The ability of a central difference approximation to generate a useful residual estimate for LPS+PAC approximate solutions seems to be compromised for the Navier-Stokes equations.

3. There is a large variation in error estimator performance depending on the approach to the linearization of the nonlinear error equations. At the high Reynolds number flow (Re=800), it was not possible to achieve an error estimate using the full linearization error equation. The semi-linearization did not experience solver difficulties. For Re=100, there is a significant difference in the estimated error depending on whether one utilizes full or semi-linearization.

5.3 Recommendations

The work presented establishes a framework with which future research can proceed. The linear equation studies seem to be conclusive and further work in this direction is not deemed necessary. To bring this technology into the purview of serious engineering CFD, two issues need to be addressed first:

Effective residual estimation for more advanced discretizations must be created and validated. There does not exist any foreseeable barriers to achieving this goal. The residual estimation step is always an $O(N)$ calculation, so it is believed that only marginal additional computational effort needs to be expended to guarantee an effective residual estimate. What is lacking currently is a coherent framework in the finite volume method for consistently generating higher accuracy discretizations.

The behavior of the linearized Navier-Stokes error equation requires additional study. Specifically, it needs to be determined conclusively if the solver difficulty for the fully linearized error equation stems from the governing equations, or the particular discretization implemented in this thesis (ie. a lumped mass discretization was utilized for the zero-order terms in the fully linearized error equation). Furthermore, it would be desirable to understand the limits of a linearized error equation.

If these issues two issues can be resolved, then the error estimation methodology is not inherently limited to the equations studied. The conceptual extension to 3D Navier-Stokes equations with auxiliary coupled equations (compressibility, conjugate heat transfer, turbulence models, etc.) is quite straightforward. Furthermore, the use of orthogonal grids in this study was strictly for convenience; there is no obstacle that the researchers are aware of to prevent one from using non-orthogonal grids.

Bibliography

[1] M. Ainsworth, J.T. Oden, "A Procedure for *a posteriori* Error Estimation for the *h-p* Finite Element Methods", Comp. Meth. in Appl. Mech. and Engrg. Vol 101, pp. 73-96. 1992.

[2] M. Ainsworth, J.T. Oden, "A Unified Approach to *a posteriori* Error Estimation Using Element Residual Methods", Numer. Math. 63, pp. 23-50, 1993

[3] M. Ainsworth, J.Z. Zhu, A.W. Craig, O.C. Zienkiewicz, "Analysis of the Zienkiewicz-Zhu *a posteriori* Error Estimator in the Finite Element Method", Int. Journ. for Numer. Meth. in Engrg. Vol.28 pp. 2161-2174, 1989.

[4] I. Babuška, W.C. Rheinboldt, "*A Posteriori* Error Estimates for the Finite Element Method", Int. Journ. for Numer. Meth. in Engrg. Vol.12 pp. 1597-1615, 1978.

[5] I. Babuška, R. Rodriguez, "The Problem of the Selection of an *A Posteriori* Error Indicator Based on Smoothening Techniques", Int. Journ. for Numer. Meth. in Engrg. Vol.36 pp. 539-567, 1993

[6] R.E. Bank, A. Weiser, "Some *a posteriori* error estimates for elliptic partial Differential Equations", Math. of Computation, 44, pp. 283-301, 1985

[7] R.E. Bank, "Analysis of a Local *a posteriori* Error Estimator for Elliptic Equations", *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*", John Wiley & Sons, 1986.

[8] D. Bradley, M. Missaghi, S.B. Chin, "A Taylor-Series Approach to Numerical Accuracy and a Third-Order Scheme for Strongly Convective Flows" Comp. Meth. in App. Mech. and Engrg. Vol.69 133-151, 1988.

[9] R.B. Broberg, "Numerical Simulation of Rotor/Stator Flow", MASc Thesis, University of Waterloo, 1994.

[10] S.C. Caruso, J.H. Ferziger, J. Oliger, "Adaptive Grid Techniques for Elliptic Fluid-Flow Problems", Report No. TF-23, Department of Mechanical Engineering, Stanford University, Stanford, California. 1985

[11] R.E. Ewing, "A Posteriori Error Estimation", Comp. Meth. in App. Mech. and Engrg. Vol.82 59-72. 1990.

[12] J.H. Ferziger, "Estimation and Reduction of Numerical Error", *Forum on Methods of Estimating Uncertainty Limits in Fluid Flow Computations. ASME Winter Annual Meeting*, 1989.

[13] J.H. Ferziger, *Numerical Methods for Engineering Applications*, John Wiley & Sons, New York, NY, 1981.

[14] D.K. Gartling, "A Test Problem For Outflow Boundary Conditions - Flow Over a Backward Facing Step", Int. Journ. Numer. Meth. Fluids, Vol 11, pp. 953-967, 1990

[15] P.M. Gresho, D.K. Gartling, K.A. Cliff, T.J. Garret, A. Spence, K.H. Winters, J.W. Goodrich, J.R. Torczynski "Is the Steady Viscous Incompressible 2D Flow Over a Backward Facing Step at Re=800 Stable ?" Int. Journ. Numer. Meth. in Fluids, Vol 17, pp. 501-541, 1993.

[16] J-F. Hétu, D. Pelletier, "Fast, Adaptive Finite Element Scheme for Viscous Incompressible Flows". AIAA Journal, Vol. 30 No. 11, Nov. 1992.

[17] J-F. Hétu, D. Pelletier, "An Adaptive Finite Element Methodology for Incompressible Viscous Flows". *Advances in Finite Element Analysis in Fluid Dynamics II: ASME Winter Annual Meeting. November 8-13, 1992*

[18] J-F. Hétu, D. Pelletier, "An Adaptive Finite Element Method for Thermal Hydraulics Problems" Presented at *AIAA 31st Aerospace Sciences Meeting and Exhibition: January 11 - 14, 1993 in Reno, NV, USA* AIAA 93-0920, 1993

[19] C. Hirsch, "*Numerical Computation of Internal and External Flows, Vol. 1*", John Wiley & Sons, San Mateo, California, 1988.

[20] D.W. Kelly, "The Self Equilibration of Residuals and Complementary *a posteriori* Error Estimates In the Finite Element Method", Int. Journ. Numer. Meth. Engrg. 20, pp. 1491-1506, 1984

[21] D.W. Kelly, "The Self-equilibration of Residuals and 'Upper Bound' Error Estimates for the Finite Element Method", *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, John Wiley & Sons, 1986

[22] D.W. Kelly, S.R. Gago, O.C. Zienkiewicz, I. Babuška, "*A Posteriori* Error Analysis and Adaptive Processes in the Finite Element Method: Part I - Error Analysis", Int. Journ. for Numer. Meth. in Engrg. Vol 19, pp. 1593-1619. 1983

[23] D.W. Kelly, R.J. Mills, J.A. Reizes, A.D. Miller, "*A Posteriori* Estimates of the Solution Error Caused by Discretization in the Finite Element, Finite Difference, and Boundary Element Methods", Int. Journ. Numer. Meth. in Engrg. Vol.24, 1921-1939, 1987

[24] R.Kessler, M.Peric, G. Scheuerer, "Solution Error Estimation in the Numerical Predictions of Turbulent Recirculating Flows", Lehstruhl für Strömungsmechanik, D-8520 Erlangen, F.R.Germany

[25] A.W. Naylor, G.R. Sell, "*Linear Operator Theory for Engineering and Science*" Springer-Verlag, 1982

[26] J.T. Oden, W. Weihan, M. Ainsworth, "An *a posteriori* Error Estimate for Finite Element Approximations of the Navier-Stokes Equations", Comp. Meth. Appl. Mech. and Engrg. vol. 111, pp. 185-202, 1994

[27] J.T. Oden, L. Demkowicz, T. Strouboulis, P. Devloo, "Adaptive Methods for Problems in Solid and Fluid Mechanics", *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*", John Wiley & Sons, 1986.

[28] S.V.Patankar, "*Numerical Heat Transfer*", Hemisphere, Washington D.C., 1980.

[29] G.D. Raithby, "Skew Upstream Differencing Schemes for Problems Involving Fluid Flow", Comp. Meth. Appl. Mech. and Engrg. vol. 9 pp. 153-164, 1976.

[30] C.M. Rhie, W.L.Chow, "A Numerical Study of the Turbulent Flow Past an Isolated Airfoil with Trailing Edge Separation", AIAA paper 82-0998, 1982.

[31] P.J. Roache, "*Computational Fluid Dynamics*", Hermosa Publishers, Albuquerque, N.M. 1982.

[32] P.J. Roache, "A Method for Uniform Reporting of Grid Convergence Studies", *Quantification of Uncertainty in Computational Fluid Dynamics*, ASME 1993

[33] G.E. Schneider, M.J. Raw, "Control Volume Finite Element Method for Heat Transfer and Fluid Flow Using Colocated Variables - 1. Computational Procedure", Numer. Heat Trans. Vol 11, pp. 363-390, 1987.

[34] R.B. Simpson, "Testing the Effects of Assymetry and Instability on the Preconditioned Iterations of the Conjugate Gradient Method", Journ. of Numer. Anal., 14, pp.1-25, 1993

[35] I. Stakgold, "*Boundary Value Problems of Mathematical Physics*", The Macmillan Company, New York, 1968.

[36] T. Strouboulis, K.A. Haque, "Recent Experiences with Error Estimation and Adaptivity, Part I: Review of Error Estimators for Scalar Elliptic Problems" Comp. Meth. in App. Mech. and Engrg. Vol.97 399-436, 1992.

[37] T. Strouboulis, J.T. Oden "A Posteriori Error Estimation of Finite Element Approximations in Fluid Mechanics", Comp. Meth. in App. Mech. and Engrg. Vol.78 201-242, 1990.

[38] G.D. Stubley, "A New Discrete Method for Convection-Dominated Flows Based on a Clear Understanding of Solution Errors", Ph.D. Thesis, University of Waterloo, 1981.

[39] D.G. de Vahl, "Natural Convection of Air in a Square Cavity: A Bench Mark Solution", Int. Journ. Numer. Meth. in Fluids, Vol. 3, No.3, pp.249-264, 1983.

[40] R. Verfurth, "A Posteriori Error Estimates for the Stokes Equations", Numer. Math. Vol. 55, pp. 309-325, 1989.

[41] WATSIT - *Waterloo Sparse, Iterative Solvers Users Guide* Scientific Computation Group, University of Waterloo, Ontario, Canada, 1993.

[42] J.D. Yang, D.W. Kelly, J.D. Isle, "*A posteriori* Pointwise Upper Bound Error Estimates in the Finite Element Method", Int. Journ. for Numer. Meth. in Engrg. Vol 36, pp. 1279-1298. 1993.

[43] O.C. Zienkiewicz, J. Z. Zhu, "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis", Int. Journ. for Numer. Meth. in Engrg. Vol.24 pp. 337-357, 1987.

[44] O.C. Zienkiewicz, J. Z. Zhu, "The Superconvergent Patch Recovery (SPR) and Adaptive Finite Element Refinement", Comp. Meth. in Appl. Mech. and Engrg. Vol 101, pp. 207-224, 1992

[45] O.C. Zienkiewicz, R.L. Taylor, "*The Finite Element Method. Fourth Edition, Vol.1 & 2*", McGraw-Hill Book Company, Maidenhead England. 1989 (Vol.1), 1991 (Vol.2).

Appendix A

Related Research

In this appendix, a more thorough description is given for other contemporary error estimation techniques than could be given in §1.5. This is to provide a fully referenced resource for the interested reader.

A.1 Element Residual Methods

The Element Residual Methods are given in a Finite Element context. For the explanation given here, the mathematical rigor will be suppressed for the benefit of understanding (The referenced articles have more than enough formalism for the interested reader). A familiarity with the Finite Element Method is required in this section [1].

First some preliminaries need to be presented. Consider a standard elliptic PDE with a Dirichlet and a Neumann boundary condition:

$$L(\Phi) = -\nabla \cdot (a \nabla \Phi) = f \text{ on } \Omega$$

$$\Phi = 0 \text{ on } \Gamma_D$$

$$\frac{\partial \Phi}{\partial n} = g \text{ on } \Gamma_N$$

The first step for a finite element calculation would be to partition the domain Ω into a set of finite elements. On these finite elements are defined shape functions. The set of all these shape functions will be called $V(\Omega)$.
Now, finding the approximate solution ϕ, which uses the shape functions as a basis, $V(\Omega)$, can be expressed as:

Find $\phi \in V(\Omega)$ *such that:*

$$\int_\Omega (a \nabla \phi \cdot \nabla v) dx = \int_\Omega (fv) dx + \int_{\Gamma_N} gv \, ds \tag{A.1}$$

for all $v \in V(\Omega)$
For the finite element method, the space $V(\Omega)$ is composed of shape functions restricted to the elements (for example, piecewise linear shape functions). Hence, we can define an

[1] Zienkiewicz has written a thorough and widely published reference [45]

element-wise integral and represent the integrals as summations over all the individual elements k (we can also introduce the *inner product* form $(\cdot,\cdot)$) :

$$a(\phi, v)_k = \int_k (a\nabla\phi \cdot \nabla v)dx \tag{A.2}$$

$$l(v)_k = \int_k (fv)dx + \int_{\Gamma_N \cap k} gv\, ds \tag{A.3}$$

$$a(\phi, v) = \sum_k a(\phi, v)_k \tag{A.4}$$

$$l(v) = \sum_k l(v)_k \tag{A.5}$$

Hence, the FEM may be stated more succinctly as:
Find $\phi \in V(\Omega)$ *such that:*

$$(a, v) = l(v) \tag{A.6}$$

for all $v \in V(\Omega)$

The first step in an Element Residual Method is the estimation of the residual function over the element. The residual of the approximate solution $\phi \in V(\Omega)$ is the function r such that on each element

$$a(\phi, w)_k - l(w)_k = (r, w)_k \tag{A.7}$$

for all admissible w *functions.*

Error estimation now involves selecting an error representation space V_E (ensuring that V_E is not a subspace of $V(\Omega)$), and finding the function $e \in V_E$ such that

$$a(e, w) = (r, w) + \text{boundary conditions} \tag{A.8}$$

for all $w \in V_E$

Before equation A.8 is broken down and analyzed, let's examine it as a whole. What we have is an *error equation*, where the error is expressed as the satisfying the original partial differential formula, with the *residual* acting in the role of the source term from the original equation.

To effect a local computation for (A.8) one can select V_E to be the set of *bubble functions*. Bubble functions are like the familiar shape functions, in that they are defined over an element, however, they are defined to be zero at the element nodes. Thus, a bubble function is completely localized to a particular element. Higher order bubble functions were originally proposed for hierarchical finite element methods [45].

The bubble functions are chosen to be order higher than was used to solve the original problem (V). So if V was first order, $p = 1$ (piecewise linear interpolation), then the bubble functions would typically be second order, $p = 2$ (piecewise quadratic).

The use of bubble functions, howver, results in a singular system on each element. When the element shares an edge with the boundary of the domain Ω, then the closure on $\Gamma \cap k$ is straight forward. Where the boundary condition of the original equation was a Dirichlet condition, a Dirichlet condition can be imposed, using the known value of error. Where

the boundary condition was a Neumann condition, a Neumann error equation boundary condition can be imposed.

For interelment boundaries, a number of approaches have been proposed and used. The earliest proposals used a homogeneous Dirichlet boundary condition (Babuŝka and Rheinboldt [4]). These techniques were superseded by later work from Bank and Weiser [6] and Bank [7] which proposed setting a Neumann boundary conditions, in which, at the element boundary:

$$\frac{\partial e}{\partial n} = \alpha \left[\left| \frac{\partial \phi}{\partial n} \right| \right]_j$$

where the right-hand side represents the jump in the first derivative across the element boundary. The weighting parameters α must be solved in an auxiliary calculation called *residual flux equilibration* (The reader is referenced to [6] for a description of this process)

The local Neumann problem formulation of the Element Residual Method has further been refined by incorporating a splitting approach, first proposed by Kelly [20, 21], and extended by Ainsworth and Oden [2], where, for a given element edge bordered by elements l and r (left and right):

$$\left(\frac{\partial e}{\partial n} \right)_l = \alpha_{lr} \left[\left| \frac{\partial \phi}{\partial n} \right| \right]_{j,lr} + (1 - \alpha_{rl}) \left(\frac{\partial \phi}{\partial n} \right)_{average}$$

$$\left(\frac{\partial e}{\partial n} \right)_r = \alpha_{rl} \left[\left| \frac{\partial \phi}{\partial n} \right| \right]_{j,rl} + (1 - \alpha_{lr}) \left(\frac{\partial \phi}{\partial n} \right)_{average}$$

The derivation of this form is quite involved, but essentially the Neumann condition is created from a weighted average of the jumps in solution derivative, and the solution derivatives. The residual flux equilibration has to be modified to take this into account.

In moving to Navier-Stokes equations (Oden, Weihan and Ainsworth [26]), the only change is the inclusion of the additional advection term in the residual calculation. The local error equation remains a purely elliptic computation.

A.2 Flux Projection Methods

This class of error estimators has its origin in a very humble observation from elasticity problems. For the displacement approximation for linear elastic problems we generally assume a continuous displacement field, resulting in discontinuous interelement stresses $\hat{\sigma}$. An obviously more accurate solution would have a continuous stress distribution σ^*.

Out of this observation comes the idea of Flux Projection Methods. The name derives from the process by which the smoother stress field is created by proposing a continuous distribution of strain energy flux (hence these estimators are also said to be based on *smoothening techniques*). Some commercial post-processors codes will smooth the stress field by taking the average value of stress across the interface but this can be misleading. A better approach would be to perform what is called *variational recovery* or *projection* [45] which

involves solving another variational problem for the stress field where we use the same basis functions for the stress field as for the the underlying displacement field. The problem which now must be solved turns out to equivalent to solving the least squares fit for stress, or the minimization of the functional

$$\int_{\Omega} (\sigma^* - \hat{\sigma})^2.$$

Now an error estimate for this solution would be [43, 3, 45]

$$e_\sigma \approx \sigma^* - \hat{\sigma}.$$

This is commonly referred to as the *Zienkiewicz-Zhu* error estimator, after their classic paper first detailing the approach [43].

Consider a simple one-dimensional example, namely, a simple elliptic PDE as done by Babuška and Rodriguez [5]:

$$-\frac{\partial^2 \Phi}{\partial x^2} = f \text{ on } I := (0,1),$$

$$\left.\frac{\partial \Phi}{\partial x}\right|_{x=0} = 0,$$

$$\Phi|_{x=1} = \Phi_1.$$

For this problem Φ would represent a displacement field for a linear elastic problem, while $\frac{\partial \Phi}{\partial x}$ would represent stress.

Imagine the forcing function f was specified such that the FEM solution for ϕ is as shown in figure A.1. Also plotted is $\frac{\partial \phi}{\partial x}$ of the approximate solution, and the appearance of a projected stress distribution would look like, $\frac{\partial \phi}{\partial x}^*$.

The error estimator is invariably reported in the *energy norm*. For the simple linear problem presented, the energy norm of the error on a given finite element is simply

$$\left[\int_{x_i}^{x_{i+1}} \left(\frac{\partial(\Phi - \phi)}{\partial x}\right)^2 dx\right]^{\frac{1}{2}},$$

which can now be approximated as

$$\left[\int_{x_i}^{x_{i+1}} \left(\frac{\partial \phi}{\partial x}^* - \frac{\partial \phi}{\partial x}\right)^2 dx\right]^{\frac{1}{2}}.$$

It can be seen that this concept applies to any purely elliptic phenomenon. For the heat equation, one would perform a smoothening of the heat conduction field; for scalar diffusion, the diffusive flux would be smoothed. Hence the terminology *flux* projection.

The nice thing about this estimator is its simplicity of implementation. This error estimator has been used successfully by various researchers to drive adaptive schemes for linear and nonlinear elasticity problems.

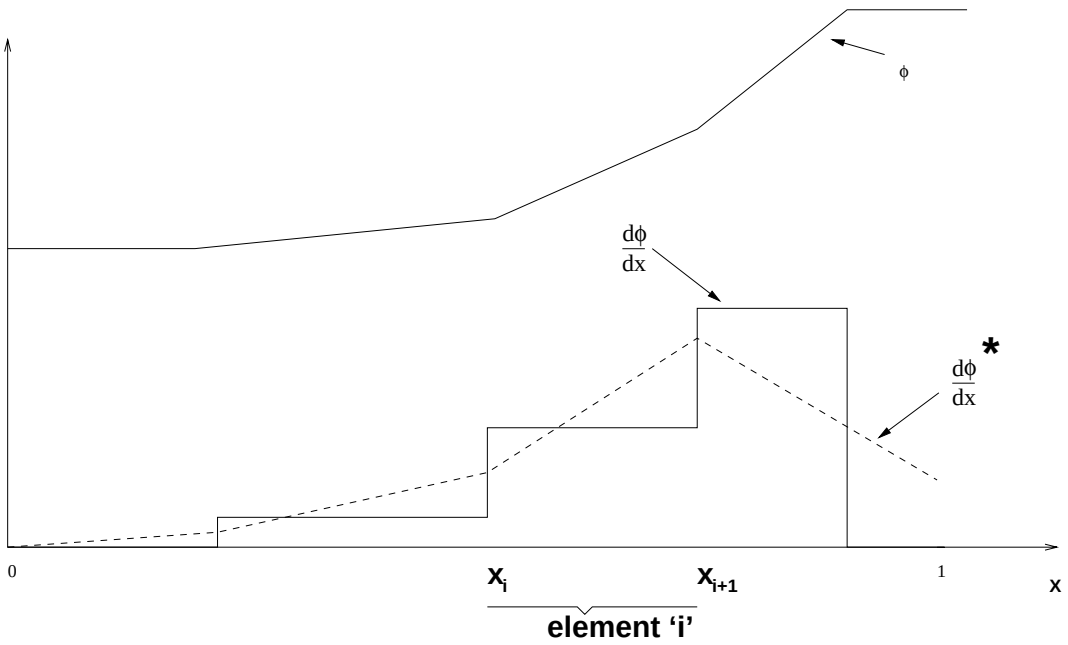


Figure A.1: One Dimensional Flux Projection

The extension of this approach to fluid flow problems is based on the parallel found between plane strain equations and creeping, or Stokes flow equations (ref. [45]). The fluid analogy for σ for fluid transport would be fluid normal and shear stress. This fact can be used to derive a valid error estimator for creeping flows based on a projection of the flow stress components. This approach has been applied to the Navier-Stokes equations by Hétu and Pelletier [16, 17] for error estimation to guide their own grid adaption scheme. The estimator is only theoretically valid for creeping flow; therefore, the convergence behavior and mathematical rigor are not translatable. Nonetheless, the estimator has the required sensitivity to high stress/strain regions of flow, and hence serves quite well for detecting insufficient refinement in boundary layers and at stagnation points.

The projection estimator, as described, is unfortunately insensitive to pressure gradient dominated flows and turbulence effects. Hence flow features like re-attachment points are ignored. Recent attempts have been made by Hétu and Pelletier [18] to correct this defect by incorporating projection corrections of other discontinuous fields: pressure, turbulent kinetic energy flux, heat flux, etc. into a hybrid local error norm. Despite the fact that there is no theoretical justification for such an approach, this seems to yield an estimator which is sensitive to most flow phenomenon found in incompressible flows. The lack of theoretical underpinning is not entirely surprising; it was two years after Zienkiewicz and Zhu proposed their projection error estimator based on physical intuition that it was proven convergent and consistent with the underlying variational principle by Ainsworth, Zhu, Craig and Zienkiewicz [3].

At the current time, the extension from Stokes flow to Navier-Stokes flow is, strictly speaking, unjustified. Although the estimator sensitivity has been experimentally shown to be qualitatively valid for driving grid adaption schemes. The error in the flux estimate does

not have a simple local correlation to the distributed error in the dependent variables. It is also unlikely that a least-square projection, despite being easy to calculate and numerically well behaved, is the correct variational recovery that should be employed. The least-square equivalence applies only to strictly elliptic partial differential problems. A more rigorous derivation of the correct variational recovery for the discontinuous quantities in the discrete Navier-Stokes equations would be worthwhile, but beyond the scope of this thesis.

A.3 Extrapolation Methods

Extrapolation was first used by Richardson in 1910. The basic premise, as described by Roache [32], assumes that the discrete solution ϕ, on a grid with characteristic dimension h, relates to the exact solution Φ, as

$$\Phi = \phi + F_1 h + F_2 h^2 + F_3 h^3 + \dots$$

for a first order discretization scheme, and

$$\Phi = \phi + G_2 h^2 + G_3 h^3 + \dots$$

for a second-order scheme.

The idea now is to derive values for these higher order error terms based on successive solutions for ϕ computed with different values of h.

For instance, suppose we have two successive computations from a first-order accurate scheme ϕ_1 and ϕ_2, with corresponding discrete spacing h_1 and h_2. These values relate as follows

$$\Phi = \phi_1 + F_1 h_1 + O(h^2) + \dots$$

$$\Phi = \phi_2 + F_1 h_2 + O(h^2) + \dots$$

We can then solve for F_1 from these two equations

$$F_1 = \frac{\phi_2 - \phi_1}{h_1 - h_2} + O(h^2)$$

An estimate of the error on the ϕ_2 solution; E_2, would be

$$\begin{aligned} E_2 &= \Phi - \phi_2 \\ &= F_1 h_2 + O(h^2) \\ &= \frac{\phi_2 - \phi_1}{\frac{h_1}{h_2} - 1} + O(h^2) \end{aligned} \tag{A.9}$$

Applying this correction to the original solution ϕ_2 should result in the classic second order accurate scheme from extrapolation. This is referred to as h^2 *extrapolation.*

The methodology of solving for progressively higher order error terms with multiple levels of refinement has been used successfully in many areas of numerical analysis. Ferziger

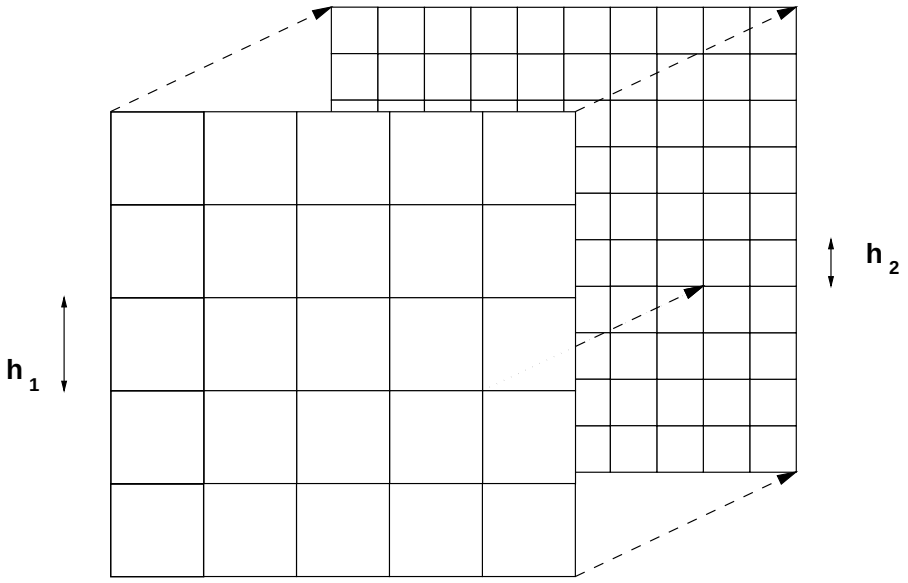


Figure A.2: Grid-Inside-Grid Arrangement

[13, 12], deVahl [39],and Kessler, Peric and Scheurer [24] Caruso, Ferziger and Oliger [10] Bradley, Missaghi and Chin [8] apply this principal to finite difference and finite volume computations of complex fluid flow.

The method can become quite general in its applicability. Functionals of the solution vector will also display this behavior. Therefore quantities such as lift coefficient, total head rise, and heat flux, can undergo the same type of error estimate. The principle laid out here is independent of the equations being solved for or the dimensionality of the the problem.

For a successful application of an extrapolation method, there is the requirement that every solution can be fully characterised by the same single parameter, h, the grid spacing. This single parameter family of approximate solutions is not a trivial matter to achieve in practice. Nor are there any means currently published to determine if one has been successful in creating such set of solutions. Furthermore, the discrete solutions need to be compared at the same location in the domain. If one wants to avoid the added uncertainty of interpolating from one grid to the next, the one needs to ensure that at least *some* of the grid points in grid set are coincident. A common method to ensure grid point coincidence, and enhance the chances of a single parameter solution set, is the *grid-inside-grid* arrangement. As shown in figure A.2.

In addition to grid nesting, there are other requirements for the successful application of Richardson extrapolation. The solution, for all levels of refinement the analyst intends to employ, must be in the *asymptotic range* (meaning that *any* degree of refinement,on any of the grids utilized, brings the approximate solution *proportionately* closer to the exact solution). Simply put, this is the requirement that the higher order terms of equation A.9 are negligible compared to the error.

An additional requirement is that the order of the numerical scheme needs to be known *a priori*. This can be difficult for modern CFD codes which utilize different numerical schemes depending on the local flow properties and types of boundary conditions employed.

The researcher may resort to cruder numerical approximations to guarantee a successful determination of the numerical order, as done by Kessler, Peric and Scheuerer [24].

The refinement process must be *homogeneous* and *isotropic* throughout the entire solution domain, in all dimensions (including `time` for transient problems). This is necessary because an inherent assumption in using extrapolation for a local value of the solution is that the error in the solution is distributed in a consistent manner from one grid to the next.

Appendix B

Analytic Solutions of 1D Finite Difference Equations

Consider the 1D steady homogeneous advection-diffusion equation:

$$\begin{aligned} -\Gamma\frac{\partial^2\Phi}{\partial x^2} + V\frac{\partial\Phi}{\partial x} = 0 \quad V,\Gamma > 0, \\ \Phi = \Phi_l \quad at\ x = 0, \\ \Phi = \Phi_r \quad at\ x = 1. \end{aligned} \tag{B.1}$$

Divide the domain $0 \leq x \leq 1$ into N partitions, each having size $h = 1/(N+1)$. The grid Peclet number will be defined as $a = Vh/\Gamma$.

If this problem is discretized using a standard upwind based finite volume method, then we will generate a system of algebraic equations for the approximate solution ϕ:

$$(2+a)\phi_{i+1} - \phi_{i+2} = (1+a)\Phi_l, \quad i = 0, \tag{B.2}$$

$$-(1+a)\phi_{i-1} + (2+a)\phi_i - \phi_{i+1} = 0, \quad i = 1...N-1, \tag{B.3}$$

$$-(1+a)\phi_{i-1} + (2+a)\phi_i = \Phi_r, \quad i = N, \tag{B.4}$$

where ϕ_i represents the solution in cell i .

We will try to find solutions of the form

$$\phi_i = Ap^i. \tag{B.5}$$

From equation (B.3) we see that

$$\begin{aligned} -p^2 + (2+a)p - (1+a) = 0, \\ \text{thus: } p = 1 \quad or \quad p = (1+a). \end{aligned} \tag{B.6}$$

Hence, solutions of (B.3) are of the form

$$\phi_i = A + B(1+a)^i. \tag{B.7}$$

From equation (B.2) we can determine:

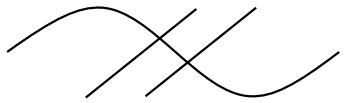
$$\begin{aligned} (2+a)(A+B(1+a)) - A - B(1+a)^2 &= (1+a)\Phi_l, \\ A+B &= \Phi_l, \end{aligned} \tag{B.8}$$

and from equation (B.4) we can determine

$$\begin{aligned} -(1+a)(A+B(1+a)^{N-2}) + (2+a)(A+B(1+a)^{N-1}) &= \Phi_r, \\ A+B(1+a)^N &= \Phi_r. \end{aligned} \tag{B.9}$$

Combining (B.7),(B.8), and (B.9);

$$\phi_i = \Phi_l + (\Phi_r - \Phi_l)\frac{(1+a)^i - 1}{(1+a)^N - 1}. \tag{B.10}$$



Consider the 1D inhomogenous error equation

$$\begin{aligned} -\Gamma\frac{\partial^2 \mathcal{E}}{\partial x^2} + V\frac{\partial \mathcal{E}}{\partial x} &= -R \quad V, \Gamma > 0, \\ \mathcal{E} &= 0 \quad at \;\; x = 0, \\ \mathcal{E} &= 0 \quad at \;\; x = 1. \end{aligned} \tag{B.11}$$

As with the original problem (B.2, an upwind difference discretization will result in a system of algebraic equations:

$$E_0 = 0 \qquad i = 0 \tag{B.12}$$

$$-(1+a)E_{i-1} + (2+a)E_i - E_{i+1} = a^3\frac{(\Phi_l - \Phi_r)(1+a)^{i-1}}{2(1+a)^N - 1} \qquad i = 1...N-1 \tag{B.13}$$

$$E_N = 0 \qquad i = N \tag{B.14}$$

Assume E_i has solutions of the form:

$$E_i = \left[A + B(1+a)^i + C(i)\right] a^3 \frac{(\Phi_l - \Phi_r)}{2(1+a)^N - 1}. \tag{B.15}$$

Substitute (B.15) into (B.13) and one is left with:

$$-(1+a)C(i-1)+(2+a)C(i)-C(i+1)=(1+a)^{i-1}$$

which can be written in a recursive form as

$$C(i+1)=(2+a)C(i)-(1+a)C(i-1)-(1+a)^{i-1}. \tag{B.16}$$

One valid solution of (B.16) is

$$C(i)=-\frac{i}{a}(1+a)^{i-1}. \tag{B.17}$$

From equation (B.12) we conclude

$$A+B=0. \tag{B.18}$$

From equation (B.14) it is seen that

$$\begin{aligned} A-A(1+a)^N-\frac{N}{a}(1+a)^{N-1} &= 0, \\ A &= \frac{N(1+a)^{N-1}}{a(1-(1+a)^N)}. \end{aligned} \tag{B.19}$$

From (B.18) we know that $B=-A$, thus,

$$E_i=\frac{a^2(\Phi_r-\Phi_l)}{2(1-(1+a)^N)^2}\left[(1-(1+a)^N)i(1+a)^{i-1}+N(1+a)^{N+i-1}-N(1+a)^{N-1}\right]. \tag{B.20}$$

Appendix C

Analytic Solutions of 2D Homogeneous Advection Diffusion Equations

C.1 Transformation to Helmholtz Equation

Start with the steady homogeneous constant coefficient advection diffusion equation, given as

$$-\Gamma\triangle\Phi + \vec{V}\cdot\vec{\nabla}\Phi = f. \tag{C.1}$$

First, since $\vec{V}$ is a constant vector, we can select a new orthogonal coordinate system with the x-direction parallel to $\vec{V}$. Thus

$$\vec{V}\cdot\vec{\nabla}\Phi \longrightarrow |\vec{V}|\frac{\partial\Phi}{\partial x}.$$

Now, let $V = |\vec{V}|$ and (C.1) can be rewritten as

$$-\Gamma\triangle\Phi + V\frac{\partial\Phi}{\partial x} = f. \tag{C.2}$$

We make the following substitution:

$$\Phi(x,y) = e^{\frac{Vx}{2\Gamma}}w(x,y). \tag{C.3}$$

This yields

$$\begin{aligned} e^{\frac{Vx}{2\Gamma}}(-\Gamma\triangle w + (\frac{V}{2\Gamma})^2 w) &= f, \\ -\Gamma\triangle w + (\frac{V}{2\Gamma})^2 w &= f, \end{aligned} \tag{C.4}$$

which is the Helmholtz Equation. We define the Helmholtz operator L as

$$L(w) = -\Gamma \triangle w + (\frac{V}{2\Gamma})^2 w.$$

The associated Green's Function for the Helmholtz operator, i.e., the solution of $L(G((x,y),(r,s))) = \delta((x,y),(r,s))$ (δ being a unit delta function centred at the point (x,y)=(r,s)), is given as

$$G((x,y),(r,s)) = \frac{1}{2\pi} K_0(\frac{V}{2\Gamma}||(x,y),(r,s)||), \tag{C.5}$$

(see Stakgold [35] for analysis of Helmholtz operators). $||a,b||$ represent the Euclidean metric. K_0 is the zero order modified Bessel function of the second kind.

The point source solutions are generated using a delta function at the origin. In this case, the Green's function *is* the desired solution of the Helmholtz equation. Equation (C.3) can then be used to calculate Φ.

C.1.1 Half Plane Problems

From the properties of the delta function, we know that

$$w(x,y) = \int_\Omega w(r,s)\delta((x,y),(r,s))d\Omega,$$

where Ω is expressed in source coordinates (r,s). We will define w to be the solution of the homogeneous Helmholtz equation, $L(w) = 0$. Letting H represent the appropriate Green's function for this problem we can state

$$w = \int_\Omega wL(H) - HL(w)d\Omega.$$

Applying Green's Theorem to this integral gives

$$w = \int_{\partial\Omega} H\frac{\partial w}{\partial n} - w\frac{\partial H}{\partial n}.dr \tag{C.6}$$

For the half plane problems, the boundary, $\partial\Omega$, will be the r-axis, $s = 0$.
For Dirichlet problems we can choose a Green's Function such that

$$H_{\partial\Omega} = H_{s=0} = 0.$$

Then we can represent the solution for w as a known line integral. That is, given the dirichlet boundary condition:

$$\Phi_{y=0}(x) = m(x),$$

$$m(x) = e^{\frac{Vx}{2K}} p(x),$$

$$w_{y=0}(x) = p(x).$$

we can write

$$w(x) = \int_{s=0} -p(r)\frac{\partial H}{\partial n}.dr \tag{C.7}$$

For this problem, a good choice for H would be

$$H = G((x,y),(r,s)) - G((x,-y),(r,s)),$$

which has the property of $H_{y=0} = 0$.

For the upper half plane we can see that

$$\frac{\partial H}{\partial n} = -\frac{\partial H}{\partial s}.$$

If we let $z = V/(2k) \cdot \sqrt{(x-r)^2 + (y-s)^2}$ then along the line integral where $s = 0$ we have

$$\begin{aligned}
-\frac{\partial H}{\partial s} &= \frac{1}{2\pi}\left(\frac{\partial}{\partial s}K_0\left(\frac{V}{2K}||(x,y),(r,0)||\right) - \frac{\partial}{\partial s}K_0\left(\frac{V}{2K}||(x,-y),(r,0)||\right)\right) \\
&= \frac{1}{\pi}\frac{\partial}{\partial s}(K_0(z)) \\
&= \frac{1}{\pi}\frac{\partial}{\partial z}(K_0(z))\frac{\partial z}{\partial s} \\
&= \frac{1}{\pi}(-K_1(z))\frac{-Vy}{2kz},
\end{aligned}$$

where K_1 is a first order modified Bessel function of the second kind.

Thus, putting (C.3), (C.7), and (C.8) together the form for the solution can be seen to be

$$\Phi(x,y) = e^{\frac{Vx}{2k}}\frac{Vy}{2k}\int_{-\infty}^{\infty}\frac{K_1(z)m(r)e^{\frac{Vr}{2k}}}{z}dr. \tag{C.8}$$

For calculation purposes, each point of the grid requires a values of Φ associated with it. In this case, the integral above in numerically approximated using an auto-refining quadrature integration routine until the desired accuracy is achieved. Keeping in mind that for a compact boundary condition (ie. $m(x) \neq 0$ on only a finite interval) the integration need only be performed in the region where $m(x)$ is non-zero.

The modification to the procedure above for a Neumann problem is not very difficult. Begin again at (C.6), but this time we require a Green's function such that at $s = 0$

$$\frac{\partial H}{\partial n} = 0.$$

A valid candidate would be

$$H = G((x,y),(r,s)) + G((x,-y),(r,s)).$$

which leaves

$$w(x,y) = \int_{s=0} H \frac{\partial w}{\partial n} dr. \tag{C.9}$$

If we specify a Neumann boundary condition along the x-axis as

$$\begin{aligned} \frac{\partial \Phi}{\partial y} &= m(x), \\ \frac{\partial w}{\partial n} &= -m(x), \end{aligned}$$

we can derive the solution to the Neumann half plane problem,

$$\Phi(x,y) = e^{\frac{Vx}{2k}} \frac{1}{\pi} \int_{-\infty}^{\infty} -m(r) K_0(z) dr. \tag{C.10}$$

Appendix D

Numerical Solutions of Navier-Stokes Equations

In this appendix, the methodology for solving the Navier-Stokes for this thesis is presented. This work only deals with our particular approach and is not intended to characterise this vast topic. A similar description of this method can be found in Broberg [9], with only a few exceptions.

A reduced nomenclature table is given next in §D.0.2 for more convenient reference.

D.0.2 Nomenclature

A	coefficient matrix
i	control volume index
k	number of neighboring nodes
L	distance along streamline from integration point to element boundary
N	number of control volumes in mesh
$\vec{n} = (n_x, n_y)$	outward normal unit vector
P	fluid pressure of exact solution
p	fluid pressure of approximate solution
$prev$	associated with the solution from the previous iteration
S	surface of arbitrary volume
t	time
s, t	local element coordinates
U	X-component of fluid velocity at nodal point
V	Y-component of fluid velocity at nodal point
$\vec{V} = (u, v)$	fluid velocity vector
$\mathcal{V}$	arbitrary control volume
x, y	Cartesian coordinates
α	relaxation parameter
μ	fluid dynamic viscosity
ρ	fluid density
τ, σ	shear and normal fluid stress

Superscripts

p	associated with the conservation of mass equation
P	contribution from pressure to algebraic equations
u	associated with the x-momentum equation
U	contribution from x-velocity to algebraic equations
v	associated with the y-momentum equation
V	contribution from y-velocity to algebraic equations
t	contribution from transient terms to algebraic equations
s	contribution from additional sources to algebraic equations

Subscripts

$1, 2, 3, 4$	associated with local nodes numbered 1 through 4 respectively
$interp$	linearly interpolated integration point velocity
ip	integration point
n	normal to the surface
SCV	subcontrol volume
$spec$	specified as part of the boundary condition definition
t	tangential to the surface
$upstream$	advected velocity from upwinding scheme
x, y	in the x and y coordinate directions respectively

D.1 Mathematical Model

The starting point for a numerical simulation of fluid flow is the governing equations employed. Our work utilizes the laminar form of the two-dimensional Navier-Stokes equations. The flow is considered to be incompressible. Body forces were neglected. This leaves three unknowns to solve for in two space dimensions : the two orthogonal components of velocity, and pressure.

D.1.1 Equations of Motion

The laminar two-dimensional incompressible Navier-Stokes equations are a coupled partial differential equation system representing the conservation of mass and linear momentum. These equations are presented here in their control volume formulation. $\mathcal{V}$ represents an arbitrary control volume which is contained by the surface S.

Conservation of Mass

$$\int_S \rho \vec{V} \cdot \vec{n} dS = 0. \tag{D.1}$$

The term $\rho\vec{V}$ represents the mass flux vector, and $\rho\vec{V} \cdot \vec{n}dS$ then is the mass flux crossing a particular differential volume surface dS . Thus, the Conservation of Mass equation, (D.1), states that the net mass leaving a volume bounded by S equals zero.

Conservation of Linear Momentum

The conservation of linear momentum in the x and y coordinate directions can be stated as

$$\int_{\mathcal{V}} \frac{\partial}{\partial t}(\rho u)dx + \int_S \rho u \vec{V} \cdot \vec{n} dS = \int_S \mu \nabla u \cdot \vec{n} dS - \int_S p n_x dS \tag{D.2}$$

and

$$\int_{\mathcal{V}} \frac{\partial}{\partial t}(\rho v)dx + \int_S \rho v \vec{V} \cdot \vec{n} dS = \int_S \mu \nabla v \cdot \vec{n} dS - \int_S p n_y dS. \tag{D.3}$$

For both equations, the first term describes the accumulation of momentum while the second term represent the advection of momentum across the volume boundaries. The first term on the right describes the viscous force acting on the volume boundaries (both normal and shear forces). The last term is the contribution from the pressure field.

D.1.2 Boundary Conditions

There were three types of boundary conditions used in this study:

1. Solid walls
2. Inflow regions
3. Outflow regions

Each is described in its own section below.

Solid Walls

At a solid wall, the *no slip* condition is imposed. The velocity of the fluid relative to the boundary is zero. In addition, the normal viscous stress at the wall is assumed to be zero. This latter constraint is valid for most bounded flows, but may introduce inaccuracies near stagnation points. It is invoked to simplify the numerical implementation.

1. $\vec{V} = 0$
2. $\sigma = 0$

Inflow Regions

Inflow boundary conditions were of the *velocity specified* type. As with the solid wall condition, the normal viscous stress is assumed to be zero. Thus:

1. $\vec{V} = \vec{V}_{spec}$
2. $\sigma = 0$

Outflow Regions

A *pressure specified* outlet condition is used. Although, for incompressible flows, the pressure level is arbitrary, it still requires a fixed reference within the computational domain; this is done at the outlet.

Also, the viscous terms at the outlet were ignored. Again, this is for numerical expediency, but it is also true that if there were strong viscous contributions to the momentum equation at the outflow region, then almost all numerical implementations outflow conditions fail. This being stated, setting the viscous terms to zero has only a minor and acceptable accuracy penalty.

The outflow boundary condition can then be stated as:

1. $p = p_{spec}$
2. $\tau = 0$
3. $\sigma = 0$

Here p_{spec} is the specified pressure profile at the outflow boundary condition

D.2 Discretization

For this research, the equations of motion were solved with a element-based finite volume method. This entails dividing the domain into a finite number of elements, out of which control volumes were generated. The conservation equations are applied to the control volumes. The integral relations presented in D.1.1, and the boundary conditions of D.1.2 were approximated to produce an algebraic equation set.

The solutions were generated from their primitive variable representations, (u, v, p), and solved using a co-located strategy (all variables stored at the nodal locations), in contrast to the commonly used staggered grid structure. The co-located formulation requires special consideration (as discussed in §D.2.3) to prevent oscillatory solutions.

As might have been noticed, the time dependent form of the Navier-Stokes equations has been presented, despite the stated interest in steady solutions. This is because the steady-state solution can be arrived at by advancing an initial solution profile through time until it converges on the steady-state solution. The method here is referred to as *pseudo-time-stepping*, not a full transient calculation, since each timestep is not iterated to achieve a solution of the non-linear equation set, but only the current linear approximation.

D.2.1 The Finite Element Grid

The meshes used are made up of regular brick elements with control volumes generated as shown in Schneider and Raw [33]. Extensions to other types of meshes are possible, but not utilized in this thesis.

The quadrilateral elements have a node at each of the four corners. To generate control volumes, straight lines were created bisecting the opposite sides of the quadrilateral (see figure D.1).

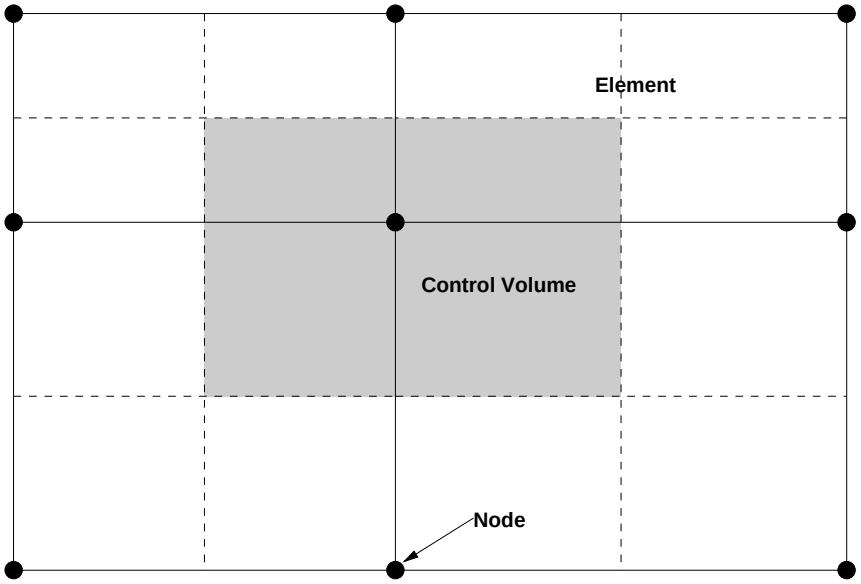


Figure D.1: Finite Element Mesh and Control Volume

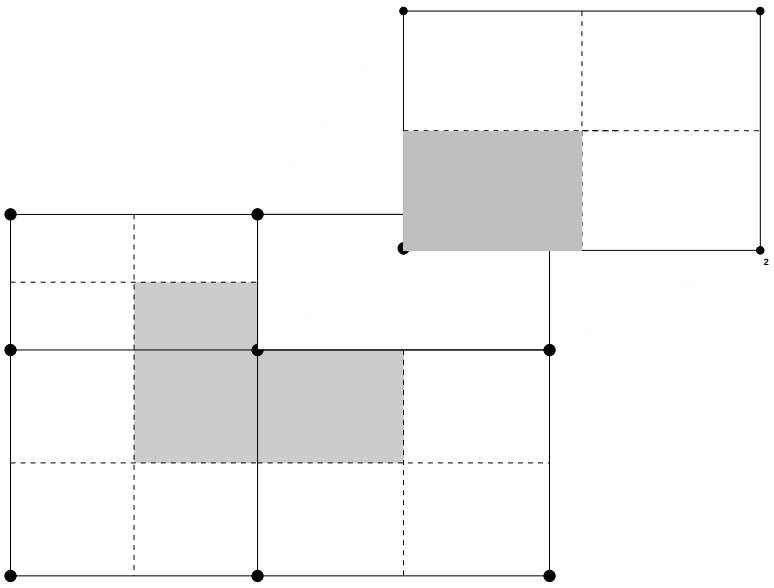


Figure D.2: One Element From a Mesh

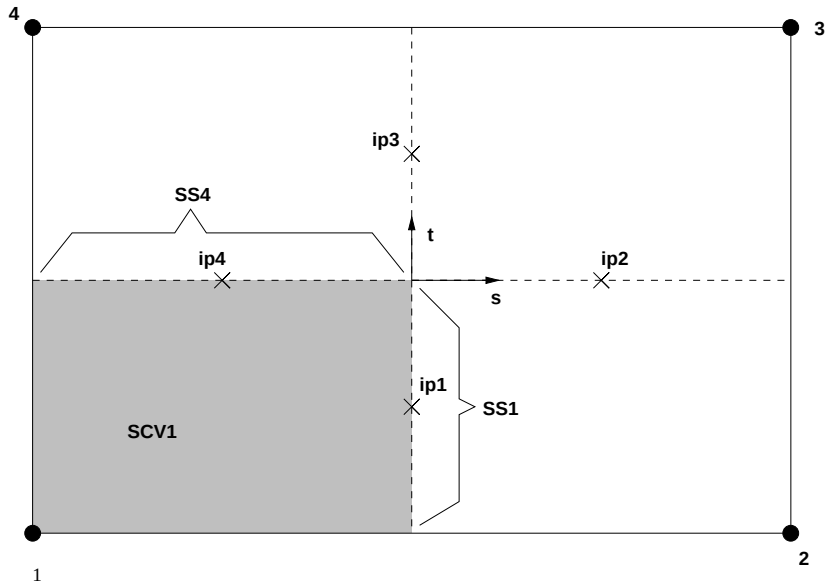


Figure D.3: The Quadrilateral Finite Element

Within each element, a local coordinate system is employed and denoted If we turn our attention from a mesh of elements, to just one element (see figure D.2), we can see that each element participates in four control volumes.

The element is spanned by s and t coordinates, which range from -1 to 1 (see figure D.3). The boundaries between control volumes are on the lines $s = 0$ and $t = 0$. The bilinear shape functions for the element, expressed in local coordinates, is given by:

$$N_1(s,t) = \frac{1}{4}(1-s)(1-t) \quad \text{(D.4)}$$

$$N_2(s,t) = \frac{1}{4}(1+s)(1-t) \quad \text{(D.5)}$$

$$N_3(s,t) = \frac{1}{4}(1+s)(1+t) \quad \text{(D.6)}$$

$$N_4(s,t) = \frac{1}{4}(1-s)(1+t) \quad \text{(D.7)}$$

The shape functions are used to describe the variation of the flow properties within the element based on the values at the nodes.

During the assembly process, each element is visited in turn. The contributions to each *sub-control volumes* (SCV) conservation equation is calculated based on the flow properties at the *integration points* (ip#) (except where noted in the sections to follow). These contributions are then assembled into the global system of algebraic equations to generate a complete conservative control volume. For example: For the sub-control volume highlighted in figure D.3, the flux through the faces **ip1** and **ip4** need to be considered. These coefficients then contribute to the complete algebraic equation set for the control volume highlighted in figure D.1

D.2.2 Discrete Momentum Equations

This section details the discrete form of the u-momentum equation (the v-momentum equation is similar).

Consider one sub-control volume of an element, SCV1 from figure D.3, which is bounded by the subsurfaces 1 and 4. First consider the transient term. *prev* refers to the use of the previous iteration value for this term. The time derivative will be approximated as a lumped mass, given by

$$\int_{SCV1} \frac{\partial}{\partial t}(\rho u)dx \approx \frac{\rho V_{SCV1}}{\Delta t}(U_1 - U_1(prev)), \tag{D.8}$$

where we denote the nodal value of the u component of the velocity with the uppercase U . Δt is the user defined time-step. The subscript 1 indicates which node we are referring to locally.

Next, consider the viscous term. The net force due to viscous stress acting on subsurfaces 1 and 4 can be approximated by

$$\int_{SS1} \mu \nabla u \cdot \vec{n} dS \approx \frac{\mu \Delta y}{2} \frac{\partial u}{\partial x}_1 \tag{D.9}$$

and

$$\int_{SS4} \mu \nabla u \cdot \vec{n} dS \approx \frac{\mu \Delta x}{2} \frac{\partial u}{\partial y}_4 , \tag{D.10}$$

where the subscripts indicate that the derivatives are evaluated at the integration points.

The pressure term is simply approximated by

$$\int_{SS1} p n_x dS \approx p_1 \Delta y/2 \tag{D.11}$$

and

$$\int_{SS4} p n_x dS = 0. \tag{D.12}$$

The second term is zero since $n_x = 0$ on SS4 .

This leaves the convective term, $\int_S \rho u \vec{V} \cdot \vec{n} dS$. If we simply linearly interpolate a value for velocity at the integration points based the nodal values, then the coupling between the momentum equations of adjacent control volumes becomes very weak (or non-existent for one-dimensional flows) as discussed by Patankar [28]. This can result in arbitrary non-physical solution oscillations.

To combat this problem, we introduce two additional concepts. First, there is a need to distinguish between the *advecting* velocity at the integration point, $\vec{V} \cdot \vec{n}$, and the *advected* velocity, u. The advecting velocity is used to determine the mass flux passing through the subsurface and is discussed in detail in §D.2.3.

The advecting velocity is calculated based on a pressure-velocity coupling scheme outlined in §D.2.3. The advected velocity is determined by an upwind based approach. In this study two forms of upwinding schemes are implemented: simple Upwind Differencing (UDS), and Linear Profile Skew Upwind Differencing with Physical Advection Correction (LPS+PAC).

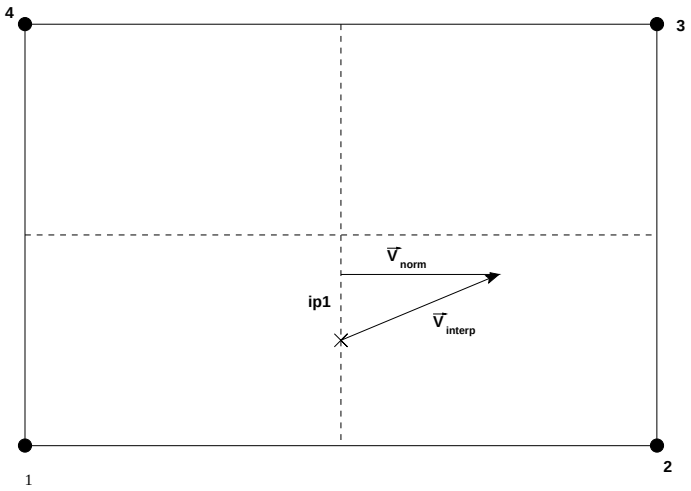


Figure D.4: Upwind Differencing Within Element

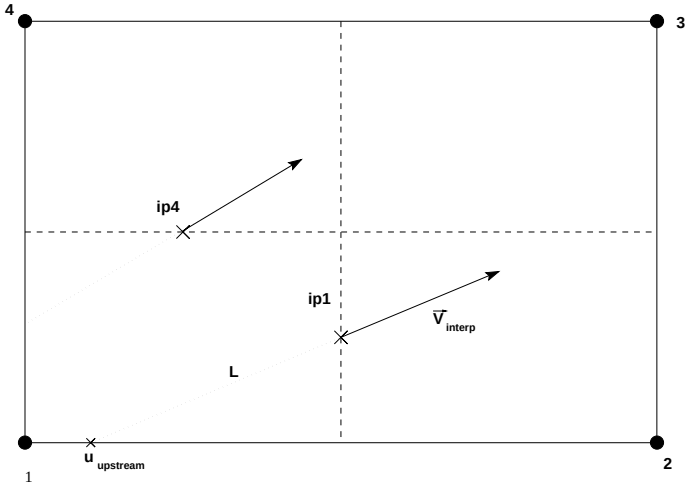


Figure D.5: Linear Profile Skew Upwinding with Advection Correction

Upwind Differencing

Referring to figure D.4, a value for the velocity vector at **ip1** is interpolated from the adjacent nodes, referred to as V_{interp}. Depending on the sign of the normal vector, either U_1 or U_2 is used for the advected velocity. In the example figure, U_1 would be used.

Linear Profile Skew Scheme with Physical Advection Correction

As with the upwind differencing scheme, an initial interpolation of the integration point velocity is made. This time however, we now track a straight line (a streamline) back to the element boundary (see figure D.5) A value for $u_{upstream}$ is calculated using the element shape function (The *Linear Profile*). We now express the integration point convected velocity as

$$\hat{u}_{ip1} = u_{upstream} + L\frac{\partial u}{\partial s}, \tag{D.13}$$

where L is the distance from the integration point and the element boundary along the

fictitious streamline, and $\frac{\partial u}{\partial s}$ represents how u changes across this distance. This term is the Physical Advection Correction (PAC) term and is what makes the scheme higher accuracy. Different forms of the advection correction term can result in different numerical techniques. In this thesis, the representation employed by Raithby [29] is employed, but modified by the inclusion of the pressure gradient as in Schneider and Raw [33] and the effects of viscosity. We now have

$$\frac{\partial u}{\partial s} = \frac{1}{\rho \mathcal{V}} \left(\mu \frac{\partial^2 u}{\partial x^2} + \frac{\partial p}{\partial x} + \dot{S}^u \right), \tag{D.14}$$

where $\dot{S}^u$ would be any additional source terms in the x-momentum equation. (D.14) is what is referred to as a PAC term. It needs to be evaluated at the integration point (where we need $\frac{\partial u}{\partial s}$). In the numerical code, this is accomplished by calculating the PAC term at every node using the previous iteration values of U, V, P, then interpolating the nodal values of the PAC term to the integration point, using the harmonic mean. The PAC terms, since they are lagged (calculated based on the previous iteration solution instead of the current solution), do not end up in the active implicit matrix, but show up in the right hand side vector B^{us}, as shown later.

In addition, This lagging effect can result in solution oscillations from one iteration to the next, hence, the value used for the nodal PAC term evaluation is relaxed using the previous nodal PAC terms as such:

$$PAC_{used} = \alpha PAC_{current} + (1 - \alpha) PAC_{previous} \tag{D.15}$$

Thus, all components of equation (A.2) have an algebraic representation. The algebraic equations for the subcontrol volumes can then be summed over the entire control volume i to form a complete algebraic conservation equation, (D.16). The first superscript refers to what momentum equation is being assembled. The t superscript indicate a term from the transient portion of the equation, as the s indicates source terms. The U, V, P superscripts indicate what variable these assembled coefficients modify. The j index cycle from 1 to k, where k is the number of neighboring nodes that participate in the ith control volume.

$$A_i^{ut} U_i + \sum_{j=1}^{k} A_{ij}^{uU} U_j + \sum_{j=1}^{k} A_{ij}^{uV} V_j + \sum_{j=1}^{k} A_{ij}^{uP} P_j = B_i^{ut} + B_i^{us} \tag{D.16}$$

$$A_i^{vt} V_i + \sum_{j=1}^{k} A_{ij}^{vU} U_j + \sum_{j=1}^{k} A_{ij}^{vV} V_j + \sum_{j=1}^{k} A_{ij}^{vP} P_j = B_i^{vt} + B_i^{vs} \tag{D.17}$$

D.2.3 Discrete Mass Equation

The mass flow through SS1 and SS4 can be approximated by

$$\int_{SS1} \rho \vec{V} \cdot \vec{n} dS \approx \dot{m}_1 \tag{D.18}$$

and

$$\int_{SS4} \rho \vec{V} \cdot \vec{n} dS \approx \dot{m}_4, \tag{D.19}$$

where the mass flows, $\dot{m}_i$, are determined from the integration point advecting velocity, $\hat{u}$,

$$\dot{m}_1 \equiv \rho(\hat{u}\Delta y/2)_1. \tag{D.20}$$

As mentioned earlier, directly interpolating a value for $\hat{u}$ can lead to unwanted equation decoupling for a co-located variable method . It is desired to couple the mass conservation equation to pressure field. The technique of Rhie and Chow [30] is a pressure/velocity coupling method that is stable for a wide variety of fluid flow problems. It is summarized here.

An o superscript means that the term is calculated from the previous iteration. The overbar indicates a harmonic average of the adjacent nodal values for that variable. As before, uppercase U indicates the nodal value. For SS1, the u-component of advecting velocity is expressed as

$$\hat{u}_{ip1} = \frac{1}{2}(U_1 + U_2) + f^u_{ip1}\left[\left(\frac{\partial p}{\partial x}\right)_{ip1} - \left(\frac{\bar{\partial p}}{\partial x}\right)^o_{ip1}\right] - \frac{\rho f^u_1}{\Delta t}(\hat{u}^o_{ip1} - \bar{\hat{u}}^o_{ip1}), \tag{D.21}$$

where

$$f^u_{ip1} = d^u_{ip1}/(1 - \rho d^u_{ip1}/\Delta t) \tag{D.22}$$

and

$$\bar{\hat{u}}^o_1 = \frac{1}{2}(U^o_1 + U^o_2). \tag{D.23}$$

The coefficient d^u_1 is created by an arithmetic average of the adjacent nodal values D^u_i, where

$$D^u_i = \Delta x \Delta y / \left(4\sum_{j=1}(A^{uU}_{ij} + A^{uV}_{ij})^o\right) \tag{D.24}$$

and $\sum_{j=1}(A^{uU}_{ij} + A^{uV}_{ij})$ represents the sum of the diffusive and convective terms in the algebraic equation for the x-momentum conservation equation. This generates a scaling term which ensures a stable matrix solution.

Equation D.21 does indeed look elaborate. In simpler terms, it represents an interpolated value for $\hat{u}$, modified by a second-order pressure and temporal correction. The pressure terms ensure that the pressure solution does not decouple between adjacent control volumes. The temporal terms ensure that the final solution converged to is independent of the timestep Δt used for the solution.

Once the advecting velocity is calculated, then all the required contributions to the discrete mass equation can be assembled. Equation (A.1) now becomes the algebraic form

$$\sum_{j=1}^{k} A^{pU}_{ij} U_j + \sum_{j=1}^{k} A^{pV}_{ij} V_j + \sum_{j=1}^{k} A^{pP}_{ij} P_j = B^{ps}_i. \tag{D.25}$$

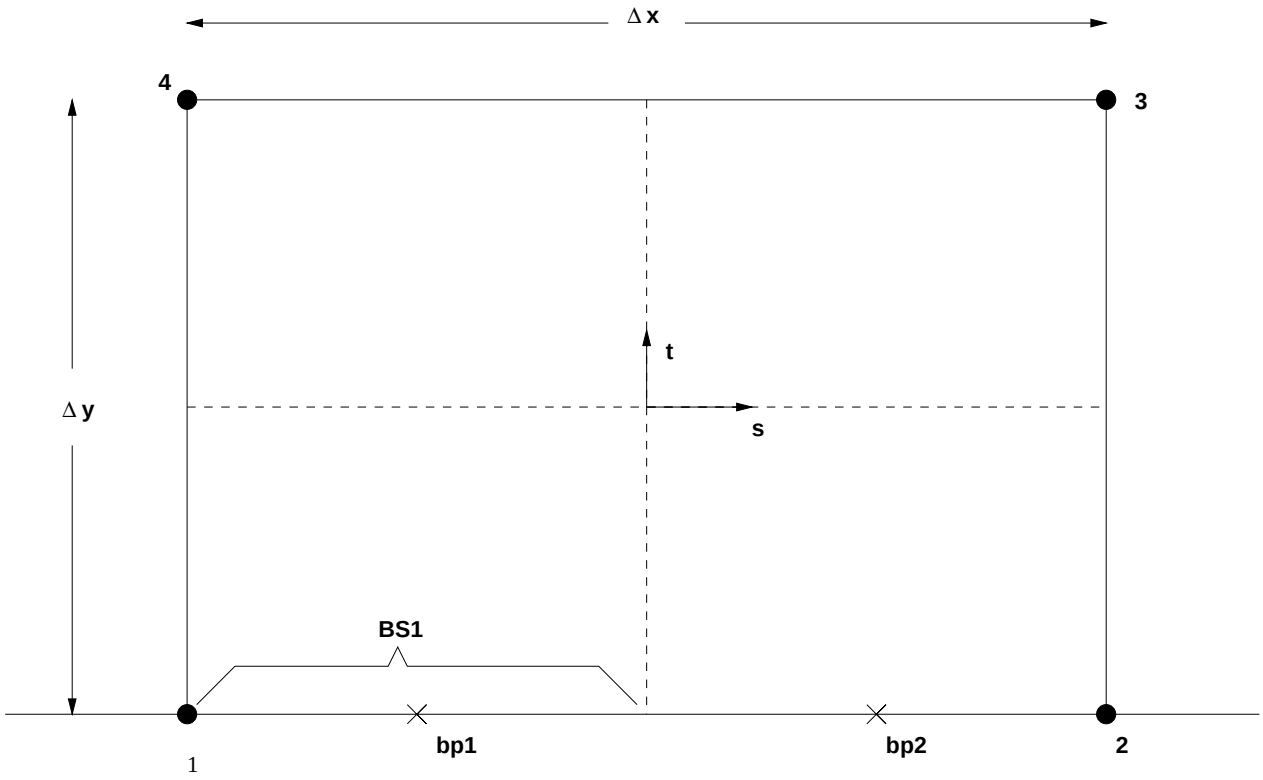


Figure D.6: Element on Boundary

D.2.4 Boundary Conditions

Boundary conditions represent the fluid equations integrated along the boundary subsurfaces. A representative boundary element is shown in figure D.6. The boundary conditions are formulated in a conservative equation form.

Solid Walls

For a solid wall, the conservation of mass equation requires that no mass travel through the wall. For the boundary element shown in figure D.6 this means

$$\dot{m}_{bp1} = \dot{m}_{bp2} = 0. \tag{D.26}$$

Hence, no contribution to the mass conversation equation need be added.

Due to the zero mass flow condition, there is also no need to add a convective transport term for the momentum equations.

As was stated in §D.1.2, the normal viscous stress terms are ignored. This leaves an approximation for the tangential shear stress on boundary surface BS1 as

$$\int_{BS1} \mu \nabla u \cdot \vec{n} dS \approx \frac{\mu \Delta x}{2} \left. \frac{\partial u}{\partial y} \right|_{bp1}, \tag{D.27}$$

where the derivative is approximated using the element shape functions.

The pressure force along the boundary surface is approximated as

$$\int_{BS1} p\vec{j} \cdot \vec{n} dS \approx -p_{bp1} \Delta x/2, \tag{D.28}$$

where p_{bp1} is arrived at via the element shape functions.

Inflow Boundary

For a velocity specified inlet condition the values of u and v will be specified at the integration points. Thus, with reference to figure D.6, we have

$$\dot{m}_{bp1} \approx \rho v_{bp1,spec} \Delta x/2 \tag{D.29}$$

and

$$\dot{m}_{bp2} \approx \rho v_{bp2,spec} \Delta x/2. \tag{D.30}$$

These terms do not implicitly depend on the nodal values during a given linear iteration, and hence can be considered as mass source terms for these boundary control volumes.

For the convection of momentum we have:

$$\int_{BS1} \rho u \vec{V} \cdot \vec{n} dS \approx \dot{m}_{bp1} u_{bp1,spec} \tag{D.31}$$

and

$$\int_{BS1} \rho v \vec{V} \cdot \vec{n} dS \approx \dot{m}_{bp1} v_{bp1,spec}. \tag{D.32}$$

The tangential viscous and pressure forces would be calculated as in equations D.27 and D.28, respectively.

Outflow Boundary

Since specifying the velocity at the outflow boundary can sometimes be numerically unstable (from discrepancies in the approximate mass into and out of the computational domain), the fluid velocity at the boundary points is expressed implicitly in terms of the nodal values. Hence the flow out of surfaces BS1 and BS2 is expressed as

$$\dot{m}_{bp1} \approx \rho v_{bp1} \Delta x/2 \tag{D.33}$$

and

$$\dot{m}_{bp2} \approx \rho v_{bp2} \Delta x/2, \tag{D.34}$$

where v_{bp1} is interpolated via the element shape functions.

The convected component for the momentum equations is similar. For BS1 we have

$$\int_{BS1} \rho u \vec{V} \cdot \vec{n} dS \approx \dot{m}_{bp1} u_{bp1} \tag{D.35}$$

and

$$\int_{BS1} \rho v \vec{V} \cdot \vec{n} dS \approx \dot{m}_{bp1} v_{bp1}. \tag{D.36}$$

As explained in §D.1.2, the normal and tangential viscous stress at the outlet is ignored. The pressure force is based on the specified pressure level. Thus

$$\int_{BS1} p\vec{j} \cdot \vec{n} dS \approx -p_{bp1,spec} \Delta x/2. \tag{D.37}$$

D.3 Solution of Coupled Equation Set

Generating a steady-state solution of the equations of motion is a process of advancing the solution through time, making linear approximations of non-linear terms at each timestep, until a steady solution is achieved.

At each timestep, the previous timestep solution will not satisfy the algebraic equation set (until the steady solution is achieve). This results in an equation imbalance, the *outer loop iterative residual*, given as (recalling equations D.16, D.17, and, D.25):

$$R_i^u = A_i^{ut} U_i + \sum_{j=1}^{k} A_{ij}^{uU} U_j + \sum_{j=1}^{k} A_{ij}^{uV} V_j + \sum_{j=1}^{k} A_{ij}^{uP} P_j - B_i^{ut} - B_i^{us} \tag{D.38}$$

$$R_i^v = A_i^{vt} V_i + \sum_{j=1}^{k} A_{ij}^{vU} U_j + \sum_{j=1}^{k} A_{ij}^{vV} V_j + \sum_{j=1}^{k} A_{ij}^{vP} P_j - B_i^{vt} - B_i^{vs} \tag{D.39}$$

$$R_i^p = \sum_{j=1}^{k} A_{ij}^{pU} U_j + \sum_{j=1}^{k} A_{ij}^{pV} V_j + \sum_{j=1}^{k} A_{ij}^{pP} P_j - B_i^{ps} \tag{D.40}$$

It is this residual that is used to drive the non-linear equation set to convergence.

To advance a solution from timestep t to $t + \Delta t$ the following algorithm is used:

1. Calculate subsurface advecting velocities using previous timestep solution. Calculate subsurface mass flux.
2. If required, calculate PAC terms using previous timestep solution.
3. Assemble the coefficients ($A^{uU}, A^{uV} ...., A^{pP}$) using the mass flows from step 1 for the convective terms.
4. Substitute previous solution into D.38,D.39, D.40. Calculate outer loop iterative residual.
5. Assemble coefficient matrix into sparse YSMP format and pass to WATSIT[1] with residual.
6. The newly solved U, V, P field become the previous timestep values for the next outer loop.

[1] A proprietary Conjugate Gradient iterative solver package licensed by the Scientific Computation Group at the University of Waterloo. See ref [41]