HIGH-ORDER FINITE-VOLUME DISCRETISATIONS FOR
SOLVING A MODIFIED ADVECTION-DIFFUSION PROBLEM
ON UNSTRUCTURED TRIANGULAR MESHES

by

MICHAEL VAN ALTENA
B.Math., The University of Waterloo, 1997

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in
THE FACULTY OF GRADUATE STUDIES
(Department of Mechanical Engineering)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
October 1999
©Michael Van Altena, 1999
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering

The University of British Columbia
Vancouver, Canada

Date Oct 12, 1999
Abstract

A fourth-order accurate diffusive flux calculation scheme has been developed which can be incorporated into any existing reconstruction-based unstructured mesh inviscid flow solver. The flux scheme makes use of a Laplace problem to model the viscous terms and of the existing least-squares reconstruction code used in Dr. Carl F. Ollivier-Gooch's inviscid flow solver. Boundary conditions on the solution are enforced during the reconstruction, which requires boundary constraints to be included in the least-squares matrix. The Taylor Constraints method for generating boundary constraints is developed and then discarded due to its drawbacks. The Gauss Constraints method is then developed which is simpler and avoids the drawbacks of the Taylor Constraints method. The flux integration is validated to demonstrate fourth-order accuracy on square, circle and cardioid meshes with a combination of Dirichlet and Neumann boundary conditions. The flux integration scheme is shown to be fourth-order accurate for a modified advection-diffusion problem of the same form as the Navier-Stokes' momentum equations. The diffusive flux calculation scheme can later be combined with an existing inviscid flow solver to solve the viscous compressible Navier-Stokes' equations.
Contents

Abstract ii

List of Tables v

List of Figures vi

Acknowledgements ix

1 Introduction 1
   1.1 Objectives and Scope ..................................................... 3

2 Flow Solvers 5
   2.1 Finite-Volume Methods ................................................... 5
   2.2 Inviscid Flow Solver ...................................................... 11
   2.3 Viscous Modelling ......................................................... 12
   2.4 Hybrid Scheme ............................................................. 13

3 High-Order Flux Integrals 17
   3.1 Gradient Approximation .................................................. 18
   3.2 Flux Integration: Cell-Centred Control Volumes ...................... 18
   3.3 Flux Integration: Vertex-Centred Control Volumes ................... 21

4 Boundary Conditions 24
   4.1 Taylor Constraints .......................................................... 25
      4.1.1 Dirichlet Boundary Conditions ..................................... 25
      4.1.2 Neumann Boundary Conditions ..................................... 30
      4.1.3 Dirichlet & Neumann Combined Boundary Conditions .......... 34
      4.1.4 Drawbacks ............................................................. 38

iii
CONTENTS

4.2 Gauss Constraints ........................................................................................................... 39
  4.2.1 Dirichlet Boundary Conditions ................................................................................. 40
  4.2.2 Neumann Boundary Conditions ................................................................................ 40
  4.2.3 Dirichlet & Neumann Combined Boundary Conditions .............................................. 42
  4.2.4 Constraint Comparison ............................................................................................. 44

4.3 Constrained Least-Squares Reconstruction Method ......................................................... 46

5 Validation ............................................................................................................................ 49
  5.1 Interior Control Volume Flux Integral Validation ........................................................... 51
  5.2 Dirichlet Boundary Conditions on a Square Mesh ......................................................... 53
    5.2.1 Accuracy Analysis — Taylor Constraints ............................................................... 53
    5.2.2 Accuracy Analysis — Gauss Constraints ............................................................... 55
  5.3 Neumann Boundary Conditions on a Square Mesh ....................................................... 58
  5.4 Combined Dirichlet & Neumann Boundary Conditions on a Square Mesh .................... 60
  5.5 Singular Point on a Square Mesh ................................................................................... 62
  5.6 Curved Mesh Boundaries ............................................................................................. 63
    5.6.1 Circle Mesh Boundary ......................................................................................... 65
    5.6.2 Cardioid Mesh Boundary ....................................................................................... 67
  5.7 Summary ....................................................................................................................... 70

6 Application ........................................................................................................................ 71
  6.1 Modified Advection-Diffusion Problem ......................................................................... 71
  6.2 Applying the Finite-Volume Scheme ............................................................................. 72
  6.3 Validation ...................................................................................................................... 73

7 Summary and Conclusions .................................................................................................. 76
  7.1 Issues During Development ......................................................................................... 78
  7.2 Computer Resource Usage ......................................................................................... 80
  7.3 Direction for Future Work ........................................................................................... 84

Bibliography ............................................................................................................................ 87

Appendix A ................................................................................................................................ 88
List of Tables

4.1 Constraint Comparison for Control Volumes with Taylor Constraints 38
4.2 Constraint Comparison for Control Volumes with Gauss Constraints 45

5.1 Error Norms of the Interior Control Volumes 52
5.2 Error Norms with Dirichlet Taylor Constraints 54
5.3 Error Norms with Dirichlet Gauss Constraints 56
5.4 Error Norms with Neumann Gauss Constraints 59
5.5 Error Norms with Gauss Constraints 61
5.6 Error Norms with Gauss Constraints 64
5.7 Error Norms on a Circle Mesh 66
5.8 Error Norms on a Cardioid Mesh 69

6.1 Error Norms for the Modified Advection-Diffusion Problem 75

7.1 Mesh Information for Square Meshes 80
7.2 Memory Data for Square Meshes 81
# List of Figures

2.1 Cell-Centred Control Volume ...................................................... 7
2.2 Vertex-Centred Control Volume .................................................... 7
2.3 High-order reconstruction with vertex-centred control volumes .......... 8
2.4 Hybrid Scheme ............................................................................ 15

3.1 Cell-Centred Flux Calculation ...................................................... 19
3.2 Cell Gradients ............................................................................. 20
3.3 Cell Average Gradients ................................................................. 20
3.4 Vertex-Centred Gauss Points........................................................ 23

4.1 Cell-centred control volume's Gauss point locations ....................... 45
4.2 Vertex-centred control volume's Gauss point locations ................... 45
4.3 Example of the least-squares matrix ............................................. 48
4.4 Example of the constrained least-squares matrix ............................. 48

5.1 Error Norm Plot Legend ................................................................. 51
5.2 Initial temperature function for interior control volumes ................ 52
5.3 Exact flux for interior control volumes ......................................... 52
5.4 $L_2$ error norm plots for interior cell-centred control volumes .......... 52
5.5 $L_2$ error norm plots for interior vertex-centred control volumes ...... 52
5.6 Flux error for Cell scheme using Dirichlet Taylor constraints .......... 54
5.7 Flux error for Vertex scheme using Dirichlet Taylor constraints ....... 54
5.8 $L_2$ error norm plots for Cell scheme using Dirichlet Taylor constraints 54
5.9 $L_2$ error norm plots for Vertex scheme using Dirichlet Taylor constraints 54
5.10 Flux error for Cell scheme using Dirichlet Gauss constraints .......... 56
5.11 Flux error for Vertex scheme using Dirichlet Gauss constraints ....... 56
5.12 $L_2$ error norm plots for Cell scheme using Dirichlet Gauss constraints 56
LIST OF FIGURES

5.13 L₂ error norm plots for Vertex scheme using Dirichlet Gauss constraints ... 56
5.14 Initial function for Vertex scheme with Neumann boundary conditions ... 58
5.15 Flux error for Cell scheme using Neumann Gauss constraints .................. 59
5.16 Flux error for Vertex scheme using Neumann Gauss constraints .................. 59
5.17 L₂ error norm plots for Cell scheme using Neumann Gauss constraints ... 59
5.18 L₂ error norm plots for Vertex scheme using Neumann Gauss constraints ... 59
5.19 Initial function for Vertex scheme for combined boundary conditions ......... 60
5.20 Flux error for Cell scheme using combined Gauss constraints .................. 61
5.21 Flux error for Vertex scheme using combined Gauss constraints .................. 61
5.22 L₂ error norm plots for Cell scheme using combined Gauss constraints ... 61
5.23 L₂ error norm plots for Vertex scheme using combined Gauss constraints ... 61
5.24 Initial function of singular point test case on cell-centred control volumes ... 63
5.25 Flux error for singular test case using Cell scheme with Gauss constraints ... 64
5.26 Flux error for singular test case using Vertex scheme with Gauss constraints ... 64
5.27 L₂ error norm plots for Cell scheme using Gauss constraints .................. 64
5.28 L₂ error norm plots for Vertex scheme using Gauss constraints .................. 64
5.29 Initial function for cell-centred control volumes on a circle mesh ............... 65
5.30 Flux error for Cell scheme on a circle mesh ...................................... 66
5.31 Flux error for Vertex scheme on a circle mesh ...................................... 66
5.32 L₂ error norm plots for Cell scheme on a circle mesh ....................... 66
5.33 L₂ error norm plots for Vertex scheme on a circle mesh ....................... 66
5.34 Initial function for Vertex scheme on a cardioid mesh ....................... 68
5.35 Flux error for Cell scheme on a cardioid mesh ...................................... 69
5.36 Flux error for Vertex scheme on a cardioid mesh ...................................... 69
5.37 L₂ error norm plots for Cell scheme on a cardioid mesh ....................... 69
5.38 L₂ error norm plots for Vertex scheme on a cardioid mesh ....................... 69

6.1 Initial function for advection-diffusion problem ................................. 74
6.2 Flux error for Cell scheme for advection-diffusion problem ...................... 75
6.3 Flux error for Vertex scheme for advection-diffusion problem ...................... 75
6.4 L₂ error norm plots for Cell scheme for advection-diffusion problem ............... 75
6.5 L₂ error norm plots for Vertex scheme for advection-diffusion problem ............... 75

7.1 Plot of number of control volumes vs. amount of memory ......................... 81
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Plot of $L_2$ error norm vs. CPU time for Cell scheme</td>
<td>83</td>
</tr>
<tr>
<td>7.3</td>
<td>Plot of $L_2$ error norm vs. CPU time for Vertex scheme</td>
<td>83</td>
</tr>
<tr>
<td>A.1</td>
<td>square0.mesh</td>
<td>88</td>
</tr>
<tr>
<td>A.2</td>
<td>square1.mesh</td>
<td>89</td>
</tr>
<tr>
<td>A.3</td>
<td>square2.mesh</td>
<td>89</td>
</tr>
<tr>
<td>A.4</td>
<td>circle0.mesh</td>
<td>90</td>
</tr>
<tr>
<td>A.5</td>
<td>circle1.mesh</td>
<td>90</td>
</tr>
<tr>
<td>A.6</td>
<td>cardioid0.mesh</td>
<td>91</td>
</tr>
</tbody>
</table>
I would like to thank my research supervisor, Dr. Carl F. Ollivier-Gooch, for his constant support and help throughout the course of this thesis. Being the guinea pig graduate student was a pleasure, and his future students will be lucky to have him. I would also like to thank Jody Kirchner; her friendship throughout my graduate school experience I will never forget.
Chapter 1

Introduction

The ultimate goal of the computational fluid dynamics (CFD) field is to obtain accurate solutions to engineering problems with a rapid turnaround time. This is done by finding an approximate solution to the equations which govern the dynamics of fluids, known as the Navier-Stokes’ equations, and their prescribed boundary conditions. A fluids problem is solved numerically by first decomposing the computational domain into a mesh. The Navier-Stokes’ equations are then discretised and solved over the mesh. The accuracy of the solution and the efficiency with which it is obtained are very important. The computer program should compute an accurate solution in the least amount of time possible.

The geometry of a problem can be as simple as a flat plate, or as complex as a space shuttle. This domain must be decomposed into a mesh for the flow solver. In general there are two types of meshes, structured and unstructured. In a structured mesh the identity of a cell’s neighbours can be determined from the identity of the cell. The mesh cells are almost always quadrilaterals and hexahedra. Structured meshes are difficult and labour intensive to
generate for complex geometries, due to difficulties in meshing around holes, sharp corners, etc. Typically, different regions of a complex domain are meshed separately and then the component meshes are sewn together.

Meshing can be greatly simplified by using unstructured meshes. An unstructured mesh differs from a structured mesh in that the neighbours for any particular cell cannot be computed, but must be stored in a data structure in the computer's memory. An unstructured mesh can be constructed with a variety of polygons in two dimensions (usually triangles and/or quadrilaterals) or polyhedra in three dimensions (usually tetrahedra and/or hexahedra). The topological flexibility of unstructured meshes makes it easier to mesh complex geometries consistently. This thesis uses two dimensional triangular meshes exclusively, although the results should generalize easily to other types of meshes.

With the mesh generated, the governing equations need to be discretised to compute the solution over the mesh. One common approach to discretisation in CFD is the finite-volume method, which performs fluids control volume analysis by calculating the fluxes through the many small control volumes in a mesh. The development of such a method is at the heart of this thesis.

The two most important issues in developing a numerical flow solver are efficiency and accuracy. The problem is that efficiency and accuracy are inversely related. Most flow solvers compute solutions in the control volumes to second-order accuracy, as this can be done quickly. To improve the accuracy of the solution, the mesh is made finer, increasing the number of control volumes while decreasing their size. However the larger number of control volumes also means a larger number of calculations, which require more CPU time.
Another way to improve accuracy is to use the original coarse mesh and compute solutions in the control volumes more accurately. This increases the computation time per control volume, but uses fewer control volumes than mesh refinement.

Which method is more efficient? The trade-off between computational accuracy, mesh refinement and efficiency for viscous flow solvers on unstructured meshes must be quantified. To study this trade-off, a method must first be developed to solve the Navier-Stokes' equations to high-order accuracy on unstructured meshes. This work takes the first step towards building a high-order accurate flow solver by developing a high-order accurate flux integration scheme for diffusive fluxes.

1.1 Objectives and Scope

The main objective of this research is to develop and validate a fourth-order accurate diffusive flux calculation scheme. The flux scheme will make use of a Laplace problem to model the viscous terms and of the existing reconstruction code used in Dr. Carl F. Ollivier-Gooch's inviscid flow solver. This diffusive flux calculation scheme can later be combined with the existing inviscid flow solver to solve the complete viscous compressible Navier-Stokes' equations.

The existing inviscid flow solver is discussed in Chapter 2. Next, the viscous terms in the Navier-Stokes' equations are examined and a simple model problem is proposed. Then the commonly-used second-order Hybrid scheme for diffusive flux calculation is explained. This scheme cannot efficiently be extended to higher-order accuracy, motivating the development
CHAPTER 1. INTRODUCTION

of new finite-volume methods.

A new high-order finite-volume scheme for diffusive flux calculations is discussed in Chapter 3. This reconstruction-based scheme uses the same reconstruction data calculated for the inviscid terms. The description includes a detailed example of the flux integral calculation.

Chapter 4 describes how the boundary conditions are enforced during reconstruction. Two methods are discussed for deriving the equations to constrain the reconstructed solution near the boundary. The method of imposing these constraints is then explained.

The newly developed finite-volume method is validated in Chapter 5. Methodology for measuring the accuracy of the method is described. The method is tested for the interior control volumes on a simple square mesh. Next, several test cases are shown using square, circle and cardioid meshes to prove the scheme works using either cell-centred or vertex-centred control volumes at second-, third- and fourth-order accuracy. A comparison is made between the Taylor Constraints and Gauss Constraints methods for creating the boundary constraint equations. At the end of the chapter an example of a problem with a singularity is discussed.

Finally, in Chapter 6, the control volume finite-volume method is applied to a modified advection-diffusion problem. This scalar model problem has the same structure as the Navier-Stokes' momentum equations, to prove the scheme can be used to evaluate the viscous terms in the Navier-Stokes' equations.

The last chapter summarises the work done and presents conclusions about the developed schemes.
Chapter 2

Flow Solvers

A method is to be developed to extend a high-order accurate inviscid finite-volume flow solver to a high-order accurate viscous flow solver. This chapter summarises how finite-volume methods work in general before going into the inner workings of the existing inviscid flow solver. Next, a model problem for the viscous terms is discussed, since the viscous terms themselves are too complex to study easily. Finally, the commonly used Hybrid finite-volume method is explored for computing high-order fluxes.

2.1 Finite-Volume Methods

Finite-volume methods are based on the fully conservative control volume form of the governing equations being solved. These methods compute control volume averages of the unknowns, rather than computing values at the mesh vertices.

Both time advance and steady state finite-volume methods compute the net flux of the
unknown variables through the boundary of each control volume. The flux integral can be used for both time advance and relaxation schemes. One method to accurately advance the solution in time is to use the flux integral with the fourth-order accurate Runge-Kutta scheme to advance the mean values in the control volumes. For steady-state problems the flux integral can be used to drive a successive over-relaxation process.

The accuracy of the solution hinges on the accuracy of the flux integral. A high-order accurate flux integral in a control volume requires an accurate flux and accurate integration. An accurate flux requires an accurate approximation of the solution in the control volume. One way to achieve this is to reconstruct a polynomial approximation to the solution in the control volume.

Before describing a reconstruction method, two types of control volumes a finite-volume scheme can use to break up an unstructured mesh are described. The finite-volume schemes described in this thesis can use either cell-centred control volumes or vertex-centred control volumes. Cell-centred control volumes are defined as the cells of the mesh. The perimeter of vertex-centred control volumes are defined as lines surrounding a vertex, connecting the midpoint of the triangle edges to the centroid of the cells. Examples of the two control volumes are shown in Figures 2.1 and 2.2. The finite-volume schemes use a local reference point in the control volume as the origin for the reconstruction polynomial. These local reference points are the centroid of the triangle and the encompassed vertex, for the cell-centred and vertex-centred control volumes respectively.

The flow solver approximates the inviscid terms' unknown variables in the control volumes with the cubic Taylor polynomial centered about the control volume’s reference point
Figure 2.1: Cell-Centred Control Volume  Figure 2.2: Vertex-Centred Control Volume

\[ T(x, y) \approx T_c + \frac{\partial T}{\partial x}|_c (x - x_c) + \frac{\partial T}{\partial y}|_c (y - y_c) + \frac{\partial^2 T}{\partial x^2}|_c \frac{(x - x_c)^2}{2} + \]

\[ + \frac{\partial^2 T}{\partial x \partial y}|_c (x - x_c)(y - y_c) + \frac{\partial^2 T}{\partial y^2}|_c \frac{(y - y_c)^2}{2} + \frac{\partial^3 T}{\partial x \partial y^2}|_c \frac{(x - x_c)(y - y_c)^2}{2} + \]

\[ + \frac{\partial^3 T}{\partial x^2 \partial y}|_c \frac{(x - x_c)^2(y - y_c)}{2} + \frac{\partial^3 T}{\partial y^3}|_c \frac{(y - y_c)^3}{6} \]

(2.1)

Here and throughout this thesis, the unknown variable will be represented by the temperature \( T \).

The coefficients of the Taylor polynomial are determined by reconstructing the solution using a least-squares reconstruction method [17, 18]. This reconstruction method is used in this thesis. The coefficients of the Taylor polynomial approximation are chosen to predict the nearby control volume averages well. The Taylor polynomial for the control volume where reconstruction is being done is averaged analytically over near control volumes, and the
CHAPTER 2. FLOW SOLVERS

Figure 2.3: High-order reconstruction with vertex-centred control volumes on a partial mesh. The line along the bottom represent a boundary of the mesh. The neighbouring control volumes used during reconstruction are highlighted for different orders of accuracy for two control volumes.

results are compared with the real averages. The equations are weighted according to how close the nearby control volume is to the one being reconstructed. The closer neighbours are more important (i.e. weighted heavily) than the control volumes that are further away. The number of neighbouring control volumes used to create the system of equations depends on the order of accuracy desired. Figure 2.3 shows the vertex centred-control volumes used for each order of accuracy. As the order of accuracy increases, the neighbouring control volumes
of the lower-orders are also taken into account. For example, for a fourth-order accurate
reconstruction, all the neighbours of the reconstructed control volume, at the bottom of the
mesh shown in Figure 2.3, for second-, third- and fourth-order accuracy are taken into ac-
count. By design, this process creates a system of equations with more equations than there
are unknowns for the desired order of accuracy. The system of equations is over-determined,
which is why a least-squares algorithm is used to solve for the unknown coefficients.

A mean constraint equation is also added to the system before solving, to ensure the
mean value in the reconstructed control volume is conserved — a requirement of finite-
volume schemes. The mean constraint is:

\[
\begin{align*}
\bar{T} &= T_c + I_x \frac{\partial T}{\partial x} |_c + I_y \frac{\partial T}{\partial y} |_c + \frac{I_{xx}}{2} \frac{\partial^2 T}{\partial x^2} |_c + I_{xy} \frac{\partial^2 T}{\partial x \partial y} |_c + \frac{I_{yy}}{2} \frac{\partial^2 T}{\partial y^2} |_c + \\
&+ \frac{I_{xxx}}{6} \frac{\partial^3 T}{\partial x^3} |_c + \frac{I_{xxy}}{2} \frac{\partial^3 T}{\partial x^2 \partial y} |_c + \frac{I_{yy}}{2} \frac{\partial^3 T}{\partial x \partial y^2} |_c + \frac{I_{yyy}}{6} \frac{\partial^3 T}{\partial y^3} |_c
\end{align*}
\]

(2.2)

where \( T \) is the unknown value and the \( I \)'s are the control volume's moments, for example:

\[
I_{xxy} = \frac{1}{A} \int_{cv} x^2 y \mathrm{d}A 
\]

(2.3)

where \( A \) is the area of the control volume.

The least-squares matrix contains the mean constraint equation, but this alone does not
ensure the mean value in the control volume will be conserved. For the mean constraint to
be satisfied exactly, the first column of the matrix is reduced using Gaussian elimination
with the mean constraint equation. This ensures the mean constraint is enforced during
reconstruction.

The least-squares algorithm is then used on the column-reduced matrix to solve for the
unknown coefficients. The Taylor polynomial with the reconstructed coefficients gives an
approximation to the solution in the control volume. This process is repeated for each control volume in the mesh.

Flux integrals are formed from the Taylor polynomial approximation of the solutions in each control volume. The flux integrals are evaluated using Gauss quadrature [21]. Gauss quadrature gives the capability of evaluating a definite integral to high-order accuracy with the integrand evaluated at only a few points.

An alternate approach to reconstruction is the essentially non-oscillatory (ENO) scheme, which ensures that all control volumes with a smooth neighbourhood have a uniformly high-order accurate reconstruction. Some ENO schemes search for the smoothest stencil for reconstruction in each control volume [1, 6, 22]. These schemes can have difficulty with steady-state calculations. To overcome these problems weighted ENO (WENO) schemes were developed [7, 10]. Instead of selecting the smoothest stencil, they use a weighted sum of all the stencils. ENO reconstruction is compatible with the method developed in this thesis for evaluating viscous terms.

In summary, the general steps for finite-volume methods are as follows:

1. Reconstruct a high-order approximation to the solution, based on the current mean values in local control volumes.

2. Compute the high-order flux integral through the boundaries of the control volumes.

3. Use a time advance or relaxation scheme to advance the current solution with the flux integral.

4. Repeat all steps until desired time or steady-state solution is reached.
2.2 Inviscid Flow Solver

Dr. Carl F. Ollivier-Gooch has developed a fourth-order accurate finite-volume flow solver for the inviscid terms of the compressible Navier-Stokes' equations:

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.4}
\]

\[
\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla P + F_B \tag{2.5}
\]

\[
\frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u} (E + P)) = 0 \tag{2.6}
\]

where \((\mathbf{u} \otimes \mathbf{u})_{ij} = u_i u_j\)

\[F_B = \text{net vector body force for the control volume}\]

In inviscid compressible flow, the only flux normal to a boundary wall is the flux due to pressure. The pressure is obtained by extrapolation and used explicitly in the flux calculation at the boundary wall. This is all that is needed for boundary conditions at a wall in inviscid compressible flow. Inflow and outflow boundary conditions are based on characteristic theory. Roe's scheme [20] is used to compute the flux at the control volume boundary to match the compressible inviscid physics.

The net fluxes are then computed, and an appropriate time advance scheme is used to update the solution, completing a single time step.

In addition to upwind methods, such as the one used in the existing inviscid flow solver [3, 4, 5, 17, 18], there are other successful methods based on centred differences [12, 13, 15, 16]. Centred difference codes do not perform reconstructions, so the method for viscous discretisation presented here will not fit into that context.
2.3 Viscous Modelling

A high-order viscous flow solver is to be developed by extending the methods in the existing inviscid flow solver to include the viscous terms of the governing equations. The full viscous compressible Navier-Stokes’ equations can be written as given in equations 2.7 to 2.9. The boxed terms in equations 2.8 and 2.9 represent the viscous terms to be evaluated.

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.7)
\]

\[
\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla P + [\nabla \cdot \mathbf{\tau}] + F_B \tag{2.8}
\]

\[
\frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u} (E + P)) = [\nabla \cdot (\mathbf{\tau} \cdot \mathbf{u})] + [\nabla \cdot (\kappa \nabla T)] \tag{2.9}
\]

where \((\mathbf{u} \otimes \mathbf{u})_{ij} = u_i u_j\)

\[
\tau_{ij} = 2\mu \left(D_{ij} - \frac{1}{3} \delta_{ij} D_{mm}\right)
\]

\[
D = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \quad \text{or} \quad D_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)
\]

\[
F_B = \text{net vector body force for the control volume}
\]

The viscous terms themselves are too complex to study easily since it is a coupled system involving two unknowns, so a model equation is used. As a model problem the Laplace equation for temperature is used with no forcing term:

\[
\nabla^2 T = 0 \quad (2.10)
\]

This equation is integrated over each control volume in the mesh and the Divergence Theorem is applied to obtain a flux integral:

\[
\int_{\partial (CV)} \nabla T \cdot \hat{n} \, ds = 0 \quad (2.11)
\]
where \( \hat{n} \) is the unit normal vector to the shared control volume edge.

To solve the homogeneous Laplace equation to high-order accuracy, the flux integral must be calculated to high-order accuracy. Because the integrand contains the temperature gradient, \( \nabla T \) must be known to high-order accuracy.

The core of this thesis describes how to calculate the flux integral \( \int_{\partial(CV)} \nabla T \cdot \hat{n} \, ds \) and its boundary conditions to high-order accuracy. The types of boundary conditions discussed are Dirichlet alone, Neumann alone and linear combinations of Dirichlet and Neumann.

### 2.4 Hybrid Scheme

Before discussing the new high-order method, this section looks at one of the most common finite-volume methods for discretising the Laplace equation. Such a method is the Hybrid scheme, so named because the control volumes are vertex-centred and the gradients are calculated over the cells of the mesh. This scheme calculates the flux through vertex-centred control volumes.

Many researchers use essentially this scheme, often for computing viscous terms of the Navier-Stokes' equations, using either finite-element or finite-volume methods. The finite-volume community seems to use second-order accuracy exclusively \([2, 13, 14, 15, 16]\), and even finite-element researchers do not typically exploit high-order accuracy \([8, 9, 11, 19]\).

For solving the Laplace equation, as shown in the previous section, the temperature gradient is needed to calculate the flux through the control volume boundary. The Hybrid scheme approximates the gradient of the solution on the triangles of the mesh and uses it
CHAPTER 2. FLOW SOLVERS

to calculate the flux through the control volume edges inside the triangle.

The gradient is approximated by applying Green's Theorem:

$$\int_{CV} \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) \, dA = \oint_{\partial(CV)} (P \, dx + Q \, dy)$$  \hspace{1cm} (2.12)

By letting both $Q$ and $P$ be equal to temperature $T$, equation 2.12 can be broken up into:

$$\int_{CV} \frac{\partial T}{\partial x} \, dA = \oint_{\partial(CV)} T \, dy \quad \quad - \int_{CV} \frac{\partial T}{\partial y} \, dA = \oint_{\partial(CV)} T \, dx$$ \hspace{1cm} (2.13)

The temperature gradient is assumed constant throughout the control volume, which causes a second-order error. This assumption allows the integral equations in 2.13 to be written as:

$$\frac{\partial T}{\partial x} = \frac{1}{A} \oint_{\partial(CV)} T \, dy \quad \quad \frac{\partial T}{\partial y} = -\frac{1}{A} \oint_{\partial(CV)} T \, dx$$ \hspace{1cm} (2.14)

These equations provide the needed temperature gradient for the flux calculation. The vertices and the area of the triangle are known. In order to evaluate the contour integrals, the Hybrid scheme assumes the control volume average value applies at the respective vertex, meaning the temperature located at vertex 0 is equal to the mean temperature $T_0$. This assumption also causes a second-order error, since the vertex is not located at the centre of the control volume. Using the assumption, the temperatures at each of the triangle vertices
are known. With the equations in 2.14, the gradient can be evaluated as:

\[
\frac{\partial T}{\partial x} = \frac{1}{A} \left( \int_{y_0}^{y_1} T \, dy + \int_{y_1}^{y_2} T \, dy + \int_{y_2}^{y_0} T \, dy \right) \\
= \frac{1}{A} \left[ (\bar{T}_1 - \bar{T}_0) (y_1 - y_0) + (\bar{T}_2 - \bar{T}_1) (y_2 - y_1) + (\bar{T}_0 - \bar{T}_2) (y_0 - y_2) \right] \tag{2.15}
\]

\[
\frac{\partial T}{\partial y} = -\frac{1}{A} \left( \int_{y_0}^{y_1} T \, dx + \int_{y_1}^{y_2} T \, dx + \int_{y_2}^{y_0} T \, dx \right) \\
= -\frac{1}{A} \left[ (\bar{T}_1 - \bar{T}_0) (x_1 - x_0) + (\bar{T}_2 - \bar{T}_1) (x_2 - x_1) + (\bar{T}_0 - \bar{T}_2) (x_0 - x_2) \right] \tag{2.16}
\]

With the temperature gradient known for the triangle, the flux through the edges between the control volumes located inside the triangle is calculated, using the flux integral. Because the gradient is constant, the flux integral is simply the dot product of the gradient and the unit normal to the shared edge, multiplied by the length of the shared control volume edge.

Consider the situation in Figure 2.4. The gradient in the shaded triangle is approximated to second-order accuracy using equations 2.15 and 2.16. This gradient is then used in
evaluating the flux integral through the shared edge of the vertex-centred control volumes inside the triangle. The fluxes are then subtracted from the source control volume, and added to the destination control volume, as determined by the direction of the normal to the edge used in the flux integral calculation. In Figure 2.4 the flux calculated across the upper edge is subtracted from the current total flux for control volume 0 and added to the current total flux for control volume 2. Similarly the total fluxes for control volumes along the remaining two edges are adjusted.

Once this process has been repeated for each cell in the mesh, the flux integral for each control volume is known. The fluxes are then used with an appropriate time advance scheme to update the mean values in their corresponding control volumes.

Improvement of the accuracy of this scheme requires high-order gradients, which implies a reconstruction over the triangles. This means if the high-order extension of the Hybrid method were applied to solving the Navier-Stokes' equations, two reconstructions would need to be performed: one over the vertex-centred control volumes for the inviscid terms and one over triangles for the viscous terms. For this reason attempting to improve the accuracy of the Hybrid scheme efficiently would be difficult. It was decided to develop other high-order schemes, which would share the reconstructed solution from the inviscid terms and thus be more efficient.
Chapter 3

High-Order Accurate Gradient Approximation and Flux Integration

This chapter introduces two reconstruction-based schemes that use the Taylor polynomial coefficients, found during reconstruction, to calculate the viscous fluxes through the control volume boundaries. A high-order accurate approximation to the flux integral \( \int_{\partial(CV)} \nabla T \cdot \hat{n} \, ds \) requires a high-order accurate gradient. The existing reconstruction scheme in the inviscid flow solver finds a cubic Taylor polynomial approximation to the solution expanded about the reference point of the control volumes.

\[
T(x, y) \approx T|_c + \frac{\partial T}{\partial x}|_c (x - x_c) + \frac{\partial T}{\partial y}|_c (y - y_c) + \frac{\partial^2 T}{\partial x^2}|_c \frac{(x - x_c)^2}{2} + \\
\frac{\partial^2 T}{\partial x \partial y}|_c \frac{(x - x_c)(y - y_c)}{2} + \frac{\partial^2 T}{\partial y^2}|_c \frac{(y - y_c)^2}{2} + \frac{\partial^3 T}{\partial x^3}|_c \frac{(x - x_c)^3}{6} + \\
\frac{\partial^3 T}{\partial x^2 \partial y}|_c \frac{(x - x_c)^2(y - y_c)}{2} + \frac{\partial^3 T}{\partial x \partial y^2}|_c \frac{(x - x_c)(y - y_c)^2}{2} + \\
\frac{\partial^3 T}{\partial y^3}|_c \frac{(y - y_c)^3}{6} \tag{3.1}
\]
This equation is the basis for obtaining an approximation to the gradient of the solution. This chapter describes how the gradient is approximated and how the flux integral is calculated for both cell-centred and vertex-centred control volumes.

3.1 Gradient Approximation

The solution gradient is approximated in the control volume by taking the gradient of the high-order Taylor polynomial. The gradient of the cubic Taylor polynomial (equation 3.1) gives the gradient of the solution within each control volume:

\[ \nabla T(x, y) \approx \left( \frac{\partial T}{\partial x} \bigg|_c + \frac{\partial^2 T}{\partial x^2} \bigg|_c (x - x_c) + \frac{\partial^2 T}{\partial x \partial y} \bigg|_c (y - y_c) + \frac{\partial^3 T}{\partial x^3} \bigg|_c (x - x_c)^2 \right) + \\
\frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c (x - x_c) (y - y_c) + \frac{\partial^3 T}{\partial x \partial y^2} \bigg|_c (y - y_c)^2, \]

\[ + \left( \frac{\partial T}{\partial y} \bigg|_c + \frac{\partial^2 T}{\partial x \partial y} \bigg|_c (x - x_c) + \frac{\partial^2 T}{\partial y^2} \bigg|_c (y - y_c) + \frac{\partial^3 T}{\partial x \partial y^2} \bigg|_c (x - x_c)^2 \right) + \\
\frac{\partial^3 T}{\partial x^2 \partial y^2} \bigg|_c (x - x_c) (y - y_c) + \frac{\partial^3 T}{\partial y^3} \bigg|_c (y - y_c)^2 \right) \]

(3.2)

3.2 Flux Integration: Cell-Centred Control Volumes

In this section, an example will be used to describe the method by which the flux integral is calculated for cell-centred control volumes. It is assumed that fourth-order accuracy is desired.

Consider the two interior control volumes shown in Figure 3.1. The previous section described how the gradient is approximated in each control volume. The gradients used
along edge $\overline{ab}$ in the flux integrals $\oint_{\partial(CV0)} \nabla T \cdot \hat{n} \, ds$ and $\oint_{\partial(CV1)} \nabla T \cdot \hat{n} \, ds$ must be identical. This requirement comes from global conservation: the flux leaving a control volume must enter its neighbour. However, along the shared edge between two control volumes there are two gradients, which are not in general the same, as can be seen in the one dimensional example in Figure 3.2. The curves represent the cubic Taylor polynomial approximations and the straight lines are the gradients at the shared edge for each control volume. The flux through this edge is calculated by taking the average of the two gradients, weighted by the control volumes’ areas:

$$\nabla T = \frac{A_0 \nabla T_0 + A_1 \nabla T_1}{A_0 + A_1}$$

If this is done along the shared edge in the Figure 3.2 example, the result would be as shown in Figure 3.3. With the gradient along the edge known, and the unit normal to the shared edge given by the geometry of the mesh, the flux integral can be evaluated.

The flux integral is evaluated to fourth-order accuracy using Gauss quadrature. Ob-
haining fourth-order accuracy implies that a cubic polynomial must be integrated exactly.
To do this using Gauss quadrature requires two Gauss points for every straight edge to be
integrated. For the shared edge between the cell-centred control volumes in Figure 3.1 the
integral is broken up into two equal length segments, each with a specified Gauss point,
normal and weight. The integral can then be evaluated as:

\[
\text{Flux} = (\nabla T(x_{g1}) \cdot \hat{n}_{g1}) W_{g1} + (\nabla T(x_{g2}) \cdot \hat{n}_{g2}) W_{g2}
\]

(3.4)

where \(x_{g1}\) and \(x_{g2}\) are the Gauss point locations along the shared edge and \(W_{g1}\) and \(W_{g2}\)
are the corresponding Gauss weights.

For fourth-order accuracy, the Gauss points are given by equations 3.5 and 3.6. The
Gauss weights are simply half the length of the edge. For second- and third-order accuracy
only one Gauss point is needed, located at the midpoint of the edge with a corresponding
Gauss weight equal to the length of the edge. The normal is the unit normal of the edge,
CHAPTER 3. HIGH-ORDER FLUX INTEGRALS

given in equation 3.7.

\[ x_{g1} = \frac{x_b + x_a}{2} + \frac{x_b - x_a}{2\sqrt{3}} \]  
(3.5)

\[ x_{g2} = \frac{x_b + x_a}{2} - \frac{x_b - x_a}{2\sqrt{3}} \]  
(3.6)

\[ \hat{n}_{g1} = \hat{n}_{g2} = \frac{(y_a - y_b, x_b - x_a)}{|x_a - x_b|} \]  
(3.7)

The flux is then subtracted from the flux integral for control volume 0 and added to the flux integral for control volume 1 (following the direction of the normal). This process is repeated for each internal edge of the mesh.

For the boundary edges the flux integration process is the same except the gradient does not need to be averaged, and the fluxes only adjust the boundary control volumes' flux integral. The method for boundary condition enforcement is discussed in Chapter 4.

3.3 Flux Integration: Vertex-Centred Control Volumes

The only difference when applying this scheme to vertex-centred control volumes is the number and location of the Gauss points. Figure 3.4 shows an example of a shared edge between two control volumes with the normals and Gauss points on the edge for second-, third- and fourth-order accuracy. The formulae for the Gauss points, weights and normals,
for the different orders of accuracy are given here:

\[ L_1 = |x_a - x_b| \]

\[ L_2 = |x_b - x_c| \]

\[ x_g = \frac{\frac{L_1 x_a + x_b}{2} + \frac{L_2 x_b + x_c}{2}}{L_1 + L_2} \] \hspace{1cm} (3.8)

\[ W_g = |x_a - x_c| \] \hspace{1cm} (3.9)

\[ \hat{n}_g = \frac{(y_c - y_a, x_a - x_c)}{W_g} \] \hspace{1cm} (3.10)

\[ x_{g1} = \frac{x_a + x_b}{2} \] \hspace{1cm} (3.11)

\[ x_{g2} = \frac{x_b + x_c}{2} \] \hspace{1cm} (3.12)

\[ W_{g1} = L_1 \] \hspace{1cm} (3.13)

\[ W_{g2} = L_2 \] \hspace{1cm} (3.14)

\[ \hat{n}_{g1} = \frac{(y_b - y_a, x_a - x_b)}{L_1} \] \hspace{1cm} (3.15)

\[ \hat{n}_{g2} = \frac{(y_c - y_b, x_b - x_c)}{L_2} \] \hspace{1cm} (3.16)

\[ x_{g1} = \frac{x_a + x_b}{2} + \frac{x_a - x_b}{2\sqrt{3}} \] \hspace{1cm} (3.17)

\[ x_{g2} = \frac{x_a + x_b}{2} - \frac{x_a - x_b}{2\sqrt{3}} \] \hspace{1cm} (3.18)

\[ x_{g3} = \frac{x_b + x_c}{2} + \frac{x_b - x_c}{2\sqrt{3}} \] \hspace{1cm} (3.19)

\[ x_{g4} = \frac{x_b + x_c}{2} - \frac{x_b - x_c}{2\sqrt{3}} \] \hspace{1cm} (3.20)

\[ W_{g1} = W_{g2} = \frac{L_1}{2} \] \hspace{1cm} (3.21)

\[ W_{g3} = W_{g4} = \frac{L_2}{2} \] \hspace{1cm} (3.22)

\[ \hat{n}_{g1} = \hat{n}_{g2} = \frac{(y_b - y_a, x_a - x_b)}{L_1} \] \hspace{1cm} (3.23)

\[ \hat{n}_{g3} = \hat{n}_{g4} = \frac{(y_c - y_b, x_b - x_c)}{L_2} \] \hspace{1cm} (3.24)
CHAPTER 3. HIGH-ORDER FLUX INTEGRALS

Figure 3.4: Vertex-Centred Gauss Points

2nd-order: 1 Gauss Point

3rd-order: 2 Gauss Points

4th-order: 4 Gauss Points

Figure 3.4: Vertex-Centred Gauss Points
Chapter 4

Boundary Conditions

For the scheme described in Chapter 3 to be completely high-order accurate, the high-order reconstruction in boundary control volumes must be consistent with the boundary condition. That is, the Taylor approximation of the solution in the control volume must be constrained to satisfy the boundary condition. These boundary constraints are in addition to the mean value constraint for each control volume.

Most of this chapter deals with the derivation of the boundary constraints. Section 4.1 describes Taylor Constraints, which combine the Taylor polynomial approximation of the solution in the control volume with the Taylor polynomial approximation of the boundary condition to create appropriate constraints for Dirichlet boundary conditions, Neumann boundary conditions and a linear combination of Dirichlet and Neumann boundary conditions. Unfortunately there are a number of drawbacks to the Taylor Constraints approach. Section 4.2 describes a different and simpler way of creating the boundary constraints called Gauss Constraints. Whereas the Taylor Constraints method enforces the boundary con-
dition all along the boundary edge of the control volume, the Gauss Constraints method enforces it only at the Gauss points where the solution gradient is needed for the Gauss quadrature flux integral calculation. This method is also derived for Dirichlet boundary conditions, Neumann boundary conditions and a linear combination of Dirichlet and Neumann boundary conditions. Finally, the last section describes how the constraints are added to the least-squares matrix, turning it into a constrained least-squares problem.

4.1 Taylor Constraints

The Taylor Constraints method was the first attempt at enforcing the boundary conditions in the reconstruction over the control volume. The constraints are created using a Taylor polynomial to approximate the boundary condition. This Taylor polynomial is matched term by term to the Taylor polynomial approximation of the solution in the control volume evaluated along the boundary edge, giving a set of constraints on the unknown coefficients of the Taylor polynomial approximation of the solution. The following derivations are for an arbitrary straight boundary edge, where the boundary condition is a function of the variable $s$ along the edge.

4.1.1 Dirichlet Boundary Conditions

The Dirichlet boundary condition is given by $T(s) = f(s)$, where $f$ is $C^3$. The boundary function $f$ is replaced with a cubic Taylor polynomial about the midpoint of the edge $s_m$:

$$f(s) \approx f|_m + \frac{\partial f}{\partial s}|_m (s - s_m) + \frac{\partial^2 f}{\partial s^2}|_m \frac{(s - s_m)^2}{2} + \frac{\partial^3 f}{\partial s^3}|_m \frac{(s - s_m)^3}{6}$$  \hfill (4.1)
CHAPTER 4. BOUNDARY CONDITIONS

Now a similar approximation is needed for the solution function, so the terms in the two equations can be matched to form the boundary constraints. The Taylor expansion of the solution must be in terms of Taylor coefficients at the reconstruction reference point of the control volume, because these are the terms solved for during reconstruction.

The fourth-order accurate Taylor polynomial approximation of the solution about the reference point of the control volume is given again in equation 4.2. The idea is to rewrite this equation, evaluated along the boundary, into a Taylor polynomial about the midpoint of the boundary edge, while keeping the desired derivatives at the reference point of the control volume.

\[
T(x, y) \approx T(x, y) \approx T_c + \frac{\partial T}{\partial x}
|_{c} (x - x_c) + \frac{\partial T}{\partial y}
|_{c} (y - y_c) + \frac{\partial^2 T}{\partial x^2}
|_{c} \frac{(x - x_c)^2}{2} + \\
\frac{\partial^2 T}{\partial x \partial y}
|_{c} (x - x_c)(y - y_c) + \frac{\partial^2 T}{\partial y^2}
|_{c} \frac{(y - y_c)^2}{2} + \frac{\partial^3 T}{\partial x^3}
|_{c} \frac{(x - x_c)^3}{6} + \\
\frac{\partial^3 T}{\partial x^2 \partial y}
|_{c} \frac{(x - x_c)^2(y - y_c)}{2} + \frac{\partial^3 T}{\partial x \partial y^2}
|_{c} \frac{(x - x_c)(y - y_c)^2}{2} + \\
\frac{\partial^3 T}{\partial y^3}
|_{c} \frac{(y - y_c)^3}{6}
\]  

(4.2)

To evaluate this polynomial along the boundary, \(x\) and \(y\) in equation 4.2 are replaced by:

\[
x(s) = x_m + (s - s_m) \left( \frac{x_b - x_a}{s_b - s_a} \right) \\
= x_m + (s - s_m) D_x 
\]

(4.3)

\[
y(s) = y_m + (s - s_m) \left( \frac{y_b - y_a}{s_b - s_a} \right) \\
= y_m + (s - s_m) D_y 
\]

(4.4)

where \((x_a, y_a)\) and \((x_b, y_b)\) are the endpoints of the boundary edge in Cartesian coordinates.
and $s_a$ and $s_b$ are the endpoints in the boundary coordinate. After considerable manipulation
the solution on the boundary can be written as:

\[
T(s) \approx \left[ T_c + \frac{\partial T}{\partial x} \bigg|_c (x_m - x_c) + \frac{\partial T}{\partial y} \bigg|_c (y_m - y_c) + \\
\frac{\partial^2 T}{\partial x^2} \bigg|_c \frac{(x_m - x_c)^2}{2} + \frac{\partial^2 T}{\partial x \partial y} \bigg|_c (x_m - x_c)(y_m - y_c) + \frac{\partial^2 T}{\partial y^2} \bigg|_c \frac{(y_m - y_c)^2}{2} + \\
\frac{\partial^3 T}{\partial x^3} \bigg|_c \frac{(x_m - x_c)^3}{6} + \frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c \frac{(x_m - x_c)^2(y_m - y_c)}{2} + \\
\frac{\partial^3 T}{\partial x \partial y^2} \bigg|_c \frac{(x_m - x_c)(y_m - y_c)^2}{2} + \frac{\partial^3 T}{\partial y^3} \bigg|_c \frac{(y_m - y_c)^3}{6} \right] + \\
\left[ \frac{\partial T}{\partial x} \bigg|_c D_x + \frac{\partial T}{\partial y} \bigg|_c D_y + \\
\frac{\partial^2 T}{\partial x^2} \bigg|_c D_x (x_m - x_c) + \frac{\partial^2 T}{\partial x \partial y} \bigg|_c (D_y (x_m - x_c) + D_x (y_m - y_c)) + \\
\frac{\partial^2 T}{\partial y^2} \bigg|_c D_y (y_m - y_c) + \frac{\partial^3 T}{\partial x^3} \bigg|_c D_x \frac{(x_m - x_c)^2}{2} + \\
\frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c \left(D_x (x_m - x_c)(y_m - y_c) + D_y \frac{(x_m - x_c)^2}{2}\right) + \\
\frac{\partial^3 T}{\partial x \partial y^2} \bigg|_c \left(D_y (x_m - x_c)(y_m - y_c) + D_x \frac{(y_m - y_c)^2}{2}\right) + \\
\frac{\partial^3 T}{\partial y^3} \bigg|_c D_y \frac{(y_m - y_c)^3}{2} \right] (s - s_m) + \\
\left[ \frac{\partial^2 T}{\partial x^2} \bigg|_c D_x^2 + \frac{\partial^3 T}{\partial x \partial y} \bigg|_c 2D_x D_y + \frac{\partial^3 T}{\partial y^2} \bigg|_c D_y^2 \right] + \\
\frac{\partial^2 T}{\partial x^3} \bigg|_c D_x^2 (x_m - x_c) + \frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c (2D_x D_y (x_m - x_c) + D_x^2 (y_m - y_c)) + \\
\frac{\partial^3 T}{\partial x \partial y^2} \bigg|_c \left(2D_x D_y (y_m - y_c) + D_y^2 (x_m - x_c)\right) + \\
\frac{\partial^3 T}{\partial y^3} \bigg|_c D_y^2 (y_m - y_c) \right] \frac{(s - s_m)^2}{2} + \\
\left[ \frac{\partial^3 T}{\partial x^3} \bigg|_c D_x^3 + \frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c 3D_x D_y + \frac{\partial^3 T}{\partial y^3} \bigg|_c \frac{D_y^3}{6} \right] (s - s_m)^3
\]  

(4.5)
CHAPTER 4. BOUNDARY CONDITIONS

Now the boundary constraints can be determined by matching up the terms in equation 4.1 with the terms in equation 4.5. The full Dirichlet Taylor constraints for any straight boundary are:

\[
\left. f \right|_m = T\bigg|_c + \frac{\partial T}{\partial x}\bigg|_c (x_m - x_c) + \frac{\partial T}{\partial y}\bigg|_c (y_m - y_c) + \frac{\partial^2 T}{\partial x^2}\bigg|_c \frac{(x_m - x_c)^2}{2} + \\
+ \frac{\partial^2 T}{\partial x\partial y}\bigg|_c (x_m - x_c) (y_m - y_c) + \frac{\partial^2 T}{\partial y^2}\bigg|_c \frac{(y_m - y_c)^2}{2} + \\
+ \frac{\partial^3 T}{\partial x^3}\bigg|_c \frac{(x_m - x_c)^3}{6} + \frac{\partial^3 T}{\partial x^2\partial y}\bigg|_c \frac{(x_m - x_c)^2 (y_m - y_c)}{2} + \\
+ \frac{\partial^3 T}{\partial x\partial y^2}\bigg|_c \frac{(x_m - x_c)^2}{2} + \frac{\partial^3 T}{\partial y^3}\bigg|_c \frac{(y_m - y_c)^3}{6} \tag{4.6}
\]

\[
\left. \frac{\partial f}{\partial s} \right|_m = \left. \frac{\partial T}{\partial x} \right|_c D_x + \left. \frac{\partial T}{\partial y} \right|_c D_y + \left. \frac{\partial^2 T}{\partial x^2} \right|_c D_x (x_m - x_c) + \\
+ \left. \frac{\partial^2 T}{\partial x\partial y} \right|_c (D_y (x_m - x_c) + D_x (y_m - y_c)) + \left. \frac{\partial^2 T}{\partial y^2} \right|_c D_y (y_m - y_c) + \\
+ \left. \frac{\partial^3 T}{\partial x^3} \right|_c D_x \frac{(x_m - x_c)^2}{2} + \\
+ \left. \frac{\partial^3 T}{\partial x^2\partial y} \right|_c \left( D_x (x_m - x_c) (y_m - y_c) + D_y \frac{(x_m - x_c)^2}{2} \right) + \\
+ \left. \frac{\partial^3 T}{\partial x\partial y^2} \right|_c \left( D_y (x_m - x_c) (y_m - y_c) + D_x \frac{(y_m - y_c)^2}{2} \right) + \\
+ \left. \frac{\partial^3 T}{\partial y^3} \right|_c D_y \frac{(y_m - y_c)^2}{2} \tag{4.7}
\]

\[
\left. \frac{\partial^2 f}{\partial s^2} \right|_m = \left. \frac{\partial^2 T}{\partial x^2} \right|_c D_x^2 + \left. \frac{\partial^2 T}{\partial x\partial y} \right|_c 2D_x D_y + \left. \frac{\partial^2 T}{\partial y^2} \right|_c D_y^2 + \left. \frac{\partial^3 T}{\partial x^3} \right|_c D_x^2 (x_m - x_c) + \\
+ \left. \frac{\partial^3 T}{\partial x^2\partial y} \right|_c (2D_x D_y (x_m - x_c) + D_y^2 (y_m - y_c)) + \\
+ \left. \frac{\partial^3 T}{\partial x\partial y^2} \right|_c (2D_x D_y (y_m - y_c) + D_y^2 (x_m - x_c)) + \left. \frac{\partial^3 T}{\partial y^3} \right|_c D_y^2 (y_m - y_c) \tag{4.8}
\]

\[
\left. \frac{\partial^3 f}{\partial s^3} \right|_m = \left. \frac{\partial^3 T}{\partial x^3} \right|_c D_x^3 + \left. \frac{\partial^3 T}{\partial x^2\partial y} \right|_c 3D_x^2 D_y + \left. \frac{\partial^3 T}{\partial x\partial y^2} \right|_c 3D_x D_y^2 + \left. \frac{\partial^3 T}{\partial y^3} \right|_c D_y^3 \tag{4.9}
\]

The constraints shown above are greatly simplified if the boundary edge is parallel to either the x- or y-axis. For example, if the boundary edge were parallel to the x-axis, \( y_b - y_a \)
would be zero, meaning that $D_y = 0$, and the resulting boundary constraints are:

$$f|_m = \left. T\right|_c + \left. \frac{\partial T}{\partial x}\right|_c (x_m - x_c) + \left. \frac{\partial T}{\partial y}\right|_c (y_m - y_c) + \frac{\partial^2 T}{\partial x^2}\left|_c \right. \frac{(x_m - x_c)^2}{2} + \frac{\partial^2 T}{\partial x \partial y}\left|_c \right. (x_m - x_c) (y_m - y_c) + \frac{\partial^2 T}{\partial y^2}\left|_c \right. \frac{(y_m - y_c)^2}{2} + \frac{\partial^3 T}{\partial x^3}\left|_c \right. \frac{(x_m - x_c)^3}{6} + \frac{\partial^3 T}{\partial x^2 \partial y}\left|_c \right. \frac{(x_m - x_c)^2 (y_m - y_c)}{2} + \frac{\partial^3 T}{\partial x \partial y^2}\left|_c \right. \frac{(x_m - x_c) (y_m - y_c)^2}{2} \frac{\partial^3 T}{\partial y^3}\left|_c \right. \frac{(y_m - y_c)^3}{6}$$ (4.10)

$$\frac{\partial f}{\partial s}|_m = \left. \frac{\partial T}{\partial x}\right|_c \left. D_x\right|_c + \left. \frac{\partial^2 T}{\partial x^2}\right|_c \left. D_x\right|_c (x_m - x_c) + \left. \frac{\partial^2 T}{\partial x \partial y}\right|_c \left. D_x\right|_c (y_m - y_c) + \frac{\partial^3 T}{\partial x^3}\left|_c \right. \left. D_x\right|_c \frac{(x_m - x_c)^2}{2} + \frac{\partial^3 T}{\partial x^2 \partial y}\left|_c \right. \left. D_x\right|_c (x_m - x_c) (y_m - y_c) + \frac{\partial^3 T}{\partial x \partial y^2}\left|_c \right. \left. D_x\right|_c \frac{(y_m - y_c)^2}{2}$$ (4.11)

$$\frac{\partial^2 f}{\partial s^2}|_m = \left. \frac{\partial^2 T}{\partial x^2}\right|_c \left. D_x^2\right|_c + \left. \frac{\partial^3 T}{\partial x^3}\right|_c \left. D_x^2\right|_c (x_m - x_c) + \left. \frac{\partial^3 T}{\partial x^2 \partial y}\right|_c \left. D_x^2\right|_c (y_m - y_c)$$ (4.12)

$$\frac{\partial^3 f}{\partial s^3}|_m = \left. \frac{\partial^3 T}{\partial x^3}\right|_c \left. D_x^3\right|_c$$ (4.13)

### 4.1.2 Neumann Boundary Conditions

The Neumann boundary condition $\frac{\partial T(s)}{\partial n_m} = g(s)$, where $g$ is $C^3$, is dealt with in much the same manner as the Dirichlet boundary condition. However, a little extra work is required in dealing with the normal derivative. Applying the definition of the directional derivative, with the knowledge that $x = x(s)$, the Neumann boundary condition is rewritten as:

$$\frac{\partial T}{\partial x} \hat{n}_x + \frac{\partial T}{\partial y} \hat{n}_y = g(s)$$ (4.14)

The boundary function $g$ is written using the quadratic Taylor polynomial about the
CHAPTER 4. BOUNDARY CONDITIONS

midpoint of the boundary edge:

\[ g(s) \approx g_m + \frac{\partial g}{\partial s}|_m (s - s_m) + \frac{\partial^2 g}{\partial s^2}|_m \frac{(s - s_m)^2}{2} \]  \hspace{1cm} (4.15)

This is done so the terms can be compared with the terms in the Taylor polynomial approximation to the left-hand side of equation 4.14, which is also a quadratic function.

The spatial derivatives are approximated with a Taylor polynomial. The Taylor polynomial is obtained by taking the spatial partial derivatives of equation 4.2. The derivative approximations are:

\[
\frac{\partial T_{bdry}(x, y)}{\partial x} \approx \frac{\partial T}{\partial x}|_c + \frac{\partial^2 T}{\partial x^2}|_c (x - x_c) + \frac{\partial^2 T}{\partial x \partial y}|_c (y - y_c) + \frac{\partial^3 T}{\partial x^3}|_c \frac{(x - x_c)^2}{2} + \frac{\partial^3 T}{\partial x^2 \partial y}|_c (x - x_c) (y - y_c) + \frac{\partial^3 T}{\partial x \partial y^2}|_c \frac{(y - y_c)^2}{2} \]  \hspace{1cm} (4.16)

\[
\frac{\partial T_{bdry}(x, y)}{\partial y} \approx \frac{\partial T}{\partial y}|_c + \frac{\partial^2 T}{\partial x \partial y}|_c (x - x_c) + \frac{\partial^2 T}{\partial y^2}|_c (y - y_c) + \frac{\partial^3 T}{\partial x^2 \partial y}|_c \frac{(x - x_c)^2}{2} + \frac{\partial^3 T}{\partial x \partial y^2}|_c (x - x_c) (y - y_c) + \frac{\partial^3 T}{\partial y^3}|_c \frac{(y - y_c)^2}{2} \]  \hspace{1cm} (4.17)

To convert this to a polynomial containing \((s - s_m)\) terms, the substitutions shown in equations 4.3 and 4.4 are used in equations 4.16 and 4.17. After considerable manipulation
the solution on the boundary can be written as:

\[
\frac{\partial T(x,y)}{\partial x} \approx \left[ \frac{\partial T}{\partial x} \right]_c + \left[ \frac{\partial^2 T}{\partial x^2} \right]_c (x_m - x_c) + \left[ \frac{\partial^2 T}{\partial x \partial y} \right]_c (y_m - y_c) +
\]

\[
\left[ \frac{\partial^3 T}{\partial x^3} \right]_c \frac{(x_m - x_c)^2}{2} + \left[ \frac{\partial^3 T}{\partial x^2 \partial y} \right]_c (x_m - x_c)(y_m - y_c) + \left[ \frac{\partial^3 T}{\partial x \partial y^2} \right]_c \frac{(y_m - y_c)^2}{2} +
\]

\[
\left[ \frac{\partial^2 T}{\partial x^2} \right]_c D_x + \left[ \frac{\partial^2 T}{\partial x \partial y} \right]_c D_y + \left[ \frac{\partial^3 T}{\partial x^2 \partial y} \right]_c (x_m - x_c) D_x +
\]

\[
\left[ \frac{\partial^3 T}{\partial x \partial y} \right]_c \frac{(x_m - x_c)^2}{2} + \left[ \frac{\partial^3 T}{\partial x \partial y^2} \right]_c (x_m - x_c)(y_m - y_c) + \left[ \frac{\partial^3 T}{\partial y^3} \right]_c \frac{(y_m - y_c)^2}{2} +
\]

\[
\left[ \frac{\partial^2 T}{\partial x \partial y} \right]_c D_x + \left[ \frac{\partial^2 T}{\partial x \partial y} \right]_c D_y + \left[ \frac{\partial^3 T}{\partial x \partial y} \right]_c (x_m - x_c) D_x +
\]

\[
\left[ \frac{\partial^2 T}{\partial x^2} \right]_c D_x + \left[ \frac{\partial^2 T}{\partial x \partial y^2} \right]_c 2D_x D_y + \left[ \frac{\partial^3 T}{\partial x \partial y^2} \right]_c D_y \frac{(s - s_m)^2}{2} +
\]

\[
\left[ \frac{\partial^3 T}{\partial x^2 \partial y} \right]_c \frac{(x_m - x_c)^2}{2} + \left[ \frac{\partial^3 T}{\partial x^2 \partial y} \right]_c (x_m - x_c)(y_m - y_c) + \left[ \frac{\partial^3 T}{\partial y^3} \right]_c \frac{(y_m - y_c)^2}{2} +
\]

\[
\left[ \frac{\partial^2 T}{\partial x \partial y} \right]_c D_x + \left[ \frac{\partial^2 T}{\partial x \partial y} \right]_c D_y + \left[ \frac{\partial^3 T}{\partial x \partial y} \right]_c (x_m - x_c) D_x +
\]

Now equations 4.18 and 4.19 are substituted back into 4.14. The boundary constraints are then determined by matching up the terms with those in 4.15. The full Neumann Taylor
As for the Dirichlet case, the constraints shown above are greatly simplified if the boundary edge is parallel to either the $x$- or $y$-axis. For example, if the boundary edge were parallel to the $x$-axis, $y_b - y_a$ would be zero, meaning that $D_y = 0$, and the resulting boundary con-
4.1.3 Dirichlet & Neumann Combined Boundary Conditions

The Taylor constraint equations for the Dirichlet and Neumann boundary conditions can be coupled together (combining appropriate terms) for a general boundary condition of the form $\alpha T(s) + \beta \frac{dT(s)}{dn} = h(s)$, where $\alpha$ and $\beta$ are constants and $h$ is $C^3$, to create new boundary constraint equations for the least-squares reconstruction. The resulting boundary
constraint equations are:

\[
\begin{align*}
    h|_m &= (\alpha) T|_c + (\alpha (x_m - x_c) + \beta \hat{n}_x) \frac{\partial T}{\partial x}|_c + (\alpha (y_m - y_c) + \beta \hat{n}_y) \frac{\partial T}{\partial y}|_c + \\
    &\left(\frac{\alpha (x_m - x_c)^2}{2} + \beta (x_m - x_c) \hat{n}_x\right) \frac{\partial^2 T}{\partial x^2}|_c + \\
    &\left(\frac{\alpha (y_m - y_c)^2}{2} + \beta (y_m - y_c) \hat{n}_y\right) \frac{\partial^2 T}{\partial y^2}|_c + \\
    &\left(\frac{\alpha (x_m - x_c)^3}{6} + \beta \frac{(x_m - x_c)^2}{2} \hat{n}_x\right) \frac{\partial^3 T}{\partial x^3}|_c + \\
    &\left(\frac{\alpha (x_m - x_c)^2 (y_m - y_c)}{2} + \beta (x_m - x_c) (y_m - y_c) \hat{n}_x + \beta \frac{(x_m - x_c)^2}{2} \hat{n}_y\right) \frac{\partial^3 T}{\partial x^2 \partial y}|_c + \\
    &\left(\frac{\alpha (y_m - y_c)^2}{2} + \beta \frac{(y_m - y_c)^2}{2} \hat{n}_x + \beta (x_m - x_c) (y_m - y_c) \hat{n}_y\right) \frac{\partial^3 T}{\partial x \partial y^2}|_c + \\
    &\left(\frac{\alpha (y_m - y_c)^3}{6} + \beta \frac{(y_m - y_c)^2}{2} \hat{n}_y\right) \frac{\partial^3 T}{\partial y^3}|_c
\end{align*}
\]

(4.26)
\[ \frac{\partial h}{\partial s} \bigg|_m = (\alpha D_x) \frac{\partial T}{\partial x} \bigg|_c + (\alpha D_y) \frac{\partial T}{\partial y} \bigg|_c + (\alpha (x_m - x_c) D_x + \beta D_x \hat{n}_x) \frac{\partial^2 T}{\partial x^2} \bigg|_c + \\
(\alpha ((x_m - x_c) D_y + (y_m - y_c) D_x) + \beta D_y \hat{n}_x + \beta D_x \hat{n}_y) \frac{\partial^2 T}{\partial x^2 \partial y} \bigg|_c + \\
(\alpha (y_m - y_c) D_y + \beta D_y \hat{n}_y) \frac{\partial^2 T}{\partial y^2} \bigg|_c + \\
\left( \alpha \frac{(x_m - x_c)^2}{2} D_x + \beta (x_m - x_c) D_x \hat{n}_x \right) \frac{\partial^3 T}{\partial x^3} \bigg|_c + \\
\left( \alpha (x_m - x_c)(y_m - y_c) D_x + \alpha \frac{(x_m - x_c)^2}{2} D_y + \\
\beta ((x_m - x_c) D_y + (y_m - y_c) D_x) \hat{n}_x + \beta (x_m - x_c) D_x \hat{n}_y \right) \frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c + \\
\left( \alpha (x_m - x_c)(y_m - y_c) D_y + \alpha \frac{(y_m - y_c)^2}{2} D_x + \\
\beta (y_m - y_c) D_y \hat{n}_x + \beta ((x_m - x_c) D_y + (y_m - y_c) D_x) \hat{n}_y \right) \frac{\partial^3 T}{\partial y^3} \bigg|_c \]

\[ \frac{\partial^2 h}{\partial s^2} \bigg|_m = (\alpha D_x^2) \frac{\partial^2 T}{\partial x^2} \bigg|_c + (2\alpha D_x D_y) \frac{\partial^2 T}{\partial x \partial y} \bigg|_c + (\alpha D_y^2) \frac{\partial^2 T}{\partial y^2} \bigg|_c + \\
(\alpha (x_m - x_c) D_x^2 + \beta D_x^2 \hat{n}_x) \frac{\partial^3 T}{\partial x^3} \bigg|_c + \\
(2\alpha (x_m - x_c) D_x D_y + \alpha (y_m - y_c) D_x^2 + 2\beta D_x D_y \hat{n}_x + \beta D_x^2 \hat{n}_y) \frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c + \\
(2\alpha (y_m - y_c) D_x D_y + \alpha (x_m - x_c) D_x^2 + \beta D_y^2 \hat{n}_x + 2\beta D_x D_y \hat{n}_y \hat{n}_y) \frac{\partial^3 T}{\partial x \partial y^2} \bigg|_c + \\
(\alpha (y_m - y_c) D_y^2 + \beta D_y^2 \hat{n}_y) \frac{\partial^3 T}{\partial y^3} \bigg|_c \]

\[ \frac{\partial^3 h}{\partial s^3} \bigg|_m = (\alpha D_x^3) \frac{\partial^3 T}{\partial x^3} \bigg|_c + (3\alpha D_x^2 D_y) \frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c + (3\alpha D_x D_y^2) \frac{\partial^3 T}{\partial x \partial y^2} \bigg|_c + (\alpha D_y^3) \frac{\partial^3 T}{\partial y^3} \bigg|_c \]

The constraints shown above are greatly simplified if the boundary edge is parallel to either the x- or y-axis. For example, if the boundary edge were parallel to the x-axis, \( y_b - y_a \)
would be zero, meaning that \( D_y = 0 \), and the resulting boundary constraints are:

\[
| h \rangle_m = (\alpha) T \langle c + (\alpha (x_m - x_c) + \beta \hat{n}_x) \left. \frac{\partial T}{\partial x} \right|_c + (\alpha (y_m - y_c) + \beta \hat{n}_y) \left. \frac{\partial T}{\partial y} \right|_c + \\
\left( \frac{\alpha (x_m - x_c)^2}{2} + \beta (x_m - x_c) \hat{n}_x \right) \left. \frac{\partial^2 T}{\partial x^2} \right|_c + \\
(\alpha (x_m - x_c) (y_m - y_c) + \beta (y_m - y_c) \hat{n}_x + \beta (x_m - x_c) \hat{n}_y) \left. \frac{\partial^2 T}{\partial x \partial y} \right|_c + \\
\left( \frac{\alpha (y_m - y_c)^2}{2} + \beta (y_m - y_c) \hat{n}_y \right) \left. \frac{\partial^2 T}{\partial y^2} \right|_c + \\
\left( \frac{\alpha (x_m - x_c)^3}{6} + \beta \frac{(x_m - x_c)^2}{2} \hat{n}_x \right) \left. \frac{\partial^3 T}{\partial x^3} \right|_c + \\
(\alpha (x_m - x_c) (y_m - y_c)^2 + \beta (y_m - y_c)^2 \hat{n}_x + \beta (x_m - x_c) (y_m - y_c) \hat{n}_y) \left. \frac{\partial^3 T}{\partial x^2 \partial y} \right|_c + \\
\left( \frac{\alpha (y_m - y_c)^3}{6} + \beta \frac{(y_m - y_c)^2}{2} \hat{n}_y \right) \left. \frac{\partial^3 T}{\partial y^3} \right|_c
\]

\( (4.30) \)

\[
\left. \frac{\partial h}{\partial s} \right|_m = (\alpha D_x) \left. \frac{\partial T}{\partial x} \right|_c + (\alpha (x_m - x_c) D_x + \beta D_x \hat{n}_x) \left. \frac{\partial^2 T}{\partial x^2} \right|_c + (\alpha (y_m - y_c) D_x) \left. \frac{\partial^2 T}{\partial x \partial y} \right|_c + \\
\left( \frac{\alpha (x_m - x_c)^2}{2} D_x + \beta (x_m - x_c) D_x \hat{n}_x \right) \left. \frac{\partial^3 T}{\partial x^3} \right|_c + \\
(\alpha (x_m - x_c) (y_m - y_c) D_x + \beta (y_m - y_c) D_x \hat{n}_x + \beta (x_m - x_c) D_x \hat{n}_y) \left. \frac{\partial^3 T}{\partial x^2 \partial y} \right|_c + \\
\left( \frac{\alpha (y_m - y_c)^2}{2} D_x + \beta (y_m - y_c) D_x \hat{n}_y \right) \left. \frac{\partial^3 T}{\partial y^3} \right|_c
\]

\( (4.31) \)

\[
\left. \frac{\partial^2 h}{\partial s^2} \right|_m = (\alpha D_x^2) \left. \frac{\partial^2 T}{\partial x^2} \right|_c + (\alpha (x_m - x_c) D_x^2 + \beta D_x^2 \hat{n}_x) \left. \frac{\partial^3 T}{\partial x^3} \right|_c + \\
(\alpha (y_m - y_c) D_x^2 + \beta D_x^2 \hat{n}_y) \left. \frac{\partial^3 T}{\partial x^2 \partial y} \right|_c
\]

\( (4.32) \)

\[
\left. \frac{\partial^3 h}{\partial s^3} \right|_m = (\alpha D_x^3) \left. \frac{\partial^3 T}{\partial x^3} \right|_c
\]

\( (4.33) \)
4.1.4 Drawbacks

There are several drawbacks with the Taylor Constraints method for boundary condition enforcement.

- The total number of constraints can be excessive. Table 4.1 compares the orders of accuracy, the total number of constraints (assuming Dirichlet boundary conditions and including the mean constraint) for control volumes with one and two boundary edges and the number of unknowns being solved in the reconstruction. Several of the values in the tables are stressed in bold to indicate there are a greater number of constraints than unknown variables. The least-squares problem in these cases is over-constrained. There are situations, emphasized in italics, where there are the same number of constraints, or one less, than there are unknowns, which makes the reconstruction less reliable, since there is only one degree of freedom for the least-squares algorithm to calculate the unknown value accurately. At second-order this is not a problem, since the second-order least-squares matrix will never degenerate with two constraints and one integral constant.

Table 4.1: Constraint Comparison for Control Volumes with Taylor Constraints

<table>
<thead>
<tr>
<th>Order of Accuracy</th>
<th>Cell Boundary Edges</th>
<th>Vertex Boundary Edges</th>
<th>Unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One</td>
<td>Two</td>
<td>Two</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

- Another limitation is that the boundary condition equation must be $C^3$ (to assume it
can be approximated with a cubic Taylor polynomial for fourth-order accuracy). This is the case for any scheme to be fourth-order accurate.

- The derived Taylor constraints were for arbitrary straight boundary edges only. While it is possible in principle to derive the Taylor constraints for curved boundary edges, by replacing equations 4.3 and 4.4 by approximating $x(s)$ with a cubic Taylor polynomial in $s$, the method for doing so is quite complex and, considering the existing problems with the scheme, the derivation was not done.

Because of these problems, a different approach was developed for creating the boundary constraints to add to the least-squares reconstruction problem.

### 4.2 Gauss Constraints

The Gauss Constraints method was the second attempt at generating boundary constraints. This method enforces the boundary condition at the Gauss points used in the flux integral, instead of enforcing it along the entire boundary edge of the control volume. The reconstruction polynomial is evaluated at the Gauss points and is set equal to the boundary condition at that point. This gives one constraint on the coefficients of the reconstruction polynomial per Gauss point. These boundary constraints may be applied to any boundary edge, straight or curved, where the Gauss point locations and normals are known. The Gauss Constraints method is simpler to implement than the Taylor Constraints method, and does not suffer from the same drawbacks.
4.2.1 Dirichlet Boundary Conditions

The solution in the boundary control volume \( T(x,y) \) is approximated with a cubic Taylor polynomial about the reference point of the control volume. This polynomial is then evaluated at each Gauss point along the boundary edge of the control volume and equated to the known boundary temperature \( f(x_{bdry},y_{bdry}) \). The result is one boundary constraint per Gauss point along the boundary edge:

\[
\begin{align*}
T(x_g, y_g) &= T|_c + \left( \frac{\partial T}{\partial x} \right)|_c (x_g - x_c) + \left( \frac{\partial T}{\partial y} \right)|_c (y_g - y_c) + \\
&+ \left( \frac{\partial^2 T}{\partial x^2} \right)|_c \frac{(x_g - x_c)^2}{2} + \left( \frac{\partial^2 T}{\partial x \partial y} \right)|_c (x_g - x_c) (y_g - y_c) + \left( \frac{\partial^2 T}{\partial y^2} \right)|_c \frac{(y_g - y_c)^2}{2} + \\
&+ \left( \frac{\partial^3 T}{\partial x^3} \right)|_c \frac{(x_g - x_c)^3}{6} + \left( \frac{\partial^3 T}{\partial x^2 \partial y} \right)|_c \frac{(x_g - x_c)^2 (y_g - y_c)}{2} + \\
&+ \left( \frac{\partial^3 T}{\partial x \partial y^2} \right)|_c \frac{(x_g - x_c) (y_g - y_c)^2}{2} + \left( \frac{\partial^3 T}{\partial y^3} \right)|_c \frac{(y_g - y_c)^3}{6}
\end{align*}
\] (4.34)

4.2.2 Neumann Boundary Conditions

The Gauss constraint method for the Neumann boundary condition \( \frac{\partial T}{\partial n} = g(s) \) is almost as simple as for the Dirichlet boundary condition. The Gauss constraint forces the normal derivative at the Gauss point on the boundary to the specified condition. Applying the definition of the directional derivative, with the knowledge that \( x = x(s) \), the Neumann boundary condition is rewritten as:

\[
\frac{\partial T}{\partial x} \hat{n}_x + \frac{\partial T}{\partial y} \hat{n}_y = g(x(s), y(s))
\] (4.35)

The spatial partial derivatives of \( T \) can be determined by taking the partial derivatives of the Taylor polynomial approximation of \( T \). This gives the following Taylor polynomial
approximation to the partial derivatives:

\[
\frac{\partial T}{\partial x} = \frac{\partial T}{\partial x}_c + \frac{\partial^2 T}{\partial x^2}_c (x - x_c) + \frac{\partial^2 T}{\partial x \partial y}_c (y - y_c) + \frac{\partial^3 T}{\partial x^3}_c \frac{(x - x_c)^2}{2} + \\
\frac{\partial^3 T}{\partial x^2 \partial y}_c (x - x_c) (y - y_c) + \frac{\partial^3 T}{\partial x \partial y^2}_c \frac{(y - y_c)^2}{2} 
\]

(4.36)

\[
\frac{\partial T}{\partial y} = \frac{\partial T}{\partial y}_c + \frac{\partial^2 T}{\partial x \partial y}_c (x - x_c) + \frac{\partial^2 T}{\partial y^2}_c (y - y_c) + \frac{\partial^3 T}{\partial x^2 \partial y}_c \frac{(x - x_c)^2}{2} + \\
\frac{\partial^3 T}{\partial x \partial y^2}_c (x - x_c) (y - y_c) + \frac{\partial^3 T}{\partial y^3}_c \frac{(y - y_c)^2}{2} 
\]

(4.37)

Substituting equations 4.36, 4.37 into equation 4.35, the full Gauss constraint can be written as follows, for any given Gauss point \((x_g, y_g)\):

\[
g(x_g, y_g) = \left( \frac{\partial T}{\partial x}_c + \frac{\partial^2 T}{\partial x^2}_c (x_g - x_c) + \frac{\partial^2 T}{\partial x \partial y}_c (y_g - y_c) + \frac{\partial^3 T}{\partial x^3}_c \frac{(x_g - x_c)^2}{2} + \\
\frac{\partial^3 T}{\partial x^2 \partial y}_c (x_g - x_c) (y_g - y_c) + \frac{\partial^3 T}{\partial x \partial y^2}_c \frac{(y_g - y_c)^2}{2} \right) \hat{n}_x + \\
\left( \frac{\partial T}{\partial y}_c + \frac{\partial^2 T}{\partial x \partial y}_c (x_g - x_c) + \frac{\partial^2 T}{\partial y^2}_c (y_g - y_c) + \frac{\partial^3 T}{\partial x^2 \partial y}_c \frac{(x_g - x_c)^2}{2} + \\
\frac{\partial^3 T}{\partial x \partial y^2}_c (x_g - x_c) (y_g - y_c) + \frac{\partial^3 T}{\partial y^3}_c \frac{(y_g - y_c)^2}{2} \right) \hat{n}_y 
\]

(4.38)
Collecting like terms, the general Neumann Gauss constraint becomes:

\[ g(x, y) = (\hat{n}_x) \left. \frac{\partial T}{\partial x} \right|_c + (\hat{n}_y) \left. \frac{\partial T}{\partial y} \right|_c + (\hat{n}_x(x_g - x_c)) \left. \frac{\partial^2 T}{\partial x^2} \right|_c + \\
(\hat{n}_x(y_g - y_c)) + \hat{n}_y(x_g - x_c) \left. \frac{\partial^2 T}{\partial x \partial y} \right|_c + (\hat{n}_y(y_g - y_c)) \left. \frac{\partial^2 T}{\partial y^2} \right|_c + \\
\left( \hat{n}_x \frac{(x_g - x_c)^2}{2} \right) \left. \frac{\partial^3 T}{\partial x^3} \right|_c + \\
\left( \hat{n}_x(x_g - x_c)(y_g - y_c) + \hat{n}_y \frac{(x_g - x_c)^2}{2} \right) \left. \frac{\partial^3 T}{\partial x^2 \partial y} \right|_c + \\
\left( \hat{n}_y \frac{(y_g - y_c)^2}{2} + \hat{n}_y(x_g - x_c)(y_g - y_c) \right) \left. \frac{\partial^3 T}{\partial x \partial y^2} \right|_c + \\
\left( \hat{n}_y \frac{(y_g - y_c)^2}{2} \right) \left. \frac{\partial^3 T}{\partial y^3} \right|_c \]  

(4.39)

### 4.2.3 Dirichlet & Neumann Combined Boundary Conditions

The Gauss constraint equations for the Dirichlet and Neumann boundary conditions can be coupled together (combining appropriate terms) for a general boundary condition of the form \( \alpha T(s) + \beta \frac{\partial T(s)}{\partial n} = h(s) \), where \( \alpha \) and \( \beta \) are constants, to create new boundary constraint equations for the least-squares reconstruction. The resulting boundary constraint equation
for the Gauss point \((x_g, y_g)\), along the boundary, is:

\[
\begin{align*}
\frac{\partial T}{\partial x} \bigg|_c &= \alpha \left[ T(\alpha) + \left( x_g - x_c \right) + \frac{\partial^2 T}{\partial x^2} \bigg|_c \left( y_g - y_c \right) + \frac{\partial^2 T}{\partial x^2} \bigg|_c \frac{(x_g - x_c)^2}{2} + \
\frac{\partial^3 T}{\partial x^3} \bigg|_c \right] + \\
\frac{\partial T}{\partial y} \bigg|_c &= \beta \left[ \frac{\partial^2 T}{\partial x^2} \bigg|_c \frac{(x_g - x_c)^2}{2} + \frac{\partial^3 T}{\partial x^2 \partial y} \bigg|_c \frac{(x_g - x_c)}{2} + \
\frac{\partial^3 T}{\partial y^3} \bigg|_c \right]
\end{align*}
\]  

(4.40)
Collecting like terms results in:

\[
\begin{align*}
  h(x_g, y_g) &= (\alpha) T|_c + (\alpha (x_g - x_c) + \beta \hat{n}_x) \frac{\partial T}{\partial x}|_c + (\alpha (y_g - y_c) + \beta \hat{n}_y) \frac{\partial T}{\partial y}|_c + \\
  &\left(\frac{\alpha (x_g - x_c)^2}{2} + \beta (x_g - x_c) \hat{n}_x\right) \frac{\partial^2 T}{\partial x^2}|_c + \\
  &\left(\frac{\alpha (y_g - y_c)^2}{2} + \beta (y_g - y_c) \hat{n}_y\right) \frac{\partial^2 T}{\partial y^2}|_c + \\
  &\left(\frac{\alpha (x_g - x_c)^3}{6} + \beta \frac{(x_g - x_c)^2}{2} \hat{n}_x\right) \frac{\partial^3 T}{\partial x^3}|_c + \\
  &\left(\frac{\alpha (x_g - x_c)^2 (y_g - y_c)}{2} + \beta (x_g - x_c) (y_g - y_c) \hat{n}_x + \beta \frac{(x_g - x_c)^2}{2} \hat{n}_y\right) \frac{\partial^3 T}{\partial x^2 \partial y}|_c + \\
  &\left(\frac{\alpha (x_g - x_c)^2 (y_g - y_c)}{2} + \beta \frac{(y_g - y_c)^2}{2} \hat{n}_x + \beta (x_g - x_c) (y_g - y_c) \hat{n}_y\right) \frac{\partial^3 T}{\partial x \partial y^2}|_c + \\
  &\left(\frac{\alpha (y_g - y_c)^3}{6} + \beta \frac{(y_g - y_c)^2}{2} \hat{n}_y\right) \frac{\partial^3 T}{\partial y^3}|_c
\end{align*}
\]

(4.41)

### 4.2.4 Constraint Comparison

It was previously shown there are too many Taylor constraints, but what about the number of Gauss constraints? Table 4.2 shows there are enough degrees of freedom in each case to be able to perform the least-squares reconstruction reliably. At second-order accuracy, there are still situations where there is only one or no degrees of freedom for the least-squares reconstruction. Fortunately this is not a problem, since the second-order least-squares matrix will never degenerate with two constraints and one integral constant.

Figure 4.1 shows, on a partial mesh, the two ways cell-centred control volumes can appear in the corners. The upper left shows an example of a control volume with two
Table 4.2: Constraint Comparison for Control Volumes with Gauss Constraints

<table>
<thead>
<tr>
<th>Order of Accuracy</th>
<th>Cell Boundary Edges</th>
<th>Vertex Boundary Edges</th>
<th>Unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One</td>
<td>Two</td>
<td>Two</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 4.1: Cell-centred control volume’s Gauss point locations

Figure 4.2: Vertex-centred control volume’s Gauss point locations

different boundary edges. This gives it a total of four Gauss points. However the upper right control volume shows how it is possible to have only two Gauss points, the same number as any regular cell-centred boundary control volume. Figure 4.2 shows, on the same partial mesh, the two different ways vertex-centred control volumes appear in corners. In each case the control volume has two different boundary edges. For regular vertex-centred control volumes, this is also the case, as shown by the hollow circle Gauss points. So no matter the location of a vertex-centred control volume, the number of Gauss points remains constant.
4.3 Constrained Least-Squares Reconstruction Method

This section describes how the constraints are applied to the least-squares reconstruction problem and the method by which the problem is solved. The first step is to add the previously derived constraints to the top of the least-squares matrix. With the constraints in place the reconstruction is a constrained least-squares problem. An example of this matrix is shown in Figure 4.3.

The constraint equations must be satisfied exactly, not just in a least-squares sense. To accomplish this, Gaussian elimination is performed on as many rows as the total number of constraints $m$ (including the mean constraint). The method is described here for each constraint $i = [1..m]$.

1. Determine the largest constraint coefficient in row $i$ and divide that row through by it, including the right-hand side.

2. Exchange the column with this largest constraint coefficient with column $i$ (column pivoting). This has the effect of exchanging corresponding rows in the solution vector.

3. The new column $i$, now containing a 1 in row $i$, is then reduced using Gaussian Elimination from row $i + 1$ onwards.

The result is a partially reduced least-squares matrix. An example of this is shown in Figure 4.4. The least-squares algorithm, by use of Householder transformations, is then applied to the matrix, which computes the unknown Taylor polynomial coefficients for the approximation to the solution for the control volume, with proper constraint enforcement.
The least-squares algorithm by use of Normal equations can also be used with a little more work. The entire constrained matrix, shown in Figure 4.4, is not used. Instead, the least-squares algorithm by use of Normal equations solves the matrix given by $y_{1,2}$ to $y_{n,10}$ for the corresponding unknowns. The solved unknowns are then used to solve the constraint equations for the remaining unknown coefficients.
CHAPTER 4. BOUNDARY CONDITIONS

Figure 4.3: Example of the least-squares matrix. The c's represent the constraint equations and the x's represent the rest of the least-squares matrix.

\[
\begin{bmatrix}
c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} & \cdots & c_{1,10} \\
c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & c_{2,5} & \cdots & c_{2,10} \\
c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & c_{3,5} & \cdots & c_{3,10} \\
\end{bmatrix}
\begin{bmatrix}
x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} & \cdots & x_{1,10} \\
x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{2,5} & \cdots & x_{2,10} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
x_{n,1} & x_{n,2} & x_{n,3} & x_{n,4} & x_{n,5} & \cdots & x_{n,10} \\
\end{bmatrix}
\begin{bmatrix}
T |_
C \\
\frac{\partial T}{\partial x} |_
C \\
\frac{\partial T}{\partial y} |_
C \\
\frac{\partial^2 T}{\partial x^2} |_
C \\
\frac{\partial^2 T}{\partial x \partial y} |_
C \\
\frac{\partial^2 T}{\partial y^2} |_
C \\
\end{bmatrix}
= 
\begin{bmatrix}
C_1 \\
C_2 \\
C_3 \\
X_1 \\
X_2 \\
\vdots \\
X_n \\
\end{bmatrix}
\]

Figure 4.4: Example of the constrained least-squares matrix, where columns 2 and 4 were exchanged by column pivoting. The d's represent the altered constraint equations and the y's represent the rest of the altered least-squares matrix.

\[
\begin{bmatrix}
1 & d_{1,4} & d_{1,3} & d_{1,2} & d_{1,5} & \cdots & d_{1,10} \\
0 & 1 & d_{2,3} & d_{2,2} & d_{2,5} & \cdots & d_{2,10} \\
0 & 0 & 1 & d_{3,2} & d_{3,5} & \cdots & d_{3,10} \\
0 & 0 & 0 & y_{1,2} & y_{1,5} & \cdots & y_{1,10} \\
0 & 0 & 0 & y_{2,2} & y_{2,5} & \cdots & y_{2,10} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & y_{n,2} & y_{n,5} & \cdots & y_{n,10} \\
\end{bmatrix}
\begin{bmatrix}
T |_
C \\
\frac{\partial^2 T}{\partial x^2} |_
C \\
\frac{\partial T}{\partial y} |_
C \\
\frac{\partial^2 T}{\partial x \partial y} |_
C \\
\frac{\partial^2 T}{\partial y^2} |_
C \\
\end{bmatrix}
= 
\begin{bmatrix}
D_1 \\
D_2 \\
D_3 \\
Y_1 \\
Y_2 \\
\vdots \\
Y_n \\
\end{bmatrix}
\]
Chapter 5

Validation

The finite-volume flux scheme, developed in the previous chapters for two types of control volumes, was validated in several stages. The accuracy of the fluxes was first validated for the interior control volumes, then over an entire simple square mesh with Dirichlet boundary conditions, for both boundary constraint methods. The Gauss Constraints method was shown to be superior and verification continued with more complex problems, including combined Dirichlet and Neumann boundary conditions and meshes with curved boundaries.

Accuracy was assessed by examining the behaviour of the norms of the error in the flux integral with mesh refinement. The flux error is calculated by subtracting the numerically calculated flux integral from the exact flux integral. The exact flux integral is calculated by evaluating $\int_{CV} \nabla^2 T \, dA$ using Gauss quadrature over control volumes, where $T$ is the known initial solution. The Gauss quadrature for the exact flux integral is performed to sixth-order accuracy to ensure that errors in this integral do not contaminate the results.

The error of each test case is examined using the $L_k$ error norms, where $k = 1, 2, \ldots, \infty$. 
CHAPTER 5. VALIDATION

The $L_1$ norm describes the error globally — all errors are weighted equally. As $k$ increases, the error norm includes more local errors, until the error is strictly a local error at infinity.

The $L_k$ error norm, weighted by control volume area, is calculated using the formula shown in equation 5.1, where $N$ is the number of control volumes, $A_i$ is the control volume area and $E_i$ is the flux error, calculated as the exact flux integral minus the numerical flux integral, for the $i^{th}$ control volume. The error norms used during validation are the $L_1$, $L_2$ (which is a measure of both global and local error) and the $L_\infty$. The $L_\infty$ error norm is simply the highest error magnitude in the mesh.

$$L_k \text{ Error Norm} = \left( \frac{\sum_{i=1}^{N} A_i E_i^k}{\sum_{i=1}^{N} A_i} \right)^{\frac{1}{k}} \quad (5.1)$$

The order of accuracy was determined by plotting the number of control volumes versus the error norm on a log-log plot. The data points are obtained by running a flux calculation program using the same initial solution and boundary conditions over several identical geometries, meshed with a variety of coarsenesses. Examples of the coarsest meshes can be found in Appendix A. Ideally the data points on the log-log plot give a straight line whose slope represents negative one half of the order of accuracy obtained. The reason for this is explained as follows: the area of a control volume in the mesh is proportional to $1/N$, where $N$ is the number of control volumes in the mesh. So the length scale $\Delta L$ is proportional to $1/\sqrt{N}$. Suppose that the error is $O(\Delta x^n)$. Then the $L_k$ error norm is $\sqrt{\frac{N^{-\frac{k}{2}} (N^{-\frac{n}{2}})^k}{N^{-\frac{k}{2}}}}$, which reduces to $L_k = N^{-\frac{n}{2}}$. Taking the log of both sides results in $\frac{\log(L_k)}{\log(N)} = -\frac{n}{2}$. In practice the slope of the line of best fit through the data points is used to determine the order of accuracy.
For the rest of this thesis, the legend for the error norm plots is shown in Figure 5.1. It is also important to note that the flux error plots shown are for coarse meshes, since the finer meshes do not print well on paper due to the smallness of the control volumes.

\[ \begin{align*}
\text{--- 2nd-Order} \\
\text{--- 3rd-Order} \\
\text{--- 4th-Order}
\end{align*} \]

Figure 5.1: Error Norm Plot Legend

5.1 Interior Control Volume Flux Integral Validation

The interior control volume fluxes were validated before attempts at computing the boundary conditions were made. The interior control volume fluxes were obtained by running the flux integral calculation program with an initial temperature distribution of \( T(x, y) = \sin(\pi x) \sin(\pi y) \) on square meshes with a domain and range of \([-1, 1]\) with all boundary conditions set to zero. Figure 5.2 shows a plot of the initial temperature distribution with cell-centred control volumes. Figure 5.3 shows a plot of the exact flux with vertex-centred control volumes.

Table 5.1 contains the calculated error norms of the error data collected during testing of the cell-centred and vertex-centred schemes for the interior control volumes. The error norms were calculated using only the interior control volumes of the meshes. Figure 5.4 shows the \( L_2 \) error norm plot for the interior cell-centred control volumes and Figure 5.5 shows the same for vertex-centred control volumes. The plots clearly show the slopes of the
Figure 5.2: Initial temperature of \( \sin(\pi x) \sin(\pi y) \) on cell-centred control volumes

Figure 5.3: Exact flux for \( \sin(\pi x) \sin(\pi y) \) on vertex-centred control volumes

Figure 5.4: \( L_2 \) error norm plots for interior cell-centred control volumes

Figure 5.5: \( L_2 \) error norm plots for interior vertex-centred control volumes

Table 5.1: Error Norms of the Interior Control Volumes

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th>Vertex-Centred Control Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( 2^{nd} )-Order</td>
<td>( 3^{rd} )-Order</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>1.94</td>
<td>2.94</td>
</tr>
<tr>
<td>( L_2 )</td>
<td>1.94</td>
<td>2.94</td>
</tr>
<tr>
<td>( L_\infty )</td>
<td>1.71</td>
<td>2.67</td>
</tr>
</tbody>
</table>
lines increasing with order of accuracy. The slopes are almost exactly half the desired orders of accuracy, indicating the correctness of the scheme with both control volumes.

5.2 Dirichlet Boundary Conditions on a Square Mesh

With the interior control volumes proven to be high-order accurate, the boundary flux calculation becomes the next target to validate high-order accuracy. The first test case run which included boundary conditions used the same mesh and test function as for the interior control volume testing, with compatible Dirichlet boundary conditions. Figure 5.2 shows a plot of the function using cell-centred control volumes and Figure 5.3 shows a plot of the exact flux used to compare the numerical results for vertex-centred control volumes.

5.2.1 Accuracy Analysis — Taylor Constraints

This section analyses the results of running the flux calculation program on a square mesh using the Taylor Constraints method for generating the boundary constraints for the reconstruction.

For cell-centred control volumes Figure 5.6 shows the error between the numerical flux, calculated to fourth-order accuracy, and the exact flux. The flux errors in some of the boundary control volumes are significantly larger than the interior error, known to be high-order accurate.

The $L_2$ error norms for each of the three orders of accuracy for the cell-centred scheme are plotted in Figure 5.8. The scheme is shown to be second-order accurate by the solid
CHAPTER 5. VALIDATION

Figure 5.6: Flux error for the cell-centred control volume scheme using Dirichlet Taylor constraints

Figure 5.7: Flux error for the vertex-centred control volume scheme using Dirichlet Taylor constraints

Figure 5.8: $L_2$ error norm plots for the cell-centred control volume scheme using Dirichlet Taylor constraints

Figure 5.9: $L_2$ error norm plots for the vertex-centred control volume scheme using Dirichlet Taylor constraints

Table 5.2: Error Norms with Dirichlet Taylor Constraints

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$2^{nd}$-Order</td>
<td>$3^{rd}$-Order</td>
<td>$4^{th}$-Order</td>
<td>$2^{nd}$-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>2.16</td>
<td>3.31</td>
<td>4.14</td>
<td>2.20</td>
</tr>
<tr>
<td>$L_2$</td>
<td>2.16</td>
<td>3.09</td>
<td>3.88</td>
<td>1.76</td>
</tr>
<tr>
<td>$L_{\infty}$</td>
<td>1.88</td>
<td>2.00</td>
<td>3.09</td>
<td>1.08</td>
</tr>
</tbody>
</table>
straight line with a slope of -1. However the scheme does not perform as well at higher orders of accuracy, as indicated by the slightly crooked lines for third- and fourth-orders of accuracy. The high orders of accuracy do improve the solution, but not consistently, most likely because of the over-constraining problem described in the Taylor constraint method for generating boundary control volumes.

For the vertex-centred control volumes, the story is even worse. Figure 5.7 shows the error between the numerical flux, calculated to fourth-order accuracy, and the exact flux. The figure shows that the flux error of the boundary control volumes greatly outweighs that of the high-order accurate interior control volumes.

The $L_2$ error norms for each of the three orders of accuracy for the vertex-centred scheme are plotted in Figure 5.9. The scheme is shown to be second-order accurate, regardless of desired order of accuracy, most likely because of the over-constraining problem described in the Taylor constraint method for generating boundary constraints.

Table 5.2 shows the different error norms for each order of accuracy for each control volume type. The orders of accuracy are computed from the slopes of the lines of best fit to the data points. The vertex-centred control volumes do not achieve high orders of accuracy with the Taylor constraints.

### 5.2.2 Accuracy Analysis — Gauss Constraints

This section analyses the results of running the flux calculation program on a square mesh using the Gauss Constraints method for generating the boundary constraints to add to the
Figure 5.10: Flux error for the cell-centred control volume scheme using Dirichlet Gauss constraints

Figure 5.11: Flux error for the vertex-centred control volume scheme using Dirichlet Gauss constraints

Figure 5.12: $L_2$ error norm plots for the cell-centred control volume scheme using Dirichlet Gauss constraints

Figure 5.13: $L_2$ error norm plots for the vertex-centred control volume scheme using Dirichlet Gauss Constraints

Table 5.3: Error Norms with Dirichlet Gauss Constraints

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th>Vertex-Centred Control Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2nd-Order</td>
<td>3rd-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>1.99</td>
<td>3.24</td>
</tr>
<tr>
<td>$L_2$</td>
<td>1.99</td>
<td>3.26</td>
</tr>
<tr>
<td>$L_{\infty}$</td>
<td>1.78</td>
<td>2.78</td>
</tr>
</tbody>
</table>
CHAPTER 5. VALIDATION

reconstruction.

Figure 5.10 shows the error between the numerical flux, calculated to fourth-order accuracy, and the exact flux. As with the Taylor boundary constraints, the figure shows that some of the flux errors in the boundary control volumes outweigh the interior error, known to be high-order accurate. However, unlike using Taylor constraints, the boundary error diminishes as rapidly as the interior error when the mesh is refined. The $L^2$ error norms for each of the three orders of accuracy for the cell-centred scheme are plotted in Figure 5.12. The scheme is shown to reach each of the orders of accuracy successfully.

For the vertex-centred control volumes, the story is the same. Figure 5.11 shows the error between the numerical flux, calculated to fourth-order accuracy, and the exact flux. The figure shows that some of the flux errors of the boundary control volumes outweigh that of the high-order accurate interior control volumes. Again this error decreases rapidly as the mesh is refined. The $L^2$ error norms for each of the three orders of accuracy for the vertex-centred scheme are plotted in Figure 5.13. The scheme is again shown to reach each of the orders of accuracy successfully. The reconstruction is no longer over-constrained and is able to complete the calculations to the proper orders of accuracy.

Table 5.3 shows the different error norms for each order of accuracy for each control volume type. The orders of accuracy are computed from the slopes of the lines of best fit to the data points. Each error norm meets the target order of accuracy, although the $L_\infty$ error norm is less well-behaved than the others, because of its sensitivity to local error, which is greatest along the boundary. This proves the Gauss Constraints method for generating the constraint equations works to high-order accuracy.
Since the Gauss constraints proved superior to the Taylor constraints, the Taylor con­
straints were discarded. The rest of the validation deals only with cases using Gauss con­
straints.

5.3 Neumann Boundary Conditions on a Square Mesh

The function for testing the Neumann boundary conditions was $T(x, y) = \sin(\pi x) \sinh(\pi y)$.
The boundary conditions were set up such that the left and right vertical walls used $\frac{\partial T}{\partial n} = \pm \pi \cos(\pi x) \sinh(\pi y)$ and the upper and lower walls used $\frac{\partial T}{\partial n} = \pm \pi \sin(\pi x) \cosh(\pi y)$ (the alternating signs depend on the direction of the inward boundary normal). A plot of the function is shown in Figure 5.14 for vertex-centred control volumes. Since the Laplacian of the initial function is zero, the exact flux is also zero everywhere in the mesh.

![Initial function of $\sin(\pi x) \sinh(\pi y)$ for vertex-centred control volumes for Neumann boundary conditions](image)

Figures 5.15 and 5.16 show the flux integral error for the Cell scheme and the Vertex scheme respectively. In both cases the error along the lower and upper boundaries is rather
CHAPTER 5. VALIDATION

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th>Vertex-Centred Control Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2nd-Order</td>
<td>3rd-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>1.93</td>
<td>3.05</td>
</tr>
<tr>
<td>$L_2$</td>
<td>1.92</td>
<td>3.04</td>
</tr>
<tr>
<td>$L_\infty$</td>
<td>1.73</td>
<td>2.69</td>
</tr>
</tbody>
</table>
large. This remains so as the mesh is refined. This is because the gradient along these walls is very large for the boundary control volumes, a value of over 36 at the peaks where the boundary error is largest. Looking at the $L_2$ error norm plots in Figures 5.17 and 5.18, the slopes indicate the orders of accuracy are obtained. This is confirmed by the data collected in Table 5.4. The $L_\infty$ error norms represent a measure of the local error, which is sensitive, due to the gradients at the boundary control volumes. High orders of accuracy are still achieved, even though the approximation of the very large gradients at the boundaries are locally inaccurate.

5.4 Combined Dirichlet & Neumann Boundary Conditions on a Square Mesh

![Initial function for $\sin(\frac{\pi}{4}x) \sinh(\frac{\pi}{4}y)$ vertex-centred control volumes for combined boundary conditions](image)

Figure 5.19: Initial function for $\sin(\frac{\pi}{4}x) \sinh(\frac{\pi}{4}y)$ vertex-centred control volumes for combined boundary conditions

In an effort to see if the size of the gradients really is the problem with the Neumann
CHAPTER 5. VALIDATION

Figure 5.20: Flux error for the cell-centred control volume scheme using combined Gauss constraints

Figure 5.21: Flux error for the vertex-centred control volume scheme using combined Gauss constraints

Figure 5.22: $L_2$ error norm plots for the cell-centred control volume scheme using combined Gauss constraints

Figure 5.23: $L_2$ error norm plots for the vertex-centred control volume scheme using combined Gauss constraints

Table 5.5: Error Norms with Gauss Constraints

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th></th>
<th>Vertex-Centred Control Volumes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2nd-Order</td>
<td>3rd-Order</td>
<td>4th-Order</td>
<td>2nd-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>2.01</td>
<td>3.06</td>
<td>4.26</td>
<td>2.38</td>
</tr>
<tr>
<td>$L_2$</td>
<td>2.04</td>
<td>3.08</td>
<td>4.27</td>
<td>2.08</td>
</tr>
<tr>
<td>$L_{\infty}$</td>
<td>1.92</td>
<td>2.80</td>
<td>3.72</td>
<td>1.12</td>
</tr>
</tbody>
</table>
boundary conditions, the initial problem was adjusted slightly to \( T(x, y) = \sin(\frac{x}{4}) \sinh(\frac{y}{4}) \).

This time Neumann boundary conditions were used on the upper and left walls, and the lower and right walls used a linear combination of Dirichlet and Neumann boundary conditions of the form \( \frac{3}{2} T + \frac{\partial T}{\partial n} \). A plot of the function is shown in Figure 5.19 for vertex-centred control volumes. Since the Laplacian of the initial function is still zero, the exact flux is also zero everywhere in the mesh.

Figures 5.20 and 5.21 show the flux integral error for the Cell scheme and the Vertex scheme respectively. In both cases the error along the lower and upper boundaries has decreased, indicating that the large gradients were indeed the problem in the flux integral calculation for the case in Section 5.3. Looking at the \( L_2 \) error norm plots in Figures 5.22 and 5.23, the slopes indicate the orders of accuracy are obtained. This is confirmed by the data collected in Table 5.5.

5.5 Singular Point on a Square Mesh

Previous cases had smooth solutions. What if that is not the case? A test case with \( \frac{\partial T}{\partial n} = 0 \) on the upper and left walls, \( \frac{\partial T}{\partial n} = 1 \) on the right wall and \( T = 0 \) on the bottom proved to be interesting. The exact solution to this problem was computed analytically to be

\[
T(x, y) = \sum_{k=\text{odd}} \frac{16 \sin\left(\frac{k\pi(x+1)}{4}\right) \cosh\left(\frac{k\pi(x+1)}{4}\right)}{k^2 \pi^2 \sinh\left(\frac{k\pi}{2}\right)}.
\]

A plot of the function is shown in Figure 5.24. Since the Laplacian of the initial solution is zero, then the exact flux integral is also zero.

Figures 5.25 and 5.26 show there is a problem with the flux integral calculation in the control volumes around the bottom right corner of the mesh. The reason for this is that
the corner vertex is a singular point, at which $\frac{\partial T}{\partial y} \to \infty$. As with the first Neumann test case in Section 5.3, there will be a large gradient (at least in the $y$ direction) which causes problems with the flux integral calculation. As the mesh is refined, the error in the corner decreases, as indicated in the $L_2$ error norm plots in Figures 5.27 and 5.28. However the error at that location remains much greater than elsewhere in the mesh, resulting in, at best, second-order accuracy as shown in Table 5.6. This is a recurring problem whenever the gradients get too large for the flux schemes to handle.

### 5.6 Curved Mesh Boundaries

With the Cell and Vertex flux integration schemes shown to work on a simple square mesh with boundaries parallel to the Cartesian axes, the next test involves curved mesh boundaries. Curved mesh boundaries do not affect the flux integral calculations of the interior control volumes: only the boundary control volumes are affected. Two different curved
Figure 5.25: Flux error for singular test case using the cell-centred control volume scheme with Gauss constraints

Figure 5.26: Flux error for singular test case using the vertex-centred control volume scheme with Gauss constraints

Figure 5.27: $L_2$ error norm plots for the cell-centred control volume scheme using Gauss constraints

Figure 5.28: $L_2$ error norm plots for the vertex-centred control volume scheme using Gauss constraints

Table 5.6: Error Norms with Gauss Constraints

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th>Vertex-Centred Control Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$2^{nd}$-Order</td>
<td>$3^{rd}$-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>1.97</td>
<td>2.46</td>
</tr>
<tr>
<td>$L_2$</td>
<td>1.66</td>
<td>1.60</td>
</tr>
<tr>
<td>$L_{\infty}$</td>
<td>0.70</td>
<td>0.59</td>
</tr>
</tbody>
</table>
5.6.1 Circle Mesh Boundary

The circle mesh boundary was defined as a unit circle centred on the origin. Examples of the two coarsest meshes are given in Figures A.4 and A.5 in Appendix A. The test function used was \( T(x, y) = \cos\left(\frac{\pi}{4}x\right) \sinh\left(\frac{\pi}{4}y\right) \). A plot of the initial test function is given in Figure 5.29 for cell-centred control volumes. The boundary conditions used were

\[
T + \frac{4}{\pi} \frac{\partial T}{\partial n} = \left[\cos\left(\frac{\pi}{4}x\right) + x \sin\left(\frac{\pi}{4}x\right)\right] \sinh\left(\frac{\pi}{4}y\right) - y \cos\left(\frac{\pi}{4}x\right) \cosh\left(\frac{\pi}{4}y\right), \quad (5.2)
\]

for \( \theta = [0, \pi) \)

for the upper half of the circle, and the lower half is divided into two equal parts with the lower left quarter set to the exact Dirichlet condition

\[
T = \cos\left(\frac{\pi}{4}x\right) \sinh\left(\frac{\pi}{4}y\right), \quad \text{for} \quad \theta = \left[\frac{\pi}{2}, \frac{3\pi}{2}\right) \quad (5.3)
\]
Figure 5.30: Flux error for the cell-centred control volume scheme on a circle mesh

Figure 5.31: Flux error for the vertex-centred control volume scheme on a circle mesh

Figure 5.32: $L_2$ error norm plots for the cell-centred control volume scheme on a circle mesh

Figure 5.33: $L_2$ error norm plots for the vertex-centred control volume scheme on a circle mesh

Table 5.7: Error Norms on a Circle Mesh

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th>Vertex-Centred Control Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$2^{nd}$-Order</td>
<td>$3^{rd}$-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>2.00</td>
<td>3.14</td>
</tr>
<tr>
<td>$L_2$</td>
<td>2.00</td>
<td>3.20</td>
</tr>
<tr>
<td>$L_\infty$</td>
<td>1.81</td>
<td>2.87</td>
</tr>
</tbody>
</table>
and the lower right quarter set to the exact Neumann condition

\[
\frac{\partial T}{\partial n} = \frac{\pi}{4} x \sin \left(\frac{\pi}{4} x\right) \sinh \left(\frac{\pi}{4} y\right) - \frac{\pi}{4} y \cos \left(\frac{\pi}{4} x\right) \cosh \left(\frac{\pi}{4} y\right), \quad \text{for} \quad \theta = \left[\frac{3\pi}{2}, 2\pi\right) \quad (5.4)
\]

The Laplacian of the initial function is zero, so the exact flux is also zero everywhere in the mesh.

Figures 5.30 and 5.31 show the flux integral error for the Cell and Vertex schemes respectively on the coarsest mesh. The error for the interior control volumes is smaller than the error in the boundary control volumes. As the mesh is refined, the error in the boundary control volumes decreases, more quickly than the interior error. Figures 5.32 and 5.33 and Table 5.7 show the orders of accuracy for the \(L_1\) and \(L_2\) error norms for each control volume type. The \(L_2\) error norms for the vertex-scheme at third-order accuracy are approaching fourth-order. However, the \(L_\infty\) error norms for second- and fourth-order accuracy are somewhat lower than desired. The reason for this is currently unknown, but may be typical since these results were also obtained running different test cases.

### 5.6.2 Cardioid Mesh Boundary

The more complex cardioid mesh boundary was defined with the polar function \(r = 1 + \cos \theta\). The coarsest mesh is given in Figure A.6 in Appendix A. What makes this curved boundary mesh interesting is the cusp located at the origin. The mesh is refined as it approaches the cusp to accurately compute the flux integral. The test function used over this mesh was \(T(x, y) = \sin \left(\frac{x}{4}\right) \sinh \left(\frac{y}{4}\right)\). A plot of the initial test function is given in Figure 5.34 for
Figure 5.34: Initial function of $\sin \left( \frac{\pi}{4}x \right) \sinh \left( \frac{\pi}{4}y \right)$ for vertex-centred control volumes on a cardioid mesh

vertex-centred control volumes. The boundary conditions used were

$$T + \frac{4}{\pi} \frac{\partial T}{\partial n} = \left[ \sin \left( \frac{\pi}{4}x \right) - x \sin \left( \frac{\pi}{4}x \right) \right] \sinh \left( \frac{\pi}{4}y \right) - y \sin \left( \frac{\pi}{4}x \right) \cosh \left( \frac{\pi}{4}y \right)$$

(5.5)

for the upper half of the cardioid,

$$\frac{\pi}{4} T + \frac{\partial T}{\partial n} = \frac{\pi}{4} \left\{ \left[ \sin \left( \frac{\pi}{4}x \right) - x \sin \left( \frac{\pi}{4}x \right) \right] \sinh \left( \frac{\pi}{4}y \right) - y \sin \left( \frac{\pi}{4}x \right) \cosh \left( \frac{\pi}{4}y \right) \right\}$$

(5.6)

for the lower half. The Laplacian of the initial function is zero, so the exact flux is also zero everywhere in the mesh.

Figures 5.35 and 5.36 show the flux integral error for the Cell and Vertex schemes respectively on the coarsest mesh. The error is greatest in the boundary control volumes where the initial solution is largest. As the mesh is refined, the error in the boundary control volumes decreases. Figures 5.37 and 5.38 and Table 5.8 confirm the orders of accuracy for most of the error norms for each control volume type. However, the $L_{\infty}$ error norms are once again somewhat low at most orders of accuracy, for both control volume types. The $L_{2}$ error norm at fourth-order accuracy for the vertex-centred control volume is also too low. Again, the
CHAPTER 5. VALIDATION

Figure 5.35: Flux error for the cell-centred control volume scheme on a cardioid mesh

Figure 5.36: Flux error for the vertex-centred control volume scheme on a cardioid mesh

Figure 5.37: $L_2$ error norm plots for the cell-centred control volume scheme on a cardioid mesh

Figure 5.38: $L_2$ error norm plots for the vertex-centred control volume scheme on a cardioid mesh

Table 5.8: Error Norms on a Cardioid Mesh

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th>Vertex-Centred Control Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2nd-Order</td>
<td>3rd-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>1.98</td>
<td>2.97</td>
</tr>
<tr>
<td>$L_2$</td>
<td>1.97</td>
<td>2.95</td>
</tr>
<tr>
<td>$L_\infty$</td>
<td>1.80</td>
<td>2.37</td>
</tr>
</tbody>
</table>
CHAPTER 5. VALIDATION

reason for this is unknown.

5.7 Summary

The finite-volume flux scheme was successfully validated for the interior control volumes. When Dirichlet boundary conditions were introduced, the Taylor Constraints method gave erratic results while the Gauss Constraints method provided clean, high-order accurate flux integrals. The Gauss Constraints method did provide problems at boundaries where the gradients are high — leading to second-order accurate results regardless of the desired order of accuracy. Verification continued as the scheme was executed over meshes with curved boundaries. For both the circle mesh and the more complex cardioid mesh, the scheme produced high-order accurate flux integrals.
Chapter 6

Application

The previous chapters have demonstrated a high-order finite-volume method for the Laplace equation. What remains to be shown is that the scheme can handle the viscous terms of the Navier-Stokes' equations to high-order accuracy. This chapter describes a modified advection-diffusion problem. It is a scalar model problem which has the same structure as the Navier-Stokes' momentum equations.

6.1 Modified Advection-Diffusion Problem

Recall from Chapter 2 the Navier-Stokes' equations:

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{6.1}
\]

\[
\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla P + \left[ \nabla \cdot \mathbf{T} \right] + F_B \tag{6.2}
\]

\[
\frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u} (E + P)) = \left[ \nabla \cdot (\mathbf{T} \cdot \mathbf{u}) \right] + \left[ \nabla \cdot (\kappa \nabla T) \right] \tag{6.3}
\]
where \((u \otimes u)_{ij} = u_i u_j\)

\[
\tau_{ij} = 2\mu \left(D_{ij} - \frac{1}{3} \delta_{ij} D_{mm}\right)
\]

\[
D = \frac{1}{2} \left(\nabla u + (\nabla u)^T\right) \quad \text{or} \quad D_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)
\]

\[
F_B = \text{net vector body force for the control volume}
\]

The viscous terms of the Navier-Stokes' equations (the boxed terms) are of the form \(\nabla \cdot \tau\), where \(\tau\) for three dimensions is:

\[
\tau = 2\mu \begin{bmatrix}
\frac{2}{3} \frac{\partial u}{\partial z} - \frac{1}{3} \left(\frac{\partial u}{\partial y} + \frac{\partial u}{\partial x}\right) & \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\
\frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \frac{2}{3} \frac{\partial v}{\partial y} - \frac{1}{3} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial z}\right) & \frac{1}{2} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\
\frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \frac{1}{2} \left(\frac{\partial w}{\partial z} + \frac{\partial w}{\partial y}\right) & \frac{2}{3} \frac{\partial w}{\partial z} - \frac{1}{3} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial y}\right)
\end{bmatrix}
\]

The scalar model problem used to validate the claim that the viscous terms can be evaluated is:

\[
\frac{\partial T}{\partial t} + a \frac{\partial T}{\partial x} + b \frac{\partial T}{\partial y} + \mu_1 (\nabla^2 T) + 2\mu_2 \frac{\partial^2 T}{\partial x \partial y} = 0,
\]

where \(a, b, \mu_1\) and \(\mu_2\) are constants.

### 6.2 Applying the Finite-Volume Scheme

The flux integral for this model problem is derived by using the same method as for the Laplace equation. Equation 6.5 is integrated over the control volume. The Divergence Theorem is then applied to obtain the flux integral shown in equation 6.6, where \(\hat{n}\) is the unit vector normal to the edge of the control volume. The \(\frac{\partial^2 T}{\partial z \partial y}\) term cannot be handled...
uniquely, and is broken up into two terms with the Divergence Theorem.

\[
\text{Flux Integral} = \oint_{\partial(CV)} \left( aT + \mu_1 \frac{\partial T}{\partial x} + \mu_2 \frac{\partial T}{\partial y}, bT + \mu_1 \frac{\partial T}{\partial y} + \mu_2 \frac{\partial T}{\partial x} \right) \cdot \hat{n} \, ds \tag{6.6}
\]

The flux integral is evaluated using the same technique as for the Laplace equation. A least-squares reconstruction is performed, including the Gauss constraints specified by the boundary conditions to give a Taylor polynomial approximation to \( T \) in each control volume. From this the value of \( T \) and \( \nabla T \) can be determined at the Gauss points. Along control volume boundaries, the area-weighted averages of \( T \) and \( \nabla T \) are used to calculate the flux. The flux integral for each edge segment of the control volume is calculated using Gauss quadrature and added to the control volumes' total flux integral.

### 6.3 Validation

This section demonstrates the modified advection-diffusion flux integral can be computed to high-order accuracy on a square mesh. The numerical flux integral is compared to the exact flux integral, calculated to sixth-order accuracy by integrating equation 6.5 over the control volumes. The error norms are calculated and plotted to demonstrate that each order of accuracy can be obtained.

The test function used to validate the modified advection-diffusion problem was \( T(x, y) = \sin(\pi x) \cos(\pi y) \). A picture of the initial function is shown in Figure 6.1 for cell-centred control volumes. The boundary conditions used were Dirichlet, \( T = \sin(\pi x) \cos(\pi y) \), along the top and bottom boundaries and Neumann, \( \frac{\partial T}{\partial n} = \sin(\pi x) \cos(\pi y) \), along the left and right boundaries. The constants \( a, b, \mu_1 \) and \( \mu_2 \) are all equal to one.
CHAPTER 6. APPLICATION

Figure 6.1: Initial function of $\sin(\pi x)\cos(\pi y)$ for the modified advection-diffusion problem for cell-centred control volumes

Figures 6.2 and 6.3 show the flux integral error for the cell- and vertex-centred control volumes respectively. In each case, the error in the boundary control volumes is comparable to that in the interior control volumes. The $L_2$ error norm plots in Figures 6.4 and 6.5 show the correct relationship between the error norm and the number of control volumes for each control volume type. All the orders of accuracy are validated by the three error norms shown in Table 6.1.
CHAPTER 6. APPLICATION

Figure 6.2: Flux error for the cell-centred control volume scheme for the modified advection-diffusion problem

Figure 6.3: Flux error for the vertex-centred control volume scheme for the modified advection-diffusion problem

Figure 6.4: $L_2$ error norm plots for the cell-centred control volume scheme for the modified advection-diffusion problem

Figure 6.5: $L_2$ error norm plots for the vertex-centred control volume scheme for the modified advection-diffusion problem

Table 6.1: Error Norms for the Modified Advection-Diffusion Problem

<table>
<thead>
<tr>
<th>Error Norm</th>
<th>Cell-Centred Control Volumes</th>
<th>Vertex-Centred Control Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$2^{nd}$-Order</td>
<td>$3^{rd}$-Order</td>
</tr>
<tr>
<td>$L_1$</td>
<td>2.16</td>
<td>3.21</td>
</tr>
<tr>
<td>$L_2$</td>
<td>2.16</td>
<td>3.25</td>
</tr>
<tr>
<td>$L_{\infty}$</td>
<td>1.98</td>
<td>2.90</td>
</tr>
</tbody>
</table>
Chapter 7

Summary and Conclusions

A fourth-order accurate diffusive flux calculation scheme has been developed which can be incorporated into any existing reconstruction-based unstructured mesh inviscid flow solver. Commonly used finite-volume methods were examined, concluding that a new method is needed to easily calculate high-order accurate flux integrals, using the same least-squares reconstruction method as with the inviscid scheme in order to avoid performing multiple time-consuming reconstructions. Using the Laplace equation as a model of the viscous terms of the Navier-Stokes' equations, a scheme was developed to calculate flux integrals to high-order accuracy by using data from a high-order accurate least-squares reconstruction. The boundary conditions are enforced during the reconstruction, which requires boundary constraints to be included in the least-squares matrix. Initially the Taylor Constraints method, which approximates the boundary condition with a Taylor polynomial and matches it term by term to the Taylor polynomial approximation of the solution evaluated along the boundary edge, was developed to derive the boundary constraints for straight boundaries.
However, there were drawbacks: too many constraints made the constrained least-squares reconstruction matrix over-constrained, the boundary condition had to be $C^3$, and extending the scheme to curved boundaries was complex. The Taylor Constraints method was discarded and the simpler Gauss Constraints method was developed that eliminated the drawbacks. The Gauss Constraints method simply equates the reconstruction polynomial at the boundary Gauss points to the boundary condition at that point.

The flux integration was validated by demonstrating high-order accuracy on square, circle and cardioid meshes with a combination of Dirichlet and Neumann boundary conditions. During testing two cases were revealed which cause problems with the flux integration scheme: initial solutions which have large gradients at the boundaries produce large errors locally at those boundaries, but still maintain high-order accuracy, and singular points along the boundary cause the high-order scheme to fail. The flux integration scheme was shown to be high-order accurate for a modified advection-diffusion problem of the same form as the Navier-Stokes' momentum equations. This demonstrates that the developed flux integral calculation scheme can be used to evaluate the viscous terms of the Navier-Stokes' equations.

The rest of this chapter describes the important issues that arose during development of the code. Conclusions are then made about the scheme, describing the limitations and comparing different aspects of the two different control volume types, including memory usage and CPU time.
CHAPTER 7. SUMMARY AND CONCLUSIONS

7.1 Issues During Development

There are several areas in the code which must be implemented precisely correctly in order to achieve and verify high-order accuracy. Each of the items listed below was at some point a significant problem. They have been listed here in the hopes that others may avoid them.

1. The boundary vertices of the generated meshes must lie precisely on the geometry’s boundary. The mesh file must also contain enough significant digits for the boundary vertices. Without either of these, attempts at verifying that the flux integral is high-order accurate will fail. The Gauss points on the boundary are computed from the boundary vertices. If the boundary’s vertices are not accurate enough, the Gauss points will be in the incorrect place. This causes a small error in the high-order solution, but causes big problems during verification, as the computations of the initial and exact solutions are also affected.

2. Likewise, the location of the Gauss points must be computed accurately so they lie on the straight or curved boundary of the geometry.

3. Gauss quadrature for integration of the moments along the curved mesh boundaries must be performed with respect to the correct variable. The accuracy of the reconstruction, and hence flux integral calculation, absolutely depends on this at high orders of accuracy. In the code used for this thesis, the moments are integrated analytically in $x$ and numerically in $y$. It is important to compute the Gauss points and weights with respect to $y$ and not along the arc parameter $s$, as done for the flux integration.
During validation, the mean values of the initial solution in the boundary control volumes of the mesh need to be calculated in the same way as the moments. In either case, if the integration is done with respect to s, only second-order accuracy can be obtained.

4. During validation, the initial function and its exact flux function (the Laplace or advection-diffusion operator applied to the initial solution) must be correct with respect to each other. If they are not, the computed error norms will not give the correct order of accuracy and the plots of the flux integral error will be erratic. This was detected when the interior control volumes gave large flux integration errors when they were already validated to be correct. As part of the testing code, the analytic integration with respect to x of both the initial function and its flux function must also be done correctly, otherwise similar problems will occur.

5. Another technicality in validating the code is to ensure the specified boundary conditions match the initial solution. If this is not done, the flux integration error will be large where the boundary condition is incorrect, possibly leading to the false conclusion that the boundary flux integration code is incorrect. Also the boundary conditions cannot have any discontinuities along the boundary, otherwise results such as the validation case in Section 5.5 will occur.
7.2 Computer Resource Usage

It is difficult to make conclusions about which control volume is preferable. In addition to accuracy, memory usage and computation time need to be weighed carefully.

When memory usage is considered, the cell-centred control volumes are the better choice. Memory usage was computed for two of the sample square meshes used during testing; size statistics for each mesh are listed in Table 7.1. The finer mesh is about 66 times larger than the coarser mesh.

Table 7.1: Mesh Information for Square Meshes

<table>
<thead>
<tr>
<th>Mesh Name</th>
<th># Cells</th>
<th># Edges</th>
<th># Boundary Edges</th>
<th># Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>square1.mesh</td>
<td>550</td>
<td>857</td>
<td>64</td>
<td>308</td>
</tr>
<tr>
<td>square4.mesh</td>
<td>37726</td>
<td>56841</td>
<td>504</td>
<td>19116</td>
</tr>
</tbody>
</table>

There are two main data structures which consume memory: the coefficients of the reconstructed solution and the mesh data structure. Table 7.2 shows the approximate amount of memory in kilobytes used for each order of accuracy by each mesh. It is important to note that these can be considered an upper bound for memory usage, as the data structures used in the thesis code are by no means optimal. The reconstruction data structures increases in size as the order of accuracy increases, since the number of terms, and hence coefficients, of the Taylor polynomial approximation increases. In the mesh data structure the number of moments increases with order of accuracy, as does the number of neighbouring control volumes for each individual control volume.

The cell-centred control volumes use less memory than the vertex-centred control vol-
UMES. And since there are about twice as many cells as vertices, cell-centred control volumes are much less memory intensive per control volume. The vertex-centred control volumes use significantly more memory for fourth-order accuracy than for third-order accuracy. This is because the interior Gauss points for the cell-centred case are computed during the flux integral calculation, while the interior Gauss information for the vertex-centred case is computed during preprocessing and stored in the mesh data structure. The Vertex scheme could use less memory by computing the Gauss information on the fly like the Cell scheme, but this would increase the CPU time.

![Figure 7.1: Plot of number of control volumes versus the amount of memory used for each control volume scheme.](image-url)
CHAPTER 7. SUMMARY AND CONCLUSIONS

Figure 7.1 shows the relationship between the order of accuracy and amount of memory used by each control volume scheme. The memory increase between meshes is approximately linearly proportional to the number of control volumes, about 65 times for cell-centred control volumes and 62 times for vertex-centred control volumes.

Overall memory use may or may not be an issue, depending on the amount of memory in the computer and the size of the finest mesh for the problem's geometry.

Although the Vertex scheme uses more memory, it executes more quickly than the Cell scheme. Figures 7.3 and 7.2 show the relationship between desired orders of accuracy and the CPU time to compute the flux integrals for the two different schemes. The times were recorded for the sinusoidal test case on the finest square mesh used in Section 5.2.2. A Pentium 233 MHz personal computer, running the Windows98 operating system, with 40 MBytes of RAM was used to perform the calculations. As with memory usage, the times can be considered an upper bound, as efficiency was not taken into consideration during implementation. However, implementation is similar between the two schemes, and thus a basic comparison can be made.

In each figure the lines cross at approximately the same point. After this point, fourth-order accurate methods reach lower error norms more quickly than second- and third-order accurate methods. This strongly suggests that solving problems over coarse meshes to high-order accuracy is more efficient than solving to second-order accuracy over fine meshes.

As the order of accuracy increases, the CPU time to compute the flux integral over cell-centred control volumes essentially doubles. The vertex-centred scheme's times increase
linearly (a factor of one) with order of accuracy. The main explanation for the time difference between control volumes types is that there are fewer vertex-centred control volumes than cells in any mesh — roughly half as many. Because the most time consuming part of the calculation is the reconstruction over each control volume, having fewer control volumes to reconstruct greatly decreases the amount of CPU time needed.

Because the Cell scheme is working with twice as many control volumes as the Vertex scheme, it should be more accurate on a given mesh. When the CPU time difference is a factor of two or less, the cell scheme is more efficient on a per control volume basis.

One last consideration is the possibility of extending the two-dimensional schemes to three dimensions. The details for cell-centred control volumes in three dimensions are far easier than for vertex-centred control volumes because the flux integration for oddly-shaped control volumes is quite complex.
CHAPTER 7. SUMMARY AND CONCLUSIONS

7.3 Direction for Future Work

The flux integral of a modified advection-diffusion equation in a form similar to the Navier-Stokes' momentum equations was calculated to high-order accuracy, indicating the developed diffusive flux scheme can be incorporated into Dr. Carl F. Ollivier-Gooch's existing inviscid compressible flow solver to evaluate the viscous terms and make it a viscous compressible flow solver.

This could be done by performing the constrained least-squares reconstruction and then using the approximate control volume solutions to calculate the flux integrals of each of the terms (inviscid and viscous) in the Navier-Stokes' equations in a manner similar to the modified advection-diffusion example.

The diffusive flux scheme has been developed and validated only for scalar quantities. This can be a problem for boundary conditions which are a function of more than one unknown (e.g. no-slip wall). One way the scheme can be expanded for coupled boundary conditions is to include all of the unknown coefficients of both Taylor polynomial approximations of the variables in the unknown vector of the least-squares problem for the reconstruction. This essentially doubles the size of the constrained least-squares matrix, as well as the total number of constraints, as all the unknown coefficients are solved simultaneously. This coupled reconstruction problem will be very sparse, which can be taken advantage of to help the efficiency of the reconstruction.

It was shown in Section 5.5 that the scheme does not work well if there are singular points in the solution. The error in the control volumes around the singular point are larger
than elsewhere in the solution. This problem might be solved using a different coordinate system (e.g. polar coordinates) around the problem area.

The boundary constraint scheme is also currently limited to linear combinations of Dirichlet and Neumann boundary conditions, although other boundary conditions, such as radiation or acceleration, can be derived by similar methods.

Extending the diffusive finite-volume scheme to three dimensions is, in principle, straightforward. The flux integral is now an integral through the faces of the control volumes, as opposed to the edges. Gauss quadrature can still be performed with Gauss points located on the surface of the face and the Gauss Constraints method can still be used to add constraints to the reconstruction matrix. However implementation must be done very carefully in order to avoid the problems described in Section 7.1.
Bibliography


Appendix A

These are some of the meshes used in validating the flux integral calculations. They are the meshes coarse enough to be viewable when printed:

Figure A.1: square0.mesh — 145 cells, 89 vertices
Figure A.2: square1.mesh — 550 cells, 308 vertices

Figure A.3: square2.mesh — 2280 cells, 1204 vertices
Figure A.4: circle0.mesh — 414 cells, 226 vertices

Figure A.5: circle1.mesh — 1617 cells, 854 vertices
Figure A.6: cardioid0.mesh — 1673 cells, 879 vertices