Multi-Moment Advection Schemes for Cartesian Grids and Cut Cells

Richard J. Ferrier

Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2013

To my parents,
Carol and Hugh

Abstract

Computational fluid dynamics has progressed to the point where it is now possible to simulate flows with large eddy turbulence, free surfaces and other complex features. However, the success of these models often depends on the accuracy of the advection scheme supporting them. Two such schemes are the constrained interpolation profile method (CIP) and the interpolated differential operator method (IDO). They share the same space discretisation but differ in their respectively semi-Lagrangian and Eulerian formulations. They both belong to a family of high-order, compact methods referred to as the multi-moment methods.

In the absence of sufficient information in the literature, this thesis begins by taxonomising various multi-moment space discretisations and appraising their linear advective properties. In one dimension it is found that the CIP/IDO with order $(2N-1)$ has an identical spectrum and memory cost to the Nth order discontinuous Galerkin method. Tests confirm that convergence rates are consistent with nominal orders of accuracy, suggesting that CIP/IDO is a better choice for smooth propagation problems. In two dimensions, six Cartesian multi-moment schemes of third order are compared using both spectral analysis and time-domain testing. Three of these schemes economise on the number of moments that need to be stored, with one CIP/IDO variant showing improved isotropy, another failing to maintain its nominal order of accuracy, and one of the conservative variants having eigenvalues with positive real parts: it is stable only in a semi-Lagrangian formulation. These findings should help researchers who are interested in using multi-moment schemes in their solvers but are unsure as to which are suitable.

The thesis then addresses the question as to whether a multi-moment method could be implemented on a Cartesian cut cell grid. Such grids are attractive for supporting arbitrary, possibly moving boundaries with minimal grid regeneration. A pair of novel conservative fourth order schemes is proposed. The first scheme, occupying the Cartesian interior, has unprecedented low memory cost and is proven to be conditionally stable. The second, occupying the cut cells, involves a profile reconstruction that is guaranteed to be well-behaved for any shape of cell. However, analysis of the second scheme in a simple grid arrangement reveals positive real parts, so it is not stable in an Eulerian formulation. Stability in a hybrid formulation remains open to question.

Acknowledgements

Foremost, I would like to express my gratitude to my supervisor, Professor David Ingram, who allowed me to pursue this research with autonomy. It was a great gift that will not be forgotten. His guidance was always valuable, and I enjoyed our discussions about all things numerical. Most of all he provided enthusiasm, optimism and gentle encouragement at times when they were sorely needed.

My second supervisor, Dr. Vengatesan Venugopal, played an important role in making sure my project progressed as it should. His door was always open. I am further indebted to Liz Paterson and Sue Simpson of the Graduate School of Engineering for their assistance.

At Hokkaido University, Dr. Yasunori Watanabe gave me important insights into his work from which this project grew. He and Dr. Ayumi Saruwatari were the most gracious hosts one could ask for.

Everyone at the Institute for Energy Systems made it a joyous place to work. Those particularly relevant to my project were: Mathew Topper, for his words of wisdom and the excellent LaTeX template by which this document is typeset; Bill Edwards, for his cautionary advice that I probably should have heeded more closely; Bruce Duncan, *especially*, for his unfailing humour in addressing my many software-related questions; Angus Creech, for expressing such interest in my work and generously lending his support at every turn; and Pauline Clark, for holding the place together and helping me out at critical times.

My friends – Nicol, Ania (both of them), Mark, Ben and Julie – are credited with preserving my sanity. Finally I would like to thank my parents, my brother and my sister, for their love and support in whichever direction I ended up taking.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Richard James Ferrier

Contents

Appendices

Figures and Tables

Figures

Tables

Nomenclature

Symbols

$a, \mathbf{a}$	wave velocity (scalar, Euclidean vector)
$\mathbf{A}$	Vandermonde matrix
$\mathbf{c}, \mathbf{C}$	profile coefficients (column vector or vectors)
G	amplification factor
i, j	grid point/element indices
I	imaginary unit, $\sqrt{-1}$
$\mathbf{I}$	identity matrix
k	relative time level; wavenumber, $2\pi/\lambda$
K	non-dimensional wavenumber, $k\Delta x$
K_c	maximum resolvable wavenumber (Hu *et al.*, 1999)
L^λ	generalised norm
m	number of cells; number of points minus one
n	time level; number of time steps
$\mathbf{n}$	surface normal unit vector
N	(discontinuous Galerkin) order of accuracy
p	polynomial degree
q	temporal order of accuracy; source term
$\mathbf{r}$	physical position (Euclidean vector); products of ξ and η (row vector)
S	cell or domain surface
Δt	time step size
$u, \mathbf{u}$	advected quantity (scalar, column vector)
V	cell or domain volume
$\mathbf{V}$	fluid velocity (Euclidean vector)
$w, \mathbf{w}, \mathbf{W}$	nondimensional weights (scalar, row vector, matrix)
$\Delta x, \Delta y$	grid cell size; point spacing
z	amplification factor eigenvalue

α,β	relative position of grid point/element
γ	grid cell aspect ratio, $\Delta y/\Delta x$
ε	error
$\Vert\varepsilon\Vert_{\lambda}$	generalised error norm or L^{λ}-error
$\Vert\varepsilon(K)\Vert^{\text{phys}}$	physical-mode error norm, see Equation (3.47)
η	nondimensional y-coordinate, $(y-y_{i,j})/\Delta y$
θ	direction of wave propagation
λ	wavelength; generalised norm power
ξ	nondimensional x-coordinate, $(x-x_{i,j})/\Delta x$
σ	Courant number, $a\Delta t/\Delta x$
ϕ	generic quantity; phase angle
ω	angular frequency
Ω	non-dimensional frequency, $\omega\Delta t$

Abbreviations

CIP	constrained interpolation profile method
CIP–CSL	CIP conservative semi-Lagrangian
CIP/IDO/MM–FVM	CIP/IDO/multi-moment finite volume method
CUL	cubic Lagrange scheme
DG	discontinuous Galerkin method
FDM	finite difference method
FEM	finite element method
FOU	first order upwind
FVM	finite volume method
HWENO	Hermite weighted essentially non-oscillatory method
IDO	interpolated differential operator method
MCV	multi-moment constrained finite volume method
SOC	second order central
TEC	time evolution converting formula
TOU	third order upwind
VSIAM	volume- and surface-integrated average multi-moment method

Chapter 1

Introduction

1.1 Computational Fluid Dynamics

In the engineering industry, it is often necessary to make an assessment of strength or performance. One might need to assess the aerodynamic lift on an aircraft wing, the energy losses in a pipeline, or the forces inflicted on a structure by water currents. All of these are examples of fluid dynamics, which is the study of the motion and forces of liquids and gases. Sometimes the problem can be reduced to a simple calculation based on theory or empirical rules; for instance, the energy losses in a pipeline will be proportional to the square of the flow rate. In more complex cases, experiments must be carried out using scale models. Wind tunnel testing is the obvious example. In all cases engineers will try to use both theory and experiment to gain confidence in their method.

Computational fluid dynamics (CFD) is a relatively new addition to the range of techniques available to the engineer. CFD involves simulating flows using a computer, so instead of constructing a physical experiment the engineer now constructs a numerical one. Much of the groundwork of CFD development was done in the early 20th Century with, for example, Richardson (1922) and Thom (1933) applying the finite difference method to problems in fluid dynamics, and Courant, Friedrichs, and Lewy (1928) providing a particularly important work on numerical stability. But it was not until the arrival of digital computing in the 1940s that CFD development took off. The Cold War, and the need to simulate nuclear explosions, drove much of development in the mid-20th Century, culminating in the very first CFD simulations by Los Alamos National Laboratory's T-3 group in the late 1950s. The aerospace industry became involved in the 1970s, with developments focusing on transonic and other flows relevant to flight. By the 1980s the wider engineering industry had taken an interest, with CHAM Ltd. releasing the first general-purpose CFD package, PHOENICS, in 1981, and Creare Inc. releasing FLUENT in 1983. In the last few decades, as computer processing power has become faster and cheaper, the attraction of CFD as an alternative to physical testing has become more apparent. In the aerospace industry, the increased use of CFD has even been accompanied by a decrease in the number of wind tunnel tests being carried out (Hirsch, 2007, p. 7).

CFD is made possible through the Navier–Stokes equations, the equations governing fluid flow. The equations are famously unsolvable analytically for all but the simplest flow conditions and geometries – for example, non-turbulent flow in a straight pipe or between parallel plates. However, when the geometry of interest is broken up or 'discretised' into a grid of cells, the equations can be applied to the cells individually. For each cell, an approximate solution is calculated using the information from neighbouring cells. The result is a simulated flow field that evolves in time. Quantities such as force and changes in energy are easily derived from the flow field. The finer the grid and time increment used, the more accurate the overall solution, but the higher the computational burden.

To gain confidence in the results of a simulation, a CFD practitioner will generate results on a coarse grid and then successively refine the grid until the flow field is seen to converge to a solution. If the grid is well designed and the flow reasonably well behaved enough, the practitioner can even quantify the errors associated with the grid resolution. A number often quoted in numerical methods is the order of accuracy. This refers to how a refinement of the grid or time step affects the reduction in the solution error. For a solver with first order spatial accuracy, halving the cell size in one dimension also halves the error. With second order accuracy, the error is reduced to a quarter. With arbitrary Nth order accuracy, the error is reduced to $^1/_{2^N}$.

Commercial CFD solvers typically implement up to second order methods in space. There is a certain mathematical convenience behind this that will be explained in due course. While on the surface this seems like a good deal – a factor two refinement in the grid yielding a factor four reduction in the errors – in practice, grids are not one-dimensional but three-dimensional. This means a factor two grid refinement also requires a *factor eight* increase in the number of cells. It does not take long to exhaust the memory and processing capabilities of the hardware, unless the computation is parallelised on a sufficiently large cluster of machines. Clearly there is a strong case to be made for having higher-order methods. Of course, such methods carry their own computational expense, and this must be taken into account when developing any new CFD code.

1.1.1 The Importance of Advection

Today engineers enjoy access to commercial solvers boasting capabilities in modelling many different kinds of flow. Yet these capabilities offer a dangerous sense of security to engineers seeking to simulate more unusual flows. The existence of turbulence poses a particular threat to the reliability of solutions. Turbulence is a stochastic phenomenon involving a spectrum of different-sized eddies. To resolve the smallest eddies on a computer becomes prohibitively expensive, and CFD researchers have introduced the so-called Reynolds Averaged Navier Stokes (RANS) turbulence models to keep simulations economical. These models take advantage of the emergent properties of turbulence, such as its apparent viscosity, to reduce the computational burden. However, while the models have been used successfully for decades in the established (oil and gas, aeronautical) industries, they are based on assumptions that do not hold up under certain conditions. It is the responsibility of the CFD practitioner to identify when and where to deploy these models.

Free surfaces – the interfaces between two or more phases such as air and water – pose another numerical challenge. The difficulty lies in the discontinuous jump in fluid properties across the interface. Conventional numerical methods depend on the flow field being smooth; discontinuities cause them to fail. Shock waves also exhibit a discontinuous jump. In commercial codes, the Volume Of Fluid method (VOF) is typically used to treat free surface flows. VOF computes the volume fraction of each phase filling the grid cells, then a separate algorithm reconstructs the interface using piecewise segments. The method preserves mass but occasionally suffers from pieces of fluid being non-physically ejected as flotsam and jetsam (Losasso *et al.*, 2006). VOF is good enough for problems where only the general shape of the interface is important.

When faced with complex interacting phenomena such as turbulence and free surfaces, researchers in the academic community tend to eschew general-purpose numerical methods like the RANS and VOF models in favour of specialised, high-fidelity ones. The simulation of breaking water waves is a case in point. The evolution of a breaking wave involves complex surface-vortex interactions that give rise to the formation of surface scars and finger jets (Brocchini and Peregrine, 2001). These jets may go on to experience break-up and coalescence, and air entrainment may cause the water to become drastically more compressible (Peregrine and Thais, 1996). RANS turbulence models are unable to reproduce the relevant vortices, while VOF is unsuitable because the reconstructed surface is too crude to represent finger jets, droplets and bubbles. Yet it is a worthwhile goal to be able to model breaking waves accurately using CFD, because it enables better assessments of extreme wave loading and overtopping of coastal structures, ship hulls and wave energy converters.

A CFD code developed at Hokkaido University, Japan is able to capture the surface-vortex interactions in breaking waves (Watanabe and Saeki, 1999; Saruwatari *et al.*, 2009). The code currently only treats a single incompressible fluid phase and does not take into account air entrainment, but the formation of finger jets is convincingly reproduced. A screenshot is shown in Figure 1.1. The method has three key features:

- Large eddy simulation (LES). This is a relatively expensive treatment of turbulence that filters the velocity field, resolves eddies on the same scale as the grid cells, and applies a stochastic model to the smaller eddies.
- The level set method. Popular in the computer graphics community, this is an interface capturing scheme that yields smoother free surfaces than VOF. A weakness is that it is unable to preserve fluid mass. The level set implicitly handles complex topological changes such as break-up and coalescence (Losasso *et al.*, 2006).
- The constrained interpolation profile method (CIP), a third order advection scheme. Conventional advection schemes have a tendency to generate non-physical oscillations when extended to higher orders; CIP to a large extent avoids this.

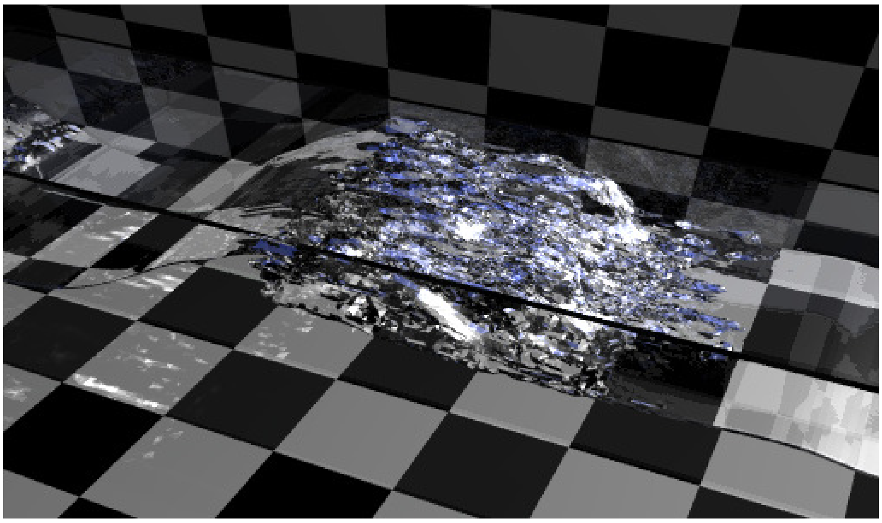


Figure 1.1: Breaking wave simulated using code developed at Hokkaido University. Courtesy of D. M. Ingram, Y. Watanabe, and A. Saruwatari.

Why does the advection scheme play such an important role in this solver? It turns out that the fidelity of the advection scheme underpins the success of the other two features. Advection, also known as convection or transport, is a prominent term in the Navier–Stokes equations. Usually the equations can be simplified or posed in a different form depending on the whether the flow is predominantly compressible or incompressible, viscous or inviscid, steady or transient. But unless the viscosity is predominant and the flow can be classed as ‘creeping’, advection always appears. An advection scheme is therefore needed to model the transport of a given quantity.

In the Navier–Stokes equations, the advection term is nonlinear in the sense that the velocity of the fluid transports *itself.* It is this nonlinearity that causes turbulence. At a certain Reynolds number, when the inertia of the fluid is insufficiently damped by viscosity, a flow becomes unstable and develops a large eddy; this in turn goes on to induce smaller eddies, and so on. Resolution of the large eddies in LES depends directly on the advection scheme applied to the velocity field. And the same velocity field may be responsible for transporting other quantities. In the level set method, a scalar field representing distance from the free surface is created. The isosurface at zero thus represents the free surface. The velocity field transports the scalar, allowing pieces of each phase to move, merge together and pinch apart naturally. Like the evolution of the velocity field, the evolution of the scalar field depends on the applied advection scheme. And so when simulating the interactions between the free surface and subsurface eddies in a breaking wave, a high-order advection scheme applied to both quantities is crucial.

Similar studies have been carried out by Yang and Stern (2009) and Christensen (2006). Yang and Stern use a coupled level set/volume of fluid formulation that preserves mass and keeps the free surface smooth. The velocity field is treated with a third order advection scheme, while the level set field is treated with a fifth order scheme. Christensen uses VOF and a third order scheme. In both cases, LES is used.

1.1.2 Towards Non-Conforming Grids

The academic community has led the way in developing and applying high-order numerical methods for studying complex flow phenomena. But often these methods are not easily implemented on anything other than a Cartesian grid with rectilinear boundaries: the flow is confined to a box. Of the three solvers introduced in the previous section, only the solver belonging to Yang and Stern (2009) can accommodate arbitrary boundaries. By contrast, the numerical schemes of commercial CFD solvers are flexible enough to handle arbitrary grids without difficulty. This allows for some impressive grid generation capabilities.

Users of modern CFD packages may design structured grids that conform to the domain geometry. A 2D example is shown in Figure 1.2. The cells of a structured grid map straightforwardly with (i, j, k) addresses to an array stored on the computer. Because the relative positions of the cells are regular, operations are fast. However, such grids may be time consuming to design.

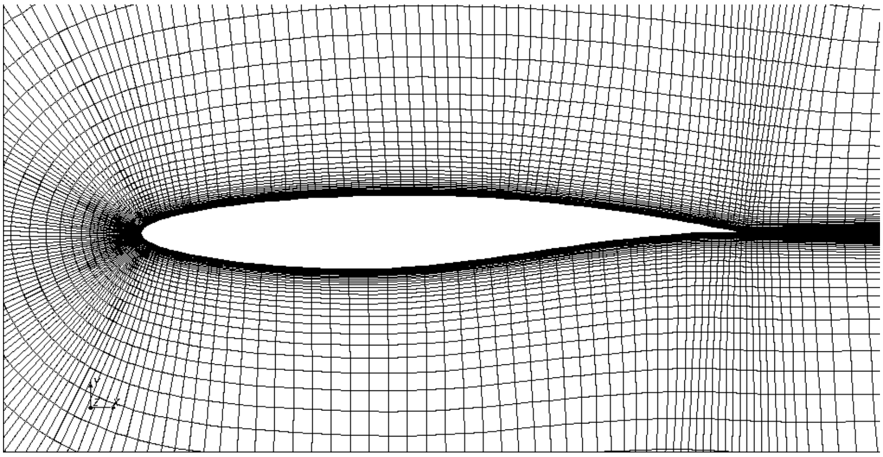


Figure 1.2: Example of an structured grid around a RAE2822 Airfoil. From the 'Transonic Flow over an Airfoil' tutorial in STAR-CCM+ Version 6.04.014.

Unstructured grids (Figure 1.3) are popular in that the user can fill a domain of any shape with cells at the click of a button. Features exist for controlling the local cell density, adding thin boundary cells, and so on. Unstructured grids are not foolproof. Poor quality cells can be generated that adversely affect the stability and accuracy of the solution, so the user must carefully check the quality of the grid at each generation. Furthermore, the simplicial (triangular/tetrahedral) cells commonly found in unstructured grids do not have particularly good numerical properties compared to their structured counterparts. And unstructured grids are computationally slower than structured grids, because operations must be preceded by a lookup of cell connectivity.

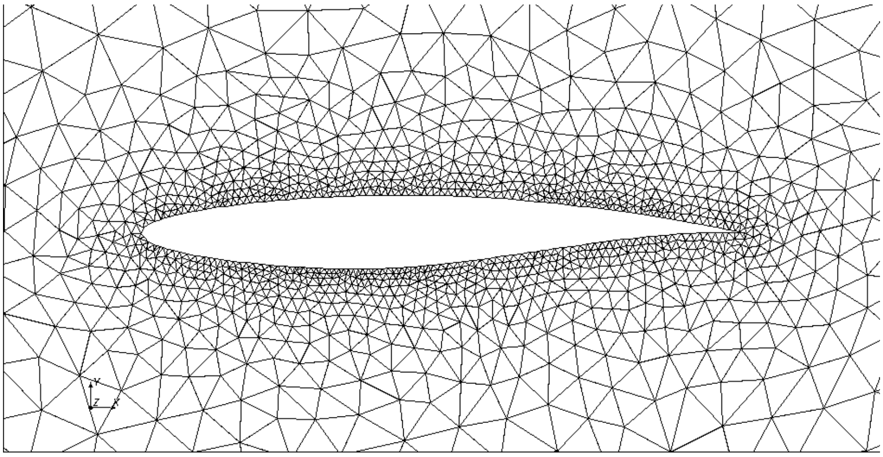


Figure 1.3: Example of an unstructured grid around a RAE2822 Airfoil. Generated using STAR-CCM+ Version 6.04.014.

Both grids are considered ‘body-conforming’ in that cells are fitted to the boundary. There is another class of grid, the ‘non-conforming’ grid, in which a boundary is superimposed upon a background grid of cells. The background grid is usually Cartesian, and so the literature sometimes refers to ‘Cartesian grid methods’. An example of such a grid is shown in Figure 1.4. The boundary cells, or flow values inside the boundary, must be given special numerical treatment. The interior can remain unmodified and structured.

In the example shown, the grid is also locally refined around the boundary. This demonstrates an immediate advantage over unstructured grids. Any practical grid should be able to support high cell density in regions where the flow field changes rapidly. This commonly happens when there is shear between regions of fast and slow moving fluid, for instance on the peripheries of wakes and jets. The grid needs to be refined locally rather than globally to avoid wasting cells. Cartesian cells can be split, in any of the x–y–z directions or isotropically, without the need for regenerating the whole grid and without adversely affecting the shape of the resulting cells.

The advantages of non-conforming grids go further. Suppose we wanted to model a solid body moving in the flow field. In a marine hydrodynamic context, this might be a wave energy converter. A structured grid would have to be designed so that the cells deform with the motion of the body. This may be difficult to implement and extreme motions could result in degenerate cells. An unstructured grid is easier to implement, but the need to regenerate it at every time step represents a severe runtime expense. A non-conforming grid is the most appropriate solution in that it could conceivably handle the incremental motion both automatically and with minimal modification to the cells and flow field.

The one objection frequently levelled at non-conforming grids is that they do not capture the boundary layers that develop on walls at high Reynolds numbers. By their viscous nature, fluids stick to walls; their velocity at the boundary is zero. Moving away from the boundary, the flow

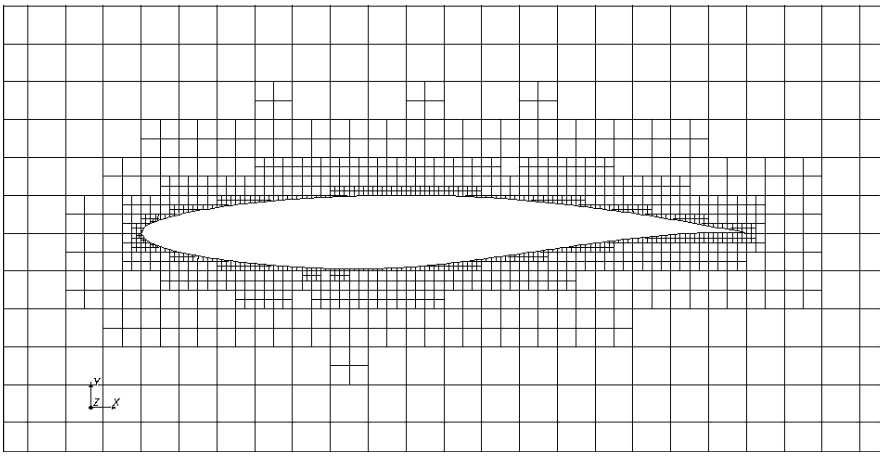


Figure 1.4: Example of an Cartesian non-conforming grid around a RAE2822 Airfoil. Generated using STAR-CCM+ Version 6.04.014.

is laminar (stratified) with increasing velocity until the flow transitions to turbulence. This happens over a relatively short distance because turbulent mixing is effective at transferring energy from the bulk of the flow to the near-wall fluid. We say there is a thin boundary layer. Unless the behaviour of the boundary layer is so predictable that it can be ignored or modelled empirically, thin cells are needed to resolve the high gradients down to the laminar sub-layer. To get an sense of how expensive this can become, consider water flowing in a 1 m diameter pipe at an average velocity of 1 m/s – a Reynolds number of approximately one million. To resolve the laminar sub-layer in this case would require cells less than 0.1 mm in height. Clearly this situation is out of the question for non-conforming grids where the local aspect ratio and orientation of cells cannot be controlled. A compromise, suggested by Coirier (1994), is to have a hybrid grid whereby thin prismatic cells are fitted to the body of interest and the outermost faces of these cells cut the Cartesian background grid.

There is great potential for Cartesian non-conforming grids. Dawes (2007), introducing a code called 'BOXER', calls for a new paradigm shift in CFD – full parallelisation of the process chain, from geometry import through gridding and solving to results visualisation. Dawes argues that traditional manipulation of grid elements represents a serial bottleneck in the process chain and is therefore an impediment in the design and analysis cycle. BOXER innovatively uses the level set method to represent the topology of the boundary. This opens the way for interactive volume sculpting, since adding and removing material is simply a case of manipulating the distance field. Gridding is minimal; the distance field allows prismatic cells to grow naturally out from the body before cutting a background Cartesian grid. And because this is all done in parallel, it offers the potential of rapid prototyping via simultaneous flow analysis and solid modelling – *flow sculpting*, as Dawes puts it.

1.2 Identification of a Need

The above discussion has shown that high-order advection schemes and arbitrary grids are two aspects of CFD that play important roles in the academic and commercial spheres, respectively. Clearly there is a need for crossover between the two areas. There is, moreover, a specific need which is of interest to researchers at the University of Edinburgh and Hokkaido University who are involved in coastal and offshore engineering.

The United Kingdom and Japan are two countries with regional similarities – maritime climates, an abundance of coastline – where coastal and offshore engineering has become a prominent research focus. It is in this field that the University of Edinburgh has enjoyed a long-standing collaboration with Hokkaido University. While the University of Edinburgh has conducted various studies on coastal structures and wave energy converters (Shiach *et al.*, 2004; Payne *et al.*, 2008a,b), Hokkaido University has, by virtue of the solver discussed in §1.1.1, exposed details of the underlying hydrodynamics (Watanabe *et al.*, 2008; Saruwatari *et al.*, 2009).

Present studies on violent wave overtopping at sea walls are worth noting. Such studies often solve the shallow water equations as an economical alternative to the full Navier–Stokes equations. In the study of Shiach *et al.* (2004), representation of the sea wall is achieved via the Cartesian cut cell method, a non-conforming gridding method. The study found that the shallow water equations provide an adequate prediction of volumetric overtopping and yet underpredict the number of overtopping events at the depth of water tested. On the other hand, Orszaghova *et al.* (2012) report favourable agreement between their model and physical experiments. In any case, prediction of complex flow features and forces from wave impacts would not be possible without a full Navier–Stokes simulation. Hokkaido University's solver would be an ideal choice if only the solver could handle arbitrary boundaries.

The applications of the solver could extend to analyses of wave energy converters such as the Pelamis and the Oyster. These devices are currently in their second generation. Wave energy converter developers are interested in bringing greater degrees of accuracy to their analyses using Navier–Stokes simulations, but commercial solvers are simply not up to the task. All wave energy converters will experience turbulent eddies, possibly shed from the solid boundary, interacting with the free surface. In the case of the Oyster, overtopping occurs and is serious enough to result in significant energy losses (Henry *et al.*, 2010).

A long-term aim, then, is to introduce arbitrary boundaries into Hokkaido University's solver via a non-conforming gridding method. And the first task is to implement its CIP advection scheme on a non-conforming grid, since the advection scheme underpins the fidelity and stability of the free surface and turbulence models.

1.3 Thesis Overview

The purpose of the present chapter has been to introduce the research aim, framing it in the wider context of CFD and discussing motivations on a more local scale. But the task of implementing the constrained interpolation profile method (CIP) on a non-conforming grid is not straightforward.

As Chapter 2 explains, the CIP scheme in question is just one of a number of variants known collectively as the multi-moment methods. Before attempting to adapt this particular scheme, it must be asked whether it is really the best candidate or whether one of the more recent variants would be more appropriate. The literature review will show that in fact not much is known about the properties of multi-moment schemes in relation to one another, let alone in relation to other modern methods. A priority is to make sure that the adaptation of a multi-moment scheme is relevant and viable in the first place. The review goes on to look at unstructured gridding – since it is a mature field that yields insights into high-order implementations – and non-conforming grids.

It becomes clear that a methodology is needed by which one can (a) appraise the performance of existing multi-moment schemes, and (b) design a stable and accurate scheme. To this end, Chapter 3 introduces the techniques of frequency-domain analysis, which uncovers spatial properties for linear wave advection, and time-domain testing, which verifies the order of accuracy for other various test cases. In addition to these established techniques, the use of a ‘physical-mode error norm’ is proposed. This error converts frequency-domain information into expected time-domain performance and therefore provides a useful means of verification. One-dimensional multi-moment schemes are analysed and tested in an introductory study, corroborating existing studies. The 1D techniques are extended to two spatial dimensions in preparation for the study of 2D multi-moment schemes.

The thesis then divides into two parts, with the first part concerning the properties of existing multi-moment schemes. In Chapter 4, comparisons are made with the discontinuous Galerkin method – the most popular compact method in the CFD literature – in one and two spatial dimensions. It is confirmed that the multi-moment approach is competitive, and furthermore an interesting connection between the methods is uncovered. In Chapter 5, spatial properties of six existing 2D Cartesian multi-moment schemes are analysed and compared. To verify the analysis, five of these schemes are further subjected to time-domain simulations. This chapter contains possibly the first comprehensive review of multi-moment methods in which multidimensional properties are quantified.

The second part of the thesis details the design of a novel cut cell multi-moment method. The design is based on an existing conservative, fourth order scheme for triangular cells rather than one of the early CIP variants. The cut cell method divides naturally into two schemes, with one scheme occupying the uncut background grid and the other occupying cut cells. The background scheme is designed in Chapter 6 according to the frequency-domain techniques previously established. This scheme economises on stored moments using an arrangement not previously considered in the literature. It also has remarkably high isotropy.

The scheme for the cut cells is more difficult. Being able to fit a polynomial profile to arbitrary cell shapes is central to the success of the scheme, and so Chapter 7 is mostly concerned with development of a profile fitting algorithm. It is shown that fitting a single polynomial over a cut cell is not a valid solution. A novel alternative is proposed: the cut cell is triangulated into simplices, with each simplex assigned a profile; but the profiles within the cut cell are constructed in such a way as to respect a single integration over the cell. It is then the cut cell that acts as the finite volume, with the simplices supporting high-order interpolations and flux reconstructions. Analysis of the second scheme in a simple grid arrangement reveals positive real parts, so it is not stable in an Eulerian formulation; however, stability in a hybrid formulation remains possible.

Further investigations are beyond the scope of the present work. Chapter 8, in addition to summarising the main findings of thesis, provides a road map for further work. The possible unconditional instability of the proposed cut cell scheme is taken into account: a novel concept that still exploits the cut cell paradigm is suggested.

Chapter 2

Literature Review

2.1 Preliminaries

Chapter 1 introduced a central aim for the present work, which was to implement the constrained interpolation profile method (CIP) or other multi-moment scheme on a Cartesian non-conforming grid. Exactly which multi-moment scheme and type of grid to use, and how to approach the implementation, are questions driving the literature review. But in order to structure the review appropriately, it is important to recognise that the themes of advection and gridding are actually part of a wider problem: how to discretise the flow field in space using a grid of points or cells. The different techniques give rise to methods for modelling partial differential equations.

This section provides the relevant background information that for brevity was left out of Chapter 1. The various approaches to modelling partial differential equations are introduced in the first subsection, §2.1.1. Some well-known issues associated with implementing a high-order method on non-rectilinear grids are then explained in §2.1.2. Following these two subsections is an outline of the main review, given in §2.1.3.

2.1.1 Methods for Modelling Partial Differential Equations

Here the following methods are introduced.

- Finite difference and finite volume methods;
- Finite element methods, including the discontinuous Galerkin method;
- Multi-moment methods;
- The spectral volume and spectral difference methods;
- The Hermite WENO schemes.

The finite difference and finite volume methods are the seminal methods in computational fluid dynamics, remaining popular to this day. However, it will be seen that their ‘non-compactness’ complicates their extension to higher orders of accuracy on irregular grids. The other items on the list are all compact methods applicable to advection, with the discontinuous Galerkin method being the most widely researched. The multi-moment methods, which include CIP, form the main focus of this thesis. The spectral difference and spectral volume methods join the discontinuous Galerkin and multi-moment methods as competing compact methods. The Hermite WENO schemes are not competitive by themselves, and so their relevance to this thesis is deprecated. But they are prominent in the literature and are included here for completeness.

Not included in this list is the flux reconstruction approach, which was introduced by Huynh (2007) and subsequently reduced to a family of linearly stable methods by Vincent *et al.* (2011b). This family is actually a superset which includes the nodal discontinuous Galerkin and spectral difference methods. A scalar parameter c controls stability and accuracy properties and determines which subset will be recovered. The analysis of Vincent *et al.* (2011a) suggests that accuracy is maximal when the discontinuous Galerkin method is recovered, while a less stringent stability condition results from other values of c. For brevity, our discussions will be limited to the discontinuous Galerkin and spectral difference methods.

In the following discussions, the term 'method' will be favoured when describing a general approach to discretising a differential equation or operator in some way. The term 'scheme' will be favoured when describing some subset or component of a method. For example, 'the CIP–CSL method may be implemented with a quadratic polynomial to yield a third order scheme'; or 'the discontinuous Galerkin method may be supplemented by a Hermite WENO scheme to maintain accuracy in the presence of discontinuities'. The distinction is somewhat arbitrary as the two terms appear to be interchangeable in the literature.

Finite Difference and Finite Volume Methods

The governing conservation laws – conservation of mass, momentum and energy – are usually represented as a set of partial differential equations, which describe the flow at a point. In the finite difference method (FDM), partial differentials are rewritten as a combination of Taylor series expansions involving neighbouring points. The algebraic terms form a finite difference, while the continuous higher order terms are discarded in a truncation error. In the truncation error, the dominant term is proportional to the grid spacing raised to some power, with the power indicating the spatial order of accuracy. If we want to increase the order of accuracy of a finite difference method, we must include more neighbouring points so that leading higher order terms vanish. The pattern of points supporting a finite difference is called the computational stencil (also known as the computational molecule, or simply the support).

In a variation on the finite difference method, the compact difference method (Lele, 1992) does not necessarily increase the size of the stencil to achieve higher accuracy. Rather, it takes advantage of the fact that the stencil associated with a given point differential overlaps identical stencils for neighbouring point differentials. From a global point of view, this overlap represents redundancy. It is possible to manipulate the weights on the point values when they are implicitly connected in a linear system of equations. Consequently it is possible to minimise the stencil size while maintaining the order of accuracy.

The computational stencil is an important factor when it comes to extending a method to irregular grids. The problem with finite differences is that they are impossible to implement on unstructured points without inconsistent stencils leading to conservation being violated. It explains the success of the finite volume method (FVM), which does not suffer from such a restriction. The finite volume method is based on the integral equations, which describe the flow in a control volume.

In the integral equations, partial differentials are replaced by surface integrals using Gauss' divergence theorem. The surface integrals represent fluxes, or the transfer per unit time of quantities between cells. The order of accuracy here is related to the degree of interpolation and quadrature polynomials used to reconstruct and integrate the fluxes over each cell face. A degree-p polynomial yields a $(p+1)$th order method. Higher-degree interpolation polynomials can be constrained using information from more neighbouring cells. The computational stencil, defined by the pattern of cells supporting the interpolation, may be arbitrary. Conservation holds as long as the loss of a quantity from one cell sees the gain in another: the fluxes are 'telescoping'.

Despite the flexibility and popularity of the finite volume method, *high-order* finite volume schemes have not gained a wide following in the research community (Ollivier-Gooch *et al.*, 2009). Today's commercial solvers certainly do not implement more than second order accuracy. The challenges surrounding high-order finite volume schemes are explored in greater detail in §2.3, but we can expect large stencils to be a major problem. Not only do the algorithms for constructing large stencils over arbitrary cell arrangements need to be robust, but in parallel processing the increased overlap between domain partitions inevitably compromises speedup. This is where the third approach to modelling partial differential equations – the finite element method (FEM) – becomes relevant, because large computational stencils are conveniently circumvented.

Finite Element Methods

The goal of a finite element method is to find a linear combination of local basis functions that best approximates the solution to the governing equations. The equations are represented in weak form in order to admit a piecewise continuous solution. Each basis function belongs to a node which is resident on some element (or between elements), and the function is local in the sense that it extends only to immediately neighbouring nodes. A linear system of equations corresponding to the basis functions is formed. In the Galerkin approach, each row describes the residual of the governing equation when the corresponding basis function is 'switched on' in the absence of the others. Because of the function's locality, the linear system is sparse and can be solved efficiently. Importantly, arbitrary $(p+1)$th order accuracy can be achieved simply by adding more nodes to the elements – extended neighbours are not needed.

The continuous Galerkin method allows nodal values, also called the degrees of freedom, to be shared by elements along the element boundaries. This results in all of the elements being connected in the linear system: the method is implicit, requiring an inversion of globally defined matrices. In the case of treating advection, the matrices must be assembled and inverted once per time step – possibly more if nonlinear iterations are needed – since the computational stencil must be upwind-biased for stability. The repeated assembly and inversion can become expensive compared to an explicit method.

An increasingly popular paradigm in CFD is the discontinuous Galerkin (DG) method, which is a hybrid of the continuous Galerkin and finite volume method. As before, the domain is divided into elements populated by nodes; but this time the elements do not share nodal values along the element boundaries. Segregating the elements in this way leads to block matrices that are inverted on a local rather than global basis. Communication between the elements is restored by a Riemann solver, which is used to transmit fluxes through element boundaries. Riemann solvers are commonly used in the FVM to admit a piecewise-discontinuous solution. It is now much easier to control advection, because the flux calculation straightforwardly uses information from the upwind element(s); furthermore, high-order fluxes can be calculated using only the immediately neighbouring elements. The discontinuous Galerkin method therefore reproduces the best qualities from finite element and finite volume methods. A disadvantage is that additional degrees of freedom per element have to be stored.

Multi-Moment Methods

The reason for introducing the DG method, aside from its increasing popularity, is that it is closely related to the family of methods to which CIP belongs. The latter methods shall be referred to as the multi-moment methods. They evolved out of CIP, and they all have compact stencils. The word ‘moment’ usually refers to a moment of inertia, such as the zeroth moment: integration of a quantity over a line, polygon or polyhedron (or simply the value at a point). The definition is sometimes extended to include any derivative at a point (Imai and Aoki, 2006b). We will adopt this extended definition for the present thesis. The moments play a similar role to the degrees of freedom in the FEM: they help define a high-order solution, and they do so without requiring information from extended neighbours. They differ from the FEM in that they do not have to be solved as part of a linear system. In fact, the strategies for updating the different types of moment are not defined: they vary according to the individual methods.

CIP, for example, uses sets of point values and gradients which are advected using a semi-Lagrangian formulation (a depiction of this is given in Figure 2.1 on p. 20). This formulation is in contrast to the traditional Eulerian approach, which relies upon the local time derivative in the governing equations to describe the evolution of the flow field on a fixed grid. In a Lagrangian approach, the substantial derivative is invoked so that grid elements move according to the velocity field. The latter has favourable properties but can suffer from grid entanglement.

The semi-Lagrangian approach represents a compromise. At each time step, grid elements are displaced, but they are reset to their original fixed positions through interpolation. Semi-Lagrangian schemes have proved to be accurate and stable for longer time steps than their Eulerian counterparts (Robert, 1981) and have since become popular in meteorological models.

It is clear that it is the configuration of moments, rather than the advection formulation, that will determine compatibility with a non-conforming grid structure. This is why it is more important to examine multi-moment methods in the context of compact methods than to compare CIP against, say, upwind FDMs. Of the compact methods, DG is the most prevalent in the literature and may be considered the gold standard against which to measure multi-moment methods. Other competing methods include the spectral volume and spectral difference methods.

Spectral Volume and Spectral Difference Methods

Developed by Wang (2002) and Wang and Liu (2002), the spectral volume (SV) method divides the domain into cells, or spectral volumes, which are further partitioned into control volumes. The control volumes support the cell in the same way that degrees of freedom support a finite element in the DG method: they are used to constrain a local polynomial profile. The solution is piecewise-discontinuous between cells. Therefore, like the DG method, a Riemann solver is used to transmit fluxes between cells. Conversely, the solution *inside* the cell is piecewise-continuous. This means internal fluxes may be calculated analytically. The internal fluxes are then used to update individual control volumes.

The cells are simplicial in shape and use the same local coordinate systems, which means that the pattern of partitioned control volumes can be made the same across cells. This in turn allows a single set of profile coefficients and quadrature data to be stored, making the method CPU- and memory-efficient. On the other hand, the choice of pattern profoundly affects convergence properties, and optimising the pattern in 3D for higher orders of accuracy becomes increasingly more difficult (Liu *et al.*, 2006).

The spectral difference (SD) method developed by Liu *et al.* (2006) and Wang *et al.* (2007) overcomes the complications inherent in the SV method by supporting the cells with nodes rather than control volumes. Again, a local reconstruction of the solution is performed inside each cell, and a Riemann solver is used to transmit fluxes between cells. But the updating of the nodal values comes from a finite difference formulation instead of a finite volume formulation. It is done by additionally reconstructing a profile of the fluxes across the cell: the surface integrals are replaced by a volume integral. Because of the differential form, a degree-$(p+1)$ rather than a degree-p polynomial is required for the flux profile. As a result, it becomes efficient to store both variable values and flux values at different sets of quadrature points. As expected, the SD method has similar accuracies and more robust convergence properties compared with SV.

Hermite WENO Schemes

One unfortunate aspect of high-order schemes is their tendency to generate spurious oscillations at jump discontinuities such as shocks, in a manner reminiscent of the Gibbs phenomenon in Fourier series. They are a result of Godunov's (1959) theorem, which states that all linear monotone schemes are necessarily first order accurate. 'Monotone' means having the property of not generating new extrema. We can restore the monotonicity of high-order schemes using nonlinear devices such as slope limiters and adaptive stencils, or by adding artificial viscosity to provide damping. Such devices are beyond the scope of the current review, but the idea of an adaptive stencil will be pursued in order to introduce the HWENO schemes.

In the previous discussion on the finite volume method, it was mentioned that a high-order scheme requires a large, but arbitrary, computational stencil to reconstruct the fluxes. This flexibility to choose the stencil is exploited in the essentially non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) schemes pioneered by Harten *et al.* (1987), Shu and Osher (1988), Liu *et al.* (1994) and Jiang and Shu (1996). In an ENO scheme, the stencil is chosen from a number of candidates according to which gives the smoothest solution. Since the candidates crossing a discontinuity will result in non-smooth solutions, they ought to be automatically excluded. A WENO scheme does not choose one candidate, but instead assigns a nonlinear weight to each so that the fluxes may be represented as a convex combination of all the polynomials. The non-smooth candidates will have near-zero contributions. The WENO approach has been shown to improve upon the stability, smoothness and accuracy of the ENO schemes.

More recently, Hermite WENO (HWENO) schemes have been developed by Qiu and Shu (2003, 2005). Like multi-moment methods such as CIP, the HWENO approach stores and evolves the gradients in addition to the unknown values. As a result, it is compact. Unlike the multi-moment methods, it is free to use a combination of moments on both upwind and downwind sides in order to achieve accuracy when the solution is sufficiently smooth.

Despite the similarity to the multi-moment methods, the HWENO schemes are not competitive as standalone spatial discretisations due to the cost of reconstructing and evaluating the candidates at each time step. Qiu and Shu (2003) point out that their HWENO5-RK3 scheme (fifth order in space, third order in time) is almost twice as costly as WENO5-RK3 for the same mesh while only marginally improving the accuracy. The HWENO schemes are instead more applicable as limiters for the DG method. The philosophy is as follows: conventional slope limiters, while being an inexpensive way to enforce smoothness, necessarily degrade the solution to first order accuracy in the region of the discontinuity (and sometimes elsewhere). By contrast, the ENO/WENO/HWENO schemes converge to smooth solutions with the correct order of accuracy, but they are expensive. So Qiu and Shu use total variation bounding (TVB) limiters to mark 'troubled cells' and subsequently treat those cells with HWENO, thus ensuring a smooth, accurate and relatively economical solution. One might speculate that HWENO schemes could also be applied to the other compact methods to achieve the same effect.

2.1.2 High-Order Methods and Gridding

From the perspective of implementing a high-order method on a body non-conforming grid, it is tempting to restrict the literature review to that type of grid. But actually it is necessary to consider unstructured grids as well. Not only is the unstructured grid literature more mature; researchers have spent more effort rigorously implementing high-order unstructured methods. It will be shown in §2.4 that researchers of non-conforming methods often do not give much regard to maintaining high-order accuracy at the boundary.

In going from any rectilinear grid implementation to a non-rectilinear one, maintaining high-order accuracy in space becomes difficult for two reasons. The first reason is that the approximation of the domain boundary itself carries an order of accuracy. In the most primitive approximation, blanking out boundary cells or elements to resemble a staircase-like profile leads to first order accuracy. For second order accuracy, the boundary should be represented by piecewise-linear segments. For higher orders, the requirement is stricter still; curvature must be represented. In the case of compact methods, the higher-order approximation can be achieved by adding more points to the grid elements, thus lending curvature to the elements. Non-compact methods, being confined to low-order elements, must approximate the curved surface by shaping the large computational stencils to include a sufficient footprint of boundary elements. Construction of such stencils can bring complications. On the other hand, most non-Cartesian grid generating software produces unstructured linear elements. Being able to represent curvature through the stencil rather than through curved elements has the distinct advantage that one is not locked into some non-standard grid format.

The second reason is that integration of quantities over cell volumes, and fluxes over cell faces, becomes harder for accuracies greater than second order. In fully Cartesian grids, exact integration over the Cartesian elements is possible when the profiles of quantities are known analytically. But in grids with arbitrarily shaped, arbitrarily orientated elements, numerical quadrature must be used. Numerical quadrature involves interpolating the quantity at a prescribed set of points, assigning prescribed weights to the interpolations, and summing. The resulting value will correspond to the exact integration of a polynomial constrained according to the quadrature points. The degree of this polynomial corresponds to the quadrature's order of accuracy.

Second order quadrature is simply interpolation at the centroid of the element. In other words, quantities stored at element centroids may be regarded as element averages integrated to second order accuracy. This type of quadrature is so common that it is known as the midpoint rule. However, the skewness of elements complicates interpolations. For example, in the FVM, integration of face fluxes would ideally involve interpolation between the two connecting cell centroids. But if the cells are skew, the line of interpolation will not intersect the face centroid. In this case linear interpolation can still be guaranteed by including immediate cell neighbours in the reckoning. The stencil does not need to be extended beyond these immediate neighbours.

> *"On structured non-orthogonal grids one can use higher-order integration and interpolation techniques to approximate convective fluxes . . . However, if the grid is unstructured and involves CVs of arbitrary numbers of faces, use of linear interpolation and the midpoint rule approximation seem to offer the best compromise among accuracy, generality, and simplicity. Indeed, a computer code which uses these techniques is simple, even for grids of arbitrary shape."* "
>
> – Ferziger and Perić (1996)

High-order quadratures are more commonly used in the finite element method, where basis functions are compact and lead to straightforward interpolations. For arbitrary shapes, a theoretically infinite number of configurations of quadrature points is possible. Optimisation of these is a small but active area of research (Mousavi *et al.*, 2010). For a simplex (i.e. a triangle or tetrahedron), which is the most primitive class of polytope, optimum configurations are well established (Hammer and Stroud, 1956). In the worst case, an arbitrary polytope can be decomposed into simplices before being integrated piecewise, which is a standard strategy in the FEM.

In summary, both the geometry representation and the interpolation/integration methods are associated with orders of accuracy in space. The latter affect the scheme's *formal* order of accuracy. It is an obligation of the developer of any numerical scheme to maintain the formal order of accuracy. The developer of a *compact*, *non-Cartesian* numerical scheme is additionally faced with an choice: should high-order representations of the boundary be accommodated? This question will be returned to in §2.3.

2.1.3 Organisation of the Review

The review is organised in three parts. The first part, §2.2, explores the multi-moment methods in more detail and in particular tries to assess their numerical properties. CIP is included, but it is not the focus of the section. It will be seen that much of the multi-moment literature concentrates on 1D schemes and multidimensional schemes in rectilinear Cartesian grids. The latter are relevant to a Cartesian non-conforming grid implementation as they can be used in the background grid where the cells or points are unmodified.

It must then be considered what kind of difficulties will be encountered in the extension to non-conforming grids. The literature does not offer easy answers, because it appears that existing non-conforming methods often do not succeed in achieving more than second order accuracy at the boundary. The issue can be better understood by looking at attempts to implement high-order methods on unstructured triangular/tetrahedral grids: these represent the prototypical irregular grid. The implementations and surrounding issues are studied in §2.3. The scope is broad, with high-order FVM, DG and multi-moment methods included.

The above study of unstructured grid implementations puts us in a better position to finally make a critical review of Cartesian non-conforming methods in §2.4. Traditionally these have included the immersed boundary method and Cartesian cut cells. The majority of non-conforming methods are second order, but potentially high-order implementations of DG and multi-moment methods will be examined. The review concludes in §2.5 by establishing the objectives of the present research.

2.2 Multi-Moment Methods

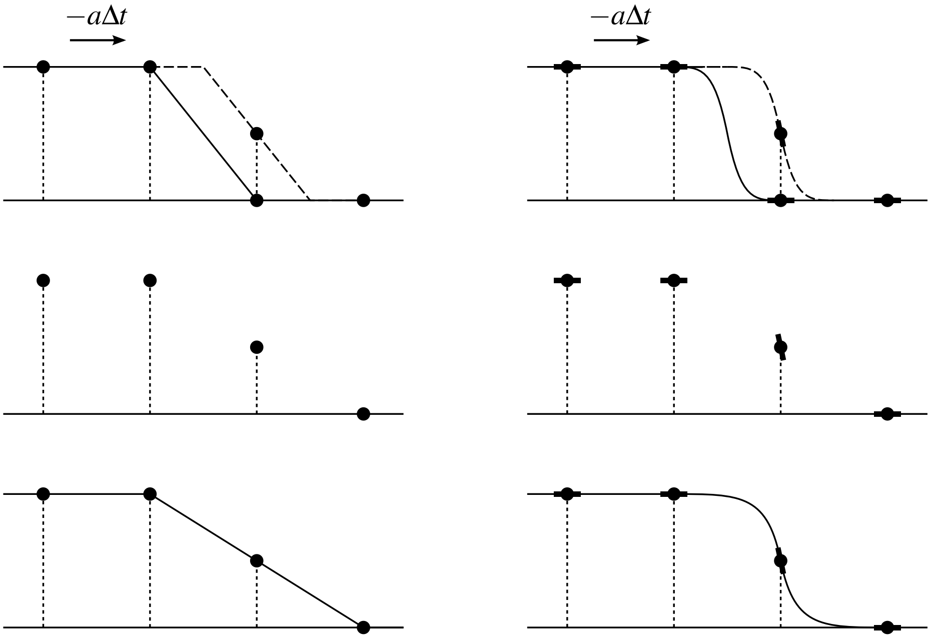
The defining feature of multi-moment methods is that they store more information than conventional FVM/FDM in order to raise the order of accuracy. The methods began with CIP, a semi-Lagrangian advection scheme. The original CIP stores gradients in addition to undifferentiated values at grid points. This allows a cubic interpolation polynomial to be fitted between the points. At each time step, departure points are found by backtracking along the trajectories of the arrival points (coincident with grid points) according to the velocity field. The new point values and gradients are interpolated at the departure points using the appropriate polynomial. A visualisation of this process is shown in Figure 2.1. CIP was first introduced by Takewaki *et al.* (1985). The abbreviation CIP originally stood for 'cubic-interpolated pseudo-particle' or 'cubic interpolation polynomial', but this was later changed to 'constrained interpolation profile' for generality.

CIP is implemented on a Cartesian grid of points. It is not conservative. Conservative terms such as the inviscid momentum term in the Euler/Navier–Stokes equations must be split into advective and divergent parts according to the decomposition

$$\nabla \cdot (\phi \mathbf{V}) = \mathbf{V} \cdot \nabla \phi + \phi \nabla \cdot \mathbf{V} \tag{2.1}$$

where ϕ is the advected quantity and $\mathbf{V}$ is the velocity vector. The two parts are resolved sequentially in fractional steps, with the linear advective term treated by CIP and the divergence term treated by a sum of (usually second order central) finite differences (Takewaki and Yabe, 1987; Yabe and Aoki, 1991). It may be concluded that the original CIP scheme is inherently unsuited to implementations on irregular grids, unless it can be adapted in the manner of immersed/embedded boundary methods (§2.4).

Yabe and Aoki (1991), Xiao *et al.* (1996a) and Xiao and Yabe (2001) offer CIP variants that help suppress oscillations at discontinuities. But the pivotal development came from Tanaka *et al.* (2000), who proposed the introduction of a cell-integrated average in order to restore conservation of mass. This is easiest to explain in one dimension, where a cell is defined as a line joining two points. As before, each point value and (possibly) point gradient is updated by evaluating an interpolation polynomial at the corresponding departure point. This time,



(a) Linear Lagrange **(b)** CIP

Figure 2.1: Reproduction of Figure 1 from Yabe *et al.* (2001b) showing the principle of the CIP method. The solid line is the initial profile and the dashed line is an exact solution after advection. a is the advection velocity, Δt is the time step. In the linear Lagrange scheme, points values are linearly interpolated and numerical diffusion appears. In CIP, the spatial derivative also propagates and the profile inside a grid cell is retrieved to a higher order of accuracy.

however, the polynomial is further constrained by an integration over the cell. To update the cell-integrated average, the volumes swept through the cell boundaries during the time step must be computed. Each swept volume is equal to an integration of the appropriate polynomial in space between arrival and departure points, by virtue of the characteristics. This is still a semi-Lagrangian procedure, but conservation is upheld because the loss of mass in one cell sees the gain in another. The new method is called CIP–CSL, where CSL is an abbreviation of ‘conservative semi-Lagrangian’. While reminiscent of the finite volume method, CIP–CSL remains compact and appears to be free from the stability limitations of the Eulerian form.

A closely related method to CIP is the Interpolated Differential Operator (IDO) method (Aoki, 1997). Like CIP, IDO stores both point values and gradients, using them to constrain a polynomial between grid points. But whereas CIP performs a semi-Lagrangian advection step, IDO differentiates the polynomial directly at the point of interest to arrive at an Eulerian formulation. This approach can be extended to diffusion terms in the governing equation by making the polynomial span *both* sides of the point of interest. In the simplest IDO scheme, the number of moments available for constraining the polynomial is four for advection and six for diffusion, leading to degree-three and degree-five polynomials respectively. Researchers of CIP have often contributed to the development of IDO, and vice versa. But the development of IDO has lagged somewhat; for example, only recently has the cell-based modification been adopted (Imai *et al.*, 2008).

The term ‘multi-moment’ first appeared in articles by Xiao (2004) and Ii *et al.* (2005). Here CIP is incorporated into multidimensional frameworks called, respectively, the volume/surface integrated average multi-moment method (VSIAM) and the CIP/multi-moment finite volume method (CIP/MM–FVM). Both frameworks conserve the cell-integrated average. VSIAM additionally uses face-integrated averages, interleaving CIP–CSL operators so that all moments are advanced by the semi-Lagrangian formulation. It therefore has the desirable property of being both conservative and stable for large time steps. And, because face-normal velocities are stored at the faces, the pressure–velocity coupling problem of incompressible flows is easily solved (Xiao *et al.*, 2006). On the other hand, the reliance on CIP–CSL operators restricts the framework to Cartesian grids. Modifications for handling arbitrary boundaries can be found in Xiao *et al.* (2005), although we defer discussion of that article to §2.4.1.

The more flexible CIP/MM–FVM uses point moments which are populated around the boundaries of the cell. These are advanced by CIP. Multidimensional profile reconstruction replaces operator splitting, which means the method lends itself to non-Cartesian grids. This is an extremely useful application of CIP/MM–FVM, but again we defer discussion to a later section (§2.3.2). The cell-integrated average is advanced by a finite volume formulation, with the fluxes over the cell boundaries reconstructed to a high order using the local point moments. The method is therefore a hybrid between semi-Lagrangian and Eulerian formulations, in contrast to the completely semi-Lagrangian CIP–CSL. We can expect the price of this flexibility to be a restricted time step.

The most recent major development is the multi-moment constrained finite volume method (MCV) by Ii and Xiao (2009). Like CIP/MM–FVM, point moments are used and a finite volume formulation is exploited in order to achieve mass conservation. However, the cell integrated average is no longer stored. Instead, the n point moments on each cell are updated as part of a linear system containing the FVM equation and $(n-1)$ arbitrary evolution equations. There need not be a direct correspondence between each point moment and its semi-Lagrangian advection. One can instead use the set of point moments on each cell to constrain an interpolation profile, then evolve the profile according to a different set of moments and associated equations. This is not a new idea; Crowley (1968) showed how high-order noncompact schemes can be made conservative by a similar construction. The advantage of MCV therefore seems to be greater flexibility to mix and match various existing numerical schemes. Ii and Xiao drop the semi-Lagrangian formulation entirely, adopting interpolated differential operators to evolve the point values. Another, arguably more important, feature of MCV is that point values are duplicated at cell boundaries, allowing profiles to be discontinuous between cells. A Riemann solver is used to resolve the fluxes at the cell boundaries. Presumably this makes MCV better at shock capturing in compressible flows compared to the other multi-moment methods.

The aim of the following subsections is to quantify the performance of CIP and other multi-moment schemes, both between themselves and in relation to competing methods such as DG, as far as possible. This is not a straightforward task; it will be seen that there are hidden factors complicating analyses, and published results are few. It raises questions about how complete the existing analyses of multi-moment methods are.

The reader unfamiliar with CIP and other multi-moment schemes is referred to §3.3.1, where one-dimensional formulations and introductory analyses are provided.

2.2.1 One Spatial Dimension

In this section we examine the properties and performance of one-dimensional multi-moment methods. There are two classic approaches to analysing the numerical properties: the von Neumann stability analysis and the matrix method (Hirsch, 2007). The von Neumann analysis is restricted to interior schemes as it assumes a uniform grid with periodic boundary conditions. But it is simple to perform, and it produces a continuous spectrum of amplification factors with respect to wavenumber. On the other hand, the matrix method is evidently needed to break down schemes with multiple types of moment (Utsumi *et al.*, 1997; Imai *et al.*, 2008) or degrees of freedom (Hu *et al.*, 1999). In any case the governing differential equation must be linear or linearised. The two methods are introduced here for the purpose of discussion – more details are given in §3.2.1.

Traditionally the von Neumann analysis applies to fully discretised equations. Actually, it is possible to semi-discretise the equations and consider the space discretisation in isolation. The governing equation is written with the continuous time differential on the left hand side and discrete space operators on the right hand side. When there are several types of moment in the domain, under periodic boundary conditions, the right hand side appears as a square matrix multiplying a column vector of moment values. This matrix encapsulates the properties of the spatial scheme.

The matrix is decomposed into a set of eigenvalues and eigenvectors corresponding to Fourier modes. The imaginary parts of the eigenvalues tell us how well the modes match the wavespeed of the exact solution, indicating the dispersion accuracy, while the real parts tell us by how much they are damped, indicating the dissipation accuracy. When there is more than one moment type, further eigenvalues will be present at a given wavenumber. These ought to be more heavily damped than the eigenvalues representing the physical solution. All of the eigenvalues must have negative real parts to ensure stability.

For Eulerian methods treating the linear wave advection problem, the eigenvalues scale with the Courant number, $\sigma := a\Delta x/\Delta t$, where a is the exact wavespeed, Δx the grid spacing, and Δt the time step. This is extremely convenient, because it means the spectrum can be scaled to fit the region of stability of a compatible time scheme, leading to a precise stability criterion. It also means that the discretisation error of the space scheme is independent of the time scheme. In short, the space and time discretisations are decoupled from one another.

When the time discretisation is introduced into the analysis, the eigenvalues are transformed into amplification factors. A fully discrete scheme is now represented. The gain and phase errors correspond to dissipation and dispersion, respectively, and a gain less than or equal to unity is needed everywhere for stability. These fully discrete properties are also obtainable from the von Neumann analysis.

Given that the space discretisation is the defining feature of any multi-moment method, one might expect researchers of the multi-moment methods to favour a semi-discrete stability analysis over a fully discrete one. But in the few works dedicated to analysing the properties of multi-moment methods (Utsumi *et al.*, 1997; Imai and Aoki, 2006b; Konno *et al.*, 2008), a fully discrete approach is always taken. There is a good reason for this in the case of Utsumi *et al.*, who analyse the original CIP method: in a semi-Lagrangian formulation, it is not possible to decouple the space and time discretisations. Spatial interpolation at the departure point depends on a time integration along the trajectory to get the point coordinates.

Another way of looking at this dependence is to examine the error term. Xiu and Karniadakis (2001), citing Falcone and Ferretti (1998), point out and go on to demonstrate that the overall error of a semi-Lagrangian scheme has the form

$$\varepsilon_T = \mathcal{O}\left(\Delta t^q, \frac{\Delta x^{p+1}}{\Delta t}\right) \tag{2.2}$$

where q is the order of time integration (changed from k to avoid confusion with wavenumber) and p the degree of interpolation polynomial. The scheme in question is presumably in differential form. The error contrasts with that of a purely Eulerian scheme in differential form, where the equivalent polynomial is differentiated:

$$\varepsilon_T = \mathcal{O}\left(\Delta t^q, \Delta x^p\right) \tag{2.3}$$

and for a purely Eulerian scheme in integral form, where the polynomial is used to interpolate face fluxes:

$$\varepsilon_T = \mathcal{O}\left(\Delta t^q, \Delta x^{p+1}\right) \tag{2.4}$$

The conflated spatial–temporal term in (2.2) explains why Utsumi *et al.* and other analysts of semi-Lagrangian schemes (e.g. Lauritzen, 2007) take a fully discrete approach. Utsumi *et al.* and Imai and Aoki do incorporate matrices with modal decompositions on account of the variety of moment types, but the matrices correspond to the fully discrete equations, not semi-discrete ones.

Comparison with Non-Compact Methods

The gain and phase errors of CIP/IDO are favourable compared to non-compact alternatives. One particularly striking plot from Utsumi *et al.*, which reappears in Yabe *et al.* (2001b), is reproduced in Figure 2.2. The wavenumber has been nondimensionalised as $K := k\Delta x$. CIP can be seen following the exact wave speed closely for the wavenumber range $0 \leq K \leq \pi$. By contrast, other schemes follow the exact speed until $K \approx \pi/2$, at best, before dropping to zero at $K = \pi$. We shall henceforth refer to the wavenumber at which the numerical wavespeed returns to zero as K_0. The gain curves of CIP are also superior, but the comparison is somewhat unfair since the other schemes are only of first and second order.

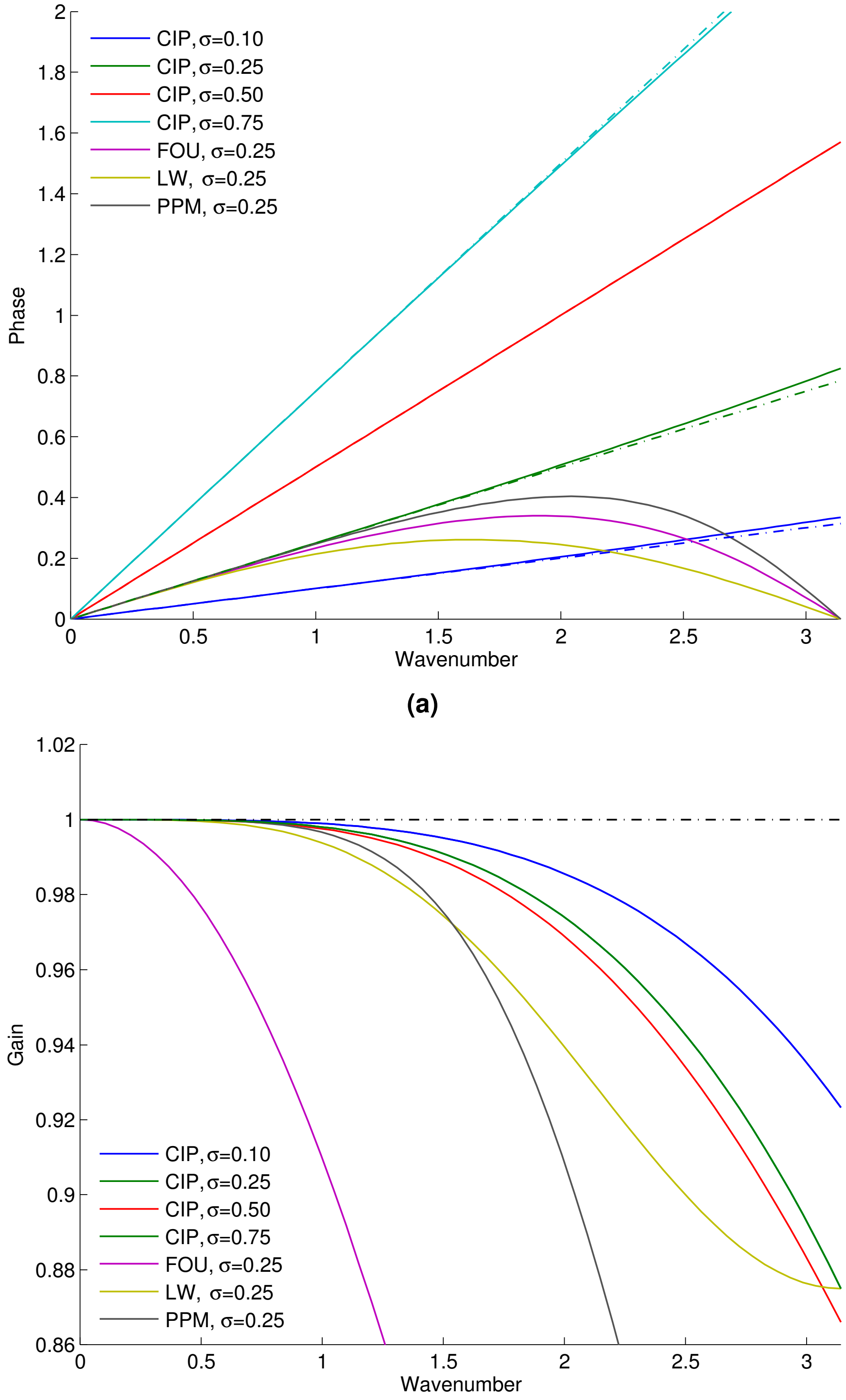
Imai and Aoki (2006b) present properties of the IDO scheme for advection, diffusion and the pressure equation. We are most interested in advection, although the phase curves for diffusion are also notable: they show that even compact difference schemes are limited to a resolvable wavenumber range defined by $K_0 = \pi$, whereas IDO, like CIP, approximates the exact solution well beyond $K = \pi$.

IDO is an Eulerian method, and so a semi-discrete approach could have been taken in the article. But the fully discrete analysis does allow a useful comparison with semi-Lagrangian schemes. Most usefully, Imai and Aoki compare schemes of the same spatial order at low Courant numbers, which means the dominance of the space discretisation can be observed. The relevant phase and gain curves are reproduced in Figure 2.3. The schemes include CIP, IDO, and the cubic Lagrange (CUL) and third order upwind (TOU) schemes. CUL is evidently a semi-Lagrangian reformulation of the Eulerian TOU scheme.

The curves are normalised and plotted at Courant numbers of 0.1, 0.2 and 0.4. Looking at the plots in reverse order, the spectra can be seen to converge upon their semi-discrete counterparts as the influence of the time integration diminishes. The Eulerian schemes in fact resemble their semi-discrete representations to begin with, owing to their coupling with the highly accurate, fourth order Runge–Kutta (RK4) time scheme. The CIP and IDO curves merge, confirming the two schemes are identical from the point of view of the space discretisation. The CUL and TOU curves behave in exactly the same way.

Comparison with Compact Methods

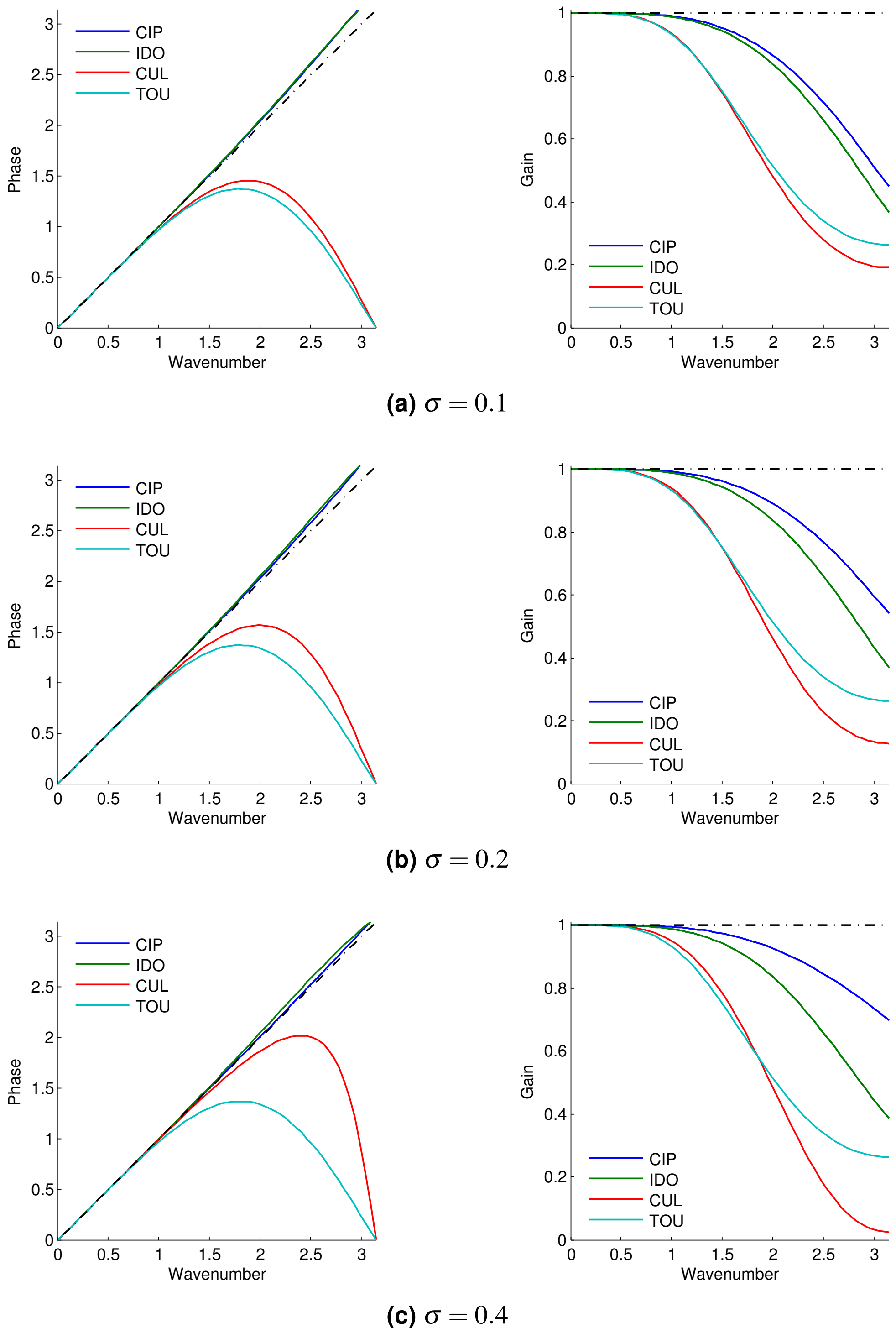
It is clear that multi-moment methods are superior to non-compact alternatives in terms of dispersion and dissipation accuracy, but it would be more useful to extend the comparison to other compact methods. Hu *et al.* (1999) provide a good benchmark analysis for the DG method at various orders of spatial accuracy. Their analysis is semi-discrete. Immediately it can be seen that the multi-moment methods are not alone in approximating the wave speed well beyond $K = \pi$; the DG method also has this quality. In fact the resolvable wavenumber range is defined exactly by $K_0 = \pi(p+1)$, where p is the degree of polynomial reconstruction within



(a)

(b)

Figure 2.2: (a) Phase and (b) gain curves of CIP at the first time step at Courant numbers 0.1, 0.25, 0.5 and 0.75. The curves for the first order upwind, Lax–Wendroff and Piecewise Parabolic Method at 0.25 are also shown. Exact solutions are indicated by dash-dotted lines. Reproduced from Utsumi *et al.* (1997); Lauritzen (2007).



(a) $\sigma = 0.1$

(b) $\sigma = 0.2$

(c) $\sigma = 0.4$

Figure 2.3: Comparison of CIP, IDO, cubic Lagrange (CUL) and third order upwind (TOU) schemes at increasing Courant numbers. Phases have been normalised as $\overline{\phi} := \phi/\sigma$; gains as $\overline{|G|} := |G|^{1/\sigma}$. The Eulerian IDO and TOU schemes have been coupled with a fourth order Runge-Kutta time integration. Reproduced from Imai and Aoki (2006b).

the element, leading to $(p+1)$th order spatial accuracy. It makes the plots of Utsumi *et al.* and Imai and Aoki seem incomplete, since they do not show the full range of wavenumbers resolvable by CIP and IDO.

Hu *et al.* do not compare the DG spectra with other methods, but they do go on to propose a useful measure of performance: the resolution of a scheme is quantified as the largest wavenumber K_c satisfying the criteria

$$\left|\Omega_r(K)/\sigma\right| < 0.005 \quad \text{(dissipation)} \tag{2.5a}$$

$$\text{and} \quad \left|-\Omega_i(K)/\sigma - K\right| < 0.005 \quad \text{(dispersion)} \tag{2.5b}$$

Here the eigenvalue Ω corresponds to the physical mode and may be interpreted as nondimensional frequency, $\Omega := \omega\Delta t$. Note that Hu *et al.* include the division by σ in their definition of Ω. Also, they formulate their analysis in terms of the dispersion relation, so in their paper the real and imaginary parts of the eigenvalues correspond to dispersion and dissipation, respectively. In (2.5) the subscripts are the other way around, and the minus sign is needed on the imaginary part, to suit conventional analysis (Hirsch, 2007).

Zhang and Shu (2005) analyse both the DG and spectral volume methods, providing exact formulae for the eigenvalues of the second order schemes rather than processing them numerically. K_c can be calculated from these formulae (using e.g. the bisection method) in order to compare the resolution of SV against DG. Returning to the work of Imai and Aoki, we are fortunate to have Eulerian schemes with high-order time integrations presented at low Courant numbers, because the resulting spectra can be interpreted as approximately semi-discrete. They can be used to make estimates of K_c. Finally, Xiao (2012), discussing the third order MCV scheme and two variants, includes plots of the eigenvalue spectra on the complex plane. Like IDO, MCV is fully Eulerian and the semi-discrete analysis makes sense. The analysis is not exhaustive and Xiao does not give full details of the spectra, but he does give Taylor series expansions which again can be used to estimate K_c. The bisection method was used except in the case of Imai and Aoki, for which the plot data from Figure 2.3 were simply spline-interpolated.

On the surface, it appears as though the multi-moment methods are not as competitive as the SV and DG methods at the same order of accuracy. But it is interesting that the second order DG scheme has almost exactly the same resolution as the third order IDO scheme. These schemes also have the same storage cost, with two degrees of freedom or types of moment per grid cell. One might speculate whether the two schemes are in fact equivalent, despite citing different orders of accuracy. Unfortunately, the ambiguity makes it difficult to gauge the relative performance of schemes without further analysis. We will return to this point later in the review, but the interested reader is referred forward to Chapter 4 where such an analysis is carried out.

Table 2.1: Resolvable wavenumber K_c, defined by Hu *et al.* (1999), evaluated for various schemes. Imai and Aoki (2006b) and Xiao (2012) provide information from which K_c has been estimated. Zhang and Shu (2005) provide formulae from which K_c has been calculated accurately. Hu *et al.* give the values explicitly.

Method/scheme	Order	K_c	Source
TOU	3	0.500	Imai and Aoki (2006b)
IDO	3	0.787	
MCV3	3	0.774	Xiao (2012)
MCV3_UPCC	3	0.921	
MCV3_CPCC	4	1.19	
SV	2	0.509	Zhang and Shu (2005)
DG	2	0.786	
DG	2	0.8	Hu *et al.* (1999)
	3	1.8	
	4	3.2	

Performance in the Time Domain

With comparisons in the frequency domain failing us, we turn to the time domain. Testing in the time domain is more flexible than spectral analysis in that nonlinear equations can be studied. On the other hand, the time and space discretisations are necessarily coupled together so that only the overall accuracy can be observed. A number of standard 1D problems exist, some of which involve non-smooth solutions for demonstrating behaviour in the presence of discontinuities. However, only tests with smooth solutions are of interest to us. Linear methods such as the multi-moment methods will not converge upon non-smooth solutions in the absence of limiters, due to the presence of oscillations.

For a given grid resolution and time step, the accuracy of the numerical methods under test can be quantified by calculating the errors between the numerical solution and the known continuum solution, and aggregating the errors by a norm. Convergence can be observed by comparing the norms on successively fine grids. The time step size and Courant number are not critical parameters in these tests, because the temporal error tends to be subdominant. This will be demonstrated in §3.3.2. What is important is that the Courant number is small in the case of semi-Lagrangian schemes (so that they approximate their Eulerian counterparts; c.f. Figure 2.3), and that the simulated time period is consistent. Unfortunately some articles use the same standard test cases but take results at different simulated times, so including them in comparisons would be unfair.

Starting with comparisons between multi-moment schemes, it is interesting to note that when an interpolating polynomial of degree p is used, the resulting accuracy is pth order for CIP and

$(p+1)$th order for the CIP-CSL schemes. This is proven by Taylor series expansion in Utsumi *et al.* (1997) and Ii and Xiao (2007), respectively, and is verified by tests on CIP (cubic) and CIP-CSL2 (quadratic) yielding almost exactly the same errors with respect to grid resolution in Yabe *et al.* (2001a). One can see how this fits the error expressions (2.2) and (2.4) in the context of linear advection. If the Δt denominator in (2.2) is converted to Δx via a constant Courant number, and if the spatial error remains dominant, then the semi-Lagrangian error reduces to $\mathcal{O}\left(\Delta x^p\right)$ compared with the Eulerian integral-form error $\mathcal{O}\left(\Delta x^{p+1}\right)$. What is interesting is that the presence of the integral formulation appears to promote the otherwise differential-form accuracy by an order of magnitude, rather than the differential form demoting the otherwise integral-form accuracy. In short, by including the cell-integrated average and an appropriate integral formulation, it appears that not only do we earn conservation; we also earn an extra order of magnitude in spatial accuracy. In this instance, however, the two schemes still store the same number of moments per grid cell, so there is no change in terms of the storage cost.

Comparison with other compact methods is more difficult and requires collating results from individual articles. Three common test cases are identified as:

- Linear advection $u_t + u_x = 0$ with initial condition $u(x,0) = \sin(\pi x)$ and final time $t = 1$ (i.e. advection of one wavelength)
- the Burgers equation $u_t + \left(u^2/2\right)_x = 0$ with initial condition $u(x,0) = 0.5 + \sin(\pi x)$ and final time $t = 0.5/\pi$;
- the Euler equations (2.6), with equation of state $p = (1-\gamma)\left(e - \rho u^2/2\right)$, $\gamma = 1.4$, initial conditions $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $u(x,0) = p(x,0) = 1$, and final time $t = 2$.

$$\mathbf{u}_t + \mathbf{f}_x = 0, \qquad \mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix} \tag{2.6}$$

Figure 2.4 collects the various results of Ii and Xiao (2007, 2009), Qiu and Shu (2003), and Wang (2002) in a tableau of plots. Results were also available from Cockburn and Shu (1989), Cockburn (1999), and Zhang and Shu (2005), but these had to be excluded on account of different periods of time simulated.

Whether meaningful comparisons can be drawn from the plots is doubtful. At first glance, the multi-moment schemes compare favourably with the other methods. But while they perform almost identically with DG for the Burgers equation, DG method surpasses them by several orders of magnitude for the Euler equations. Is this due to the nature of the equations, or are there hidden effects that have not been exposed here? The data are too sparse to draw any correlations.

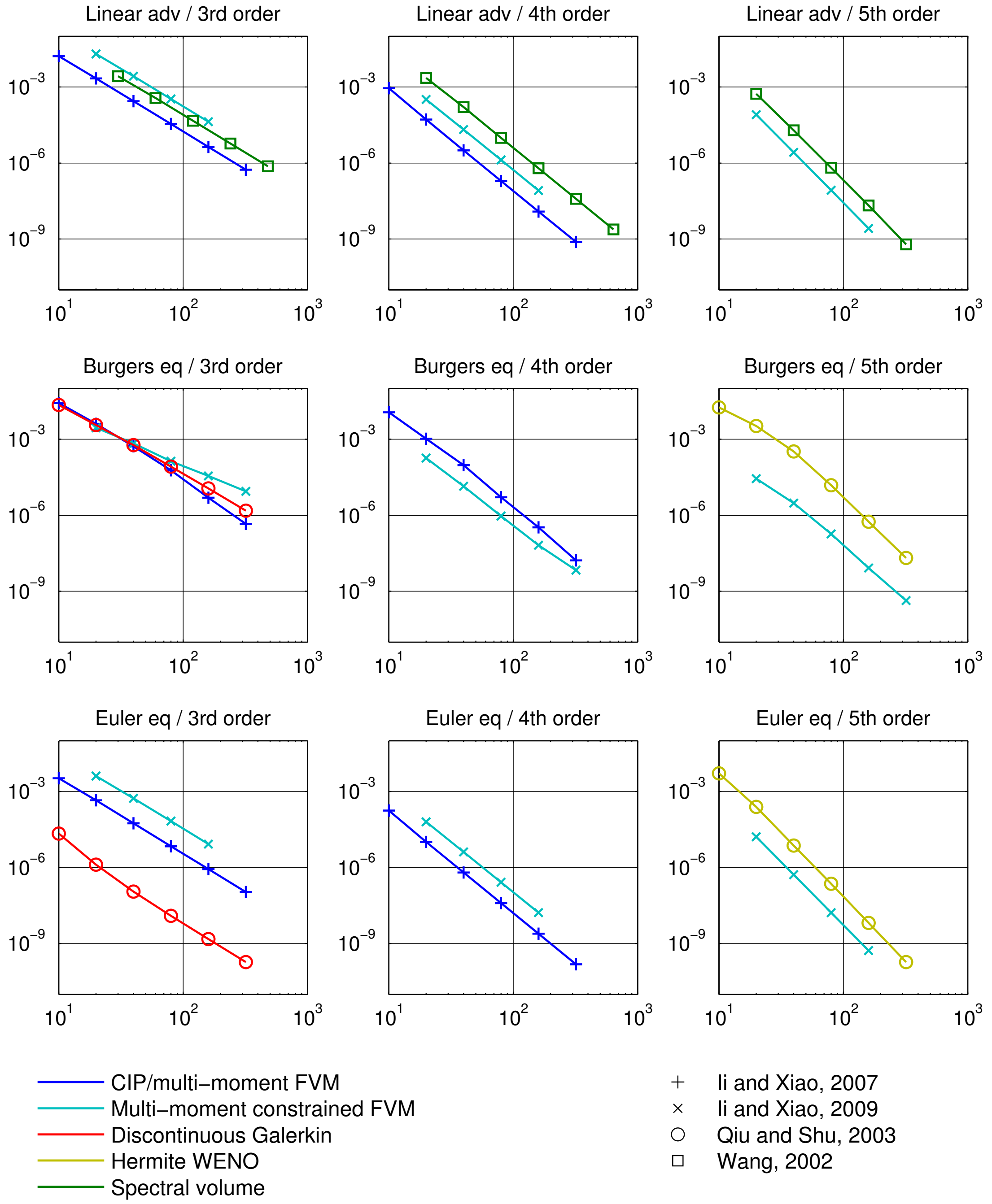


Figure 2.4: L_∞ convergence of various compact schemes with respect to grid size, modelling: (row 1) linear advection; (row 2) the Burgers equation; (row 3) Euler equations. Schemes are of third to fifth order accuracy (columns 1 to 3, respectively). Reproduced from Li and Xiao (2007, 2009); Wang (2002); Qiu and Shu (2003).

Two more articles are worth mentioning for their direct comparison of methods, albeit with different tests or test parameters to those above. Zhang and Shu (2005) compare SV and DG in linear advection. The L^∞-errors of SV are 3.9–4.1 times those of DG in third order and 3.1 times in fourth order. If we digress to multidimensional schemes, we find a fourth order, triangular cell-based CIP/MM–FVM scheme (Ii *et al.*, 2005) that can be compared to SV and DG counterparts (Sun and Wang, 2004) in diagonal linear advection. The L^∞-errors relative to DG on a regular grid are 13.3 for CIP/MM–FVM and 3.5 for SV. One may conclude that the studies of Zhang and Shu and Sun and Wang comparing SV and DG are consistent. But the relative performance of CIP/MM–FVM does not fit into the plots of Figure 2.4. Clearly there is a need for a more systematic study of performance between compact methods.

2.2.2 Extension to Multidimensions

So far, review of multi-moment methods has been limited to one-dimensional implementations in order to explain general concepts and properties. The bulk of the multi-moment literature falls under this category. We now consider extensions to multidimensional grids, but still limit ourselves to rectilinear, Cartesian grid elements. There are a number of ways of extending a given method to multidimensions, each one having an impact on the method's numerical properties and computational costs. But the jump in complexity in going to from one dimensional to multidimensional grid elements is not as severe as going from Cartesian to non-Cartesian elements (and hence to grids with arbitrary boundaries). Because of this, and because there are practical applications in multidimensions that do not require the presence of arbitrary boundaries, there is a wealth of articles on Cartesian multidimensional schemes.

Reviewing these schemes should allow us to make an informed decision when it comes to implementing the unmodified part – the background – of a Cartesian non-conforming grid. Here it is more important to pay attention to numerical details, such as performance or arrangement of moments, than the proclaimed end applications.

Classifications

Schemes can be subdivided into those using directional splitting (also known as operator splitting) and unsplit (or 'direct') schemes. With splitting, one or more 1D operators are applied to each dimension in sequence. In the case of multi-moment methods, this means a sequence of 1D profile reconstructions and interpolations. The situation is somewhat complicated by the fact that a multi-moment operator needs a compatible set of moments in each direction to work. In the x-direction, for example, CIP can advect point values when x-gradients are present, and vice-versa. It *cannot* advect y-gradients in that direction unless xy-gradients (cross-derivatives) are present, and vice-versa.

It means that if we want to use high-order operators everywhere, we must supply extra moments that are redundant in the sense that they do not serve to increase the overall order of accuracy compared to the equivalent 1D scheme. Adding the xy-gradients was the approach taken by Aoki (1995) to yield a CIP scheme that was later called ‘Type-C’ (Yabe *et al.*, 2004). The alternative is to mix in lower-order operators to update problem moments at each step; for example, y-gradients can be updated in the x-direction by taking finite differences with neighbouring y-gradients. This approach yielded the earliest multidimensional CIP scheme (Takewaki and Yabe, 1987), later called ‘Type-M’. In this scheme a first order upwinding (FOU) is used to update the transverse gradients.

Two more examples may be found in the CIP-CSL schemes. These schemes often use a set of point, line, face and/or cell values (where the value is integrated and averaged), rather than a set of point derivatives, to construct high-order profiles. In 2D, considering the x-direction again, CIP-CSL2 constructs quadratic profiles from either (i) two point values and one horizontal line value, or (ii) two vertical line values and the face value. If the complete set is present, we have the scheme of Nakamura *et al.* (2001). If the point values are omitted, we must update the horizontal line values by some other means. Xiao *et al.* (2006) use a so-called ‘time evolution converting’ (TEC) formula, a second-order interpolation of the time evolution of the two neighbouring face values.

To generalise, we shall refer to the first type of split scheme as having uniform operators, and to the second type as having mixed operators. We can expect uniform operators to give greater accuracy, but at the expense of additional memory required to store the extra moments. The question that must be asked is how this trade-off can be quantified. Split schemes may additionally suffer from a temporal error if the sequence of 1D operators does not properly cancel the cross-derivatives in the Taylor series expansion.

Unsplit methods involve a multidimensional profile reconstruction over each grid element followed by direct interpolation or differentiation. They may be more expensive at higher orders of spatial accuracy, since the number of profile coefficients increases as $\mathcal{O}\left(p^N\right)$, where p is the degree of polynomial and N the number of dimensions. To be precise: the minimum number of coefficients, and hence moments, needed to complete the order of accuracy is $\frac{1}{2}p(p+1)$ in two dimensions (a triangular number) and $\frac{1}{6}p(p+1)(p+2)$ in three dimensions (a tetrahedral number). It will be seen later that these numbers lend themselves to triangular and tetrahedral grids, because the moments can be arranged naturally for these shapes. For Cartesian grids, it is more natural, albeit less economical, to use p^N moments (square or cubic numbers). We shall refer to these two types of profile as the *order-complete* basis and the *tensor product* basis, after Hu *et al.* (1999).

Takizawa *et al.* (2002) implement CIP-CSL2 in an unsplit, tensor product-based configuration. Like Nakamura *et al.*, they store a full set of point, line, face and cell values. By contrast, Yabe *et al.* (1991) construct an order-complete profile between each block of points in a scheme that later became known as ‘Type-A’. In 2D, because the 12 available moments (four point values, x-gradients and y-gradients) outnumber the 10 profile coefficients required, the x- and

y-gradients furthest away from the arrival point of interest are omitted from the reconstruction. A tensor-product profile could have been used instead, but then an extra set of moments – the xy-gradients – would be needed to make the profile well-determined. This is reminiscent of the trade-off between the Type-M and Type-C schemes.

The schemes discussed above are summarised in Table 2.2.

Table 2.2: Multidimensional Cartesian implementations of CIP and CIP-CSL.

Classification		Method	Source
split	uniform operators	CIP (Type-C)	Aoki (1995)
		CIP-CSL2	Nakamura *et al.* (2001)
	mixed operators	CIP (Type-M)	Takewaki and Yabe (1987)
		CIP-CSL2/3	Xiao *et al.* (2006)
unsplit	tensor product basis	CIP-CSL2	Takizawa *et al.* (2002)
	order-complete basis	CIP (Type-A)	Yabe *et al.* (1991)

Relative Performance

Despite the number of articles introducing multidimensional CIP and CIP-CSL schemes into the literature, the articles by themselves are of no use in assessing relative performance. Even when an article includes a standard test case with quantified errors (Nakamura *et al.*, 2001; Takizawa *et al.*, 2002), comparison with an equivalent scheme of a different class is rarely carried out. Nakamura *et al.* do compare CIP-CSL2 with a monotone-preserving alternative, R-CIP-CSL2, but we are less interested in monotonic qualities at this stage.

We also have little hope of comparing articles with the same test cases. The only recurring test cases with consistent parameters seem to be rotations of Zalesak's disk or some such scalar field. This is the multidimensional equivalent of a step function where a region of high values is separate from a field of low values by a C^0 discontinuity. The discontinuity is in the shape of a slotted disk, and the whole field is subjected to advection by a rotational velocity field. Schemes of higher accuracy will be able to maintain the shape of the discontinuity for longer. However, such test cases are unsuitable for comparing linear schemes since the discontinuities cause spurious oscillations and spoil convergence.

There are nevertheless a few articles that directly compare two or more existing schemes. Yabe *et al.* (2004) compare Type-C and Type-M CIP, but they implement the schemes on a non-Cartesian structured 2D grid. The grid resembles a convergent-divergent nozzle with straight edges. For a discontinuous cosine bell function advected at a shallow angle across the grid, Type-C approaches the expected third-order convergence faster as the grid size is reduced, with the consequence that it is 7–8 times more accurate than Type-M on fine grids. Yabe *et al.* (2004) see slower convergence for Type-M but claim that greater than second order accuracy is still obtained, which suggests that the presence of low-order operators does not necessarily impact the overall high-order accuracy.

A more useful analysis can be found in a communication by Konno *et al.* (2008). A 2D linear wave advection problem with Courant number 0.8 is considered. The dissipation and dispersion errors of Type-M CIP, Type-C CIP and Takizawa *et al.*'s CIP-CSL2 are computed for a range of propagation angles. As one would expect, the errors are identical for propagations in the x-direction ($\theta = 0°$) and y-direction ($\theta = 90°$), since the treatment of the non-aligned moments is of no consequence in these directions. As θ sweeps from 0° to 45°, the dissipation for Type-M CIP worsens, whereas for the other two schemes it gets better. The normalised phase speed for all schemes gets worse, although it appears to be slightly better for Type-M than for the other two.

Interestingly, the errors for Type-C CIP and Takizawa *et al.*'s CIP-CSL2 are identical. Recalling that the 1D CIP and CIP–CSL schemes yielded identical results in Yabe *et al.* (2001a), one might suppose that the 1D operators are spectrally equivalent, and furthermore that split schemes with uniform operators (such as Type-C CIP) are spectrally equivalent to unsplit schemes with tensor product bases (such as Takizawa *et al.*'s CIP-CSL2). This would not be surprising, because in either case there is a square (or cubic) number of moments whose evolution is purely CIP-based. One might go on to speculate whether there is also a correlation between split schemes with mixed operators and unsplit schemes with order-complete bases. It is assumed that the splitting error is subdominant or eliminated in the analysis, but Konno *et al.* do not leave details of their method open for inspection.

It is even more interesting to revisit the article by Hu *et al.* (1999) concerning the analysis of the DG method for wave propagation problems. The authors extend their work to 2D, analysing the discontinuous Galerkin method with both tensor product and order-complete bases. As the propagation angle sweeps from 0° to 45°, the dissipation rate for the tensor product basis gets better, whereas for the order-complete basis it gets worse. These behaviours compare favourably with those of Type-C CIP/CIP-CSL2 and Type-M CIP respectively. In §2.2.1 it was asked whether one-dimensional DG and multi-moment methods might be equivalent; we could extend the question to multidimensions. On the other hand, the dispersion accuracy does not show the same correlation, with the order-complete basis improving as θ sweeps from 0°. But it must be kept in mind that the analysis of Konno *et al.* concerns fully discrete schemes, while Hu *et al.* take a semi-discrete perspective. The analyses are not directly comparable.

One more article worth mentioning follows loosely from the work of Konno *et al.* (2008). Ara *et al.* (2012) compare Type-M and Type-C CIP in the simulation of a sound wave propagating radially outwards. As expected, Type-C was found to be more accurate, while Type-M was more economical in terms of memory. The authors also recorded computation time and found Type-M to take 15% less time than Type-C due to the substitution of lower-order operators.

We now have a better idea of the costs and benefits of various multidimensional extensions to multi-moment methods. The picture is not complete in that the CIP-CSL schemes remain largely uninvestigated in terms of their performance, but we can make reasonable predictions about a scheme's behaviour based on its classification. Additionally, we have uncovered an interesting research question in whether there is a connection between discontinuous Galerkin and multi-moment methods. This question will be addressed in Chapter 4.

Ideally, we would be able to perform a spectral analysis of any 2D multi-moment method. Besides eradicating doubt as to the performance of the CIP-CSL schemes, being able to perform such analyses would allow us to custom-design a scheme to occupy the background of a non-conforming grid implementation. This is exactly what Chapters 3, 5 and 6 set out to do.

2.3 Unstructured Grids

The following subsections review unstructured grids and their relationships with the finite volume, finite element and multi-moment methods. Finite difference schemes are excluded. While the flux of quantities between cells is accounted for by construction in the FVM, in the FDM it is not. Consistent, conservative stencils are difficult to form for arbitrary point arrangements, and for this reason the literature is devoid of FDM implementations.

The first subsection (§2.3.1) discusses how unstructured grids affect the FVM and FEM. The aim here is simply to highlight issues and techniques relevant to this thesis. The grids reviewed are exclusively of the simplicial (triangular or tetrahedral) kind. These have a particular relevance in that, were we to develop the cut-cell type of non-conforming grid, we could also decompose the cut cells into simplices and directly use the techniques under review.

The second subsection (§2.3.2) is mainly a case study of a multi-moment implementation on an unstructured, triangular grid.

2.3.1 Finite Volume and Finite Element Issues

In the finite volume method, the main threat to high-order accuracy is the unpredictable arrangement of cells. To compute the flux through a given cell face, a stencil must be constructed that spans more than just the immediate cell neighbours in order to sufficiently determine an order-complete flux profile. Unfortunately it is not possible to say in advance how many degrees of separation must be traversed before enough cells have been included. Even worse, it is possible that the selected cells lead to a rank-deficient matrix.

The standard approach (Barth and Frederickson, 1990) is to err on the side of too many cells. Instead of a matrix inversion, the method of least squares is used to recover the profile coefficients or cell weights. More sophisticated methods involve the ENO/WENO schemes introduced in §2.1.1. The least squares approach is worth noting as a way of resolving overdetermined profiles. Robust algorithms for cell selection, however, are not relevant to the present thesis.

The problem of how to shape the stencil is exacerbated at boundaries. There is a tendency for the stencil to be biased towards the domain interior, leading to weakly enforced boundary conditions. One way round this is to create 'ghost' cells, fictitious cells exterior to the domain, which nevertheless bring their own complications (Krivodonova and Berger, 2006). Ollivier-Gooch and Van Altena (2002) have a more elegant solution. Using curvature information about the boundary, boundary faces are populated with high-order quadrature points. Each point has an associated row in the reconstruction matrix representing a boundary condition. The associated point values are not evolved according to governing equations. When reconstructing the profile for a boundary cell, which is still overdetermined due to the presence of randomly arranged interior cells, a *constrained* least squares treatment is performed. Boundary conditions are constrained exactly, while interior cells are treated by least squares and therefore have a weaker influence. It remains to integrate the fluxes over the cell faces, which is facilitated by the boundary points also being quadrature points. Ollivier-Gooch and Van Altena give details of how to constrain the least squares method, but actually it is a standard linear algebra problem (Golub and Van Loan, 1983; Björck, 1996) that can be solved using modern numerical libraries.

The technique of Ollivier-Gooch and Van Altena may be regarded as a modification of the FVM to make it semi-compact at the boundary. As such, there is a reliance on curved grid elements. In §2.1.2, the question was posed as to whether a developer of a compact, high-order method could get away with using a piecewise-linear representation of the boundary. Looking to the DG community, the consensus seems to be an emphatic 'no'. Bassi and Rebay (1997) demonstrate that the use of linear elements impacts not only the accuracy near the boundary; it can have a disastrous effect on the solution as a whole. In a simple experiment involving inviscid flow over a sphere, a non-physical wake is observed which persists even after repeated grid refinements. Bassi and Rebay and other proponents of the DG method (Cockburn *et al.*, 2000) conclude that the use of high-order elements is not optional but mandatory.

In fact, problems with linear elements have already been known in the FEM for decades. Babuška (1963) proved that when a loaded circular solid plate is modelled as a polygon, the solution does not converge. The phenomenon became known as the circle–polygon paradox and gave rise to a new area of research. Fortunately it also prompted the mesh generation community to respond with techniques for generating and working with curved elements. Isoparametric elements were developed in the late 1960s. An isoparametric element is formed by mapping a linear element using a polynomial shape function. The mathematics for transforming the element's local coordinate system and modifying its quadrature are now standard; see e.g. Zienkiewicz and Taylor (2000).

At a higher level, and more current, is the idea of transforming a standard linear-element grid into a high-order one. This can be done by giving the elements elastic properties, populating them with nodes, and solving the resulting elliptic system as an energy minimisation problem. This does not get away from needing high-order boundary information: to provide the boundary constraint, either the original geometry specification must be available, or surface curvature must be reconstructed from a stencil of element vertices. But clearly it is attractive in that it defers the more difficult part of grid generation to off-the-shelf software. Additional optimisations can be built in to cope with anisotropic elements or strong curvatures. Implementations can be found in Sherwin and Peiró (2002), Oliver (2008), and Persson and Peraire (2009). The application to the present work may be limited, since a cut cell grid would not be as modifiable as an unstructured one, but the situation may arise where transformation of a cut cell's constituent simplices is called for.

Finally, Krivodonova and Berger (2006) propose a high-order boundary condition that relies on neither curved elements nor the original geometry. The method is based on the observation that streamlines should follow the contour of the physical boundary, not the numerical one. For a given element, this physical surface is reconstructed as a circular arc passing through local vertices. A numerical boundary condition is still imposed at quadrature points on the piecewise-linear boundary, but it is specified such that the velocity vector projected onto the *physical* surface is orthogonal to the *physical* surface normal.

The above discussion has highlighted techniques and pitfalls relevant to implementing a compact high-order method on a non-Cartesian grid. It has been necessarily broad, since unstructured methods make up a wide field, but a key point is that high-order representations of curved boundaries should be supported.

2.3.2 Multi-Moment Implementations

The two notable works implementing multi-moment methods on simplicial grids are from Ii *et al.* (2005) and Ii and Xiao (2010). In both works, grids are two-dimensional. Point values populating the edges and vertices of the triangular cell are used to constrain an order-complete profile, and conservation is achieved by using a flux formulation to evolve the cell-integrated average. A cubic profile leads to fourth order accuracy. This is the most convenient degree of polynomial because the 10 coefficients are well determined by one point value per vertex, two point values per edge, and the cell-integrated average. A quadratic profile for third order accuracy is also considered in the latter article. Neither of the articles deal with curved boundaries, but that is beyond their scope.

The articles differ mainly in their method of evolving the point values and storing the cell value. Ii and Xiao (2010), implementing MCV, use a point-wise Riemann solver to update the values. In both works a flux formulation updates the cell-integrated average, but in the MCV method the cell centroid value is stored instead of the integrated average. By construction, the centroid value is converted to the integrated average for the update operation, and converted back again for storage.

Because of the similarities in the space discretisation, only the article by Ii *et al.* (2005) will be reviewed further. This article is more interesting for being CIP-based; we can learn more about implementing the semi-Lagrangian procedure for this grid type. Note that, this being 2D, we shall regard edges and faces as synonymous in order to speak generally about face fluxes, cell-integrated averages, etc.

An implementation for CIP/MM–FVM is presented under the conservative advection equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{V}) = 0 \tag{2.7}$$

For the point values, this equation is split into linear and nonlinear parts according to (2.1) and advanced by fractional steps. The linear advection step is advanced using CIP. That is, for each arrival point, a departure point is found by tracking backwards along the trajectory according to the velocity and time step. Interpolation at the departure point using the local cell profile gives the new arrival point value. This is followed by the nonadvection step, but the authors stop here – the problem of computing velocity divergence in the unstructured grid is not addressed.

With the point values computed, it remains to integrate the fluxes over the cell faces. The authors use Gaussian quadrature points along the edges. Conveniently, two points are needed for fourth-order quadrature, so the arrival points on each edge can be made to coincide with the Gaussian points. The points are shown as black triangles in Figure 2.5. But the integration over the faces must also be integrated in time. The authors take advantage of the characteristics – in other words, the relationship $\partial_t \phi + \mathbf{V} \cdot \nabla \phi = 0$ along the trajectory – to treat the time integration as another spatial integration. And so the facial area to be integrated is extruded upwind according to the previously computed departure points, with four more quadrature points being created in the upwind region. The final integration is given by interpolating, weighting and summing the point fluxes at the upwind quadrature points.

The authors verify the method with linear advection of a plane sine wave, rotation of a Gaussian cone, and rotation of a square pulse. Regular and irregular triangular grids are tested. The linear advection test is repeated using discontinuous Galerkin and spectral volume methods for comparison. It is found that CIP/MM–FVM converges with fourth order accuracy in linear advection, which is consistent with the DG and SV methods. However, when the Courant number is 0.1, the L^∞-errors are 13.3 times those of DG on a regular grid (as previously noted on p. 29) and 17.2–20.4 on an irregular grid. SV's errors are somewhere between the two. Convergence of CIP/MM–FVM is not achieved for the rotational tests, but this is to be expected since the profiles contain discontinuities and the method is not slope-limited.

This article raises some questions. The first is that there is no mention of an algorithm for locating departure cells from departure point coordinates. In a Cartesian grid, cell location is trivial; a simple floating point number to integer conversion will give the desired $(i,\ j,\ k)$ address. But in an unstructured grid some logic is inevitably needed. Its runtime impact on the

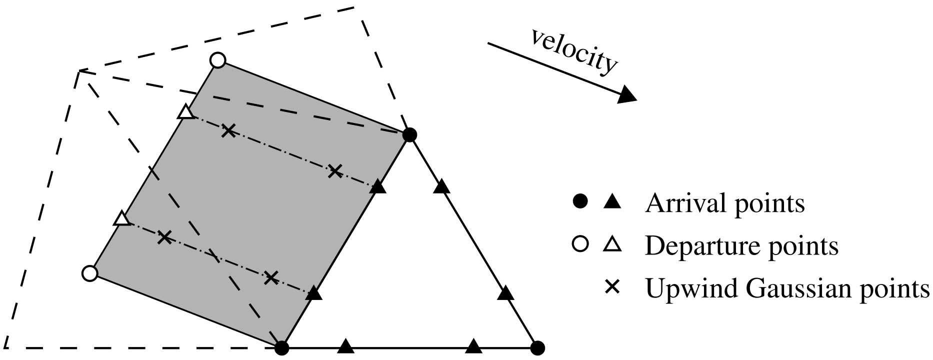


Figure 2.5: Reproduction of Figure 3 from li *et al.* (2005) showing how face fluxes are integrated in space and time.

above method is an open question, especially given the high number of interpolations. A good example of an algorithm comes from Xiu and Karniadakis (2001). The algorithm marches from cell to cell according to connectivity in the direction of the trajectory. For each cell, dot product operations test whether or not the departure point lies within all faces of the convex cell. It can be assumed that the tests are fast, and only a few cells will need to be traversed if the Courant number is small.

The second concern is that the nonadvection term has gone untreated. Indeed the authors do not need to treat it, since the verification tests involve divergence-free velocity fields wherein the term vanishes. It is important to take the nonadvection term into account when simulating the Euler and Navier-Stokes equations. An isotropic stencil would be needed. For example, in a Cartesian grid one would use a five-point (centre–north–south–east–west) stencil. In an unstructured grid the stencil is ambiguous. In this case the article of Ollivier-Gooch and Van Altena (2002) may be of help. Their method also relies on cell-wise profile reconstructions, but the profiles are used to compute viscous fluxes in addition to advective fluxes. On the face of interest, *two* flux quadratures are performed: one according to each cell profile on either side. The results are averaged. One could apply the same strategy to the edge values in Figure 2.5. Alternatively, one could take the IDO approach of spanning both cells in the profile reconstruction before differentiating at the face. For the vertex values, multiple cells are involved and it may be necessary to weight the results or the collocation point values.

Finally, the authors appear to rely on straight line trajectories to achieve high-order time integration of the fluxes. One may wonder whether the straight lines compromise temporal accuracy. Just as linear elements cannot ensure high-order boundary conditions, so linear trajectories cannot ensure high-order evolution in time. This can be shown from a Taylor series expansion of some moving point coordinates:

$$\begin{aligned}\mathbf{r}^{n-1} &= \mathbf{r}^n - \Delta t \frac{\mathrm{d}\mathbf{r}^n}{\mathrm{d}t} + \frac{\Delta t^2}{2} \frac{\mathrm{d}^2\mathbf{r}^n}{\mathrm{d}t^2} + \mathcal{O}\left(\Delta t^3\right) \\ &= \mathbf{r}^n - \Delta t \mathbf{V}^n \ + \frac{\Delta t^2}{2} \frac{\mathrm{d}\mathbf{V}^n}{\mathrm{d}t} + \mathcal{O}\left(\Delta t^3\right) \end{aligned} \tag{2.8}$$

which implies that the departure point $\mathbf{r}^{n-1}$ may be calculated to first order accuracy using the arrival point coordinates $\mathbf{r}^n$ and velocity $\mathbf{V}^n$, but substantial derivatives of the velocity are needed to achieve higher orders. One way around this is to approximate the substantial derivatives with spatial derivatives (McGregor, 1993). Xiu and Karniadakis (2001) express doubt about this approach in light of the non-monotonic semi-Lagrangian error (2.2). A more common strategy is to use a two-stage Runge-Kutta scheme such as the midpoint rule (Staniforth and Côté, 1991; McDonald, 1999).

Despite concerns about the order of trajectory integration, high-order integration only becomes important for the large Courant numbers afforded by purely semi-Lagrangian schemes – hence why most of the research on this topic comes from the metereological physics community. The scheme of Xiao *et al.* is a hybrid between the semi-Lagrangian CIP and Eulerian FVM. This would explain the combination of low-order time integration in the semi-Lagrangian step and high-order integration in the Eulerian step. The latter integration needs to be high for stability, and indeed Xiao *et al.* report stable Courant numbers of up to 0.8 on unstructured grids. Fourth order convergence is still observed at this Courant number, confirming that the temporal error remains subdominant.

Revisiting CIP/MM–FVM in one dimension in a subsequent paper, Ii and Xiao (2007) examine the complete time integration more rigorously. A total variation diminishing, third order Runge-Kutta (TVD RK3) integration is used. Like the midpoint rule, TVD RK3 relies on mid-level predictors, so it is assumed that the mid-level velocity is available or can be extrapolated accordingly. As before, Ii and Xiao update arrival point values by interpolating at the final departure points. But now they integrate the fluxes in time using the intermediate trajectory points as quadrature points. This makes sense and could easily be transplanted to the 2D case.

2.4 Cartesian Non-Conforming Grids

The merits of non-conforming grids were discussed in §1.1.2, but to recapitulate:

- In the bulk of the domain, grid elements are structured and fast to compute;
- Elements are of regular shape, promoting accuracy, stability, and element-wise refinement;
- When there are moving boundaries, the grid does not need a global regeneration at every time step.

Non-conforming grids fall roughly into two categories: cut cell methods and immersed boundary methods. Cut cell methods are the more conceptually straightforward. Cells on a Cartesian grid are cut by a sharp interface, and the computational stencil in those cells is modified to respect the boundary conditions on the interface. There are two problems at hand: not only are robust algorithms needed for manipulating cell geometry; there is no universal treatment for an arbitrary interior scheme. This has led to a large number of 2D implementations for the Euler equations (Berger and Leveque, 1989; De Zeeuw and Powell, 1993; Quirk, 1994), shallow water equations (Causon *et al.*, 2000; Ingram *et al.*, 2003; Liang *et al.*, 2007) and, to a lesser extent, Navier–Stokes equations (Udaykumar *et al.*, 1996; Ye *et al.*, 1999; Chung, 2006). Some of these are able to refine adaptively the cells near the boundary using quadtree algorithms, or handle moving solid bodies or other contact discontinuities. Cell merging or local time stepping is used to overcome the stiffening of the equations arising from the presence of small cut cells. Methods inevitably require more logic in 3D (Yang *et al.*, 2000; Luo *et al.*, 2012) due to the variety of ways a cell can be cut. However, in the present work, we are interested in the problem of scheme modification, *not* geometry manipulation.

In the immersed boundary method (Peskin, 1972), the flow field is discretised on a Cartesian grid of points, and the solid boundary is represented as a force field which is added to the source term in the governing equations. This is convenient because it allows the interior finite difference scheme to go unmodified; many existing solvers will support the immersed boundary method. However, the method relies on a semi-discrete representation of the Dirac δ-function which must be spread over several grid points, leading to a reduction in accuracy. The method was originally used to study fluid–structure interactions, with the solid boundary modelled as a set of Lagrangian points connected by arbitrarily rigid springs. For cases where the solid boundary is stationary or its motion is known explicitly, Goldstein *et al.* (1993) introduced a feedback controller for computing the appropriate force, albeit with stability issues and a need to apply smoothing at the boundary. Mohd-Yusof (1997) provide a more direct expression for the force by manipulating the time-discretised equations. Lai and Peskin (2000) implement the immersed boundary method with formal second order accuracy, but they still acknowledge a first-order error at the boundary.

Between the two types of Cartesian grid exist several hybrids. These aim to preserve the stencil of the interior scheme right up to the boundary, as the immersed boundary method does, while respecting boundary conditions exactly on the discretised interface, thus recovering cut cell accuracy. They are sometimes referred to as embedded boundary methods (Yang and Balaras, 2006). The seminal development came from Fadlun *et al.* (2000) who took Mohd-Yusof's formulation and added linear interpolations to achieve global second order accuracy. The interpolations are performed at the points or cells immediately outside the boundary. A variation is to interpolate at ghost points or cells inside the boundary (Kim *et al.*, 2001; Tseng and Ferziger, 2003).

In most of the above methods, piecewise-linear cuts are assumed and the midpoint rule is invoked for face fluxes and cell values. As discussed in §2.1.2 this leads to, at best, second order accuracy. High-order methods have started appearing in the literature, but in the context of finite volume/finite difference methods these involve manipulations of large stencils (Popescu *et al.*, 2008; Duan *et al.*, 2010). More interesting is the method by Kirkpatrick *et al.* (2003), which takes place on a staggered grid. Staggered grids are a standard approach for restoring pressure–velocity coupling in incompressible flows. Although the schemes here are still second order, they are supported by a quadric representation of the boundary which is generated at the preprocessing stage. This means that surface normals are correctly represented. One wonders whether this would solve the non-physical problems reported by the finite element community with respect to piecewise-linear boundary representations (§2.3.1).

In summary, developments in non-conforming grids have taken place mainly in the context of the FDM and FVM. We now go on to examine developments for compact methods.

2.4.1 Compact Implementations

In the finite element literature, implementations on non-conforming grids are commonly known as unfitted methods (Barrett and Elliott, 1987). The discontinuous Galerkin method has a notable advantage here. Since DG profiles do not need to be continuous between elements, they need not conform to the element geometry. The basis functions can therefore be mapped to the original, uncut elements. Exploiting this fact, Lew and Buscaglia (2008) are able to impose strong Dirichlet conditions and still achieve acceptable convergence. The same goes for Qin and Krivodonova (2012), who impose the streamline-based boundary conditions of Krivodonova and Berger (2006). Bastian and Engwer (2009) on the other hand choose to incorporate the boundary conditions in the weak DG formulation. In all cases, the cut elements are decomposed into their constituent simplices for the purpose of numerical integration only.

Despite the ability of weak formulations to handle cut elements elegantly, it is not obvious how this ability might transfer to multi-moment methods. The most promising implementation comes from Sakurai and Aoki (2001) who incorporate cut cells into the IDO method to evolve contact discontinuities. A level set function defines an interface between two materials. The function can be used to calculate points cutting the Cartesian gridlines. Values and gradients for each material are evolved as usual with IDO, but profile reconstructions are segregated by these cutting points. As shown in Figure 2.6, instead of a single continuous interpolation polynomial spanning the cell, two discontinuous polynomials exist on either side of the cutting point. Each material's field values and gradients must be stored at the cutting point in order to make this possible. At each time step, the level set function is advected according to IDO, the cutting points are recalculated, and the material fields are advanced according to IDO and the governing equations.

Every time the cutting points are moved, the values and gradients stored there have to be recalculated. Sakurai and Aoki perform linear interpolations which inevitably compromise IDO's high-order accuracy at the interface. Indeed in the rotation of a scalar field with a square-shaped C^0 discontinuity, the sharpness of the square corners is lost. However, if we limit our review to static boundaries, it seems as though Sakurai and Aoki's procedure does not suffer from low-order approximations. The split IDO operators are applied along Cartesian gridlines and require no further information about the interface geometry between cutting points.

The simplicity and apparent high-order accuracy of Sakurai and Aoki's cut cell method is certainly encouraging. However, it assumes that all field values and gradients are known at the interface. This may be true in C^0-discontinuous scalar field rotation tests, but what about advection around a solid obstacle? One would typically want to apply Neumann (normal gradient) boundary conditions for tangential velocity components and passive scalars. To do this requires extrapolating from the interior field to the cutting points along *orthogonal* paths. In Sakurai and Aoki's method, information is only available along Cartesian gridlines. It seems that we are back to the problem of how to reconstruct multidimensionally the flow field in the vicinity of the boundary: to preserve the method's nominal order of accuracy, we require high-order information about the interface geometry.

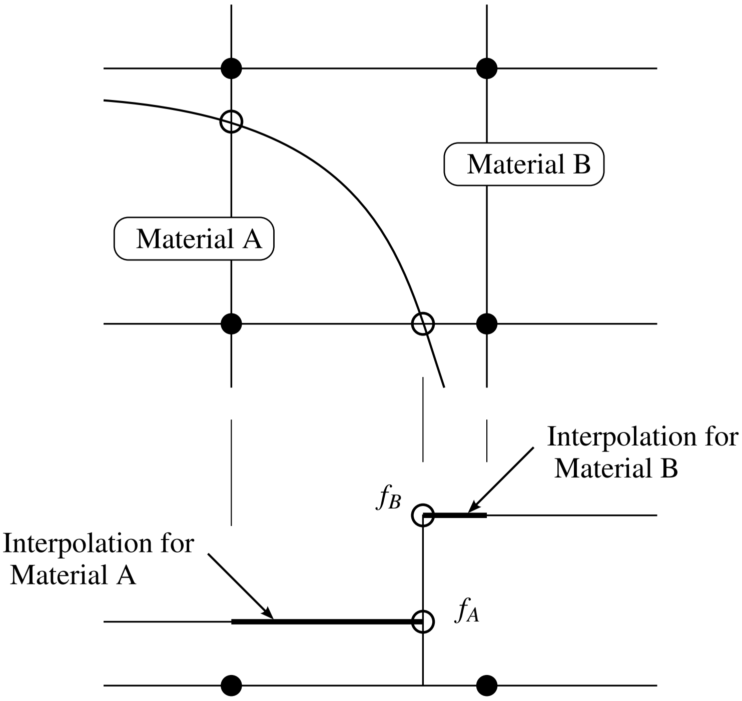


Figure 2.6: Interpretation of the 2D cut cell method of Sakurai and Aoki (2001), with their Figure 5 reproduced in the bottom part of the diagram to show 1D interpolations.

In other implementations, low-order compromises at the interface are easier to identify. Hu and Kashiwagi (2004) use CIP–CSL3 with a tangent hyperbola function for interface capturing. In boundary cells, a 'volume-fraction weighting' of velocity is applied. This is simply a linear blending between the solid body and the interior field.

Xiao *et al.* (2005) exploit the fractional area volume obstacle representation (FAVOR) method of Hirt and Sicilian (1985) to apply VSIAM3 to arbitrary boundaries. FAVOR introduces a scalar parameter to be assigned to cell faces which represents the fractional area open to the flow. However, this parameter is not sufficient to describe the geometric details of linear segments, let alone curved interfaces. Hirt and Sicilian acknowledge that FAVOR has first order accuracy at the boundary.

In contrast to the warnings of the finite element community regarding low-order approximations of the boundary, the above non-conforming multi-moment methods appear to give reasonable results. One explanation for this is that the presence of viscosity damps non-physical effects that might originate at cusps on the piecewise-linear interface. Another is that approximations such as FAVOR do not equate precisely to a piecewise-linear representation of the boundary, but rather to something more 'smeared'. In any case, one would be wise to support high-order boundary representations in the design of a non-conforming compact scheme. With low-order representations, simulations will at best be disadvantaged at resolving boundary layers. Boundary layers can be critical flow features, especially if they separate. At worst, simulations will result in non-physical, non-convergent solutions such as those presented by Bassi and Rebay (1997).

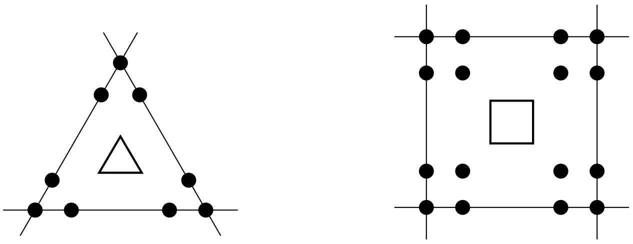
2.5 Conclusions and Research Objectives

We began this review with a study of existing multi-moment methods and their extension to multidimensions. The superiority of multi-moment schemes over traditional, non-compact alternatives is undeniable. But in §2.2 it was quickly found that fair comparisons between (i) schemes within the multi-moment family, and (ii) multi-moment methods and other compact methods such as the discontinuous Galerkin method, are severely lacking. A researcher wishing to implement a high-order numerical scheme in his or her solver would have difficulty making an informed decision as to whether a multi-moment method would be the right choice in the first place, let alone which of the multi-moment schemes to use. Additionally, the question was raised as to whether there was a connection between the multi-moment and discontinuous Galerkin methods. It may be concluded that a systematic comparison of multi-moment schemes, both between themselves and with the discontinuous Galerkin method, is in order. Such a comparison would be of interest to the academic community in addition to serving the end goal of the present research. The DG analysis by Hu *et al.* (1999) provides a convenient benchmark.

The end goal, to recapitulate, is to implement a multi-moment scheme on a Cartesian non-conforming grid. In §2.4 it was found that there are two ways of doing this: either apply a Cartesian operator to the entire grid and modify the source terms in the vicinity of the boundary, thus representing the boundary by a force field, or apply the Cartesian operator to interior cells and modify it (or find a compatible operator) in cells cut by the boundary. The first approach is difficult if not impossible to implement with greater than second order accuracy at the boundary – and in §2.3.1 it was argued that such approximations do not bode well in the context of compact methods.

It seems that the cut cell approach is the only sensible choice. Here the triangular cell-based scheme of Ii *et al.* (2005) becomes relevant. This scheme is economical for its fourth order accuracy: as discussed in §2.3.2 and shown in Figure 2.7a, the number of moments in the stencil exactly determines the 10 coefficients of the order-complete basis. It is at a disadvantage compared to the other CIP-based schemes in that it is partly Eulerian and thus has a restricted time step. But it is conservative, and importantly it has a multidimensional profile reconstruction that invites adaptation to arbitrarily shaped cells.

With the above scheme in mind, we have an incentive for finding a compatible fourth order Cartesian scheme for use in the background grid of uncut cells. For instance, one might choose a tensor product basis for the Cartesian cell. Figure 2.7b shows the natural arrangement of points. A constrained least squares treatment or omission of the farthest point would be needed during reconstruction due to the overdetermination of the system by the cell-integrated average.



(a) Scheme of Ii *et al.* (2005) **(b)** Cartesian equivalent

Figure 2.7: Fourth order multi-moment schemes. Dots represent point values; polygons represent cell-integrated averages.

However, when the arrangement in Figure 2.7b is expanded in two dimensions, the number of moments that need to be stored works out at 10 per grid cell (one vertex value, four edge point values, four cell point values, and the cell-integrated average). Compared with the third order multi-moment schemes, which only need to store three or four, this is an enormous penalty. Perhaps edge and cell point values could be taken away and replaced with point gradients at the vertices, since the latter are shared by more profiles. And perhaps the tensor product basis could be exchanged for an order-complete basis which would need fewer moments for support.

But could such modifications impair the accuracy or even the stability of the scheme? The literature does not provide answers, being concerned mainly with third-order or split schemes.

In summary, there are four major research objectives:

1. Compare Cartesian multi-moment schemes to analogous discontinuous Galerkin schemes.
2. Compare the performance of an array of existing Cartesian multi-moment schemes in multidimensions.
3. Design a stable, conservative, fourth order-accurate Cartesian multi-moment scheme to occupy the uncut background of a Cartesian cut cell grid.
4. Redesign the triangular cell-based scheme of Ii *et al.* (2005) for arbitrarily shaped cells.

In order to achieve these objectives, a methodology for analysing and testing numerical schemes is needed. This methodology is established in Chapter 3. Techniques such as Fourier/eigenvalue decomposition and grid convergence analysis, which were alluded to in the above review, now become central topics. While they are standard for finite difference and finite volume methods (Hirsch, 2007), Chapter 3 extends them to multi-moment methods – which are also exposed in more detail and analysed in their one-dimensional forms – and to multidimensions, in the spirit of Hu *et al.* (1999). Additionally, some measures of performance and other conventions are established. The resulting methodolody equips us to handle the Cartesian multi-moment schemes in Objectives 1–3. Some modifications will be needed to achieve Objective 4, but they will be introduced in due course. The four objectives are addressed in turn in Chapters 4 through 7.

Chapter 3

Theory and Methods

3.1 Introduction

The purpose of the present chapter is to establish a framework by which multi-moment schemes can be analysed and tested. In §2.2.1, the idea of stability analysis was introduced. It was explained that there were two methods available, the von Neumann analysis and the matrix method, and that the principles behind both methods can be applied to both semi-discrete and fully discrete equations. The outcome of the analysis is a spectrum that describes the dispersion and dissipation properties, and by extension the stability, of a scheme.

It was argued that we should be concerned with the semi-discrete analysis, not the fully discrete one, because the multi-moment schemes under review are characterised by original approaches to the space discretisation: their methods of time integration remain standard. However, most of the multi-moment literature concerns *semi-Lagrangian* schemes that are not technically possible to semi-discretise. Although these schemes tend to be limited to either forward Euler or third order Runge-Kutta (RK3) integration of the departure point, the choice of integration is somewhat arbitrary and can make comparisons unfair. Worse, the spectra of fully discrete schemes are dependent on the Courant number, adding another variable to take into account.

To get around this dilemma, we might consider again the relationship between semi-Lagrangian and Eulerian schemes as demonstrated in Figure 2.3 (p. 26). It was observed that, when modelling linear advection, the properties of a semi-Lagrangian scheme converge on those of an equivalent Eulerian scheme as the Courant number goes to zero. This may be explained as follows. In either case, a polynomial $U(x)$ is constructed between grid points. For example, the constrained interpolation profile method (CIP) and its Eulerian equivalent, the interpolated differential operator method (IDO), both construct a cubic curve. The schemes differ in how quantity u, stored at grid point i, is updated from time level n to time level $n+1$. The semi-Lagrangian scheme interpolates the polynomial at departure point $x_i - a\Delta t$, where a is the advection velocity and Δt the time step.

$$u_i^{n+1} = U(x_i - a\Delta t) \tag{3.1}$$

In contrast, the Eulerian scheme differentiates the polynomial to derive fluxes at the grid point. The semi-discrete representation when modelling linear advection is

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -a \left.\frac{\mathrm{d}U(x)}{\mathrm{d}x}\right|_{x_i} \tag{3.2}$$

Now suppose Δt in Equation (3.1) is brought to zero. The semi-Lagrangian scheme approaches a semi-discrete form as

$$\begin{aligned}\frac{\mathrm{d}u_i}{\mathrm{d}t} &= \lim_{\Delta t \to 0} \frac{u_i^{n+1} - u_i^n}{\Delta t} && (3.3)\\ &= \lim_{\Delta t \to 0} \frac{U(x_i - a\Delta t) - U(x_i)}{\Delta t} && (3.4)\end{aligned}$$

and, after substituting Δt with $\sigma \Delta x / a$, where σ is the Courant number, Δx is the grid spacing, and a and Δx are constant,

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = a \lim_{\sigma \Delta x \to 0} \frac{U(x_i - \sigma \Delta x) - U(x_i)}{\sigma \Delta x} \tag{3.5}$$

By inspection, Equation (3.5) approaches Equation (3.2) as $\sigma \to 0$. This explains why the properties of a semi-Lagrangian scheme converge on those of its Eulerian counterpart as the Courant number is reduced. One may conclude that to derive the spatial properties of a multi-moment scheme, it suffices to replace the scheme's semi-Lagrangian operations with Eulerian ones before subjecting the scheme to a semi-discrete analysis.

This strategy of converting semi-Lagrangian formulations to Eulerian ones will be adopted throughout the thesis whenever a multi-moment scheme is analysed. To avoid confusion, converted schemes will be renamed such that the abbreviation ‘IDO’ replaces each instance of ‘CIP’. For example, Type-M CIP becomes Type-M IDO. There is a complication when it comes to renaming the conservative semi-Lagrangian (CIP–CSL) schemes, because the CSL suffix makes little sense following the Eulerian conversion. We shall instead call the Eulerian version IDO–FVM, since it can be considered a hybrid between IDO and the finite volume method. In multidimensions we will run into further naming issues as the conservative CIP-based schemes divide into two subgroups, but these issues will be dealt with in the next chapter.

The present chapter outlines the theory behind analysis and testing in 1D (§3.2), how analysis may be extended to multi-moment schemes (§3.3), the theory behind analysis and testing in 2D (§3.4), and finally, some conventions and techniques that will be adopted in the following chapters (§3.5).

All analysis in the present work was done using MATLAB® R2009a. Double precision was used everywhere; the smallest representable floating point number (machine epsilon) was 2.2204 $\times 10^{-16}$.

3.2 One Spatial Dimension

In the next subsection, stability analysis of semi-discrete systems is expanded upon in more technical detail. We follow the approach of Hirsch (2007, Ch. 9). Here the matrix method is introduced first, due to its generality, before it is replaced by a Fourier decomposition in the spirit of the von Neumann analysis. Hirsch limits his discussion to finite difference/finite volume schemes, but it will be shown in §3.3 that that the matrix method can be combined with the Fourier decomposition to represent schemes involving more than one type of moment.

Fully discrete analysis will be explained in less depth, although it is still relevant in the context of time-domain simulations. It is useful to know how a semi-discrete spectrum may be augmented with a given time scheme to yield a numerical amplification factor, since it provides an intuitive means to determine the stability condition.

Finally, we turn our attention from the frequency domain to the time domain. Time-domain testing is relevant in that (i) simulation results can be used to verify the frequency-domain analysis, and (ii) we can extend to nonlinear situations and check that the observed rate of convergence continues to match the nominal order of accuracy. In this last subsection it will also be shown how the errors arising from linear advection can be predicted using an expression derived from the spectrum. This expression is particularly useful in that it provides an alternative to the K_c metric of Hu *et al.* (1999) as a means of quantifying the performance of schemes.

3.2.1 Semi-Discrete Analysis

Suppose we have a grid of points, and we want to evolve the point values in time according to some linear differential equation. The problem can be expressed as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{S}\mathbf{u} + \mathbf{Q} \tag{3.6}$$

where **u** is a column vector containing all the point values, **S** is a matrix representing the space discretisation, and **Q** is a source term containing contributions from the boundary conditions. To simplify the analysis, we insist on a uniform mesh and periodic boundary conditions. One consequence of this is that **Q** vanishes.

As an example, consider a 1D grid of point values evolved according to linear advection using the first order upwind (FOU) scheme. The linear advection equation is

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x} \tag{3.7}$$

which is semi-discretised with the FOU operator as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -a\frac{u_i - u_{i-1}}{\Delta x} \tag{3.8}$$

Equation (3.6) becomes

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -\frac{a}{\Delta x}\begin{pmatrix} 1 & & & & & -1 \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & -1 & 1 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_n \end{pmatrix} \tag{3.9}$$

Subjecting $\mathbf{S}$ to an eigendecomposition yields n eigenvalues and eigenvectors. It requires the solution of

$$\mathbf{S}\mathbf{V} = \mathbf{V}\mathbf{D} \tag{3.10}$$

where $\mathbf{V}$ is the matrix of eigenvectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(n)}$, arranged in columns, and $\mathbf{D}$ is the diagonal matrix of complex eigenvalues $\omega_1, \omega_2, \ldots, \omega_n$.

The exact solution to Equation (3.6) may be expressed in terms of an exact solution $\tilde{\mathbf{u}}(t)$ in the modal space,

$$\mathbf{u}(t) = \mathbf{V}\tilde{\mathbf{u}}(t) \tag{3.11}$$

and because the modal system is formulated as a system of independent, homogeneous differential equations, it can be solved straightforwardly.

$$\frac{\mathrm{d}\tilde{\mathbf{u}}}{\mathrm{d}t} = \mathbf{D}\tilde{\mathbf{u}}(t) \tag{3.12}$$

$$\tilde{\mathbf{u}}(t) = \mathbf{E}(t)\,\tilde{\mathbf{u}}^0 \tag{3.13}$$

$\mathbf{E}(t)$ is a diagonal matrix containing $\mathrm{e}^{\omega_1 t}, \mathrm{e}^{\omega_2 t}, \ldots, \mathrm{e}^{\omega_n t}$, and $\tilde{\mathbf{u}}^0$ is the initial condition. The exponential terms imply that *the real part of all eigenvalues must be zero or negative to ensure stability.*

The periodic boundary conditions mean that the eigendecomposition produces the same result as a Fourier decomposition. Eigenvector $\mathbf{v}^{(\alpha)}$ describes a Fourier mode with wavenumber k_α or phase ϕ_α. The *i*th element of the eigenvector corresponds to the *i*th point on the grid; it relates to k_α and ϕ_α as

$$v_i^{(\alpha)} = \mathrm{e}^{\mathrm{I}i\phi_\alpha} = \mathrm{e}^{\mathrm{I}k_\alpha x_i} \tag{3.14}$$

$$\phi_\alpha = k_\alpha \Delta x = \frac{\alpha\pi}{n} \quad \alpha = -n, \ldots, n \tag{3.15}$$

where I is the imaginary number $\sqrt{-1}$. Hirsch points out that the Fourier decomposition is the main strategy behind the von Neumann analysis. The difference here is that we have not yet introduced the time integration.

Relationship (3.14) concerns a finite set of wavenumbers, but it can be generalised to a continuous range. This has the useful consequence that the space operator matrix collapses to some scalar expression $S(K)$, where $K := k\Delta x$. For example, the FOU matrix equation (3.9) can be reformulated and decomposed as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{a}{\Delta x}\begin{pmatrix} 1 & -1 \end{pmatrix}\begin{pmatrix} u_i \\ u_{i-1} \end{pmatrix} \tag{3.16}$$

$$S(K)\,\mathrm{e}^{\mathrm{I}Kx_i/\Delta x} = -\frac{a}{\Delta x}\left(1-\mathrm{e}^{-\mathrm{I}K}\right)\mathrm{e}^{\mathrm{I}Kx_i/\Delta x} \tag{3.17}$$

Although the characteristics of the space scheme are encapsulated entirely by $S(K)$, we shall henceforth characterise space schemes in terms of nondimensional weights w_α. The FOU operator above is characterised by $w_0 = 1$ and $w_{\overline{1}} = -1$, corresponding to u_i and u_{i-1} respectively. The overline on the weight subscript has been used to denote a negative α and hence an upwind position relative to i. This is simply a convention that will make it easier to group subscripts as more dimensions are introduced.

Putting w_α and u_α into row and column vectors $\mathbf{w}$ and $\mathbf{u}$, the semi-discrete equation can be rewritten

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{a}{\Delta x}\mathbf{w}\,\mathbf{u} \tag{3.18}$$

The writing of $\mathbf{w}\,\mathbf{u}$ instead of $\mathbf{w}^\intercal\mathbf{u}$ is somewhat awkward, but row vectors such as $\mathbf{w}$ will feature heavily in this thesis. Allowing them to be written like column vectors will help prevent a proliferation of transpose operators later on.

Keeping in mind that the subscript α refers to relative point position, not element index for the column vector, $S(K)$ is recovered by

$$S(K) = -\frac{a}{\Delta x}\sum_\alpha w_\alpha \mathrm{e}^{\mathrm{I}\alpha K} \tag{3.19}$$

From (3.10) and (3.14), it is clear that $S\mathrm{e}^{\mathrm{I}Kx_i/\Delta x} = \omega\mathrm{e}^{\mathrm{I}Kx_i/\Delta x}$, and so (3.19) can be used to generate the spectrum of frequencies. We finally nondimensionalise with $\Omega := \omega\Delta t$ (differing from Hirsch, who keeps Ω dimensional) and substitute the Courant number σ to arrive at

$$\frac{\Omega(K)}{\sigma} = -\sum_\alpha w_\alpha \mathrm{e}^{\mathrm{I}\alpha K} \tag{3.20}$$

In the FOU example,

$$\begin{aligned}\Omega(K)/\sigma &= -1+\mathrm{e}^{-\mathrm{I}K}\\ &= -1+\cos K-\mathrm{I}\sin K \qquad (3.21)\end{aligned}$$

which is a unit circle centred on $(-1,0)$ in the complex plane of Ω/σ. Its stability (before time discretisation) is indicated by the fact that $\Omega_r(K) \leq 0$ for all K.

We may go further than Hirsch and use the spectrum to get the order of accuracy of the space operator. If $\mathcal{L}(u)$ represents the numerical approximation to $\partial u/\partial x$, then by the governing equation (3.7)

$$\mathcal{L}(u) = -\frac{1}{a}\frac{\mathrm{d}u}{\mathrm{d}t} \qquad (3.22)$$

In the modal space, $\mathrm{d}\tilde{u}/\mathrm{d}t = \omega(k\Delta x)\,\tilde{u}(t)$. Substituting this gives

$$\begin{aligned}\mathcal{L}(\tilde{u}) &= -\frac{\omega(k\Delta x)}{a}\tilde{u}(t) \qquad (3.23)\\ &= -\frac{\Omega(k\Delta x)}{\sigma\Delta x}\tilde{u}(t) \qquad (3.24)\end{aligned}$$

Since $\tilde{u}(t)$ is not a function of physical space, we can discard it and go on to perform a Maclaurin series expansion of $\Omega(k\Delta x)/\sigma\Delta x$ in terms of Δx. There will be a leading $-\mathrm{I}k$ term, representing the exact dispersion relation, plus a series of terms corresponding to the truncation error. The lowest power of Δx in the series indicates the order of accuracy. Real parts correspond to dissipation; imaginary parts to dispersion. In the FOU example,

$$\frac{\Omega(k\Delta x)}{\sigma\Delta x} = -\mathrm{I}k - \frac{k^2\Delta x}{2} + \mathrm{I}\frac{k^3\Delta x^2}{6} + \mathcal{O}\left(\Delta x^3\right) \qquad (3.25)$$

indicating that the space operator is overall first order accurate, with first order accuracy in dissipation and second order accuracy in dispersion.

When an analytical expression for the spectrum is too difficult to derive, it is still possible to infer the orders numerically. Let $\varepsilon_{T\,r}$ and $\varepsilon_{T\,i}$ correspond to the real and imaginary parts of the truncation error, respectively. Then

$$\begin{aligned}\varepsilon_{T\,r}(K) &= \Omega_r(K)/\sigma \qquad (3.26)\\ \varepsilon_{T\,i}(K) &= -\left(\Omega_i(K)/\sigma + K\right) \qquad (3.27)\end{aligned}$$

Recall that $K := k\Delta x$, so as $K \to 0$ we would expect the dissipation or dispersion error to decrease at a rate according to $p+1$, where p is the corresponding order of accuracy. The $+1$ compensates for the Δx in the denominator of $\Omega(k\Delta x)/\sigma\Delta x$. Specifically, if one were to choose a small, arbitrary wavenumber K_a, then the error after a factor-two refinement should have decreased by

$$\frac{\varepsilon_T\left(K_a\right)}{\varepsilon_T\left(K_a/2\right)} \approx 2^{p+1} \tag{3.28}$$

Therefore the order of accuracy is verified by evaluating

$$p \approx \log_2\left(\frac{\varepsilon_T\left(K_a\right)}{\varepsilon_T\left(K_a\big/2\right)}\right) - 1 \tag{3.29}$$

When it comes to computing and plotting the spectrum of a space discretisation, there are some subtle issues on how to represent the dispersion relation and how to identify the existence of additional modes. These issues will be discussed in the analysis of IDO (§3.3.1).

3.2.2 Fully Discrete Analysis

We shall skip over the von Neumann analysis and instead show how the space discretisation may be coupled to a time integration method to yield final information about the scheme's stability, dissipation and dispersion properties. This is relevant when it comes to testing multi-moment schemes, because we want to be able to select a suitable time integration method and know the resulting stability criteria. However, it must be kept in mind that the approach below is only valid for Eulerian formulations. The analysis of semi-Lagrangian schemes cannot be separated into space and time components: it must be done in one step according to, say, the von Neumann analysis. We do not go into the procedure here but instead defer to §3.3, where the fully discrete CIP analysis of Utsumi *et al.* (1997) is reproduced.

A time scheme will have a representation in the solution space, but it can also be written in terms of some modal quantity at various time levels $\tilde{u}^{n+k}$ and the spatial eigenvalues Ω. For instance, the forward Euler method is defined respectively in the solution space and modal space by

$$u^{n+1} = u^n + \Delta t \mathcal{L}\left(u^n, t_n\right) \tag{3.30}$$

$$\tilde{u}^{n+1} = \tilde{u}^n + \Omega \tilde{u}^n \tag{3.31}$$

There will be an associated amplification factor $G(F,\Omega)$, which is a function of a 'time shift operator' $\tilde{E}$ and the spatial eigenvalues. G is formulated by rearranging the modal representation of the time scheme for $\tilde{u}^{n+1}\big/\tilde{u}^n$ and replacing other instances of $\tilde{u}^{n+k}$ by $\tilde{E}^k$ ($\tilde{E}$ to the power of k). G is then decomposed into its eigenvalues z according to

$$z = G(z,\Omega) \tag{3.32}$$

The forward Euler method, for example, gives

$$G\left(\tilde{E},\Omega\right) = 1+\Omega \tag{3.33}$$

$$z = 1+\Omega \tag{3.34}$$

Stability demands that $|z| \leq 0$ for all z. Since z is a function in Ω, its magnitude can be traced on the complex Ω-plane together with the spectrum of the space discretisation. The latter must be able to fit inside the trace of $|z|$ to ensure stability. And since $\Omega\big/\sigma$ is self-similar for Eulerian schemes, the spectrum can often be scaled to fit according to the Courant number.

In the case of forward Euler, the stable region is

$$(1+\Omega_r)^2 + \Omega_i^2 \leq 1 \tag{3.35}$$

which is a unit circle centred on $(-1,0)$ in the complex Ω-plane. This suggests that coupling with the first order upwind scheme will be stable up to $\sigma = 1$. Indeed,

$$\begin{aligned}
|z| &= \left[1+\sigma\left(\cos K - 1\right)\right]^2 + \left[-\sigma \sin K\right]^2 \\
&= 1 - 2\sigma + 2\sigma^2 + 2\sigma\left(1-\sigma\right)\cos K \\
&= \begin{cases} 1 & \cos K = 1 \\ (2\sigma-1)^2 & \cos K = -1 \end{cases} \\
\therefore \quad |z| &\leq 1 \quad \text{for} \quad 0 \leq \sigma \leq 1
\end{aligned} \tag{3.36}$$

The dissipation error of the fully discretised system is the ratio of the numerical gain to the exact gain – the latter being unity – while the dispersion error is the ratio of the numerical phase to the exact phase.

$$\varepsilon_{diss} = |z| \tag{3.37}$$

$$\varepsilon_{disp} = -\frac{\angle z}{\sigma K} \tag{3.38}$$

3.2.3 Performance in the Time Domain

Unlike a frequency-domain analysis, a simulation in the time domain does not directly yield information about the numerical scheme's dissipation and dispersion characteristics. Rather, the outcome is a numerical solution u_{N_i} at each grid point (or cell) that departs from the exact solution u_{E_i} by some error. If the exact solution is known, the errors may be computed as $\varepsilon_i := u_{N_i} - u_{E_i}$ and aggregated over the grid according to some norm. This thesis defines the generalised L^{λ}-norm of $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ as

$$\|\varepsilon\|_{\lambda} := \left(\frac{1}{n} \sum_{i=1}^{n} |\varepsilon_i|^{\lambda} \right)^{1/\lambda} \tag{3.39}$$

which is also known as the mean absolute error when $\lambda = 1$, the root mean square error when $\lambda = 2$ and the maximum absolute error when $\lambda \to \infty$.

Examining Spatial Convergence

One way of verifying the frequency-domain analysis is to check that an equivalent time-domain simulation demonstrates the same order of convergence. In §3.2.2 it was explained that as $K \to 0$, $\Omega(K)$ converges on the exact solution IK at a rate dependent on the order of accuracy. A similar relationship can be found in the time domain. If a series of simulations with successively decreasing grid spacings is set up, it should be observed that a continuum solution is converged upon – i.e. the errors go to zero – at a rate dependent on the order of accuracy.

If simulations are performed on two grids A and B of differing resolutions, and $\|\varepsilon\|_{\lambda}$ is calculated for each, then the order of accuracy should be observed from

$$p \approx \log \left(\frac{\|\varepsilon_A\|_{\lambda}}{\|\varepsilon_B\|_{\lambda}} \right) \Big/ \log \left(\frac{\Delta x_A}{\Delta x_B} \right) \tag{3.40}$$

It is interesting to note that Takacs (1985) takes the mean square error and splits it into dissipative and dispersive components. This technique was adopted by Nakamura *et al.* (2001) to showcase the properties of a 2D conservative CIP variant.

$$E_{tot} := \frac{1}{n} \sum_{i=1}^{n} \left(u_{E_i} - u_{N_i} \right)^2 \tag{3.41a}$$

$$\approx \mathrm{var}\left(u_E - u_N \right) + \left(\mu_E - \mu_N \right)^2 \tag{3.41b}$$

$$= \underbrace{\left(\sigma_E - \sigma_N \right)^2 + \left(\mu_E - \mu_N \right)^2}_{E_{diss}} + \underbrace{2\left(1 - \rho_{E,N}\right) \sigma_E \sigma_N}_{E_{disp}} \tag{3.41c}$$

In the above equation, μ refers to the mean, σ to the standard deviation, and ρ to the correlation coefficient. The dimensionality of the E-metrics is that of $\|\varepsilon\|_{\lambda}$ squared, so hypothetically one could examine the convergence of $\sqrt{E_{diss}}$ and $\sqrt{E_{disp}}$ and expect to see the corresponding orders of accuracy in the spectrum. Note that (3.41a) is written as equal to (3.41b) in Takacs' paper, but the equality is not true as long as n is finite. Rather, the statements are statistically consistent and converge on one another as $n \to \infty$.

A possible objection to the above verification is that it does not take into account errors arising from the time discretisation. This concern was also raised in §2.3.2. According to Equations

(2.2)–(2.4), the overall error of an Eulerian or semi-Lagrangian scheme is dependent on both Δx and Δt. Fortunately, however, the influence of Δt turns out to be subdominant at small enough Courant numbers. This will be demonstrated in §3.3.2. Errors for Eulerian formulations in particular appear to be insensitive to the time discretisation for all stable simulations.

Prediction of Linear Advection Errors

Since each computed spectrum is based on a space discretisation of the linear advection equation, the spectrum and an accompanying time scheme should provide all the information needed to predict the evolution of a periodic wave in the time domain. In other words, $||\varepsilon||_\lambda$ can be predicted exactly from the frequency-domain analysis. This offers a more precise means of verification than matching the rates of convergence.

In fact, with a few simplifying assumptions, an approximation to $||\varepsilon||_\lambda$ can be derived which depends solely on two simulation parameters: the nondimensional wavenumber K, and the scaled spectrum $\Omega(K)/\sigma$ which is invariant for a given space discretisation. Such a metric will be able to predict $||\varepsilon||_\lambda$ in relation to other schemes. Most usefully, this means it can be used to quantify scheme performance. It may be recalled that the other metric quantifying performance is Hu *et al.*'s (1999) K_c, whose underlying criteria (2.5) is somewhat arbitrary. Having an alternative metric enables us to gauge scheme performance with more confidence.

The first assumption is exact time integration. As mentioned above, the influence of the time discretisation is often subdominant at practical Courant numbers. Another assumption is that the number of grid points m is infinite, which means that the summation in (3.39) can be replaced by a continuous integration over the wavelength. A third assumption is that modes other than the physical one will be heavily damped and can therefore be neglected.

Let U be the amplitude of the wave. An exact solution at grid point i and time level n is

$$u^n_{E_i} = U\mathrm{e}^{\mathrm{I}k(x_i - at_n)} \tag{3.42}$$

The numerical solution is a multiplication of (3.42) at time zero by the numerical amplification factor $G(\Omega)$, repeated n times. The amplification factor depends on the time scheme as discussed in §3.2.2, but exact time integration is simply represented by $G(\Omega) = \mathrm{e}^{\Omega(K)}$. Then

$$u^n_{N_i} = [G(\Omega)]^n U\mathrm{e}^{\mathrm{I}kx_i} = U\mathrm{e}^{n\Omega(K)+\mathrm{I}kx_i} \tag{3.43}$$

The pointwise error is given by the difference between (3.42) and (3.43). Further simplifications can be made by setting $k = 2\pi/x_m$ and defining $N := at_n/x_m$ and $\xi := x_i/x_m$. The pointwise error can then be written

$$\varepsilon(K,\xi) = U\left[\exp\left(\frac{2\pi N\Omega(K)}{\sigma K}\right) - \exp(-2\pi \mathrm{I} N)\right]\exp(2\pi \mathrm{I}\xi) \tag{3.44}$$

$$\equiv C(K)\,\mathrm{e}^{2\pi \mathrm{I}\xi} \tag{3.45}$$

which conveniently removes the dependence on temporal parameters and has the spectrum in its invariant form, $\Omega(K)/\sigma$. The norm of ε is expressed in terms of an integration over the wavelength,

$$\|\varepsilon(K,\xi)\|_\lambda = |C(K)|\left(\int_0^1 \left|\mathrm{e}^{2\pi \mathrm{I}\xi}\right|^\lambda \mathrm{d}\xi\right)^{1/\lambda} \tag{3.46}$$

but this expression collapses to $|C(K)|$ by virtue of $\left|\mathrm{e}^{2\pi \mathrm{I}\xi}\right|$ being unity for all ξ. The resulting metric, which is no longer a function of λ, is hereby referred to as the physical-mode error norm, $\|\varepsilon(K)\|^{\text{phys}}$.

$$\|\varepsilon(K)\|^{\text{phys}} := U\left|\exp\left(\frac{2\pi N\Omega(K)}{\sigma K}\right) - \exp(-2\pi \mathrm{I} N)\right| \tag{3.47}$$

To compare against the equivalent simulation, $\|\varepsilon(K)\|^{\text{phys}}$ would be evaluated at the simulation's nondimensional wavenumber given by $K := k\Delta x = 2\pi/m$. Unless stated otherwise, the simulation parameters U and N are assumed to be unity; i.e. the wave travels at unit velocity for a distance of one wavelength.

3.3 Introductory Study of 1D Multi-Moment Schemes

In this section, one-dimensional multi-moment schemes will be analysed and tested. The first subsection introduces four schemes – CIP, IDO, IDO–FVM and MCV – and computes their properties in the frequency domain. The second subsection compares the time-domain behaviour of CIP and IDO with that of their noncompact equivalents.

3.3.1 Formulations and Analysis

The analytical method outlined previously needs to be extended to include more than one type of moment. The Fourier decomposition still plays an important role in reducing the system to a continuous spectrum in K, but this time there is another level of discretisation. It is a nested problem: for each K, a square matrix can be formed describing the relationship between the moments. In general, n moment types require a $n \times n$ matrix. We have to appeal to the matrix method to break the solution down into eigenvalues, each of which will be a continuous function in K. These functions may or may not join up to form a single physical mode, but this point will be discussed on p. 62.

To show how it may be done, we start with CIP and reproduce the analysis of Utsumi *et al.* (1997). This provides an opportunity to introduce the original CIP algorithm, but analysis is necessarily limited to the fully discrete system. We subsequently replace all the semi-Lagrangian operations with Eulerian ones, turning CIP into the IDO scheme. A semi-discrete analysis can then be performed. The semi-discrete system is straightforwardly transformed into a fully discrete system, allowing comparisons to be made between the two formulations at different Courant numbers.

Then we introduce the modification by which CIP becomes CIP–CSL. Since we are looking to move away from fully discrete analysis, CIP–CSL is converted into the Eulerian scheme IDO–FVM. Again it is emphasised that our end goal is to uncover the spatial properties behind such schemes, making the semi-discrete analysis preferable to a fully discrete one.

Finally, we examine the multi-moment constrained finite volume method (MCV). This method is relevant because it allows cells to have discontinuous profiles, and so it is most closely related to the discontinuous Galerkin method. Since the purpose of the next chapter is to compare multi-moment methods with the DG method, MCV is analysed briefly here.

As a side note, most eigendecomposition algorithms such as MATLAB's® EIG function do not return modes in a consistent order over a range of K. The effect can be seen in the jumps in the curves in Utsumi *et al.*'s paper. To resolve this, the function of d'Errico (2009) has been used throughout this thesis to ensure that continuous curves appear in plots.

Constrained Interpolation Profile Method (CIP)

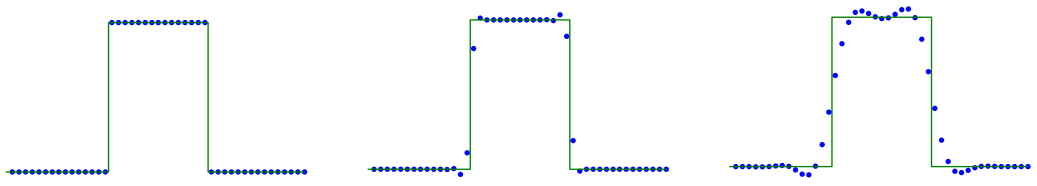
The original CIP method evolves two types of moment by a semi-Lagrangian formulation: the undifferentiated quantity f and its first order spatial derivative, or gradient, g. At each time step, a cubic function $F(x)$ is fitted between each pair of grid points using the four available moments f_i, g_i, f_{i-1}, g_{i-1}. The point of interest i is also referred to as the arrival point. $i-1$ is immediately upwind. If the wavespeed is a, then the values of f_i and g_i at time level $n+1$ are updated as

$$f_i^{n+1} = F^n(x_i - a\Delta t) \tag{3.48}$$

$$g_i^{n+1} = G^n(x_i - a\Delta t) = \frac{\partial}{\partial x}F^n(x_i - a\Delta t) \tag{3.49}$$

where $x_i - a\Delta t$ gives the position of the departure point by forward Euler integration. Other integration methods are possible, but when the velocity field is uniform and constant, all methods evaluate to the exact departure point.

The linear advection of a step function with CIP is demonstrated in Figure 3.1. Numerical dispersion manifests as spurious oscillations on either side of the discontinuities, while dissipation causes the profile to gradually lose its sharpness over time.



(a) $n=0$ **(b)** $n=10$ **(c)** $n=1000$

Figure 3.1: Linear advection with the semi-Lagrangian CIP scheme at $\sigma=0.8$. The point spacing and time step size were both unity, and the step function was initialised to have a width of 15.

It simplifies analysis and improves conditioning if profiles are written in terms of nondimensional coordinates. $F(x)$ and $G(x)$ become $F(\xi)$ and $G(\xi)$, with ξ defined by

$$\xi := \frac{x-x_i}{\Delta x} \tag{3.50}$$

We shall also normalise the point gradients. This treatment departs from the CIP literature, although it is consistent with the 'integrated average' type of moment which appears in the CIP–CSL schemes.

$$g := \Delta x \frac{\partial f}{\partial x} = \frac{\partial f}{\partial \xi} \tag{3.51}$$

$F(\xi)$ and $G(\xi)$ may be expressed as sums of powers of ξ,

$$\begin{aligned} F(\xi) &= c_0 + c_1\xi + c_2\xi^2 + c_3\xi^3 \\ &= \begin{pmatrix} 1 & \xi & \xi^2 & \xi^3 \end{pmatrix} \mathbf{c} \end{aligned} \tag{3.52}$$

$$\begin{aligned} G(\xi) &= c_1 + 2c_2\xi + 3c_3\xi^2 \\ &= \begin{pmatrix} 0 & 1 & 2\xi & 3\xi^2 \end{pmatrix} \mathbf{c} \end{aligned} \tag{3.53}$$

in which case the system of equations determining the cubic may be written

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}}_{\mathbf{c}} = \underbrace{\begin{pmatrix} f_i \\ g_i \\ f_{i-1} \\ g_{i-1} \end{pmatrix}}_{\mathbf{u}} \tag{3.54}$$

The Vandermonde matrix $\mathbf{A}$ is made up of the row vectors in Equations (3.52) and (3.53) evaluated at $\xi = 0$ and $\xi = -1$, i.e. grid points i and $i-1$. The coefficients are recovered as $\mathbf{c} = \mathbf{A}^{-1}\mathbf{u}$. When the grid is static, the inversion of $\mathbf{A}$ can be done outside simulation time. With linear advection, we can go further and derive time-invariant sets of weights $\mathbf{w}$ and $\mathbf{w}'$ such that

$$f_i^{n+1} = \mathbf{w}\ \mathbf{u}^n \equiv \begin{pmatrix} w_{f_i} & w_{g_i} & w_{f_{i-1}} & w_{g_{i-1}} \end{pmatrix} \mathbf{u}^n \tag{3.55}$$

$$g_i^{n+1} = \mathbf{w}'\ \mathbf{u}^n \equiv \begin{pmatrix} w'_{f_i} & w'_{g_i} & w'_{f_{i-1}} & w'_{g_{i-1}} \end{pmatrix} \mathbf{u}^n \tag{3.56}$$

Substituting the departure point position $(x_i - a\Delta t)$ into Equation (3.50) means that Equations (3.48) and (3.49) can be rewritten solely in terms of the Courant number:

$$f_i^{n+1} = F^n(-\sigma) = \begin{pmatrix} 1 & -\sigma & \sigma^2 & \sigma^3 \end{pmatrix} \mathbf{c}^n \tag{3.57}$$

$$g_i^{n+1} = G^n(-\sigma) = \begin{pmatrix} 0 & 1 & -2\sigma & 3\sigma^2 \end{pmatrix} \mathbf{c}^n \tag{3.58}$$

Since $\mathbf{c} = \mathbf{A}^{-1}\mathbf{u}$, premultiplying $\mathbf{A}^{-1}$ by the row vectors in (3.57) and (3.58) gives $\mathbf{w}$ and $\mathbf{w}'$ respectively. The weights can be reshaped into square matrices corresponding to relative position and linearly combined to define a nondimensional space-time operator.

$$\mathbf{W}(K) = \begin{pmatrix} w_{f_i} & w_{g_i} \\ w'_{f_i} & w'_{g_i} \end{pmatrix} + \begin{pmatrix} w_{f_{i-1}} & w_{g_{i-1}} \\ w'_{f_{i-1}} & w'_{g_{i-1}} \end{pmatrix} \mathrm{e}^{-\mathrm{I}K} \tag{3.59}$$

It remains to subject $-\mathbf{W}(K)$ to an eigendecomposition in the same fashion as $\mathbf{S}$ in Equation (3.10). There are two eigenvalues for each value of K. In the present context $\mathbf{W}$ corresponds to the fully discrete space, so the resulting spectrum describes $z(K,\sigma)$ rather than $\Omega(K)/\sigma$. Figure 3.2 shows $z(K,\sigma)$ of CIP plotted on the complex plane at various σ. This spectrum of amplification factors also yields the CIP curves previously seen in Figures 2.2 and 2.3 (pp. 25–26), where the dissipation and dispersion errors are given by Equations (3.37) and (3.38), respectively.

It may be noted that the trace of $z(K,\sigma)$ remains inside the unit circle, and hence is stable, as long as $\sigma \leq 1$. This is a remarkable property of the semi-Lagrangian formulation compared to the Eulerian formulation. It will be seen that CIP's Eulerian equivalent, IDO, is unconditionally unstable when coupled with the same time scheme, and at best suffers from a stringent stability criterion when coupled with a high-order time scheme.

In fact, it is possible to have stable Courant numbers of $\sigma > 1$ in the semi-Lagrangian formulation if one interpolates using the profile local to the departure point, instead of extrapolating using the profile immediately upwind of the arrival point. This is common practice when implementing semi-Lagrangian schemes. To keep the stability analysis linear, however, we shall continue to use the immediate upwind profile.

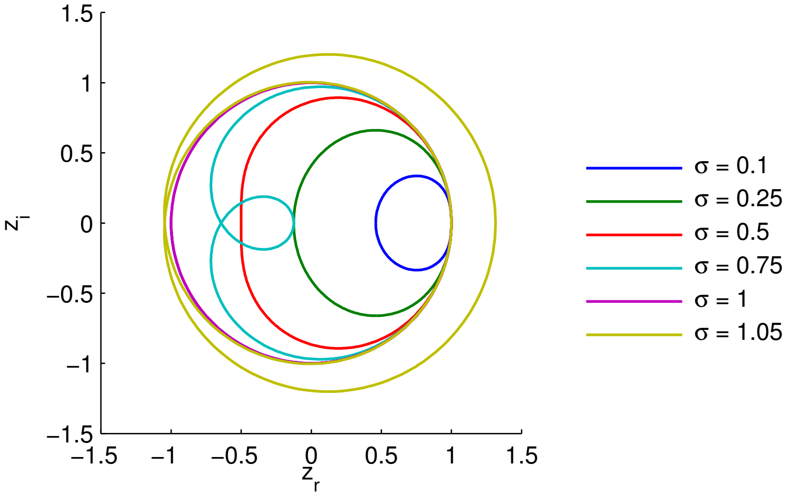


Figure 3.2: CIP amplification factors plotted on the complex z-plane at various Courant numbers

Interpolated Differential Operator Method (IDO)

As before, a cubic function $F(\xi)$ is fitted between each pair of grid points using the point values and gradients. But now the interpolation of $F(\xi)$ at some departure point is replaced by a differentiation at the arrival point. In semi-discrete terms,

$$\frac{\mathrm{d}f}{\mathrm{d}t} = -\frac{a}{\Delta x}\left.\frac{\partial F(\xi)}{\partial \xi}\right|_0 = -\frac{a}{\Delta x}\left.\frac{\partial \mathbf{r}_F}{\partial \xi}\right|_0 \mathbf{c} \tag{3.60}$$

$$\frac{\mathrm{d}g}{\mathrm{d}t} = -\frac{a}{\Delta x}\left.\frac{\partial G(\xi)}{\partial \xi}\right|_0 = -\frac{a}{\Delta x}\left.\frac{\partial \mathbf{r}_G}{\partial \xi}\right|_0 \mathbf{c} \tag{3.61}$$

where $\mathbf{r}_F$ and $\mathbf{r}_G$ are the row vectors in (3.52) and (3.53) respectively;

$$\mathbf{r}_F = \begin{pmatrix} 1 & \xi & \xi^2 & \xi^3 \end{pmatrix} \tag{3.62}$$

$$\mathbf{r}_G = \begin{pmatrix} 0 & 1 & 2\xi & 3\xi^2 \end{pmatrix} \tag{3.63}$$

Equations (3.51) through (3.54) still hold, but this time the weights of interest correspond to the semi-discrete space and are independent of σ. After substituting $\mathbf{c} = \mathbf{A}^{-1}\mathbf{u}$ into (3.60) and (3.61), the weights are identified as

$$\mathbf{w} = \left.\frac{\partial \mathbf{r}_F}{\partial \xi}\right|_0 \mathbf{A}^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \tag{3.64}$$

$$\mathbf{w}' = \left.\frac{\partial \mathbf{r}_G}{\partial \xi}\right|_0 \mathbf{A}^{-1} = \begin{pmatrix} -6 & 4 & 6 & 2 \end{pmatrix} \tag{3.65}$$

Reshaping and combining them according to (3.59) gives

$$\mathbf{W}(K) = \begin{pmatrix} 0 & 1 \\ -6 & 4 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 6 & 2 \end{pmatrix} \mathrm{e}^{-\mathrm{I}K} \tag{3.66}$$

and subjecting $-\mathbf{W}(K)$ to an eigendecomposition yields $\Omega(K)/\sigma$. The spectrum is shown in Figure 3.3.

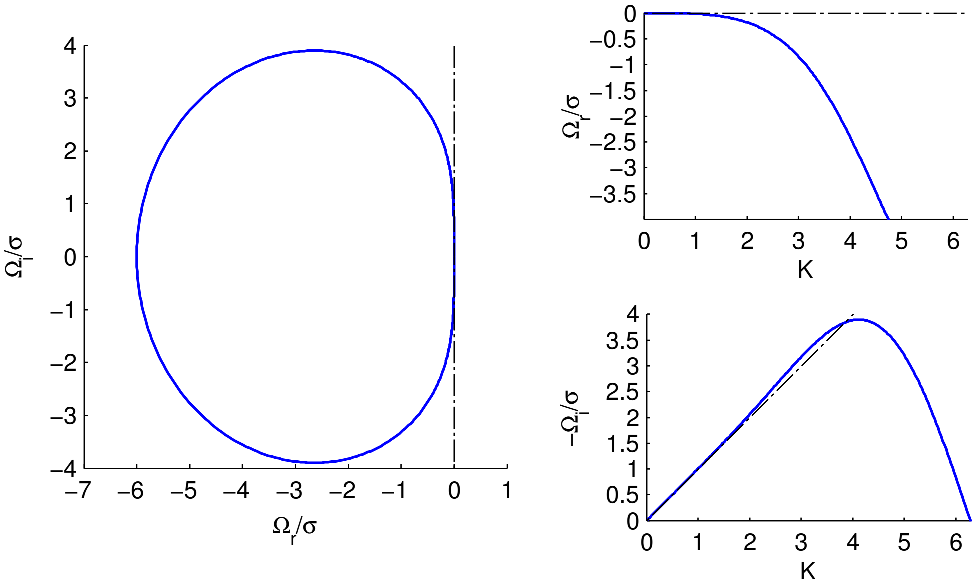


Figure 3.3: Spectrum of the interpolated differential operator method (IDO), with (left) frequencies on the complex plane; (top right) real components with respect to wavenumber, indicating the dissipation rate; and (bottom right) imaginary components with respect to wavenumber, indicating the phase speed. Exact solutions are shown by dash-dotted lines.

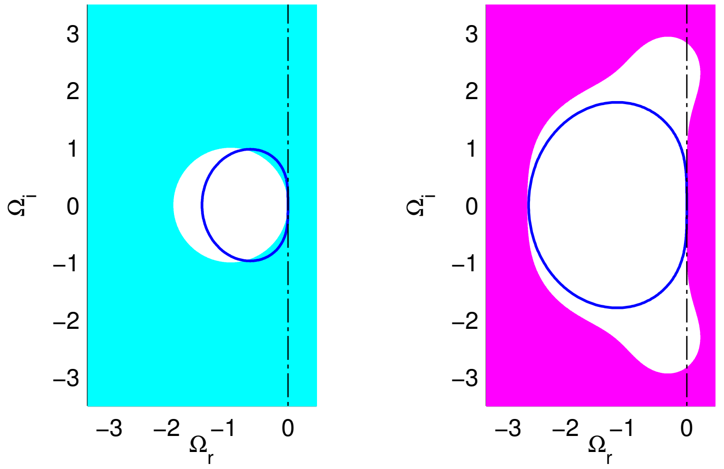
Dissipation is represented in the plot of real components Ω_r/σ against K. When representing dispersion, it is conventional to plot the relation in the positive Ω, positive K quadrant such that the exact relation appears as $\Omega = \sigma K$ (Hu *et al.*, 1999; Yabe *et al.*, 2001b). However, this requires reformulating the problem in terms of the dispersion relation; then dissipation is represented by the imaginary components Ω_i/σ and dispersion by Ω_r/σ. In contrast, the present analysis concerns the spectrum of the space operator matrix, and the exact solution appears as $\Omega = -\mathrm{I}\sigma K$. It is therefore convenient to simply plot $-\,\Omega_i/\sigma$ against K to represent the dispersion relation.

Another important detail is that, because the $-\mathbf{W}(K)$ matrix is of rank 2, two eigenvalues are produced for a given K. The physical mode is identified as the mode approaching the exact solution as $K \to 0$. Other modes may be referred to as spurious or parasite modes (Hu *et al.*), but often the eigenvalues join together to form a single loop as exemplified in Figure 3.3. Here one eigenvalue begins at the origin, the other at $(-6,0)$, and both eigenvalues proceed clockwise on the complex plane as K is increased. In this case it is clear that the modes are one and the same; the idea of a spurious mode is not meaningful. In the present work, the term ‘spurious’ will be reserved for a mode which is clearly decoupled from the physical mode to form a separate loop on the complex plane. The modes can be identified by cycling over a large enough range of K for all the loops to close. The existence of any positive real parts will be uncovered this way. However, to avoid cluttering plots in the forthcoming chapters, spurious modes will be omitted unless specified otherwise.

The amplification factors of the fully discrete system are given by passing $\Omega(K)$ to the z function of some time scheme. The IDO curves in Figure 2.3, for example, result from a coupling to the fourth order Runge–Kutta (RK4) scheme. In general, the z function for an qth order RK scheme reads

$$z(\Omega) = \sum_{i=0}^{q} \frac{\Omega^i}{i!} \tag{3.67}$$

Unlike for CIP, coupling to the forward Euler scheme is out of the question for IDO. The curvature of IDO’s spectrum at the imaginary axis on the complex Ω-plane is such that the spectrum never fits inside the forward Euler stability region, as shown in Figure 3.4a. A high-order time scheme such as RK4 will work as long as the IDO spectrum is scaled by an appropriate Courant number, as shown in Figure 3.4b.



(a) at $\sigma = 0.25$ with forward Euler (unstable)

(b) at $\sigma = 0.46$ with fourth order Runge–Kutta (stable)

Figure 3.4: IDO spectra and stability regions of candidate time schemes.

The reader may observe that the RK4 stability region reaches into the positive real half of the complex plane, apparently overcoming our previously stated criteria that the real parts of all eigenvalues must be zero or negative to ensure stability. While it is true that a spectrum could conceivably have positive real parts and fit inside such a stability region, in practice this situation could not be maintained as the Courant number is made smaller. Therefore we still insist on zero or negative real parts for Eulerian formulations.

IDO Finite Volume Method (IDO–FVM)

CIP is a nonconservative method, but with modification it can be made to guarantee exact conservation. Non-overlapping cells are defined between the grid points, each of which carries a cell-integrated average. The arrangement is shown in Figure 3.5. Like the point moments, the cell-integrated average is included in the set of constraints determining the interpolation profile. It is updated by computing the swept volumes through the cell boundaries via an integration of the interpolation profile between arrival and departure points. The flux through each boundary sees one cell gain what the other has lost: exact conservation is maintained by construction. Hence the term ‘CIP conservative semi-Lagrangian’ (CIP–CSL). We do not go into the analysis of CIP–CSL here, but we do convert the third order scheme, CIP–CSL2, into a purely Eulerian form in order to analyse the equivalent semi-discrete system. The ‘2’ suffix refers to the degree of polynomial involved, not the order of accuracy. The resulting scheme is referred to as IDO–FVM2.

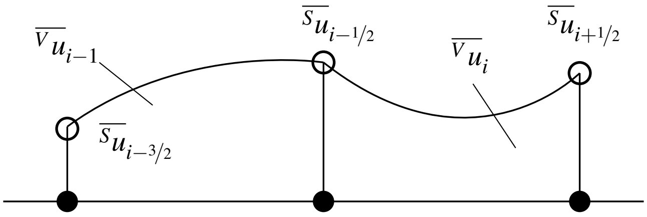

$\overline{^{V}u_{i-1}}$
$\overline{^{S}u_{i-1/2}}$
$\overline{^{V}u_{i}}$
$\overline{^{S}u_{i+1/2}}$
$\overline{^{S}u_{i-3/2}}$

Figure 3.5: The 1D arrangement for the third order CIP conservative semi-Lagrangian scheme (CIP–CSL2) and its Eulerian equivalent, the interpolated differential operator finite volume method (IDO–FVM2). Superscript V refers to a volume- or cell-integrated average, which in 1D reduces to a line-integrated average, and S refers to a surface-integrated average, which in 1D reduces to a point value.

The finite volume method requires a conservation law to be cast in integral form,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V u\,\mathrm{d}V + \oint_S \mathbf{F}\cdot\mathbf{dS} = \int_V q\,\mathrm{d}V \tag{3.68}$$

where V represents the control volume, u the prognostic variable, $\mathbf{S}$ the directed surface area, $\mathbf{F}$ the vector field of fluxes, and q additional source terms. The fluxes and source terms in the case of the advection equation are $\mathbf{F} = \mathbf{a}u$ and $q = 0$. This leads to the following semi-discretisation in 1D:

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{^{V}u_i}\Delta x + a\left(u^*_{i+1/2} - u^*_{i-1/2}\right) = 0 \tag{3.69}$$

where $\overline{{}^{V}u}_{i}$ now represents the integrated average of cell i, and $u^{*}_{i+1/2}$ and $u^{*}_{i-1/2}$ represent numerical approximations of the quantity at the cell boundaries. The latter could be interpolated from an upwind-biased profile constrained according to local cell values. For example, it is clear from (3.69) that using a degree-zero polynomial, i.e. replacing $u^{*}_{i+1/2+j}$ with $\overline{{}^{V}u}_{i+j}$, simply recovers the first order upwind scheme. A degree-p polynomial results in a $(p+1)$th order scheme. There is a subtle distinction between this approach and the finite difference approach outlined previously, where a degree-p polynomial was *differentiated* to result in a pth order scheme.

Building in the multi-moment method, it is convenient to store values at the cell boundaries and have them readily available for use in Equation (3.69). Let the boundary values be stored as $\overline{{}^{S}u}_{i+1/2}$, etc. For each cell, a quadratic profile can be fitted according to the three local moments. The matrix system reads

$$\underset{\mathbf{A}}{\begin{pmatrix} 1 & 0 & 0 \\ 1 & -1/2 & 1/3 \\ 1 & -1 & 1 \end{pmatrix}} \underset{\mathbf{c}}{\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}} = \underset{\mathbf{u}}{\begin{pmatrix} \overline{{}^{S}u}_{i+1/2} \\ \overline{{}^{V}u}_{i} \\ \overline{{}^{S}u}_{i-1/2} \end{pmatrix}} \tag{3.70}$$

where the middle row of $\mathbf{A}$ derives from an integration of the quadratic $U(\xi)$ between -1 and 0. As with any other moment, weights for the cell-integrated average can be recovered by expressing the evolution equation in terms of $-a/\Delta x$, some nondimensional row vector $\mathbf{r}$, and $\mathbf{c}$, before multiplying $\mathbf{r}$ by $\mathbf{A}^{-1}$. In this instance,

$$\begin{aligned} \frac{\mathrm{d}\overline{{}^{V}u}_{i}}{\mathrm{d}t} &= -\frac{a}{\Delta x}\left[\ U(0) \ - \ U(-1)\ \right] \\ &= -\frac{a}{\Delta x}\begin{pmatrix} 0 & 1 & -1 \end{pmatrix}\mathbf{c} \end{aligned} \tag{3.71}$$

The boundary values are updated by IDO as before. Let the weights for the boundary values and the cell-integrated average be ${}^{V}\mathbf{w} \equiv \begin{pmatrix} {}^{V}w_{S_i} & {}^{V}w_{V_i} & {}^{V}w_{S_{i-1}} \end{pmatrix}$ and ${}^{S}\mathbf{w} \equiv \begin{pmatrix} {}^{S}w_{S_i} & {}^{S}w_{V_i} & {}^{S}w_{S_{i-1}} \end{pmatrix}$, respectively. Then

$${}^{V}\mathbf{w} = \begin{pmatrix} 0 & 1 & -1 \end{pmatrix}\mathbf{A}^{-1} = \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} \tag{3.72}$$

$${}^{S}\mathbf{w} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}\mathbf{A}^{-1} = \begin{pmatrix} 4 & -6 & 2 \end{pmatrix} \tag{3.73}$$

The elements of ${}^{V}\mathbf{w}$ are more easily found from inspection of (3.69) when $\overline{{}^{S}u}_{i+1/2}$ and $\overline{{}^{S}u}_{i-1/2}$ have been substituted. But it is still important to show how the cell integrated average fits into the analytical method like the other moments, because later on it will become necessary to analyse multidimensional systems where the arrangements of moments are more complex.

Reshaping the weights this time requires padding the square matrices with zero ${}^{V}w_{V_{i-1}}$ and ${}^{S}w_{V_{i-1}}$ terms.

$$\mathbf{W}(K) = \begin{pmatrix} 0 & 1 \\ -6 & 4 \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 0 & 2 \end{pmatrix} \mathrm{e}^{-\mathrm{I}K} \tag{3.74}$$

After eigendecomposing, this operator yields the same spectrum as IDO. It confirms that the CIP-CSL2 scheme, with its quadratic profile reconstruction, recovers the same third order accuracy as the cubic-based CIP. As mentioned in §2.2.1, at first it may appear as though an extra order of accuracy has been earned for free. But there are no savings in terms of memory, because in either case two moments must be stored per repeating grid unit.

Higher-order and multidimensional variations of IDO and IDO–FVM will be explored more fully in the following chapters.

Multi-Moment Constrained Finite Volume Method (MCV)

A quadratic profile is also used in the third order MCV scheme (MCV3), but it is discontinuous between cells as shown in Figure 3.6. Point values are duplicated at the cell boundaries, with values on the upwind side denoted $\overline{{}^{S-}u}$ and on the downwind side $\overline{{}^{S+}u}$. In the literature, $\overline{{}^{V}u}$ is further transformed into a midpoint value for practical purposes. This transformation does not affect the modal decomposition and is omitted here for brevity.

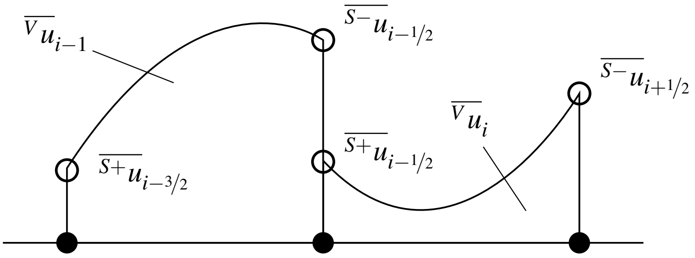

$\overline{{}^{V}u}_{i-1}$
$\overline{{}^{S-}u}_{i-1/2}$
$\overline{{}^{S-}u}_{i+1/2}$
$\overline{{}^{V}u}_{i}$
$\overline{{}^{S+}u}_{i-1/2}$
$\overline{{}^{S+}u}_{i-3/2}$

Figure 3.6: The third order multi-moment constrained finite volume method (MCV3). Superscript V refers to the volume-integrated average, which in 1D reduces to a line-integrated average, and $S-$ and $S+$ refer respectively to the upwind-side and downwind-side surface-integrated averages, which in 1D reduce to point values.

With the normalised position ξ local to each cell, the matrix system reads

$$\underset{\mathbf{A}}{\begin{pmatrix} 1 & 0 & 0 & & & \\ 1 & -1/2 & 1/3 & & & \\ 1 & -1 & 1 & & & \\ & & & 1 & 0 & 0 \\ & & & 1 & -1/2 & 1/3 \\ & & & 1 & -1 & 1 \end{pmatrix}} \underset{\mathbf{c}}{\begin{pmatrix} c_{i,0} \\ c_{i,1} \\ c_{i,2} \\ c_{i-1,0} \\ c_{i-1,1} \\ c_{i-1,2} \end{pmatrix}} = \underset{\mathbf{u}}{\begin{pmatrix} \overline{{}^{S-}u}_{i+1/2} \\ \overline{{}^{V}u}_{i} \\ \overline{{}^{S+}u}_{i-1/2} \\ \overline{{}^{S-}u}_{i-1/2} \\ \overline{{}^{V}u}_{i-1} \\ \overline{{}^{S+}u}_{i-3/2} \end{pmatrix}} \tag{3.75}$$

Using an upwind flux at the cell boundaries, the weights are computed as

$$^{V}\mathbf{w} = \begin{pmatrix} \mathbf{r}|_0 & -\mathbf{r}|_0 \end{pmatrix} \mathbf{A}^{-1} = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \end{pmatrix} \tag{3.76}$$

$$^{S-}\mathbf{w} = \begin{pmatrix} \left.\frac{\partial \mathbf{r}}{\partial \xi}\right|_0 & \mathbf{0} \end{pmatrix} \mathbf{A}^{-1} = \begin{pmatrix} 4 & -6 & 2 & 0 & 0 & 0 \end{pmatrix} \tag{3.77}$$

$$^{S+}\mathbf{w} = \begin{pmatrix} \mathbf{0} & \left.\frac{\partial \mathbf{r}}{\partial \xi}\right|_0 \end{pmatrix} \mathbf{A}^{-1} = \begin{pmatrix} 0 & 0 & 0 & 4 & -6 & 2 \end{pmatrix} \tag{3.78}$$

and reshaped as

$$\mathbf{W}(K) = \begin{pmatrix} 0 & 1 & 0 \\ -6 & 4 & 2 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ -6 & 4 & 2 \end{pmatrix} \mathrm{e}^{-\mathrm{I}K} \tag{3.79}$$

Once again, after eigendecomposing, this operator yields the same spectrum as IDO. It verifies the resolution estimated previously in Table 2.1. It is intuitive that we should not gain any accuracy from storing the additional point moment in this case, because we effectively discard downwind information when using an upwind flux. The redundancy is evidenced from the zero rows in Equation (3.79). However, as pointed out in the literature review, we can expect MCV to be better at resolving nonlinear features such as shocks when modelling compressible flow.

3.3.2 Performance in the Time Domain

As a pilot experiment, one-dimensional linear advection was simulated using both conventional and multi-moment third order space discretisations in Eulerian and semi-Lagrangian formulations. In other words, the four schemes featured in Figure 2.3 (TOU, CUL, IDO and CIP) were tested in the time domain. However, RK4 integration was replaced with the more economical RK3. The purpose of the experiment was to uncover the behaviour of errors when both Δx and Δt were varied.

Method

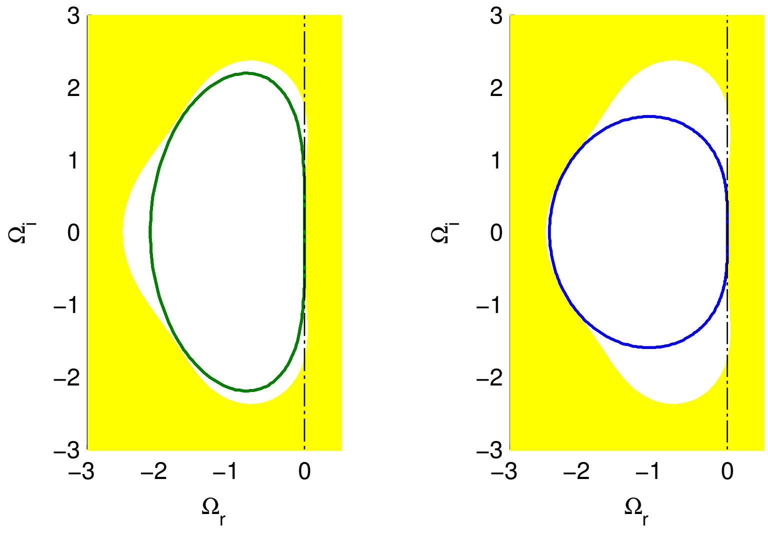
31 grids were tested with the number of grid points m ranging exponentially from 10 to 10,000. For each grid, 31 simulations were carried out with the number of time steps n ranging exponentially from 100 to 100,000. All other parameters were kept constant as follows:

- Initial condition $u_0(x) = \sin x$
- Advection velocity $a = 1$
- Domain length $x_m = 2\pi$
- Simulated time $t_n = 2\pi$
- Periodic boundary conditions

Because the velocity field is uniform and constant, the integration of the departure point in the case of the semi-Lagrangian schemes is given exactly by $x_i - u\Delta t$. In other words, the forward Euler time scheme can be used. But in the case of the Eulerian schemes, a high-order time scheme is mandatory for stability. Analysis suggests that the conventional third order upwind scheme holds an advantage over IDO in this respect: the stability criteria is $\sigma < 1.6$ compared to $\sigma < 0.4$, as shown in Figure 3.7.

With Eulerian schemes a new flow field must be computed at each RK stage, rendering them more expensive than their semi-Lagrangian counterparts. An economical RK3 algorithm (Shu and Osher, 1988) was selected. This algorithm has the further benefit of being total variation diminishing if a flux limiter is applied at each stage. It reads

$$\begin{aligned} u^{\langle 1\rangle} &= u^n + \Delta t \mathcal{L}\left(u^n, t_n\right) \\ u^{\langle 2\rangle} &= \frac{3}{4}u^n + \frac{1}{4}u^{\langle 1\rangle} + \frac{1}{4}\Delta t \mathcal{L}\left(u^{\langle 1\rangle}, t_{n+1}\right) \\ u^{n+1} = u^{\langle 3\rangle} &= \frac{1}{3}u^n + \frac{2}{3}u^{\langle 2\rangle} + \frac{2}{3}\Delta t \mathcal{L}\left(u^{\langle 2\rangle}, t_{n+1/2}\right) \end{aligned} \tag{3.80}$$



(a) TOU at $\sigma = 1.6$ **(b)** IDO at $\sigma = 0.4$

Figure 3.7: Third order spectra and stability region of third order Runge-Kutta time integration.

Results

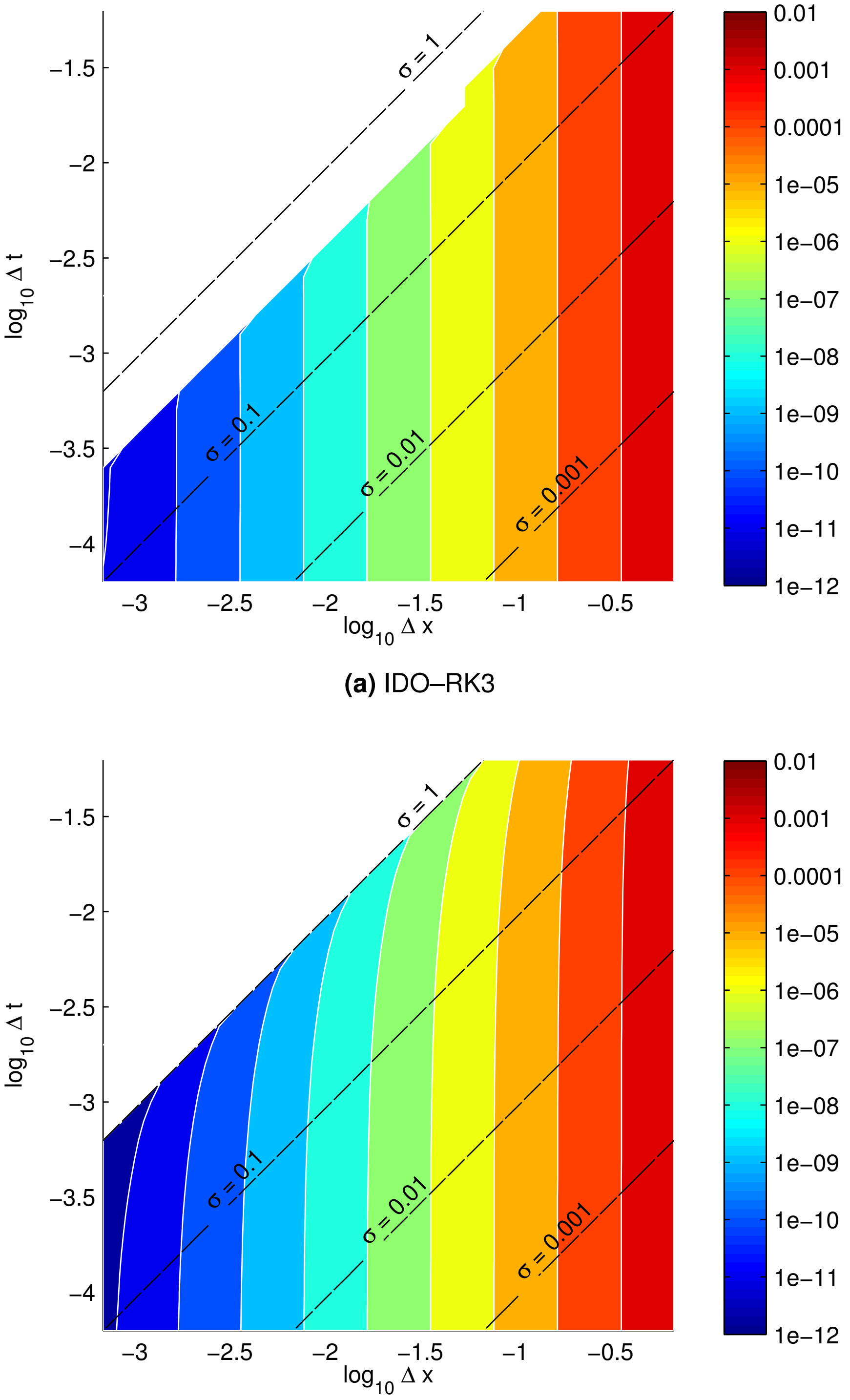
After time t_n, the initial profile should have travelled exactly one wavelength. $||\varepsilon||_1$ was computed from the numerical solution u_{N_i} and exact solution $u_{E_i} = \sin x_i$ according to Equation (3.39). It is shown plotted as filled contours with respect to Δx and Δt in Figure 3.8. The multi-moment schemes have the same shape of contours as their noncompact counterparts, and so for brevity the plots for the noncompact schemes have been omitted here.

Because of the similarity between error means of the the multi-moment schemes and noncompact schemes, the ratio between the two sets could be computed. Plots of this ratio are shown in Figure 3.9. The range narrows as the Courant number decreases, making it possible to quantify the improvement of the multi-moment schemes over their noncompact counterparts. In both formulations the ratio converges to 6.06 ± 0.04 as $\sigma \to 0$.

Discussion

In the Eulerian formulation, the vertical $||\varepsilon||_1$ contours show that convergence is purely spatial in the stable range of Courant numbers. That is, the Δx^p term in Equation (2.3) dominates. It can be confirmed that the contours have near-uniform gradient of three. There is a slight deterioration in the gradient at small Δx and Δt which could be attributed to machine rounding errors.

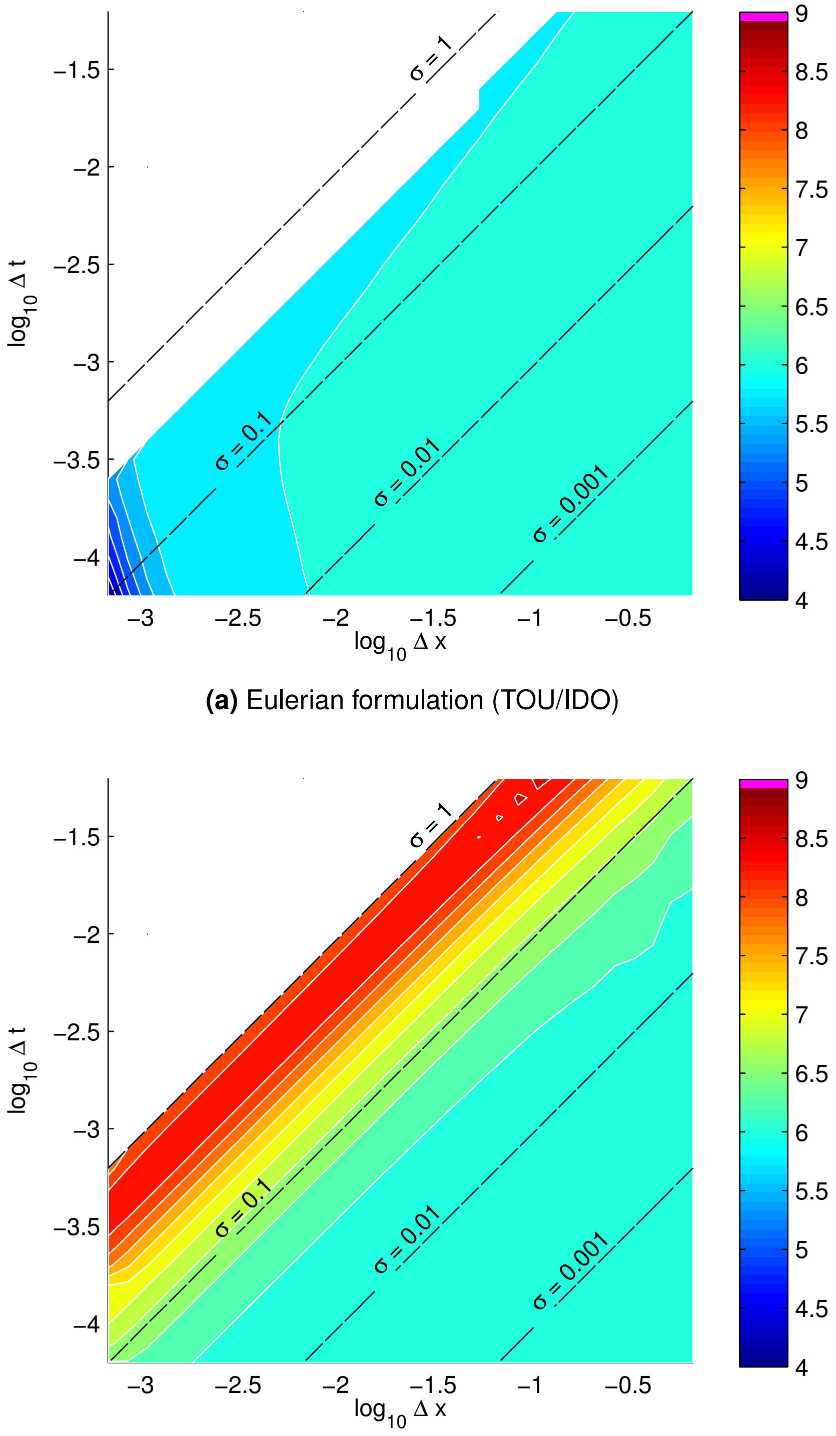
In the semi-Lagrangian formulation, if the $\Delta x^{p+1} \big/ \Delta t$ term from Equation (2.2) were to dominate, the contours would be angled at $\tan^{-1} 1/4 \approx 14°$ to the vertical. Clearly this does not happen, so perhaps linear advection is a special case that renders (2.2) inapplicable. However,



(a) IDO–RK3

(b) CIP

Figure 3.8: $\|\varepsilon\|_1$ for multi-moment schemes simulating 1D linear advection



(a) Eulerian formulation (TOU/IDO)

(b) Semi-Lagrangian formulation (CUL/CIP)

Figure 3.9: Ratio of $\|\varepsilon\|_1$ between noncompact and multi-moment schemes.

it may be observed that the curved convergence contours are self-similar along σ-contours and approximately vertical at small Courant numbers. It suggests that Equation (3.40) may still hold for semi-Lagrangian schemes as long as the Courant number is either held constant or sufficiently small. Indeed, when (3.40) is applied to the errors in Figure 3.8b with σ held constant instead of Δt, uniform third order convergence is recovered. This is encouraging because it proves that semi-Lagrangian simulations may still converge according to their nominal order of spatial accuracy.

Finally, we can compare the ratio of errors against the predicted ratio using physical-mode error norms computed from the TOU and IDO spectra. Table 3.1 shows the errors with decreasing wavenumber (increasing grid resolution). Both sets of errors converge precisely on factor six. This agrees with all Eulerian results, except at high grid numbers (which can be attributed to machine rounding errors), and with semi-Lagrangian results except at high Courant numbers.

Table 3.1: Physical-mode error norms, defined by Equation (3.47), for the third order upwind (TOU) and interpolated differential operator (IDO) schemes.

$\log_{10}(K/2\pi)$	TOU	IDO	ratio
-1.0	1.1817e−01	2.0837e−02	5.671261
-1.5	4.0848e−03	6.8227e−04	5.987053
-2.0	1.2983e−04	2.1640e−05	5.999411
-2.5	4.1070e−06	6.8450e−07	5.999963
-3.0	1.2988e−07	2.1646e−08	6.000001

Conclusion

The above experiment demonstrates that we can test multi-moment schemes in the time domain, using either an Eulerian or semi-Lagrangian formulation, and be confident that the temporal error will not pollute the observed spatial convergence. The proviso in the case of the semi-Lagrangian formulation is that the Courant number is kept small or held constant (i.e. spatial convergence is calculated after a combined grid and time step refinement). A small Courant number will also see the semi-Lagrangian scheme approximating its Eulerian counterpart.

For the purpose of verifying an analysis it is convenient to opt for the semi-Lagrangian formulation, even when the system under analysis is semi-discrete and therefore technically Eulerian. The semi-Lagrangian formulation permits coupling with a low-order time scheme and is proportionally cheaper to run than the Eulerian formulation with a high-order time scheme. And, because it affords greater stability, it enables fewer time steps to be carried out for the same grid size. The semi-Lagrangian formulation shall henceforth be used in all time-domain tests.

When simulating linear advection, the physical-mode error norms computed from the spectra may provide further verification when a ratio of errors between same-order schemes is taken. This ratio is insensitive to the choice of grid and time step for most grid resolutions in the Eulerian formulation, but it is only valid for small Courant numbers in the semi-Lagrangian formulation.

3.4 Two Spatial Dimensions

In 2D, the linear advection equation describes a plane wave travelling in space at some angle θ. When propagating in the x-direction ($\theta = 0°$) or the y-direction ($\theta = 90°$) we can expect the numerical approximation of the wave computed by the 2D scheme to be exactly the same as that computed by the equivalent 1D scheme. But at oblique angles, the shape of the grid elements causes the approximation to either improve or deteriorate. The approximation is said to be anisotropic. Anisotropy is usually impossible to avoid, but at minimum we insist on the space scheme maintaining its nominal order of accuracy in all directions.

In addition to studying anisotropy arising from the discretised 2D linear advection equation, this section introduces additional test cases for the time domain that can offer further insights into space scheme properties.

3.4.1 Semi-Discrete Analysis

The linear advection equation is expanded as

$$\begin{aligned}\frac{\partial u}{\partial t} &= -\mathbf{a}\cdot\nabla u \\ &= -a_x\frac{\partial u}{\partial x} - a_y\frac{\partial u}{\partial y}\end{aligned} \tag{3.81}$$

As with the 1D case (3.18–3.19), the semi-discrete equation and associated space operator can be expressed in terms of weights corresponding to moments in the stencil. But Equation (3.81) implies that these weights are now split into x- and y-components. The semi-discrete equation reads

$$\frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} = -\left(\frac{a_x}{\Delta x}\mathbf{w}^x + \frac{a_y}{\Delta y}\mathbf{w}^y\right)\mathbf{u} \tag{3.82}$$

where each element in $\mathbf{w}^x$ and $\mathbf{w}^y$ corresponds to a relative point position (α,β). Now let $K := k_x\Delta x$; let $\tan\theta := a_y/a_x = k_y/k_x$; and define a grid spacing aspect ratio $\gamma := \Delta y/\Delta x$. The space operator can be expressed as a function of these variables and the weights,

$$S(K,\theta,\gamma) = -\frac{a\cos\theta}{\Delta x}\sum_{\alpha,\beta} w^x_{\alpha,\beta}\mathrm{e}^{\mathrm{I}K(\alpha\cos\theta+\beta\gamma\sin\theta)} - \frac{a\sin\theta}{\gamma\Delta x}\sum_{\alpha,\beta} w^y_{\alpha,\beta}\mathrm{e}^{\mathrm{I}K(\alpha\cos\theta+\beta\gamma\sin\theta)} \tag{3.83}$$

leading to the expression for the nondimensional spectrum

$$\frac{\Omega(K,\theta,\gamma)}{\sigma} = -\cos\theta \sum_{\alpha,\beta} w^x_{\alpha,\beta} \mathrm{e}^{\mathrm{I}K(\alpha\cos\theta+\beta\gamma\sin\theta)} - \frac{1}{\gamma}\sin\theta \sum_{\alpha,\beta} w^y_{\alpha,\beta} \mathrm{e}^{\mathrm{I}K(\alpha\cos\theta+\beta\gamma\sin\theta)} \tag{3.84}$$

Here the Courant number $\sigma := a\Delta t / \Delta x$ remains a function of velocity magnitude and Cartesian grid spacing. One might argue that a different definition of the Courant number should be used in multidimensions, but other definitions are sensitive to θ and γ and are therefore not appropriate for normalising the spectrum.

The following two subsections show how the two sets of weights may be computed for split schemes and unsplit schemes. We shall henceforth assume $\gamma = 1$ to reduce the number of variables to take into account.

Split Approach

As explained in §2.2.2, a multidimensional space scheme may be decomposed into a set of 1D operators applied in sequence. Thus, Equation (3.81) may be treated as

$$\frac{\partial u^*}{\partial t} = -a\cos\theta \frac{\partial u}{\partial x} \tag{3.85a}$$

$$\frac{\partial u}{\partial t} = -a\sin\theta \frac{\partial u^*}{\partial y} \tag{3.85b}$$

The above equations correspond to the fully discrete space (the use of continuous operators is expedient to avoid cluttering the equations with discrete operators). Each equation is discretised according to the selected time and space scheme, so integration in time can be done step by step. But from a semi-discrete point of view, the integration has not happened yet: u^* and u are not coupled quantities; they are one and the same. For the purpose of analysis we can continue to use Equation (3.81) which simply adds together the space differentials.

Returning to the example of the first order upwind scheme, our aim is to compute the weights in Eqs. (3.82–3.84). It can be inferred from the 1D FOU formulation (3.16) that

$$w^x_{00} = 1; \quad w^x_{\bar{1}0} = -1; \quad w^x_{0\bar{1}} = 0 \tag{3.86a}$$

$$w^y_{00} = 1; \quad w^y_{\bar{1}0} = 0; \quad w^y_{0\bar{1}} = -1 \tag{3.86b}$$

Here the dummy terms $w^x_{0\bar{1}}$ and $w^y_{\bar{1}0}$ have been inserted to acknowledge the fact that the effective 2D stencil encompasses three points.

Unsplit Approach

In the direct or unsplit approach, a 2D profile $U(x,y)$ is constructed over the stencil of points and differentiated with respect to x and y at the point of interest. As previously noted, it helps to nondimensionalise coordinates. Therefore $U(x,y)$ is rewritten $U(\xi,\eta)$, where

$$\xi := \frac{x - x_{i,j}}{\Delta x} \qquad (3.87a) \qquad\qquad \eta := \frac{y - y_{i,j}}{\Delta y} \qquad (3.87b)$$

We consider a polynomial profile, which is a sum of monomials $c_{p,q}\xi^p\eta^q$. Putting the coefficients $c_{p,q}$ into the column vector $\mathbf{c}$ and the $\xi^p\eta^q$ terms into the row vector $\mathbf{r}(\xi,\eta)$ enables the profile to be expressed as

$$U(\xi,\eta) = \mathbf{r}(\xi,\eta)\,\mathbf{c} \qquad (3.88)$$

The semi-discrete equation then evaluates to

$$\begin{aligned}\frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} &= -a_x \left.\frac{\partial U}{\partial x}\right|_{x_i,y_j} - a_y \left.\frac{\partial U}{\partial y}\right|_{x_i,y_j} \\ &= -\left(\frac{a_x}{\Delta x}\left.\frac{\partial \mathbf{r}}{\partial \xi}\right|_{0,0} - \frac{a_y}{\Delta y}\left.\frac{\partial \mathbf{r}}{\partial \eta}\right|_{0,0}\right)\mathbf{c}\end{aligned} \qquad (3.89)$$

The profile is constrained by a Vandermonde matrix $\mathbf{A}$ and column vector of values $\mathbf{u}$, such that $\mathbf{A}\,\mathbf{c} = \mathbf{u}$. The rows of the matrix are simply $\mathbf{r}(\xi,\eta)$ evaluated at each collocation point. The profile coefficients are given by $\mathbf{c} = \mathbf{A}^{-1}\mathbf{u}$. However, it is the weights in Equation (3.82) that are of interest. After substituting $\mathbf{c} = \mathbf{A}^{-1}\mathbf{u}$ into (3.89), the weights are identified as

$$\mathbf{w}^x = \left.\frac{\partial \mathbf{r}}{\partial \xi}\right|_{0,0}\mathbf{A}^{-1} \qquad (3.90a) \qquad\qquad \mathbf{w}^y = \left.\frac{\partial \mathbf{r}}{\partial \eta}\right|_{0,0}\mathbf{A}^{-1} \qquad (3.90b)$$

The profile-based approach presented above is consistent with 1D analysis. For example, the 1D FOU operator effectively constructs a linear profile between points i and $i-1$. In 2D, we might select the tensor product-based profile

$$U(\xi,\eta) = c_{00} + c_{10}\xi + c_{01}\eta + c_{11}\xi\eta = \begin{pmatrix} 1 & \xi & \eta & \xi\eta \end{pmatrix}\mathbf{c} \qquad (3.91)$$

The 2D profile is constrained by the point values u_{00}, $u_{\bar{1}0}$, $u_{0\bar{1}}$ and $u_{\bar{1}\bar{1}}$. Note that the subscripts on the point values refer to the corresponding relative positions ξ and η, whereas on the coefficients they refer to powers of ξ and η on the corresponding monomials. The system of constraints may be written in matrix form as

$$\underset{\mathbf{A}}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & -1 \end{pmatrix}} \underset{\mathbf{c}}{\begin{pmatrix} c_{00} \\ c_{10} \\ c_{01} \\ c_{11} \end{pmatrix}} = \underset{\mathbf{u}}{\begin{pmatrix} u_{00} \\ u_{\bar{1}0} \\ u_{0\bar{1}} \\ u_{\bar{1}\bar{1}} \end{pmatrix}} \tag{3.92}$$

It so happens that this particular Vandermonde matrix inverts to itself. Eqs. (3.90a) and (3.90b) therefore evaluate to

$$\mathbf{w}^x = \begin{pmatrix} 1 & -1 & 0 & 0 \end{pmatrix} \tag{3.93a}$$

$$\mathbf{w}^y = \begin{pmatrix} 1 & 0 & -1 & 0 \end{pmatrix} \tag{3.93b}$$

which are the same weights as those obtained from the split approach (3.86). This is a trivial example, but it demonstrates the identity between the split scheme with uniform operators and the unsplit scheme with the tensor-product basis, when the time stepping is made vanishingly small. For the first order upwind scheme, the four-point stencil of the tensor product basis further reduces to the three-point stencil of the order-complete basis.

Interpreting the Results

In 1D, the analysis of a scheme produces a spectrum of frequencies as a function of wavenumber, $\Omega(K)$. The real and imaginary components can be plotted on the complex plane to indicate the scheme's stability and compatibility with time schemes, or they may be plotted with respect to wavenumber to demonstrate the dissipation and dispersion characteristics. A high-resolution scheme is characterised by the frequencies not deviating from the exact dispersion relation $-\mathrm{I}K$ for a wide range of K.

In 2D, the frequencies are also a function of the propagation angle θ (assuming a constant grid aspect ratio $\gamma = 1$). To visualise the spectra, we may either choose a few values of θ and plot with respect to K, or choose a few values of K and plot with respect to θ. The former is more useful for checking stability, as the frequencies could stray into the positive real region for just a small wavenumber range and render the scheme unconditionally unstable. Also, being able to vary K enables us to check the nominal order of accuracy is being maintained.

We choose to evaluate spectra at three propagation angles: 0°, $\tan^{-1} 0.5 \approx 26.6^\circ$, and 45°. The significance of the middle value is that, as for 0° and 45°, a ray can be traced exactly between grid points as shown in Figure 3.10. The resulting trace of the frequencies on the complex plane forms a closed loop. Specifically, if the spectrum associated with 0° closes at $K = K_{\max}$, then the spectra associated with $\tan^{-1} 0.5$ and 45° are guaranteed to close at $\sqrt{5}K_{\max}$ and $\sqrt{2}K_{\max}$, respectively. The relative lengths of the arrows in Figure 3.10 illustrate this point. By contrast, an angle such as 22.5° would cause the frequencies to orbit inside some envelope but never close, resulting in a messy, hard-to-read plot on the complex plane.

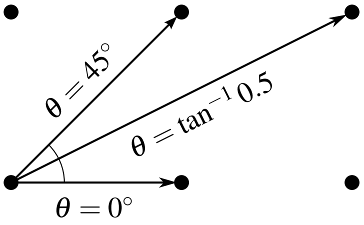


Figure 3.10: Three directions of propagation for which spectra shall be evaluated against K.

For example, plots for the first order upwind scheme are shown in Figure 3.11. It can be confirmed that, for the oblique angles, FOU retains first order accuracy in dissipation and second order accuracy in dispersion, and that the real components of its frequencies are always positive.

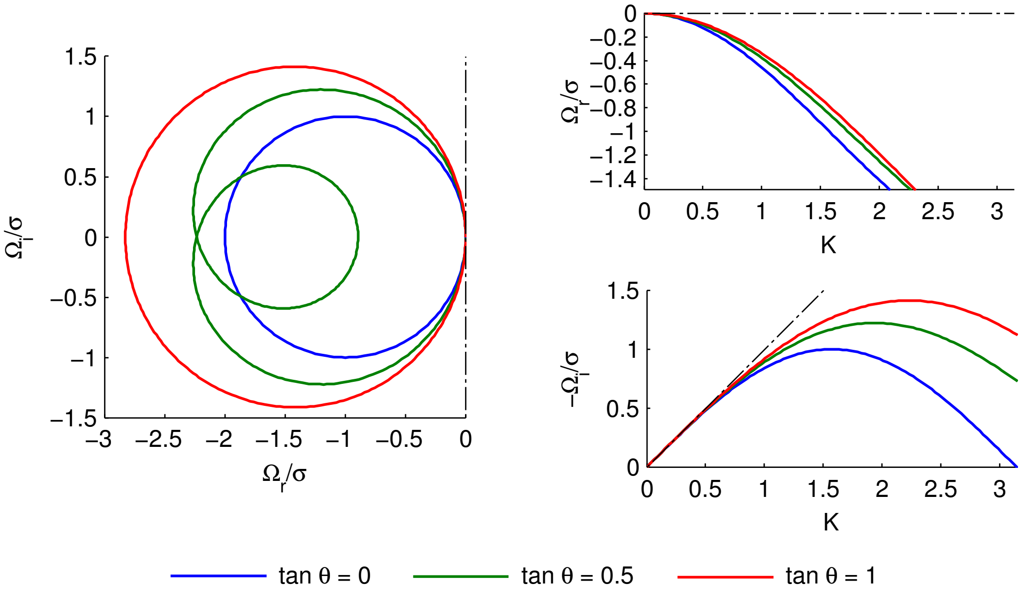


Figure 3.11: Spectra of the first order upwind scheme at three propagation angles. (Left) frequencies on the complex plane; (top right) real components with respect to wavenumber, indicating the dissipation rate; (bottom right) imaginary components with respect to wavenumber, indicating the phase speed. Exact solutions are shown by dash-dotted lines.

The other approach, choosing a few values of K and varying θ, is useful for studying anisotropy. It is common to visualise dissipation and dispersion by polar plots of normalised phase speed $-\,\Omega_i(\theta)\big/\sigma K$ and dissipation factor $e^{\Omega_r(\theta)/\sigma}$ (see e.g. Li, 1997; Hu *et al.*, 1999). The dissipation factor is equivalent to numerical gain when the time integration is exact. In the case of high-order schemes, it may be adjusted to represent dissipation over m cells, $e^{m\Omega_r(\theta)/\sigma}$. Both normalised phase speed and dissipation factor are idealised by a circular contour of unit radius. As K increases, the contour may grow or shrink according to the numerical approximation. A perfectly isotropic scheme would maintain the circular shape, but in practice the contour becomes distorted, indicating anisotropy.

Figure 3.12 demonstrates the effect for FOU. It can be seen that both dissipation and dispersion accuracies improve towards 45°, although the latter eventually decays at high wavenumbers. On another note, it so happens that FOU shares exactly the same dispersion properties as the second order central (SOC) scheme (c.f. Lele, 1992). This is not surprising given the second order dispersion accuracy in both cases. However, SOC differs by having no numerical damping: its dissipation factor is always 1.

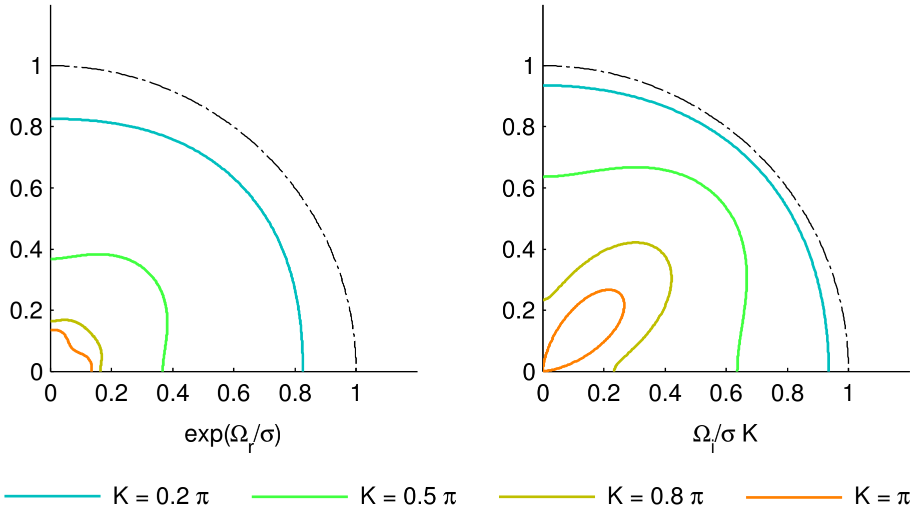


Figure 3.12: Polar plots of (left) dissipation factor and (right) normalised phase speed for the first order upwind scheme at various wavenumbers.

Although polar plots such as those in Figure 3.12 are good at showing where anisotropy occurs, it is not obvious how one should gauge the 'general' isotropy of a scheme. There are several variables to aggregate: propagation angle θ, nondimensional wavenumber K, and the real and imaginary parts of the eigenvalues.

We can begin to quantify isotropy by observing that anisotropy is at its strongest at $\theta = 45^\circ$ (this is not always the case, but it will be seen in the following chapter that it is generally true). Furthermore, we know the isotropic ideal: it is the spectrum evaluated at $\theta = 0^\circ$, which should be the same as the equivalent 1D scheme. The K_c performance metric can be modified to describe deviation from the isotropic ideal rather than deviation from the exact solution. Thus,

Equation (2.5) becomes

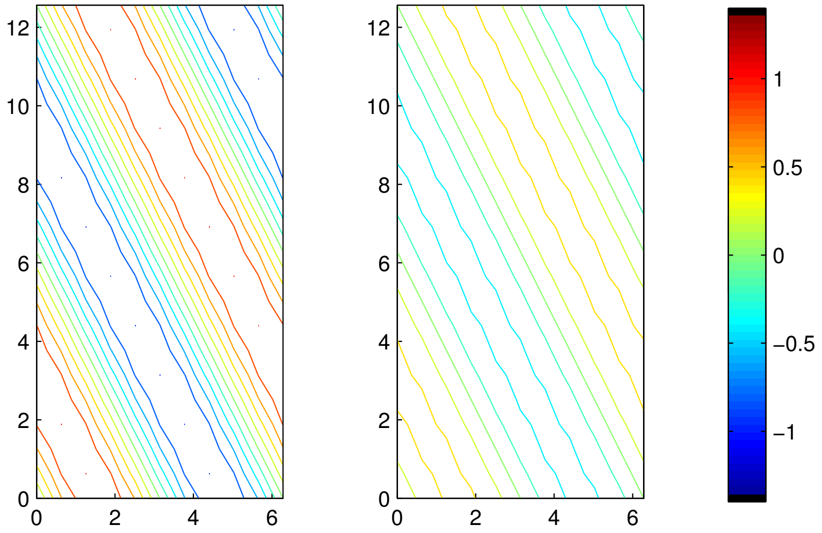
$$\left|\Omega_r(K) - \Omega_r^{\text{iso}}(K)\right| \big/ \sigma < 0.005 \quad \text{(dissipation)} \tag{3.94a}$$

$$\text{and} \quad \left|\Omega_i(K) - \Omega_i^{\text{iso}}(K)\right| \big/ \sigma < 0.005 \quad \text{(dispersion)} \tag{3.94b}$$

The largest wavenumber satisfying this criteria gives an ‘isotropic resolution’, K_c^{iso}.

3.4.2 Performance in the Time Domain

As with the one-dimensional pilot experiment, it is important for experiments in 2D to be able to reproduce the linear advection conditions assumed in analyses. On a domain of finite size, periodic boundary conditions generally prohibit the propagation of plane waves in arbitrary directions. However, it is possible to test the three directions in Figure 3.10 on a domain of aspect ratio 1:2. As shown in Figure 3.13, the periodic conditions are still respected when the propagation angle is $\tan^{-1} 0.5 \approx 26.6^\circ$. Also shown in this figure is the evolution of one wavelength by the linear Lagrange scheme, which is a semi-Lagrangian reformulation of the first order upwind scheme. Fast attenuation of the waveform due to the low dissipation accuracy can be observed.



(a) $T = 0$ **(b)** $T = 8\pi/5$

Figure 3.13: Linear advection by the linear Lagrange scheme at $\tan\theta = 1/2$, $\sigma = 1/\sqrt{5}$.

Staying with the linear Lagrange example, it is standard practice when studying spatial convergence to present two norms or means of the errors. Table 3.2 shows the convergence of $||\varepsilon||_1$ and $||\varepsilon||_\infty$ for the three angles over four grid resolutions. The expected first order convergence is observed for all three angles.

Table 3.2: Error means and their convergence for linear advection by the linear Lagrange scheme.

(a) $\theta = 0,\ \sigma = 0.1$

n_x	K	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
8	7.85e−01	5.5464e−01	—	8.9763e−01	—
16	3.93e−01	4.2623e−01	0.380	6.7060e−01	0.421
32	1.96e−01	2.7124e−01	0.652	4.2579e−01	0.655
64	9.82e−02	1.5432e−01	0.814	2.4233e−01	0.813
128	4.91e−02	8.2502e−02	0.903	1.2958e−01	0.903

(b) $\theta = \tan^{-1} 0.5 \approx 26.6^\circ,\ \sigma = {}^{2}\!/\!\sqrt{5} \times 0.1 \approx 0.0894$

n_x	K	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
8	8.78e−01	5.5076e−01	—	8.6550e−01	—
16	4.39e−01	4.0014e−01	0.461	6.2700e−01	0.465
32	2.20e−01	2.4808e−01	0.690	3.8926e−01	0.688
64	1.10e−01	1.3921e−01	0.833	2.1860e−01	0.832
128	5.49e−02	7.3890e−02	0.914	1.1606e−01	0.914

(c) $\theta = 45^\circ,\ \sigma = {}^{1}\!/\!\sqrt{2} \times 0.1 \approx 0.0707$

n_x	K	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
8	1.11e+00	5.5464e−01	—	8.9763e−01	—
16	5.55e−01	4.2623e−01	0.380	6.7060e−01	0.421
32	2.78e−01	2.7124e−01	0.652	4.2579e−01	0.655
64	1.39e−01	1.5432e−01	0.814	2.4233e−01	0.813
128	6.94e−02	8.2502e−02	0.903	1.2958e−01	0.903

A important detail is that the nondimensional wavenumber, formally defined as $K := k\Delta x$, and Courant number, $\sigma := a\Delta t / \Delta x$, are not consistent between the three propagation angles. This is because the wavelength and wavespeed are modified by a factor of $\cos\theta$, as illustrated in Figure 3.14. It explains why the errors in Tables 3.2a and 3.2c are identical. Recalling Figure 3.11, $\Omega(K)/\sigma$ for $\theta = 45°$ is the same as that for $\theta = 0°$ but scaled up by $\sqrt{2}$ in Ω and down by $\sqrt{2}$ in K. Modifying the Courant number by $\cos(45°) = 1/\sqrt{2}$ and the nondimensional wavenumber by $1/\cos(45°) = \sqrt{2}$ therefore makes the $\theta = 45°$ spectrum identical to the $\theta = 0°$ one.

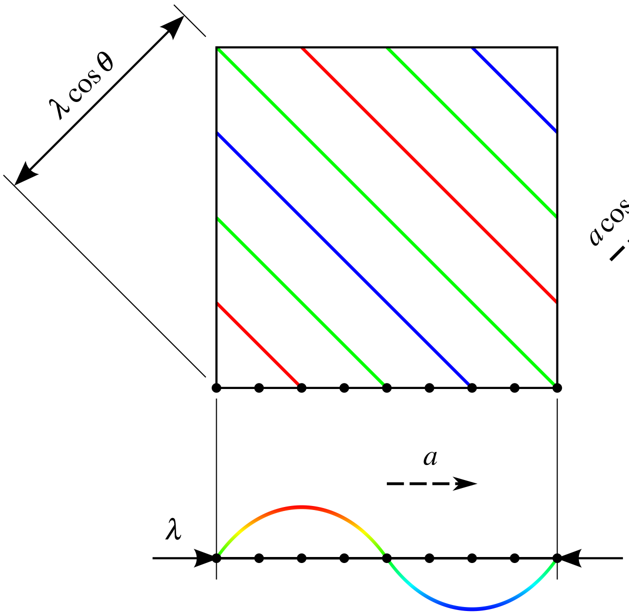


Figure 3.14: How the wavelength and wavespeed become modified at oblique propagation angles. If these did not affect the formal nondimensional wavenumber and Courant number, anisotropy could be measured by sampling the 2D solution at a Cartesian boundary and subtracting the results from an equivalent 1D simulation.

Since the Courant numbers are small, the physical-mode error norm (3.47) derived from the first order upwind spectra can be used to verify that the relative magnitudes of errors are correct. In fact, Figure 3.15 shows that there is favourable agreement between $||\varepsilon(K)||^{\text{phys}}$ and $||\varepsilon||_\infty$.

Unfortunately, the inconsistency of the Courant number between the three sets of simulations makes it difficult to quantify anisotropy in the results. If the Courant number were unmodified, one could simply sample the 2D solution at one of the Cartesian boundaries, as shown in Figure 3.14, and subtract the results of an equivalent 1D simulation with matching points. But the modification of the Courant number by an irrational number makes it impossible to construct an equivalent 1D simulation, since the numbers of grid points and time steps must remain integers. In the present work, then, anisotropy will be measured using spectral

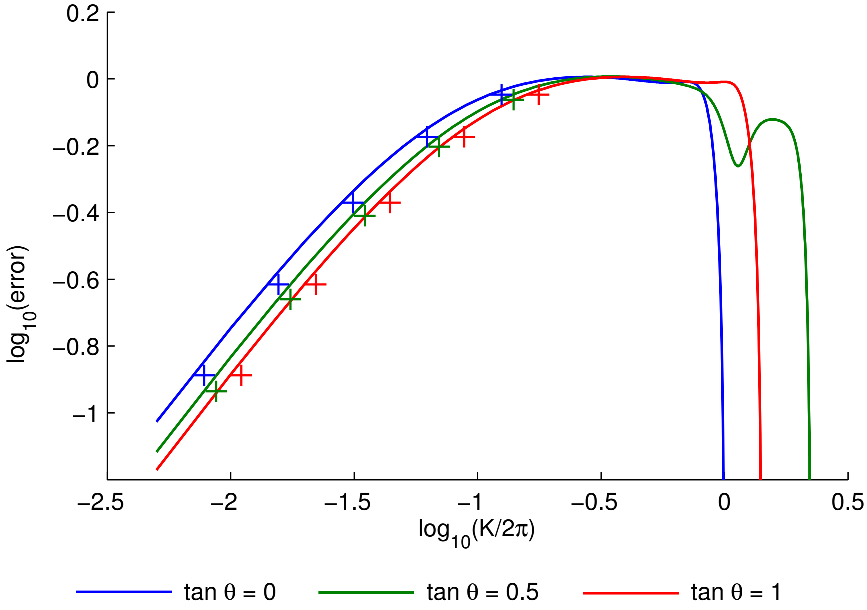


Figure 3.15: 2D linear advection errors with (lines) physical-mode error norm $\|\varepsilon(K)\|^{\text{phys}}$, computed from first order upwind spectra, and (crosses) $\|\varepsilon\|_{\infty}$, from the linear Lagrange simulation results.

information only. An appropriate modification to the physical-mode error norm reads

$$\|\varepsilon(K)\|^{\text{phys,iso}} := U\left|\exp\left(\frac{2\pi N\Omega(K)}{\sigma K}\right) - \exp\left(\frac{2\pi N\Omega^{\text{iso}}(K)}{\sigma K}\right)\right| \tag{3.95}$$

Although this expression is still a function of K, the relative values for same-order schemes should converge as $K \to 0$.

3.5 Preliminaries for the Study of 2D Multi-Moment Schemes

Up to this point, theory and methods that uncover the numerical properties of space and time discretisations have been explored. Where appropriate, modifications and extensions have been made to help strengthen the methodology of the present thesis. The examinations of 1D/2D first order upwind and 1D multi-moment schemes appear only for illustrative purposes, since these schemes have already been investigated thoroughly in the literature. We now arrive at the point at which 2D multi-moment schemes may be analysed and original results generated. This section serves as a prologue to Chapters 4–7, presenting common theory and methods but deferring all analysis. We begin by augmenting the 2D Fourier analysis with the matrix method, just as we did for the 1D analysis.

In §3.4.1 it was explained that any 2D space scheme could be characterised by two sets of weights $w^x_{\alpha,\beta}$ and $w^y_{\alpha,\beta}$, with each weight corresponding to a grid point (α,β) in the stencil.

Equation (3.84) then puts the spectrum, which is a function of wavenumber, propagation angle and grid spacing aspect ratio, in terms of these weights. This is straightforwardly extended to multi-moment schemes. Any 2D multi-moment scheme can be characterised by two sets of matrices $\mathbf{W}^{x}_{\alpha,\beta}$ and $\mathbf{W}^{y}_{\alpha,\beta}$, with each matrix corresponding to a grid element. They may be combined to form a final matrix, analogous to (3.84),

$$\mathbf{W}(K,\theta,\gamma) = -\cos\theta \sum_{\alpha,\beta} \mathbf{W}^{x}_{\alpha,\beta} \mathrm{e}^{\mathrm{I}K(\alpha\cos\theta+\beta\gamma\sin\theta)} - \frac{1}{\gamma}\sin\theta \sum_{\alpha,\beta} \mathbf{W}^{y}_{\alpha,\beta} \mathrm{e}^{\mathrm{I}K(\alpha\cos\theta+\beta\gamma\sin\theta)} \quad (3.96)$$

which when subjected to an eigendecomposition yields $\Omega(K,\theta,\gamma)/\sigma$.

3.5.1 Notation

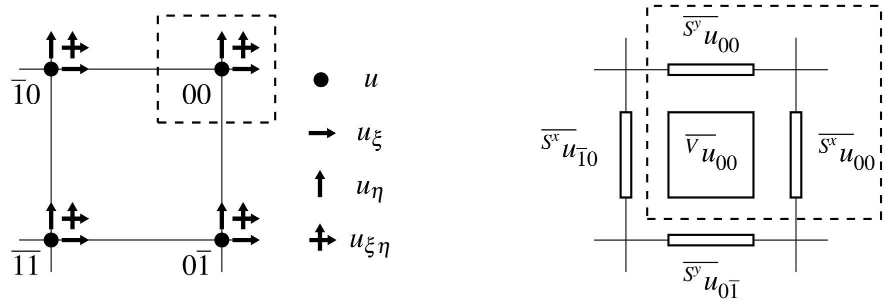
This subsection establishes conventions for describing moments, their relative location, and their associated weights.

- Grid point coordinates are normalised according to Equations (3.87) so that they can be expressed as integers. For instance, the point value $u_{i,j-1}$ can be rewritten $u_{0\bar{1}}$. Negative (overlined) integers correspond to upwind points.
- Point derivatives are denoted u_ξ, u_η, etc. To be consistent with integrated averages, which are essentially integrations normalised by the grid spacing, we have chosen to differentiate with respect to the normalised coordinates rather than by x and y. This normalises differentiations by the grid spacing; c.f. Equation (3.51).
- Integrated averages (VIAs and SIAs) are denoted $\overline{^{S^x}u}$, $\overline{^{S^y}u}$ and $\overline{^{V}u}$ as introduced for IDO–FVM in §3.3.1. The superscript on S signifies the normal direction of the surface, so S^x corresponds to vertical faces and S^y to horizontal faces. Faces may sometimes be referred to as lines when discussing 2D cell geometry. The term ‘integrated average’ may sometimes be shortened to ‘value’.

When describing profile reconstructions, it is often useful to depict the layout of the constraining moments. Figure 3.16 defines a schematic notation. Higher-dimensional grid elements share the same coordinate notation as the grid points but are necessarily offset in space. Given two elements with the same coordinates, the higher-dimensional element is defined to be on the upwind side to dispel ambiguity.

Finally, when using symbols to represent weights in the matrices $\mathbf{W}^{x}_{\alpha,\beta}$ and $\mathbf{W}^{y}_{\alpha,\beta}$, it is important to distinguish, for each weight, (i) the type of moment being evolved, and (ii) the moment supporting the calculation, including its location in the stencil. We have previously seen that weights tend to be computed in row vectors aggregating the supporting moments, with each vector associated with an evolving moment type.

Anterior superscripts will indicate the evolving moment type, and posterior subscripts will indicate the supporting moment. Unsplit schemes will additionally have a posterior superscript


$\bar{1}0$
00
$\bar{1}\bar{1}$
$0\bar{1}$
u
u_ξ
u_η
$u_{\xi\eta}$
$\overline{{}^{S^y}u}_{00}$
$\overline{{}^{S^x}u}_{\bar{1}0}$
$\overline{{}^{V}u}_{00}$
$\overline{{}^{S^x}u}_{00}$
$\overline{{}^{S^y}u}_{0\bar{1}}$

(a) Point moments

(b) Integrated averages

Figure 3.16: Notation for the layout of moments in a cell. The dashed box emphasises that the '00' moments are being evolved. The wind direction is to the right and/or upwards.

indicating the direction of spatial differentiation (x or y). Split schemes are based on 1D operators that do not need to distinguish this direction; however, when the operators are mixed, it becomes necessary to modify the anterior superscripts to make them unique.

Weights for 1D operators that may reappear in split 2D schemes are given the following symbols.

$$\text{First order upwind (FOU)}: \quad {}^F\mathbf{w} := \begin{pmatrix} {}^F w_0 & {}^F w_{\bar{1}} \end{pmatrix} \tag{3.97}$$

$$\text{IDO for values}: \quad {}^I\mathbf{w} := \begin{pmatrix} {}^I w_0, & {}^I w_{\xi 0}, & {}^I w_{\bar{1}}, & {}^I w_{\xi\bar{1}} \end{pmatrix} \tag{3.98}$$

$$\text{IDO for gradients}: \quad {}^{I\xi}\mathbf{w} := \begin{pmatrix} {}^{I\xi} w_0, & {}^{I\xi} w_{\xi 0}, & {}^{I\xi} w_{\bar{1}}, & {}^{I\xi} w_{\xi\bar{1}} \end{pmatrix} \tag{3.99}$$

$$\text{IDO}-\text{FVM for cell values}: \quad {}^V\mathbf{w} := \begin{pmatrix} {}^V w_{S0}, & {}^V w_{V0}, & {}^V w_{S\bar{1}} \end{pmatrix} \tag{3.100}$$

$$\text{IDO}-\text{FVM for boundary values}: \quad {}^S\mathbf{w} := \begin{pmatrix} {}^S w_{S0}, & {}^S w_{V0}, & {}^S w_{S\bar{1}} \end{pmatrix} \tag{3.101}$$

3.5.2 Dealing with Redundancy

There will be certain situations whereby the moments available in the stencil outnumber the profile coefficients to be determined. An example is Type-A CIP/IDO (Yabe *et al.*, 1991). In 2D, the scheme has a four-point stencil of three moments per point – a total of 12 moments at its disposal. However, the order-complete profile to be constrained only needs 10 coefficients. Yabe *et al.* deal with this situation by omitting two of the moments from the reconstruction.

We may choose instead to keep the matrix $\mathbf{A}$ overdetermined, then take the Moore–Penrose pseudoinverse $\mathbf{A}^+$. While not respecting the moments exactly, the resulting profile will minimise the error in a least squares (L^2) sense. This is especially useful when it is not obvious which moments to omit from the reconstruction, for instance when the omitted moments would lead to an asymmetric stencil.

Still, it is sometimes useful or even mandatory to have one or more of the moments exactly constrained, relegating the others to the least squares space. An example is the cell-integrated average of the FVM-augmented methods, which needs to be exactly constrained to ensure conservation. In these cases we can apply a constrained least squares treatment. The constrained least squares approach was encountered in §2.3.1, where it was found that Ollivier-Gooch and Van Altena (2002) use it to deal with arbitrarily large stencils of control volumes while respecting boundary conditions exactly.

The treatment used in the present work is based on the nullspace method of Björck (1996) which makes use of QR factorisations. We begin with the $m \times n$ matrix $\mathbf{A}$. The matrix is partitioned into $\mathbf{A}_E$, consisting of p rows corresponding to exact constraints, and $\mathbf{A}_L$, consisting of $(m-p)$ rows to be treated by least squares. The right hand side column vector is correspondingly partitioned into $\mathbf{u}_E$ and $\mathbf{u}_L$. $\mathbf{A}_E$ is required to be of full rank, i.e. $\mathrm{rank}\,(\mathbf{A}_E) = p$.

The strategy is to decompose the solution vector $\mathbf{c}$ according to

$$\mathbf{c} = \mathbf{x} + \mathbf{Q}_{E2}\,\mathbf{y} \tag{3.102}$$

where $\mathbf{x}$ is a particular solution generated from the row space of $\mathbf{A}_E$, $\mathbf{Q}_{E2}$ is an orthogonal basis for the nullspace of $\mathbf{A}_E$, and $\mathbf{y}$ is the least squares solution to the reduced system

$$\min_{\mathbf{y}} \|\, (\mathbf{A}_L\,\mathbf{Q}_{E2})\,\mathbf{y} - (\mathbf{u}_L - \mathbf{A}\,\mathbf{x})\,\| \tag{3.103}$$

However, we reformulate Björck's method so that the end result is not a solution vector but some pseudoinverse $\mathbf{M}$. This removes the dependence on $\mathbf{u}$ so that spectral analysis can proceed. The terms are related by

$$\mathbf{c} = \mathbf{M}\,\mathbf{u} \equiv \begin{pmatrix} \mathbf{M}_E & \mathbf{M}_L \end{pmatrix} \mathbf{u} \tag{3.104}$$

$\mathbf{M}_E$ and $\mathbf{M}_L$ are computed as follows:

1. Factorise $\mathbf{A}_E^{\mathsf{T}}$ into an orthogonal matrix $\mathbf{Q}_E$ and upper-triangular matrix $\mathbf{R}_E$. The first p columns of $\mathbf{Q}_E$ form a basis for the row space of $\mathbf{A}_E$, while the remaining $n-p$ columns form a basis for the null space. These sets of columns are denoted $\mathbf{Q}_{E1}$ and $\mathbf{Q}_{E2}$ respectively. $\mathbf{R}_E$ can be trimmed to its first p rows to leave $\mathbf{R}_{E1}$.

$$\mathbf{A}_E^{\mathsf{T}} = \mathbf{Q}_E\mathbf{R}_E \equiv \begin{pmatrix} \mathbf{Q}_{E1} & \mathbf{Q}_{E2} \end{pmatrix} \begin{pmatrix} \mathbf{R}_{E1} \\ \mathbf{0} \end{pmatrix} \tag{3.105}$$

2. Factorise $\mathbf{A}_L\,\mathbf{Q}_{E2}$ into orthogonal and upper triangular matrices.

$$\mathbf{A}_L\,\mathbf{Q}_{E2} = \mathbf{Q}_L\mathbf{R}_L \tag{3.106}$$

3. The particular solution to the exactly constrained part is given by $\mathbf{x} = \mathbf{Q}_{E1}\,\mathbf{R}_{E1}^{-\mathsf{T}}\,\mathbf{u}_E$. This can be represented by the linear operation $\mathbf{x} = \mathbf{M}_{1E}\,\mathbf{u}_E$. Therefore, compute:

$$\mathbf{M}_{1E} = \mathbf{Q}_{E1}\,\mathbf{R}_{E1}^{-\mathsf{T}} \tag{3.107}$$

4. The least squares solution to the remaining part is given by $\mathbf{y} = \mathbf{R}_L^{-1}\,\mathbf{Q}_L^{\mathsf{T}}\,(\mathbf{u}_L - \mathbf{A}_L\mathbf{x})$. Expand this and extract the two matrices representing the operations on $\mathbf{u}_L$ and $\mathbf{u}_E$.

$$\mathbf{M}_{2L} = \mathbf{R}_L^{-1}\,\mathbf{Q}_L^{\mathsf{T}} \tag{3.108}$$

$$\mathbf{M}_{2E} = -\mathbf{M}_{2L}\,\mathbf{A}_L\,\mathbf{M}_{1E} \tag{3.109}$$

5. Finally expand Equation (3.102) and extract the matrices $\mathbf{M}_E$ and $\mathbf{M}_L$ for which $\mathbf{c} = \begin{pmatrix} \mathbf{M}_E & \mathbf{M}_L \end{pmatrix}\mathbf{u}$.

$$\mathbf{M}_L = \mathbf{Q}_{E2}\,\mathbf{M}_{2L} \tag{3.110}$$

$$\mathbf{M}_E = \mathbf{M}_{1E} + \mathbf{Q}_{E2}\,\mathbf{M}_{2E} \tag{3.111}$$

3.6 Chapter Summary

In this chapter, it has been argued that a multi-moment scheme is characterised more prominently by its space discretisation than by its semi-Lagrangian or Eulerian formulation. Therefore it is appropriate to analyse the scheme's semi-discrete system, converting all semi-Lagrangian operators to Eulerian ones. Some standard frequency-domain techniques have been introduced. Additionally, it has been shown how dissipative and dispersive orders of accuracy may be derived.

In the time domain, errors are to be aggregated and measured by norms. The rate of convergence with grid resolution indicates the scheme's order of accuracy. Takacs (1985) further gives us a means of separating the errors into dissipation and dispersion components. A possibly original contribution in this chapter is the definition of a 'physical-mode error norm', $\|\varepsilon(K)\|^{\mathrm{phys}}$. It provides a more direct means of cross-verification between spectra and simulations than simply comparing the rates of convergence.

Fourier series decomposition can be combined with the matrix method to analyse systems with both periodicity and multiple types of moment. Analysis of the fully discrete CIP yielded amplification factors that corroborate the results of Utsumi *et al.* (1997). Analysis of the semi-discrete IDO, the conservative IDO–FVM and the discontinuous MCV yielded spectra that are identical. That is not to say that these schemes will perform identically in various nonlinear situations, however.

In a pilot experiment comparing various third order schemes in 1D, it was found that the temporal error does not affect the observed spatial convergence as long as the Courant number is controlled. A small Courant number will also see the semi-Lagrangian scheme approximating its Eulerian counterpart. This is a great convenience because it means the economical semi-Lagrangian formulation can be used for testing. Also, the compact schemes were found to be precisely six times more accurate than their noncompact counterparts.

With some modification to the mathematics, the analysis can be extended to two spatial dimensions. Schemes can be analysed as split or unsplit. In the case of first order upwind, it makes no difference to the spectrum, which demonstrates that split and unsplit schemes are often identical in a semi-discrete sense. To visualise the results, we can generate polar plots or evaluate the spectra at certain wave propagation angles. Anisotropy is generally strongest at 45°; measures for isotropy can be defined by modifying $||\varepsilon\,(K)\,||^{\mathrm{phys}}$ and Hu *et al.*'s (1999) K_c and evaluating at that angle. Unfortunately, for periodic waves propagating at oblique angles, the formally defined Courant number is modified by an irrational number. This means we have to rely on frequency-domain information rather than time-domain tests when quantifying isotropy.

Some notation and algebra has been introduced in preparation for the main study of 2D schemes, including linear algebra for profile fitting with constrained least squares.

Chapter 4

Comparison with the Discontinuous Galerkin Method

4.1 Introduction

Having established a methodology for analysing and testing Cartesian multi-moment schemes, we can begin to address the objectives given at the end of Chapter 2. The first objective relates to the observation that the third order interpolated differential operator (IDO) and second order discontinuous Galerkin (DG2) schemes have almost the same value for K_c, the resolution defined by Hu *et al.* (1999). IDO's K_c was estimated to be 0.787, which was based on the amplification factors at a Courant number of 0.1 after fourth order time integration. DG's K_c was calculated to be 0.786 using formulae by Zhang and Shu (2005), which is consistent with the value of 0.8 given by Hu *et al.*. These similarities beg the question: could IDO spectra in fact be identical to DG spectra? And more importantly, why do two methods with the same 'resolution' and possibly the same spectra have different orders of accuracy? The first part of this chapter addresses these by comparing DG and IDO in one dimension.

One might object to the choice of IDO for the comparison. The multi-moment constrained finite volume method (MCV) is arguably more appropriate, since it has the same spectrum as IDO (§3.3) but greater similarity to DG in its piecewise-discontinuous profile reconstructions. For interest, then, the third order MCV scheme is also included here. Conversely, one might argue that the continuous Galerkin (CG) method is more appropriate for comparison with CIP/IDO, given the piecewise-continuous reconstructions. But CG involves the inversion of a *global* mass matrix, lending it an implicit semi-discrete form and making it troublesome for more general advection problems. It is therefore not considered in this chapter.

It is interesting to extend the question of spectral equivalence to two dimensions, since Hu *et al.* show how isotropy of the DG method is affected by whether a tensor product or order-complete basis is selected. We do not have to look far to find corresponding multi-moment schemes. As mentioned in §2.2.2, the Type-A constrained interpolation profile (CIP) scheme (Yabe *et al.*, 1991) has an order-complete basis, while Type-C CIP (Aoki, 1995), despite being a split scheme with homogeneous operators, has a set of moments that fully supports a tensor product basis. However, the profile reconstruction in Type-A is not unique. Moments farthest from the point of interest are omitted, and the resulting bias may affect the spectrum. Variations on Type-A can be created using constrained least squares fitting as described in §3.5.2. In particular, an unconstrained least squares treatment of all moments would mean equal influence of the moments on the space operator. It is hypothesised that if any of the Type-A variants match the order-complete DG spectrum, it is this one.

4.2 One Spatial Dimension

This section documents the analysis of various orders of one-dimensional DG and IDO followed by the testing of DG and the constrained interpolation profile method (CIP). CIP was chosen to replace IDO in tests since, as noted in §3.3.2, semi-Lagrangian formulations are more efficient and stable than their Eulerian counterparts and yield approximately the same results at small Courant numbers. It may be recalled that analysis and testing of third order CIP/IDO was already carried out in Chapter 3. However, this section goes further by extending the CIP/IDO discretisation to higher orders. The results make it possible to quantify exactly how CIP/IDO and DG compare with one another.

4.2.1 Formulations and Simulation Method

It will be recalled that the original CIP/IDO method involves a degree-three polynomial which is fitted between points according to two point values and two point gradients. Higher degrees of polynomial can be determined by storing additional higher order spatial derivatives at the points. Second order derivatives lead to degree-five profiles, third order derivatives lead to degree-seven profiles, and so on. The polynomial degree can be increased by two each time because there are two more constraints: one at point i and another at $i-1$. We could alternatively store moments at intermediate locations such as $i-1/2$, but this would not be efficient use of memory. Each such moment could only be used by one profile, whereas each moment at point i, $i-1$, etc. is shared by two profiles.

With the above strategy for constructing higher-degree IDO scheme, matrices of weights can be generated by applying the analytical method in §3.3.1. The matrices are given in Appendix A.

For the analysis of the DG method, the method of Hu *et al.* (1999) was followed up to the calculation of matrices $\mathbf{Q}$, $\mathbf{N}_0$ and $\mathbf{N}_{-1}$. These were converted to weights according to

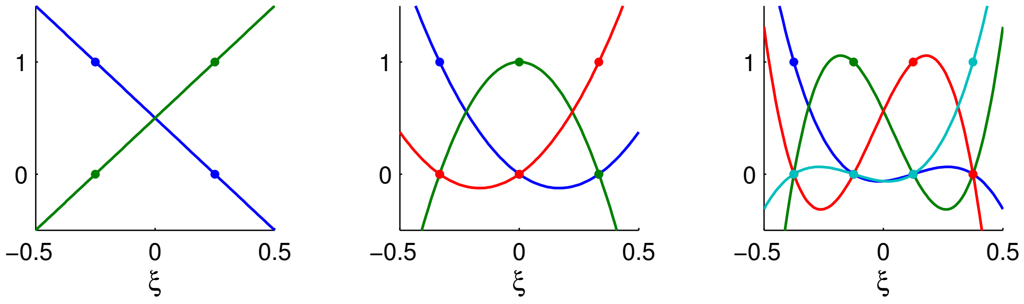
$$\mathbf{W}(K) = \Delta\xi \mathbf{Q}^{-1}\left(\mathbf{N}_0 + \mathbf{N}_{-1}\mathrm{e}^{-\mathrm{I}K}\right) \tag{4.1}$$

where $\Delta\xi$ represents the width of the local basis function, which is one by our convention and two if using the more standard finite element mapping onto $[-1,1]$. In the work of Hu *et al.*, the local basis functions go unspecified. As long as they reproduce the correct degree of polynomial, they are incidental to the analysis. For the purpose of testing, however, the example of Zhang and Shu (2005) was followed which defines basis functions according to equispaced point values. This enables the initial condition to be specified by collocation points.

Basis functions for polynomial degrees $p = 1$ through $p = 3$ are depicted in Figure 4.1. Each basis function is given by

$$\upsilon_j(\xi) = \sum_{i=0}^{p} c_{i,j}\xi^i \tag{4.2}$$

where c is the ith coefficient of the jth basis function (in this context i counts monomials, not grid points). The relationship between the Vandermonde matrix and assembled profile coefficients is $\mathbf{A}\,\mathbf{C} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, and so the coefficients are simply given by $\mathbf{A}^{-1}$.



(a) $p = 1$ **(b)** $p = 2$ **(c)** $p = 3$

Figure 4.1: Basis functions for the one-dimensional discontinuous Galerkin (DG) method.

Linear advection tests used the same parameters as the pilot experiment in §3.3.2: a sine wave of unit height and wavenumber was advected by unit velocity over a distance of one wavelength. The Courant number was made equal to 0.1 which constrained the relationship between grid size m and number of time steps n. It was possible to give all of the CIP schemes a forward Euler integration of the departure point; the uniform velocity meant this integration would be exact anyway. The Eulerian schemes were given third order Runge-Kutta time integrations. For high-order DG schemes this would mean the temporal error would start to dominate, so the Courant number was reduced in these instances until the temporal error was clearly subdominant.

4.2.2 Results

Figure 4.2 shows the DG spectra with IDO spectra superimposed as dots. It is confirmed that the third order IDO scheme has an identical spectrum to the second order DG scheme. In general, the IDO scheme with order $(2N-1)$ has an identical spectrum to the Nth order DG scheme. Properties of these spectra are summarised in Table 4.1.

Curves of the physical-mode error norm $\|\varepsilon(K)\|^{\text{phys}}$ were computed from spectra, while values of $\|\varepsilon\|_\infty$ were taken from test results. The errors are shown plotted together in Figure 4.3.

4.2.3 Discussion

Figure 4.3 clearly shows that IDO and DG do not perform identically in the time domain, despite having identical spectra. Although it is counterintuitive that the spectrum of the space operator combined with some time scheme should not hold all the information needed to predict the development of errors, that is what happens with the DG method. One may conclude that the weak formulation is responsible for an impaired order of accuracy compared with multi-moment and finite difference schemes.

To shed more light on this issue, we turn to the literature on DG and its convergence. Lowrie (1996) defines three different types of norm:

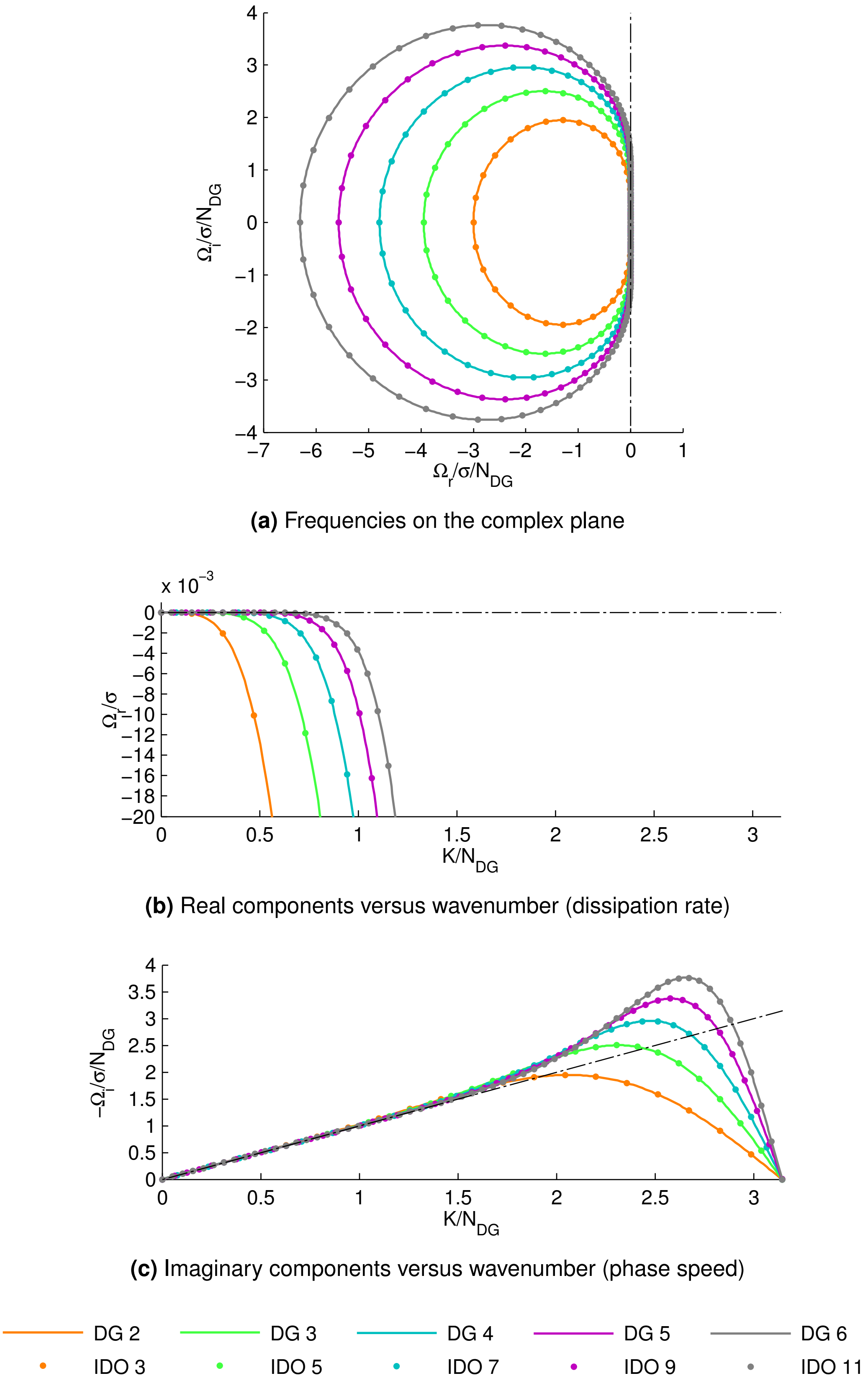
$$L_\lambda(u) := \left(\frac{1}{|V_{dom}|}\int_{V_{dom}} |u_N(\mathbf{r}) - u_E(\mathbf{r})|^\lambda \, \mathrm{d}V\right)^{1/\lambda} \tag{4.3}$$

$$\overline{L}_\lambda(u) := \left(\frac{1}{m}\sum_{i=1}^{m} |P_i(u_N) - P_i(u_E)|^\lambda\right)^{1/\lambda} \tag{4.4}$$

$$\overline{L}^{ev}_\lambda(u) := \left(\frac{1}{m}\sum_{i=1}^{m} |P^{ev}_i(u_N) - P^{ev}_i(u_E)|^\lambda\right)^{1/\lambda} \tag{4.5}$$

where some of the symbols have been changed to suit our nomenclature. V_{dom} represents the volume of the domain and $\mathbf{r}$ is the Euclidean vector representing position. $P_i(u)$ is the cell-integrated average of flux function $F(\mathbf{r})$, computed using the nominal order of quadrature. $P^{ev}_i(u)$ is $P_i(u)$ projected onto the physical-mode eigenvalue of the update operator; therefore the last norm, $\overline{L}^{ev}_\lambda(u)$, measures only the 'evolution' error. There is another error associated with the projection of the initial condition onto the physical mode, and this gets added to the evolution error.

It will be recalled that DG's nominal order of accuracy N is $(p+1)$, where p is the degree of polynomial basis. According to Lowrie, this is indeed the convergence rate in L_2 for many smooth solutions, although degradation to $\mathcal{O}\left(\Delta x^{p+1/2}\right)$ is also possible. He goes on to show that $\overline{L}^{ev}_\lambda(u)$ converges with $\mathcal{O}\left(\Delta x^{2p+1}\right)$ in linear advection. The latter rate is interesting because it is equal to the nominal order of accuracy for a multi-moment method with the same spectrum,



(a) Frequencies on the complex plane

(b) Real components versus wavenumber (dissipation rate)

(c) Imaginary components versus wavenumber (phase speed)

Figure 4.2: High-order interpolated differential operator (IDO) and DG spectra. In accordance with Hu *et al.* (1999), axes except for the dissipation rate have been normalised by DG scheme order N.

Table 4.1: Spectral properties of IDO and DG in one dimension.

DG order	IDO order	Spectral radius	K_c
2	3	6.0	0.786
3	5	11.8	1.883
4	7	19.2	3.195
5	9	27.8	4.637
6	11	37.8	6.164

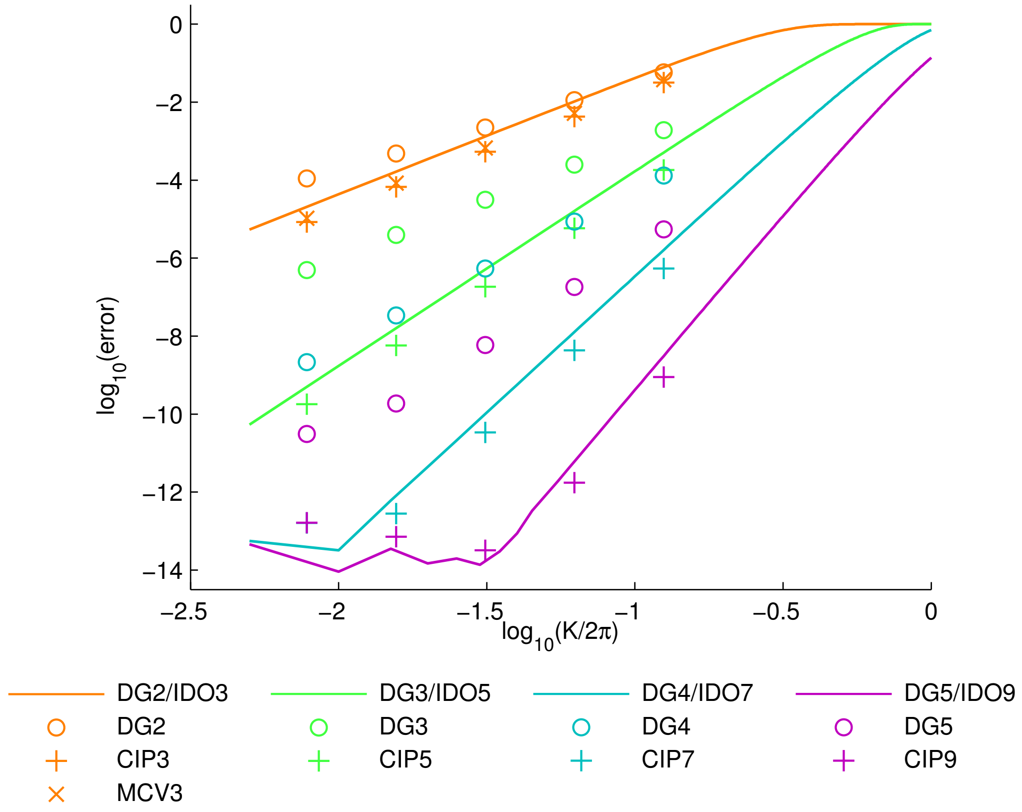


Figure 4.3: Linear advection errors with (lines) physical-mode error norm $\|\varepsilon(K)\|^{\text{phys}}$, computed from spectra, and (markers) $\|\varepsilon\|_\infty$, from test results.

$(2N-1)$. Or, looking at it another way, the rate is equal to that of our physical-mode error norm. It may be concluded that DG's observed rate of $\mathcal{O}\left(\Delta x^{p+1}\right)$ is dominated by the initial projection error which is not taken into account in the spectral analysis.

Before attempting to make a fair comparison between DG and multi-moment methods, it must be asked whether or not DG's initial projection error should be discounted. For the present discussion, it will be assumed that it is difficult or impossible to eliminate the initial projection error in practical applications, and this is why the DG order of accuracy is widely accepted to be $(p+1)$. Therefore we ought to compare methods of the same order rather than the same spectrum. Measures derived solely from the spectrum, such as K_c and $||\varepsilon(K)||^{\text{phys}}$, cannot be used to compare the relative performance of DG and multi-moment methods. But we can continue to use these measures to quantify the performance of multi-moment schemes, both among themselves and in relation to conventional finite difference/volume methods. The validity of $||\varepsilon(K)||^{\text{phys}}$ for predicting finite difference errors was demonstrated in Chapter 3.

When the ratios of errors between CIP (or MCV) and DG are computed for third and fifth orders on the fine grids, it is found that DG is the more accurate method. For third order, the ratio is 17.1 in $||\varepsilon||_\infty$, rising to 25.1 in $||\varepsilon||_1$. For fifth order, the ratio is 30.8 in $||\varepsilon||_\infty$, rising to 54.6 in $||\varepsilon||_1$. The latter ratios are based on the second finest grid ($m=64$) since DG5 suffers from machine rounding errors on the finest grid. It may be noted that the calculation of $||\varepsilon||_\lambda$ is sensitive to point arrangement; Lowrie's integration-based L_λ is arguably a better measure. Nevertheless, the accuracy improvement of DG over multi-moment methods is more or less consistent with observations made in §2.2.1 (p. 29).

On the other hand, we are now comparing methods with different spectra and moment storage costs. Where DG and MCV store N moments or degrees of freedom per cell, CIP/IDO only needs to store $(N+1)/2$ rounded up. For Eulerian schemes, the spectral radius is also significant. In general, the largest stable Courant number is inversely proportional to the spectral radius. IDO has a smaller spectrum than DG for a given order of accuracy, allowing us to reduce the number of time steps by approximately one half. This also applies to MCV.

The costs and benefits of choosing CIP/IDO over DG are summarised in 4.2. MCV shares the same spectral properties and errors as CIP/IDO, but there is no improvement over DG in terms of storage.

Table 4.2: Properties of CIP/IDO relative to DG.

Order	Storage	Spectral radius	$\|\|\varepsilon\|\|_1$	$\|\|\varepsilon\|\|_\infty$
3	$2/3$	0.507	25.1	17.1
5	$3/5$	0.425	54.6	30.8

In the end, CIP/IDO is clearly superior in the context of linear advection and, one might suppose, other smooth solutions. For a given number of moments or degrees of freedom N stored per cell, CIP/IDO converges at a rate of $(2N-1)$ compared to DG's rate of N. We can be confident that a multi-moment scheme such as CIP/IDO is therefore a good choice when, going back to Chapter 1, a level set function is being used to model incompressible multiphase flow. An alternative is the continuous Galerkin method if one is prepared to use an implicit formulation. For compressible flows which possibly contain shocks, it makes sense to choose MCV or DG over CIP/IDO. Here MCV will converge at the same rate as DG but with inferior accuracy; on the other hand, like IDO, it will allow for a greater stable Courant number which will be greater still if its point moments are advanced by a semi-Lagrangian formulation.

4.3 Two Spatial Dimensions

At the beginning of this chapter, it was asked whether the spectra of two-dimensional DG and multi-moment methods might be identical. If they were, it would provide further evidence for a connection between the two methods. But it has been shown above that while the respective one-dimensional methods do have identical spectra, they do not give the same errors. This makes the question of spectral equivalence in two spatial dimensions less relevant. Nevertheless, we can still learn something about isotropy if we extend comparison of the spectra to 2D.

As with the 1D method, 2D CIP/IDO schemes can be extended to higher orders of accuracy. This section presents the analysis of Type-C and Type-A CIP schemes in their Eulerian (IDO) formulations. For brevity, further tests were not carried out. Two-dimensional testing will play a more important role in Chapter 5.

4.3.1 Formulations

The semi-discrete representation of each scheme is encapsulated by the sets of matrices $\mathbf{W}^x_{\alpha,\beta}$ and $\mathbf{W}^y_{\alpha,\beta}$. How these matrices are arrived at is described below. Actual numbers for the third order schemes are included in Appendix A, but the higher order matrices are too big to reproduce in this document.

It will be seen that the orders of accuracy corresponding to a given number of stored moments are consistent with the findings for one spatial dimension. For Type-C CIP/IDO, N^2 moments stored at each point lead to $(2N-1)$th order accuracy, whereas DG's N^2 tensor product basis coefficients lead to Nth order accuracy. For Type-A CIP/IDO, $N(N+1)/2$ moments stored at each point lead to $(2N-1)$th order accuracy, whereas DG's $N(N+1)/2$ order-complete basis coefficients lead to Nth order accuracy.

Type-C IDO, Unsplit

If we store the undifferentiated values u, first order spatial derivatives u_ξ and u_η, and cross-derivatives $u_{\xi\eta}$ at the grid points, we have up to 16 moments to reconstruct a profile between four grid points. It is natural, then, to consider the third order tensor product basis which has 16 coefficients. Weights can be computed by the unsplit method presented in §3.4.1. Let

$$\mathbf{r} = \begin{pmatrix} 1 & \xi & \xi^2 & \xi^3 & \eta & \xi\eta & \xi^2\eta & \xi^3\eta & \eta^2 & \xi\eta^2 & \ldots & \xi^3\eta^3 \end{pmatrix} \tag{4.6}$$

The rows of the Vandermonde matrix $\mathbf{A}$ are made up of $\mathbf{r}$, $\partial\mathbf{r}/\partial\xi$, $\partial\mathbf{r}/\partial\eta$ and $\partial^2\mathbf{r}/\partial\xi\partial\eta$ evaluated at the four point positions. The weights in row vector form are

$$\begin{aligned} \mathbf{w}^x &= \left.\frac{\partial\mathbf{r}}{\partial\xi}\right|_{0,0} \mathbf{A}^{-1} \\ {}^{\xi}\mathbf{w}^x &= \left.\frac{\partial^2\mathbf{r}}{\partial\xi^2}\right|_{0,0} \mathbf{A}^{-1} \\ {}^{\eta}\mathbf{w}^x &= \left.\frac{\partial^2\mathbf{r}}{\partial\xi\partial\eta}\right|_{0,0} \mathbf{A}^{-1} \\ {}^{\xi\eta}\mathbf{w}^x &= \left.\frac{\partial^3\mathbf{r}}{\partial\xi^2\partial\eta}\right|_{0,0} \mathbf{A}^{-1} \end{aligned} \tag{4.7a}$$

$$\begin{aligned} \mathbf{w}^y &= \left.\frac{\partial\mathbf{r}}{\partial\eta}\right|_{0,0} \mathbf{A}^{-1} \\ {}^{\xi}\mathbf{w}^y &= \left.\frac{\partial^2\mathbf{r}}{\partial\xi\partial\eta}\right|_{0,0} \mathbf{A}^{-1} \\ {}^{\eta}\mathbf{w}^y &= \left.\frac{\partial^2\mathbf{r}}{\partial\eta^2}\right|_{0,0} \mathbf{A}^{-1} \\ {}^{\xi\eta}\mathbf{w}^y &= \left.\frac{\partial^3\mathbf{r}}{\partial\xi\partial\eta^2}\right|_{0,0} \mathbf{A}^{-1} \end{aligned} \tag{4.7b}$$

and it remains to arrange these into $\mathbf{W}^x_{\alpha,\beta}$ and $\mathbf{W}^y_{\alpha,\beta}$ matrices according to the ordering of the point moments in $\mathbf{A}$.

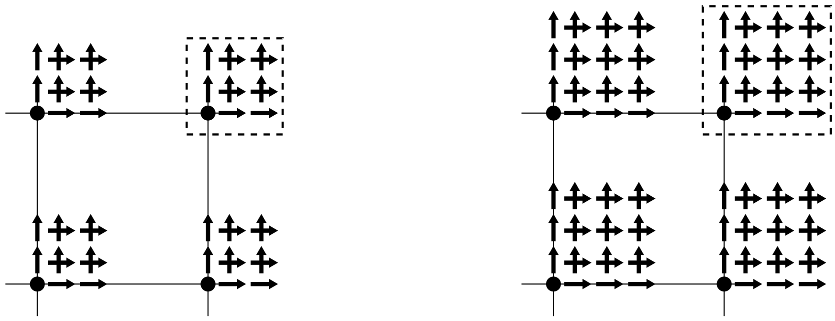
Extensions to higher degrees of reconstruction are straightforward. A 36-coefficient quintic profile is determined by nine spatial derivatives stored at each point (Figure 4.4a), a 64-coefficient septic profile by 16 spatial derivatives at each point (Figure 4.4b), and so on to higher orders.

Type-A IDO

This scheme stores u, u_ξ and u_η and uses an order-complete basis. However, there are 12 available moments, as shown in Figure 4.5a, and only 10 coefficients in the profile. Some redundancy has therefore been introduced. In this instance, the redundancy is dealt with simply by omitting the two point gradients farthest upwind as shown in Figure 4.5b. It is reminded that in such diagrams the moments being supported are located in the top and/or right hand side of the stencil (in this instance they are the top right-hand point moments) and the velocity is directed upwards and towards the right. Care must be taken during computation to rotate the stencil depending on the quadrant of the velocity direction.

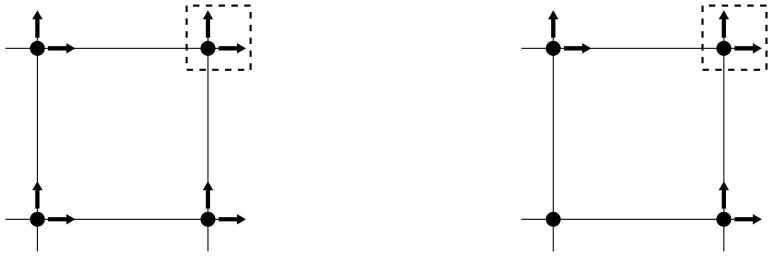
The products of ξ and η taking part in the order-complete basis are

$$\mathbf{r} = \begin{pmatrix} 1 & \xi & \xi^2 & \xi^3 & \eta & \xi\eta & \xi^2\eta & \eta^2 & \xi\eta^2 & \eta^3 \end{pmatrix} \tag{4.8}$$



(a) $p = 5$ **(b)** $p = 7$

Figure 4.4: Type-C stencils for higher degrees of profile reconstruction. The notation of §3.5.1 has been extended with higher-order moments stacked upon u_ξ, u_η and $u_{\xi\eta}$. The dashed box indicates the moments being evolved, i.e. moments subscripted '00', and the wind direction is towards the right and upwards.



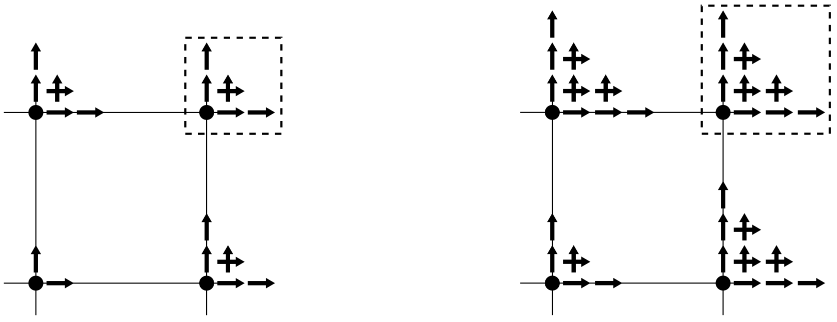
(a) available **(b)** used in the reconstruction

Figure 4.5: Moments in Type-A CIP/IDO. The dashed box indicates the moments being evolved, i.e. moments subscripted '00', and the wind direction is towards the right and upwards.

This time, the rows of $\mathbf{A}$ are made up of $\mathbf{r}$, $\partial\mathbf{r}/\partial\xi$ and $\partial\mathbf{r}/\partial\eta$ evaluated at the four point positions, but with $\partial\mathbf{r}/\partial\xi\,|_{-1,-1}$ and $\partial\mathbf{r}/\partial\eta\,|_{-1,-1}$ omitted.

Equations (4.7) can be reused when computing the row vectors of weights. However, in rearranging them to $\mathbf{W}^x_{\alpha,\beta}$ and $\mathbf{W}^y_{\alpha,\beta}$ it will be found that $w^x_{\xi\overline{11}}$, $w^x_{\eta\overline{11}}$, $w^y_{\xi\overline{11}}$ and $w^y_{\eta\overline{11}}$ are missing. As always, missing terms equate to zeros.

Extension to a quintic profile can be done by additionally storing the second-order derivatives u_{ξ^2}, $u_{\xi\eta}$ and u_{η^2}. The 21-coefficient profile would be outnumbered by the 24 moments between the four points, but we can omit from the reconstruction the three second-order derivatives farthest downwind. This arrangement is depicted in Figure 4.6a. Septic profiles can be determined by additionally storing third order derivatives and omitting those four farthest downwind (Figure 4.6b), and so on to higher orders.



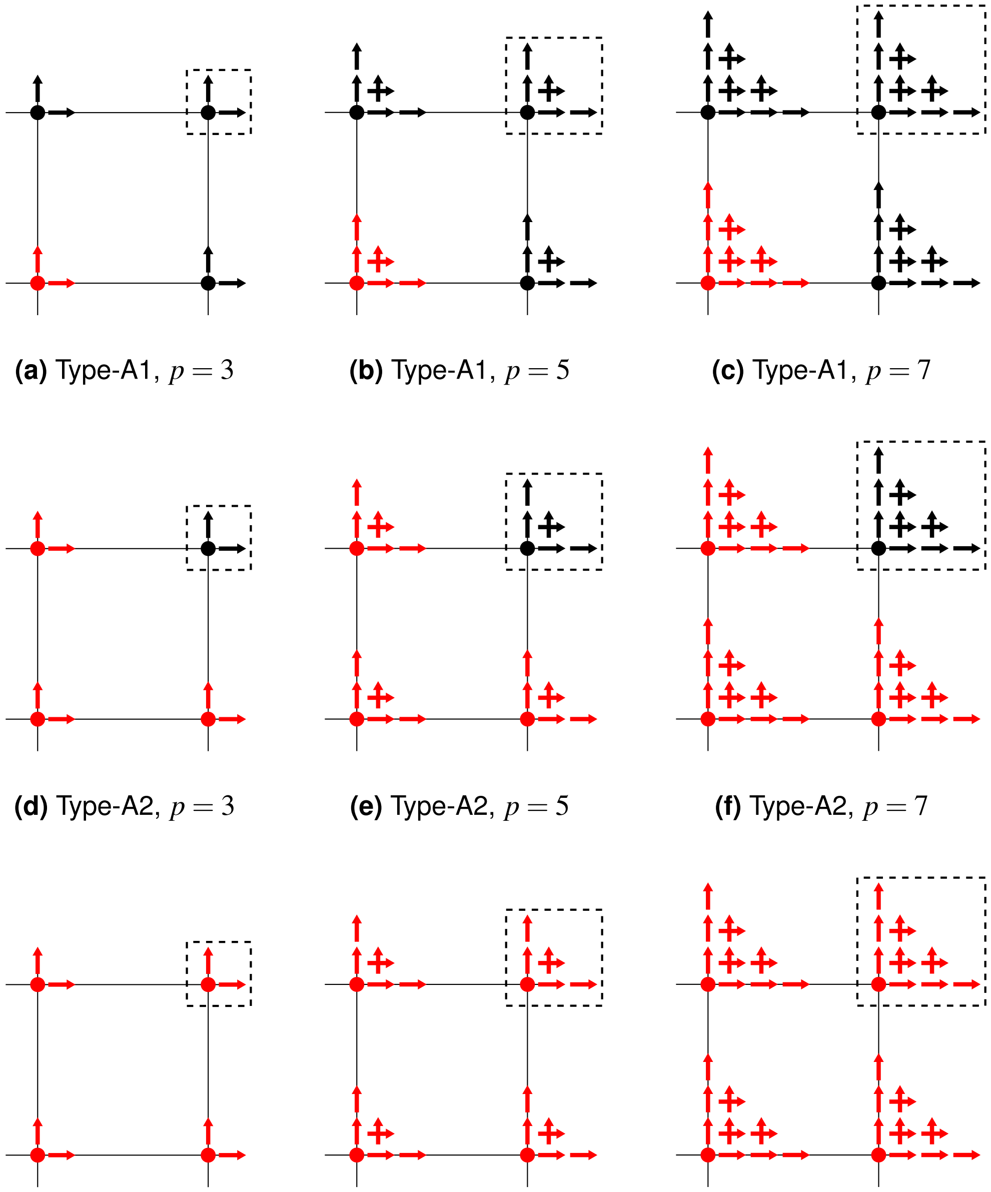
(a) $p = 5$ **(b)** $p = 7$

Figure 4.6: Type-A stencils for higher degrees of profile reconstruction. The notation of §3.5.1 has been extended with higher-order moments stacked upon u_ξ, u_η and $u_{\xi\eta}$. The dashed box indicates the moments being evolved, i.e. moments subscripted '00', and the wind direction is towards the right and upwards.

Variations on Type-A IDO

In Type-A IDO, redundancy exists and the profile reconstruction is not unique. If we take away one or more moments from the exactly constrained solution space, we can add as many moments as we like to the least squares space. In this way the influence of the available moments on the resulting space operator becomes more spread out.

Using the methodology of §3.5.2, three more variants using the order-complete basis were subjected to analysis. The first, which has been called Type-A1, is depicted in Figures 4.7a through 4.7c. The motivation here was to preserve the moments available to the point of interest in the Cartesian directions. The second, Type-A2 (Figures 4.7d through 4.7f) consigns all moments to the least squares space except those at the point of interest. The third, Type-A3 (Figures 4.7g through 4.7i), treats all moments by least squares: all moments have equal influence.



(a) Type-A1, $p = 3$ **(b)** Type-A1, $p = 5$ **(c)** Type-A1, $p = 7$

(d) Type-A2, $p = 3$ **(e)** Type-A2, $p = 5$ **(f)** Type-A2, $p = 7$

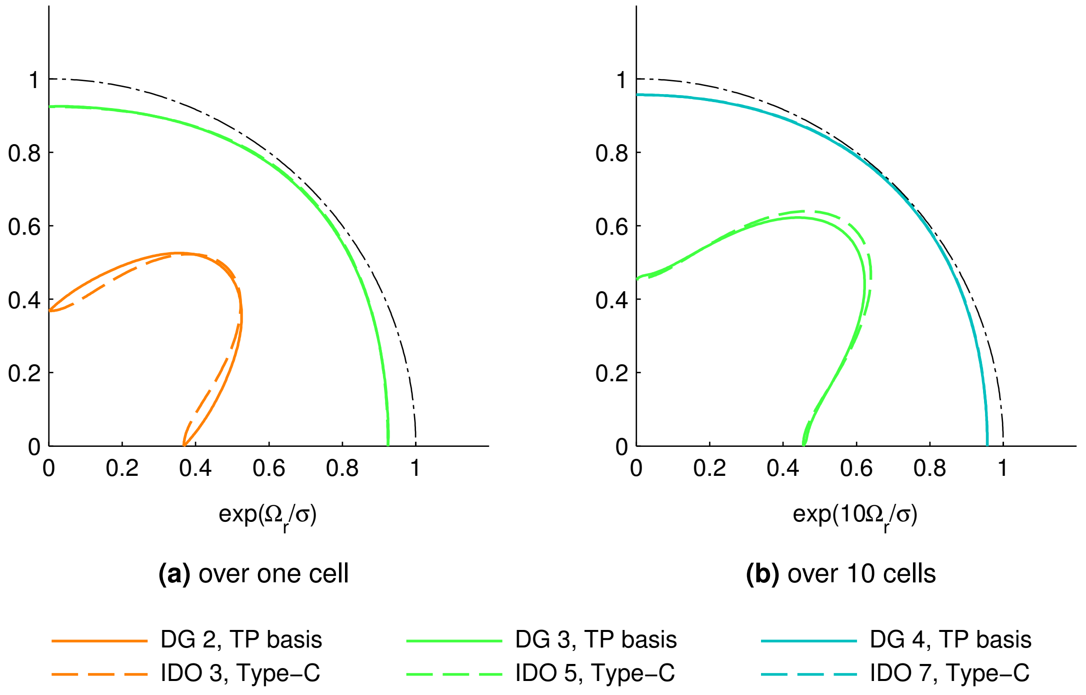
(g) Type-A3, $p = 3$ **(h)** Type-A3, $p = 5$ **(i)** Type-A3, $p = 7$

Figure 4.7: Alternative Type-A stencils. Moments in red are consigned to the least squares space.

4.3.2 Results

Polar dissipation plots for the tensor product and order-complete bases are shown in Figures 4.8 and 4.9 respectively. Plots for the Type-A IDO variations are also shown in Figure 4.10. In these plots, dissipation over 10 cells is included. This reproduces the results of Hu *et al.* (1999) and makes higher order contours more visible. Dispersion plots have been excluded since the high order contours are difficult to distinguish.

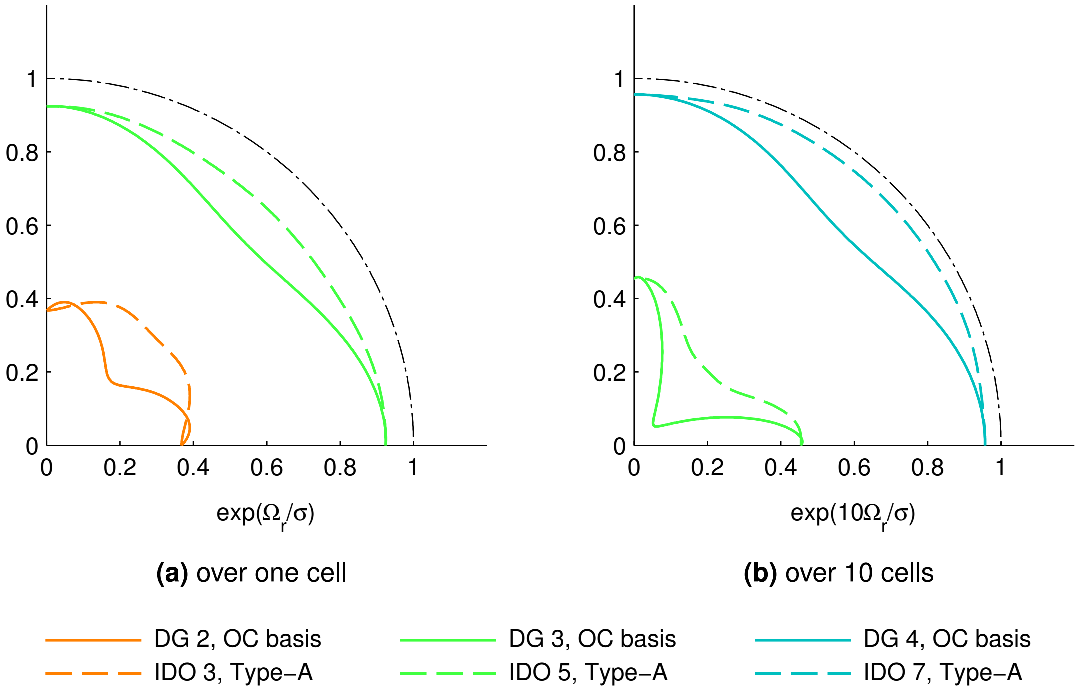
During the analyses it was found that some of the spurious modes and even physical modes of the Type-A IDO variations had positive real eigenvalue parts (a discussion on physical versus spurious modes is given on p. 62). These can be identified in Figure 4.11, which shows spectral detail at three propagation angles. The original Type-A IDO, Type-C IDO and DG schemes did not show real eigenvalue parts that were significantly greater than machine rounding error. The largest real part was found to be 1.17×10^{-13} for Type-A IDO 7 when K was stepped through by $\pi/100$.



(a) over one cell

(b) over 10 cells

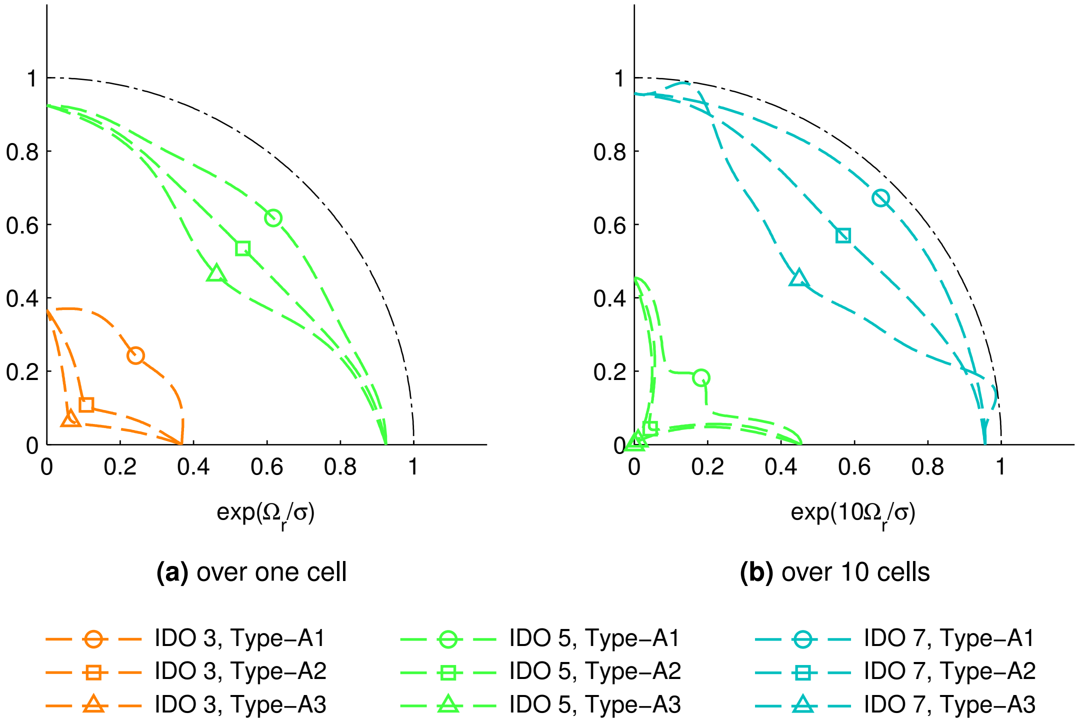
Figure 4.8: Dissipation factor for Type-C IDO and tensor product-based discontinuous Galerkin schemes at $K = \pi$.



(a) over one cell

(b) over 10 cells

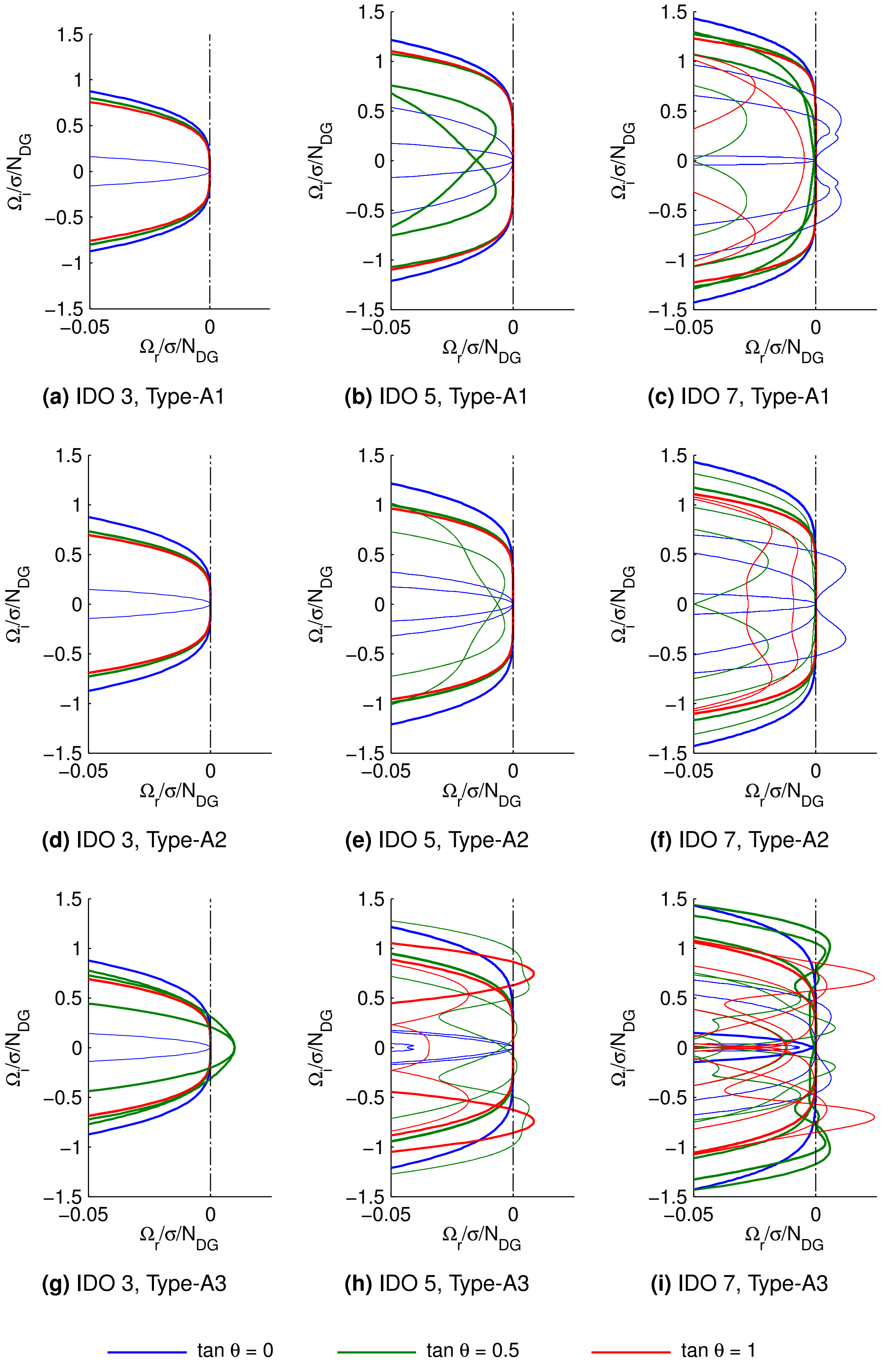
Figure 4.9: Dissipation factor for Type-A IDO and order-complete-based discontinuous Galerkin schemes at $K = \pi$.



(a) over one cell

(b) over 10 cells

Figure 4.10: Dissipation factor for variations of Type-A IDO at $K = \pi$.



(a) IDO 3, Type-A1

(b) IDO 5, Type-A1

(c) IDO 7, Type-A1

(d) IDO 3, Type-A2

(e) IDO 5, Type-A2

(f) IDO 7, Type-A2

(g) IDO 3, Type-A3

(h) IDO 5, Type-A3

(i) IDO 7, Type-A3

Figure 4.11: Normalised spectra of Type-A IDO variations, showing detail near the origin. Physical modes are represented by bold lines, spurious modes by thin lines.

4.3.3 Discussion

Against the hypothesis, Type-C spectra do not exactly match the corresponding DG tensor product-based spectra in oblique propagation directions. The contours are very close, however. They confirm that a tensor product basis will tend to increase accuracy anisotropically. The slight mismatch could be attributed to DG's piecewise-discontinuous reconstruction taking information from two upwind cells in the x- and y-directions, compared with IDO's piecewise-continuous reconstruction which takes information from three upwind points. In may be recalled that in one dimension, either method only needs upwind information from one grid element. The differing piecewise continuities may also help to explain why DG's contours do not intersect the Cartesian axes at a perpendicular angle compared with IDO's contours.

Further work might involve analysing two-dimensional MCV schemes, which are piecewise-discontinuous and therefore stand a better chance of matching the two-dimensional DG analysis. The relevant formulations are given in Ii and Xiao (2009). Hu *et al.* (1999) show how the flux vector splitting between the cells may be represented in the analysis. However, analysis of MCV schemes would detract from the focus of this thesis, and besides, it is not obvious how to adapt MCV to the order-complete basis from its native tensor product basis.

It light of the above, it is unsurprising that none of the Type-A variants' contours match those of the order-complete-based DG method. The closest appear to be those of Type-A2, not Type-A3 as hypothesised. The original Type-A and Type-A1 are the most isotropic, suggesting that it is favourable to bias information towards the point of interest. Indeed, when considering variations A1 through A3, it is seen that accuracy in the diagonal direction decreases as the influence of the upwind moments increases. The contours also lose perpendicularity with the Cartesian axes, which is consistent with the loss in C^0 continuity from least squares reconstruction.

An interesting finding is that a least squares reconstruction can threaten stability, at least in the Eulerian formulation. Unconstrained least squares in particular leads to physical modes having positive real eigenvalue parts. The implied unconditional instability is perhaps understandable, since differentiations of the profile at the point of interest are no longer consistent with the moments resident there. Constrained least squares reconstructions are more forgiving. For the cubic and quintic profiles, all real parts appear to be zero or negative. However, the septic profile shows wayward spurious modes. These modes can be expected to destroy the stability of the otherwise stable Eulerian scheme. It is unclear as to why the septic profile should exhibit unstable modes compared with lower degrees of polynomial, but it underscores the need for careful examination of the eigenvalues when designing a scheme.

Note that the existence of positive real eigenvalue parts does not necessarily cause unconditional instability in the semi-Lagrangian formulation. An example will be seen in Chapter 5.

4.4 Conclusions

It is evident that CIP/IDO is superior to DG when modelling smooth problems such as linear wave advection. In one dimension, when N moments or degrees of freedom are stored, the order of accuracy for CIP/IDO is $(2N-1)$ compared with N for DG, even though the two methods share the same spectral properties. This has important implications in two (or three) dimensions, where the number of moments that need to be stored increases quadratically (or cubically). However, the MCV scheme may be more appropriate for modelling nonsmooth problems, and although it inherits the same favourable stability as IDO, it is outperformed by DG in terms of accuracy for a given number of stored moments.

The spectral equivalence of IDO and DG does not hold in multidimensions, but the tensor product-based methods are still similar in their spectra. The order-complete-based Type-A IDO spectrum actually has superior isotropy to the corresponding DG spectrum. Also, the Type-A stencil is non-unique and can be adjusted with constrained least squares fitting. Accuracy in the diagonal direction and hence isotropy decreases as the influence of the upwind moments increases, which is worth keeping in mind for scheme design. However, least squares fitting, especially when unconstrained, can lead to positive real eigenvalue parts destroying stability in the Eulerian formulation.

Chapter 5

Comparison of Third Order Schemes

5.1 Introduction

In this chapter we compare existing Cartesian multi-moment schemes in two dimensions. It may be recalled that for a given multi-moment method, several multidimensional implementations may exist. An implementation may be split or unsplit; it may economise on moment storage by mixing in other operators or minimising the number of coefficients in the profile; it may uphold conservation of mass; and it may be semi-Lagrangian, Eulerian, or a hybrid between the two formulations.

The most important common factor is the order of accuracy. In the interest of conducting a fair study, only schemes of the same order of accuracy are considered here. Third order is the natural choice, because all Cartesian multi-moment schemes in the literature are introduced as such. Most of these schemes – variants of the constrained interpolation profile method (CIP) and CIP conservative semi-Lagrangian method (CIP–CSL) – were listed in Table 2.2. Of these, we can assume that the split CIP–CSL2 scheme of Nakamura *et al.* (2001) and the unsplit scheme of Takizawa *et al.* (2002) are one and the same from a semi-discrete point of view. The equivalence of split schemes with uniform operators and unsplit, tensor product-based schemes was observed in §2.2.2 and demonstrated in §3.4.1. There is another third order Cartesian scheme not included in that table, which is the scheme of Chen *et al.* (2011). This is a conservative, unsplit scheme with a biquadratic profile. The authors cite CIP–CSL2 as their underlying method, but their scheme's similarity to the triangular cell-based scheme of Ii *et al.* (2005) – point values which are advanced by CIP, a cell-integrated average which is advanced by the finite volume method – makes it more accurately classified as CIP/MM–FVM.

With respect to the advection formulation, it has been shown that conversion to an Eulerian formulation is appropriate for comparing spatial properties. CIP-based schemes become IDO-based, and we should adjust their names accordingly. CIP/MM–FVM2 naturally becomes IDO/MM–FVM2. In one dimension, CIP–CSL2 was previously renamed IDO–FVM2, but in multidimensions this is not very descriptive. We can look to Xiao (2004) who introduces the volume- and surface-integrated average multi-moment method (VSIAM). VSIAM is actually a general framework for solving different components of the Navier–Stokes equations, but the name serves as an appropriate description of the space discretisation. Therefore we shall rename the CIP–CSL schemes the IDO/VSIAM schemes. Also, the CIP–CSL2 schemes of Nakamura *et al.* and of Xiao *et al.* (2006) differ in having uniform and mixed operators, respectively. We distinguish the latter as having a second order central, time evolution converting formula (SOC TEC). This formula will be explained in the next section.

Table 5.1 lists the six CIP-based schemes of interest and their IDO-based counterparts. Additionally, the storage cost is given in terms of number of moments per cell. This will be of interest after performance has been evaluated. The following sections document how the spectra of the IDO-based versions were analysed and then verified with time-domain tests. In the tests, the CIP-based versions were used, at small Courant numbers where appropriate, since they were faster to run.

Table 5.1: Third order CIP-based schemes and their IDO-based counterparts.

Original scheme	Eulerian version	Storage cost
Nonconservative		
CIP, Type-M (Takewaki and Yabe, 1987)	IDO, Type-M	3
CIP, Type-A (Yabe *et al.*, 1991)	IDO, Type-A	3
CIP, Type-C (Aoki, 1995)	IDO, Type-C	4
Conservative		
CIP–CSL2 (Nakamura *et al.*, 2001)	IDO/VSIAM2	4
CIP–CSL2 (Xiao *et al.*, 2006)	IDO/VSIAM2 (SOC TEC)	3
CIP/MM–FVM2 (Chen *et al.*, 2011)	IDO/MM–FVM2	4

5.2 Formulations

This section describes how the sets of matrices $\mathbf{W}^{x}_{\alpha,\beta}$ and $\mathbf{W}^{y}_{\alpha,\beta}$ are arrived at for each scheme, with the exception of Type-A, which was described in Chapter 4. The presentation of the actual matrices is deferred to Appendix A. The moments will be ordered by descending dimensionality of the associated grid element, then by ascending order of spatial derivative. In other words, the ordering is $\overline{^{V}u}, \overline{^{S^x}u}, \overline{^{S^y}u}, u, u_{\xi}, u_{\eta}, u_{\xi\eta}$.

Type-M IDO

Like Type-A, the Type-M scheme stores u, u_ξ and u_η. Type-M is a split scheme, however. In the x-direction, the 1D IDO operator evolves u and u_ξ, and in the y-direction, it evolves u and u_η. But u_η also needs to be evolved in the x-direction, and u_ξ needs to be evolved in the x-direction. So a first order upwind operator is applied. The fully discrete system of equations reads

$$\begin{aligned}
u^*_{00} &= \text{IDO}\left(u^n_{00}, u^n_{\xi 00}, u^n_{\bar{1}0}, u^n_{\xi\bar{1}0}\right)\\
u^*_{\xi 00} &= \text{IDO}'\left(u^n_{00}, u^n_{\xi 00}, u^n_{\bar{1}0}, u^n_{\xi\bar{1}0}\right)\\
u^*_{\eta 00} &= \text{FOU}\left(u^n_{\eta 00}, u^n_{\eta 0\bar{1}}\right)
\end{aligned} \tag{5.1a}$$

$$\begin{aligned}
u^{n+1}_{00} &= \text{IDO}\left(u^*_{00}, u^*_{\eta 00}, u^*_{\bar{1}0}, u^*_{\eta\bar{1}0}\right)\\
u^{n+1}_{\xi 00} &= \text{FOU}\left(u^*_{\xi 00}, u^*_{\xi\bar{1}0}\right)\\
u^{n+1}_{\eta 00} &= \text{IDO}'\left(u^*_{00}, u^*_{\eta 00}, u^*_{0\bar{1}}, u^*_{\eta 0\bar{1}}\right)
\end{aligned} \tag{5.1b}$$

where the prime symbol ($'$) on the IDO operator indicates evolution of the point gradient, not the point value. The four IDO arguments correspond directly to the weights given in Equations (3.64) and (3.65), while the two FOU arguments correspond to weights of 1 and -1 derived in §3.2.1. These weights can be mapped to the sets of matrices $\mathbf{W}^x_{\alpha,\beta}$ and $\mathbf{W}^y_{\alpha,\beta}$ according to Equations (5.1a) and (5.1a), respectively.

Type-C IDO

Type-C IDO was presented in an unsplit, tensor product-based form in Chapter 4, but here it is presented in its original form as a split scheme with uniform IDO operators. It is reminded that $u_{\xi\eta}$ must be stored in addition to u, u_ξ and u_η, making this more expensive in terms of memory than Type-M and Type-A. The fully discrete system of equations reads

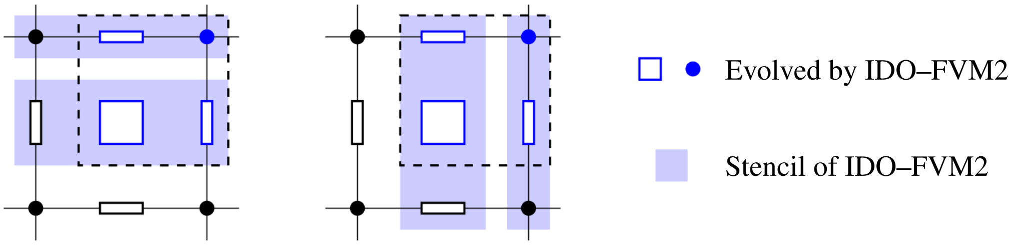
$$\begin{aligned}
u^*_{00} &= \text{IDO}\left(u^n_{00}, u^n_{\xi 00}, u^n_{\bar{1}0}, u^n_{\xi\bar{1}0}\right)\\
u^*_{\xi 00} &= \text{IDO}'\left(u^n_{00}, u^n_{\xi 00}, u^n_{\bar{1}0}, u^n_{\xi\bar{1}0}\right)\\
u^*_{\eta 00} &= \text{IDO}\left(u^n_{\eta 00}, u^n_{\xi\eta 00}, u^n_{\eta\bar{1}0}, u^n_{\xi\eta\bar{1}0}\right)\\
u^*_{\xi\eta 00} &= \text{IDO}'\left(u^n_{\eta 00}, u^n_{\xi\eta 00}, u^n_{\eta\bar{1}0}, u^n_{\xi\eta\bar{1}0}\right)
\end{aligned} \tag{5.2a}$$

$$\begin{aligned}
u^{n+1}_{00} &= \text{IDO}\left(u^*_{00}, u^*_{\eta 00}, u^*_{\bar{1}0}, u^*_{\eta\bar{1}0}\right)\\
u^{n+1}_{\xi 00} &= \text{IDO}\left(u^*_{\xi 00}, u^*_{\xi\eta 00}, u^*_{\xi\bar{1}0}, u^*_{\xi\eta\bar{1}0}\right)\\
u^{n+1}_{\eta 00} &= \text{IDO}'\left(u^*_{00}, u^*_{\eta 00}, u^*_{0\bar{1}}, u^*_{\eta 0\bar{1}}\right)\\
u^{n+1}_{\xi\eta 00} &= \text{IDO}'\left(u^*_{\xi 00}, u^*_{\xi\eta 00}, u^*_{\xi 0\bar{1}}, u^*_{\xi\eta 0\bar{1}}\right)
\end{aligned} \tag{5.2b}$$

which leads to a more regular mapping of weights compared to Type-M IDO. The weight matrices are confirmed to be identical to those of the unsplit form presented in §4.3.1.

IDO/VSIAM2

This is also a split scheme with uniform operators, but the stored moments are now $\overline{{}^{V}u}$, $\overline{{}^{S^x}u}$, $\overline{{}^{S^y}u}$ and u, and the 1D operator is IDO–FVM2. In each direction, the operator evolves (i) point values and longitudinal line values, identical to the 1D situation, and (ii) lateral line values and cell values, as though the 1D grid elements had been extruded into the second dimension. The operations are illustrated in Figure 5.1. The resulting weights have the same pattern as Type-C CIP/IDO, albeit with fewer nonzero terms.



(a) x-direction **(b)** y-direction

Figure 5.1: Evolution of moments in CIP–CSL2 or IDO/VSIAM2. The dashed box indicates the moments being evolved, i.e. moments subscripted ‘00’ according to the notation of §3.5.1. The wind direction is towards the right in (a) and upwards in (b).

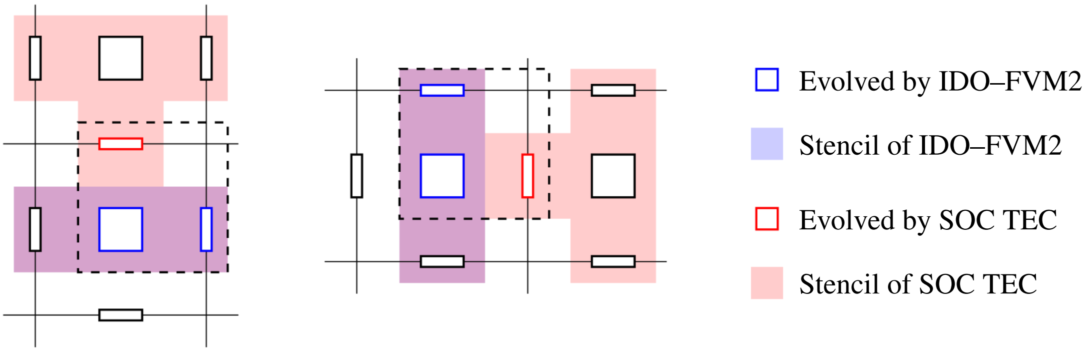
IDO/VSIAM2 with SOC TEC

In an effort to make 2D CIP-CSL2 (analogous to IDO/VSIAM2) as memory-efficient as Type-M and Type-A CIP, Xiao *et al.* (2006) remove the point values and have the CIP-CSL2 operator only evolve the lateral line values and cell values. The longitudinal line values are updated by a ‘time evolution converting’ (TEC) formula. This formula does a spatial interpolation of the rate of change of moments in time. Essentially it is an interpolation of space operators in a semi-discrete sense. The second order formula adopted by Xiao *et al.* reads

$$\frac{\partial}{\partial t}\overline{{}^{S^y}u}_{00} = \frac{1}{2}\left(\frac{\partial}{\partial t}\overline{{}^{V}u}_{00} + \frac{\partial}{\partial t}\overline{{}^{V}u}_{01}\right) \qquad (5.3a)$$

$$\frac{\partial}{\partial t}\overline{{}^{S^x}u}_{00} = \frac{1}{2}\left(\frac{\partial}{\partial t}\overline{{}^{V}u}_{00} + \frac{\partial}{\partial t}\overline{{}^{V}u}_{10}\right) \qquad (5.3b)$$

which means the stencil has been extended to encompass the cell values immediately downwind. In an Eulerian context, the TEC operator is represented by the ${}^{V}\mathbf{w}$ weights of the IDO–FVM2 operator divided between neighbouring stencils. Because of its resemblance to the second order central finite volume scheme, we shall refer to this operator as ‘SOC TEC’. Superposed is the original IDO–FVM2 operator, which is represented by a single set of ${}^{V}\mathbf{w}$ and ${}^{S}\mathbf{w}$ weights as before. Figure 5.2 illustrates the evolution of the moments by the two operators.



(a) x-direction **(b)** y-direction

Figure 5.2: Evolution of moments in CIP–CSL2 or IDO/VSIAM2 with second order central time evolution converting formula (SOC TEC). The dashed box indicates the moments being evolved, i.e. moments subscripted '00' according to the notation of §3.5.1. The wind direction is towards the right in (a) and upwards in (b).

IDO/MM–FVM2

Chen *et al.* (2011) propose a scheme comparable to the point value-based CIP/MM–FVM of Ii *et al.* (2005). However, they use a degree-2 tensor product basis conforming to a Cartesian cell, rather than a degree-3 order-complete basis conforming to a simplex.

$$\mathbf{r} = \begin{pmatrix} 1 & \xi & \xi^2 & \eta & \xi\eta & \xi^2\eta & \eta^2 & \xi\eta^2 & \xi^2\eta^2 \end{pmatrix} \tag{5.4}$$

Storing u on the vertices and edge midpoints, as shown in Figure 5.3, is sufficient to determine the nine profile coefficients.

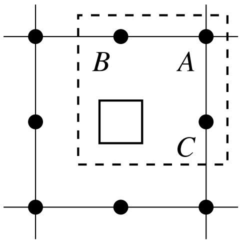


Figure 5.3: Layout of moments in CIP/MM–FVM2 or IDO/MM–FVM2. The dashed box indicates the moments being evolved, i.e. moments subscripted '00' according to the notation of §3.5.1. The wind direction is towards the right and upwards. In addition to vertex values (A), there are now values at the edge midpoints (B and C).

The Vandermonde matrix $\mathbf{A}$ is made up of $\mathbf{r}$ evaluated at the eight points, plus the following row representing the cell-integrated average:

$$\mathbf{A}_V = \iint \mathbf{r}\,\mathrm{d}\xi\,\mathrm{d}\eta\Big|_{0,0} - \iint \mathbf{r}\,\mathrm{d}\xi\,\mathrm{d}\eta\Big|_{-1,0} - \iint \mathbf{r}\,\mathrm{d}\xi\,\mathrm{d}\eta\Big|_{0,-1} + \iint \mathbf{r}\,\mathrm{d}\xi\,\mathrm{d}\eta\Big|_{-1,-1} \tag{5.5}$$

Because the Fourier analysis requires moments to have periodicity, the vertex values, horizontal edge midpoint values and vertical edge midpoint values must be treated as three distinct moment types. We distinguish them and their weights with the superscripts A, B and C, respectively. The complete weights are

$$\begin{gathered}
{}^{V}\mathbf{w}^{x} = \left(\int \mathbf{r}\,\mathrm{d}\eta\Big|_{0,0} - \int \mathbf{r}\,\mathrm{d}\eta\Big|_{-1,0} - \int \mathbf{r}\,\mathrm{d}\eta\Big|_{0,-1} + \int \mathbf{r}\,\mathrm{d}\eta\Big|_{-1,-1}\right)\mathbf{A}^{-1} \\
{}^{A}\mathbf{w}^{x} = \frac{\partial \mathbf{r}}{\partial \xi}\Big|_{0,0}\mathbf{A}^{-1} \qquad {}^{B}\mathbf{w}^{x} = \frac{\partial \mathbf{r}}{\partial \xi}\Big|_{-\frac{1}{2},0}\mathbf{A}^{-1} \qquad {}^{C}\mathbf{w}^{x} = \frac{\partial \mathbf{r}}{\partial \xi}\Big|_{0,-\frac{1}{2}}\mathbf{A}^{-1}
\end{gathered} \tag{5.6a}$$

$$\begin{gathered}
{}^{V}\mathbf{w}^{y} = \left(\int \mathbf{r}\,\mathrm{d}\xi\Big|_{0,0} - \int \mathbf{r}\,\mathrm{d}\xi\Big|_{-1,0} - \int \mathbf{r}\,\mathrm{d}\xi\Big|_{0,-1} + \int \mathbf{r}\,\mathrm{d}\xi\Big|_{-1,-1}\right)\mathbf{A}^{-1} \\
{}^{A}\mathbf{w}^{y} = \frac{\partial \mathbf{r}}{\partial \eta}\Big|_{0,0}\mathbf{A}^{-1} \qquad {}^{B}\mathbf{w}^{y} = \frac{\partial \mathbf{r}}{\partial \eta}\Big|_{-\frac{1}{2},0}\mathbf{A}^{-1} \qquad {}^{C}\mathbf{w}^{y} = \frac{\partial \mathbf{r}}{\partial \eta}\Big|_{0,-\frac{1}{2}}\mathbf{A}^{-1}
\end{gathered} \tag{5.6b}$$

and the elements of the resulting row vectors correspond to $\overline{{}^{V}u_{00}}$, ${}^{A}u_{00}$, ${}^{A}u_{\bar{1}0}$, ${}^{A}u_{1\bar{1}}$, ${}^{A}u_{\bar{1}\bar{1}}$, ${}^{B}u_{00}$, ${}^{B}u_{0\bar{1}}$, ${}^{C}u_{00}$ and ${}^{C}u_{\bar{1}0}$ according to the ordering of the rows in $\mathbf{A}$.

5.3 Simulation Methods

To test the schemes, a solver was developed in MATLAB® R2009a. The solver framework allowed for any combination of vertex, line and cell values and their derivatives. Polymorphic (placeholder) components included the space discretisation, the advection formulation and the time discretisation, so it was possible to formulate a scheme to be either semi-Lagrangian or Eulerian and assign it an appropriate order of time integration.

The semi-Lagrangian formulation was preferred. The tested schemes were consequently CIP-based, not IDO-based. The three CIP variants and two CIP/VSIAM2 variants were included in tests. CIP/MM-FVM2 was omitted. It was reasoned that the gain in time-domain information would not be worth the extra work needed to incorporate midpoint values into the framework.

For each test case the exact solution was known and differentiable, so for the three CIP schemes it was possible to populate the points with an exact initial solution. For the CIP/VSIAM2 schemes, integrations of the solution over lines and rectangles had to be carried out. Here it was easier to adopt numerical quadrature than exact integration. Fourth order Gaussian quadrature was used. For each element, the exact solution was interpolated at the Gauss points shown in Figure 5.4 and averaged.

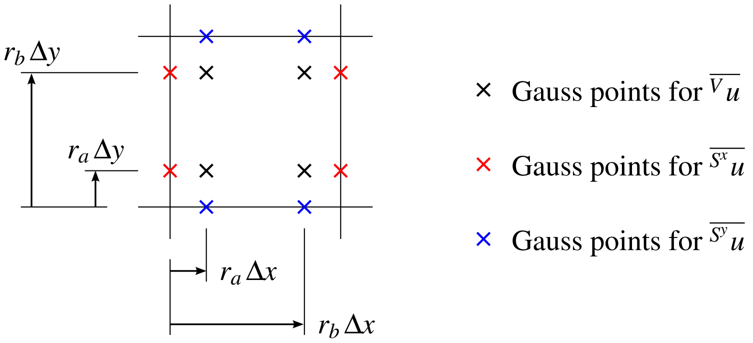

$r_b\Delta y$
$r_a\Delta y$
$r_a\Delta x$
$r_b\Delta x$
× Gauss points for $\overline{{}^{V}u}$
× Gauss points for $\overline{{}^{S^x}u}$
× Gauss points for $\overline{{}^{S^y}u}$

Figure 5.4: Fourth order Gaussian quadrature points for cells and faces in 2D. r_a and r_b are given by $1/2\,(1 \mp 1/\sqrt{3})$.

Linear advection tests, outlined in §3.4.2, were performed using a small reference Courant number of $\sigma_0 = 0.1$ so that the resulting errors could be compared with those derived from the corresponding spectra. The formal Courant number was $\sigma = \sigma_0 \cos\theta$, with θ being the wave propagation angle.

The schemes were then subjected to more demanding tests. Rotational velocity fields were of particular interest, because they exercised the schemes over the full range of propagation angles. If there were certain angles at which accuracy deteriorated, the deterioration would show up in the convergence study. The Courant number could be raised for these tests: $\sigma = \pi/4$ was selected.

The first such test case had a velocity field which was uniform in space but changed direction in time. This quasi-linear advection did not so much rotate the scalar profile as translate it in a circle about $(0, -2\pi)$. The profile was given by a bicosine function which enabled periodic boundary conditions to be exploited. The velocity and scalar fields were specified as

$$\begin{gathered}
\mathbf{a}(t) = \begin{pmatrix} 2\pi\cos t \\ -2\pi\sin t \end{pmatrix}, \quad u(x,y,t) = \tfrac{1}{4}\left[1+\cos\left(x - x_c(t)\right)\right]\left[1+\cos\left(y-y_c(t)\right)\right] \\
x_c(t) = 2\pi\sin t \\
y_c(t) = 2\pi\cos t
\end{gathered} \tag{5.7}$$

The second test case involved rotation of a scalar field describing some feature, which is standard in the literature for testing advection schemes. The Gaussian cone function of Xiu and Karniadakis (2001) provided a sufficiently smooth profile. The initial scalar field and constant velocity field, depicted in Figure 5.5, were specified as

$$\mathbf{a}(x,y) = \begin{pmatrix} y \\ -x \end{pmatrix}, \qquad u(x,y,t) = \exp\left(-\frac{[x - x_c(t)]^2 + [y - y_c(t)]^2}{2\lambda^2} \right)$$
$$x_c(t) = x_0 \cos t + y_0 \sin t$$
$$y_c(t) = -x_0 \sin t + y_0 \cos t \tag{5.8}$$

with $x_0 = -0.5$, $y_0 = 0$ and $\lambda = 1/8$. There was a question as to how to implement the boundary conditions, which could no longer be periodic. Since the purpose of the simulation was purely to check spatial convergence, it made sense to substitute the exact solution wherever the scheme attempted to take information from outside the domain. For the CIP/VSIAM2 schemes, Gaussian quadrature simplified the required integration over notional departure lines.

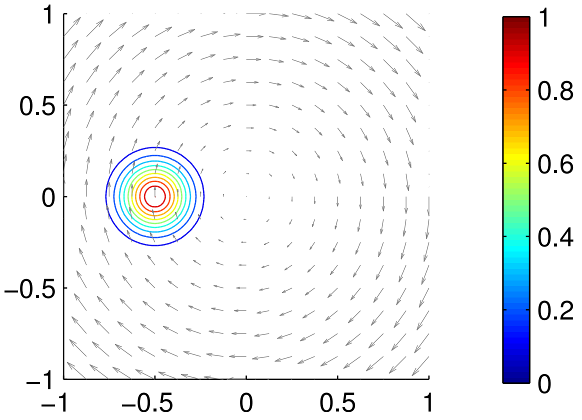


Figure 5.5: Gaussian cone rotation

More complex test cases can be found in the literature. A particularly rigorous test is frontogenesis associated with an idealised vortex (Doswell III, 1984), which was used by Nakamura *et al.* (2001). Here relative scheme performance becomes more pronounced as dissipative schemes fail to resolve the shape of the spiral. However, this test required high grid resolution before any convergence was seen, and it did not offer much more insight into the space scheme characteristics. It was not added to the test suite.

Rotations of features with sharply defined edges, such as Zalesak's slotted disk (see e.g. Yabe *et al.*, 1991; Xiao *et al.*, 1996b), were also of little use. The presence of discontinuities in the solution leads to oscillations which do not decay with increased grid resolution – a manifestation of Gibb's phenomenon. Without nonlinear devices such as slope limiters to curb these oscillations, convergence is not achieved. The focus of this thesis is on linear schemes, therefore such cases were not considered further.

5.4 Results

Results are presented in two parts. The first part concerns the spectral analysis of the semi-discretised systems, while the second part concerns time-domain performance. Not all the results in the second part are simulation results: $||\varepsilon(K)||^{\text{phys}}$ from the spectra is included for the purpose of verifying linear advection behaviour, and $||\varepsilon(K)||^{\text{phys,iso}}$ is used to infer isotropic performance.

5.4.1 Semi-Discrete Analysis

In analysing the weight matrices associated with the six space discretisations, it was found that Type-C IDO and IDO/VSIAM2 have identical spectra. In the plots that follow, their curves have been assigned the same colour. Spurious modes have not been plotted for any of the spectra (a discussion on this point is given on p. 62).

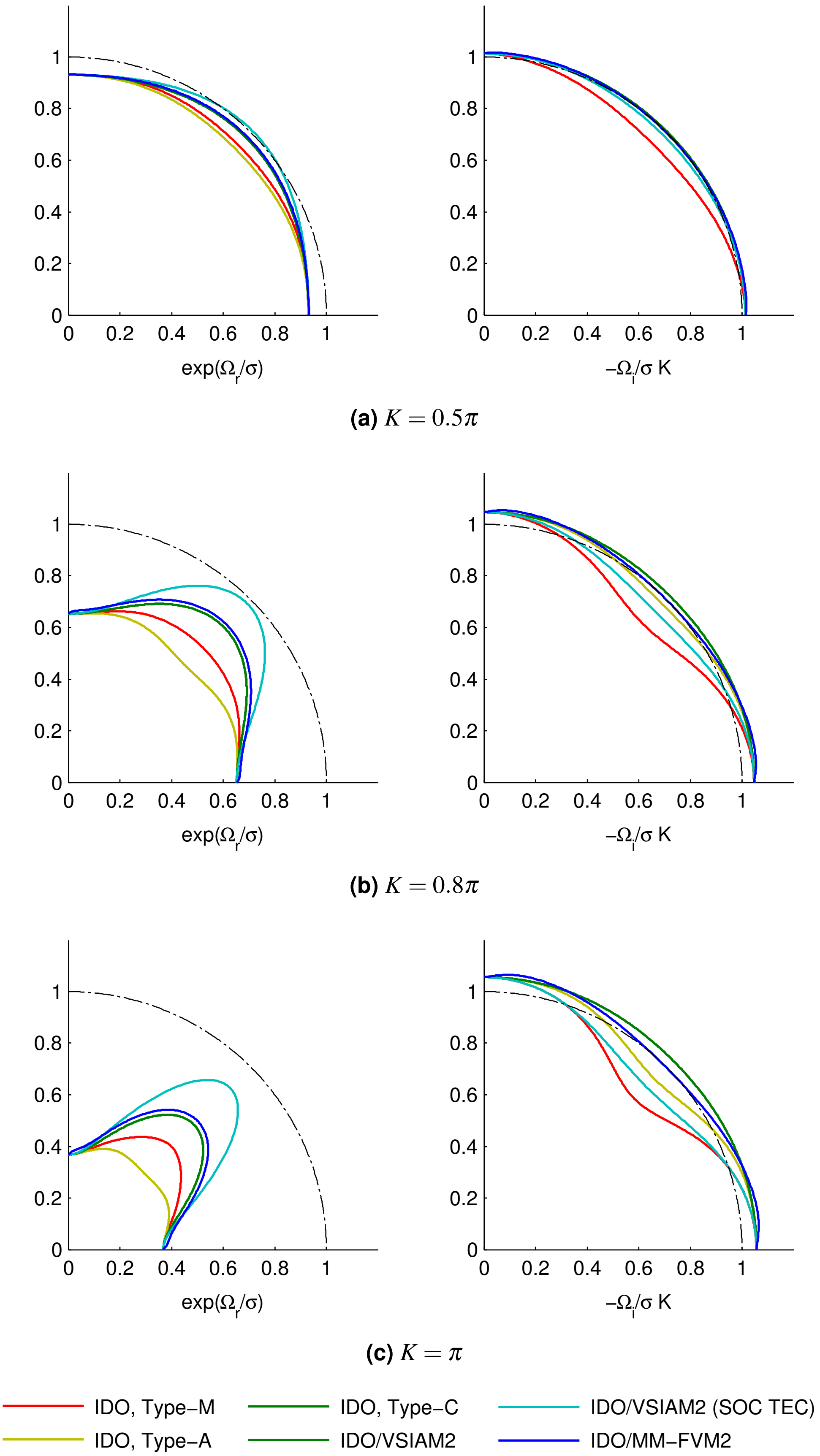
The schemes' relative anisotropy is best conveyed in polar plots. Figure 5.6 shows polar plots for $K = 0.5\pi$ and above. For smaller wavenumbers, relative performance of the three schemes becomes difficult to see in polar form, and we must turn to the spectra at fixed propagation angles.

Figure 5.7 shows the spectra plotted on the complex Ω/σ plane. The 1D IDO spectrum is included to show how the 2D scheme spectra deviate from the isotropic ideal as θ increases. Detail around the origin is also shown, highlighting the existence of positive real parts of the IDO/VSIAM2 (SOC TEC) eigenvalues when $\theta = 45^\circ$.

Dissipation and dispersion characteristics are shown in Figures 5.8 and 5.9. To distinguish the dispersive behaviour of the schemes at low wavenumbers, the usual plotting of $\Omega_i(K)$ is omitted; $-(\Omega_i(K) + K)$ has been plotted instead. This means that the errors of the eigenvalues with the exact dispersion relation, $-\mathrm{I}K$, are represented on the vertical axis and can be zoomed in on over a wide range of K. There is a similar zoom on the vertical axis for $\Omega_r(K)$, which represents the dissipation error.

Table 5.2 presents the calculated rates of convergence, and hence orders of accuracy, for the two oblique propagation angles. Table 5.3 presents the K_c metric of Hu *et al.* (1999), and Table 5.4 presents the equivalent measure of isotropy K_c^{iso}. Although the latter is the more appropriate of the two measures when comparing same-order multidimensional schemes, it is useful to see how K_c compares with our other measure of performance, $||\varepsilon(K)||^{\text{phys}}$, which is plotted in the next subsection.

Since the measure of a scheme's isotropy should be succinct, Table 5.4 corresponds to the strongest anisotropic direction only, $\theta = 45^\circ$.

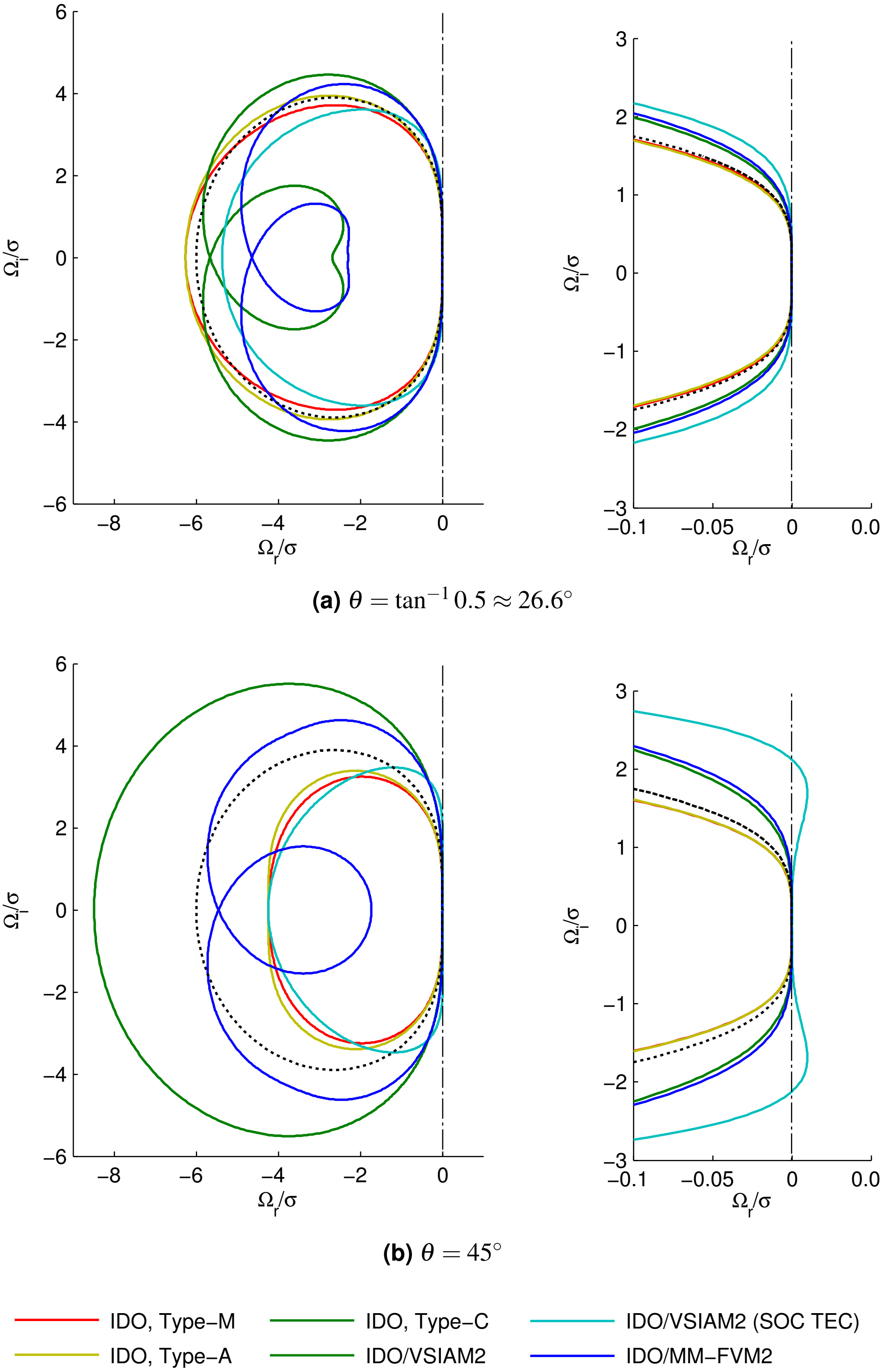


(a) $K = 0.5\pi$

(b) $K = 0.8\pi$

(c) $K = \pi$

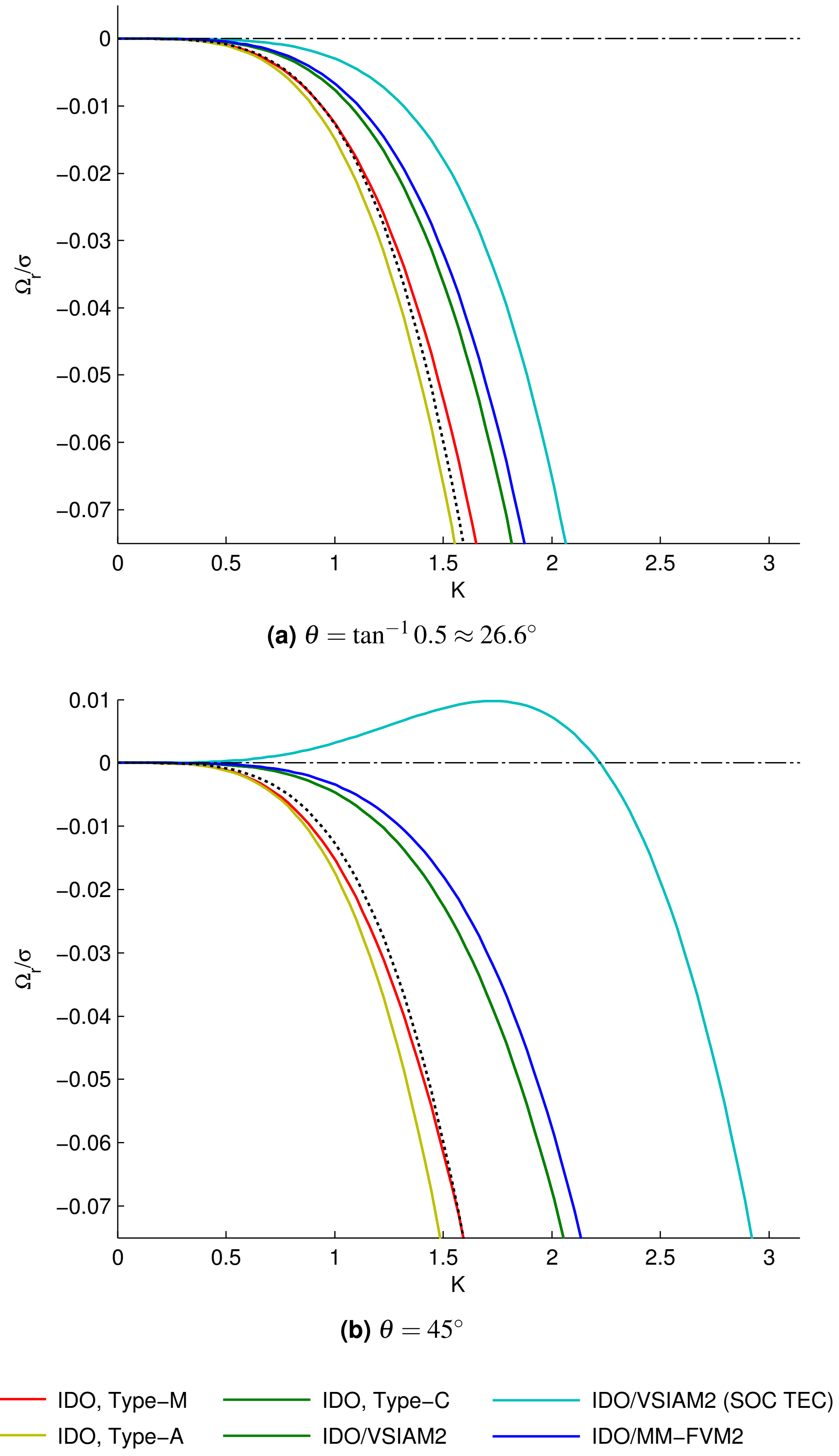
Figure 5.6: Polar plots for third order multi-moment schemes, showing (left) dissipation factor and (right) normalised phase speed.



(a) $\theta = \tan^{-1} 0.5 \approx 26.6^\circ$

(b) $\theta = 45^\circ$

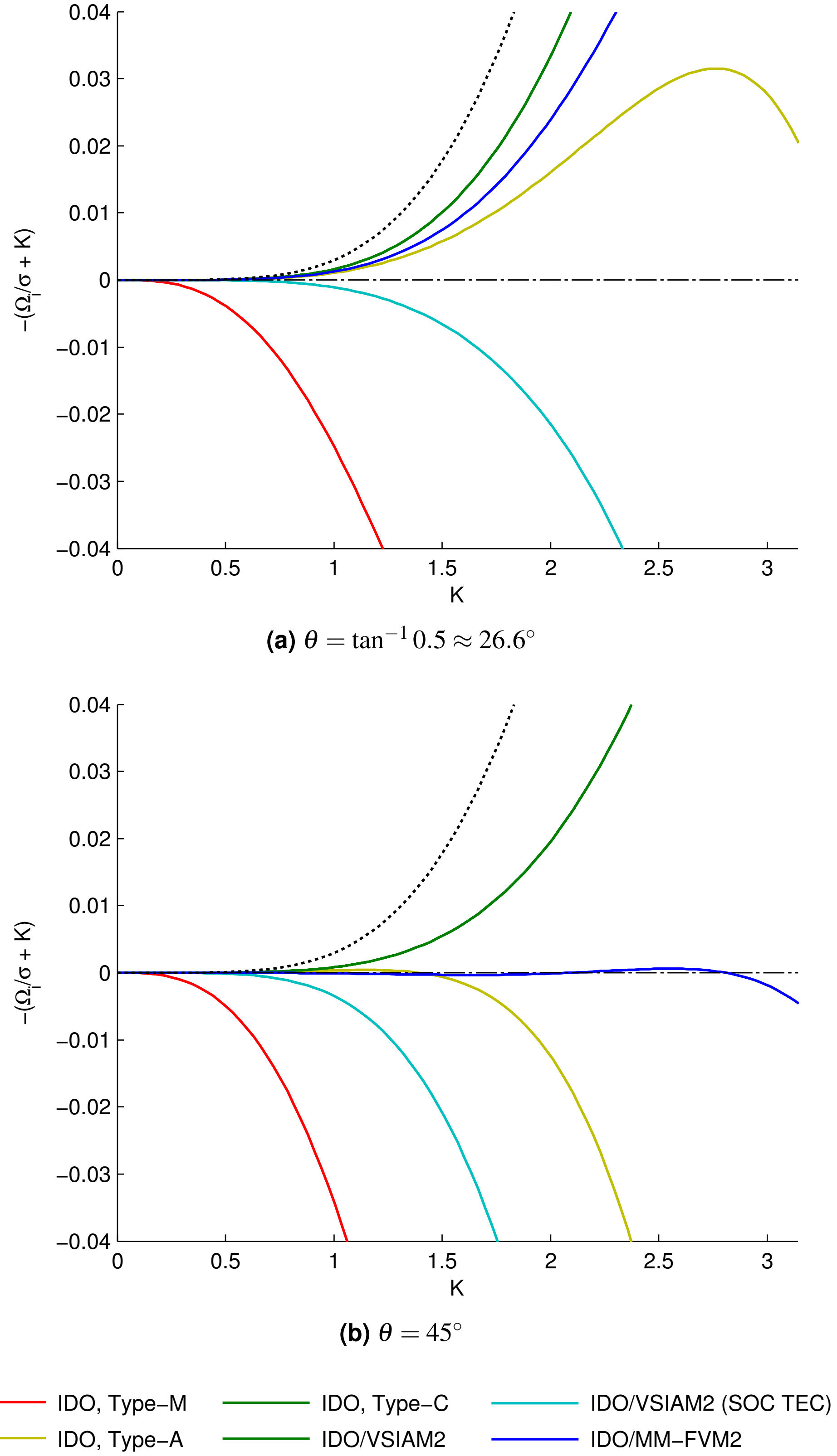
Figure 5.7: Spectra of third order multi-moment schemes plotted on the complex plane for two propagation angles. (Left) full range of K; (right) detail near the origin. The isotropic ideal is shown as a dotted trace.



(a) $\theta = \tan^{-1} 0.5 \approx 26.6^\circ$

(b) $\theta = 45^\circ$

Figure 5.8: Dissipation error of third order multi-moment schemes for two propagation angles. The isotropic ideal is shown as a dotted curve.



(a) $\theta = \tan^{-1} 0.5 \approx 26.6^\circ$

(b) $\theta = 45^\circ$

Figure 5.9: Dispersion error of third order multi-moment schemes for two propagation angles. The isotropic ideal is shown as a dotted curve.

Table 5.2: Orders of accuracy as calculated from Equation (3.29) with $K_a = \pi/25$.

θ	Scheme	Convergence rate		
		Dissipation	Dispersion	Overall
$\tan^{-1} 0.5$	IDO, Type-M	2.9952	1.9943	2
$\approx 26.6°$	IDO, Type-A	2.9972	3.9936	3
	IDO, Type-C	2.9985	3.9967	3
	IDO/VSIAM2	2.9985	3.9967	3
	IDO/VSIAM2 (SOC TEC)	3.0065	3.9954	3
	IDO/MM–FVM2	2.9985	3.9957	3
45°	IDO, Type-M	2.9948	1.9958	2
	IDO, Type-A	2.9976	3.9854	3
	IDO, Type-C	2.9991	3.9979	3
	IDO/VSIAM2	2.9991	3.9979	3
	IDO/VSIAM2 (SOC TEC)	2.9926	3.9961	3
	IDO/MM–FVM2	3.0010	3.9910	3

Table 5.3: Measures of resolution as calculated from Equation (2.5). For reference, the resolution of 1D IDO which represents the isotropic ideal is 0.7863 in dissipation and 1.1220 in dispersion, i.e. 0.79 overall.

θ	Scheme	K_c		
		Dissipation	Dispersion	Overall
$\tan^{-1} 0.5$	IDO, Type-M	0.7800	0.5488	0.55
$\approx 26.6°$	IDO, Type-A	0.7504	1.4471	0.75
	IDO, Type-C	0.8972	1.2805	0.90
	IDO/VSIAM2	0.8972	1.2805	0.90
	IDO/VSIAM2 (SOC TEC)	1.1278	1.4067	1.13
	IDO/MM-FVM2	0.9296	1.3664	0.93
45°	IDO, Type-M	0.7373	0.5030	0.50
	IDO, Type-A	0.7225	1.7875	0.72
	IDO, Type-C	1.0175	1.4700	1.02
	IDO/VSIAM2	1.0175	1.4700	1.02
	IDO/VSIAM2 (SOC TEC)	1.1711	1.0840	1.08
	IDO/MM-FVM2	1.0973	3.1611	1.10

Table 5.4: Measures of isotropy as calculated from Equation (3.94) at $\theta = 45°$.

Scheme	K_c^{iso}		
	Dissipation	Dispersion	Overall
IDO, Type-M	1.7254	0.4992	0.50
IDO, Type-A	1.0200	1.1414	1.02
IDO, Type-C	0.8840	1.2113	0.88
IDO/VSIAM2	0.8840	1.2113	0.88
IDO/VSIAM2 (SOC TEC)	0.7360	0.9473	0.74
IDO/MM-FVM2	0.8501	1.1149	0.85

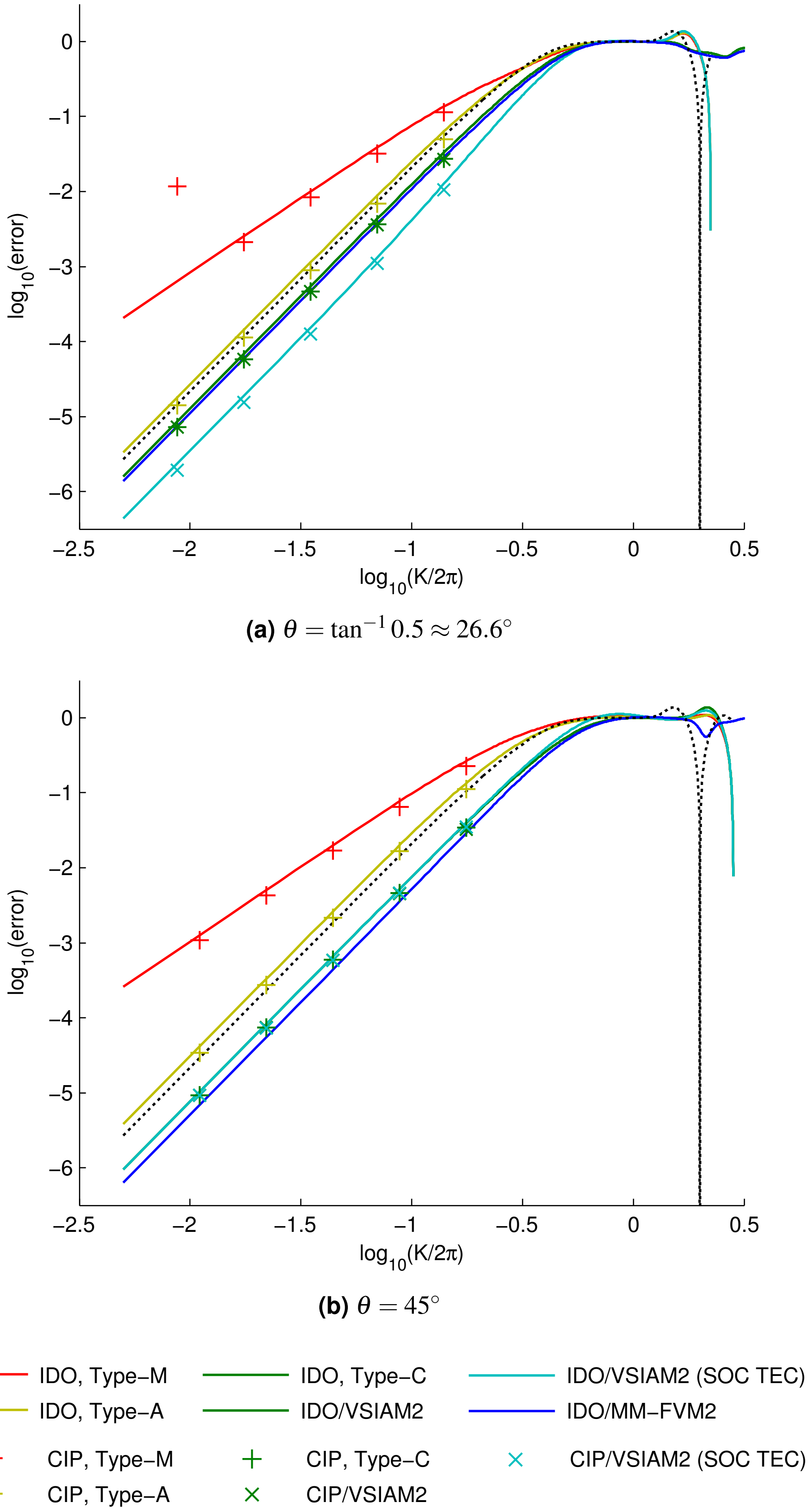
5.4.2 Performance in the Time Domain

Maximum absolute errors from the linear advection tests are plotted in Figure 5.10. Curves of $||\varepsilon(K)||^{\mathrm{phys}}$ derived from the spectra are included for comparison. In Figure 5.11, curves of $||\varepsilon(K)||^{\mathrm{phys,iso}}$ are plotted alone since, as discussed in §3.4.2, it is not possible to get the equivalent isotropic errors from simulations. The latter corresponds to the strongest anisotropic direction only, $\theta = 45°$.

Table 5.5 summarises convergence rates of the square roots of dissipation and dispersion errors defined by Takacs (1985). The second finest grid was selected since the rates begin to deteriorate for the finest grid. For brevity, only two test cases are shown. Linear advection convergence rates at $\theta = \tan^{-1} 0.5 \approx 26.6°$ were similar to those at $\theta = 45°$, and circulation and rotation convergence rates were similar to those of time-varying advection.

Rates of convergence are summarised in Table 5.6, where the rate is calculated from errors on the finest two grids – that is, the grids with $m_x = 64$ and $m_x = 128$. m_x is the number of cells, or number of points less one, in the x-direction. Some of the schemes exhibited divergence at the finest grid resolutions, in which case the next best grid pairing is used. The linear advection test cases are omitted from this table since their rates of convergence are evident from Figure 5.10.

Full details of the errors and their convergence are included in Appendix B.



(a) $\theta = \tan^{-1} 0.5 \approx 26.6^\circ$

(b) $\theta = 45^\circ$

Figure 5.10: 2D linear advection errors with (lines) physical-mode error norm $\|\varepsilon\,(K)\,\|^{\text{phys}}$, computed from spectra, and (crosses) $\|\varepsilon\|_\infty$, from simulation results. The isotropic ideal is shown as a dotted trace. At $\theta = 45^\circ$, for most of K, IDO/VSIAM2 (SOC TEC) is indistinguishable from Type-C IDO and ISO/VSIAM2.

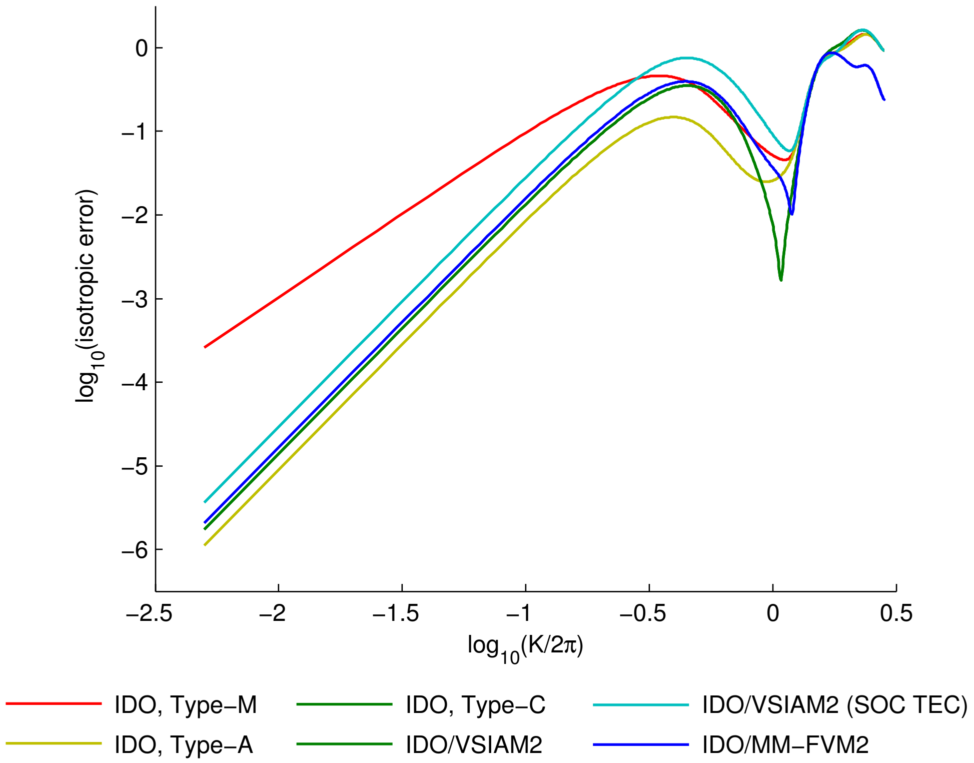


Figure 5.11: Norm of the physical-mode error with the isotropic ideal, or $\|\varepsilon(K)\|^{\mathrm{phys,iso}}$, computed from spectra.

Table 5.5: Summary of convergence rates of dissipation and dispersion errors (Takacs, 1985). $m_x = 64$.

Test case	Scheme	$\sqrt{E_{diss}}$ rate	$\sqrt{E_{disp}}$ rate
Linear advection	CIP, Type-M	2.963	1.974
at $\theta = 45°$	CIP, Type-A	2.982	3.861
	CIP, Type-C	2.989	3.968
	CIP/VSIAM2	2.985	3.953
	CIP/VSIAM2 (SOC TEC)	2.956	3.969
Time-varying	CIP, Type-M	2.597	1.877
advection of a	CIP, Type-A	2.970	2.960
bicosine profile	CIP, Type-C	2.998	2.998
	CIP/VSIAM2	2.985	2.985
	CIP/VSIAM2 (SOC TEC)	3.010	3.125

Table 5.6: Summary of convergence rates of mean absolute errors. Some of the schemes exhibited divergence at the finest grid resolutions ($m_x = 128$), in which case the next best grid pairing is used ($m_x = 64$).

Test case	Scheme	$\|\|\varepsilon\|\|_1$ rate	$\|\|\varepsilon\|\|_\infty$ rate
Time-varying advection of a bicosine profile	CIP, Type-M ($m_x = 64$)	1.913	1.861
	CIP, Type-A ($m_x = 64$)	2.966	2.957
	CIP, Type-C	2.995	2.994
	CIP/VSIAM2	2.994	2.992
	CIP/VSIAM2 (SOC TEC)	3.002	3.004
Rotation of a Gaussian cone	CIP, Type-M	—	—
	CIP, Type-A	—	—
	CIP, Type-C	0.989	0.947
	CIP/VSIAM2	2.946	2.886
	CIP/VSIAM2 (SOC TEC)	2.968	2.969

5.5 Discussion

This section is also divided into two parts, with the first part discussing spatial properties associated with discretisation of the linear advection equation, and the second part discussing convergence issues in the more complex test cases.

5.5.1 Linear Advection

The first point to make is that the agreement between analysis and simulation results in Figure 5.10 lends confidence that the weight matrices have been formulated correctly for analysis, and that the schemes have been coded correctly for simulations. The only exception is in Type-M CIP at $\theta = \tan^{-1} 0.5$, where there is an anomalous error on the finest grid. It is speculated that the temporal error has become dominant, possibly as a result of the operator splitting.

It is also interesting that the correct orders of dissipation and dispersion accuracy are reflected in the convergence rates of $\sqrt{E_{\text{diss}}}$ and $\sqrt{E_{\text{disp}}}$ for all schemes.

The plots of the spectra fit in with observations already made in Chapters 2 and 3. It has already been pointed out that one-dimensional IDO and IDO–FVM2 have identical spectra. Since Type-C IDO and IDO/VSIAM2 consist, respectively, of these operators interleaved in two dimensions, it is unsurprising that their spectra should also be identical.

Furthermore, an equivalence was noted between split schemes with uniform operators and unsplit schemes with tensor product bases. Although the unsplit IDO/MM–FVM2 is not exactly equivalent to split Type-C IDO or IDO/VSIAM2, the similarities make sense. Specifically, the schemes can be seen to be more accurate in the 45° direction than in 0° and 90° directions. This effect was previously observed for the first order upwind scheme. The spectra are also bigger in the diagonal direction, which will give rise to a more stringent Eulerian stability condition (because the allowable Courant number is generally inversely proportional to spectrum size).

Conversely, schemes that use mixed operators or order-complete bases, which means they can use three types of moment instead of four, can be seen to reduce spectrum size but sacrifice accuracy in the 45° direction. This is favourable in the case of Type-A CIP, which becomes more isotropic as a result: according to both K_c^{iso} and $\|\varepsilon(K)\|^{\text{phys,iso}}$, it is the most isotropic of the six schemes. However, Type-M CIP's order of accuracy has been compromised in oblique directions. Clearly it has inherited the dispersion accuracy of the FOU operator, leaving it second order accurate overall.

This drop in the order of accuracy is confirmed by the simulations, although it is not explicitly acknowledged in the literature. Yabe *et al.* (2004), after performing tests with the Type-C and Type-M schemes, conclude:

> *"The third-order accuracy of the Type-C is as is expected because it uses the CIP procedure all through the process. Although the Type-M uses the first-order scheme (linear interpolation) in estimating the derivative in perpendicular direction, it gives the accuracy better than second-order."*
>
> – Yabe *et al.* (2004)

The remaining scheme, IDO/VSIAM2 with second order central time evolution conversion (SOC TEC), has a remarkable feature. Some of its eigenvalues have positive real parts, which in the Eulerian formulation would lead to unconditional instability. However, it is clear from the simulations that its semi-Lagrangian counterpart is both stable and accurate. Furthermore, the scheme contradicts the general pattern whereby having fewer moment types leads to sacrificed accuracy in the 45° direction. It is among the most accurate of the schemes in that direction, second only to IDO/MM–FVM2.

The potential of semi-Lagrangian schemes to overcome unconditional instability in their Eulerian counterparts is worth keeping in mind. One might speculate, for instance, that replacing the FOU operators in the Type-M CIP scheme with SOC TEC operators would restore the scheme's order of accuracy while preserving stability.

Lastly, there is encouraging correlation between K_c and $\|\varepsilon(K)\|^{\text{phys}}$ at low wavenumbers. Comparing Table 5.3 with Figure 5.10, it can be seen that the schemes are in the same rank. An exception is between IDO/VSIAM2 with SOC TEC and Type-C IDO (or IDO/VSIAM2) at $\theta = 45°$. IDO/VSIAM2 with SOC TEC marginally outperforms the other in K_c, whereas it is the other way round in $\|\varepsilon(K)\|^{\text{phys}}$. However, those schemes are barely distinguishable at either angle.

The same agreement in scheme ranking can be seen between K_c^{iso} and $\|\varepsilon(K)\|^{\text{phys,iso}}$. It may be concluded that we can choose either metric when quantifying scheme isotropy. K_c^{iso} is easier to quote, since an arbitrarily small K must be chosen when evaluating $\|\varepsilon(K)\|^{\text{phys,iso}}$.

5.5.2 Other Test Cases

The circular translation case, being quasi-linear, simply confirms that the overall orders of convergence observed in linear advection are maintained for all propagation directions. However, the rates of $\sqrt{E_{\text{diss}}}$ and $\sqrt{E_{\text{disp}}}$ no longer distinguish between the orders of dissipation and dispersion accuracy. This is not completely surprising when one considers that E_{disp} is based on the correlation coefficient between numerical and exact solutions: it is a measure of the displacement of features (e.g. peaks) in the solution. If the features undergo rotation – that is, if velocity changes direction in time or space – then representing the phase errors by displacement is perhaps questionable.

In the rotation test, there were unexpected deteriorations in convergence, with only the two CIP/VSIAM2 variants exhibiting the correct third order accuracy. The mean errors of Type-M and Type-A CIP diverged completely at high grid resolutions. This latter problem can be traced to instabilities that grew from the boundaries as shown in Figure 5.12. It is speculated that these instabilities were caused by small inconsistencies that developed between the interior numerical solution and boundary values, the latter of which were forced to take information from the exact solution in the absence of suitable boundary conditions.

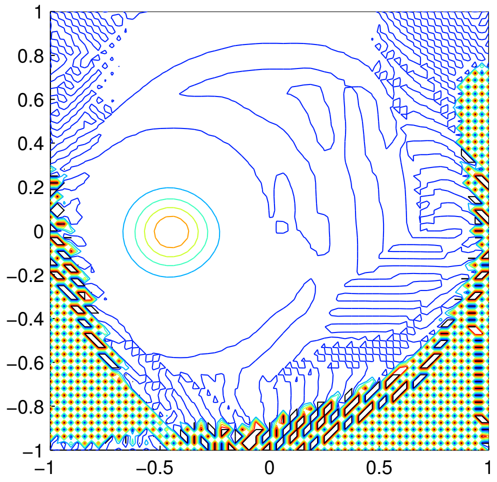


Figure 5.12: Instabilities arising from Gaussian cone rotation. This plot shows one revolution by Type-A CIP on the $m_x = 64$ grid. The oscillations growing from boundary are too large in magnitude to be rendered by the contours and hence appear as alternating red and blue dots.

Although the interior solution appears to remain intact in Figure 5.12, it might be further speculated that such inconsistencies nevertheless impair the global convergence rate. This would explain why Type-C CIP exhibited first-order accuracy. The VSIAM schemes are evidently immune to the phenomenon, protected perhaps by the cell faces implementing conservative reconstructions at the domain boundary. This is an important point: in spite of the adverse convergence rates of the nonconservative schemes, the VSIAM schemes have been proven capable of achieving nominal accuracy in nonlinear conditions.

5.6 Conclusions

The main findings are presented in the context of linear advection. Tests involving a rotational velocity field saw deteriorations to the nominal rates of convergence, possibly owing to boundary effects, whereas the linear advection tests straightforwardly verified the frequency-domain analyses. The main findings are as follows:

- The original multidimensional variant of CIP, Type-M (Takewaki and Yabe, 1987), is formally second order-accurate, despite claims that greater than second order accuracy is recovered (Yabe *et al.*, 2004).
- CIP–CSL2 with the second order central time evolution converting formula (Xiao *et al.*, 2006) unusually features positive real eigenvalues in its spectrum, and so one can expect its Eulerian formulation to be unconditionally unstable. However, the semi-Lagrangian formulation presented by Xiao *et al.* is stable and has high accuracy in diagonal propagation directions. It is the most attractive for being conservative and having a moment storage cost of $^3/_4$ that of the other conservative schemes.
- Type-A CIP (Yabe *et al.*, 1991) may be considered the next ‘best’ scheme in the sense of having the same low storage cost as CIP–CSL2 and remaining stable after conversion to Type-A IDO. This economy of storage is made possible by the scheme’s order-complete basis and high isotropy. However, Type-A CIP does not uphold conservation.

The results confirm observations made in Chapters 2 and 4 that there is a trade-off between accuracy and economy of stored moments. But the Type-A scheme shows that as long as accuracy is not affected in the Cartesian propagation directions, the trade-off is acceptable and even desirable in the sense that it reduces anisotropy and leads to a more uniform stability condition.

In terms of methodology, it is suggested that the dissipation and dispersion errors of Takacs (1985) can be useful indicators of scheme properties, but only when derived from linear advection simulations.

Chapter 6

A New Conservative, Fourth Order Scheme

6.1 Introduction

The concept of a Cartesian cut cell grid was introduced in Chapter 1 and explored in more detail in §2.4. Starting with a background grid of Cartesian cells, an arbitrarily-shaped boundary is superimposed such that some of the cells are cut by the boundary. In order to implement a multi-moment method on such a grid, two new multi-moment schemes need to be formulated: one for the background grid of uncut cells, the other for the cut cells. These ought to have similar properties and, ideally, matching arrangements of moments at the uncut/cut cell interfaces. The present chapter deals with the background Cartesian scheme.

It has been established that the space discretisation is the fundamental feature of any multi-moment scheme. The full space–time discretisation, whether Eulerian, semi-Lagrangian or a hybrid formulation, is of secondary importance, although it has been shown that choosing the semi-Lagrangian formulation can rescue a scheme that would otherwise be unstable. A methodology for analysing the stability and accuracy of a Cartesian multi-moment space discretisation was developed in Chapter 3 and subsequently verified in Chapters 4 and 5. This methodology provides a straightforward means by which scheme design can proceed. One can simply choose some combination of derivatives and integrated averages to populate a cell, analyse it, and check the resulting eigenvalues over a range of wavenumbers and propagation angles.

Of the various multi-moment methods, one in particular stood out in Chapter 2 as having a space discretisation suited to implementation on a cut cell grid. This is the CIP- or IDO-based multi-moment finite volume method (hereby referred to as CIP/IDO/MM–FVM). Conservation of the advected variable is maintained within the cell, which is a desirable property and consistent with the choice of a cell-based grid. In this sense CIP/IDO/MM–FVM is superior to the original constrained interpolation profile (CIP) or interpolated differential operator (IDO) method. On the other hand, the CIP-based version (CIP/MM–FVM) is not a purely semi-Lagrangian method, so stability at higher Courant numbers is sacrificed. This is a reasonable price to pay given that the method is flexible enough to be implemented in non-Cartesian cells, as proven by Ii *et al.* (2005).

The fourth order triangle-based scheme of Ii *et al.* is an ideal candidate for adaptation to cut cells. At this point, we defer attempts to adapt it and concentrate on finding a compatible fourth order Cartesian CIP/IDO/MM–FVM scheme. This is an interesting problem in its own right, since such a scheme has not been presented in the literature. Of even greater interest would be a scheme that is *economical* in terms of the number of moments stored. Fourth order Cartesian schemes currently appearing in the multi-moment literature are invariably split-operator or tensor-product based. In general, the number of moments that need to be stored is proportional to the number of coefficients in the polynomial basis. The tensor product basis starts to become expensive for cubic and higher polynomials: 16 profile coefficients are needed compared with 10 for the more economical order-complete basis. The cost is exacerbated in three dimensions: 64 coefficients are needed compared with 20 for the order-complete basis. One method that gets round the escalating storage costs is the volume- and surface-integrated average method of Xiao *et al.* (2006). Its third order 2D implementation was analysed in the last chapter. The fourth order 2D scheme has a storage cost of just five moments per cell (two derivatives, two face-integrated averages, and the cell-integrated average). However, the method only works under the assumption that all operators are semi-Lagrangian and neighbouring cells are Cartesian – hence our favouring CIP/IDO/MM–FVM.

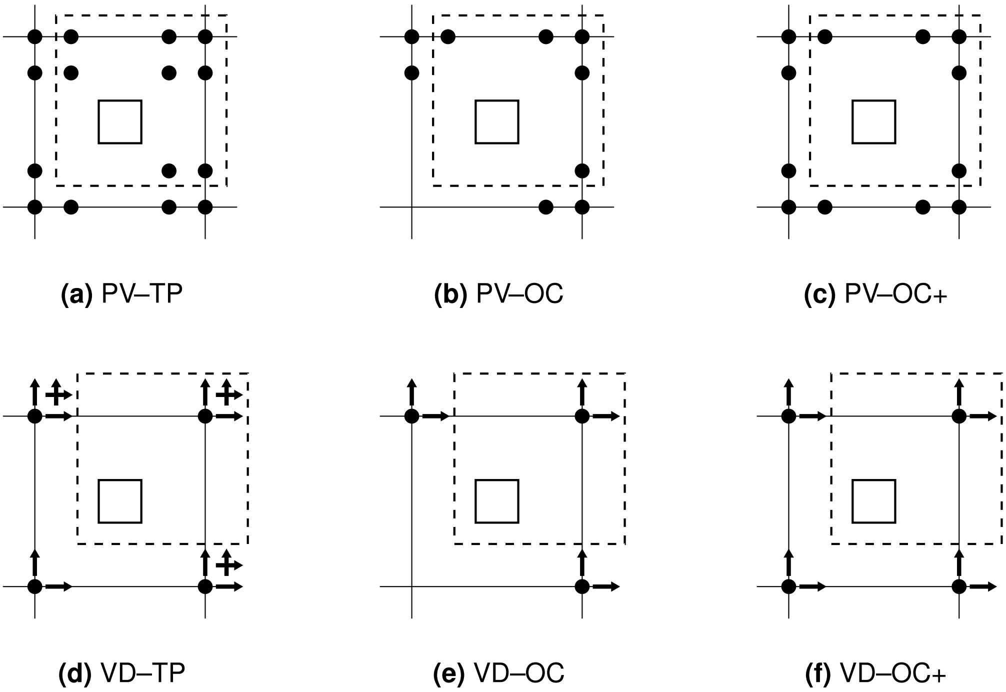
There is clearly an incentive to economise on the number of stored moments by reducing the polynomial basis. Another possible strategy is to populate the lowest-dimensional grid elements – vertices – with derivatives, instead of populating higher-dimensional grid elements with point values or integrated averages. More cells share lower-dimensional elements, so fewer moments are needed to determine the profiles. Both strategies are explored in this chapter.

Time-domain testing is deprecated at this stage in the thesis. Having supported the analysis of a wide range of multi-moment methods in the previous chapters, it has served its purpose in verifying the frequency-domain methodology. The latter methodology is more useful for uncovering stability and accuracy properties, and we can now apply it with confidence to any proposed multi-moment scheme.

6.2 Formulations

Moment arrangements for six schemes are depicted in Figure 6.1, with two further schemes shown in Figure 6.2. Their formulations are explained as follows.

The first scheme (Figure 6.1a) is referred to as the point value, tensor product-based scheme (PV–TP). It was first proposed in §2.5. Here it serves as a baseline in the sense that it is the least economical but the most likely to work. In accordance with Ii *et al.* (2005), points are spaced such that they may double as Gaussian quadrature points (the exact locations can be recalled from Figure 5.4). For symmetry, four point values populate the cell interior; but because the resulting 16 point values plus the cell integrated average overdetermine the tensor product basis, the most upwind interior point value is omitted from the reconstruction. An alternative strategy is to treat the four interior points by least squares (PV–TP–L, Figure 6.2a). The storage cost of PV–TP or PV–TP–L is 10 moments per cell: one vertex value, four edge point values, four interior point values and the cell-integrated average.

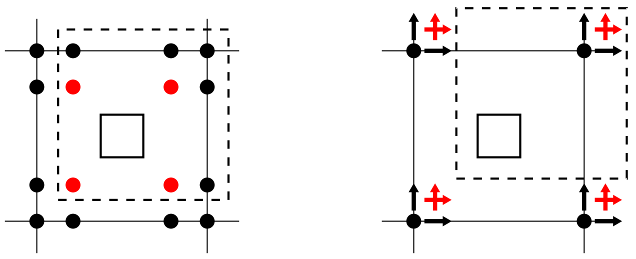

(a) PV–TP **(b)** PV–OC **(c)** PV–OC+

(d) VD–TP **(e)** VD–OC **(f)** VD–OC+

Figure 6.1: Six candidates for a new fourth-order CIP/IDO/MM–FVM scheme. The first three use point values (PV); the latter three use vertex derivatives (VD). Polynomial bases are tensor product (TP), order-complete (OC) and order-complete-plus (OC+). The dashed box indicates the moments being evolved, i.e. moments subscripted ‘00’ according to the notation of §3.5.1. The wind direction is towards the right and upwards.

The second scheme (PV–OC, Figure 6.1b) uses the order-complete basis. The four interior point values can be deleted entirely, leading to a storage cost of six moments per cell. The edge point values still overdetermine the profile, however, and three upwind point moments must be omitted from the reconstruction.

There is something inconsistent in the PV–OC arrangement. It may be recalled that, unlike point-based schemes such as Type-A CIP/IDO, the cell-based CIP/IDO/MM–FVM relies on an integration of fluxes over the cell faces. The order of flux reconstruction and integration should be consistent with the nominal order of the scheme. In other words, the point moments populating the edges in Figure 6.1a should be able to support a cubic curve over each edge.



(a) PV–TP–L **(b)** VD–TP–L

Figure 6.2: Alternative tensor product-based candidates. Moments in red are consigned to the least squares space.

This is true of PV–TP, where there are always four point values per edge. However, two of the edges in PV–OC are deficient by two moments.

We might try excluding vertex values from the exactly constrained space in order to make space for the eight edge point values. The fluxes could still be reconstructed and integrated to fourth order accuracy, because the edge points are also fourth order Gaussian quadrature points. This is how Ii *et al.* (2005) integrate the fluxes. However, when the corresponding eight rows are formed according to the order-complete basis and then assembled, the resulting rank is only seven: the system is deficient. So it seems that such a scheme would be inconsistent.

The situation may be remedied by adding three more terms to the order-complete basis in order to admit all the edge and vertex values into the exactly constrained space. Equation 6.1 gives a new polynomial basis, with new terms highlighted in red. This basis will be referred to as the 'order-complete-plus' (OC+) basis. Fortunately it is well-determined by the arrangement shown in Figure 6.1c: the rank of the resulting Vandermonde matrix is 13.

$$
\begin{aligned}
U_{3OC+}(\xi,\eta) := \quad & c_{00} && +c_{10}\xi && +c_{20}\xi^2 && +c_{30}\xi^3 \\
& +c_{01}\eta && +c_{11}\xi\eta && +c_{21}\xi^2\eta && +c_{31}\xi^3\eta \\
& +c_{02}\eta^2 && +c_{12}\xi\eta^2 && +c_{22}\xi^2\eta^2 \\
& +c_{03}\eta^3 && +c_{13}\xi\eta^3
\end{aligned}
\tag{6.1}
$$

The arrangements shown in Figures 6.1d through 6.1f, and the constrained least squares treatment shown in Figure 6.2b, are simply vertex derivative-based analogues of the point value-based arrangements. That is, edge point values are replaced by vertex first order derivatives, and interior point values are replaced by vertex cross derivatives. The effect of these replacements is a reduction in the number of moments that need to be stored: the cost of VD–TP(–L) is five moments per cell, compared with PV–TP(–L)'s 10; and the cost of VD–OC(+) is four moments per cell, compared with PV–OC(+)'s six. The bases remain well-determined.

6.3 Method

For these eight schemes in the fully Eulerian formulation (IDO/MM–FVM), matrices of weights were computed and subjected to the frequency-domain analysis outlined in Chapter 3. As usual, three propagation angles θ were tested: 0°, $\tan^{-1} 0.5 \approx 26.6°$, and 45°.

For the three angles, it was necessary to check for (a) positive real eigenvalue parts and (b) subnominal convergence on the exact solution as $K \to 0$. The former are a sign of instability, although it was shown in Chapter 5 that real eigenvalue parts can end up being slightly positive – for instance when manipulation of the operators introduces downwind information – with the scheme still maintaining stability in a semi-Lagrangian formulation. In other words, even if the IDO-based scheme shows positive real eigenvalue parts, it may be possible for the CIP-based scheme to have conditional stability. However, one cannot tell for certain with only the semi-discrete analysis.

In any case, we insist on all propagation angles supporting the nominal order of accuracy, otherwise the scheme is considered to have failed. It was previously found that a scheme could be acceptable at $\theta = 0°$, only to deteriorate at oblique propagation angles – thus defeating the multidimensional representation of the method.

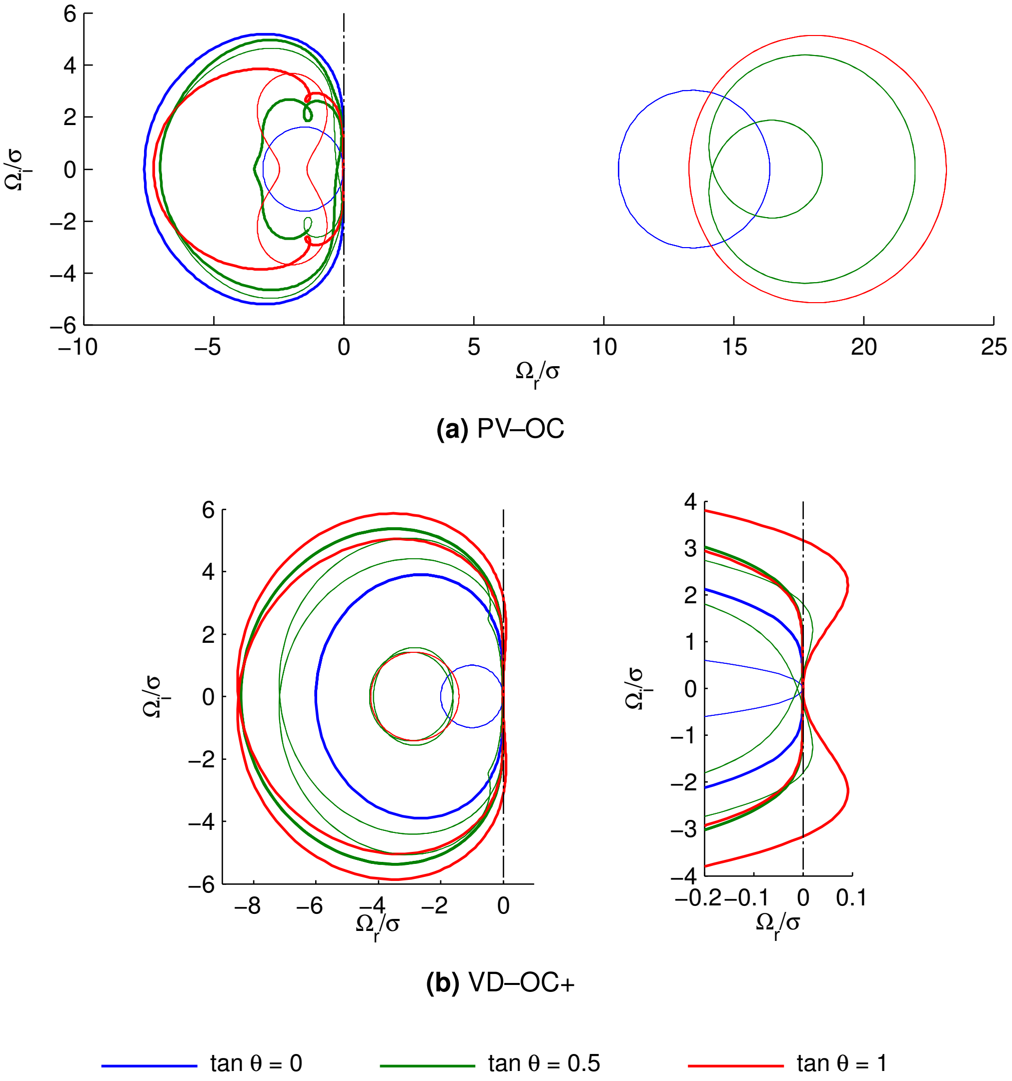
The eigenvalues were sampled in steps of $\Delta K = \pi/50$, up to $K_{\max} = 6\pi\left(1, \sqrt{5}, \sqrt{2}\right)$ for the three respective angles. The factor 6π was chosen to match the K-range of fifth order IDO. It may be recalled from §2.2.1 and §4.2.2 that the wavenumber K_0 at which the numerical phase speed drops to zero is 2π for IDO3/DG2, 3π for IDO5/DG3, and so on. In general, the spectrum closes at $K_{\max} = 2K_0$. This is modified for oblique angles as discussed in §3.4.1.

6.4 Results

Maximum real eigenvalue parts and convergence rates of the eight schemes are given in Table 6.1. Positive real eigenvalue parts and subnominal convergence rates are highlighted in red. It can be seen that PV–TP, PV–TP–L and PV–OC+ are the only acceptable fourth order schemes, although VD–TP works as a third order scheme. The spectra of two of the unstable schemes are shown for interest in Figure 6.3, with decoupled spurious modes included (a discussion on physical versus spurious modes is given on p. 62).

Spectra, physical-mode error norms $||\varepsilon(K)||^{\text{phys}}$ and polar dissipation factors for the acceptable schemes are presented in Figures 6.4 through 6.6. The third order VD–TP is included for interest. The PV–TP and PV–TP–L curves are coincident, indicating that the spectra are identical. Spectral radii and measures of isotropy K_c^{iso} are presented in Table 6.2.

The PV–OC+ $||\varepsilon(K)||^{\text{phys}}$ curves can be seen to intersect the isotropic ideal: the $\theta = \tan^{-1} 0.5 \approx 26.6°$ curve does this at $K = 0.980\pi$ (or 2.04 points per wavelength) and the $\theta = 45°$ curve does it at $K = 0.881\pi$ (2.27 points per wavelength). With respect to the polar plots, normalised phase speeds are not shown because the fourth order accuracy makes them difficult to distinguish even at $K = \pi$.



(a) PV–OC

(b) VD–OC+

Figure 6.3: Spectra of two IDO/MM–FVM schemes showing instability. Physical modes are represented by bold lines, spurious modes by thin lines.

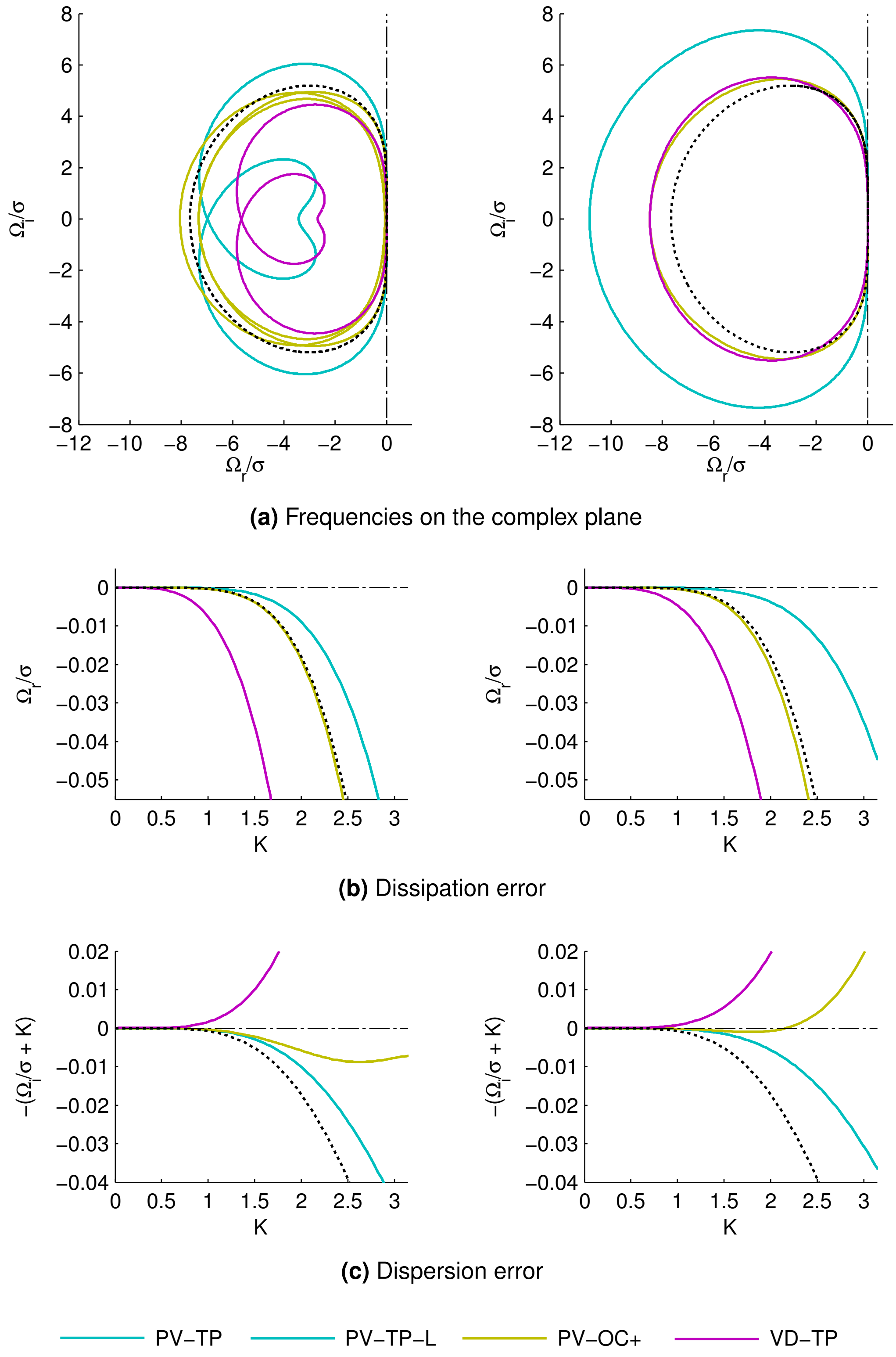
Table 6.1: Critical spectral details for the fourth order IDO/MM–FVM candidates. Convergence rates have been calculated from Equation (3.29) with $K_a = \pi/50$.

(a) Main schemes (well-determined)

Scheme	$\tan\theta$	Max real Ω/σ	Convergence rate	
			Dissipation	Dispersion
PV–TP	0.0	$2.354\text{e}{-14}$	5.000	3.998
	0.5	$3.914\text{e}{-14}$	5.005	3.998
	1.0	$4.685\text{e}{-14}$	5.015	3.999
PV–OC	0.0	$1.639\text{e}{+01}$	4.999	3.998
	0.5	$2.199\text{e}{+01}$	3.001	3.999
	1.0	$2.318\text{e}{+01}$	3.001	3.997
PV–OC+	0.0	$1.599\text{e}{-14}$	4.999	3.998
	0.5	$-1.776\text{e}{-15}$	4.999	3.997
	1.0	$-3.553\text{e}{-15}$	4.998	3.995
VD–TP	0.0	$9.237\text{e}{-14}$	2.998	3.996
	0.5	$7.998\text{e}{-14}$	2.999	3.997
	1.0	$5.024\text{e}{-14}$	2.999	3.998
VD–OC	0.0	$4.000\text{e}{+00}$	2.998	3.996
	0.5	$6.452\text{e}{+00}$	2.998	3.996
	1.0	$6.749\text{e}{+00}$	2.999	3.995
VD–OC+	0.0	$2.638\text{e}{-07}$	2.998	3.996
	0.5	$1.969\text{e}{-02}$	4.999	3.995
	1.0	$9.118\text{e}{-02}$	4.998	3.988

(b) Alternative schemes (constrained least squares)

Scheme	$\tan\theta$	Max real Ω/σ	Convergence rate	
			Dissipation	Dispersion
PV–TP–L	0.0	$4.575\text{e}{-14}$	5.002	3.998
	0.5	$-4.300\text{e}{-15}$	4.994	3.998
	1.0	$-3.348\text{e}{-15}$	4.984	3.999
VD–TP–L	0.0	$2.097\text{e}{-14}$	2.998	3.976
	0.5	$3.372\text{e}{-03}$	4.997	3.993
	1.0	$5.761\text{e}{-16}$	4.998	3.994



(a) Frequencies on the complex plane

(b) Dissipation error

(c) Dispersion error

Figure 6.4: Spectra of working IDO/MM–FVM schemes at two propagation angles: (left) $\theta = \tan^{-1} 0.5 \approx 26.6°$; (right) $\theta = 45°$. The fourth order isotropic ideal is shown dotted.

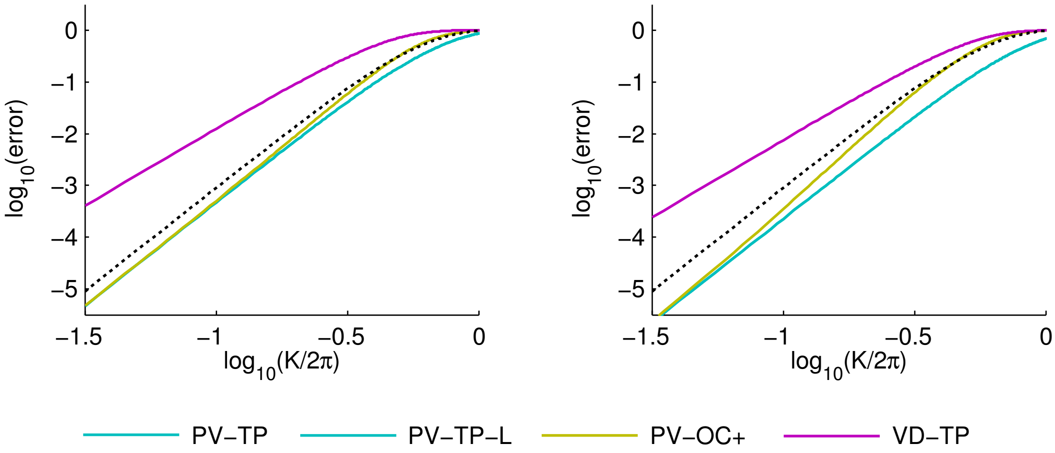
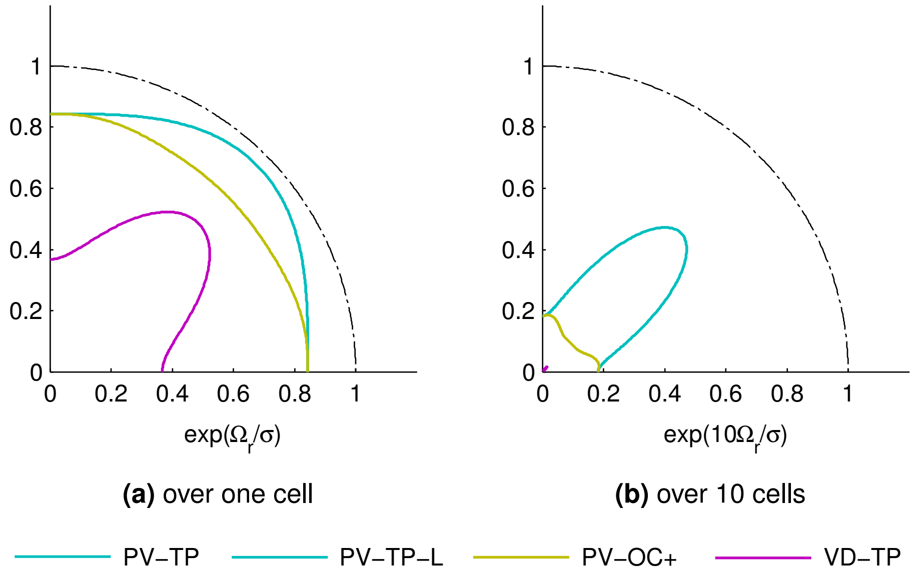


Figure 6.5: Physical-mode error norm $\|\varepsilon(K)\|^{\text{phys}}$ of working IDO/MM–FVM schemes at two propagation angles: (left) $\theta = \tan^{-1} 0.5 \approx 26.6^{\circ}$; (right) $\theta = 45^{\circ}$. The fourth order isotropic ideal is shown dotted.



(a) over one cell

(b) over 10 cells

Figure 6.6: Polar dissipation factor of working IDO/MM–FVM schemes at $K = \pi$.

Table 6.2: Spectral radii and measures of isotropy, as calculated from Equation (3.94), of working IDO/MM–FVM schemes.

Scheme	Spectral radius		K_c^{iso} at $\theta = 45^{\circ}$		
	$\theta = 0^{\circ}$	$\theta = 45^{\circ}$	Dissipation	Dispersion	Overall
PV–TP(–L)	7.660	10.834	1.6429	1.6109	1.61
PV–OC+	7.660	8.485	2.2283	1.5358	1.53
VD–TP	6.000	8.485	0.8840	1.2113	0.88

6.5 Discussion

The baseline PV–TP is stable and accurate, with pronounced accuracy in the diagonal propagation direction as one would expect. It is interesting that the constrained least squares treatment of PV–TP–L should give identical results. It suggests that the way moments are distributed *inside* the cell does not impact the resulting spectra. This would be consistent with the discontinuous Galerkin analysis (Hu *et al.*, 1999) which can be done entirely without needing to specify collocation points. Indeed, when PV–TP is reanalysed with different combinations of interior points relative to the wind direction, the same spectra are recovered. It is concluded that only three interior point values need be stored, reducing the cost of the baseline scheme to nine moments per cell. However, the arrangement of moments at the cell boundary *does* matter, as evidenced by VD–TP versus VD–TP–L. Consistent with the findings of Chapter 4, omitting upwind moments is the better treatment.

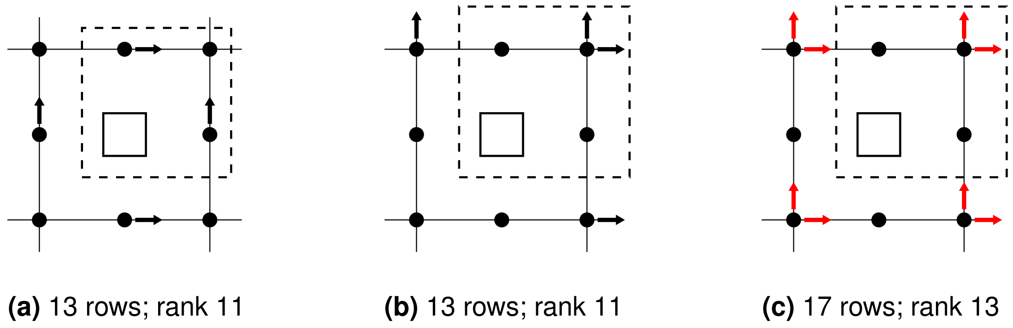
An economical alternative to the baseline has been successfully identified in PV–OC+. The accuracy of this scheme diminishes in oblique directions owing to the removal of interior point values, but what is remarkable is that this only happens at large K. Figure 6.5 shows that as $K \to 0$, the scheme appears to approach the accuracies of PV–TP. The schemes therefore have similar isotropies as formally defined by K_c^{iso}. Looking at it another way, the accuracy of PV–OC+ is better than isotropic at grid resolutions all the way down to 2.27 points per wavelength. Considering that the scheme stores two thirds of the moments of the baseline arrangement, this is an excellent result. Furthermore, the spectra of PV–OC+ at oblique angles are smaller than those of PV–TP by up to 28%, allowing for proportionally larger Courant numbers.

As suspected, the order-complete arrangements failed. Given that there is no way for fluxes at all the cell boundaries to be represented with fourth order accuracy, it is not surprising that the schemes' orders of accuracy deteriorate in oblique directions. A bigger problem is that the eigenvalues do not just stray into the positive real half of the complex plane; in the case of PV–OC they can be seen forming decoupled spurious modes with completely positive real parts (Figure 6.3a). It underscores the fact that one cannot get away with misrepresenting fluxes at the cell boundaries.

Besides reducing the polynomial basis, it was hoped that further savings in storage costs could be made by storing derivatives on the vertices rather than point values around the cell. Unfortunately, it appears that vertex derivative-based schemes lead to reduced orders of accuracy and/or eigenvalues that stray into the positive real half of the complex plane. VD–TP actually makes for a stable third order scheme, but it is of little practical interest since Chen *et al.* (2011) present a more economical arrangement. Incidentally, VD–TP appears to have the same spectra as Type-C IDO and IDO/VSIAM2 (§5.4.1). VD–OC is highly unstable due to its order-complete basis. VD–OC+ and VD–TP–L are intriguing because their accuracies deteriorate to third order, but only in the Cartesian propagation direction: they effectively reduce to third-order IDO along gridlines.

It was previously inferred that the order of one-dimensional CIP or IDO schemes could be promoted by introducing the cell-integrated average into the reconstruction. The present experiment has shown that this is not necessarily valid for multidimensional CIP/IDO/MM–FVM. Furthermore, the order of accuracy of a multi-moment scheme does not simply depend on the degree of polynomial basis. The number of moments stored per cell is more important. It was noted in §4.3.1 that for a 2D CIP/IDO scheme of order $(2N-1)$, $N(N+1)/2$ moments were needed to support the order-complete basis and N^2 were needed to support the tensor product basis. A similar rule could apply here, if fourth order accuracy corresponds to $N = 3$. Six moments are enough to support a stable fourth order accurate scheme; five or fewer moments are not; and nine moments can support a stable, fourth order accurate, tensor product-based scheme.

In the interest of improving either the storage cost or accuracy of PV–OC+, we might finally ask whether there is a middle ground between the point value-based and vertex derivative-based arrangement. Instead of populating each edge with two point values, we might store only the midpoint value. Derivatives can then be introduced to complete the basis, with the proviso that each cell face has access to four moments to support a cubic flux profile reconstruction. Figure 6.7 shows three such arrangements. The first thing to say about them is that there is no saving on storage cost – arrangements storing fewer than six moments per cell could not be found. The second is that they are all rank-deficient systems. Single points on edges evidently do not combine well with derivatives. Of the three arrangements, only the last one (Figure 6.7c) has sufficient rank to complete the OC+ basis. But on closer inspection this arrangement is no better than VD–OC+. Midpoint values have been added, but they do not improve the rank of the overall system. Analysis confirms that the new arrangement has similar properties to VD–OC+.



(a) 13 rows; rank 11 **(b)** 13 rows; rank 11 **(c)** 17 rows; rank 13

Figure 6.7: Alternative arrangements for the OC+ basis.

6.6 Conclusions

The aim of the present chapter was to uncover a new, economical, fourth-order CIP/IDO/MM–FVM scheme in two dimensions, and this has been successfully achieved by the point value, order-complete-plus (PV–OC+) variant. While a baseline tensor product-based CIP/IDO/MM–FVM scheme needs to store nine moments per cell, PV–OC+ only needs to store six. This scheme has the same high accuracies as the baseline at low K (high grid resolutions) and remains more accurate than the isotropic ideal up to high K. It also allows for a higher Courant number in oblique propagation directions.

The saving in storage cost is made by reducing the polynomial basis while preserving the ability to represent high-order fluxes at the cell boundaries. One is also tempted to support the basis with shared vertex derivatives instead of edge point values, but this approach generally leads to a reduced order of accuracy at best and instabilities at worst. It seems inviolable that a minimum of six stored moments per cell are needed to support a stable, fourth order-accurate CIP/IDO/MM–FVM scheme.

Further work might involve testing the scheme in the time domain, perhaps with the nonlinear Euler equations in addition to linear and rotational advections, and analysing the fully discrete CIP/MM–FVM system in order to ascertain stability conditions in terms of the Courant number. In the meantime, however, it can be assumed that the allowable Courant number of the CIP/MM–FVM formulation is greater than that of IDO/MM–FVM.

Chapter 7

Cut Cell Scheme Design

7.1 Introduction

In this chapter, we arrive at the difficult task of adapting a multi-moment scheme to cut cells. The scheme in question is the fourth order, triangular cell-based constrained interpolation profile/multi-moment finite volume method (CIP/MM–FVM) introduced by Ii *et al.* (2005). The merits of this scheme were discussed in Chapters 2 and 6, with Chapter 6 proceeding to find a compatible scheme to occupy the background Cartesian grid. The cut cell and background schemes together form the implementation of CIP/MM–FVM on a cut cell grid.

The numerical procedure of Ii *et al.* consists of the following steps:

(i) reconstruct an order-complete cubic profile over the cell according to the nine point values on the cell boundary plus the cell-integrated average;

(ii) advance the point values on the cell boundaries by CIP;

(iii) advance the cell-integrated average by a fourth order integration of fluxes reconstructed over the cell boundaries.

Applying the same algorithm to a hypothetical Cartesian cut cell, steps (ii) and (iii) remain straightforward. Step (ii) in fact becomes easier: in an unstructured grid, in order to interpolate at some departure point coordinates, some logic is needed to identify the departure cell profile. In a Cartesian cut cell grid this is trivial; a single calculation will return the desired (i, j, k) address. This leaves step (i) as the most complex part of the algorithm. The problem of profile fitting, and the implications for scheme stability and accuracy, are the focus of this chapter.

7.1.1 Considerations

In contrast to a simplex (triangle or tetrahedron), a cut cell has an arbitrary collection of vertices, edges and faces to be populated with moments. It may be tempting to enumerate and individually treat the various combinations; for example, in representing the cell-integrated average, one could select from a set of previously defined quadrature rules according to the cell shape. However, enumeration should be avoided because it does not scale well to 3D. In fact, it may well prove impossible if cusps are admitted into the grid (more on this below). In our example, then, it is preferable to use a triangulation algorithm to decompose the volume into a set of simplices, then use a standard pattern of quadrature points to integrate over each simplex.

High order profile reconstruction may become especially troublesome in cut cells. The moments must be configured in such a way that the profile is suitably constrained over the cell. There is a danger that cutting close to preexisting vertices may lead to badly-conditioned or rank-deficient matrices. It is therefore important to test sensitivity of the reconstruction algorithm to extremes in cell shape.

According to the literature review we must be able to represent high-order, curved boundary conditions. It is not sufficient to assume piecewise-linear segments between cutting points (points where the boundary intersects the Cartesian gridlines), as traditional cut cell methods do. Fortunately it is straightforward to make provisions for curved boundaries, even if the actual implementation of them is another matter. In 2D, one can simply introduce the boundary conditions at Gaussian points on each boundary edge (Ollivier-Gooch and Van Altena, 2002). A pair of points correctly spaced ensures fourth order integration of boundary fluxes. This ties in conveniently with the findings of Chapter 6. In that chapter it was found that the fourth order CIP/IDO/MM–FVM reconstruction requires each cell boundary to support a cubic curve, and that the support should come from point values rather than vertex derivatives. By extending these rules to cut cells, we automatically accommodate fourth order boundary conditions.

There is another issue that has not been brought up before now. What happens when there are cusps (C^0 discontinuities) in the boundary? In a traditional cut cell grid, these do not pose much of a problem. Any boundary vertex present within a cell may simply be ignored. The immediate cutting points will still represent a piecewise-linear cut, and successive grid refinements should eventually converge on the continuum solution. This situation is depicted in Figure 7.1. But for higher orders of boundary edge, the collocation points represent a continuous curve that could become increasingly distorted with refinements. Such distortions could disrupt convergence or even destroy stability. One remedy is to admit extra vertices into the grid, although this approach brings its own challenges.

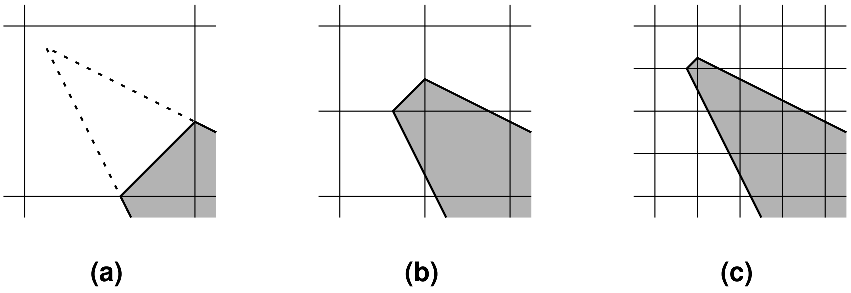


Figure 7.1: Piecewise-linear truncation of a cusp, and ensuing grid refinement.

Suppose all of the above considerations have been taken into account, and an algorithm has been developed that produces ‘nice’-looking profiles over a range of test cells. How do we know whether we have really been successful? The proof is surely in the stability and accuracy of the resulting scheme. Here there are two options: either modify the frequency-domain methodology to represent some periodic tessellation of cut cells, or develop a new time-domain solver with cut cell architecture. The former option is limited in scope but involves much less work and should be attempted first.

7.2 Profile Fitting

The present task is to develop an algorithm that fits a polynomial profile to an arbitrarily cut cell. Specifications for the task follow directly from the above discussion. The cell-integrated average should be represented to fourth order accuracy, and each face of the cut cell should be able to support a cubic curve via collocation points. In 2D, this means each vertex or cutting point should hold a point value, and each connecting edge should hold two point values. In addition, we insist that the reconstruction should not be adversely affected by the proximity of cuts to Cartesian grid points, or by the presence of discontinuities such as cusps. Finally, the algorithm should scale easily to 3D.

It is helpful to define a suite of 2D test cells. The cells are shown in Figure 7.2. Since shapes arising from piecewise-linear cuts are invariably convex polygons with three to five sides, such polygons are represented in Figures 7.2a through 7.2c. When a cusp is introduced into the boundary, concave polygons are possible. Three of them are represented in Figures 7.2d through 7.2f. Curved boundaries are not included since, as discussed above, boundaries with cusps are considered more challenging.

7.2.1 Algorithms

Two profile fitting algorithms are detailed in the following subsections. One thing common to both of them is the spacing of collocation points along edges. Points are positioned $^1/_2\,(1 \mp {^1/\sqrt{3}})$ of the way along the edge length, consistent with the Gaussian quadrature rule based on cubic interpolation.

Another common factor is the representation of the cell-integrated average, which is done by simplex quadrature. Given a set of cut cell vertices, the Delaunay triangulation of Barber *et al.* (1996, via MATLAB® R2009a) is applied to get a set of N_T triangles. It may be objected that concave hulls (Figures 7.2d through 7.2f) are not properly triangulated by the Delaunay algorithm. Indeed, more sophisticated triangulations such as Chew’s second algorithm (1993) could be used instead, but in the present work it was sufficient to simply delete the extraneous triangles.

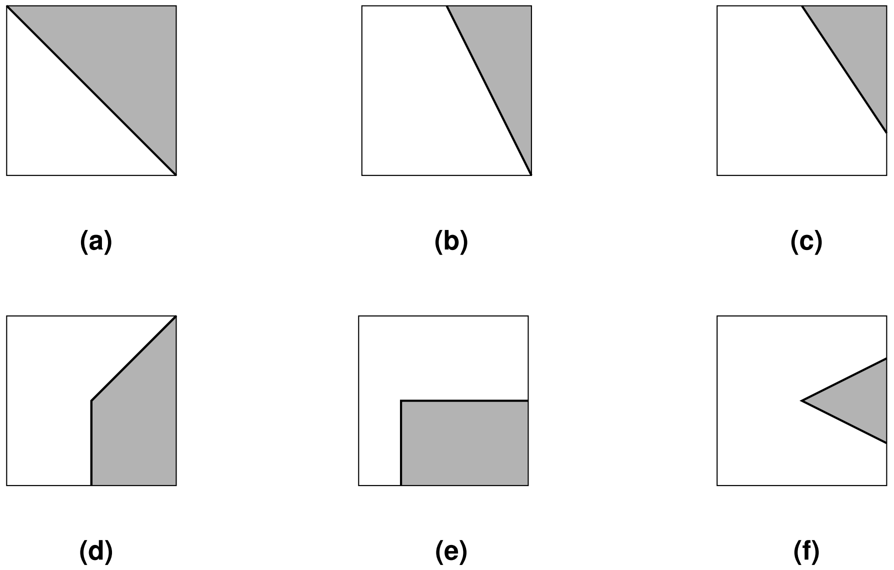


Figure 7.2: Sample cut cells.

The kth triangle has an associated set of vertices Ξ_{Vk} in terms of the local coordinates ξ and η. From this a local area T_k can be calculated.

$$\Xi_{Vk} := \begin{pmatrix} \xi_{Vk}^{\langle 1\rangle} & \eta_{Vk}^{\langle 1\rangle} \\ \xi_{Vk}^{\langle 2\rangle} & \eta_{Vk}^{\langle 2\rangle} \\ \xi_{Vk}^{\langle 3\rangle} & \eta_{Vk}^{\langle 3\rangle} \end{pmatrix} \qquad T_k = \frac{1}{2}\left|\det\begin{pmatrix} \Xi_{Vk} & \mathbf{1} \end{pmatrix}\right| \tag{7.1}$$

A set of simplex quadrature points guaranteeing exact integration of a cubic is provided by Hammer and Stroud (1956). For a triangle, barycentric coordinates $\boldsymbol{\Lambda}_Q$ and weights $\boldsymbol{\rho}_Q$ are given as

$$\boldsymbol{\Lambda}_Q = \begin{pmatrix} {}^{1}/_{3} & {}^{1}/_{3} & {}^{1}/_{3} \\ {}^{3}/_{5} & {}^{1}/_{5} & {}^{1}/_{5} \\ {}^{1}/_{5} & {}^{3}/_{5} & {}^{1}/_{5} \\ {}^{1}/_{5} & {}^{1}/_{5} & {}^{3}/_{5} \end{pmatrix} \qquad \boldsymbol{\rho}_Q = \begin{pmatrix} -{}^{9}/_{16} \\ {}^{25}/_{48} \\ {}^{25}/_{48} \\ {}^{25}/_{48} \end{pmatrix} \tag{7.2}$$

Each triangle's set of vertices converts the quadrature points' barycentric coordinates into local coordinates Ξ_{Qk}. The ith row in Ξ_{Qk} gives $\left(\xi_{Qk}^{\langle i\rangle}, \eta_{Qk}^{\langle i\rangle}\right)$. Also, the proportion of the triangle's area in the cut cell rescales the native weights.

$$\Xi_{Qk} = \boldsymbol{\Lambda}_Q \Xi_{Vk} \qquad \boldsymbol{\rho}_{Qk} = \frac{T_k}{\sum_{k=1}^{N_T} T_k} \boldsymbol{\rho}_Q \tag{7.3}$$

It will be recalled that the polynomial basis can be represented by a row vector **r** of local coordinate products. The Vandermonde matrix row $\mathbf{A}_{Qk}$ representing the contribution of the kth triangle to the cell-integrated average is computed by summing the rescaled quadrature weights with **r** evaluated at the respective point coordinates. The row $\mathbf{A}_Q$ representing the cell-integrated average is computed by summing $\mathbf{A}_{Qk}$ for all simplices.

$$\mathbf{r} = \begin{pmatrix} 1 & \xi & \dots & \xi^p \eta^q \end{pmatrix}$$

$$\mathbf{A}_{Qk} = \sum_{i=1}^{4} \rho_{Qk}^{\langle i \rangle} \mathbf{r} \Big|_{\xi_{Qk}^{\langle i \rangle}, \eta_{Qk}^{\langle i \rangle}} \tag{7.4}$$

$$\mathbf{A}_Q = \sum_{k=1}^{N_T} \mathbf{A}_{Qk} \tag{7.5}$$

Single Polynomial, Varying Basis

For the first concept, a single polynomial function is constrained over the polygon. The cell-integrated average and all point values – one one per vertex, two per edge – must be exact constraints. Consequently the polynomial must have no fewer than $3N_S + 1$ coefficients, where N_S is the number of polygon sides. It has been established that the 10 coefficients of the order-complete cubic basis are naturally suited to a triangle, and three more coefficients may be added for a quadrilateral.

What about a pentagon? One might try the tensor product basis, which has 16 coefficients. However, when the Vandermonde matrix is assembled for the cell in Figure 7.2c, it turns out to have a rank of only 15. It is necessary to go to a higher polynomial. The quartic order-complete basis is short by one coefficient, so a quartic 'order-complete-plus' basis is hereby created; see Equation (7.6). The matrix pseudoinverse will resolve the underdetermined system by minimising the L^2-norm of all coefficients. This is not claimed to be the best treatment – minimisation of only the highest order coefficients is undoubtedly better – but it suffices for the present investigation.

$$\begin{array}{llllll} U_{4OC+}(\xi,\eta) := & c_{00} & +c_{10}\xi & +c_{20}\xi^2 & +c_{30}\xi^3 & +c_{40}\xi^4 \\ & +c_{01}\eta & +c_{11}\xi\eta & +c_{21}\xi^2\eta & +c_{31}\xi^3\eta & +c_{41}\xi^4\eta \\ & +c_{02}\eta^2 & +c_{12}\xi\eta^2 & +c_{22}\xi^2\eta^2 & +c_{32}\xi^3\eta^2 & \\ & +c_{03}\eta^3 & +c_{13}\xi\eta^3 & +c_{23}\xi^2\eta^3 & & \\ & +c_{04}\eta^4 & +c_{14}\xi\eta^4 & & & \end{array} \tag{7.6}$$

It is clear that there is a risk of rank deficiency that will only get worse as the number of cell edges increases. Therefore, it is necessary to limit the number of edges populated with point values. Discontinuous boundaries can be regarded as single edges within the cell, with quadrature points spaced according to the total edge length. This means that in the worst case, assuming only one cusp ever appears within a cell, a cell will have no more than six edges (19 moments). This situation corresponds to Figure 7.2f, where one edge is penetrated by a sharp boundary.

Unfortunately, a full-rank matrix could not be created for the last test cell, even with sextic and septic tensor product bases. This is perhaps not surprising given that only collinear points are being added. There is no choice but to disallow such cells: grid refinement would eventually lead to five or fewer faces per cell anyway. For interest, this last cell was still included in tests. It was assigned a sextic order-complete-plus basis.

The difficulties encountered in constructing non-singular matrices for multifaceted cells can only be expected to get worse in 3D, and it must be questioned whether such an extension to 3D could ever be robust.

Piecewise Polynomial, Shared Integrated Average

If the first concept should fail, a fallback approach might be to take the cut cell triangulation previously used to calculate the cell-integrated average and use it instead to fully decompose the cell into smaller finite volumes. The result would be a hybrid grid of Cartesian and simplicial cells. The scheme of Ii *et al.* (2005) could be used in the latter cells without modification. But besides defeating the spirit of the cut cell method, there is a serious problem with this approach. Very small or thin sliver cells may be created (Figure 7.3a) leading to a proportionally severe restriction on the global time step. One way around this situation would be to merge the offending cut cell with an appropriate neigbour before re-triangulating (Figure 7.3b). But it is not ideal. One can foresee clusters of Cartesian cells needing re-triangulation, and it may require further algorithms to choose the best mergers and control the size and quality of resulting simplices. In short, the approach may end up becoming no less complicated than a fully unstructured grid implementation.

For the second profile fitting concept, a compromise is proposed. The cut cell is triangulated into N_T simplices, with each simplex having its own order-complete polynomial reconstruction. The reconstruction is partly constrained by the triangle's three vertex values and three pairs of edge point values. In this respect the concept does not differ from that of Ii *et al.*. However, the cut cell retains the role of finite volume; it holds a single integrated average which is shared between the simplices. The N_T reconstructions are therefore linked by a master linear system for the cut cell.

The justification for this concept is as follows. It is reasoned that if the point values are evolved by the semi-Lagrangian CIP, they do not contribute to the restriction on stability that would fatally impair a triangulation with arbitrarily thin simplices. The only restriction comes from the Eulerian evolution of the cut cell's integrated average, and this has been shown in traditional

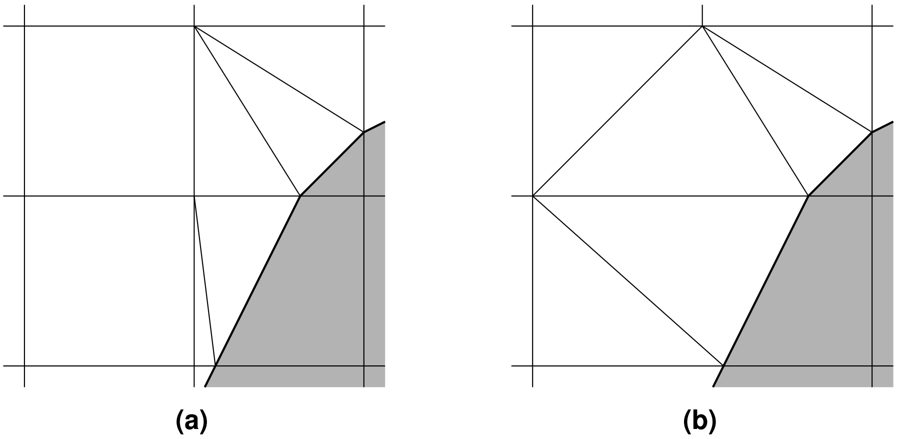


Figure 7.3: Re-triangulation to avoid sliver cells.

cut cell methods to be tolerable. Some merging of the cut cells may be needed, but appropriate merging techniques exist in the cut cell literature.

Since there are N_T order-complete polynomials, there are $10N_T$ coefficients that need to be solved for. Constraining the master linear system are $3N_T$ vertex values, $(3 \times 2)N_T$ edge point values and one integrated average. This means it is underdetermined by $(N_T - 1)$ constraints. If we then add a point value at each simplex centroid, the system becomes overdetermined, but only by one moment. This redundancy needs to be resolved without harming scheme stability and accuracy. Fortunately, as found in the last chapter, interior moments do not play a critical role compared with moments residing on the cell boundaries. We can therefore treat the N_T simplex centroid values with least squares and have all other moments serve as exact constraints. Figure 7.4 depicts the master linear system. The cell-integrated average has been assigned the symbol $\mathbf{u}_Q$ instead of $\overline{{}^{V}u}$ to avoid confusion with the vertex subscript created during this chapter.

Formulating a large linear system such as the one shown in Figure 7.4 is inefficient in practice, because there are blocks of zeros that will become more numerous as more simplices populate the cut cell. The inefficiency can be resolved in a two-step procedure. In the first step, each underdetermined subsystem of vertex and edge point values $\mathbf{A}_{Ek}\mathbf{c}_k = \mathbf{u}_{Ek}$ is partially solved, reducing the number of unknowns n to $(n-p)$ where p is the number of rows (and the rank) of $\mathbf{A}_{Ek}$. This can be done by computing a particular solution $\mathbf{x}_{Ek}$, compatible with the solution space of $\mathbf{A}_{Ek}$, and a set of column vectors $\mathbf{Q}_{Ek,2}$, representing the null space. $\mathbf{c}_k$ can be written as a linear combination of these vectors.

$$\mathbf{x}_{Ek} = \mathbf{A}_{Ek}^{+}\mathbf{u}_{Ek} \tag{7.7}$$

$$\mathbf{Q}_{Ek,2} = \mathrm{null}\left(\mathbf{A}_{Ek}\right) \tag{7.8}$$

$$\mathbf{c}_k = \mathbf{x}_{Ek} + \mathbf{Q}_{Ek,2}\,\mathbf{c}'_k \tag{7.9}$$

$\mathbf{A}_{Ek}^{+}$ is a pseudoinverse (usually the Moore-Penrose) of $\mathbf{A}_{Ek}$. To get the basis for the null space, an orthogonal (e.g. QR) decomposition can be applied to $\mathbf{A}_{Ek}^{\mathsf{T}}$; then the null space is represented in the last $(n-p)$ columns of the orthogonal matrix. The same strategy was applied when formulating constrained least squares in §3.5.2. $\mathbf{c}'_k$ is a vector of the remaining unknowns, or slack variables. In this instance, $n = 10$ and $p = 9$, so there is only one column in $\mathbf{Q}_{Ek,2}$ and one slack variable per simplex.

The N_T slack variables form the solution vector for a new linear system whose matrix is now only $(N_T + 1)$ rows by N_T columns. By substituting Equation (7.9) into equations for the least squares-treated centroid values and the exactly constrained cell-integrated average, it can be shown that

$$\mathbf{A}'_{Lk}\mathbf{c}'_k = \mathbf{u}'_{Lk} \quad \text{where} \quad \mathbf{A}'_{Lk} = \mathbf{A}_{Lk}\mathbf{Q}_{Ek,2} \quad \text{and} \quad \mathbf{u}'_{Lk} = \mathbf{u}_{Lk} - \mathbf{A}_{Lk}\mathbf{x}_{Ek} \tag{7.10}$$

$$\sum_{k=1}^{N_T}\mathbf{A}'_{Qk}\mathbf{c}'_k = \mathbf{u}'_Q \quad \text{where} \quad \mathbf{A}'_{Qk} = \mathbf{A}_{Qk}\mathbf{Q}_{Ek,2} \quad \text{and} \quad \mathbf{u}'_Q = \mathbf{u}_Q - \sum_{k=1}^{N_T}\mathbf{A}_{Qk}\mathbf{x}_{Ek} \tag{7.11}$$

The reduced system is assembled as shown in Figure 7.5 and solved by constrained least squares. After the slack variables have been computed, they can be substituted back into Equation (7.9) to finally recover the simplex profile coefficients.

This algorithm will scale to 3D, but a modification is needed. Populating each tetrahedron with one point value per vertex, two per edge and one per triangular face leads to a total of 20 moments on the tetrahedron boundary. The order-complete cubic polynomial happens to have 20 coefficients: each tetrahedral profile is already well-determined. Adding the cut cell integrated average means that the master linear system is overdetermined by one moment. This poses a problem. Without tetrahedron centroid values to act as ‘padding’, which moments can be consigned to the least squares space? Those residing on the cut cell’s faces are out of the question; they need to be exact constraints to support order-complete flux reconstructions.

One option is to consign point values on those triangular faces that are strictly *inside* the cut cell, not on its boundary. This unfortunately implies that the tetrahedral profiles will no longer be C^0-continuous. However, this may not matter since the flux reconstructions remain unaffected. Another option would be to add more coefficients to the polynomials. A minimum of three, corresponding to x^2yz, xy^2z and xyz^2, would be added for symmetry. Then tetrahedron centroid values could be added and the underdetermined systems solved by L^2-minimisation of the new coefficients.

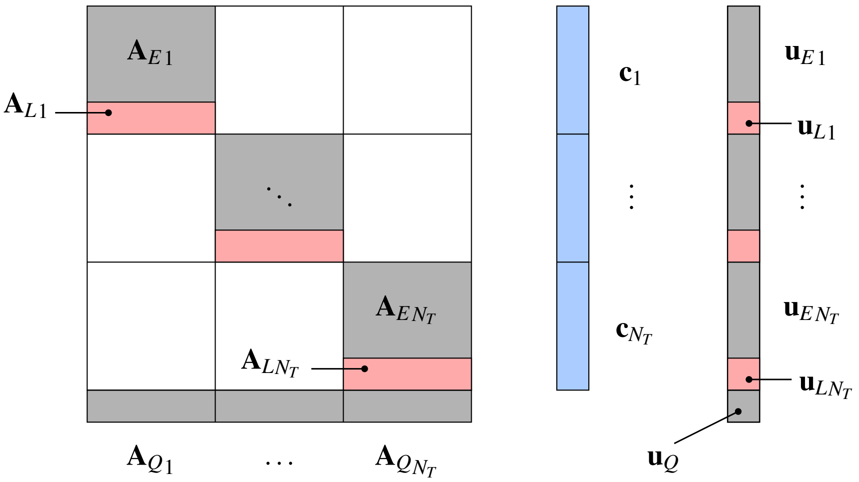


Figure 7.4: Master linear system for the *piecewise polynomial, shared integrated average* concept. Subscript E refers to the exactly constrained point values; L refers to the least squares-treated point values; Q refers to the integrated average constraint.

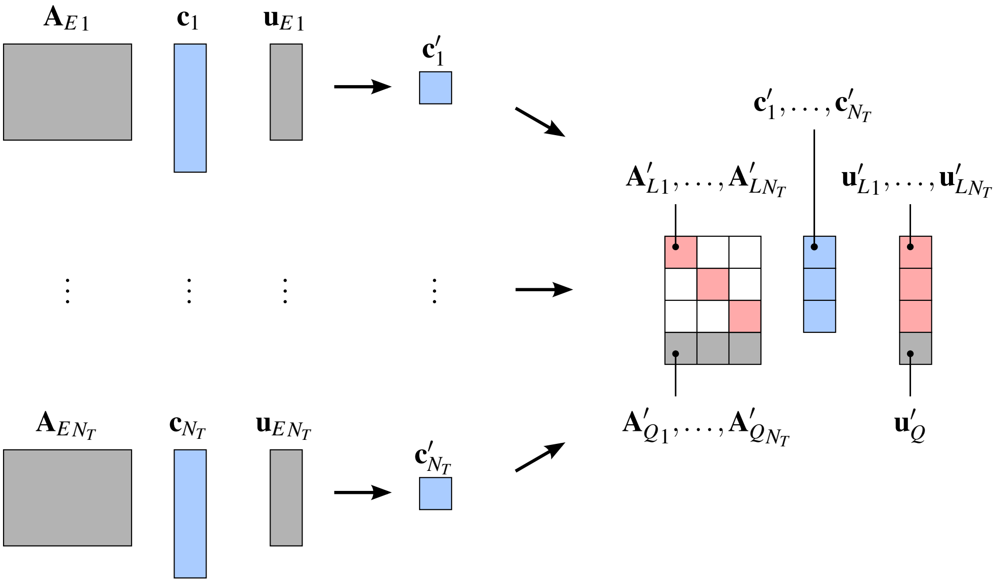


Figure 7.5: Simplex subsystems and reduced master system. Subscript E refers to the exactly constrained point values; L refers to the least squares-treated point values; Q refers to the integrated average constraint.

7.2.2 Testing Method

The algorithms were tested on the cells in Figure 7.2 with arbitrary field values. The values were chosen to emulate a uniform initial condition in the domain interior and a differing Dirichlet condition on the boundary. All interior point values and integrated averages were assigned $u = 0$, while boundary vertices and edge points were assigned $u = 1$.

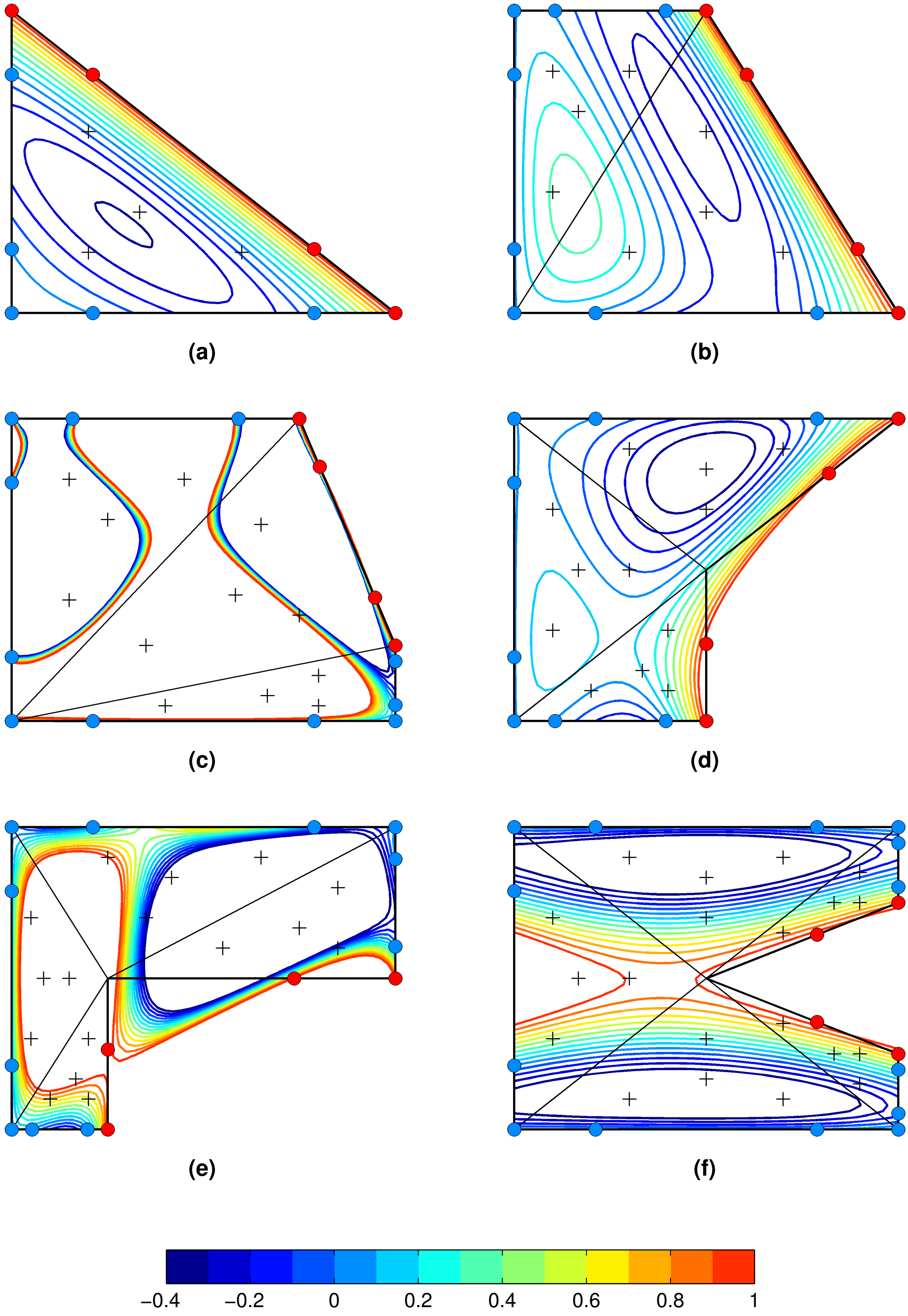
The resulting profiles could be visualised by plotting contours of $U(\xi, \eta)$ over a 51×51 array of ξ and η. It was straightforward to check that these respected the collocation points on vertices and edges.

The quadrilateral cell (Figure 7.2b) was chosen as an exemplar for further checks. To check that the profile respected the integrated average of zero, the sampling resolution was successively doubled and the mean of $U(\xi, \eta)$ calculated each time. Points lying outwith the cell boundary were excluded from the calculation, lending the oblique boundary a staircase representation: as a consequence, first order convergence to zero could be expected.

The effect of proximity of the boundary to the Cartesian grid points was checked by measuring profile boundedness. The top edge of the quadrilateral cell has a nominal length of 0.5. This was successively halved and the maximum and minimum of $U(\xi, \eta)$ within the cell recorded each time. This was done at the original sampling resolution of 51×51 points. The maximum and minimum were expected to converge upon, or at least not exceed, certain limits.

7.2.3 Results

Profiles generated by the two algorithms are shown in Figures 7.6 and 7.7 respectively. Only the range of $-0.4 \leq U \leq 1.0$ is plotted. For the quadrilateral cell (b), measurement and convergence of the integrated average is reported in Table 7.1. The effect of shortening the top edge length is reported in Table 7.2 and demonstrated graphically in Figures 7.8 and 7.9. In the latter figure, the quadrilateral profile appears to converge on that of the triangle (Figure 7.7a). Indeed, the triangular profile's minimum value, $\min(U_0) = -0.4088\text{e}{-01}$, can be used to verify convergence of the minima in Table 7.2. An informal analysis is given in Table 7.3.



(a)

(b)

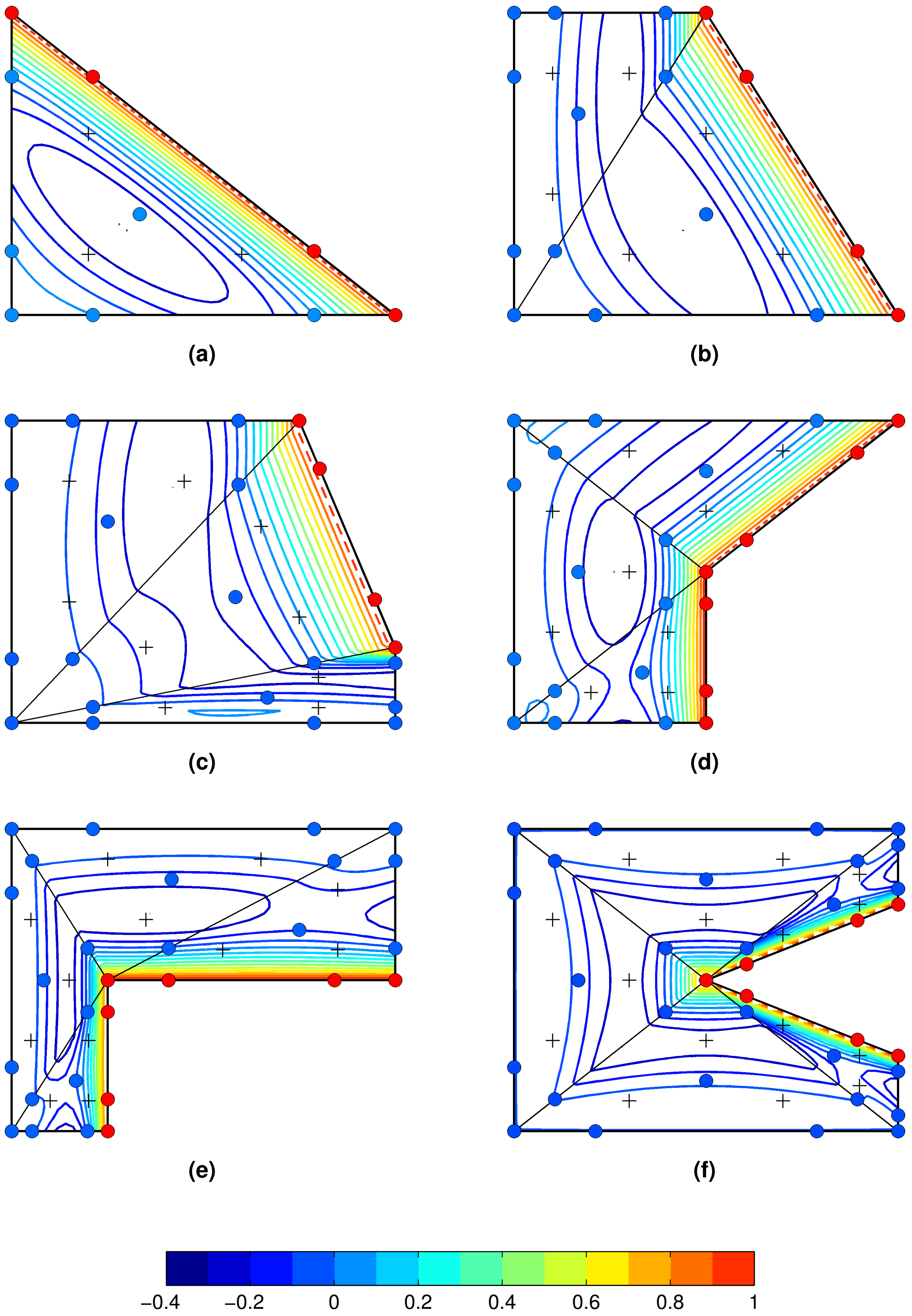
(c)

(d)

(e)

(f)

Figure 7.6: Profiles generated with the *single polynomial, varying basis* algorithm. The reconstruction in Figure 7.6f is always rank-deficient (inexact) regardless of the chosen basis.


(a)

(b)

(c)

(d)

(e)

(f)

Figure 7.7: Profiles generated with the *piecewise polynomial, shared integrated average* algorithm.

Table 7.1: Integrated average check for the quadrilateral cell profile fitting.

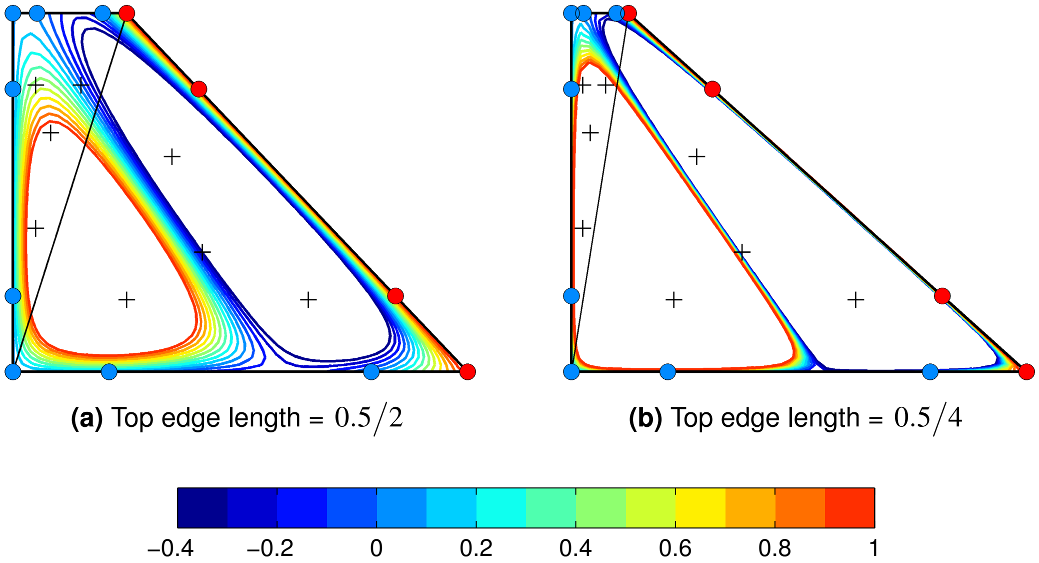
Algorithm	Sampling resolution	mean (U)	Rate
Single polynomial,	51×51	5.2278e−02	–
varying basis	101×101	4.9920e−02	0.067
	201×201	4.8737e−02	0.035
	401×401	4.8144e−02	0.018
Piecewise polynomial,	51×51	6.6839e−03	–
shared integrated average	101×101	3.3372e−03	1.002
	201×201	1.6676e−03	1.001
	401×401	8.3355e−04	1.000

Table 7.2: Boundedness check for the quadrilateral cell profile fitting. The sampling resolution was 51×51.

Algorithm	Top edge length	min (U)	max (U)
Single polynomial,	0.5	−3.5657e−01	1.0016e+00
varying basis	$0.5/2$	−2.1095e+00	2.8188e+00
	$0.5/4$	−1.6261e+01	1.9182e+01
	$0.5/8$	−1.2752e+02	1.3861e+02
Piecewise polynomial,	0.5	−3.2418e−01	1.0000e+00
shared integrated average	$0.5/2$	−3.7985e−01	1.0000e+00
	$0.5/4$	−3.9920e−01	1.0000e+00
	$0.5/8$	−4.0536e−01	1.0000e+00

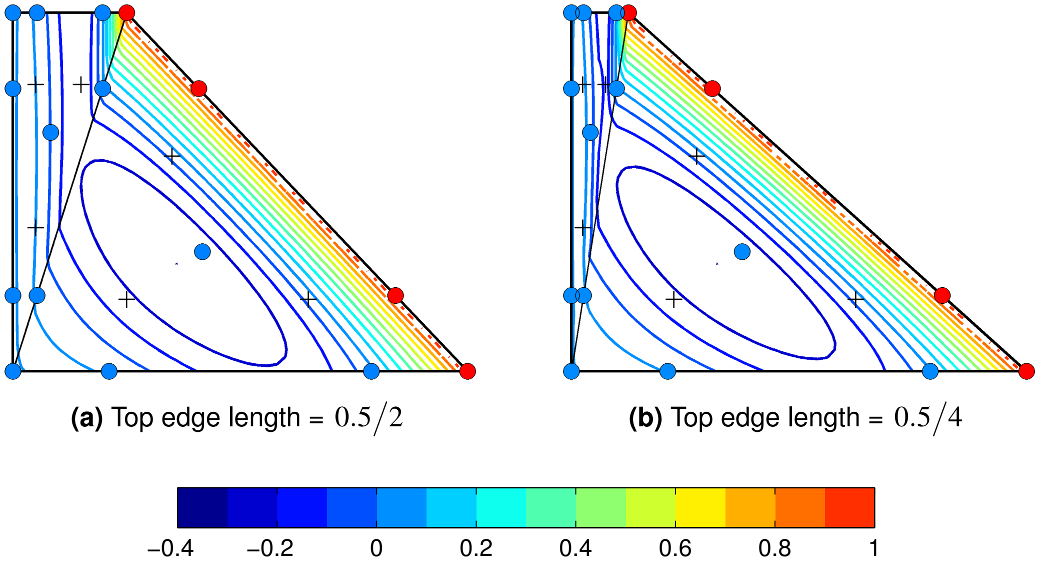
Table 7.3: Convergence of the *piecewise polynomial, shared integrated average* quadrilateral cell profile minimum on the equivalent triangular profile minimum, $\min(U_0) = -0.4088\text{e}{-01}$, as the top edge is progressively shortened. The sampling resolution was 51×51.

Top edge length	$\min(U) - \min(U_0)$	Rate
0.5	8.4672e−02	–
$0.5/2$	2.8999e−02	1.546
$0.5/4$	9.6517e−03	1.587
$0.5/8$	3.4923e−03	1.467


(a) Top edge length = $0.5/2$

(b) Top edge length = $0.5/4$

Figure 7.8: Sensitivity of the *single polynomial, varying basis* quadrilateral cell profile to changes in geometry.


(a) Top edge length = $0.5/2$

(b) Top edge length = $0.5/4$

Figure 7.9: Sensitivity of the *piecewise polynomial, shared integrated average* quadrilateral cell profile to changes in geometry.

7.2.4 Discussion

The first algorithm has failed. It appears to perform particularly badly on cells with short edges (Figures 7.6c and 7.6e), resulting in densely packed contours between regions of extreme minima and maxima. This sensitivity to edge length is seen more clearly in Figure 7.8, where the otherwise acceptable quality of the quadrilateral cell profile deteriorates. Table 7.2 confirms that the minimum and maximum are divergent as the top edge becomes shorter. These problems are reminiscent of Runge's phenomenon in which use of high-order polynomial interpolation leads to nonconvergent, oscillatory behaviour.

Furthermore, there is a problem with the integrated average. It does appear to converge, but not to zero: the continuum value is more like 0.048. The problem does not happen with the triangular cut cell, so it is evidently inconsistent to use the quadrature points of Hammer and Stroud (1956) with polynomials that are not strictly cubic and order-complete.

The second algorithm performs much better. Because all vertex and edge collocation points are respected, the contours are C^0-continuous between simplices. The cell-integrated average converges to zero at the expected first order rate, indicating that it is correctly represented according to Gaussian quadrature. The profile naturally accommodates extreme cell shapes while remaining bounded, as evidenced by the convergence rate in Table 7.3. It is emphasised that the latter analysis is informal. Justification of the observed rate in terms of a Taylor series expansion is not attempted here; the important thing is that the rate is equal to or greater than first order.

Cell shapes naturally handled by the second algorithm include those with cusps in the boundary. There was a concern expressed in §7.1.1 that populating such boundaries with collocation points could destroy the quality of the profile; here it is not an issue. In fact, there is no reason why the algorithm should not handle arbitrary polygons in addition to cut Cartesian cells.

One of the proposed strategies for extending the second algorithm to 3D, whereby coefficients are added to the order-complete profile, is possibly complicated in light of the first algorithm's failure to maintain the correct integrated average using Hammer and Stroud's set of quadrature points. If this strategy were to be taken up, a higher-order set of quadrature points may be needed.

In summary, the *single polynomial, varying basis* algorithm clearly fails to fit profiles that lead to a stable and accurate scheme. It is excluded from further consideration. By contrast, the *piecewise polynomial, shared integrated average* algorithm passes the criteria posed at the beginning of §7.2, with the exception that extension to 3D potentially poses a problem. This second algorithm is put forward for further analysis.

7.3 Scheme Analysis

Up to this point in the thesis, only Cartesian schemes have been analysed in the frequency domain. It is not too difficult to extend the methodology to non-Cartesian schemes if the grid elements can be tessellated in some periodic arrangement. For instance, a pair of isosceles triangles has periodicity in two directions. The Fourier decomposition can be applied to this small repeating unit. Hu *et al.* (1999) analyse the third order discontinuous Galerkin method for the three triangular grid patterns shown in Figure 7.10.

As a matter of interest, and to provide an important benchmark analysis, the IDO/MM–FVM scheme of Ii *et al.* (2005) will be similarly analysed in this section. However, for brevity, only the pattern in Figure 7.10a will be considered, and only at propagation angles of $0^\circ \leq \theta \leq 90^\circ$.

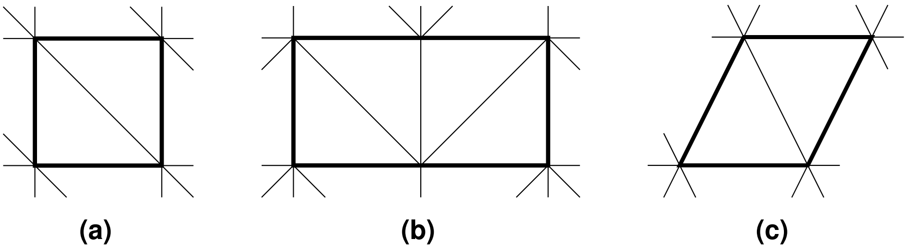


Figure 7.10: Triangular grid patterns considered by Hu *et al.* (1999).

The same grid pattern can be used to check that the *piecewise polynomial, shared integrated average* concept is conducive to a stable scheme. We need not introduce cutting boundaries into the analysis; it is sufficient to form a Cartesian cell which is 'cut' only in the sense that it is triangulated and subjected to the proposed profile fitting algorithm. In other words, the bold box around the two triangles in Figure 7.10 defines the cell, and the piecewise polynomial is fitted according to the enclosed triangles.

The analytical procedure of the previous chapter can be used to determine whether the proposed scheme is stable and accurate. Again it is pointed out that the equivalent semi-Lagrangian (CIP-based) formulation can still be stable even when some of the real eigenvalue parts of the Eulerian (IDO-based) scheme are slightly positive. This is certainly worth keeping in mind for the proposed scheme, where the influence of the supporting point values on the evolution of the cell-integrated average has been manipulated.

A stable and accurate scheme in this interior configuration does not guarantee success for configurations involving boundaries, but it is an important first step.

7.3.1 Formulations

Scheme of li *et al.* (2005)

When the Cartesian IDO/MM-FVM2 scheme was formulated (§5.2), it was noted that weights had to be computed for the vertical and horizontal edge midpoint values in addition to the vertex values. It is the same situation in the triangular IDO/MM-FVM3 scheme, only now there are more point values with associated weight vectors. Figure 7.11 shows the arrangement. Point values with weight vectors are labelled P1 through P7. There are also two cell-integrated averages with individual weight vectors, labelled Q1 and Q2.

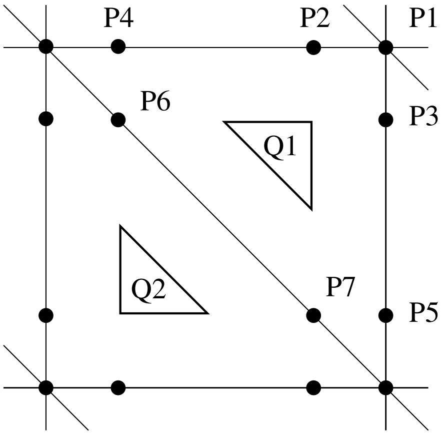


Figure 7.11: Moment arrangement for the scheme of li *et al.* (2005) in a simple grid pattern. Weight vectors must be computed for the values at points labelled P1–7 and at cells labelled Q1 and Q2.

There are now two profile reconstructions represented by two Vandermonde matrices, $\mathbf{A}_1$ and $\mathbf{A}_2$. The order complete cubic basis is used.

$$\mathbf{r} = \begin{pmatrix} 1 & \xi & \xi^2 & \xi^3 & \eta & \xi\eta & \xi^2\eta & \eta^2 & \xi\eta^2 & \eta^3 \end{pmatrix} \tag{7.12}$$

The kth of the two matrices is made up of nine rows $\mathbf{A}_{Ek}$ representing the surrounding point values, and a row $\mathbf{A}_{Qk}$ representing the respective cell-integrated average. The latter can be computed from quadrature points according to Equation (7.4), although the quadrature weights are not rescaled by area since each triangle defines a cell.

$$\mathbf{A}_{E1} = \left(\; \mathbf{r}_{00}; \quad \mathbf{r}_{\bar{a}0}; \quad \mathbf{r}_{0\bar{a}}; \quad \mathbf{r}_{\bar{b}0}; \quad \mathbf{r}_{0\bar{b}}; \quad \mathbf{r}_{\bar{1}0}; \quad \mathbf{r}_{\overline{ba}}; \quad \mathbf{r}_{\overline{ab}}; \quad \mathbf{r}_{0\bar{1}} \; \right) \tag{7.13}$$

$$\mathbf{A}_{E1} = \left(\; \mathbf{r}_{\bar{1}0}; \quad \mathbf{r}_{\overline{ba}}; \quad \mathbf{r}_{\overline{ab}}; \quad \mathbf{r}_{0\bar{1}}; \quad \mathbf{r}_{\overline{1a}}; \quad \mathbf{r}_{\overline{a1}}; \quad \mathbf{r}_{\overline{1b}}; \quad \mathbf{r}_{\overline{b1}}; \quad \mathbf{r}_{\overline{11}} \; \right) \tag{7.14}$$

$$\mathbf{A}_k = \begin{pmatrix} \mathbf{A}_{Ek} \\ \mathbf{A}_{Qk} \end{pmatrix} \tag{7.15}$$

The values a and b are the Gaussian point spacings, ${}^{1}\!/\!{}_{2}(1 \mp {}^{1}\!/\!\sqrt{3})$. The semicolons denote separation of rows instead of separation of columns. A shorthand notation $\mathbf{r}_{\overline{\xi\eta}} := \mathbf{r}|_{-\xi,-\eta}$ has been introduced for compactness.

There is a critical detail in the computation of the weight vectors. By our convention, the velocity direction is upwards and/or to the right. This means that the point fluxes at P1 through P5 should be computed from the first profile, and the point fluxes at P6 and P7 should be computed from the second profile. Thus we arrive at

$${}^{Pj}\mathbf{w}^x = \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\xi}\right|_{\xi_{Pj},\eta_{Pj}} \mathbf{A}_{k(j)}^{-1} \qquad {}^{Qk}\mathbf{w}^x = \sum_{i=1}^{4} \rho_{Qk}^{\langle i\rangle} \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\xi}\right|_{\xi_{Qk}^{\langle i\rangle},\eta_{Qk}^{\langle i\rangle}} \mathbf{A}_k^{-1} \tag{7.16a}$$

$${}^{Pj}\mathbf{w}^y = \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\eta}\right|_{\xi_{Pj},\eta_{Pj}} \mathbf{A}_{k(j)}^{-1} \qquad {}^{Qk}\mathbf{w}^y = \sum_{i=1}^{4} \rho_{Qk}^{\langle i\rangle} \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\eta}\right|_{\xi_{Qk}^{\langle i\rangle},\eta_{Qk}^{\langle i\rangle}} \mathbf{A}_k^{-1} \tag{7.16b}$$

where $k(j) = 1$ when $1 \leq j \leq 5$, and $k(j) = 2$ when $6 \leq j \leq 7$. The coordinates of Pj are respectively $(0,0)$; $(-a,0)$; $(0,-a)$; $(-b,0)$; $(0,-b)$; $(-b,-a)$; $(-a,-b)$. The weights and coordinates of the quadrature points in Qk are computed from Equation (7.3).

Proposed scheme

The analysis of the IDO/MM–FVM scheme with the proposed profile fitting algorithm is complicated by the two-stage reconstruction involved. However, the problem can be approached in the same way that constrained least squares fitting was approached in §3.5.2. Instead of computing coefficient vectors (or slack variables) directly, we compute appropriate matrix pseudoinverses to remove the dependence on right hand side values.

The moment arrangement is represented in Figure 7.12. This time there is only one cell-integrated average, but there are two new points whose values are to be treated by least squares. Their coordinates are $(-{}^{1}\!/\!{}_{3}, -{}^{1}\!/\!{}_{3})$ and $(-{}^{2}\!/\!{}_{3}, -{}^{2}\!/\!{}_{3})$ respectively.

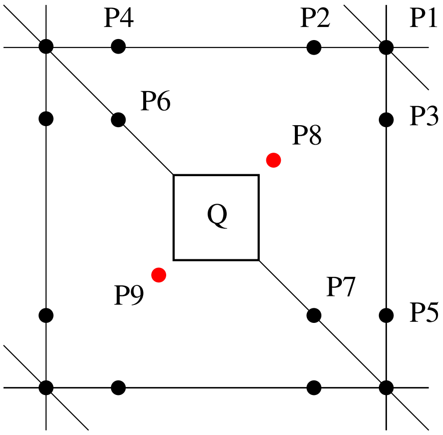


Figure 7.12: Moment arrangement when the proposed profile fitting algorithm is applied to a simple triangulation of Cartesian cells. The values at the points in red are to be treated by least squares.

The goal is to compute a pseudoinverse $\mathbf{M}_k$ for each triangle such that

$$ {}^{Pj}\mathbf{w}^x = \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\xi}\right|_{\xi_{Pj},\eta_{Pj}} \mathbf{M}_{k(j)} \qquad {}^{Q}\mathbf{w}^x = \sum_{k=1}^{2}\sum_{i=1}^{4} \rho_{Qk}^{\langle i\rangle} \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\xi}\right|_{\xi_{Qk}^{\langle i\rangle},\eta_{Qk}^{\langle i\rangle}} \mathbf{M}_k \tag{7.17a}$$

$$ {}^{Pj}\mathbf{w}^y = \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\eta}\right|_{\xi_{Pj},\eta_{Pj}} \mathbf{M}_{k(j)} \qquad {}^{Q}\mathbf{w}^y = \sum_{k=1}^{2}\sum_{i=1}^{4} \rho_{Qk}^{\langle i\rangle} \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\eta}\right|_{\xi_{Qk}^{\langle i\rangle},\eta_{Qk}^{\langle i\rangle}} \mathbf{M}_k \tag{7.17b}$$

where, this time, $k(j) = 1$ when $j = (1,2,3,4,5,8)$, and $k(j) = 2$ when $j = (6,7,9)$.

We start by defining the various systems in Figure 7.4. The two matrices $\mathbf{A}_{Ek}$ representing exactly constrained point values are the same as those defined in Equations (7.13) and (7.14). The two rows representing the least squares-treated centroid values are given by

$$\mathbf{A}_{L1} = \mathbf{r}|_{-1/3,-1/3} \tag{7.18}$$

$$\mathbf{A}_{L2} = \mathbf{r}|_{-2/3,-2/3} \tag{7.19}$$

and the two rows $\mathbf{A}_{Qk}$ representing contributions to the cell-integrated average are computed from Equation (7.5). To define the transformed systems in Figure 7.4, we require bases $\mathbf{Q}_{Ek,2}$ for the null spaces of $\mathbf{A}_{Ek}$. Using a QR factorisation, let

$$\mathbf{A}_{Ek}^{\mathsf{T}} = \mathbf{Q}_{Ek}\mathbf{R}_{Ek} \equiv \begin{pmatrix} \mathbf{Q}_{Ek,1} & \mathbf{Q}_{Ek,2} \end{pmatrix} \begin{pmatrix} \mathbf{R}_{Ek,1} \\ \mathbf{0} \end{pmatrix} \tag{7.20}$$

where $\mathbf{Q}_{Ek,1}$ is the first nine columns of $\mathbf{Q}_{Ek}$, and $\mathbf{Q}_{Ek,2}$ is the last column. The elements $\mathbf{A}'_{Lk}$ and $\mathbf{A}'_{Qk}$ follow from Equations (7.11) and (7.10) respectively. The transformed matrix $\mathbf{A}'_L$ and row $\mathbf{A}'_Q$ are finally assembled as

$$\mathbf{A}'_L = \begin{pmatrix} \mathbf{A}'_{L1} & 0 \\ 0 & \mathbf{A}'_{L2} \end{pmatrix} \tag{7.21}$$

$$\mathbf{A}'_Q = \begin{pmatrix} \mathbf{A}'_{Q1} & \mathbf{A}'_{Q2} \end{pmatrix} \tag{7.22}$$

The pseudoinverse matrices $\mathbf{M}_k$, which are related to coefficient vectors and right hand sides by $\mathbf{c}_k = \mathbf{M}_k \mathbf{u}_k$, may now be computed as follows. Using the Kronecker delta,

$$\delta_{ij} := \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{7.23}$$

1. For each triangle k', compute a matrix $\mathbf{M}_{1Ek'}$ such that $\mathbf{x}_{Ek'} = \mathbf{M}_{1Ek'} \mathbf{u}_{Ek'}$, where $\mathbf{x}_{Ek'}$ is the particular solution to the untransformed exactly constrained system.

$$\mathbf{M}_{1Ek'} = \mathbf{Q}_{Ek',1} \mathbf{R}_{Ek',1}^{-\mathsf{T}} \tag{7.24}$$

2. Factorise $\left(\mathbf{A}'_Q\right)^{\mathsf{T}}$ into an orthogonal matrix $\mathbf{Q}_Q$ and column vector $\mathbf{R}_Q$. These are partitioned according to the first column and first element, respectively.

$$\mathbf{A}_Q = \mathbf{Q}_Q \mathbf{R}_Q \equiv \begin{pmatrix} \mathbf{Q}_{Q1} & \mathbf{Q}_{Q2} \end{pmatrix} \begin{pmatrix} \mathbf{R}_{Q1} \\ \mathbf{0} \end{pmatrix} \tag{7.25}$$

3. Factorise $\mathbf{A}'_L \mathbf{Q}_{Q2}$ into an orthogonal matrix $\mathbf{Q}_L$ and column vector $\mathbf{R}_L$.

$$\mathbf{A}'_L \mathbf{Q}_{Q2} = \mathbf{Q}_L \mathbf{R}_L \tag{7.26}$$

4. Constrained least squares fitting can now be applied to the transformed system. The particular solution to the exactly constrained part is given by $\mathbf{x}' = \mathbf{Q}_{Q1} \mathbf{R}_{Q1}^{-\mathsf{T}} \mathbf{u}'_Q$. Expand using Equations (7.11) and (7.24), and extract matrices representing the linear operations on $\mathbf{u}_Q$ and $\mathbf{u}_{Ek'}$.

$$\mathbf{M}_{2Q} = \mathbf{Q}_{Q1} \mathbf{R}_{Q1}^{-\mathsf{T}} \tag{7.27}$$

$$\mathbf{M}_{2Ek'} = -\mathbf{M}_{2Q} \mathbf{A}_{Qk'} \mathbf{M}_{1Ek'} \tag{7.28}$$

5. The least squares solution to the remaining part is given by $\mathbf{y}' = \mathbf{R}_L^{-1} \mathbf{Q}_L^{\mathsf{T}} \left(\mathbf{u}'_L - \mathbf{A}'_L \mathbf{x}'\right)$. Expand this using Equations (7.10) and (7.24), and extract matrices representing the

linear operations on $\mathbf{u}_Q$, $\mathbf{u}_{Ek'}$ and $\mathbf{u}_{Lk'}$.

$$\mathbf{M}_{3Q} = -\mathbf{R}_L^{-1}\,\mathbf{Q}_L^{\intercal}\,\mathbf{A}'_L\,\mathbf{M}_{2Q} \tag{7.29}$$

$$\mathbf{M}_{3Ek'} = -\mathbf{R}_L^{-1}\,\mathbf{Q}_L^{\intercal}\left[\begin{pmatrix} \delta_{1k'} \\ \delta_{2k'} \end{pmatrix}\mathbf{A}_{Lk'}\mathbf{M}_{1Ek'} + \mathbf{A}'_L\mathbf{M}_{2Ek'}\right] \tag{7.30}$$

$$\mathbf{M}_{3Lk'} = \mathbf{R}_L^{-1}\,\mathbf{Q}_L^{\intercal}\begin{pmatrix} \delta_{1k'} \\ \delta_{2k'} \end{pmatrix} \tag{7.31}$$

6. Slack variables are given by the constrained least squares solution to the transformed system, $\mathbf{c}' = \mathbf{x}' + \mathbf{Q}_{Q2}\,\mathbf{y}'$. Expand and extract matrices.

$$\mathbf{M}_{4Q} = \mathbf{M}_{2Q} + \mathbf{Q}_{Q2}\,\mathbf{M}_{2Q} \tag{7.32}$$

$$\mathbf{M}_{4Ek'} = \mathbf{M}_{2Ek'} + \mathbf{Q}_{Q2}\,\mathbf{M}_{3Ek'} \tag{7.33}$$

$$\mathbf{M}_{4Lk'} = \mathbf{Q}_{Q2}\,\mathbf{M}_{3Lk'} \tag{7.34}$$

7. For each triangle k, substitute $\mathbf{c}'_k$ back into Equation (7.9). Expand and extract matrices.

$$\mathbf{M}_{5Qk} = \mathbf{Q}_{Ek,2}\,\mathbf{M}_{4Q} \tag{7.35}$$

$$\mathbf{M}_{5Ekk'} = \delta_{kk'}\mathbf{M}_{1Ek} + \mathbf{Q}_{Ek,2}\,\mathbf{M}_{4Ek'} \tag{7.36}$$

$$\mathbf{M}_{5Lkk'} = \mathbf{Q}_{Ek,2}\,\mathbf{M}_{4Lk'} \tag{7.37}$$

8. Assemble $\mathbf{M}_k$ from $\mathbf{M}_{5Qk}$, $\mathbf{M}_{5Ekk'}$ and $\mathbf{M}_{5Lkk'}$, superposing columns representing shared moments.

7.3.2 Method

The same method as §6.3 was used. Matrices of weights were computed and subjected to the frequency-domain analysis outlined in Chapter 3. Eigenvalues were sampled in steps of $\Delta K = \pi/50$ up to $K_{\max} = 6\pi\left(1, \sqrt{5}, \sqrt{2}\right)$ for the three respective angles, the 6π factor being appropriate for fourth and fifth order multi-moment methods to ensure that the full range of eigenvalues was represented.

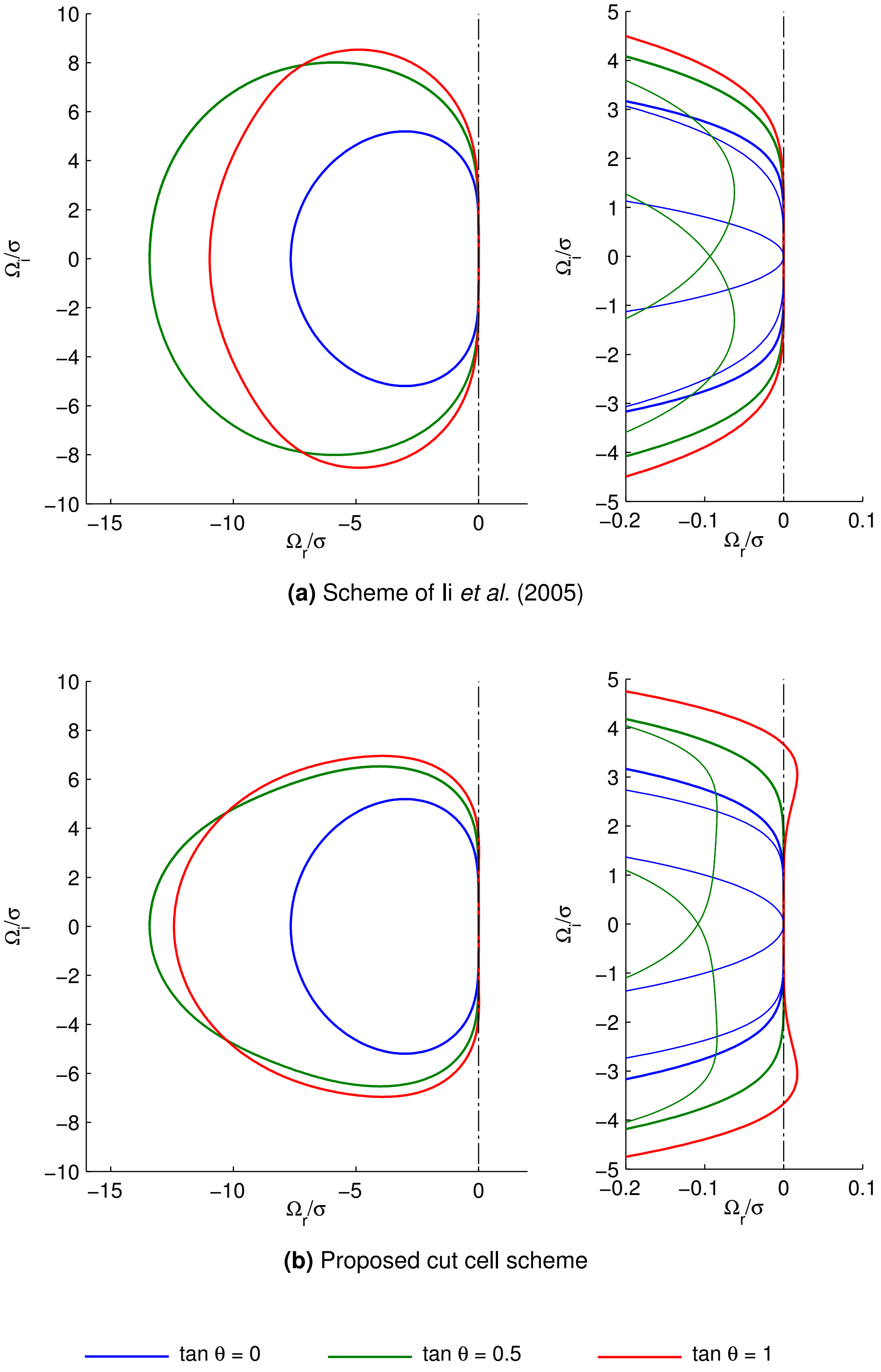
7.3.3 Results

Maximum real eigenvalue parts and convergence rates of the two schemes are given in Table 7.4. Positive real eigenvalue parts are highlighted in red, but it can be seen that all convergence rates are acceptable. Figure 7.13 shows the spectra plotted on the complex plane, with a close-up of both physical and spurious modes near the origin.

Polar plots and error plots have not been generated, since the focus of the current investigation is on stability and convergence, not on anisotropy.

Table 7.4: Critical spectral details for fourth order IDO/MM–FVM schemes modelling triangular arrangements. Convergence rates have been calculated from Equation (3.29) with $K_a = \pi/50$.

Scheme	$\tan\theta$	Max real Ω/σ	Convergence rate	
			Dissipation	Dispersion
Ii *et al.*	0.0	$-1.036\text{e}{-15}$	4.984	3.999
(2005)	0.5	$-5.940\text{e}{-15}$	4.919	4.000
	1.0	$1.799\text{e}{-15}$	5.002	4.000
Proposed	0.0	$3.708\text{e}{-15}$	4.989	3.999
	0.5	$3.297\text{e}{-05}$	6.075	4.001
	1.0	$1.715\text{e}{-02}$	5.220	3.998



(a) Scheme of li *et al.* (2005)

(b) Proposed cut cell scheme

Figure 7.13: Spectra of fourth order IDO/MM–FVM schemes modelling triangular arrangements. (Left) full range of K; (right) detail near the origin. The latter plots include spurious modes which are shown as thin lines.

7.3.4 Discussion

As expected, the scheme of Ii *et al.* (2005) is both stable and accurate for the three propagation directions. In comparison, while the proposed scheme has similarly-formed spectra which also show the correct rates of convergence, it has eigenvalues which deviate into the positive real half of the complex plane. This scheme may therefore be considered unstable in an Eulerian formulation. This may not be technically true at certain Courant numbers under certain time integration schemes, but as discussed on p. 63 we insist on all eigenvalues having zero or negative real parts for the scheme to be of practical use.

This deviation of the physical mode into the positive half of the real plane is reminiscent of the spectra of the volume- and surface- integrated average method with second order central time evolution converting formula (VSIAM2 SOC–TEC), presented in §5.4.1. In that scheme, the positive real eigenvalue parts are not an issue in the context of a semi-Lagrangian formulation: the more important property is that the nominal order of convergence is maintained at all propagation angles. One might ask whether a similar principle applies to the proposed scheme. Of course, a CIP/MM–FVM scheme is not purely semi-Lagrangian but a hybrid formulation, and so any concession on stability would be conditional; there would still be a restriction on the maximum allowable Courant number.

It is not possible to tell from the previous analysis whether the proposed scheme would be acceptable in a CIP-based formulation. There is no choice but to analyse the fully discrete CIP/MM–FVM system or test it in the time domain.

Analysis of the fully discrete system is complicated by the existence of Eulerian operators that must be either (a) integrated in Runge-Kutta stages – see for example Ii and Xiao (2007) – or (b) calculated from face fluxes, which in turn are approximated from quadrature of the upwind profiles along straight-line trajectories (Ii *et al.*, 2005). Approach (a) is more general and could be verified using stability criteria already determined for IDO and other one- and two-dimensional multi-moment methods (§3.3). However, it depends on multiple profile reconstructions which means the analysis would have to encompass multiple cells. Approach (b) only depends on a single reconstruction, but it is not easily verifiable from previous analyses. Neither approach is straightforward enough to warrant inclusion in the scope of the present work.

Even if analysis of the fully discrete system were performed and conditional stability established, the analysis would not guarantee stability for other cut cell configurations. In any case, further frequency-domain analysis would be overkill. It has been established that the proposed scheme is fourth order-accurate, which is good enough for our purposes; extra detail regarding accuracy with respect to wavenumber and propagation direction is not needed. It is suggested that from this point one should proceed straight to time-domain testing.

Testing brings new challenges. Besides creation of the cut cell architecture, there is an important detail to attend to: an algorithm is needed to select the correct simplex profile for any given departure point. Fortunately, the fact that the simplices are compartmentalised by Cartesian cells can be exploited to solve the problem. A novel, efficient algorithm has been developed as part of the present work. Details of the algorithm are omitted from this chapter, but the interested reader is referred to Appendix C.

It remains to develop the architecture and proceed with time-domain testing. These activities are deferred to further work. Tests might begin with all cells triangulated, but remaining interior, so as to exercise the cut cell scheme. Some possible arrangements are given in Figure 7.14. Boundary conditions would remain periodic to support a linear wave travelling at 45°.

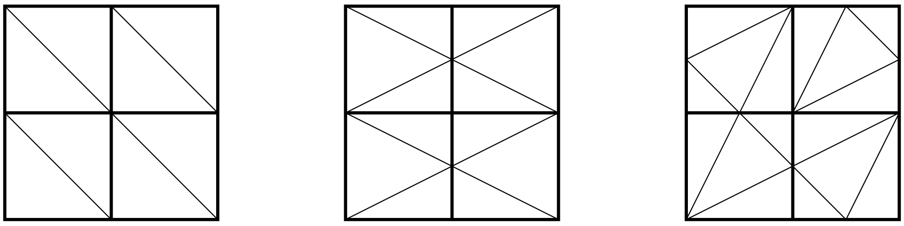


Figure 7.14: Possible cell triangulations for testing the proposed cut cell scheme.

If stability can be proven in these simple arrangements, it is grounds for more complex testing to be carried out. Further tests would involve solid boundaries and Cartesian grid refinement to check that stable, convergent solutions hold in the presence of boundary conditions.

7.4 Conclusions

At the beginning of this chapter, it was noted that the design of a cut cell multi-moment scheme depends mostly on its profile reconstruction. Two profile fitting algorithms were presented. The first algorithm, which attempts to fit a single polynomial over the entire cut cell, has no place in a numerical scheme. It is highly sensitive to discontinuities or cusps in cell geometry, with the profile showing unbounded behaviour as cusps are brought close together. The second algorithm, which assigns each simplex a profile but forces intra-cell simplices to share a cut cell-integrated average, is acceptable insofar that profiles remained bounded and the cell-integrated average is correctly maintained.

Analysis of a multi-moment scheme based on the latter algorithm – with the scheme modelling linear advection on a simple grid of triangulated cells – reveals eigenvalues with positive real parts. Unfortunately this means the scheme is practically unstable in an Eulerian formulation. Whether or not it would be stable in CIP–MM/FVM's hybrid formulation remains inconclusive: analysis or testing of the fully discrete system would need to be carried out.

Chapter 8

Conclusion

8.1 Summary of Findings

The first chapter began by identifying a need to implement a multi-moment scheme, such as CIP, on a boundary non-conforming grid. However, the literature review showed that not only does one have to be careful when adapting high-order methods to irregular grids; it is not even obvious which of the several multi-moment schemes would be best for adaptation, or whether a multi-moment scheme would be appropriate in the first place. That is why it is important to first appraise existing multi-moment schemes before attempting to implement one on a cut cell grid.

Four research objectives were specified at the end of the literature review (§2.5) and addressed in turn over Chapters 4–7. The most significant findings are summarised as follows.

1. Multi-moment methods are competitive with the current ‘gold standard’ of compact methods, the discontinuous Galerkin (DG) method. For linear advection, piecewise-continuous multi-moment methods such as the constrained interpolation profile (CIP) and interpolated differential operator (IDO) methods actually have higher orders of accuracy than DG (that is, $2N - 1$ versus N) for the same memory cost. This is a great advantage in the context of smooth problems. Although the *continuous* Galerkin method could compete in such a comparison, it is less suited to advection problems due to its globally implicit formulation. On the other hand, when comparing a *discontinuous* multi-moment method with DG, DG will be more accurate (17% or greater) but less stable (51% or lower) for the same order of accuracy and memory cost. Furthermore, multi-moment spectra are identical to DG spectra in one dimension, although the orders of accuracy do not correspond. In two dimensions, the spectra are similar but not identical.
2. For split multidimensional schemes, there are opportunities for reducing the memory cost by substituting the 1D multi-moment operators with alternative ones. However, one such scheme – Type-M CIP (Takewaki and Yabe, 1987) – is formally only second order in oblique directions, even though this is not acknowledged in the literature. Another split scheme – the CIP conservative semi-Lagrangian scheme with time evolution converting (TEC) formula (Xiao *et al.*, 2006) – is remarkable for showing high accuracy in diagonal directions but also positive real eigenvalue parts. In other words, it is stable only in a

semi-Lagrangian formulation. For unsplit schemes, the number of coefficients comprising the multidimensional profile can be reduced. This invites both isotropy and economy of storage; Type-A CIP (Yabe *et al.*, 1991) is a case in point. Schemes that do not economise on moments tend to be anisotropic, with higher accuracy but enlarged spectra (and hence stricter stability conditions) in diagonal directions. In summary, one can, and should, aim to minimise both memory cost and numerical isotropy when designing a multidimensional scheme.

3. A new conservative, fourth order scheme has been conceived for the uncut background of a Cartesian cut cell grid. While a scheme with a more standard 'tensor product' basis needs to store nine moments per cell, the new scheme only needs to store six. It takes the same approach as Type-A CIP in that it uses a reduced basis, but crucially it uses more coefficients than the order-complete basis so that cubic flux polynomials may be constructed over each face. The resulting scheme has excellent isotropy. An important point to be made is that nominal-order flux reconstructions are mandatory if a hybrid multi-moment/finite volume method is used.
4. A stable *Eulerian* fourth order multi-moment scheme could not be conceived for cut cells. It may be hypothesised that such a scheme does not exist. Fitting a single polynomial over the cut cell is out of the question due to unbounded behaviour, while a compromise – fitting polynomials over simplices while treating the cut cell as a finite volume – leads to eigenvalues with positive real parts. This does not necessarily mean the latter scheme is unstable in a hybrid semi-Lagrangian/Eulerian formulation, however. There is an encouraging fourth-order convergence of the eigenvalues for a simple grid arrangement and diagonal wave propagation.

8.2 Further Work

As discussed at the end of §7.3.4, the current priority is to establish whether or not the proposed cut cell scheme could be stable as a CIP/MM–FVM scheme. Time-domain testing is more appropriate than frequency-domain analysis at this stage. The next subsections follow on from that discussion, suggesting work that would need to be done to get a multi-moment cut cell solver to the point where it might usefully simulate nonlinear flow equations.

A road map for further work is summarised as follows.

1. Check the stability of the cut cell scheme proposed in Chapter 7 in a hybrid (CIP/MM–FVM) formulation.
2. Couple the cut cell/simplicial multi-moment scheme with the Cartesian scheme from Chapter 6, using grids containing both uncut and cut cells.
3. Introduce solid boundaries, implementing scalar boundary conditions such as Neumann conditions on the wall.

4. Implement nonadvection operators.
5. Model the Euler equations, implementing a free slip boundary condition.
6. Introduce curved boundaries. The architecture must respect the curved geometry.
7. Implement and test local grid refinement.

Should the proposed cut cell scheme (Item 1) turn out to be unconditionally unstable, one may, as a fallback strategy, investigate the use of simplicial cells with locally implicit time integration before continuing. This strategy is discussed in §8.2.1.

We will stop short of considering work needed to model the Navier–Stokes equations. The reason is that the viscous stability condition is much more stringent than the advection stability condition – it scales inversely with Δx^2, not Δx – making it more appropriate to advance the viscous term by an implicit time integration. Implicit procedures tend to increase the complexity of the method and the computational cost. Even if we choose to use an explicit time integration instead, viscous flows are often incompressible and require the pressure field to be solved implicitly.

Throughout this work, explicit time-stepping has been assumed. That is, the calculation of u_i^{n+1} is supported by grid values at time level n or earlier; a Runge-Kutta scheme computes stage $(k+1)$ in terms of stage k or lower. In an implicit time scheme, the calculation of u_i^{n+1} is further supported by grid values at time level $(n+1)$. Equivalently, a ‘diagonally’ implicit Runge-Kutta scheme computes stage $(k+1)$ in terms of stage $(k+1)$ or lower. The stability regions of implicit schemes often include the entire negative real region of the complex plane; they are said to be A-stable. The spectrum of the space operator is then free to be any size, solving our stability problems.

However, because all new grid values are now dependent on one another, a global linear system of equations must be formed. Inverting the global matrix is expensive and needs to be done at every time step when modelling moving boundaries. Furthermore, while it is possible to reformulate multi-moment schemes to be implicit – see, for example, Imai and Aoki (2006a) – irregular high-order stencils will greatly complicate the global linear system. Ferziger and Perić (1996) recommend a deferred correction approach whereby a low-order stencil appears in the linear system and a high-order correction is added explicitly in an outer iteration. In any case, the increase in complexity does not warrant an extended discussion here.

8.2.1 Locally Implicit Time Integration

Should the proposed cut cell scheme turn out to be unconditionally unstable, what then? The hypothesis that cut cells may serve as finite volumes in a multi-moment framework will have been defeated. The next best solution will be to decompose cut cells into their constituent simplices as before, but this time have the simplices serve as finite volumes with their own interpolation profiles. This means implementing the scheme of Ii *et al.* (2005) without modification. The approach was first suggested in §7.2.1. However, a major weakness was exposed: in the triangulation of cut cells, we can end up with sliver cells being created. These in turn would put a severe restriction on the allowable stable time step. One remedy might be to merge the offending cells with Cartesian neighbours before re-triangulating, but this would increase the complexity and potentially create more problems than it solves.

Another remedy could be to use an implicit time scheme instead of an explicit one. But as explained above, forming a high-order global linear system would be complicated and computationally expensive. Here the cut cell paradigm may come to the rescue after all. The time integration scheme does not have to be the same everywhere in the grid: it might be explicit for Cartesian cells and implicit within cut cells. We could then expect the cut cell's allowable Courant number to be dictated by the dimensions of the cut cell, rather than the dimensions of its constituent simplices. This technique of *locally implicit* time integration has already been introduced (Piperno, 2006) and analysed (Dolean *et al.*, 2010) in the context of DG grids whose local refinement would lead to overly stringent stability conditions. However, it has not yet been applied to Cartesian cut cells. The compartmentalisation of simplices by the cut cells makes them ideal candidates for locally implicit treatment. Of course, the compartments themselves should not be too small, and so the usual strategy of merging small cut cells with their neighbours still applies.

Locally implicit time integration is not the same as local time stepping. In the latter technique, all cell values are advanced at their maximum allowable Courant number, which means there will be different time steps in differently sized cells instead of a global time step. Local time stepping is only suitable when a steady-state solution is sought.

8.2.2 Uncut/Cut Cell Scheme Coupling

Whether the proposed cut cell CIP/MM–FVM scheme is proven to be stable, or a locally unstructured scheme with implicit time integration is pursued instead, the next task will be to test the scheme on a grid containing both uncut and cut cells. The scheme from Chapter 6 can be used for the uncut cells. To begin with, the interior, periodic arrangements of Figure 7.14 may be tested with some of the cells changed from cut to uncut.

It will then become necessary to test the effect of boundary conditions. Figure 8.1 shows an arrangement that will support the simulation of a periodic plane wave, propagating at angle

$\theta = \tan^{-1} 0.5 \approx 26.6°$, with Neumann conditions on the solid walls. The Neumann condition at point $\mathbf{r}_b$ must be implemented in the linear system for profile $\Phi(\mathbf{r})$ according to the chain rule

$$\left.\frac{\partial \phi}{\partial \mathbf{n}}\right|_{\mathbf{r}_b} = \nabla\Phi(\mathbf{r}_b) \cdot \mathbf{n}(\mathbf{r}_b) \tag{8.1}$$

where $\mathbf{n}$ is the surface normal unit vector.

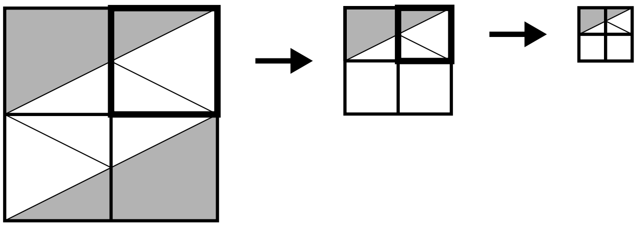


Figure 8.1: Possible grid for testing both uncut and cut cell schemes. Solid regions are shown shaded. Refinement of the cut Cartesian cells will naturally give rise to uncut cells and further cut cells.

8.2.3 Modelling the Euler Equations

If nonlinear equations such as the Euler equations are to be modelled with semi-Lagrangian operators – i.e. if a CIP/MM–FVM scheme is used – it will become necessary to implement nonadvective operators as well. This is because conservative advection terms are split into linear advection and nonadvection parts according to Equation (2.1). Even if the fully Eulerian IDO/MM–FVM method is chosen instead, it will become necessary to model nonadvective viscous and pressure gradient terms when it comes to the Navier-Stokes equations.

Recalling Equation (2.1), in the splitting of the conservative flux,

$$\nabla \cdot (\phi \mathbf{V}) = \mathbf{V} \cdot \nabla\phi + \phi\nabla \cdot \mathbf{V}$$

the nonadvective divergence operator on the right is treated in a separate fractional step. Whereas an advective operator is upwind-biased, a nonadvective operator should be supported from all directions. This leads to the problem of how one might differentiate a profile at the location of interest if that location lies between cells. One strategy could be to extend the 1D nonadvective interpolated differential operator of Aoki (1997) to multidimensions. Thus, a new polynomial profile is constructed that spans both upwind and downwind cells, and as a consequence is of higher degree than the profile for the advection operator. However, this strategy would run into difficulties at intersections between more than two cells, for instance at vertices. A more robust approach would be to follow the example of Ollivier-Gooch and Van Altena (2002). To calculate nonadvective fluxes over faces, an average of cell profiles on either side of the face is taken. For edges and vertices, a weighted average of several cells would be needed.

The Euler equations may be tested with inlet/outlet boundary conditions replacing periodic conditions and an arbitrary scalar field replacing the initial wave function. The grid of Figure 8.1 may be reused. However, because the velocity field is now modelled numerically instead of imposed exactly, it becomes necessary to introduce a free slip wall boundary condition:

$$\begin{aligned} V_n &= 0, & \frac{\partial \mathbf{V}_t}{\partial \mathbf{n}} &= \mathbf{0}; \\ V_n &:= \mathbf{V} \cdot \mathbf{n}, & \mathbf{V}_t &:= \mathbf{V} - V_n \mathbf{n} \end{aligned} \tag{8.2}$$

where **V** is the velocity vector.

A complication arises because **V** will be treated computationally in terms of its Cartesian components. Unlike scalar boundary conditions, Equations (8.2) imply that these components are coupled together in the profile. Therefore, the respective matrix rows will be two or three times the width of those corresponding to interior points or points with scalar boundary conditions. The Vandermonde matrix will then need to be enlarged by a factor of two or three to accommodate these rows. Fortunately, the enlarged system need not be solved all at once. We can apply the divide-and-conquer approach of §7.2.1 to separate the smaller rows and reduce the solution space. Combining the proposed profile fitting algorithm with such boundary conditions may lead to nested linear systems that are best implemented as recursive data structures, perhaps as part of an object-oriented design.

More complex solid boundaries may be introduced into the domain, such as circles, aerofoils, curved channels, and so on. Such cases have standard solutions that can be used to validate the cut cell method. In all cases, it should be possible to prove fourth order convergence with grid refinement. It is cautioned that boundary points and surface normal vectors must respect the exact geometry for reasons discussed in §2.3.1.

Finally, when simulating complex Euler flows with open boundaries, it is most useful to be able to control local grid resolution. Figure 1.4 shows the example of an aerofoil. Every time a 2D cell is refined, it gives rise to four smaller cells; a 3D cell gives rise to eight. These recursive data structures are known as quadtree or octree structures. Implementations are found in De Zeeuw and Powell (1993) and Liang *et al.* (2007). Additionally, locally implicit time integration could be applied to the refined cells in order to stop them reducing the stability limit.

8.3 Closing Remarks

One can expect compact, high-order methods such as the discontinuous Galerkin method to become more prominent and perhaps come to dominate the next generation of computational fluid dynamics software. Multi-moment methods are less well known, but the present work has shown that these methods are competitive with the DG method and indeed are preferable in some situations. It is certainly worth putting them to practical use and have them simulate flows in domains with arbitrary boundaries.

Much work will be needed to bring a multi-moment cut cell solver to a level where it can usefully simulate complex flows. The role of the present work has been to lay the groundwork and establish what will and will not succeed. The piecewise-continuity of multi-moment methods makes them more economical than their discontinuous counterparts, but it also makes them more difficult to implement in unusual grids. The proposed cut cell scheme is a case in point. It also means greater care must be taken when manipulating split operators or profile bases.

Nevertheless, it is possible to optimise the memory cost and isotropy of multi-moment arrangements while respecting stability and accuracy requirements. It is hoped that in this sense multi-moment schemes will enjoy greater popularity among the numerical community in the future.

Appendix A

Space Operator Weights

A.1 One Spatial Dimension

First Order

$$\begin{aligned} \mathbf{W}_0 &= {}^{F}w_0 = 1 \\ \mathbf{W}_{\bar{1}} &= {}^{F}w_{\bar{1}} = -1 \end{aligned} \tag{A.1}$$

Third Order

Interpolated Differential Operator Method (IDO)

Moments: u, u_ξ.

$$\begin{aligned} \mathbf{W}_0 &= \begin{pmatrix} {}^{I}w_0 & {}^{I}w_{\xi 0} \\ {}^{I\xi}w_0 & {}^{I\xi}w_{\xi 0} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -6 & 4 \end{pmatrix} \\ \mathbf{W}_{\bar{1}} &= \begin{pmatrix} {}^{I}w_{\bar{1}} & {}^{I}w_{\xi \bar{1}} \\ {}^{I\xi}w_{\bar{1}} & {}^{I\xi}w_{\xi \bar{1}} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 6 & 2 \end{pmatrix} \end{aligned} \tag{A.2}$$

IDO Finite Volume Method (IDO–FVM2)

Moments: $\overline{{}^{V}u}$, $\overline{{}^{S}u}$.

$$\begin{aligned} \mathbf{W}_0 &= \begin{pmatrix} {}^{V}w_{V0} & {}^{V}w_{S0} \\ {}^{S}w_{V0} & {}^{S}w_{S0} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -6 & 4 \end{pmatrix} \\ \mathbf{W}_{\bar{1}} &= \begin{pmatrix} {}^{V}w_{V\bar{1}} & {}^{V}w_{S\bar{1}} \\ {}^{S}w_{V\bar{1}} & {}^{S}w_{S\bar{1}} \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 0 & 2 \end{pmatrix} \end{aligned} \tag{A.3}$$

Multi-Moment Constrained Finite Volume Method (MCV3)

Moments: $\overline{{}^{V}u}$, $\overline{{}^{S-}u}$ (upwind boundary values), $\overline{{}^{S+}u}$ (downwind boundary values).

$$\begin{aligned} \mathbf{W}_0 &= \begin{pmatrix} 0 & 1 & 0 \\ -6 & 4 & 2 \\ 0 & 0 & 0 \end{pmatrix} \\ \mathbf{W}_{\bar{1}} &= \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ -6 & 4 & 2 \end{pmatrix} \end{aligned} \tag{A.4}$$

Higher Orders

IDO5

Moments: u, u_ξ, u_{ξ^2}.

$$\begin{aligned} \mathbf{W}_0 &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 60 & -36 & 9 \end{pmatrix} \equiv \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ 60 & \begin{pmatrix} -36 & 9 \end{pmatrix} \end{pmatrix} \\ \mathbf{W}_{\bar{1}} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -60 & -24 & -3 \end{pmatrix} \equiv \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ -60 & \begin{pmatrix} -24 & -3 \end{pmatrix} \end{pmatrix} \end{aligned} \tag{A.5}$$

IDO7

Moments: u, u_ξ, u_{ξ^2}, u_{ξ^3}.

$$\begin{aligned} \mathbf{W}_0 &= \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -840 & \begin{pmatrix} 480 & -120 & 16 \end{pmatrix} \end{pmatrix} \\ \mathbf{W}_{\bar{1}} &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ 840 & \begin{pmatrix} 360 & 60 & 4 \end{pmatrix} \end{pmatrix} \end{aligned} \tag{A.6}$$

IDO9

Moments: u, u_{ξ}, u_{ξ^2}, u_{ξ^3}, u_{ξ^4}.

$$\begin{aligned}
\mathbf{W}_0 &= \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -15120 & \begin{pmatrix} -8400 & 2100 & -300 & 25 \end{pmatrix} \end{pmatrix} \\
\mathbf{W}_{\bar{1}} &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ 15120 & \begin{pmatrix} -6720 & -1260 & -120 & -5 \end{pmatrix} \end{pmatrix}
\end{aligned} \tag{A.7}$$

IDO11

Moments: u, u_{ξ}, u_{ξ^2}, u_{ξ^3}, u_{ξ^4}, u_{ξ^5}.

$$\begin{aligned}
\mathbf{W}_0 &= \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -332640 & \begin{pmatrix} 181440 & -45360 & 6720 & -630 & 36 \end{pmatrix} \end{pmatrix} \\
\mathbf{W}_{\bar{1}} &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ 332640 & \begin{pmatrix} 151200 & 30240 & 3360 & 210 & 6 \end{pmatrix} \end{pmatrix}
\end{aligned} \tag{A.8}$$

A.2 Two Spatial Dimensions

Third Order

Type-M IDO

Moments: u, u_{ξ}, u_{η}.

$$\begin{aligned}
\mathbf{W}^x_{00} &= \begin{pmatrix} {}^{I}w_0 & {}^{I}w_{\xi 0} & \\ {}^{I\xi}w_0 & {}^{I\xi}w_{\xi 0} & \\ & & {}^{F}w_0 \end{pmatrix} \quad
\mathbf{W}^y_{00} = \begin{pmatrix} {}^{I}w_0 & & {}^{I}w_{\xi 0} \\ & {}^{F}w_0 & \\ {}^{I\xi}w_0 & & {}^{I\xi}w_{\xi 0} \end{pmatrix} \\
\mathbf{W}^x_{\bar{1}0} &= \begin{pmatrix} {}^{I}w_{\bar{1}} & {}^{I}w_{\xi\bar{1}} & \\ {}^{I\xi}w_{\bar{1}} & {}^{I\xi}w_{\xi\bar{1}} & \\ & & {}^{F}w_{\bar{1}} \end{pmatrix} \quad
\mathbf{W}^y_{0\bar{1}} = \begin{pmatrix} {}^{I}w_{\bar{1}} & & {}^{I}w_{\xi\bar{1}} \\ & {}^{F}w_{\bar{1}} & \\ {}^{I\xi}w_{\bar{1}} & & {}^{I\xi}w_{\xi\bar{1}} \end{pmatrix}
\end{aligned} \tag{A.9}$$

Type-A IDO

Moments: u, u_ξ, u_η.

$$
\begin{aligned}
\mathbf{W}^{x}_{00} &= \begin{pmatrix} & 1 & \\ -6 & 4 & \\ -1 & 1 & 1 \end{pmatrix} &
\mathbf{W}^{y}_{00} &= \begin{pmatrix} & & 1 \\ -1 & 1 & 1 \\ -6 & & 4 \end{pmatrix} \\
\mathbf{W}^{x}_{\bar{1}0} &= \begin{pmatrix} & & \\ 6 & 2 & \\ 1 & & -1 \end{pmatrix} &
\mathbf{W}^{y}_{\bar{1}0} &= \begin{pmatrix} & & \\ 1 & & -1 \\ & & \end{pmatrix} \\
\mathbf{W}^{x}_{0\bar{1}} &= \begin{pmatrix} & & \\ & & \\ 1 & -1 & \end{pmatrix} &
\mathbf{W}^{y}_{0\bar{1}} &= \begin{pmatrix} & & \\ 1 & -1 & \\ 6 & & 2 \end{pmatrix} \\
\mathbf{W}^{x}_{\bar{1}\bar{1}} &= \begin{pmatrix} & & \\ & & \\ -1 & & \end{pmatrix} &
\mathbf{W}^{y}_{\bar{1}\bar{1}} &= \begin{pmatrix} & & \\ -1 & & \\ & & \end{pmatrix}
\end{aligned}
\tag{A.10}
$$

Type-C IDO

Moments: u, u_ξ, u_η, $u_{\xi\eta}$.

$$
\begin{aligned}
\mathbf{W}^{x}_{00} &= \begin{pmatrix} {}^{I}w_{0} & {}^{I}w_{\xi 0} & & \\ {}^{I\xi}w_{0} & {}^{I\xi}w_{\xi 0} & & \\ & & {}^{I}w_{0} & {}^{I}w_{\xi 0} \\ & & {}^{I\xi}w_{0} & {}^{I\xi}w_{\xi 0} \end{pmatrix} &
\mathbf{W}^{y}_{00} &= \begin{pmatrix} {}^{I}w_{0} & & {}^{I}w_{\xi 0} & \\ & {}^{I}w_{0} & & {}^{I}w_{\xi 0} \\ {}^{I\xi}w_{0} & & {}^{I\xi}w_{\xi 0} & \\ & {}^{I\xi}w_{0} & & {}^{I\xi}w_{\xi 0} \end{pmatrix} \\
\mathbf{W}^{x}_{\bar{1}0} &= \begin{pmatrix} {}^{I}w_{\bar{1}} & {}^{I}w_{\xi\bar{1}} & & \\ {}^{I\xi}w_{\bar{1}} & {}^{I\xi}w_{\xi\bar{1}} & & \\ & & {}^{I}w_{\bar{1}} & {}^{I}w_{\xi\bar{1}} \\ & & {}^{I\xi}w_{\bar{1}} & {}^{I\xi}w_{\xi\bar{1}} \end{pmatrix} &
\mathbf{W}^{y}_{0\bar{1}} &= \begin{pmatrix} {}^{I}w_{\bar{1}} & & {}^{I}w_{\xi\bar{1}} & \\ & {}^{I}w_{\bar{1}} & & {}^{I}w_{\xi\bar{1}} \\ {}^{I\xi}w_{\bar{1}} & & {}^{I\xi}w_{\xi\bar{1}} & \\ & {}^{I\xi}w_{\bar{1}} & & {}^{I\xi}w_{\xi\bar{1}} \end{pmatrix}
\end{aligned}
\tag{A.11}
$$

IDO/Volume- and Surface- Integrated Average Method (IDO/VSIAM2)

Moments: $\overline{{}^{V}u}$, $\overline{{}^{S^x}u}$, $\overline{{}^{S^y}u}$, u.

$$\mathbf{W}^{x}_{00} = \begin{pmatrix} {}^{V}w_{V0} & {}^{V}w_{S0} & & \\ {}^{S}w_{V0} & {}^{S}w_{S0} & & \\ & & {}^{V}w_{V0} & {}^{V}w_{S0} \\ & & {}^{S}w_{V0} & {}^{S}w_{S0} \end{pmatrix} \quad \mathbf{W}^{y}_{00} = \begin{pmatrix} {}^{V}w_{V0} & & {}^{V}w_{S0} & \\ & {}^{V}w_{V0} & & {}^{V}w_{S0} \\ {}^{S}w_{V0} & & {}^{S}w_{S0} & \\ & {}^{S}w_{V0} & & {}^{S}w_{S0} \end{pmatrix}$$

$$\mathbf{W}^{x}_{\bar{1}0} = \begin{pmatrix} & {}^{V}w_{S\bar{1}} & & \\ & {}^{S}w_{S\bar{1}} & & \\ & & & {}^{V}w_{S\bar{1}} \\ & & & {}^{S}w_{S\bar{1}} \end{pmatrix} \quad \mathbf{W}^{y}_{0\bar{1}} = \begin{pmatrix} & & {}^{V}w_{S\bar{1}} & \\ & & & {}^{V}w_{S\bar{1}} \\ & & {}^{S}w_{S\bar{1}} & \\ & & & {}^{S}w_{S\bar{1}} \end{pmatrix} \tag{A.12}$$

IDO/VSIAM2 with Second Order Central Time Evolution Converting Formula (SOC TEC)

Moments: $\overline{{}^{V}u}$, $\overline{{}^{S^x}u}$, $\overline{{}^{S^y}u}$, u.

$$\mathbf{W}^{x}_{01} = \begin{pmatrix} & & \\ & & \\ {}^{V}w_{V0}/2 & {}^{V}w_{S0}/2 & \end{pmatrix} \quad \mathbf{W}^{y}_{10} = \begin{pmatrix} & & \\ {}^{V}w_{V0}/2 & & {}^{V}w_{S0}/2 \\ & & \end{pmatrix}$$

$$\mathbf{W}^{x}_{00} = \begin{pmatrix} {}^{V}w_{V0} & {}^{V}w_{S0} & \\ {}^{S}w_{V0} & {}^{S}w_{S0} & \\ {}^{V}w_{V0}/2 & {}^{V}w_{S0}/2 & \end{pmatrix} \quad \mathbf{W}^{y}_{00} = \begin{pmatrix} {}^{V}w_{V0} & & {}^{V}w_{S0} \\ {}^{V}w_{V0}/2 & & {}^{V}w_{S0}/2 \\ {}^{S}w_{V0} & & {}^{S}w_{S0} \end{pmatrix}$$

$$\mathbf{W}^{x}_{\bar{1}1} = \begin{pmatrix} & & \\ & & \\ & {}^{V}w_{S\bar{1}}/2 & \end{pmatrix} \quad \mathbf{W}^{y}_{1\bar{1}} = \begin{pmatrix} & & \\ & & {}^{V}w_{S\bar{1}}/2 \\ & & \end{pmatrix}$$

$$\mathbf{W}^{x}_{\bar{1}0} = \begin{pmatrix} & {}^{V}w_{S\bar{1}} & \\ & {}^{S}w_{S\bar{1}} & \\ & {}^{V}w_{S\bar{1}}/2 & \end{pmatrix} \quad \mathbf{W}^{y}_{0\bar{1}} = \begin{pmatrix} & & {}^{V}w_{S\bar{1}} \\ & & {}^{V}w_{S\bar{1}}/2 \\ & & {}^{S}w_{S\bar{1}} \end{pmatrix} \tag{A.13}$$

IDO/Multi-Moment Finite Volume Method (IDO–FVM2)

Moments: $\overline{{}^{V}u}$, ${}^{A}u$ (vertex value), ${}^{B}u$ (horizontal edge midpoint value), ${}^{C}u$ (vertical edge midpoint value).

$$
\begin{aligned}
\mathbf{W}^{x}_{00} &= \begin{pmatrix} & {}^{1}\!/_{6} & & {}^{2}\!/_{3} \\ & 3 & -4 & \\ & 1 & & \\ -9 & {}^{1}\!/_{4} & 1 & 4 \end{pmatrix} &
\mathbf{W}^{y}_{00} &= \begin{pmatrix} & {}^{1}\!/_{6} & {}^{2}\!/_{3} & \\ & 3 & & -4 \\ -9 & {}^{1}\!/_{4} & 4 & 1 \\ & 1 & & \end{pmatrix} \\
\mathbf{W}^{x}_{\bar{1}0} &= \begin{pmatrix} & -{}^{1}\!/_{6} & & -{}^{2}\!/_{3} \\ & 1 & & \\ & -1 & & \\ & {}^{1}\!/_{4} & & 2 \end{pmatrix} &
\mathbf{W}^{y}_{\bar{1}0} &= \begin{pmatrix} & {}^{1}\!/_{6} & & \\ & & & \\ & {}^{1}\!/_{4} & & 1 \\ & & & \end{pmatrix} \\
\mathbf{W}^{x}_{0\bar{1}} &= \begin{pmatrix} & {}^{1}\!/_{6} & & \\ & & & \\ & & & \\ & {}^{1}\!/_{4} & 1 & \end{pmatrix} &
\mathbf{W}^{y}_{0\bar{1}} &= \begin{pmatrix} & -{}^{1}\!/_{6} & -{}^{2}\!/_{3} & \\ & 1 & & \\ & {}^{1}\!/_{4} & 2 & \\ & -1 & & \end{pmatrix} \\
\mathbf{W}^{x}_{\bar{1}\bar{1}} &= \begin{pmatrix} & -{}^{1}\!/_{6} & & \\ & & & \\ & & & \\ & {}^{1}\!/_{4} & & \end{pmatrix} &
\mathbf{W}^{y}_{\bar{1}\bar{1}} &= \begin{pmatrix} & -{}^{1}\!/_{6} & & \\ & & & \\ & {}^{1}\!/_{4} & & \\ & & & \end{pmatrix}
\end{aligned}
\tag{A.14}
$$

Appendix B

Time-Domain Test Results

B.1 One Spatial Dimension

See §3.3.2 for test parameters. The Courant number was $\sigma = 0.1$ unless stated otherwise.

Table B.1: Linear advection errors for the 1D discontinuous Galerkin method with RK3 integration.

Scheme	m	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
DG2	8	2.5274e−02	—	5.7747e−02	—
	16	4.0163e−03	2.654	1.1277e−02	2.356
	32	1.0175e−03	1.981	2.2472e−03	2.327
	64	2.5533e−04	1.995	4.8325e−04	2.217
	128	6.3894e−05	1.999	1.1068e−04	2.126
DG3	8	9.0112e−04	—	1.8962e−03	—
	16	1.1030e−04	3.030	2.4800e−04	2.935
	32	1.3712e−05	3.008	3.1398e−05	2.982
	64	1.7128e−06	3.001	3.9483e−06	2.991
	128	2.1427e−07	2.999	4.9370e−07	3.000
DG4	8	4.1868e−05	—	1.3026e−04	—
($\sigma = 0.01$)	16	2.5857e−06	4.017	8.6741e−06	3.909
	32	1.5997e−07	4.015	5.4213e−07	4.000
	64	9.9383e−09	4.009	3.4027e−08	3.994
	128	6.1573e−10	4.013	2.1428e−09	3.989
DG5	8	2.0369e−06	—	5.4244e−06	—
($\sigma = 5 \times 10^{-4}$)	16	6.5595e−08	4.957	1.8225e−07	4.895
	32	2.0914e−09	4.971	5.9097e−09	4.947
	64	6.6810e−11	4.968	1.8582e−10	4.991
	128	1.6184e−11	2.045	3.0970e−11	2.585

Table B.2: Linear advection errors for the 1D constrained interpolation profile method.

Scheme	m	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
CIP3	8	2.0548e−02	—	3.1841e−02	—
	16	2.7016e−03	2.927	4.2220e−03	2.915
	32	3.4212e−04	2.981	5.3664e−04	2.976
	64	4.2939e−05	2.994	6.7424e−05	2.993
	128	5.3754e−06	2.998	8.4429e−06	2.997
CIP5	8	1.1524e−04	—	1.8102e−04	—
	16	3.7001e−06	4.961	5.8100e−06	4.961
	32	1.1650e−07	4.989	1.8297e−07	4.989
	64	3.6491e−09	4.997	5.7317e−09	4.996
	128	1.1413e−10	4.999	1.7927e−10	4.999
CIP7	8	3.3496e−07	—	5.3837e−07	—
	16	2.7295e−09	6.939	4.3131e−09	6.964
	32	2.1559e−11	6.984	3.3920e−11	6.990
	64	1.7598e−13	6.937	2.8100e−13	6.915
	128	1.0186e−13	0.789	1.6150e−13	0.799
CIP9	8	5.5693e−10	—	8.8575e−10	—
	16	1.1047e−12	8.978	1.7455e−12	8.987
	32	2.0631e−14	5.743	3.2307e−14	5.756
	64	4.7177e−14	−1.193	7.1193e−14	−1.140
	128	1.0172e−13	−1.109	1.6398e−13	−1.204

Table B.3: Linear advection errors for the 1D multi-moment constrained finite volume method with RK3 integration. Third order scheme only (MCV3).

m	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
8	2.5709e−02	—	3.7734e−02	—
16	3.3399e−03	2.944	5.1837e−03	2.864
32	4.2168e−04	2.986	6.6472e−04	2.963
64	5.3014e−05	2.992	8.3541e−05	2.992
128	6.6426e−06	2.997	1.0462e−05	2.997

B.2 Two Spatial Dimensions

See §5.3 for test parameters.

Table B.4: Linear advection at propagation angle $\theta = \tan^{-1} 0.5 \approx 26.6^\circ$. Mean absolute errors.

Scheme	m_x	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
CIP, Type-M	8	7.3172e−02	—	1.1436e−01	—
	16	2.0446e−02	1.839	3.2092e−02	1.833
	32	5.2909e−03	1.950	8.3102e−03	1.949
	64	1.3375e−03	1.984	2.1009e−03	1.984
	128	7.4319e−03	−2.474	1.1673e−02	−2.474
CIP, Type-A	8	3.1661e−02	—	4.9820e−02	—
	16	4.4024e−03	2.846	6.9145e−03	2.849
	32	5.6693e−04	2.957	8.9045e−04	2.957
	64	7.1517e−05	2.987	1.1234e−04	2.987
	128	8.9683e−06	2.995	1.4087e−05	2.995
CIP, Type-C	8	1.7752e−02	—	2.7346e−02	—
	16	2.3209e−03	2.935	3.6247e−03	2.915
	32	2.9406e−04	2.981	4.6121e−04	2.974
	64	3.6930e−05	2.993	5.7987e−05	2.992
	128	4.6249e−06	2.997	7.2640e−06	2.997
CIP/VSIAM2	8	1.6960e−02	—	2.6820e−02	—
	16	2.2927e−03	2.887	3.6106e−03	2.893
	32	2.9314e−04	2.967	4.6080e−04	2.970
	64	3.6901e−05	2.990	5.7975e−05	2.991
	128	4.6239e−06	2.996	7.2636e−06	2.997
CIP/VSIAM2	8	6.7479e−03	—	1.0450e−02	—
(SOC TEC)	16	6.9728e−04	3.275	1.0968e−03	3.252
	32	8.0512e−05	3.114	1.2639e−04	3.117
	64	9.8142e−06	3.036	1.5412e−05	3.036
	128	1.2183e−06	3.010	1.9136e−06	3.010

Table B.5: Linear advection at propagation angle $\theta = 45°$. Mean absolute errors.

Scheme	m_x	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
CIP, Type-M	8	1.4842e−01	—	2.2728e−01	—
	16	4.1715e−02	1.831	6.5211e−02	1.801
	32	1.0801e−02	1.949	1.6942e−02	1.945
	64	2.7291e−03	1.985	4.2851e−03	1.983
	128	6.8482e−04	1.995	1.0756e−03	1.994
CIP, Type-A	8	6.8269e−02	—	1.1284e−01	—
	16	1.0450e−02	2.708	1.6572e−02	2.767
	32	1.3739e−03	2.927	2.1627e−03	2.938
	64	1.7419e−04	2.980	2.7375e−04	2.982
	128	2.1874e−05	2.993	3.4364e−05	2.994
CIP, Type-C	8	2.2242e−02	—	3.4499e−02	—
	16	2.9541e−03	2.913	4.6146e−03	2.902
	32	3.7603e−04	2.974	5.8972e−04	2.968
	64	4.7317e−05	2.990	7.4295e−05	2.989
	128	5.9313e−06	2.996	9.3158e−06	2.996
CIP/VSIAM2	8	2.1117e−02	—	3.2746e−02	—
	16	2.9163e−03	2.856	4.5554e−03	2.846
	32	3.7482e−04	2.960	5.8782e−04	2.954
	64	4.7279e−05	2.987	7.4235e−05	2.985
	128	5.9301e−06	2.995	9.3139e−06	2.995
CIP/VSIAM2	8	2.2361e−02	—	3.4773e−02	—
(SOC TEC)	16	2.9374e−03	2.928	4.5776e−03	2.925
	32	3.7233e−04	2.980	5.8341e−04	2.972
	64	4.6814e−05	2.992	7.3487e−05	2.989
	128	5.8674e−06	2.996	9.2149e−06	2.995

Table B.6: Linear advection at propagation angle $\theta = 45°$. Dissipation and dispersion errors (Takacs, 1985).

Scheme	m_x	$\sqrt{E_{diss}}$	Rate	$\sqrt{E_{disp}}$	Rate
CIP, Type-M	8	7.6577e−02	—	1.4556e−01	—
	16	1.2410e−02	2.625	4.4651e−02	1.705
	32	1.6834e−03	2.882	1.1877e−02	1.910
	64	2.1591e−04	2.963	3.0234e−03	1.974
	128	2.7232e−05	2.987	7.6014e−04	1.992
CIP, Type-A	8	8.0102e−02	—	4.9961e−04	—
	16	1.1730e−02	2.772	2.0576e−04	1.280
	32	1.5296e−03	2.939	1.7638e−05	3.544
	64	1.9358e−04	2.982	1.2136e−06	3.861
	128	2.4299e−05	2.994	1.0324e−07	3.555
CIP, Type-C	8	2.4478e−02	—	4.1051e−03	—
	16	3.2662e−03	2.906	3.0757e−04	3.738
	32	4.1709e−04	2.969	2.0479e−05	3.909
	64	5.2537e−05	2.989	1.3085e−06	3.968
	128	6.5874e−06	2.996	1.0107e−07	3.695
CIP/VSIAM2	8	2.3235e−02	—	3.9096e−03	—
	16	3.2243e−03	2.849	3.0407e−04	3.685
	32	4.1575e−04	2.955	2.0428e−05	3.896
	64	5.2495e−05	2.985	1.3191e−06	3.953
	128	6.5860e−06	2.995	1.4055e−07	3.230
CIP/VSIAM2	8	1.4995e−02	—	1.9993e−02	—
(SOC TEC)	16	2.8635e−03	2.389	1.5545e−03	3.685
	32	4.0005e−04	2.840	1.0394e−04	3.903
	64	5.1564e−05	2.956	6.6378e−06	3.969
	128	6.5034e−06	2.987	3.4331e−07	4.273

Table B.7: Time-varying advection of a bicosine profile. Mean absolute errors.

Scheme	m_x	$\|\|\varepsilon\|\|_1$	Rate	$\|\|\varepsilon\|\|_\infty$	Rate
CIP, Type-M	8	7.1414e−02	—	1.6054e−01	—
	16	2.4919e−02	1.519	5.3911e−02	1.574
	32	6.8655e−03	1.860	1.4807e−02	1.864
	64	1.8229e−03	1.913	4.0761e−03	1.861
	128	2.4877e+44	−156.579	2.7907e+45	−158.906
CIP, Type-A	8	2.8302e−02	—	9.1285e−02	—
	16	4.6960e−03	2.591	1.4376e−02	2.667
	32	6.2841e−04	2.902	1.9162e−03	2.907
	64	8.0431e−05	2.966	2.4679e−04	2.957
	128	4.1102e+04	−28.929	4.5051e+05	−30.766
CIP, Type-C	8	1.2741e−02	—	4.5361e−02	—
	16	1.6975e−03	2.908	6.1595e−03	2.881
	32	2.1761e−04	2.964	7.8865e−04	2.965
	64	2.7420e−05	2.988	9.9455e−05	2.987
	128	3.4402e−06	2.995	1.2480e−05	2.994
CIP/VSIAM2	8	1.1843e−02	—	3.8890e−02	—
	16	1.6880e−03	2.811	5.9273e−03	2.714
	32	2.1680e−04	2.961	7.8111e−04	2.924
	64	2.7398e−05	2.984	9.9216e−05	2.977
	128	3.4398e−06	2.994	1.2472e−05	2.992
CIP/VSIAM2	8	1.5646e−02	—	5.0347e−02	—
(SOC TEC)	16	1.8510e−03	3.079	6.6396e−03	2.923
	32	2.2295e−04	3.054	8.1253e−04	3.031
	64	2.7632e−05	3.012	1.0045e−04	3.016
	128	3.4486e−06	3.002	1.2519e−05	3.004

Table B.8: Time-varying advection of a bicosine profile. Dissipation and dispersion errors (Takacs, 1985).

Scheme	m_x	$\sqrt{E_{diss}}$	Rate	$\sqrt{E_{disp}}$	Rate
CIP, Type-M	8	3.0156e−02	—	7.8236e−02	—
	16	7.4345e−03	2.020	2.7588e−02	1.504
	32	1.3929e−03	2.416	7.7781e−03	1.827
	64	2.3016e−04	2.597	2.1178e−03	1.877
	128	4.5729e+44	−160.443	1.5961e+22	−82.640
CIP, Type-A	8	2.4485e−02	—	2.6711e−02	—
	16	3.8777e−03	2.659	4.3883e−03	2.606
	32	5.1629e−04	2.909	5.9278e−04	2.888
	64	6.5878e−05	2.970	7.6174e−05	2.960
	128	7.2192e+04	−30.029	2.0089e+02	−21.331
CIP, Type-C	8	1.5348e−02	—	5.0750e−03	—
	16	2.0702e−03	2.890	6.8939e−04	2.880
	32	2.6465e−04	2.968	8.8208e−05	2.966
	64	3.3362e−05	2.988	1.1121e−05	2.988
	128	4.1860e−06	2.995	1.3958e−06	2.994
CIP/VSIAM2	8	1.4782e−02	—	4.8418e−03	—
	16	2.0508e−03	2.850	6.8143e−04	2.829
	32	2.6403e−04	2.957	8.7953e−05	2.954
	64	3.3343e−05	2.985	1.1113e−05	2.985
	128	4.1854e−06	2.994	1.3955e−06	2.993
CIP/VSIAM2	8	1.7443e−02	—	1.0928e−02	—
(SOC TEC)	16	2.2082e−03	2.982	1.0048e−03	3.443
	32	2.7081e−04	3.028	1.0178e−04	3.303
	64	3.3608e−05	3.010	1.1666e−05	3.125
	128	4.1953e−06	3.002	1.4168e−06	3.042

Table B.9: Rotation of a Gaussian cone. Mean absolute errors.

Scheme	m_x	$\|\varepsilon\|_1$	Rate	$\|\varepsilon\|_\infty$	Rate
CIP, Type-M	8	2.2967e+01	—	2.4302e+02	—
	16	1.6076e+06	−16.095	3.7568e+07	−17.238
	32	6.3726e+16	−35.206	2.4781e+18	−35.941
	64	2.4914e+45	−94.981	3.1024e+47	−96.660
	128	4.8340e+35	32.263	2.0537e+38	30.493
CIP, Type-A	8	2.6767e−02	—	9.2003e−01	—
	16	2.1859e−02	0.292	6.9803e−01	0.398
	32	6.5491e−02	−1.583	4.5989e+00	−2.720
	64	6.0780e+05	−23.146	1.0412e+08	−24.432
	128	1.9871e+21	−51.538	7.4735e+23	−52.673
CIP, Type-C	8	2.4065e−02	—	6.0218e−01	—
	16	9.6879e−03	1.313	3.1738e−01	0.924
	32	4.2713e−03	1.182	1.7892e−01	0.827
	64	2.1250e−03	1.007	1.0557e−01	0.761
	128	1.0709e−03	0.989	5.4768e−02	0.947
CIP/VSIAM2	8	1.4300e−02	—	1.0785e−01	—
	16	5.8871e−03	1.280	1.5127e−01	−0.488
	32	1.1735e−03	2.327	5.8959e−02	1.359
	64	1.6974e−04	2.789	1.0502e−02	2.489
	128	2.2024e−05	2.946	1.4211e−03	2.886
CIP/VSIAM2	8	1.6820e−02	—	1.1016e−01	—
(SOC TEC)	16	7.1325e−03	1.238	1.7427e−01	−0.662
	32	1.4397e−03	2.309	6.8738e−02	1.342
	64	2.0059e−04	2.843	1.1534e−02	2.575
	128	2.5640e−05	2.968	1.4730e−03	2.969

Appendix C

Simplex-Locating Algorithm

In a semi-Lagrangian method, it becomes necessary to interpolate a profile at some departure point $\mathbf{r}_d$ in order to update the value at the corresponding arrival (grid) point. The question is, where is the profile? In this appendix, a novel search algorithm is presented. The algorithm applies to Cartesian cut cells that are divided into simplices.

If the grid is Cartesian and uncut, calculating the appropriate profile address $\mathbf{i}_d$ is trivial:

$$\mathbf{i}_d = 1 + \text{floor}\left(\frac{\mathbf{r}_d - \mathbf{r}_0}{\Delta\mathbf{r}}\right) \tag{C.1}$$

where 'floor' is a function that rounds a floating-point number towards minus infinity to return an integer, $\mathbf{r}_0$ is the minimum position in the grid, and $\Delta\mathbf{r}$ is the vector of point spacings in each dimension.

If the grid is unstructured, the problem is no longer trivial; one must navigate the grid according to cell geometry and connectivity. Xiu and Karniadakis (2001) provide a good search algorithm that works as long as the cells are convex. The algorithm begins by starting with the cell immediately upwind of the arrival point. The cell is tested to determine whether or not it contains the departure point. If it does not, the next upwind cell is tested, and so on until a positive match is found.

Each cell is tested as follows. For each cell face Fj, a vector from the face to the departure point is calculated. The start point $\mathbf{r}_{Fj}$ can be anywhere on the face – Xiu and Karniadakis choose an arbitrary vertex, but in this discussion the face centroid shall be used. The dot product $(\mathbf{r}_d - \mathbf{r}_{Fj}) \cdot \mathbf{n}_{Fj}$ is evaluated, where $\mathbf{n}_{Fj}$ is the face normal pointing into the cell. If any of the dot products are negative, the departure point must lie outside the cell. An example is depicted in Fjgure C.1.

If the grid is a Cartesian cut cell grid, and the implementation involves fitting piecewise profiles over a cut cell's constituent simplices, then the cell-testing part of Xiu and Karniadakis's algorithm can be reused. But here we have a more favourable situation, because the simplices are compartmentalised according to the Cartesian grid. The procedure for traversing unstructured cells (which is somewhat ambiguous) is therefore no longer needed. Instead, the following

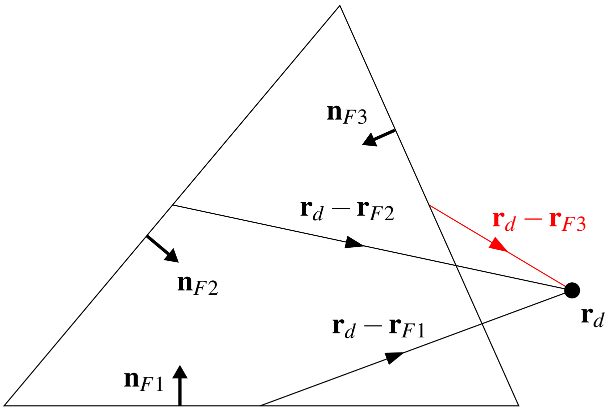

$\mathbf{n}_{F3}$
$\mathbf{r}_d - \mathbf{r}_{F2}$
$\mathbf{r}_d - \mathbf{r}_{F3}$
$\mathbf{n}_{F2}$
$\mathbf{r}_d$
$\mathbf{r}_d - \mathbf{r}_{F1}$
$\mathbf{n}_{F1}$

Figure C.1: Illustration of the cell search algorithm by Xiu and Karniadakis (2001), but with vectors to the departure point calculated from face centroids rather than vertices. In this example, $(\mathbf{r}_d - \mathbf{r}_{F3}) \cdot \mathbf{n}_{F3}$ is negative, indicating that the departure point lies outside the cell.

algorithm is proposed. It is efficient in that only one dot product needs to be performed per face, and Cartesian faces can be omitted from the tests.

A Cartesian cut cell consists of N_C simplicial sub-cells and encloses N_F oblique faces – that is, faces that are not coplanar with any Cartesian faces. In the pentagonal cell shown in Figure C.2, for instance, $N_C = N_F = 3$. The normal vectors of these faces may be stored pointing in any direction. Additionally, two $N_C \times N_F$ logical matrices must be stored, with sub-cells Ci represented in rows and oblique faces Fj in columns. The first matrix $\mathbf{C}$ describes connectivity, with element $(i,\ j)$ set to true if Fj is a face of Ci and false otherwise. The second matrix $\mathbf{D}$ describes direction of face normals relative to respective cells, with element $(i,\ j)$ set to true if the normal of Fj points into Ci, set to false if it is does not, and left with no particular value if Fj is not a face of Ci. Matrices corresponding to Figure C.2 are given in Table C.1.

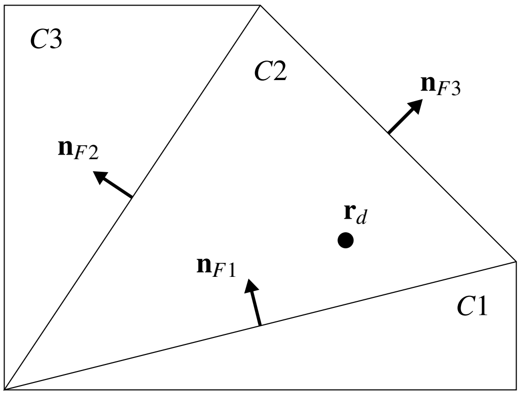


Figure C.2: A cut cell with three constituent sub-cells and three oblique faces.

Table C.1: Logical matrices corresponding to Figure C.2.

(a) Connectivity $\mathbf{C}$

	$F1$	$F2$	$F3$
$C1$	T	F	F
$C2$	T	T	T
$C3$	F	T	F

(b) Direction $\mathbf{D}$

	$F1$	$F2$	$F3$
$C1$	F	T/F	T/F
$C2$	T	F	F
$C3$	T/F	T/F	T

To locate a profile from some point coordinates $\mathbf{r}_d$, one would first identify the host Cartesian cell using Equation C.1. If that cell is uncut, there is only one profile and hence no need to call the simplex-locating algorithm. Otherwise, the algorithm proceeds as follows.

1. Create a result matrix $\mathbf{R}$ of size $N_C \times N_F$. For each oblique face Fj in the Cartesian cell, perform the dot product test and copy the result to all elements in the jth column of $\mathbf{R}$.

$$\mathbf{R}[i,j] = ((\mathbf{r}_d - \mathbf{r}_{Fj}) \cdot \mathbf{n}_{Fj} > 0) \tag{C.2}$$

2. Compute a query matrix $\mathbf{Q}$, with element $(i,\ j)$ equal to true if $\mathbf{r}_d$ falls on the correct side of Fj for each Ci.

$$\mathbf{Q}[i,j] = (\mathbf{D}[i,j] == \mathbf{R}[i,j]) \tag{C.3}$$

where the relational $==$ operator tests for equivalence between elements.

3. Default non-applicable comparisons to acceptable (true) states. In other words, identify the false values in $\mathbf{C}$, and overwrite the corresponding elements in $\mathbf{Q}$ with true values.

$$\mathbf{Q}[i,j] = \text{true} \qquad \text{where } \mathbf{C}[i,j] == \text{false} \tag{C.4}$$

4. Collapse $\mathbf{Q}$ to a column vector $\mathbf{q}$ whose elements flag the all-true rows in $\mathbf{Q}$. There will not be more than one. The position of the true element in $\mathbf{q}$, if it exists, will correspond to the sub-cell containing $\mathbf{r}_d$.

$$\mathbf{q}[i] = \text{all}\,(\mathbf{Q}[i,j]) \tag{C.5}$$

where the 'all' operator returns a value of true if, and only if, all of the elements in the operand are true. In this case the operand is the ith row of $\mathbf{Q}$.

The matrices for the point shown in Figure C.2 are worked through in Table C.2.

To verify the algorithm, the cut cell of Figure C.2 was used again. A collection of random points was generated, and each set of point coordinates was passed into the algorithm to return the index of the corresponding sub-cell. The results can be seen in Figure C.3. They confirm the success of the algorithm.

Table C.2: Logical matrices for the point shown in Figure C.2.

(a) Dot product results $\mathbf{R}$

	$F1$	$F2$	$F3$
$C1$	T	F	F
$C2$	T	F	F
$C3$	T	F	F

(b) Query matrix $\mathbf{Q}$

	$F1$	$F2$	$F3$
$C1$	F	T/F	T/F
$C2$	T	T	T
$C3$	T/F	T/F	F

(c) Defaulted query matrix $\mathbf{Q}$

	$F1$	$F2$	$F3$
$C1$	F	T	T
$C2$	T	T	T
$C3$	T	T	F

(d) Query vector $\mathbf{q}$

	–
$C1$	F
$C2$	T
$C3$	F

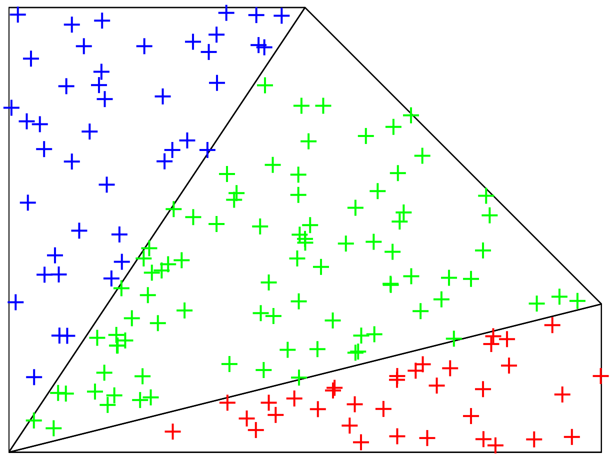


Figure C.3: Verification of the algorithm. Randomly generated points have been coloured (or omitted) according to algorithm output.

Bibliography

Aoki, T. Multi-dimensional advection of CIP (cubic-interpolated propagation) scheme. *Computational Fluid Dynamics Journal*, 4:279–291, 1995.

Aoki, T. Interpolated differential operator (IDO) scheme for solving partial differential equations. *Computer Physics Communications*, 102:132–146, 1997.

Ara, Y., Okubo, K., Tagawa, N., Tsuchiya, T., and Ishizuka, T. Performance comparison of subgrid techniques in acoustic simulation using the type-M and type-C constrained interpolation profile methods. *Japanese Journal of Applied Physics*, 51:07GG05, 2012.

Babuška, I. The theory of small changes in the domain of existence in the theory of partial differential equations and its applications. In *Differential equations and their applications, Proceedings of the Conference held in Prague in September 1962*, pages 13–26. Publishing House of the Czechoslovak Academy of Sciences, 1963.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996. http://www.qhull.org.

Barrett, J. W. and Elliott, C. M. Fitted and unfitted finite-element methods for elliptic equations with smooth interfaces. *IMA journal of numerical analysis*, 7(3):283–300, 1987.

Barth, T. J. and Frederickson, P. O. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. In *28th AIAA Aerospace Sciences Meeting*, volume 1, 1990.

Bassi, F. and Rebay, S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics*, 138(2):251–285, 1997.

Bastian, P. and Engwer, C. An unfitted finite element method using discontinuous Galerkin. *International Journal for Numerical Methods in Engineering*, 79(12):1557–1576, 2009.

Berger, M. J. and Leveque, R. J. An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries. *AIAA paper*, 1930:1–7, 1989.

Björck, Å. *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, 1996.

Brocchini, M. and Peregrine, D. H. The dynamics of strong turbulence at free surfaces. part 1: Description. *Journal of Fluid Mechanics*, 449:225–254, 2001.

Causon, D. M., Ingram, D. M., Mingham, C. G., Yang, G., and Pearson, R. V. Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in water resources*, 23 (5):545–562, 2000.

Chen, C., Xiao, F., Li, X., and Yang, Y. A multi-moment transport model on cubed-sphere grid. *International Journal for Numerical Methods in Fluids*, 67(12):1993–2014, 2011.

Chew, L. P. Guaranteed-quality mesh generation for curved surfaces. In *Proceedings of the ninth annual symposium on Computational geometry*, pages 274–280. ACM, 1993.

Christensen, E. D. Large eddy simulation of spilling and plunging breakers. *Coastal Engineering*, 53:463–485, 2006.

Chung, M.-H. Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape. *Computers & fluids*, 35(6):607–623, 2006.

Cockburn, B. Discontinuous Galerkin methods for convection-dominated problems. *Lecture Notes in Computational Science and Engineering*, 9:69–224, 1999.

Cockburn, B. and Shu, C. TVB runge-kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation*, 52(186):411–435, 1989.

Cockburn, B., Karniadakis, G. E., and Shu, C.-W. *Discontinuous Galerkin Methods: Theory, Computation and Applications*, chapter 'The Development of Discontinuous Galerkin Methods', pages 3–50. Springer, 2000.

Coirier, W. J. *An Adaptively-Refined, Cartesian, Cell-based Scheme for the Euler and Navier-Stokes Equations.* PhD thesis, The University of Michigan, 1994.

Courant, R., Friedrichs, K., and Lewy, H. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928.

Crowley, W. P. Numerical advection experiments 1. *Monthly Weather Review*, 96(1):1–11, 1968.

Dawes, W. N. Turbomachinery computational fluid dynamics: asymptotes and paradigm shifts. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 365 (1859):2553–2585, October 2007. ISSN 1364-503X.

De Zeeuw, D. L. and Powell, K. G. An adaptively refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics*, 104(1):56–68, 1993.

d'Errico, J. Eigenshuffle: Consistently sorted eigenvalue and eigenvector sequences (http://www.mathworks.co.uk/matlabcentral/fileexchange/22885-eigenshuffle). MATLAB Central File Exchange, 2009. Retrieved 28 Sep 2012.

Dolean, V., Fahs, H., Fezoui, L., and Lanteri, S. Locally implicit discontinuous Galerkin method for time domain electromagnetics. *Journal of Computational Physics*, 229(2):512–526, 2010.

Doswell III, C. A. A kinematic analysis of frontogenesis associated with a nondivergent vortex. *Journal of the atmospheric sciences*, 41(7):1242–1248, 1984.

Duan, L., Wang, X., and Zhong, X. A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness. *Journal of computational physics*, 229(19):7207–7237, 2010.

Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60, 2000.

Falcone, M. and Ferretti, R. Convergence analysis for a class of high-order semi-Lagrangian advection schemes. *SIAM Journal on Numerical Analysis*, 35(3):909–940, 1998.

Ferziger, J. H. and Perić, M. *Computational Methods for Fluid Dynamics*. Springer–Verlag, 1996.

Godunov, S. K. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 47(89)(3):271–306, 1959.

Goldstein, D., Handler, R., and Sirovich, L. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2):354–366, 1993.

Golub, G. H. and Van Loan, C. F. *Matrix Computations*. Johns Hopkins University Press, 1983.

Hammer, P. C. and Stroud, A. H. Numerical integration over simplexes. *Mathematical Tables and Other Aids to Computation*, 10(55):137–139, 1956.

Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

Henry, A., Doherty, K., Cameron, L., Whittaker, T., and Doherty, R. Advances in the design of the Oyster wave energy converter. In *RINA Marine and Offshore Renewable Energy*, London, UK, 2010.

Hirsch, C. *Numerical Computation of Internal & External Flows*. Butterworth-Heinemann, Oxford, second edition, 2007.

Hirt, C. W. and Sicilian, J. M. A porosity technique for the definition of obstacles in rectangular cell meshes. In *Proc. Fourth International Conf. Ship Hydro*, 1985.

Hu, C. and Kashiwagi, M. A CIP-based method for numerical simulations of violent free-surface flows. *Journal of Marine Science and Technology*, 9(4):143–157, 2004.

Hu, F. Q., Hussaini, M. Y., and Rasetarinera, P. An analysis of the discontinuous Galerkin method for wave propagation problems. *Journal of Computational Physics*, 151:921–946, 1999.

Huynh, H. T. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. *AIAA paper*, 4079:1–42, 2007.

Ii, S., Shimuta, M., and Xiao, F. A 4th-order and single-cell-based advection scheme on unstructured grids using multi-moments. *Computer Physics Communications*, 173:17–33, 2005.

Ii, S. and Xiao, F. CIP/multi-moment finite volume method for euler equations: A semi-Lagrangian characteristic formulation. *Journal of Computational Physics*, 222(2):849–871, March 2007.

Ii, S. and Xiao, F. A global shallow water model using high order multi-moment constrained finite volume method and icosahedral grid. *Journal of Computational Physics*, 229(5):1774–1796, 2010.

Ii, S. and Xiao, F. X. High order multi-moment constrained finite volume method. part I: Basic formulation. *Journal of Computational Physics*, 228:3669–3707, 2009.

Imai, Y. and Aoki, T. A higher-order implicit IDO scheme and its CFD application to local mesh refinement method. *Computational Mechanics*, 38(3):211–221, 2006a.

Imai, Y. and Aoki, T. Accuracy study of the IDO scheme by Fourier analysis. *Journal of Computational Physics*, 217(2):453–472, 2006b.

Imai, Y., Aoki, T., and Takizawa, K. Conservative form of interpolated differential operator scheme for compressible and incompressible fluid dynamics. *Journal of Computational Physics*, 227:2263–2285, 2008.

Ingram, D. M., Causon, D. M., and Mingham, C. G. Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61(3):561–572, 2003.

Jiang, G.-S. and Shu, C.-W. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1996.

Kim, J., Kim, D., and Choi, H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171(1):132–150, 2001.

Kirkpatrick, M., Armfield, S., and Kent, J. A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid. *Journal of Computational Physics*, 184(1):1–36, 2003.

Konno, M., Okubo, K., Tsuchiya, T., and Tagawa, N. Performances of various types of constrained interpolation profile method for two-dimensional numerical acoustic simulation. *Japanese Journal of Applied Physics*, 47(5):3962–3963, 2008.

Krivodonova, L. and Berger, M. High-order accurate implementation of solid wall boundary conditions in curved geometries. *Journal of Computational Physics*, 211:492–512, 2006.

Lai, M.-C. and Peskin, C. S. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705–719, 2000.

Lauritzen, P. H. A stability analysis of finite-volume advection schemes permitting long time steps. *Monthly Weather Review*, 135:2658–2673, 2007.

Lele, S. K. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16–42, 1992.

Lew, A. J. and Buscaglia, G. C. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering*, 76(4):427–454, 2008.

Li, Y. Wavenumber-extended high-order upwind-biased finite-difference schemes for convective scalar transport. *Journal of Computational Physics*, 133(2):235–255, 1997.

Liang, Q., Zang, J., Borthwick, A. G., and Taylor, P. H. Shallow flow simulation on dynamically adaptive cut cell quadtree grids. *International journal for numerical methods in fluids*, 53 (12):1777–1799, 2007.

Liu, X.-D., Osher, S., and Chan, T. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115:200–212, 1994.

Liu, Y., Vinokur, M., and Wang, Z. J. Spectral difference method for unstructured grids I: Basic formulation. *Journal of Computational Physics*, 216:780–801, 2006.

Losasso, F., Fedkiw, R., and Osher, S. Spatially adaptive techniques for level set methods and incompressible flow. *Computers & Fluids*, 35:995–1010, 2006.

Lowrie, R. B. *Compact higher-order numerical methods for hyperbolic conservation laws*. PhD thesis, University of Michigan, 1996.

Luo, X., Gu, Z., Lei, K., Wang, S., and Kase, K. A three-dimensional Cartesian cut cell method for incompressible viscous flow with irregular domains. *International Journal for Numerical Methods in Fluids*, 69(12):1939–1959, 2012.

McDonald, A. An examination of alternative extrapolations to find the departure point position in a 'two-time-level' semi-Lagrangian integration. *Monthly Weather Review*, 127(9):1985–1993, 1999.

McGregor, J. L. Economical determination of departure points for semi-Lagrangian models. *Monthly Weather Review*, 121(1):221–230, 1993.

Mohd-Yusof, J. Combined immersed boundaries/B-spline methods for simulations of flows in complex geometries. *Center for Turbulence Research Annual Research Briefs*, pages 317–327, 1997.

Mousavi, S., Xiao, H., and Sukumar, N. Generalized gaussian quadrature rules on arbitrary polygons. *International journal for numerical methods in engineering*, 82(1):99–113, 2010.

Nakamura, T., Tanaka, R., Yabe, T., and Takizawa, K. Exactly conservative semi-lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique. *Journal of Computational Physics*, 174:171–207, 2001.

Oliver, T. A. *A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations.* PhD thesis, Massachusetts Institute of Technology, 2008.

Ollivier-Gooch, C. and Van Altena, M. A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation. *Journal of Computational Physics*, 181:729–752, 2002.

Ollivier-Gooch, C., Nejat, A., and Michalak, K. Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations. *AIAA journal*, 47(9):2105–2120, 2009.

Orszaghova, J., Borthwick, A. G. L., and Taylor, P. H. From the paddle to the beach – a Boussinesq shallow water numerical wave tank based on Madsen and SørensenâĂŹs equations. *Journal of Computational Physics*, 231(2):328–344, 2012.

Payne, G. S., Taylor, J. R. M., Bruce, T., and Parkin, P. Assessment of boundary-element method for modelling a free-floating sloped wave energy device. part 1: Numerical modelling. *Ocean Engineering*, 35:333–341, 2008a.

Payne, G. S., Taylor, J. R. M., Bruce, T., and Parkin, P. Assessment of boundary-element method for modelling a free-floating sloped wave energy device. part 2: Experimental validation. *Ocean Engineering*, 35:342–357, 2008b.

Peregrine, D. H. and Thais, L. The effect of entrained air in violent water wave impacts. *Journal of Fluid Mechanics*, 325:377–397, 1996.

Persson, P.-O. and Peraire, J. Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In *47th AIAA Aerospace Sciences Meeting*, 2009.

Peskin, C. S. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

Piperno, S. Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems. *ESIAM: Mathematical Modelling and Numerical Analysis*, 40(5): 815–841, 2006.

Popescu, M., Vedder, R., and Shyy, W. A finite volume-based high-order, Cartesian cut-cell method for wave propagation. *International journal for numerical methods in fluids*, 56(10): 1787–1818, 2008.

Qin, R. and Krivodonova, L. A discontinuous Galerkin method for solutions of the Euler equations on Cartesian grids with embedded geometries. *Journal of Computational Science*, 4:24–35, 2012.

Qiu, J. and Shu, C.-W. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, 193:115–135, 2003.

Qiu, J. and Shu, C.-W. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method, II: Two-dimensional case. *Computers & Fluids*, 34: 642–663, 2005.

Quirk, J. J. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers & Fluids*, 23(1):125–142, 1994.

Richardson, L. F. *Weather Prediction by Numerical Process*. Cambridge University Press, 1922.

Robert, A. A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere–Ocean*, 19(1):37–41, 1981.

Sakurai, K. and Aoki, T. A numerical procedure to evolving contact discontinuity by using moving cut-cell method. *Computational Fluid Dynamics Journal*, 10(1):85–95, 2001.

Saruwatari, A., Watanabe, Y., and Ingram, D. M. Scarifying and fingering surfaces of plunging jets. *Coastal Engineering*, 56:1109–1122, 2009.

Sherwin, S. and Peiró, J. Mesh generation in curvilinear domains using high-order elements. *International journal for numerical methods in engineering*, 53(1):207–223, 2002.

Shiach, J. B., Mingham, C. G., Ingram, D. M., and Bruce, T. The applicability of the shallow water equations for modelling violent wave overtopping. *Coastal Engineering*, 51:1–15, 2004.

Shu, C.-W. and Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–471, 1988.

Staniforth, A. and Côté, J. Semi-Lagrangian integration schemes for atmospheric models – a review. *Monthly Weather Review*, 119(9):2206–2223, 1991.

Sun, Y. and Wang, Z. J. Evaluation of discontinuous Galerkin and spectral volume methods for scalar and system conservation laws on unstructured grids. *International Journal for Numerical Methods in Fluids*, 45:819–838, 2004.

Takacs, L. L. A two-step scheme for the advection equation with minimized dissipation and dispersion errors. *Monthly Weather Review*, 113(6):1050–1065, 1985.

Takewaki, H., Nishiguchi, A., and Yabe, T. Cubic interpolated pseudo-particle method (CIP) for solving hyperbolic-type equations. *Journal of Computational Physics*, 61:261–268, 1985.

Takewaki, H. and Yabe, T. The cubic-interpolated pseudo particle (CIP) method: Application to nonlinear and multi-dimensional hyperbolic equations. *Journal of Computational Physics*, 70:355–372, 1987.

Takizawa, K., Yabe, T., and Nakamura, T. Multi-dimensional semi-Lagrangian scheme that guarantees exact conservation. *Computer Physics Communications*, 148:137–159, 2002.

Tanaka, R., Nakamura, T., and Yabe, T. Constructing exactly conservative scheme in a non-conservative form. *Computer Physics Communications*, 126:232–243, 2000.

Thom, A. The flow past circular cylinders at low speeds. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, pages 651–669, 1933.

Tseng, Y.-H. and Ferziger, J. H. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, 2003.

Udaykumar, H., Shyy, W., and Rao, M. ELAFINT: a mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries. *International journal for numerical methods in fluids*, 22(8):691–712, 1996.

Utsumi, T., Kunugi, T., and Aoki, T. Stability and accuracy of the cubic interpolated propagation scheme. *Computer Physics Communications*, 101:9–20, 1997.

Vincent, P. E., Castonguay, P., and Jameson, A. Insights from von Neumann analysis of high-order flux reconstruction schemes. *Journal of Computational Physics*, 230(22):8134–8154, 2011a.

Vincent, P. E., Castonguay, P., and Jameson, A. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47(1):50–72, 2011b.

Wang, Z. J. Spectral (finite) volume method for conservation laws on unstructured grids: Basic formulation. *Journal of Computational Physics*, 178:210–251, 2002.

Wang, Z. J., Liu, Y., May, G., and Jameson, A. Spectral difference method for unstructured grids II: Extension to the euler equations. *Journal for Scientific Computing*, 32(1):45–71, 2007.

Wang, Z. Y. and Liu, Y. Spectral (finite) volume method for conservation laws on unstructured grids: II. extension to two-dimensional scalar equation. *Journal of Computational Physics*, 179:665–697, 2002.

Watanabe, Y. and Saeki, H. Three-dimensional large eddy simulation of breaking waves. *Coastal Engineering*, 41(3 & 4):281–301, 1999.

Watanabe, Y., Saruwatari, A., and Ingram, D. M. Free-surface flows under impacting droplets. *Journal of Computational Physics*, 227(4):2344–2365, February 2008.

Xiao, F., Yabe, T., and Ito, T. Constructing oscillation preventing scheme for advection equation by rational function. *Computer Physics Communications*, 93:1–12, 1996a.

Xiao, F., Yabe, T., Nizam, G., and Ito, T. Constructing a multi-dimensional oscillation preventing scheme for the advection equation by a rational function. *Computer Physics Communications*, 94(2-3):103–118, April 1996b.

Xiao, F. Unified formulation for compressible and incompressible flows by using multi-integrated moments I: one-dimensional inviscid compressible flow. *Journal of Computational Physics*, 195:629–654, 2004.

Xiao, F. Two variants of the MCV3 scheme. arXiv:1207.6844, 2012.

Xiao, F. and Yabe, T. Completely conservative and oscillationless semi-Lagrangian schemes for advection transportation. *Journal of Computational Physics*, 170(2):498–522, July 2001.

Xiao, F., Ikebata, A., and Hasegawa, T. Numerical simulations of free-interface fluids by a multi-integrated moment method. *Computers & Structures*, 83(6-7):409–423, February 2005.

Xiao, F., Akoh, R., and Ii, S. Unified formulation for compressible and incompressible flows by using multi-integrated moments II: Multi-dimensional version for compressible and incompressible flows. *Journal of Computational Physics*, 213:31–56, 2006.

Xiu, D. and Karniadakis, G. E. A semi-Lagrangian high-order method for Navier-Stokes equations. *Journal of Computational Physics*, 172:658–684, 2001.

Yabe, T. and Aoki, T. A universal solver for hyperbolic equations by cubic-polynomial interpolation: I. one-dimensional solver. *Computer Physics Communications*, 66:219–232, 1991.

Yabe, T., Ishikawa, T., and Wang, P. Y. A universal solver for hyperbolic equations by cubic-polynomial interpolation: II. two- and three-dimensional solvers. *Computer Physics Communications*, 66:233–242, 1991.

Yabe, T., Tanaka, R., Nakamura, T., and Xiao, F. An exactly conservative semi-Lagrangian scheme (CIP–CSL) in one dimension. *Monthly Weather Review*, 129:332–344, 2001a.

Yabe, T., Xiao, F., and Utsumi, T. The constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics*, 169(2):556–593, May 2001b.

Yabe, T., Mizoe, H., Takizawa, K., Moriki, H., Im, H.-N., and Ogata, Y. Higher-order schemes with CIP method and adaptive Soroban grid towards mesh-free scheme. *Journal of Computational Physics*, 194:57–77, 2004.

Yang, G., Causon, D., and Ingram, D. Calculation of compressible flows about complex moving geometries using a three-dimensional Cartesian cut cell method. *International Journal for Numerical Methods in Fluids*, 33(8):1121–1151, 2000.

Yang, J. and Balaras, E. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*, 215 (1):12–40, 2006.

Yang, J. and Stern, F. Sharp interface immersed-boundary/level-set method for wave–body interactions. *Journal of Computational Physics*, 228(17):6590–6616, September 2009.

Ye, T., Mittal, R., Udaykumar, H. S., and Shyy, W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156:209–240, 1999.

Zhang, M. and Shu, C. An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods. *Computers & fluids*, 34(4):581–592, 2005.

Zienkiewicz, O. and Taylor, R. *The Finite Element Method*, volume 1: The Basis, chapter 8, pages 164–199. Butterworth-Heinemann, Oxford, 2000.