Mississippi State University

Scholars Junction

Theses and Dissertations

Theses and Dissertations

1-1-2012

An Optimization-Based Method for High Order Gradient Calculation on Unstructured Meshes

Alcides Dallanora Busatto

Follow this and additional works at: https://scholarsjunction.msstate.edu/td

Recommended Citation

Busatto, Alcides Dallanora, "An Optimization-Based Method for High Order Gradient Calculation on Unstructured Meshes" (2012). *Theses and Dissertations*. 616.
https://scholarsjunction.msstate.edu/td/616

This Dissertation - Open Access is brought to you for free and open access by the Theses and Dissertations at Scholars Junction. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholars Junction. For more information, please contact scholcomm@msstate.libanswers.com.

An optimization-based method for high order gradient calculation on unstructured meshes

By

Alcides Dallanora Busatto

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computational Engineering
in the Bagley College of Engineering

Mississippi State, Mississippi

August 2012

Copyright by

Alcides Dallanora Busatto

2012

An optimization-based method for high order gradient calculation on unstructured meshes

By

Alcides Dallanora Busatto

Approved:

D. Keith Walters
Associate Professor of
Mechanical Engineering
(Major Professor)

J. Mark Janus
Associate Professor of
Aerospace Engineering
(Committee Member)

Seongjai Kim
Associate Professor of
Mathematics
(Committee Member)

Edward A. Luke
Associate Professor of
Computer Science and Engineering
(Committee Member)

Roger L. King
Professor of Electrical and
Computer Engineering
(Graduate Coordinator)

Sarah A. Rajala
Dean of the James Worth Bagley
College of Engineering

Name: Alcides Dallanora Busatto

Date of Degree: August 11, 2012

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr. D. Keith Walters

Title of Study: An optimization-based method for high order gradient calculation on unstructured meshes

Pages of Study: 134

Candidate for Degree of Doctor of Philosophy

A new implicit and compact optimization-based method is presented for high order derivative calculation for finite-volume numerical method on unstructured meshes. High-order approaches to gradient calculation are often based on variants of the Least-Squares (L-S) method, an explicit method that requires a stencil large enough to accommodate the necessary variable information to calculate the derivatives. The new scheme proposed here is applicable for an arbitrary order of accuracy (demonstrated here up to 3rd order), and uses just the first level of face neighbors to compute all derivatives, thus reducing stencil size and avoiding stiffness in the calculation matrix.

Preliminary results for a static variable field example and solution of a simple scalar transport (advection) equation show that the proposed method is able to deliver numerical accuracy equivalent to (or better than) the nominal order of accuracy for both 2nd and 3rd order schemes in the presence of a smoothly distributed variable field (i.e., in the absence of discontinuities).

This new Optimization-based Gradient REconstruction (herein denoted OGRE) scheme produces, for the simple scalar transport test case, lower error and demands less computational time (for a given level of required precision) for a 3rd order scheme when compared to an equivalent L-S approach on a two-dimensional framework. For three-dimensional simulations, where the L-S scheme fails to obtain convergence without the help of limiters, the new scheme obtains stable convergence and also produces lower error solution when compared to a third order MUSCL scheme.

Furthermore, spectral analysis of results from the advection equation shows that the new scheme is better able to accurately resolve high wave number modes, which demonstrates its potential to better solve problems presenting a wide spectrum of wavelengths, for example unsteady turbulent flow simulations.

Key words: high order schemes, gradient reconstruction, finite volume method

DEDICATION

To my loved family.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (NSF) under grant CBET 0645138. The findings and opinions in this dissertation belong solely to the author, and are not necessarily those of the sponsor.

I would like to express my deep appreciation to my major professor, Dr. Keith Walters, who always provided me with wonderful support. His guidance, professionalism and care earned my utmost respect. I thank my committee for their contributions on this dissertation, and in special, Dr. Janus, who I would like to convey my gratitude for his patience every time I needed his help as my former Graduate Coordinator. In addition, I would like to thank the Center for Advanced Vehicular Systems for providing financial support and facilities to this research effort.

I feel compelled to thank the legion of new friends I made in Starkville. Their friendship helped me to mitigate this time far from my family. Thank y'all.

Finally, I would like to thank my whole family for their invaluable love and support. Anytime I needed them, they were there for me. Without them, this adventure would not be possible. This accomplishment belongs to all of you as well!

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

OGRE Optimization-based Gradient REconstruction scheme

L-S Least-Squares scheme

Ansys FLUENT Commercial solver

CFD Computational fluid dynamics

G-G Green-Gauss method

E numerical error

h characteristic mesh length scale

k order of accuracy of the discretization scheme

$NDOF$ number of degrees of freedom

d physical dimension

$Cost$ numerical cost

ρ fluid density

$\vec{U}$ fluid velocity vector

ϕ arbitrary scalar variable

Ω volume integration over the cell

$\partial\Omega$ area integration over the bounding surface of the cell

$\hat{n}$ outward-pointing unit normal vector on the cell surface

N_f number of faces bounding the cell

$\partial\Omega_f$ integration over the face area

F_f convective face flux

$\dot{m}_f$ mass flow rate across the given face

$\bar{\phi}_f$ mass-averaged value of the transport scalar ϕ on face f

$H.O.T.$ high order terms

o center cell centroid

j face index

nj index for neighbor cell centroid

$\phi_a^*\big|_b$ projection of the value of ϕ (or its gradients) from a to b, through Taylor series expansion (these expansions must follow the given order of accuracy of the scheme)

$\vec{r}_{jm}$ connecting vector between quadrature point m at face j and center cell centroid

σ tuning parameter used by OGRE scheme

F quadratic functional

BC boundary condition

λ Lagrange multiplier

x, y, z space coordinates

$\hat{n}$ outward-pointing unit normal vector on the cell surface

b_n face boundary quadrature point

q_b boundary constraint

L Lagrangian function

L_p L norm where p is the norm index

CHAPTER 1

INTRODUCTION

The use of unstructured meshes for complex flow field simulations is often preferred over structured meshes because of their local mesh adaptation and boundary fitting capability. Some scientific applications require the use of high precision numerical methods, but most robust, well tested/validated high precision methods to date have been developed using structured meshed-based solvers. Such a combination of factors makes the development of robust high order accurate methods for unstructured meshed-based solvers a technological necessity.

In a typical flow field simulation, the distribution of an unknown dependent variable must be computed based on a known discrete flow field. For reconstruction-based finite-volume methods of order greater than one, a numerical derivative calculation is usually a necessary step in this process. If a high order scheme (> 2) is desired, a high order gradient calculation must be performed.

Several high order schemes have been developed, and Section 2.1.1 gives a general overview. To date, the most common scheme used to perform high order gradient calculation on structured/unstructured meshes is the Least-Squares (L-S) scheme (or its variants). One key attribute of the Least-Squares scheme is that it requires the expansion of the stencil support to account for the extra information required for high order calculation, which causes some known numerical problems, i.e., stability problems due to the increase in stiffness in the matrix of calculation, and increase of computational memory expense. In addition, expansion of the stencil leads to a large effective mesh size, which adversely impacts the error magnitude as the order of the scheme is increased.

The main goal of the work proposed in this dissertation is to develop a new scheme to address the difficulties mentioned above. This new scheme is an optimization-based high order method that is implicit and compact, and that can be used for unstructured meshed-based solvers. The primary conceptual idea behind this new Optimization-based Gradient REconstruction (herein denoted OGRE) scheme is to arrive at a specific objective function, through geometrical and numerical reasoning, that can be minimized to determine the values of the derivatives. One key aspect of this method is that it relies on stencils comprised only of immediate neighbors, but includes higher-order information via an iterative solution of a system of equations. This approach can be summarized as the minimization of a global objective function that defines the degree to which the individual cell reconstructions (of a variable and its derivatives) match the reconstructions of their immediate neighbors.

The new scheme is implemented as an User Defined Function (UDF) in the commercial solver Ansys FLUENT (v.12.0) for both 2nd and 3rd orders of accuracy, and it is evaluated for some specific two and three-dimensional test cases.

The new methodology, the boundary condition implementation, and the results for simple 2D and 3D test cases are shown in Chapters 2 and 3, respectively (which were already submitted as articles [4] and [5] to peer-reviewed journals). Chapter 4 draws conclusions and describes future work.

CHAPTER 2

A 2D OPTIMIZATION-BASED METHOD FOR HIGH ORDER GRADIENT CALCULATION ON UNSTRUCTURED MESHES

2.1 Introduction

2.1.1 Background

Calculation of local derivatives in discrete variable fields is an essential step in many different numerical methods, which are used in a wide range of application fields including computer graphics, electro-magnetics, solid mechanics, and fluid mechanics. Computational fluid dynamics (CFD), for example, often employs either a finite-difference or finite-volume methodology, either of which requires computation of local derivatives, up to order n, to obtain a solution order of accuracy equal to $n+1$. Furthermore, for a numerical method with order of accuracy $n+1$, each derivative calculation of order $p \leq n$ must itself be accurate to order $n+1-p$. To be useful in practical applications, methods for numerical gradient calculation should be accurate, stable, and computationally efficient. This paper presents a new method for high-order gradient calculation on unstructured meshes. Because of the authors' background, the study is motivated by a goal to improve finite-volume CFD simulations. However, the method is a general approach that can be used for any of the methods and applications mentioned above.

High-order-accurate schemes for gradient calculation on structured meshes have been systematically studied for decades, using finite-difference (FD), finite-element (FE) and finite-volume (FV) methods. An interested reader can refer to the comprehensive review article by Ekaterinaris [10] for a thorough discussion of high-order schemes on structured grids, with a focus on CFD applications.

For FV schemes on unstructured meshes, high-order derivative calculation has been studied less extensively, and most applications make use of 2nd-order schemes, which require calculation of only the first derivative in continuous regions of the variable field. Two of the most common approaches for derivative calculation in such 2nd-order schemes are the Green-Gauss (G-G) method and the Least-Squares (L-S) method [7]. For high-order computations, Barth and Jespersen [3] introduced a multidimensional gradient reconstruction procedure for inviscid higher-order monotonicity preserving (enforcing slope limiter) simulations, employing both cell-center and cell-vertex formulations on unstructured grids. A majority of the subsequent studies on higher order accurate unstructured mesh computations reported in the literature can be seen as a continuation of this initial multidimensional approach. The high-order k-exact method was developed by Barth and Frederickson [2]. They derived general conditions for a scheme to be higher-order accurate by developing a reconstruction procedure satisfying the following three criteria: conservation of mean, k-exactness and compact support.

Methods traditionally known in the structured-grid CFD community as Essentially Non-Oscillatory (ENO) schemes were also extended to unstructured meshes [1, 9, 22, 27], initially based on stencil searching. ENO schemes were developed to ensure uniform high-order accuracy for regions of the domain in the neighborhood of smooth variations of the dependent variable. Later, Weighted Essentially Non-Oscillatory (WENO) schemes were also extended to unstructured meshes [11, 13]. WENO schemes help in the convergence of the solution to a numerical steady-state (a known numerical problem for ENO schemes), and improve accuracy versus the ENO scheme for computation on an equivalent stencil. Ollivier-Gooch [22] proposed a variation on the WENO method by introducing an extra weight (of the order of the truncation error of the approximation) to the data that presents a discontinuous variation on the stencil of the least-squares implementation. An interested reader can refer to [26] for a review of ENO and WENO methods.

High-order methods have also been applied using Discontinuous Galerkin (DG) [6] and Spectral Volume (SV) [12] methods. Unlike FV (but similar to FE), in DG and SV methods each simplex element or cell has multiple degrees of freedom (DOF), depending on the order of accuracy. High-order spatial derivatives are required to reconstruct the dependent variable within each simplex element.

It is known that using explicit methods for high-order calculations presents a numerical problem in which the allowable CFL number must be much smaller than for low-order ones, especially for cases with viscous boundary layers, where the computational mesh must be highly clustered near solid boundaries. Many implicit algorithms have been developed and, in the past decades, adapted for unstructured mesh based solvers, e.g., Jacobi, Gauss-Seidel (GS), preconditioned GMRES, matrix-free Krylov, lower-upper symmetric Gauss-Seidel (LUSGS), line-implicit algorithms, etc. Many of them have been successfully applied to run in high-order simulations [30]. In almost all implicit approaches, the resulting non-linear system of equations is linearized and then solved with an iterative solution algorithm.

Ollivier-Gooch [23] presented results showing the importance of using high-order representation for the domain boundaries as well as the interior regions of the domain. Since most grid generation packages were developed for low-order schemes, it was recommended that high order capabilities should be added to these tools to generate at least quadratic (or higher) reconstruction at boundaries.

While much progress has been made to date, there remains significant room for improvement with regard to robust and efficient high-order methods on unstructured meshed-based solvers. For CFD applications, for example, many of the approaches mentioned above are based on the L-S approach for local gradient calculation, but they tend to become numerically stiff as the order of accuracy increases. Because unstructured meshes are often desired for their improved ability to simulate flow fields using complex geometries and for permitting flexible mesh adaptation near regions of interest, improved methods for high-order gradient calculation can lead to significant improvements in these schemes. Nonetheless, even after decades of advances, unstructured-mesh based solvers still present limitations both in terms of accuracy and efficiency, especially for simulations of complex flow fields.

2.1.2 Error Considerations

The numerical error in any simulation may be expressed in the form

$$E \propto h^k, \tag{2.1}$$

where E is the numerical error, h is a characteristic mesh length scale and k is the order of accuracy of the discretization scheme. Based on this, two approaches may be used to decrease the error of any numerical simulation: refining (globally or locally) the mesh (h-refinement) or increasing the order of the discretization scheme. The number of solution unknowns in a simulation is normally referred to as the number of degrees of freedom ($NDOF$). In the FV method (using a cell centered solver) each mesh cell has one DOF and the total $NDOF$ is related to the mesh size according to

$$NDOF \propto \frac{1}{h^d}, \tag{2.2}$$

where d is the physical dimension (1, 2 or 3). So, for example, in a 2D problem, it is possible to relate the numerical error and $NDOF$ according to

$$E \propto \left(\frac{1}{NDOF}\right)^{\frac{k}{2}}. \tag{2.3}$$

Taking the logarithm of this equation results in

$$\log(E) \propto -k\frac{\log(NDOF)}{2}. \tag{2.4}$$

This relation shows that, in 2D, by applying successive h-refinement of the mesh, and then plotting the graph $\log(E)\ vs\ [\log(NDOF)]/2$, one should get an algebraic convergence for the numerical solution where the plot slope will ideally approach the nominal order of accuracy of the scheme. It is clear that for a very fine mesh, higher-order schemes will produce less numerical error than their lower-order counterparts. If that is the case, why are they not more widely used compared to lower-order schemes? One of the main reasons is the associated costs, both human and numerical [30]. In the example of CFD, most second-order codes have already been heavily tested and validated and can provide acceptable results for most complex flow field problems the CFD community currently faces, so the human cost to switch to a higher-order code may not pay off if the need for high accuracy is not critically important. Furthermore, a higher-order scheme normally increases the stencil (compact support) required to compute the derivatives, making the higher-order solutions more costly regardless of the implementation costs.

One simple analysis to relate numerical error with numerical cost may be obtained using the following reasoning. Let us assume that the cost to converge a flow simulation to steady state is related to the $NDOF$ by

$$Cost \propto NDOF^{c(k)} \tag{2.5}$$

where $c\,(k) \geq 1$. Then

$$\log(E) \propto -\frac{k}{2c\,(k)} \log(Cost). \tag{2.6}$$

It is possible from this equation to deduce that for a low accuracy requirement, a lower-order scheme may provide an acceptable solution with smaller cost than a higher-order scheme. As the required accuracy increases, however, a high-order scheme will deliver a solution with smaller associated cost.

Similarly, observing the analogous relationship given by Eq. (2.4), in theory a lower-order scheme will result in a more accurate solution than a higher order one on a very coarse grid and vice versa as the grid is refined. In practice, however, low- and high-order schemes often yield similar results on very coarse meshes. This is due to the fact that numerical solutions only approach the nominal order of accuracy in the limit of mesh refinement, and high-order schemes tend toward low-order analogs in the coarse mesh limit.

Most of currently popular commercial CFD codes use second-order schemes. They are capable of producing design-quality Reynolds Averaged Navier-Stokes (RANS) results using a mesh with several million cells on commercial clusters after a few hours of simulation time. Second-order solvers present some known numerical problems. First, their solution tends to smear high gradients in convection dominated regions of the flow and do not capture well the total pressure in isentropic regions of the flow field. Second, this smearing feature acts like a numerical diffusivity, which is in effect equivalent to adding extra viscosity in the governing equations. Furthermore, in flows dominated by propagating vortices or acoustic waves, such as resolved turbulence or aero-acoustics simulations, second-order solvers are usually too dissipative to adequately resolve these effects [29]. High-resolution schemes have the potential to correct or mitigate these undesirable numerical features. However, high resolution schemes need to add more information (from more cells) than lower resolution ones to calculate a more accurate approximation of the variable gradient. This normally adds complexity and stiffness to the system of equations, often causing convergence problems and/or substantially increasing computational expense.

2.1.3 Paper Outline

This paper presents a proposed contribution to the problem of high-order derivative computation on unstructured meshes. Specifically, a method is developed which in theory allows arbitrary order derivative calculation based on an implicit, compact scheme.

Stencils are constructed only from nearest neighbors, and the gradient field is obtained through an iterative solution process. The method is based on minimization of an objective function that reflects the degree of mismatch between reconstructions in neighboring computational control volumes. The paper is organized as follows. Section 2.2 outlines a simple test application for the high-order scheme, namely finite-volume solution of a scalar advection equation, and briefly describes the least-squares methodology for derivative calculation. Section 2.3 presents the development of the new methodology up to 3rd order accuracy. Section 2.4 describes the method for boundary condition implementation. Section 2.5 presents numerical validation and initial results. Chapter 4 draws conclusions and describes future work.

2.2 Sample Application and Derivative Calculation

2.2.1 Finite-Volume Solution of Scalar Advection Equation

This section seeks to provide context with regard to derivative computation, using the example of least-square (L-S) schemes for finite-volume (FV) CFD methods. Consider steady-state convective transport of a passive scalar variable, ϕ, in a known velocity field $\vec{U}(x)$. The conservation equation is given by:

$$\nabla \cdot \left(\rho \vec{U} \phi\right) = \frac{\partial}{\partial x_j}\left(\rho u_j \phi\right) = 0, \tag{2.7}$$

where ρ is the fluid density, $\vec{U}$ (or u_j) is the fluid velocity vector, and ϕ is an arbitrary scalar variable. In order to apply the FV method, the flow domain is discretized into non-overlapping control volumes, and Eq. (2.7) is integrated over each control volume (cell). Applying the Gauss divergence theorem yields

$$\int_{\Omega} \nabla \cdot \left(\rho \vec{U} \phi\right) \, dV = \int_{\partial\Omega} \rho \phi \vec{U} \cdot \hat{n} \, dA = \int_{\partial\Omega} \rho \phi \left(u_j n_j\right) \, dA = 0, \tag{2.8}$$

where Ω denotes volume integration over the cell, $\partial\Omega$ denotes area integration over the bounding surface of the cell, and $\hat{n}$ is the outward-pointing unit normal vector on the cell surface. For general polygonal cells (in 2D), the bounding surface is comprised of a finite number of discrete faces (defined as line segments in 2D). The integral may therefore be expressed as

$$\int_{\partial\Omega} \rho \phi \left(u_j n_j\right) \, dA = \sum_{f=1}^{N_f} \int_{\partial\Omega_f} \rho_f \phi_f \left(u_j n_j\right)_f \, dA_f = 0. \tag{2.9}$$

The surface integration is performed as the sum of integrals over each face, N_f is the number of faces bounding the cell, and $\partial\Omega_f$ denotes integration over the face area. To simplify notation, we further adopt the conventional definition of the convective face flux, F_f, defined for each face as

$$F_f = \int_{\partial\Omega_f} \rho_f \phi_f \left(u_j n_j\right)_f \, dA_f. \tag{2.10}$$

The goal of numerical discretization is to approximate the integral in Eq. (2.10) based on available cell data, since the discrete values of the dependent variable ϕ (as well as ρ and $\vec{U}$ in general) are stored at the control volume centers.

For the purpose of developing a new discretization scheme based on the proposed gradient calculation scheme, an upwind method can be used to determine the face fluxes based on the neighbor cell variable reconstructions. For a known velocity field, the convective flux can be expressed as

$$F_f \approx \dot{m}_f \bar{\phi}_f, \tag{2.11}$$

where $\dot{m}_f$ is the mass flow rate across the given face, given by

$$\dot{m}_f = \int_{\partial\Omega_f} \rho_f \left(u_j n_j\right)_f \; dA_f, \tag{2.12}$$

and $\bar{\phi}_f$ is the mass-averaged value of the transport scalar ϕ on face f, which can be expressed as

$$\bar{\phi}_f = \frac{1}{\dot{m}_f} \int_{\partial\Omega_f} \rho_f \phi_f \left(u_j n_j\right)_f \; dA_f. \tag{2.13}$$

For a pure upwind convective scheme, the value of ϕ at any point on face f can be reconstructed by projection from the upwind cell centroid as

$$\phi_f = \phi_U + \left(\frac{\partial \phi}{\partial x_i}\right)_U x_{i,f} + H.O.T., \tag{2.14}$$

where $H.O.T.$ here represents the high order terms of the expansion. The integral in Eq. (2.13) may be evaluated using any appropriate quadrature method of sufficiently high order, in combination with Eq. (2.14). For example, Gauss quadrature is used for the test cases in this paper.

It is possible to rewrite (2.11) as

$$F_f \approx \dot{m}_f \bar{\phi}_f = \dot{m}_f \phi_U + \dot{m}_f \left(\bar{\phi}_f - \phi_U\right). \tag{2.15}$$

For an implicit upwind solution method, the first term on the right hand side can be treated implicitly during the linear solve step, and the second term, incorporating the numerical derivatives, can be treated explicitly. As a result, the contribution of all terms of order greater than one is included as a source term during each outer (Newton) iteration. This approach has been found to be stable in all of the test cases considered.

2.2.2 Derivative Calculation Methods

The approach outlined above for solution of the scalar advection equation requires numerical computation of first derivatives for a second-order scheme, and first and second derivatives for a third order scheme, etc. The derivatives are computed at the centroid of each computational cell. The necessary error order of accuracy for a scheme of order k and derivative of order m is equal to $k - m$. The test application described above is one situation in which accurate derivative calculation methods are desired, but that need is not unique to finite volume numerical methods. For example, computer graphics and image processing applications make use of local gradient information [8, 15], as do remote sensing and topography applications [14].

To compute first derivatives to first order accuracy, the discrete form of the so-called Green-Gauss theorem may be used. For higher-order gradient calculations, the most common method is the cell-based Least-Squares (L-S) method. The L-S scheme is based on a Taylor Series expansion about the centroid of the cell in which the gradient is being computed. Denoting the neighbors of this cell by the index n, and given N total neighbors, the expansion to each neighbor is expressed as

$$\phi_n = \phi_c + \left(\frac{\partial \phi}{\partial x_i}\right)_c x_{i,n} + \mathcal{O}(h^2), \tag{2.16}$$

where $x_{i,n}$ represents the component of the connecting vector between the cell centroids, the index c represents the center cell information, and $\mathcal{O}(h^2)$ indicates the truncation error in the expansion. The concept of neighbor is here defined as "face neighbor" instead of "node neighbor", which means that the neighbors of a given cell are assumed to be as the set of all cells that share a face with the cell of interest. In scalar notation, Eq. (2.16) is alternately expressed using the individual components of the derivative vector. For example, in 2-D, one obtains

$$\phi_n = \phi_c + \left(\frac{\partial \phi}{\partial x}\right)_c x_n + \left(\frac{\partial \phi}{\partial y}\right)_c y_n + \mathcal{O}(h^2). \tag{2.17}$$

Applying Eq. (2.17) to each neighbor cell yields N equations for the two unknowns $\left(\frac{\partial \phi}{\partial x}\right)_c$ and $\left(\frac{\partial \phi}{\partial y}\right)_c$. Since this is an over-determined linear system, it can be solved using a least-square approach to determine the derivative components in each cell.

A similar approach can be used to develop higher-order schemes that include derivatives of the dependent variable greater than one. To implement such a scheme, the neighbor stencil can be increased to include second-level cell neighbors (i.e., neighbors of neighbors), third-level, etc., so that there is sufficient neighbor information in the region of the cell of interest to calculate all of the necessary derivatives for a given reconstruction order.

Recent research efforts (publications by Ollivier-Gooch and his co-workers [16–24]) highlight the inherent challenges of this approach. First, as both the order of the scheme and the dimension of the problem are increased, the size of the stencil may become quite large, with the associated costs in terms of memory and calculation speed. Second, the scheme may become stiff, and require great care to ensure that it is possible to converge to a solution. One such requirement is that a fourth-order scheme must use a full Jacobian to construct the coefficient matrix during the linear solve step in order to get a well-behaved scheme, or else the time-stepping scheme must be carefully controlled to preserve stability. For incompressible flow solutions using segregated solution methods, such an approach is undesirable, and it would be better suited to use only first-order Jacobians, analogous to the approach shown in Eq. (2.15), even for simulations that do not incorporate time-stepping.

The L-S method can be classified as one type of optimization (minimum energy) based method, in which the (local) objective function is a measure of the degree of mismatch between the values of ϕ obtained from a reconstruction (polynomial expansion) about the cell centroids, and the values of ϕ in the neighbors making up the stencil support.

The new method for derivative calculation proposed in this paper seeks to address the difficulties mentioned above, and is herein referred to as the Optimization-based Gradient REconstruction (OGRE) method. The key aspect of the approach is that it relies on stencils comprised only of immediate neighbors, but includes higher-order information via an iterative solution of a system of equations. This has the potential to facilitate (speed up) convergence, by substantially reducing the total cost of calculations for the same level of accuracy versus traditional (i.e. least squares) approaches. Conceptually, the approach can be summarized as the minimization of a global objective function that defines the degree to which the individual cell reconstructions (a variable and its derivatives) match the reconstructions of their immediate neighbors. Like the L-S approach, the new method is an optimization based method. In contrast to L-S, the mismatch used to define the objective function is based on the values of ϕ as well as its derivatives, resulting in a smaller required stencil, and potentially increasing the absolute accuracy of the scheme versus the conventional L-S approach.

2.3 OGRE Scheme Methodology

This section presents the derivation of the OGRE scheme for calculating derivatives of arbitrary order on structured or unstructured computational meshes. The development is presented for the 2nd and 3rd order variants of the new scheme; extension to higher order follows the approach shown here, and is relatively straightforward. We focus specifically on a 2D implementation for simplicity of presentation and validation, however extension to 3D is likewise relatively straightforward. The development is presented in a comprehensive fashion in order to clearly indicate the reasoning behind the final equations used in the algorithm.

2.3.1 Second Order Scheme

For a second order finite-volume scheme, the goal of the method is to find the first order spatial derivatives (in x and y) for a given transport variable ϕ at the centroid of each control volume, with an error order of accuracy equal to one. The primary conceptual idea behind the OGRE method is to arrive at a specific objective function, through geometrical and numerical reasoning, that then can be minimized to determine the values of the derivatives.

Figure 2.1 shows an example of a 2D unstructured cell arrangement. The connecting vector between the center cell centroid o and the face centroid $j1$, is given by $\vec{r}_{j1}$, and the connecting vector between face centroid $j1$ and the neighbor cell centroid nj, is given by $\tilde{r}_{j1}$. Those connecting vectors are defined by:

$$\vec{r}_{j1} = \vec{X}_{j1} - \vec{X}_o = \Delta x_{j1}\vec{i}_x + \Delta y_{j1}\vec{i}_y, \tag{2.18}$$

$$\widetilde{r}_{j1} = \vec{X}_{j1} - \vec{X}_{nj} = \widetilde{\Delta x}_{j1}\vec{i}_x + \widetilde{\Delta y}_{j1}\vec{i}_y.$$

The following notation is also used throughout the derivation calculation (for both 2nd and 3rd order schemes):

o - center cell centroid.

j - face index.

nj - index for neighbor cell centroid.

$\phi_a^*\big|_b$ - projection of the value of ϕ (or its gradients) from a to b, through Taylor series expansion (these expansions must follow the given order of accuracy of the scheme).

A second order finite-volume scheme enforces that all polynomial expansions of the dependent variable are truncated at the first order derivatives. In order to calculate the first derivatives needed for the second order scheme, two measures of disagreement between neighboring cell expansions are postulated:

1) Measure of disagreement between the scalar value projections from each cell at the face quadrature point (centroid):

$$(\Delta\phi)_{j1} = \phi_o^*\big|_{j1}^{(new)} - \frac{1}{2}\left(\phi_o^*\big|_{j1} + \phi_{nj}^*\big|_{j1}\right)^{(current)}, \tag{2.19}$$

2) Measure of disagreement between the first derivative projections from each cell at the face quadrature point (centroid):

$$\Delta\left(\nabla\phi\right)_{j1} \cdot \vec{r}_{j1} = \left(\nabla\phi_o^*\big|_{j1}^{(new)} - \frac{1}{2}\left(\nabla\phi_o^*\big|_{j1} + \nabla\phi_{nj}^*\big|_{j1}\right)^{(current)}\right) \cdot \vec{r}_{j1}. \tag{2.20}$$

These two measures can be analyzed separately, starting with the first measure of disagreement.

2.3.1.1 First Measure of Disagreement

Expanding the first term on the RHS of Eq. (2.19) yields

$$\phi_o^*\big|_{j1} = \phi_o + \nabla\phi\big|_o \cdot \vec{r}_{j1} = \phi_o + \left.\frac{\partial\phi}{\partial x}\right|_o \Delta x_{j1} + \left.\frac{\partial\phi}{\partial y}\right|_o \Delta y_{j1}. \tag{2.21}$$

Also, the second term on RHS of Eq. (2.19) can be denoted $\tilde{F}_{j1}$, which represents an average of the (first-order) Taylor Series expansions of the variable from each of the cells on either side of the face to the face integration point:

$$\begin{aligned}\tilde{F}_{j1} = \frac{1}{2}\left(\phi_o^*\big|_{j1} + \phi_{nj}^*\big|_{j1}\right)^{(current)} &= \frac{1}{2}\left(\phi_o + \nabla\phi\big|_o \cdot \vec{r}_{j1} + \phi_{nj} + \nabla\phi\big|_{nj} \cdot \tilde{r}_{j1}\right)^{(curr)} \\ = \frac{1}{2}\left(\phi_o + \left.\frac{\partial\phi}{\partial x}\right|_o \Delta x_{j1} + \left.\frac{\partial\phi}{\partial y}\right|_o \Delta y_{j1} + \phi_{nj} + \left.\frac{\partial\phi}{\partial x}\right|_{nj} \widetilde{\Delta x}_{j1} + \left.\frac{\partial\phi}{\partial y}\right|_{nj} \widetilde{\Delta y}_{j1}\right)^{(curr)} &.\end{aligned} \tag{2.22}$$

Redefining the first measure of disagreement as f_{j1} and substituting Eq. (2.22), (2.19) results in

$$f_{j1} = \phi_o + \left.\frac{\partial\phi}{\partial x}\right|_o \Delta x_{j1} + \left.\frac{\partial\phi}{\partial y}\right|_o \Delta y_{j1} - \tilde{F}_{j1}. \tag{2.23}$$

2.3.1.2 Second Measure of Disagreement

Based on a zero-order Taylor Series expansion for the gradients:

$$\nabla\phi_o^*\big|_{j1} = \nabla\phi_o \tag{2.24}$$

and

$$\nabla\phi_{nj}^*\big|_{j1} = \nabla\phi_{nj}, \tag{2.25}$$

since the 2nd order (and higher) derivatives are considered to be zero in a second order scheme. This reduces (2.20) to simply

$$\Delta\left(\nabla\phi\right)_{j1}\cdot\vec{r}_{j1}=\left(\nabla\phi_o^{(new)}-\frac{1}{2}\left(\nabla\phi_o+\nabla\phi_{nj}\right)^{(current)}\right)\cdot\vec{r}_{j1}, \tag{2.26}$$

which, after redefining the second measure of disagreement as g_{j1}, (2.20) can be rewritten as

$$g_{j1}=\left.\frac{\partial\phi}{\partial x}\right|_o\Delta x_{j1}+\left.\frac{\partial\phi}{\partial y}\right|_o\Delta y_{j1}-\tilde{G}_{j1}, \tag{2.27}$$

where $\tilde{G}_{j1}$ is given by

$$\begin{aligned}\tilde{G}_{j1}&=\frac{1}{2}\left(\left(\nabla\phi_o+\nabla\phi_{nj}\right)^{(current)}\right)\cdot\vec{r}_{j1}\\&=\frac{1}{2}\left(\left.\frac{\partial\phi}{\partial x}\right|_o\Delta x_{j1}+\left.\frac{\partial\phi}{\partial y}\right|_o\Delta y_{j1}+\left.\frac{\partial\phi}{\partial x}\right|_{nj}\Delta x_{j1}+\left.\frac{\partial\phi}{\partial y}\right|_{nj}\Delta y_{j1}\right)^{(current)}.\end{aligned} \tag{2.28}$$

Equations (2.23) and (2.27) are used to develop an optimization procedure that is described in the following section.

2.3.2 Optimization Procedure for the Second Order Scheme

The goal of the optimization procedure is to minimize, in a least-squares sense, the disagreement between neighbor projections at each face represented by Eqs. (2.23) and (2.27). One consideration is the relative weight to be given to those two measures. In order to control the accuracy of the method, a tuning parameter is incorporated through the use of a weighting coefficient σ. The parameter σ modifies the relative weight given to the two measures of disagreement defined by Eqs. (2.23) and (2.27). It is expected that, in general, decreasing the value of σ should make the scheme more stable, while increasing its value should improve the accuracy. To include this parameter, the two measures f_{j1} and g_{j1}, given by Eqs. (2.23) and (2.27), respectively, can be rewritten as

$$f_{j1}^{*} = \sigma f_{j1} = \sigma \left[\phi_o + \left. \frac{\partial \phi}{\partial x} \right|_o \Delta x_{j1} + \left. \frac{\partial \phi}{\partial y} \right|_o \Delta y_{j1} - \tilde{F}_{j1} \right] \tag{2.29}$$

and

$$g_{j1}^{*} = g_{j1} = \left. \frac{\partial \phi}{\partial x} \right|_o \Delta x_{j1} + \left. \frac{\partial \phi}{\partial y} \right|_o \Delta y_{j1} - \tilde{G}_{j1}, \tag{2.30}$$

where again $\tilde{F}_{j1}$ and $\tilde{G}_{j1}$ are given by Eqs. (2.22) and (2.28), respectively.

By using (2.29) and (2.30), it is possible to construct the following quadratic functional

$$F = \sum_{j=1}^{\#neig} \left({f_{j1}^{*}}^2 + {g_{j1}^{*}}^2 \right), \tag{2.31}$$

which must be minimized in order to find the values of the first order derivatives at any cell centroid. F represents a local objective function for a particular cell. Summing over all cells yields a convex global objective function that can be minimized to find all cell (control volume) derivatives. Minimization of the function F is straightforward, producing the following two equations:

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = 0 \tag{2.32}$$

and

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = 0. \tag{2.33}$$

Those equations can be expressed as

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = 2 \sum_{j=1}^{\#neig} \left\{ f_{j1}^{*} \frac{\partial f_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} + g_{j1}^{*} \frac{\partial g_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} \right\} = 0, \tag{2.34}$$

and

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = 2 \sum_{j=1}^{\#neig} \left\{ f_{j1}^{*} \frac{\partial f_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} + g_{j1}^{*} \frac{\partial g_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} \right\} = 0, \tag{2.35}$$

where the internal partial derivatives are given by:

$$\frac{\partial f_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = \sigma \Delta x_{j1}, \tag{2.36}$$

$$\frac{\partial g_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = \Delta x_{j1}, \tag{2.37}$$

$$\frac{\partial f_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = \sigma \Delta y_{j1}, \tag{2.38}$$

$$\frac{\partial g_{j1}^{*}}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = \Delta y_{j1}. \tag{2.39}$$

Substituting those partial derivatives into (2.34) and (2.35), and dividing all equations by 2, yields

$$\sum_{j=1}^{\#neig} \left\{ f_{j1}^{*} \sigma \Delta x_{j1} + g_{j1}^{*} \Delta x_{j1} \right\} = 0. \tag{2.40}$$

and

$$\sum_{j=1}^{\#neig} \left\{ f_{j1}^{*} \sigma \Delta y_{j1} + g_{j1}^{*} \Delta y_{j1} \right\} = 0. \tag{2.41}$$

Finally, by substituting (2.29) and (2.30) into Eqs. (2.40) and (2.41) and rearranging the terms, one arrives at the system of equations $As = B$, where the matrix of coefficients A is given by

$$A = \begin{bmatrix} \sum \left(\sigma^2 + 1 \right) \Delta x_{j1}^2 & \sum \left(\sigma^2 + 1 \right) \Delta x_{j1} \Delta y_{j1} \\ \sum \left(\sigma^2 + 1 \right) \Delta x_{j1} \Delta y_{j1} & \sum \left(\sigma^2 + 1 \right) \Delta y_{j1}^2 \end{bmatrix}, \tag{2.42}$$

the solution vector s, with the values of the center cell derivatives, is given by

$$s = \left[\begin{array}{cc} \left.\dfrac{\partial \phi}{\partial x}\right|_o & \left.\dfrac{\partial \phi}{\partial y}\right|_o \end{array} \right]^T, \tag{2.43}$$

the RHS vector B is given by

$$B = \left\{ \begin{array}{c} \sum \left(\sigma^2 \Delta x_{j1} \left(\tilde{F}_{j1} - \phi_o \right) + \Delta x_{j1} \tilde{G}_{j1} \right) \\ \sum \left(\sigma^2 \Delta y_{j1} \left(\tilde{F}_{j1} - \phi_o \right) + \Delta y_{j1} \tilde{G}_{j1} \right) \end{array} \right\}, \tag{2.44}$$

and the simplification $\sum = \sum\limits_{j=1}^{\#neig}$ is used. The summation here is performed over each of the immediate face neighbors. The values of $\tilde{F}_{j1}$ and $\tilde{G}_{j1}$ contain the center cell and neighbor's scalar value and derivatives, and so the set of Eqs. given by (2.42), (2.43) and (2.44) for all cells yields a 2N×2N linear system of equations (where N here is the total number of control volumes) that can be solved for $\dfrac{\partial \phi}{\partial x}$ and $\dfrac{\partial \phi}{\partial y}$ in every control volume. The method is therefore implicit and compact.

2.3.3 Simple Comparison between OGRE and L-S 2D Second Order Schemes

Considering that the L-S scheme uses the connecting vectors between center cell centroids:

$$\vec{r}_j = \vec{X}_{nj} - \vec{X}_o = \Delta x_j \vec{i}_x + \Delta y_j \vec{i}_y, \tag{2.45}$$

but OGRE uses the connecting vectors between center cell/neighbor centroids and face centroid:

$$\vec{r}_{j1} = \vec{X}_{j1} - \vec{X}_o = \Delta x_{j1} \vec{i}_x + \Delta y_{j1} \vec{i}_y, \tag{2.46}$$

$$\widetilde{r}_{j1} = \vec{X}_{j1} - \vec{X}_{nj} = \widetilde{\Delta x}_{j1} \vec{i}_x + \widetilde{\Delta y}_{j1} \vec{i}_y,$$

it is possible to make a simple comparison between the two schemes for the special case where the mesh is regular (all cells are equilateral triangles), and also considering $\sigma = 1$. For this case:

$$\vec{r}_{j1} = -\widetilde{r}_{j1} = \vec{X}_{j1} - \vec{X}_o = \Delta x_{j1}\vec{i}_x + \Delta y_{j1}\vec{i}_y = \frac{\Delta x_j}{2}\vec{i}_x + \frac{\Delta y_j}{2}\vec{i}_y, \tag{2.47}$$

which would produce a system of equations $As = B$, where the matrix of coefficients for both cases would be given by:

$$A = \begin{bmatrix} \sum \Delta x_j^2 & \sum \Delta x_j \Delta y_j \\ \sum \Delta x_j \Delta y_j & \sum \Delta y_j^2 \end{bmatrix}, \tag{2.48}$$

and the $R.H.S.$ for the L-S scheme would be given by:

$$B = \begin{Bmatrix} \sum \Delta x_j \left(\phi_j - \phi_o\right) \\ \sum \Delta y_j \left(\phi_j - \phi_o\right) \end{Bmatrix}, \tag{2.49}$$

while the $R.H.S.$ for the OGRE scheme would be given by:

$$B = \begin{Bmatrix} \sum \Delta x_j \frac{1}{2} \left(\phi_j - \phi_o + \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_j + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_j \right) \\ \sum \Delta x_j \frac{1}{2} \left(\phi_j - \phi_o + \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_j + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_j \right) \end{Bmatrix}. \tag{2.50}$$

As it can be seen in Eqs. (2.49) and (2.50), the $R.H.S.$ of the OGRE scheme contains derivative terms, making OGRE an implicit scheme, whereas L-S is an explicit scheme. Furthermore, Eq. (2.50) is also consistent with the Taylor Series expansion:

$$\phi_j - \phi_o = \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_j + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_j, \tag{2.51}$$

which makes OGRE scheme equivalent to L-S scheme for this special case.

2.3.4 Third Order Scheme

A third order finite-volume scheme enforces that all polynomial expansions of the dependent variable are truncated at the second order derivatives. To retain third order accuracy in reconstruction of the face fluxes, two quadrature points (in 2D) must be used to perform the integration over each face. Measures of disagreement are therefore defined at each face quadrature point, denoted for each face by the index m (=1, 2). As above, j indicates face indexing. The relevant projection vectors are therefore defined by:

$$\begin{aligned} \vec{r}_{jm} &= \vec{X}_{jm} - \vec{X}_o = \Delta x_{jm}\vec{i}_x + \Delta y_{jm}\vec{i}_y, \qquad (2.52)\\ \widetilde{r}_{jm} &= \vec{X}_{jm} - \vec{X}_{nj} = \widetilde{\Delta x}_{jm}\vec{i}_x + \widetilde{\Delta y}_{jm}\vec{i}_y. \end{aligned}$$

In order to calculate the first and second derivatives needed for the third order scheme, three measures of disagreement between neighboring cell expansions are postulated for each face integration point:

1) Measure of disagreement between the scalar value projections from each cell at the face quadrature point:

$$(\Delta\phi)_{jm} = \phi_o^*\Big|_{jm}^{(new)} - \frac{1}{2}\left(\phi_o^*\big|_{jm} + \phi_{nj}^*\big|_{jm}\right)^{(current)}, \qquad (2.53)$$

2) Measure of disagreement between the first derivative projections from each cell at the face quadrature point:

$$\Delta\left(\nabla\phi\right)_{jm} \cdot \vec{r}_{jm} = \left(\nabla\phi_o^*\Big|_{jm}^{(new)} - \frac{1}{2}\left(\nabla\phi_o^*\big|_{jm} + \nabla\phi_{nj}^*\big|_{jm}\right)^{(current)}\right) \cdot \vec{r}_{jm}, \qquad (2.54)$$

3) Measure of disagreement between the second derivative projections from each cell at the face quadrature point:

$$\left(\Delta\left(\nabla\nabla\phi\right)_{jm}\cdot\vec{r}_{jm}\right)\cdot\vec{r}_{jm}=\left(\left(\nabla\nabla\phi_o^*\big|_{jm}^{(new)}-\frac{1}{2}\left(\nabla\nabla\phi_o^*\big|_{jm}+\nabla\nabla\phi_{nj}^*\big|_{jm}\right)\right)\right.$$
$$\left.\cdot\,\vec{r}_{jm}\right)\cdot\vec{r}_{jm}, \tag{2.55}$$

where again the second term on the RHS is constructed using current values of the scalar variable and its derivatives.

2.3.4.1 First Measure of Disagreement

Denoting the second term on the RHS of Eq. (2.53) as $\tilde{F}_{jm}$, a third order accurate expansion of both terms using the connecting vectors between the center cell and neighbor centroids to the integration points at their connecting face may be defined, and after redefining the first measure of disagreement as f_{jm}, Eq. (2.53) it can be rewritten as

$$\begin{aligned}f_{jm}=\phi_o+&\left.\frac{\partial\phi}{\partial x}\right|_o\Delta x_{jm}+\left.\frac{\partial\phi}{\partial y}\right|_o\Delta y_{jm}+\\&\frac{1}{2}\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o\Delta x_{jm}^2+2\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o\Delta x_{jm}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial y^2}\right|_o\Delta y_{jm}^2\right)-\tilde{F}_{jm},\end{aligned}\tag{2.56}$$

where $\tilde{F}_{jm}$ is given by

$$\begin{aligned}\tilde{F}_{jm}=\frac{1}{2}\Bigg[&\phi_o+\left.\frac{\partial\phi}{\partial x}\right|_o\Delta x_{jm}+\left.\frac{\partial\phi}{\partial y}\right|_o\Delta y_{jm}+\\&\frac{1}{2}\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o\Delta x_{jm}^2+2\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o\Delta x_{jm}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial y^2}\right|_o\Delta y_{jm}^2\right)+\\&\phi_{nj}+\left.\frac{\partial\phi}{\partial x}\right|_{nj}\widetilde{\Delta x}_{jm}+\left.\frac{\partial\phi}{\partial y}\right|_{nj}\widetilde{\Delta y}_{jm}+\\&\frac{1}{2}\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_{nj}\widetilde{\Delta x}_{jm}^{2}+2\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{nj}\widetilde{\Delta x}_{jm}\widetilde{\Delta y}_{jm}+\left.\frac{\partial^2\phi}{\partial y^2}\right|_{nj}\widetilde{\Delta y}_{jm}^{2}\right)\Bigg]^{(current)}.\end{aligned}\tag{2.57}$$

2.3.4.2 Second Measure of Disagreement

Based on first-order Taylor Series expansions for the gradients and redefining (2.54) as g_{jm}, one obtains

$$g_{jm} = \left\{ \begin{array}{c} \left.\frac{\partial \phi}{\partial x}\right|_{o} + \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{o} \Delta x_{jm} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{o} \Delta y_{jm} \\ \left.\frac{\partial \phi}{\partial y}\right|_{o} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{o} \Delta x_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{o} \Delta y_{jm} \end{array} \right\} \cdot \left\{ \begin{array}{c} \Delta x_{jm} \\ \\ \Delta y_{jm} \end{array} \right\} - \tilde{G}_{jm}, \tag{2.58}$$

where $\tilde{G}_{jm}$ is given by

$$\tilde{G}_{jm} = \frac{1}{2} \left[\left\{ \begin{array}{c} \left.\frac{\partial \phi}{\partial x}\right|_{o} + \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{o} \Delta x_{jm} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{o} \Delta y_{jm} \\ \left.\frac{\partial \phi}{\partial y}\right|_{o} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{o} \Delta x_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{o} \Delta y_{jm} \end{array} \right\} \cdot \left\{ \begin{array}{c} \Delta x_{jm} \\ \\ \Delta y_{jm} \end{array} \right\} + \right. \tag{2.59}$$

$$\left. \left\{ \begin{array}{c} \left.\frac{\partial \phi}{\partial x}\right|_{nj} + \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{nj} \widetilde{\Delta x}_{jm} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{nj} \widetilde{\Delta y}_{jm} \\ \left.\frac{\partial \phi}{\partial y}\right|_{nj} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{nj} \widetilde{\Delta x}_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{nj} \widetilde{\Delta y}_{jm} \end{array} \right\} \cdot \left\{ \begin{array}{c} \Delta x_{jm} \\ \\ \Delta y_{jm} \end{array} \right\} \right]^{(current)} .$$

2.3.4.3 Third Measure of Disagreement

Based on the zero-order expansions for the gradients of the gradients and redefining (2.55) as h_{jm}, one obtains

$$h_{jm} = \left(\left.\frac{\partial^2 \phi}{\partial x^2}\right|_{o} \Delta x_{jm}^2 + 2 \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{o} \Delta x_{jm} \Delta y_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{o} \Delta y_{jm}^2 \right) - \tilde{H}_{jm}, \tag{2.60}$$

where $\tilde{H}_{jm}$, whose expansions must be truncated at the second derivatives, is given by

$$\tilde{H}_{jm} = \frac{1}{2} \left(\left.\frac{\partial^2 \phi}{\partial x^2}\right|_{o} \Delta x_{jm}^2 + 2 \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{o} \Delta x_{jm} \Delta y_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{o} \Delta y_{jm}^2 + \right.$$

$$\left. \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{nj} \Delta x_{jm}^2 + 2 \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_{nj} \Delta x_{jm} \Delta y_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_{nj} \Delta y_{jm}^2 \right)^{(current)} . \tag{2.61}$$

2.3.5 Optimization Procedure for the Third Order Scheme

As above for the second-order version, all three measures of disagreement can be rescaled to control the accuracy and stability of the method through the use of the σ parameter. To do so, the three measures of mismatch: f_{jm}, g_{jm} and h_{jm} , given by (2.56), (2.58) and (2.60), respectively, are rewritten as

$$f^*_{jm} = \sigma^2 f_{jm} = \sigma^2 \left[\phi_o + \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_{jm} + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_{jm} + \frac{1}{2}\left(\left.\frac{\partial^2 \phi}{\partial x^2}\right|_o \Delta x^2_{jm} + 2\left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_o \Delta x_{jm}\Delta y_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_o \Delta y^2_{jm} \right) - \tilde{F}_{jm} \right], \tag{2.62}$$

$$g^*_{jm} = \sigma g_{jm} = \sigma \left[\left(\left.\frac{\partial \phi}{\partial x}\right|_o + \left.\frac{\partial^2 \phi}{\partial x^2}\right|_o \Delta x_{jm} + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_o \Delta y_{jm} \right) \Delta x_j + \left(\left.\frac{\partial \phi}{\partial y}\right|_o + \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_o \Delta x_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_o \Delta y_{jm} \right) \Delta y_j - \tilde{G}_{jm} \right], \tag{2.63}$$

$$h^*_{jm} = h_{jm} = \left(\left.\frac{\partial^2 \phi}{\partial x^2}\right|_o \Delta x^2_{jm} + 2\left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_o \Delta x_{jm}\Delta y_{jm} + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_o \Delta y^2_{jm} \right) - \tilde{H}_{jm}, \tag{2.64}$$

where again $\tilde{F}_{jm}$, $\tilde{G}_{jm}$ and $\tilde{H}_{jm}$ are given by Eqs. (2.57), (2.59) and (2.61), respectively.

By using (2.62), (2.63) and (2.64), it is possible to construct the following quadratic functional

$$F = \sum_{j=1}^{\#neig} \sum_{m=1}^{2} \left({f^*_{jm}}^2 + {g^*_{jm}}^2 + {h^*_{jm}}^2 \right), \tag{2.65}$$

which must be minimized in order to find the values of the first and second order derivatives at any given cell centroid. As above, minimization of the function F is straightforward, producing the following five equations (one for each derivative component sought):

$$\frac{\partial F}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}=0,\frac{\partial F}{\partial\left(\left.\frac{\partial\phi}{\partial y}\right|_o\right)}=0,\frac{\partial F}{\partial\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o\right)}=0,$$
$$\frac{\partial F}{\partial\left(\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o\right)}=0,\frac{\partial F}{\partial\left(\left.\frac{\partial^2\phi}{\partial y^2}\right|_o\right)}=0. \tag{2.66}$$

The first equation in (2.66) can be expressed as

$$\frac{\partial F}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}=2\sum_{j=1}^{\#neig}\sum_{m=1}^{2}\left\{f^*_{jm}\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}+g^*_{jm}\frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}+\right.$$
$$\left.h^*_{jm}\frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}\right\}=0, \tag{2.67}$$

and the other equations are expressed in a similar manner. The internal partial derivatives are given by:

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}=\sigma^2\Delta x_{jm},\frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}=\sigma\Delta x_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial x}\right|_o\right)}=0,$$
$$\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial y}\right|_o\right)}=\sigma^2\Delta y_{jm},\frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial y}\right|_o\right)}=\sigma\Delta y_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial\phi}{\partial y}\right|_o\right)}=0,$$
$$\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o\right)}=\frac{\sigma^2\Delta x^2_{jm}}{2},\frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o\right)}=\sigma\Delta x^2_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o\right)}=\Delta x^2_{jm}, \tag{2.68}$$
$$\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o\right)}=\sigma^2\Delta x_{jm}\Delta y_{jm},\frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o\right)}=2\sigma\Delta x_{jm}\Delta y_{jm},$$
$$\frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o\right)}=2\Delta x_{jm}\Delta y_{jm},\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial y^2}\right|_o\right)}=\frac{\sigma^2\Delta y^2_{jm}}{2},$$
$$\frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial y^2}\right|_o\right)}=\sigma\Delta y^2_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial^2\phi}{\partial y^2}\right|_o\right)}=\Delta y^2_{jm}.$$

Substituting Eqs. (2.68) into the Eqs. (2.66), dividing all equations by 2, and rearranging terms, the system of equations (2.66) can be rewritten in matrix form as $As = B$, where the components of the matrix of coefficients A and the RHS vector B are given in Appendix A.1.

It is worthwhile to reinforce at this point that the cell neighbors over which the summation is performed in the OGRE third order scheme takes into account just the first level of face neighbors (e.g., 3 face neighbors for an internal control volume in a triangle-based 2D unstructured mesh). This is in significant contrast to a L-S third order scheme, which would require the summation over a larger stencil made up of at least 5 neighbor cells (in 2D), typically constructed of both first and second level face neighbors. Higher-order implementations of the OGRE method would similarly require only the immediate face neighbors of each cell to populate the computational stencil.

The global system of equations obtained using the OGRE method can be solved using an iterative procedure. The local system in each cell (e.g. 5x5 for a third-order 2D implementation) can be solved using any appropriate technique, for example simple Gaussian Elimination. The derivative values can be sequentially calculated and updated for each cell, thus producing a Gauss-Seidel-type of iterative procedure. For a fixed variable field, the gradient values will converge toward the values which minimize the objective function. When the OGRE scheme is used within numerical schemes which are themselves iterative in nature, for example implicit finite volume simulations, the gradient field may be updated once per outer (Newton) iteration. Therefore, within each time step (or during a steady-state simulation), the variable field and the gradient field converge simultaneously towards the solution.

2.4 Boundary Condition Implementation

The previous section describes in detail the procedure for computing derivatives in interior cells, i.e. cells for which each bounding face is in the interior of the domain. A modified procedure must be used for boundary cells (i.e. cells for which at least one bounding face lies on the domain boundary). The boundary condition method proposed here first computes values of ϕ and its derivatives at the boundary face centroids. Boundary cells then use a procedure similar to that outlined above, but with matching conditions defined based on projections from the face centroids. The boundary conditions themselves are imposed as constraints in the optimization problem, and a procedure based on Lagrange multipliers is adopted. Two general types of boundary conditions are addressed: given boundary value (Dirichlet) and given normal gradient at the boundary (Neumann). The former yields a constraint on the value of ϕ at the boundary face, and the latter yields a constraint on the value of $\nabla\phi \cdot \hat{n}$, where $\hat{n}$ is the unit normal vector at the boundary face. The constraints are applied at the boundary face quadrature points.

Figure 2.2 illustrates in general how the procedure is implemented. For the 2nd order scheme, the face centroid denoted by b is the only face integration point; for the 3rd order scheme, the two integration points are denoted by b_1 and b_2. The values of ϕ and its derivatives must be defined at the boundary location b (face centroid). They are calculated using information from two layers of neighbor cells, for example the cells 0, 1 and 2 in

Figure 2.2. One exception occurs if the face is located at a corner and its adjacent cell is also facing a different boundary type. In this case, just two cells (the adjacent cell and its first face neighbor) are used to calculate the values at b instead of three. Similar to what is done for the internal cell calculations, measures of disagreement are defined at each of the face integration points, and the gradients and/or variable values determined based on a minimization of those measures.

2.4.1 BC Implementation for the Second Order Scheme

For the 2nd order scheme the values of ϕ and its gradient are expanded (in general) from the three closest cells 0, 1 and 2 to the face centroid b (which is the face integration location for the second order scheme) where two measures of disagreement are formed. The measures of disagreement for ϕ and its gradient in each cell k (= 0, 1, 2) are given by:

$$R_{1k} = \sigma \left(\phi_b^{(new)} - \frac{1}{2} \left(\phi_b + \phi_k^*\big|_b \right)^{(current)} \right), \tag{2.69}$$

and

$$R_{2k} = \left(\nabla\phi_b^{(new)} - \frac{1}{2} \left(\nabla\phi_b + \nabla\phi_k^*\big|_b \right)^{(current)} \right) \cdot \vec{r}_{k1}, \tag{2.70}$$

where

$$\vec{r}_{k1} = \vec{X}_b - \vec{X}_k = \Delta x_{k1} \vec{i}_x + \Delta y_{k1} \vec{i}_y \tag{2.71}$$

(notice here again the inclusion of the tuning parameter σ).

The vector used for the dot product in (2.70), if following the same approach as for the internal cells calculation, should be the connecting vector between the position where the new values are being sought and the face integration points, but since for 2nd order the face centroid and the face integration point are the same, it was decided instead to use the connecting vector between the center cell centroid k and the face integration point. To maintain consistency, the same approach is used for the 3rd order case as well. Another reason for doing so is that, if the vector chosen for the dot product was the connecting vector between the face centroid and the face integration points (for 3rd order scheme), the scheme would not converge. The cause for that probably is in the fact that, since the linear combination of those vectors would produce just one direction in 2D (always parallel to the boundary face) and this would create a geometrical inconsistency in the boundary face calculation.

The disagreement measures in Eq. (2.69) and (2.70) are squared to create a quadratic functional, which has the form

$$F\left(x^{*}\right)=\sum_{k=0}^{2}\left(R_{1k}^{2}+R_{2k}^{2}\right), \tag{2.72}$$

where x^{*} is the solution vector containing the values of ϕ and its two first derivatives at the boundary face centroid b.

The additional constraint for a Dirichlet BC is given by:

$$q_b : \phi_b-\phi_{BC}=0, \tag{2.73}$$

and the additional constraint for a Neumann BC is given by:

$$q_b : \left(\nabla\phi\cdot\hat{n}\right)_b-\left|\nabla\phi\right|_{BC}=0, \tag{2.74}$$

where the vector $\hat{n}$ represents the unit normal vector for the boundary face. The boundary condition values ϕ_{BC} or $|\nabla\phi|_{BC}$ may be constant or vary as a function of position on the boundary. For a zero-flux condition, the Neumann condition is expressed as

$$q_b : \left(\nabla\phi \cdot \hat{n}\right)_b = 0. \tag{2.75}$$

This boundary condition may be applied, for example, at a symmetry plane in finite-volume simulations.

The Lagrangian function is defined in terms of the objective function and the constraint as follows:

$$L\left(x^*, \lambda\right) = F\left(x^*\right) + \lambda q_b. \tag{2.76}$$

To minimize the Lagrangian it is necessary to solve the following system of equations:

$$\frac{\partial L}{\partial x_j^*} = 0 \;:\; j = 1, 2, 3, \tag{2.77}$$

$$\frac{\partial L}{\partial \lambda} = 0. \tag{2.78}$$

Applying Taylor Series expansions of appropriate order for (2.69) and (2.70), they can be substituted into (2.76) so the system of equations defined by (2.77) – (2.78) can be obtained. After rearranging the terms, one arrives at the system of equations $As = C$, where the matrix of coefficients A for the Dirichlet type of BC is given by

$$A = \begin{bmatrix} \sum\left(2\sigma^2\right) & 0 & 0 & 1 \\ 0 & \sum\left(2\Delta x_{k1}^2\right) & \sum\left(2\Delta x_{k1}\Delta y_{k1}\right) & 0 \\ 0 & \sum\left(2\Delta x_{k1}\Delta y_{k1}\right) & \sum\left(2\Delta y_{k1}^2\right) & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{2.79}$$

the expanded solution vector s is now given by

$$s = \left[\begin{array}{cccc} \phi_b & \left.\frac{\partial \phi}{\partial x}\right|_b & \left.\frac{\partial \phi}{\partial y}\right|_b & \lambda \end{array} \right]^T, \tag{2.80}$$

and the RHS vector C is given by

$$C = \left\{ \begin{array}{c} \sum 2\sigma^2 \overline{\phi}_{bk} \\ \sum 2\Delta x_{k1} \overline{\nabla\phi}_{bk} \cdot \vec{r}_{k1} \\ \sum 2\Delta y_{k1} \overline{\nabla\phi}_{bk} \cdot \vec{r}_{k1} \\ \phi_{BC} \end{array} \right\}. \tag{2.81}$$

The averaged values on the RHS are given by

$$\overline{\phi}_{bk} = \frac{1}{2}\left(\phi_b + \phi_k + \left.\frac{\partial \phi}{\partial x}\right|_k \Delta x_{k1} + \left.\frac{\partial \phi}{\partial y}\right|_k \Delta y_{k1} \right) \tag{2.82}$$

and

$$\overline{\nabla\phi}_{bk} \cdot \vec{r}_{k1} = \frac{1}{2}\left(\left.\frac{\partial \phi}{\partial x}\right|_b \Delta x_{k1} + \left.\frac{\partial \phi}{\partial y}\right|_b \Delta y_{k1} + \left.\frac{\partial \phi}{\partial x}\right|_k \Delta x_{k1} + \left.\frac{\partial \phi}{\partial y}\right|_k \Delta y_{k1} \right) \tag{2.83}$$

and the simplification $\sum = \sum_{k=0}^{1\ or\ 2}$ is used.

The following substitution on the fourth row/column of A (since A is symmetric) produces the matrix of coefficients for the Neumann type of BC

$$A(4,:) = A(:,4)^T = \left[\begin{array}{cccc} 0 & n_x & n_y & 0 \end{array} \right], \tag{2.84}$$

where n_x and n_y are the components of the normal unit vector for the boundary face. Also, the fourth component in C must be changed such that $C(4) = |\nabla\phi|_{BC}$.

2.4.2 BC Implementation for the Third Order Scheme

For the 3rd order scheme the values of ϕ, the gradient of ϕ and the gradient of the gradient of ϕ are expanded from the three closest cells k (= 0, 1, 2) and from the boundary face centroid to each one of the two face integration points, thus forming three measures of disagreement. In this way, 9 terms are created for each one of the two face integration points. The three measures of disagreement for a given cell k on the boundary face integration point b_m are given by:

$$R_{1km} = \sigma^2 \left(\phi_b^*\Big|_{b_m}^{(new)} - \frac{1}{2} \left(\phi_b^*\Big|_{b_m} + \phi_k^*\Big|_{b_m} \right)^{(current)} \right), \tag{2.85}$$

$$R_{2km} = \sigma \left(\nabla\phi_b^*\Big|_{b_m}^{(new)} - \frac{1}{2} \left(\nabla\phi_b^*\Big|_{b_m} + \nabla\phi_k^*\Big|_{b_m} \right)^{(current)} \right) \cdot \vec{r}_{km}, \tag{2.86}$$

$$R_{3km} = \left(\left(\nabla\nabla\phi_b^*\Big|_{b_m}^{(new)} - \frac{1}{2} \left(\nabla\nabla\phi_b^*\Big|_{b_m} + \nabla\nabla\phi_k^*\Big|_{b_m} \right)^{(current)} \right) \cdot \vec{r}_{km} \right) \cdot \vec{r}_{km}, \tag{2.87}$$

where the index m here represents boundary face integration point location and

$$\vec{r}_{km} = \vec{X}_{b_m} - \vec{X}_k = \Delta x_{km} \vec{i}_x + \Delta y_{km} \vec{i}_y \tag{2.88}$$

represents the connecting vector between the cell centroid k and the boundary face integration point b_m.

Those 18 terms (when all 3 cells are used) are squared to create a quadratic functional, which has the form

$$F\left(x^*\right) = \sum_{m=1}^{2} \sum_{k=0}^{2} \left(R_{1km}^2 + R_{2km}^2 + R_{3km}^2 \right), \tag{2.89}$$

where x^* is the solution vector containing the values of ϕ and its first and second derivatives at the boundary face centroid.

The additional constraints for a Dirichlet BC are given by:

$$q_1 : \phi_b^*\big|_{b_1} - \phi_{b1} = 0, \tag{2.90}$$

$$q_2 : \phi_b^*\big|_{b_2} - \phi_{b2} = 0, \tag{2.91}$$

and the additional constraints for a Neumann BC are given by:

$$q_1 : (\nabla\phi \cdot \hat{n})_b^*\big|_{b_1} - |\nabla\phi|_{b1} = 0, \tag{2.92}$$

$$q_2 : (\nabla\phi \cdot \hat{n})_b^*\big|_{b_2} - |\nabla\phi|_{b2} = 0. \tag{2.93}$$

Notice that the constraints are always applied at the boundary face integration points. In the previous section, it was applied at the boundary face centroid only because that is the position for the boundary face integration point for a 2nd order scheme.

The Lagrangian function is defined as follows

$$L\left(x^*, \lambda\right) = F\left(x^*\right) + \lambda_1 q_1 + \lambda_2 q_2. \tag{2.94}$$

To minimize the Lagrangian it is necessary to solve the following system of equations:

$$\frac{\partial L}{\partial x_j^*} = 0 \ : \ j = 1, 6, \tag{2.95}$$

$$\frac{\partial L}{\partial \lambda_l} = 0 \ : \ l = 1, 2. \tag{2.96}$$

Using third order expansions on (2.85), (2.86) and (2.87), they can be substituted into (2.94) so the system of equations (2.95) – (2.96) can be calculated. After rearranging the terms, one arrives at the system of equations $As = C$, whose components, the matrix of coefficients A and the RHS vector C, for the Dirichlet and Neumann types of BC, are given in the Appendix A.2.

2.5 Results

The OGRE algorithm was implemented as a User Defined Function (UDF) into the commercial flow solver Ansys Fluent v12.0. Two-dimensional versions of both the 2nd and 3rd order schemes were implemented. Results were obtained for several test cases and compared to 2nd and 3rd order least-squares methods. Test cases were performed with two objectives. The first objective was to check if the OGRE scheme is able to deliver a numerical solution with measured accuracy comparable with the nominal order of the defined scheme for two different cases: a static variable field test case and a finite-volume scalar advection case where a variable flow field is smoothly distributed throughout the domain. Both quantitative and qualitative analyses were performed. The second objective was to determine how effectively the OGRE scheme is able to reconstruct a variable field with a wide spectrum of wavelengths. The result sheds light on the spectral behavior of the scheme, which could indicate its potential ability to deal with problems that require low numerical dissipation and/or energy preservation.

2.5.1 Static Variable Field Test Case

The first test case is straightforward. Given a distribution of a (fixed) arbitrary variable, reconstruct the gradients of the variable at each point in the domain. This test case has relevance to, for example, image processing or data analysis applications. The objective is to show that the OGRE scheme is able to reproduce the gradients to the correct (nominal) order of accuracy. The computational geometry for this case is shown in Figure 2.3. All boundary faces are treated identical to the interior faces by setting them to be of periodic/shadow type. The distribution of scalar ϕ throughout the domain is given by the following analytical function

$$\phi\,(x,\ y) = sin(8\pi(x-0.5))\ cos(8\pi(y-0.5)). \tag{2.97}$$

In order to assess the numerical accuracy of this methodology, it is important to first define how the errors are quantified. The error norms in a numerical calculation are the most common metric for accuracy assessment. For 2D formulations, the general form of the error norm can be defined as

$$L_p = \left(\frac{\sum\limits_{i=1}^{N_{cv}} A_i \left|\bar{E}_i\right|^p}{\sum\limits_{i=1}^{N_{cv}} A_i} \right)^{\frac{1}{p}}, \tag{2.98}$$

where

$$\bar{E}_i = \phi_{i,exact} - \phi_{i,calculated}, \tag{2.99}$$

p is the norm index, N_{cv} is the total number of control volumes in the domain, A_i is the area of the control volume i, and $\bar{E}_i$ is the solution error in the control volume i. While L_1 and L_2 are global norms, L_∞ is a local error indicator since it shows the largest magnitude of the error in the solution domain. It is defined as

$$L_\infty = \max\left|\bar{E}_i\right|. \tag{2.100}$$

To test the numerical order of accuracy of the proposed method, five different unstructured triangular meshes were created using the commercial mesh generation system Ansys Gambit. The coarsest mesh is denoted Mesh2D 1 and contains 506 cells. To create the second mesh (Mesh2D 2), all faces on the boundaries were decreased to half of their initial size, resulting in twice as many boundary faces/nodes. Mesh2D 2 was then generated, and the number of cells increased by a factor of approximately 4 (as expected). The procedure was repeated until five successively finer meshes were created. The numbers of cells/faces/nodes for these five meshes (and Mesh2D 6 which is used in the spectral resolution test case) are given in Table 2.1. For illustration purposes, Mesh2D 1 (the coarsest mesh) is shown in Figure 2.4. It is expected that by refining the mesh, the error in the solution should follow the relation given by Eq. (2.4). Alternatively, between any two different meshes (in 2D) with a number of cells equal to $NDOF_1$ and $NDOF_2$, respectively, the numerical accuracy can be approximated by

$$k = \frac{2\log\left(L_{p1}/L_{p2}\right)}{\log\left(NDOF_2/NDOF_1\right)}, \tag{2.101}$$

where L_{p1} and L_{p2} are the respective error norms (p = 1, 2, ∞) for the solution on each mesh.

For the test case considered here, the 2nd order OGRE scheme with $\sigma = 1.1$ is used to obtain the values for $\frac{\partial \phi}{\partial y}$, and the 3rd order OGRE scheme with $\sigma = 1.4$ is used to obtain the values of $\frac{\partial \phi}{\partial y}$ and $\frac{\partial^2 \phi}{\partial x^2}$. The results for the L_2 error norms of $\frac{\partial \phi}{\partial y}$ and $\frac{\partial^2 \phi}{\partial x^2}$ in comparison with their L-S counterparts are shown in Figures 2.5 and 2.6, respectively. The numerical error accuracy orders for the different schemes, obtained from the results for the two finest meshes, are indicated in the plots. Most importantly, all error orders of accuracy compare well with the nominal accuracy of the given schemes. The results for $\frac{\partial \phi}{\partial y}$ using the 2nd order OGRE scheme compare almost exactly to the L-S 2nd order results. For the

3rd order OGRE scheme the error in the first derivative is smaller when compared to the L-S results, however the results for $\frac{\partial^2 \phi}{\partial x^2}$ show a higher numerical error when compared to L-S. This is due to the fact that the L-S scheme indicates order of accuracy in the second derivatives significantly greater than the nominal accuracy of one. The other first- and second-derivative components showed similar behavior to the ones shown here.

A qualitative examination of the results on the coarsest mesh (Mesh2D 1) when using OGRE and L-S methods for the third-order scheme is provided in Figures 2.7, 2.8 and 2.9, showing the distribution of $\frac{\partial \phi}{\partial y}$ for the OGRE 3rd order scheme, for exact (analytical) values, and for the L-S 3rd order scheme, respectively. Both the OGRE and L-S schemes yield a close approximation of the analytical result, with the OGRE scheme showing slightly better agreement, at least in the first derivative. To evaluate the differences in performance between the two schemes, it is necessary to investigate results from the sample application (finite-volume solution of advection equation), and these results are presented in the following subsections.

Because the OGRE scheme is implicit, the gradient field must be found through an iterative procedure. It was found that about 10 iterations were sufficient to obtain a gradient solution which presents no more than 10% of difference of the fully converged one for a static variable field. The L-S method, however, is an explicit scheme, and only one iteration is required to obtain the gradients in each cell. For the 3rd order scheme, this represents a tradeoff in terms of computational expense, since the stencil support is reduced in OGRE but multiple iterations are required for gradient convergence.

2.5.2 Simple Scalar Transport Test Case

The first finite-volume test case examines simple advection of a smoothly distributed profile of an arbitrary scalar transport variable ϕ. The geometry and boundary conditions for this case are shown in Figure 2.10. The domain extends one unit length in the x and y directions. A uniform convective velocity in the x-direction ($u_x = 1$) is applied throughout the domain. In this example, the left boundary is a flow inlet (Dirichlet-type boundary condition), the top and bottom boundaries are defined as symmetry conditions (zero-flux Neumann boundary condition), and the right boundary is defined as a flow outlet. The inlet profile of ϕ is given by the so-called "Mexican Hat" function:

$$\phi\left(0,y\right) = \left(1 - 200\left(y - \frac{1}{2}\right)^2\right) e^{\left(-100\left(y-\frac{1}{2}\right)^2\right)}. \tag{2.102}$$

The inlet profile is shown graphically in Figure 2.11. The exact solution for this test case is simple advection of the profile in the x-direction, i.e. $\phi(x, y) = \phi(0, y)$. The same five meshes utilized in Section 2.5.1 are used here for this test case.

To determine the optimum values to be used for σ in all test cases, the code was first run on Mesh2D 1 with different σ values and the results for total number of iterations (needed for convergence at $Residual < 10^{-16}$) and L_2 error norms were obtained. These were plotted in Figures 2.12 and 2.13, for OGRE 2nd and 3rd order schemes, respectively. As expected, there is a small tradeoff in choosing the value for the tuning coefficient σ. The results shown that a higher value for the tuning coefficient σ degrades stability (by taking more iterations to get convergence) but at the same time improves numerical accuracy. On the other hand, a very small σ value also produces a decay in stability. From the plots, it is apparent that, for the OGRE 2nd order scheme, values ranging from 0.9 to 1.2 seem adequate, and for the OGRE 3rd order scheme, values ranging from 1.4 to 1.8 seem acceptable, depending on needs of accuracy and computational resources. Specifically, the optimum value of σ for the 2nd order scheme in terms of accuracy is 1.1. The optimum value of 3rd order scheme in terms of stability is 1.4, while a value representing a reasonable tradeoff between stability and accuracy is 1.8. These values are used for the test cases presented.

The results for L_1 and L_2 error norms for the different meshes using the OGRE 2nd and 3rd order schemes, L-S 2nd and 3rd order schemes, and the Green-Gauss cell-based and node-based schemes (which are of only 2nd order accuracy) are shown in Figures 2.14 and 2.15, respectively, and in Table 2.2. The numerical accuracy orders for the different error norms, obtained from a linear fit of the results on the two finest meshes, are shown in the Table 2.3 and indicated in the plots.

The transport scalar solution on Mesh2D 1, using OGRE 3rd order scheme and L-S 3rd order scheme is shown in Figures 2.16 and 2.17. These figures show that the OGRE scheme produces a smoother distribution of the solution throughout the domain when compared to the solution obtained using the L-S scheme, which clearly seems to produce more numerical dissipation.

The global error norms (L_1 and L_2) listed in Table 2.3 indicate that the new methodology (OGRE) yields measured numerical order of accuracy equal to or higher than the nominal order of accuracy for both 2nd and 3rd order schemes, for all but the L_∞ norm. This is also clearly shown in the plots of L_1 and L_2 norms in Figures 2.14 and 2.15. The overall error for the 2nd order OGRE scheme is approximately equal to the other 2nd order schemes, all of which show better than expected order of accuracy in the L_1 and L_2 norms. The error order is almost exactly equal between the OGRE and L-S 2nd order schemes. The results therefore suggest that there is no real advantage to using the 2nd-order OGRE scheme versus the 2nd-order L-S scheme. This is not surprising, since in the 2nd order versions the two schemes use identical stencils and similar methodologies (minimizing an objective function to solve for derivatives).

For the 3rd-order results, however, the observation is quite different. The L-S scheme yields an order of accuracy almost exactly equal to 3, while the OGRE scheme yields a significantly higher measured order for both the L_1 and L_2 norms. The more rapid decrease in numerical error as mesh size is increased is clearly apparent in Figures 2.14 and 2.15. In addition, the overall error level in the OGRE scheme is consistently lower, by a significant amount, for all mesh sizes including the coarsest. As a result, the L_2 error using OGRE 3rd-order scheme is an order of magnitude lower than the equivalent L-S scheme by Mesh2D 2, and almost two orders of magnitude lower by Mesh2D 5. This highlights one of the primary advantages of the OGRE scheme, namely the use of a stencil comprised only of immediate neighbors, rather than a larger stencil as required by L-S for higher order. As discussed in Section 1, error is proportional to h^k, where h is the characteristic mesh size and k is the order of accuracy. When the stencil size is increased, the effective mesh size h is likewise increased. The results in Figures 2.14 and 2.15 show the effect of both contributions in the OGRE scheme: higher k and lower h. It is expected that this advantage

– smaller effective mesh size due to smaller stencil – will become even more pronounced in contrast to the L-S scheme as the nominal order of the scheme is increased beyond 3.

As discussed in the previous subsection, the OGRE scheme is implicit, and multiple iterations are required to converge the gradient field. This requirement led to an increase in computational expense versus the L-S scheme for the static variable case. In the finite-volume simulations, however, this expense is mitigated. Because the solution algorithm is itself implicit, the gradient field was simply updated once per iteration. Because the convergence rate for the gradients is much more rapid than for the solution of ϕ itself, this was sufficient to allow both the variable and gradient fields to converge together. As a consequence, the overall cost per iteration using OGRE was effectively equal to the cost per iteration using L-S.

As a comparison to evaluate the effect of changing the σ value, using the OGRE 3rd order scheme on Mesh2D 3 and for $\sigma = 1.8$, 694 iterations were needed to obtain full convergence (Residual $< 10^{-16}$), taking 0.1172s per iteration (average), for a total time for convergence of 81.35s; on the other hand, the L-S 3rd order scheme, running on Mesh2D 5, needs 271 iterations to obtain the same convergence, taking 0.5100s per iteration (average), which means a total time for convergence of 138.21s. This shows that OGRE 3rd order scheme obtains a 41.1% reduction in total computational time (81.35s compared to 138.21s) when compared to L-S 3rd order scheme to produce a solution with equivalent precision (e.g. for L_2, 8.01187e-05 compared to 8.65664e-05). Furthermore, using the OGRE 3rd order scheme with $\sigma = 1.4$ on Mesh2D 3 produces a result only slightly worse than L-S 3rd order on Mesh2D 5 (e.g. for L_2, 0.00011227 compared to 8.65664e-05, which is 29.7% bigger), but requires just 426 iterations to obtain convergence (total computational time of 49.93s), or in other words, a 63.9% reduction in computational time.

2.5.3 Spectral Resolution Test Case

The second transport test case seeks to assess the spectral behavior of the new gradient calculation scheme. The case used the same computational geometry described in Section 2.5.2, with a uniform x-direction velocity field. The inlet and outlet boundary condition types are the same as above, but the top and bottom surfaces are assumed to be periodic. A different mesh (mesh2D 6) was created for which each bounding side has 128 equally distributed faces, for the purposes of performing Fourier analysis of the inlet and outlet variable profiles to compare their spectral content. In order to test the new scheme for situations which contain a wide range of scales, an inlet profile for the transport scalar ϕ was synthesized as a sum of Fourier modes. The inlet profile was defined as

$$\phi\left(0,y\right)=\sum_{k=1}^{63}\left[a_k\cos\left(w_k y\right)+b_k\sin\left(w_k y\right)\right]. \tag{2.103}$$

In Eq. (2.103), the phases of the Fourier modes were randomly chosen, and the amplitude for each mode was equal to 1.0, i.e. $(a_k^2+b_k^2=1)$ and $w_k=2\pi k$. As in the previous subsection, the exact solution is given by $\phi(x,y)=\phi(0,y)$. The simulation was run using each of the different gradient calculation schemes, and the solution profile at the outlet was compared to the profile at the inlet. In both cases, the discrete values $\phi(y)$ were transformed to wavenumber space using the Fast Fourier Transform.

Figure 2.18 shows the transformed profile at the outlet for five different cases: 1st order upwind, L-S 2nd and 3rd order schemes, and OGRE 2nd (with $\sigma=1.1$) and 3rd (with $\sigma = 1.8$) order schemes. For all cases, the curves represent an average between four different reconstructed solutions (to minimize random oscillations in the results due to solution on an unstructured mesh). The plot clearly shows the lack of effectiveness in resolving higher wavenumber modes of the solution when using low order schemes. The 1st order upwind cannot even reconstruct completely the lowest wavenumber mode. Consistent with previous results, both of the 2nd order schemes yield very similar results, and are able to resolve at least 90% of the amplitude of modes up to about $k = 11$. As in the previous subsection, a significant improvement in accuracy is apparent between the OGRE and L-S 3rd order versions. OGRE resolves approximately twice the wavenumber modes compared to L-S. While the L-S scheme can resolve about 90% of the amplitudes up to about wavenumber 12, OGRE can do the same up to wavenumber 27.

Figures 2.19 and 2.20 show the transport scalar solution on Mesh2D 6 for the spectral resolution test case, using OGRE (with $\sigma = 1.8$) and L-S 3rd order schemes, respectively. The inlet profiles are the same for both cases. Again, it is clear the lower-dissipative feature of the OGRE scheme when compared to the L-S scheme.

These results not only reinforce those shown in Figures 2.12 and 2.13, they also have important implications for applications requiring high-resolution solution algorithms for a range of spatial scales. For example, in finite-volume CFD applications, these results suggest that the OGRE scheme may provide better numerical accuracy in large-eddy simulations.

Table 2.1

Mesh specifications used in all 2D test cases.

Mesh #	# of cells	# of faces	# of nodes
Mesh2D 1	506	789	284
Mesh2D 2	2,030	3,075	1,076
Mesh2D 3	8,110	12,285	4,176
Mesh2D 4	32,450	48,915	16,466
Mesh2D 5	129,758	195,117	65,360
Mesh2D 6	36,880	55,448	18,697

Table 2.2

Error norms for the 2D simple scalar transport test case on different schemes and meshes.

Mesh -	G-G	G-G	L-S	OGRE	L-S	OGRE	OGRE
error norm	2nd O	2nd O	2nd O	2nd O	3rd O	3rd O	3rd O
	(cell-based)	(node-based)		$\sigma = 1.1$		$\sigma = 1.4$	$\sigma = 1.8$
M1 - L_1	0.0724981	0.0670649	0.0695438	0.072058	0.0618172	0.0197917	0.0171327
M1 - L_2	0.124949	0.120423	0.119616	0.123871	0.106475	0.0361609	0.030788
M1 - L_∞	0.503589	0.579264	0.517218	0.530855	0.465877	0.177237	0.154563
M2 - L_1	0.0241229	0.0222775	0.0228861	0.0229495	0.0167143	0.00144461	0.00104648
M2 - L_2	0.0476369	0.0454491	0.0448132	0.0450637	0.0339241	0.00305462	0.0022206
M2 - L_∞	0.227741	0.235695	0.215202	0.212498	0.17096	0.0180364	0.0132212
M3 - L_1	0.00436268	0.00407342	0.00407909	0.00406447	0.00247131	5.18089e-05	3.83393e-05
M3 - L_2	0.00912019	0.00876585	0.00846534	0.00844122	0.00538231	0.00011227	8.01187e-05
M3 - L_∞	0.0520212	0.0585791	0.050249	0.0490998	0.0313586	0.000848392	0.000725849
M4 - L_1	0.000848163	0.000617576	0.000707685	0.000705312	0.000300152	3.12048e-06	2.82242e-06
M4 - L_2	0.00177065	0.00133975	0.00147935	0.00147122	0.000673436	7.05356e-06	6.68157e-06
M4 - L_∞	0.0155317	0.0106224	0.00902852	0.00857651	0.00422446	9.21876e-05	0.000108515
M5 - L_1	0.000189212	0.00011939	0.000139151	0.000138738	3.90086e-05	2.64245e-07	2.81981e-07
M5 - L_2	0.000387949	0.000253956	0.00027875	0.000277532	8.65664e-05	7.05961e-07	7.49947e-07
M5 - L_∞	0.00740028	0.00200627	0.00162878	0.00155126	0.000551608	1.39749e-05	1.35922e-05

Table 2.3

Numerical orders of accuracy for the error in each scheme between the two finest meshes for the 2D simple scalar transport test case.

error	G-G	G-G	L-S	OGRE	L-S	OGRE	OGRE
norm	2nd O	2nd O	2nd O	2nd O	3rd O	3rd O	3rd O
	(cell-based)	(node-based)		$\sigma = 1.1$		$\sigma = 1.4$	$\sigma = 1.8$
L_1	2.16	2.37	2.35	2.35	2.94	3.56	3.32
L_2	2.19	2.40	2.41	2.41	2.96	3.32	3.16
L_∞	1.07	2.40	2.47	2.47	2.94	2.72	3.00

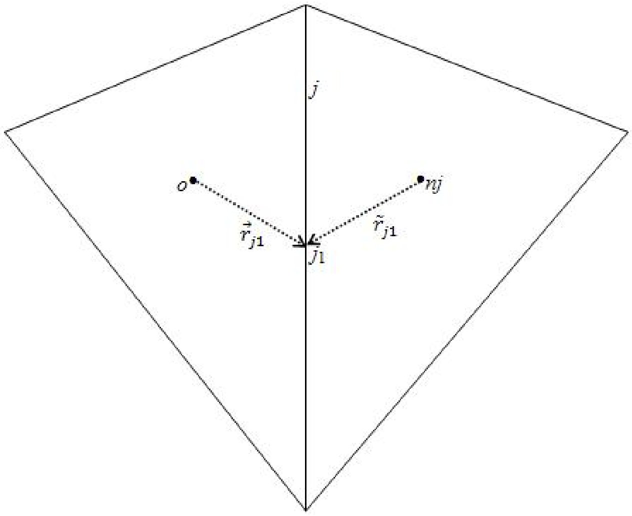

Figure 2.1

2D OGRE implementation for interior cells: center cell centroid, face neighbor centroid and face integration point (just one, for the 2nd order case), with respective connecting vectors.

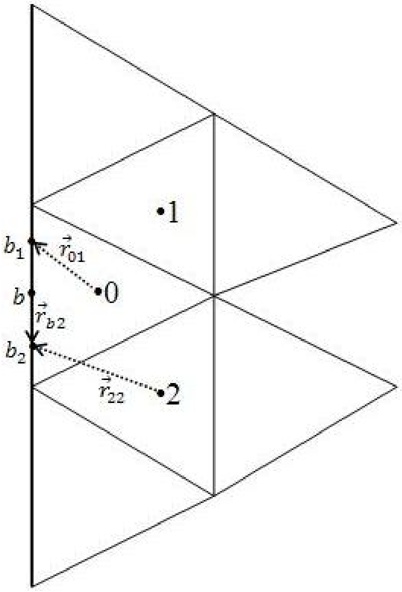

Figure 2.2

2D boundary condition implementation.

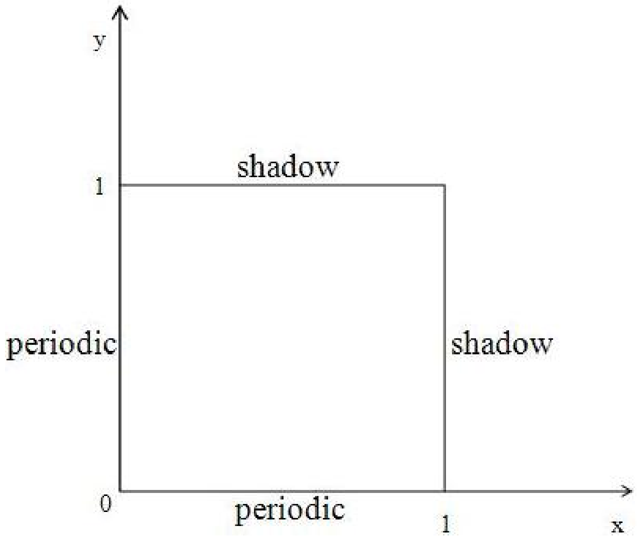

Figure 2.3

2D geometry and boundary conditions for the static variable test case.

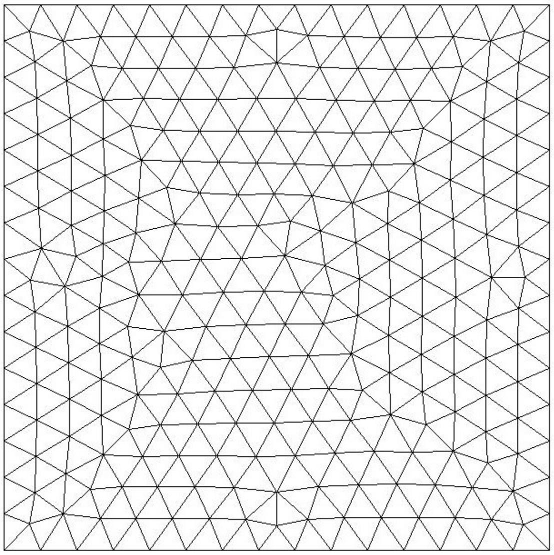

Figure 2.4

Mesh2D 1.

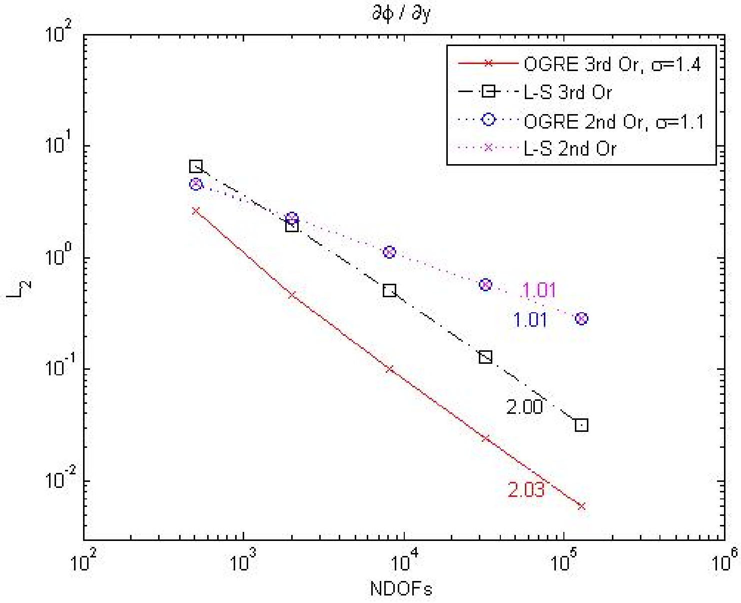

Figure 2.5

L_2 error norm values of $\dfrac{\partial \phi}{\partial y}$ for the 2D static variable test case.

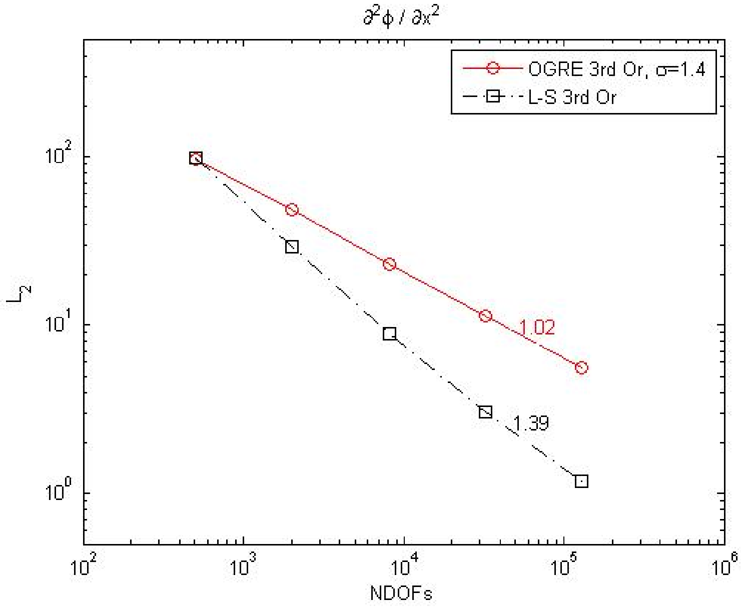

Figure 2.6

L_2 error norm values of $\dfrac{\partial^2 \phi}{\partial x^2}$ for the 2D static variable test case.

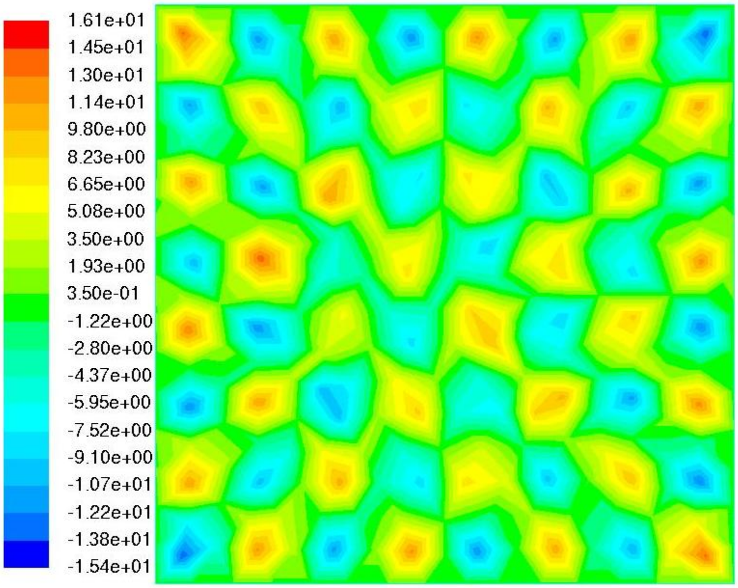

Figure 2.7

OGRE 3rd order solution of $\dfrac{\partial \phi}{\partial y}$ using σ = 1.4 for the static variable test case on Mesh2D 1.

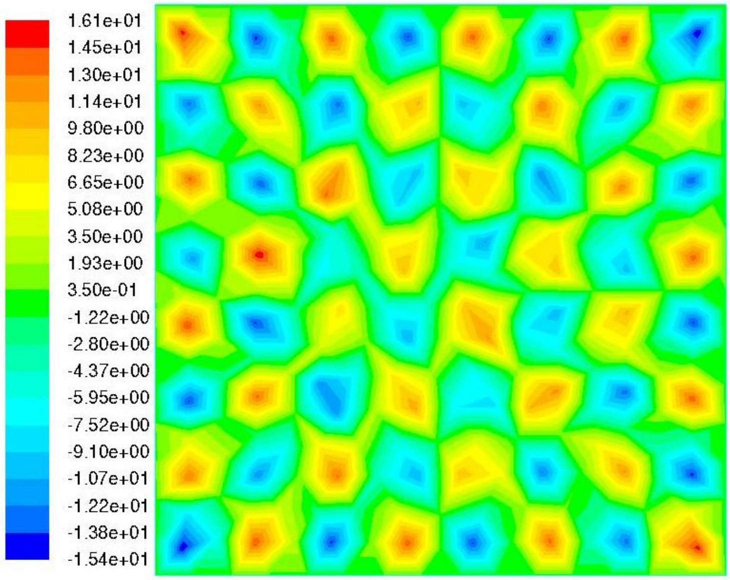

Figure 2.8

Analytical solution of $\dfrac{\partial \phi}{\partial y}$ for the static variable test case on Mesh2D 1.

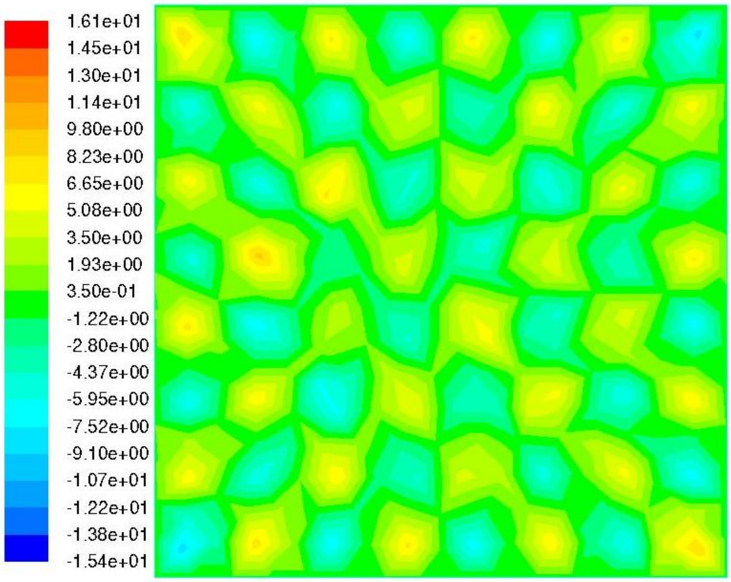

Figure 2.9

L-S 3rd order solution of $\frac{\partial \phi}{\partial y}$ for the static variable test case on Mesh2D 1.

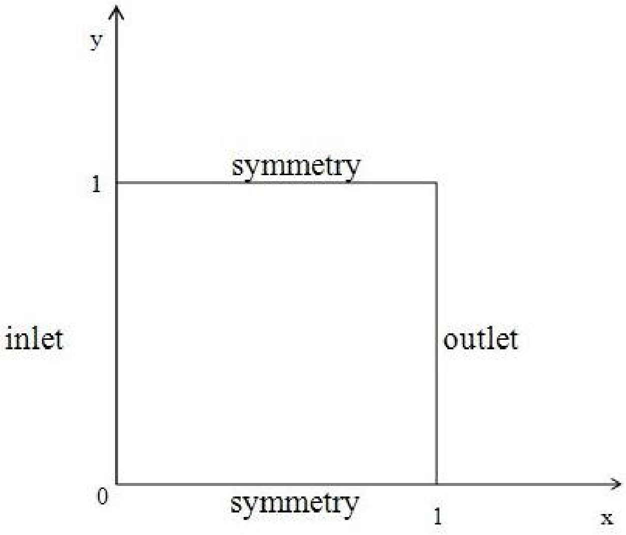

Figure 2.10

2D geometry and boundary conditions for the simple scalar transport test case.

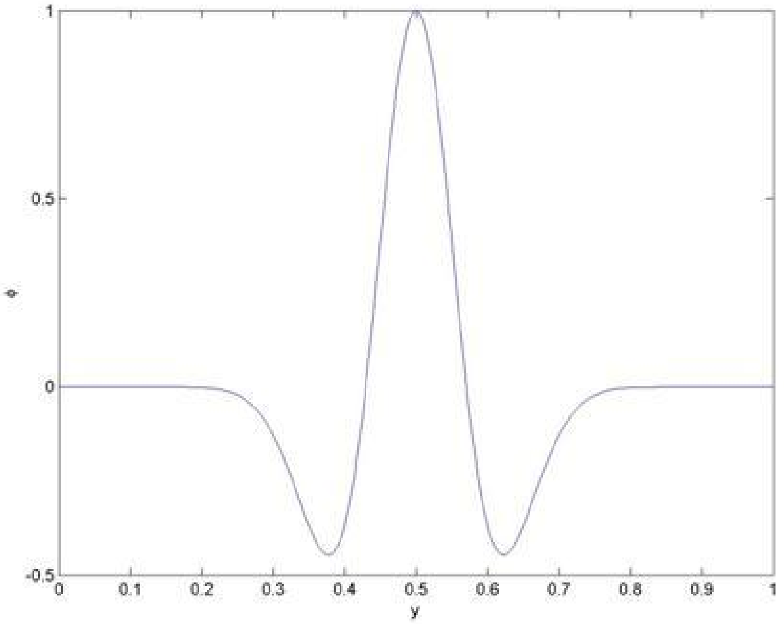

Figure 2.11

2D inlet profile: Mexican Hat function.

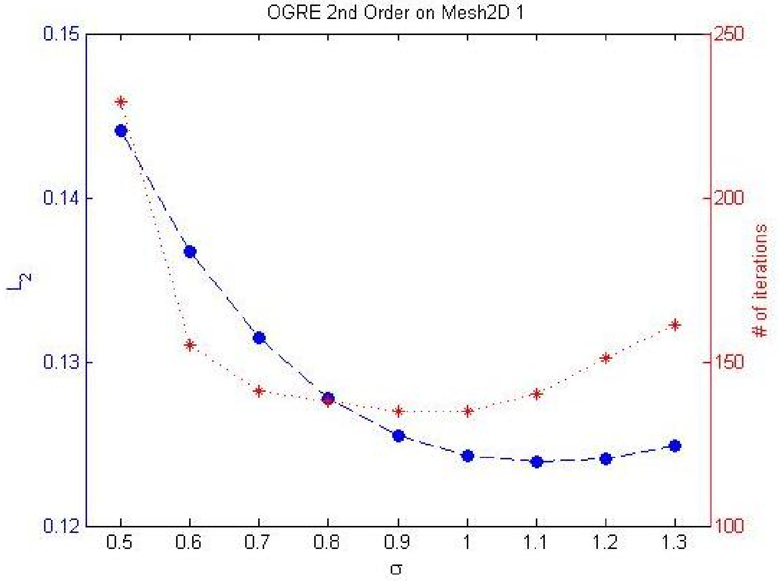

Figure 2.12

σ effect on OGRE 2nd order, Mesh2D 1.

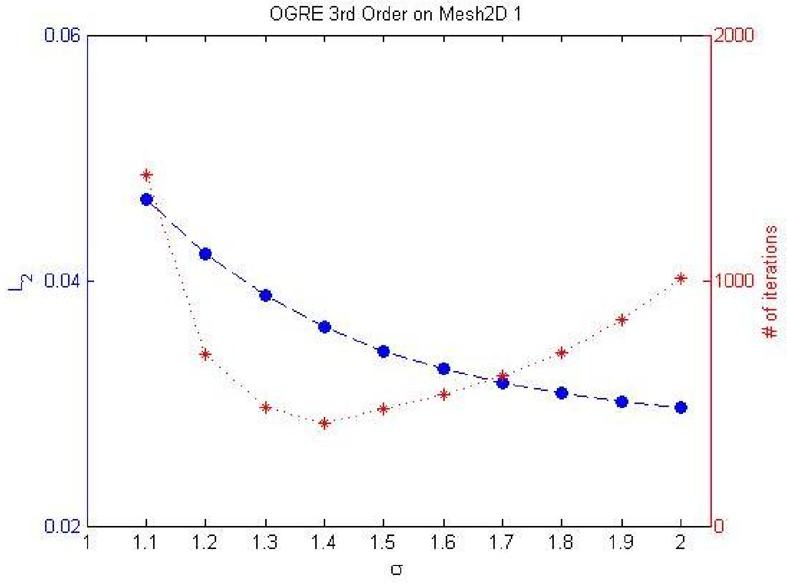

Figure 2.13

σ effect on OGRE 3rd order, Mesh2D 1.

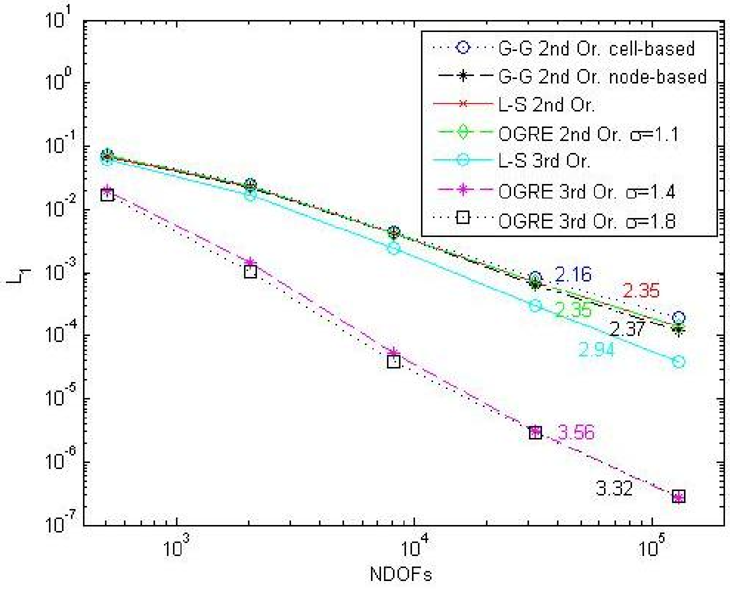

Figure 2.14

L_1 error norms for the 2D simple scalar transport test case.

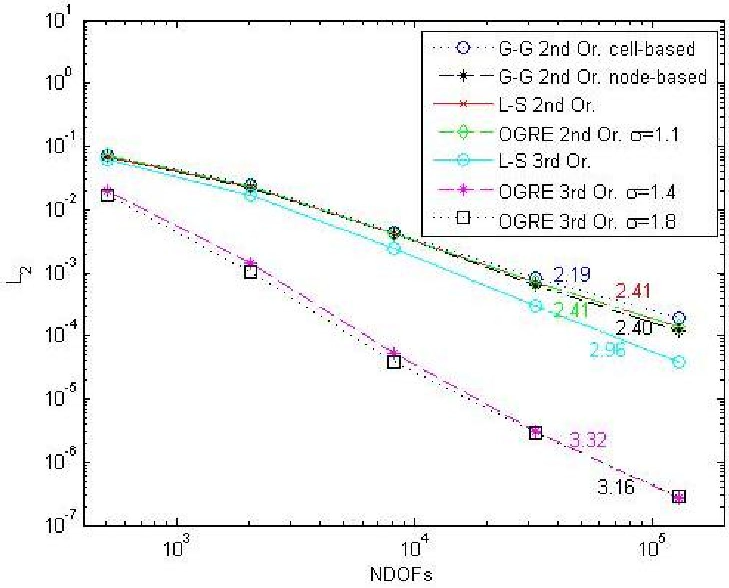

Figure 2.15

L_2 error norms for the 2D simple scalar transport test case.

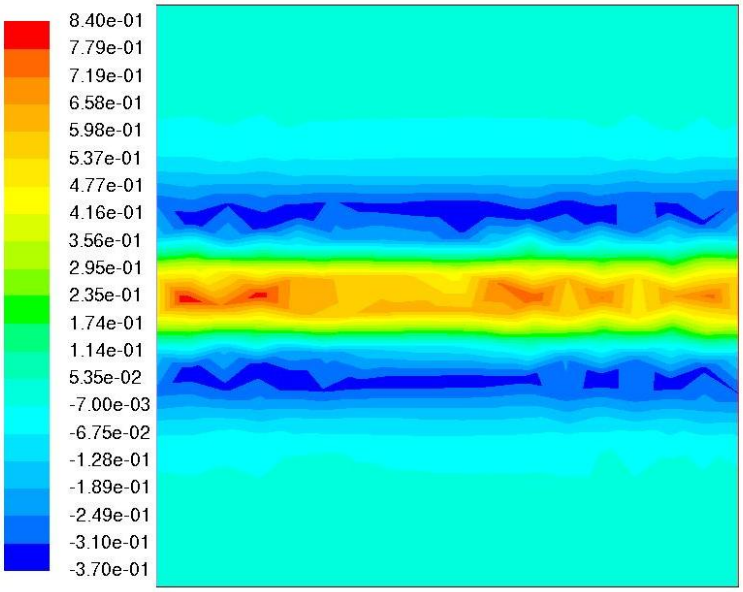

Figure 2.16

ϕ distribution throughout the domain on Mesh2D 1 using OGRE 3rd order scheme, σ=1.4, for the simple scalar transport test case.

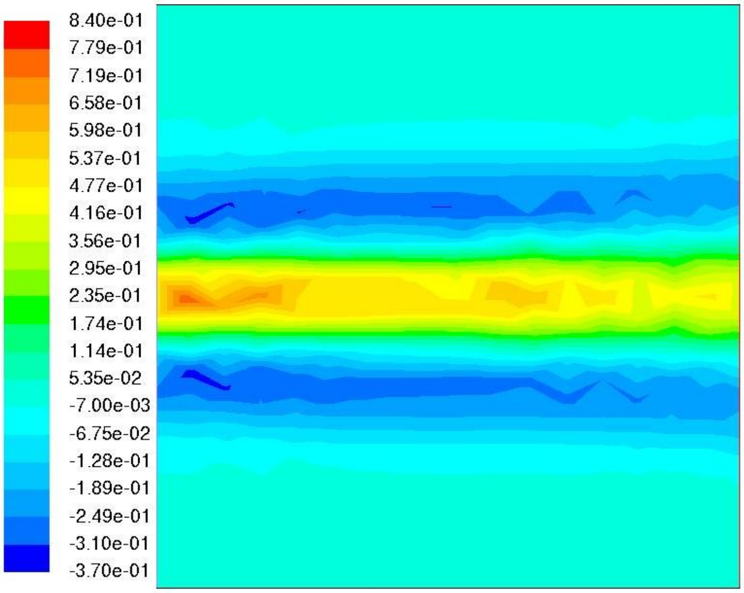

Figure 2.17

ϕ distribution throughout the domain on Mesh2D 1 using L-S 3rd order scheme, for the simple scalar transport test case.

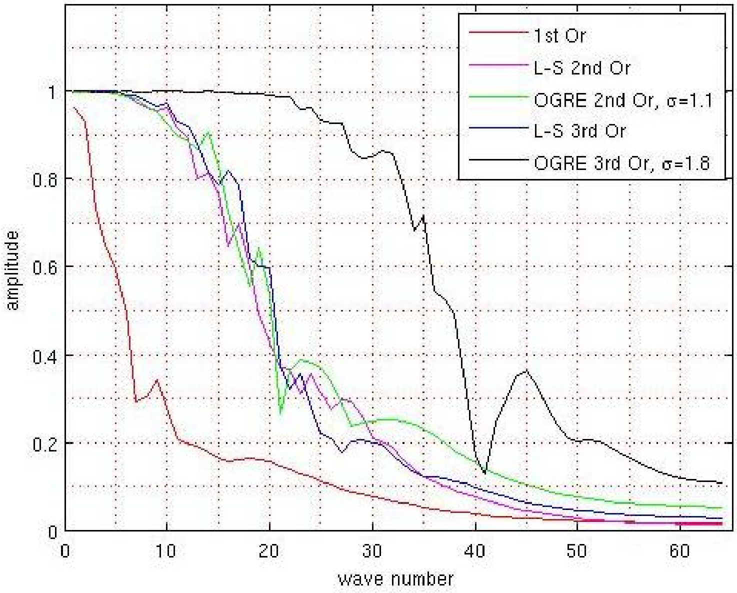

Figure 2.18

FFT outlet reconstruction using different schemes for the 2D spectral analysis test case.

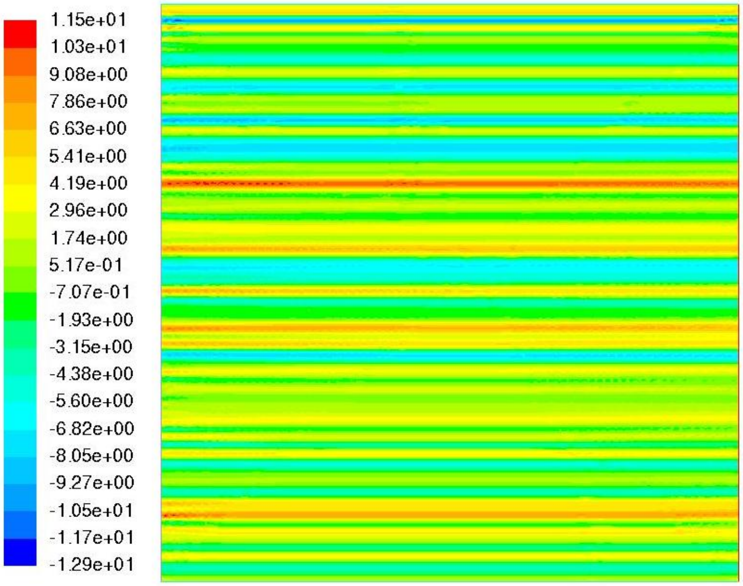

Figure 2.19

ϕ distribution throughout the domain on Mesh2D 6 using OGRE 3rd order scheme with $\sigma = 1.8$, for the 2D spectral analysis test case.

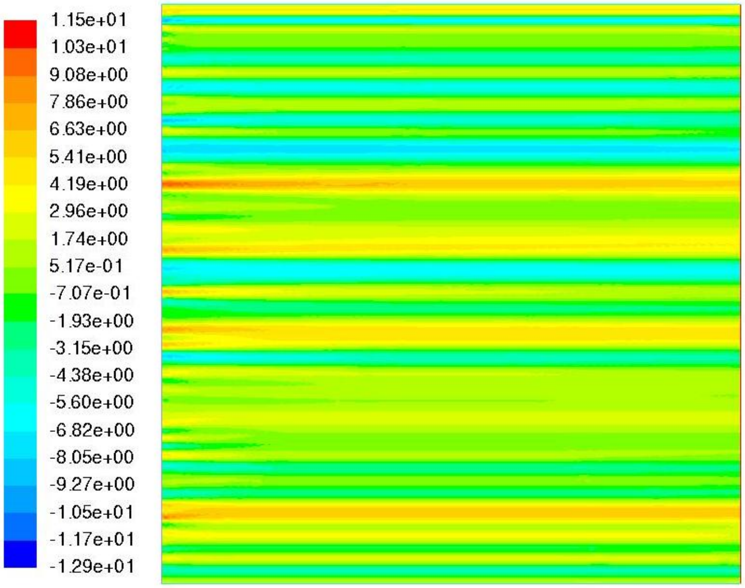

Figure 2.20

ϕ distribution throughout the domain on Mesh2D 6 using L-S 3rd order scheme, for the 2D spectral analysis test case.

CHAPTER 3

A 3D OPTIMIZATION-BASED METHOD FOR HIGH ORDER GRADIENT CALCULATION ON UNSTRUCTURED MESHES

3.1 Introduction

3.1.1 Background

In a wide range of application fields (e.g., computer graphics, electro-magnetics, solid mechanics, and fluid mechanics), an essential step in many different numerical methods is the precise calculation of local derivatives in discrete variable fields.

Methods for numerical gradient calculation should be accurate, stable, and computationally efficient to be useful in practical applications. This paper presents the 3D extension of a new method for high-order gradient calculation on unstructured meshes initially developed in [4]. Because of the authors' background, the study is motivated by a goal to improve finite-volume CFD simulations. However, the method is a general approach which can be used for any of applications mentioned above.

For structured meshes, the application of high-order-accurate schemes for gradient calculation have been systematically studied for decades, using finite-difference (FD), finite-element (FE) and finite-volume (FV) methods. An interested reader can refer to the comprehensive review article by Ekaterinaris [10] for a thorough discussion of high-order schemes on structured grids, with a focus on CFD applications.

High-order derivative calculation on unstructured meshes has been studies less extensively. Two of the most common approaches for derivative calculation in 2nd-order numerical schemes are the Green-Gauss (G-G) method and the Least-Squares (L-S) method [7]. For higher order, important contributions on unstructured-meshed based computations can be found in Barth and Jespersen [3] and Barth and Frederickson [2], who proposed k-exact schemes using a stencil variable values in the neighborhood of an area of interest. The use of non-fixed stencils were applied when Essentially Non-Oscillatory (ENO) schemes, traditionally used in the structured grid community, were extended to unstructured meshes [1,9,22,27]. Later, Weighted Essentially Non-Oscillatory (WENO) schemes were also extended to unstructured meshes [11,13]. An interested reader can refer to [26] for a review of ENO and WENO methods. High-order methods have also been applied using Discontinuous Galerkin (DG) [6] and Spectral Volume (SV) [12] methods. Ollivier-Gooch [23] presented results for finite-volume simulations using high-order reconstructions. The authors recommended that high order capabilities should be included in numerical schemes to ensure at least quadratic (or higher) reconstruction at boundaries. A more detailed discussion is available in [4].

While much progress has been made to date, there is still for improvement with regard to robust and efficient high-order methods in unstructured meshed-based solvers. For CFD applications on unstructured meshes, for example, many of the approaches mentioned above are based on the L-S scheme or its variants for local gradient calculation, but they tend to become numerically stiff as the order of accuracy (as well size of the numerical stencil) increases. Because unstructured meshes are often desired for their improved ability to simulate flow fields on using complex geometries and for permitting flexible mesh adaptation near specific regions of interest, improved methods for high-order gradient calculation can lead to significant improvements in these schemes. Nonetheless, even after decades of advances, unstructured-meshed based solvers still present limitations both in terms of accuracy and efficiency, especially for simulations of complex flow fields. Despite the potential advantages of high-order methods, the majority of CFD simulations remain second order.

3.1.2 Paper Outline

This paper presents a 3D extension to the problem of high-order derivative computation on unstructured meshes developed in [4]. Specifically, an implicit, compact scheme has been proposed which in theory allows derivative calculations to arbitrary order. Stencils are constructed only from nearest neighbors, and the gradient field is obtained through an iterative solution process. The method is based on minimization of an objective function that reflects the degree of mismatch between reconstructions in neighboring computational control volumes. The paper is organized as follows. Section 3.2 outlines a simple test application for the high-order scheme, namely finite-volume solution of a 3D scalar advection equation, and briefly describes the role of the calculated numerical derivatives in the solution of the problem. Section 3.3 presents the development of the new methodology up to 3rd order accuracy. Section 3.4 describes the method for boundary condition implementation. Section 3.5 presents numerical validation and initial results. Chapter 4 draws conclusions and describes future work.

3.2 Sample Application and Derivative Calculation

3.2.1 Finite-Volume Solution of Scalar Advection Equation

This section seeks to provide context with regard to derivative calculation, using the example of least-square (L-S) schemes for finite-volume (FV) CFD methods. Consider a steady-state convective transport equation of a passive scalar variable, ϕ, in a known velocity field $\vec{U}(x)$. The conservation equation is given by:

$$\nabla \cdot \left(\rho \vec{U} \phi\right) = \frac{\partial}{\partial x_j} \left(\rho u_j \phi\right) = 0, \tag{3.1}$$

where ρ is the fluid density, $\vec{U}$ (or u_j) is the fluid velocity vector, and ϕ is an arbitrary scalar variable. In order to apply the FV method, the flow domain is discretized into non-overlapping control volumes (cells), and Eq. (3.1) is integrated over each control volume. Applying the Gauss divergence theorem yields

$$\int_{\Omega} \nabla \cdot \left(\rho \vec{U} \phi\right) \, dV = \int_{\partial\Omega} \rho \phi \vec{U} \cdot \hat{n} \, dA = \int_{\partial\Omega} \rho \phi \left(u_j n_j\right) \, dA = 0, \tag{3.2}$$

where Ω denotes volume integration over the cell, $\partial\Omega$ denotes area integration over the bounding surface of the cell, and $\hat{n}$ is the outward-pointing unit normal vector on the cell surface. For general polyhedron cells (in 3D), the bounding surface is comprised of a finite number of discrete (polygonal) faces. The integral may therefore be expressed as

$$\int_{\partial\Omega} \rho \phi \left(u_j n_j\right) \, dA = \sum_{f=1}^{N_f} \int_{\partial\Omega_f} \rho_f \phi_f \left(u_j n_j\right)_f \, dA_f = 0. \tag{3.3}$$

The surface integration is performed as the sum of integrals over each polygonal face, N_f is the number of faces bounding the cell, and $\partial\Omega_f$ denotes integration over the face area.

To simplify notation, we further adopt the conventional definition of the convective face flux, F_f, defined for each face as

$$F_f = \int_{\partial\Omega_f} \rho_f \phi_f \left(u_j n_j\right)_f \, dA_f. \tag{3.4}$$

The goal of numerical discretization is to approximate the integral in Eq. (3.4) based on available cell data, since the discrete values of the dependent variable ϕ (as well as ρ and $\vec{U}$ in general) are stored at the control volume centers (for a cell-centered algorithm, the control volume centers are the same as the cell centroids).

For the purpose of developing a new discretization scheme based on the proposed gradient calculation scheme, a simple upwind method can be used to determine the face fluxes based on the neighbor cell variable reconstructions. For a known velocity field, the convective flux can be expressed as

$$F_f \approx \dot{m}_f \bar{\phi}_f, \tag{3.5}$$

where $\dot{m}_f$ is the mass flow rate across the given face, given by

$$\dot{m}_f = \int_{\partial\Omega_f} \rho_f \left(u_j n_j\right)_f \, dA_f, \tag{3.6}$$

and $\bar{\phi}_f$ is the mass-averaged value of the transport scalar ϕ on face f, which can be expressed as

$$\bar{\phi}_f = \frac{1}{\dot{m}_f} \int_{\partial\Omega_f} \rho_f \phi_f \left(u_j n_j\right)_f \, dA_f. \tag{3.7}$$

For a pure upwind convective scheme, the value of ϕ at any point on face f can be reconstructed by projection from the upwind cell centroid as

$$\phi_f = \phi_U + \left(\frac{\partial \phi}{\partial x_i}\right)_U x_{i,f} + H.O.T., \tag{3.8}$$

where $H.O.T.$ here represents the high order terms of the expansion. The integral in Eq. (3.7) may be evaluated using any appropriate quadrature method of sufficiently high order, in combination with Eq. (3.8). For example, Gauss quadrature ([25]) is used for the test cases in this paper.

It is possible to rewrite (3.5) as

$$F_f \approx \dot{m}_f \bar{\phi}_f = \dot{m}_f \phi_U + \dot{m}_f \left(\bar{\phi}_f - \phi_U \right). \tag{3.9}$$

For an implicit upwind solution method, the first term on the right hand side can be treated implicitly during the linear solve step, and the second term, incorporating the numerical derivatives, can be treated explicitly. As a result, the contribution of all terms of order greater than one is included as a source term during each outer (Newton) iteration. This approach has been found to be stable in all of the test cases considered.

3.2.2 Derivative Calculation Methods

The approach outlined above for solution of the scalar advection equation requires numerical computation of first derivatives for a second-order scheme, and first and second derivatives for a third order scheme, etc. Since our approach uses a cell-centered method, the derivatives are computed at the centroid of each computational cell. The necessary error order of accuracy for a scheme of order k and derivative of order m is equal to $k - m$. As discussed above, necessity of accurate derivative calculation is not unique to FV methods. Several other applications [8, 14, 15] also make use of local gradient calculation.

For higher-order gradient calculations, perhaps the most common method is the cell-based Least-Squares (L-S) method. The L-S scheme is based on a Taylor Series expansion about the centroid of the cell in which the gradient is being computed. In order to develop higher-order schemes using L-S, the Taylor series expansion includes derivatives of the dependent variable greater than one. To implement such a scheme, the neighbor stencil can be increased to include second-level cell neighbors (i.e., neighbors of neighbors), third-level, etc., so that there is sufficient neighbor information to create a system that is mathematically consistent (i.e., a determined or over-determined system) in the region of the cell of interest, in order to calculate all of the necessary derivatives for a given reconstruction order.

As shown in recent publications by Ollivier-Gooch and his co-workers [16–24], there are some inherent challenges with this approach related to memory cost, calculation speed, matrix stiffness, and stability.

The L-S method can be classified as one type of optimization (minimum energy) based method, in which the (local) objective function is a measure of the degree of mismatch between the values of ϕ obtained from a reconstruction (polynomial expansion) about the cell centroids, and the values of ϕ in the neighbors making up the stencil support. The new method for derivative calculation proposed in this paper seeks to address the difficulties mentioned above, and is herein referred to as the Optimization-based Gradient REconstruction (OGRE) method. The key aspect of the approach is that it relies on stencils comprised only of immediate neighbors, but includes higher-order information via an iterative solution of a system of equations. This has the potential to facilitate (speed up) convergence, by substantially reducing the total cost of calculations for the same level of accuracy versus traditional (i.e. least squares) approaches. Conceptually, the approach can be summarized as the minimization of a global objective function that defines the degree to which the individual cell reconstructions (a variable and its derivatives) match the reconstructions of their immediate neighbors. Like the L-S approach, the new method is an optimization based method. In contrast to L-S, the mismatch used to define the objective function is based on the values of ϕ as well as its derivatives, resulting in a smaller required stencil, and potentially increasing the absolute accuracy of the scheme versus the conventional L-S approach.

3.3 OGRE Scheme Methodology

This section shows the derivation of the OGRE scheme to calculate derivatives of arbitrary order on structured or unstructured computational meshes. The development for the 2nd and 3rd order variants of the new scheme is presented; extension to higher order follows the approach presented here, and is relatively straightforward. This work is the 3D extension of the 2D methodology presented in [4]. The development and implementation are demonstrated in detail in order to clarify the reasoning behind the final equations used in the algorithm.

Figure 3.1 shows an example of a 3D unstructured cell arrangement for the 3rd order variant of the scheme.

3.3.1 Second Order Scheme

For a 2nd order FV scheme, it is necessary to approximate the first order spatial derivatives (in x, y and z) for a given transport variable ϕ at the centroid of each control volume (cell), with an error order of accuracy equal to one. The main conceptual goal behind the

OGRE method is to obtain a specific objective function, through geometrical and numerical reasoning, that then can be minimized to determine the values of the derivatives.

The connecting vector between the center cell centroid o and the face centroid $j1$ (which is the face quadrature location for a second order scheme), is given by $\vec{r}_{j1}$, and the connecting vector between face centroid $j1$ and the neighbor cell centroid nj, is given by $\tilde{r}_{j1}$. Those connecting vectors are defined by:

$$\begin{aligned} \vec{r}_{j1} &= \vec{X}_{j1} - \vec{X}_o = \Delta x_{j1}\vec{i}_x + \Delta y_{j1}\vec{i}_y + \Delta z_{j1}\vec{i}_z, \\ \widetilde{r}_{j1} &= \vec{X}_{j1} - \vec{X}_{nj} = \widetilde{\Delta x}_{j1}\vec{i}_x + \widetilde{\Delta y}_{j1}\vec{i}_y + \widetilde{\Delta z}_{j1}\vec{i}_z. \end{aligned} \tag{3.10}$$

The following notation is also used throughout the derivation calculation (for both 2nd and 3rd order schemes):

o - center cell centroid.

j - face index.

nj - neighbor cell centroid.

$\phi^*_a\big|_b$ - projection of the value of ϕ (or its gradients) from a to b, through Taylor series expansion
(these expansions must follow the given order of accuracy of the scheme).

For a 2nd order error scheme, all polynomial expansions of the dependent variable must be truncated at the first order derivatives. In order to calculate the first derivatives needed for the 2nd order scheme, two measures of disagreement between neighboring cell expansions are postulated:

1) Measure of disagreement between the scalar value projections from each cell at the face quadrature point (face centroid):

$$(\Delta\phi)_{j1} = \phi^*_o\big|_{j1}^{(new)} - \frac{1}{2}\left(\phi^*_o\big|_{j1} + \phi^*_{nj}\big|_{j1}\right)^{(current)}, \tag{3.11}$$

2) Measure of disagreement between the first derivative projections from each cell at the face quadrature point (face centroid):

$$\Delta\left(\nabla\phi\right)_{j1}\cdot\vec{r}_{j1} = \left(\nabla\phi^*_o\Big|_{j1}^{(new)} - \frac{1}{2}\left(\nabla\phi^*_o\Big|_{j1} + \nabla\phi^*_{nj}\Big|_{j1}\right)^{(current)}\right)\cdot\vec{r}_{j1}. \tag{3.12}$$

These two measures are analyzed now separately, starting with the first measure of disagreement.

3.3.1.1 First Measure of Disagreement

Expanding the first term on the RHS of Eq. (3.11) produces

$$\phi^*_o\Big|_{j1} = \phi_o + \nabla\phi\Big|_o\cdot\vec{r}_{j1} = \phi_o + \frac{\partial\phi}{\partial x}\bigg|_o \Delta x_{j1} + \frac{\partial\phi}{\partial y}\bigg|_o \Delta y_{j1} + \frac{\partial\phi}{\partial z}\bigg|_o \Delta z_{j1}. \tag{3.13}$$

Also, the second term on RHS of Eq. (3.11) can be denoted $\tilde{F}_{j1}$, which represents an average of the (first-order) Taylor Series expansions of the variable from each of the cells on either side of the face to the face integration point:

$$\begin{aligned}\tilde{F}_{j1} &= \frac{1}{2}\left(\phi^*_o\Big|_{j1} + \phi^*_{nj}\Big|_{j1}\right)^{(current)} = \frac{1}{2}\left(\phi_o + \nabla\phi\Big|_o\cdot\vec{r}_{j1} + \phi_{nj} + \nabla\phi\Big|_{nj}\cdot\tilde{r}_{j1}\right)^{(curr)}\\ &= \frac{1}{2}\left(\phi_o + \frac{\partial\phi}{\partial x}\bigg|_o \Delta x_{j1} + \frac{\partial\phi}{\partial y}\bigg|_o \Delta y_{j1} + \frac{\partial\phi}{\partial z}\bigg|_o \Delta z_{j1} + \right.\\ &\qquad\left. \phi_{nj} + \frac{\partial\phi}{\partial x}\bigg|_{nj} \widetilde{\Delta x}_{j1} + \frac{\partial\phi}{\partial y}\bigg|_{nj} \widetilde{\Delta y}_{j1} + \frac{\partial\phi}{\partial z}\bigg|_{nj} \widetilde{\Delta z}_{j1}\right)^{(current)}.\end{aligned} \tag{3.14}$$

Redefining the first measure of disagreement as f_{j1} and substituting Eq. (3.14), (3.11) results in

$$f_{j1} = \phi_o + \frac{\partial\phi}{\partial x}\bigg|_o \Delta x_{j1} + \frac{\partial\phi}{\partial y}\bigg|_o \Delta y_{j1} + \frac{\partial\phi}{\partial z}\bigg|_o \Delta z_{j1} - \tilde{F}_{j1}. \tag{3.15}$$

3.3.1.2 Second Measure of Disagreement

Based on a zero-order Taylor Series expansion for the gradients:

$$\nabla \phi_o^*\big|_{j1} = \nabla \phi_o \tag{3.16}$$

and

$$\nabla \phi_{nj}^*\big|_{j1} = \nabla \phi_{nj}, \tag{3.17}$$

since the 2nd order (and higher) derivatives are considered to be zero in a second order scheme. This reduces (3.12) to simply

$$\Delta \left(\nabla \phi\right)_{j1} \cdot \vec{r}_{j1} = \left(\nabla \phi_o^{(new)} - \frac{1}{2}\left(\nabla \phi_o + \nabla \phi_{nj}\right)^{(current)}\right) \cdot \vec{r}_{j1}, \tag{3.18}$$

which, after redefining the second measure of disagreement as g_{j1}, (3.12) can be rewritten as

$$g_{j1} = \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_{j1} + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_{j1} + \left.\frac{\partial \phi}{\partial z}\right|_o \Delta z_{j1} - \tilde{G}_{j1}, \tag{3.19}$$

where $\tilde{G}_{j1}$ is given by

$$\begin{aligned}\tilde{G}_{j1} &= \frac{1}{2}\left(\left(\nabla \phi_o + \nabla \phi_{nj}\right)^{(current)}\right) \cdot \vec{r}_{j1} \\ &= \frac{1}{2}\left(\left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_{j1} + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_{j1} + \left.\frac{\partial \phi}{\partial z}\right|_o \Delta z_{j1} + \right. \\ &\qquad \left. \left.\frac{\partial \phi}{\partial x}\right|_{nj} \Delta x_{j1} + \left.\frac{\partial \phi}{\partial y}\right|_{nj} \Delta y_{j1} + \left.\frac{\partial \phi}{\partial z}\right|_{nj} \Delta z_{j1}\right)^{(current)} .\end{aligned} \tag{3.20}$$

Equations (3.23) and (3.19) are used to develop an optimization procedure that is described in the following section.

3.3.2 Optimization Procedure for the Second Order Scheme

The goal of the optimization procedure is to minimize, in a least-squares sense, the disagreement between neighbor projections at each face centroid represented by Eqs. (3.23) and (3.19). One consideration is the relative weight to be given to those two measures. In order to control the accuracy of the method, a tuning parameter is incorporated through the use of a weighting coefficient σ. The parameter σ modifies the relative weight given to the two measures of disagreement defined by Eqs. (3.23) and (3.19). It is expected that, in general, within an acceptable range, increasing the value of σ should improve the scheme accuracy, while decreasing its value should make the scheme more stable. To include this parameter, the two measures f_{j1} and g_{j1}, given by Eqs. (3.23) and (3.19), respectively, can be rewritten as

$$f_{j1}^* = \sigma f_{j1} = \sigma \left[\phi_o + \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_{j1} + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_{j1} + \left.\frac{\partial \phi}{\partial z}\right|_o \Delta z_{j1} - \tilde{F}_{j1} \right] \tag{3.21}$$

and

$$g_{j1}^* = g_{j1} = \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_{j1} + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_{j1} + \left.\frac{\partial \phi}{\partial z}\right|_o \Delta z_{j1} - \tilde{G}_{j1}, \tag{3.22}$$

where again $\tilde{F}_{j1}$ and $\tilde{G}_{j1}$ are given by Eqs. (3.14) and (3.20), respectively.

By using (3.21) and (3.22), it is possible to construct the following quadratic functional

$$F = \sum_{j=1}^{\#neig} \left({f_{j1}^*}^2 + {g_{j1}^*}^2 \right), \tag{3.23}$$

which must be minimized in order to find the values of the first order derivatives at any cell centroid. F represents a local objective function for any given cell. Summing over all cells yields a convex global objective function that can be minimized to find all cell (control volume) derivatives. Minimization of the function F is straightforward, producing the following three equations:

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = 0 \tag{3.24}$$

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = 0. \tag{3.25}$$

and

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial z} \right|_o \right)} = 0 \tag{3.26}$$

Those equations can be expressed as

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = 2 \sum_{j=1}^{\#neig} \left\{ f_{j1}^* \frac{\partial f_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} + g_{j1}^* \frac{\partial g_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} \right\} = 0, \tag{3.27}$$

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = 2 \sum_{j=1}^{\#neig} \left\{ f_{j1}^* \frac{\partial f_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} + g_{j1}^* \frac{\partial g_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} \right\} = 0, \tag{3.28}$$

and

$$\frac{\partial F}{\partial \left(\left. \frac{\partial \phi}{\partial z} \right|_o \right)} = 2 \sum_{j=1}^{\#neig} \left\{ f_{j1}^* \frac{\partial f_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial z} \right|_o \right)} + g_{j1}^* \frac{\partial g_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial z} \right|_o \right)} \right\} = 0, \tag{3.29}$$

where the internal partial derivatives are given by:

$$\frac{\partial f_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = \sigma \Delta x_{j1}, \tag{3.30}$$

$$\frac{\partial g_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial x} \right|_o \right)} = \Delta x_{j1}, \tag{3.31}$$

$$\frac{\partial f_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = \sigma \Delta y_{j1}, \tag{3.32}$$

$$\frac{\partial g_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial y} \right|_o \right)} = \Delta y_{j1}. \tag{3.33}$$

$$\frac{\partial f_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial z} \right|_o \right)} = \sigma \Delta z_{j1}, \tag{3.34}$$

$$\frac{\partial g_{j1}^*}{\partial \left(\left. \frac{\partial \phi}{\partial z} \right|_o \right)} = \Delta z_{j1}. \tag{3.35}$$

Substituting those partial derivatives into (3.27), (3.28) and (3.29), and dividing all equations by 2, yields

$$\sum_{j=1}^{\#neig} \left\{ f_{j1}^* \sigma \Delta x_{j1} + g_{j1}^* \Delta x_{j1} \right\} = 0. \tag{3.36}$$

$$\sum_{j=1}^{\#neig} \left\{ f_{j1}^* \sigma \Delta y_{j1} + g_{j1}^* \Delta y_{j1} \right\} = 0. \tag{3.37}$$

and

$$\sum_{j=1}^{\#neig} \left\{ f_{j1}^* \sigma \Delta z_{j1} + g_{j1}^* \Delta z_{j1} \right\} = 0. \tag{3.38}$$

Finally, by substituting (3.21) and (3.22) into Eqs. (3.36), (3.37) and (3.38) and rearranging the terms, one arrives at the system of equations $As = B$, where the matrix of coefficients A is given by

$$A = \left[\begin{array}{ccc} \sum\left(\sigma^2+1\right)\Delta x_{j1}^2 & \sum\left(\sigma^2+1\right)\Delta x_{j1}\Delta y_{j1} & \sum\left(\sigma^2+1\right)\Delta x_{j1}\Delta z_{j1} \\ \sum\left(\sigma^2+1\right)\Delta y_{j1}\Delta x_{j1} & \sum\left(\sigma^2+1\right)\Delta y_{j1}^2 & \sum\left(\sigma^2+1\right)\Delta y_{j1}\Delta z_{j1} \\ \sum\left(\sigma^2+1\right)\Delta z_{j1}\Delta x_{j1} & \sum\left(\sigma^2+1\right)\Delta z_{j1}\Delta y_{j1} & \sum\left(\sigma^2+1\right)\Delta z_{j1}^2 \end{array} \right], \tag{3.39}$$

the solution vector s, with the values of the center cell derivatives, is given by

$$s = \left[\begin{array}{ccc} \left.\frac{\partial\phi}{\partial x}\right|_o & \left.\frac{\partial\phi}{\partial y}\right|_o & \left.\frac{\partial\phi}{\partial z}\right|_o \end{array} \right]^T, \tag{3.40}$$

the RHS vector B is given by

$$B = \left\{ \begin{array}{c} \sum\left(\sigma^2\Delta x_{j1}\left(\tilde{F}_{j1}-\phi_o\right)+\Delta x_{j1}\tilde{G}_{j1}\right) \\ \sum\left(\sigma^2\Delta y_{j1}\left(\tilde{F}_{j1}-\phi_o\right)+\Delta y_{j1}\tilde{G}_{j1}\right) \\ \sum\left(\sigma^2\Delta z_{j1}\left(\tilde{F}_{j1}-\phi_o\right)+\Delta z_{j1}\tilde{G}_{j1}\right) \end{array} \right\}, \tag{3.41}$$

and the simplification $\sum = \sum_{j=1}^{\#neig}$ is used. The summation here is performed over each of the immediate face neighbors. The values of $\tilde{F}_{j1}$ and $\tilde{G}_{j1}$ contain the center cell and neighbor's scalar value and derivatives, and so the set of Eqs. given by (3.39), (3.40) and (3.41) for all cells yields a 3N×3N linear system of equations (where N here is the total number of control volumes) that can be solved for $\frac{\partial\phi}{\partial x}$, $\frac{\partial\phi}{\partial y}$ and $\frac{\partial\phi}{\partial z}$ in every control volume. The method is therefore implicit and compact.

3.3.3 Third Order Scheme

A third order finite-volume scheme enforces that all polynomial expansions of the dependent variable are truncated at the second order derivatives. To retain third order accuracy in reconstruction of the face fluxes, three quadrature points (for tetrahedral meshes in 3D) must be used to perform the integration over each face. Measures of disagreement are therefore defined at each face quadrature point, denoted for each face by the index m (=1, 2, 3). As above, j indicates face indexing. The relevant projection vectors are therefore defined by:

$$\vec{r}_{jm} = \vec{X}_{jm} - \vec{X}_o = \Delta x_{jm}\vec{i}_x + \Delta y_{jm}\vec{i}_y + \Delta z_{jm}\vec{i}_z, \tag{3.42}$$
$$\widetilde{r}_{jm} = \vec{X}_{jm} - \vec{X}_{nj} = \widetilde{\Delta x}_{jm}\vec{i}_x + \widetilde{\Delta y}_{jm}\vec{i}_y + \widetilde{\Delta z}_{jm}\vec{i}_z.$$

In order to calculate the first and second derivatives needed for the third order scheme, three measures of disagreement between neighboring cell expansions are postulated for each face integration point:

1) Measure of disagreement between the scalar value projections from each cell at the face quadrature point:

$$(\Delta\phi)_{jm} = \phi_o^*\big|_{jm}^{(new)} - \frac{1}{2}\left(\phi_o^*\big|_{jm} + \phi_{nj}^*\big|_{jm}\right)^{(current)}, \tag{3.43}$$

2) Measure of disagreement between the first derivative projections from each cell at the face quadrature point:

$$\Delta\left(\nabla\phi\right)_{jm} \cdot \vec{r}_{jm} = \left(\nabla\phi_o^*\big|_{jm}^{(new)} - \frac{1}{2}\left(\nabla\phi_o^*\big|_{jm} + \nabla\phi_{nj}^*\big|_{jm}\right)^{(current)}\right) \cdot \vec{r}_{jm}, \tag{3.44}$$

3) Measure of disagreement between the second derivative projections from each cell at the face quadrature point:

$$\left(\Delta\left(\nabla\nabla\phi\right)_{jm} \cdot \vec{r}_{jm}\right) \cdot \vec{r}_{jm} = \left(\left(\nabla\nabla\phi_o^*\big|_{jm}^{(new)} - \frac{1}{2}\left(\nabla\nabla\phi_o^*\big|_{jm} + \nabla\nabla\phi_{nj}^*\big|_{jm}\right)\right)^{(curr)}\right.$$

$$\left. \cdot \vec{r}_{jm} \right) \cdot \vec{r}_{jm}. \tag{3.45}$$

Those three measures are now analyzed separately.

3.3.3.1 First Measure of Disagreement

Denoting the second term on the RHS of Eq. (3.43) as $\tilde{F}_{jm}$, a third order accurate expansion (i.e., by truncating the Taylor series expansions at the second derivatives) of both terms using the connecting vectors between the center and neighbor cell centroids to the integration (quadrature) points at their connecting face may be defined, and after redefining the first measure of disagreement as f_{jm}, Eq. (3.43) it can be rewritten as

$$\begin{aligned} f_{jm} = \phi_o + \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_{jm} + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_{jm} + \left.\frac{\partial \phi}{\partial z}\right|_o \Delta z_{jm} + \\ \frac{1}{2}\left(\left.\frac{\partial^2 \phi}{\partial x^2}\right|_o \Delta x_{jm}^2 + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_o \Delta y_{jm}^2 + \left.\frac{\partial^2 \phi}{\partial z^2}\right|_o \Delta z_{jm}^2 \right) + \\ \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_o \Delta x_{jm} \Delta y_{jm} + \left.\frac{\partial^2 \phi}{\partial x \partial z}\right|_o \Delta x_{jm} \Delta z_{jm} + \left.\frac{\partial^2 \phi}{\partial y \partial z}\right|_o \Delta y_{jm} \Delta z_{jm} - \tilde{F}_{jm}, \end{aligned} \tag{3.46}$$

where $\tilde{F}_{jm}$ is given by

$$\begin{aligned} \tilde{F}_{jm} = \frac{1}{2}\Bigg[\phi_o + \left.\frac{\partial \phi}{\partial x}\right|_o \Delta x_{jm} + \left.\frac{\partial \phi}{\partial y}\right|_o \Delta y_{jm} + \left.\frac{\partial \phi}{\partial z}\right|_o \Delta z_{jm} + \\ \frac{1}{2}\left(\left.\frac{\partial^2 \phi}{\partial x^2}\right|_o \Delta x_{jm}^2 + \left.\frac{\partial^2 \phi}{\partial y^2}\right|_o \Delta y_{jm}^2 + \left.\frac{\partial^2 \phi}{\partial z^2}\right|_o \Delta z_{jm}^2 \right) + \\ \left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_o \Delta x_{jm} \Delta y_{jm} + \left.\frac{\partial^2 \phi}{\partial x \partial z}\right|_o \Delta x_{jm} \Delta z_{jm} + \left.\frac{\partial^2 \phi}{\partial y \partial z}\right|_o \Delta y_{jm} \Delta z_{jm} + \\ \phi_{nj} + \left.\frac{\partial \phi}{\partial x}\right|_{nj} \widetilde{\Delta x}_{jm} + \left.\frac{\partial \phi}{\partial y}\right|_{nj} \widetilde{\Delta y}_{jm} + \left.\frac{\partial \phi}{\partial z}\right|_{nj} \widetilde{\Delta z}_{jm} + \end{aligned} \tag{3.47}$$

$$\frac{1}{2}\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_{nj}\widetilde{\Delta x}_{jm}^{\,2}+\left.\frac{\partial^2\phi}{\partial y^2}\right|_{nj}\widetilde{\Delta y}_{jm}^{\,2}+\left.\frac{\partial^2\phi}{\partial z^2}\right|_{nj}\widetilde{\Delta z}_{jm}^{\,2}\right)+$$

$$\left.\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{nj}\widetilde{\Delta x}_{jm}\widetilde{\Delta y}_{jm}+\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{nj}\widetilde{\Delta x}_{jm}\widetilde{\Delta z}_{jm}+\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{nj}\widetilde{\Delta y}_{jm}\widetilde{\Delta z}_{jm}\right]^{(curr)}.$$

3.3.3.2 Second Measure of Disagreement

Based on first-order Taylor Series expansions for the gradients and redefining (3.44) as g_{jm}, one obtains

$$\begin{aligned}g_{jm}=&\left\{\begin{array}{l}\left.\frac{\partial\phi}{\partial x}\right|_{o}+\left.\frac{\partial^2\phi}{\partial x^2}\right|_{o}\Delta x_{jm}+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{o}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{o}\Delta z_{jm}\\ \left.\frac{\partial\phi}{\partial y}\right|_{o}+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{o}\Delta x_{jm}+\left.\frac{\partial^2\phi}{\partial y^2}\right|_{o}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{o}\Delta z_{jm}\\ \left.\frac{\partial\phi}{\partial z}\right|_{o}+\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{o}\Delta x_{jm}+\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{o}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial z^2}\right|_{o}\Delta z_{jm}\end{array}\right\}\cdot\left\{\begin{array}{c}\Delta x_{jm}\\ \Delta y_{jm}\\ \Delta z_{jm}\end{array}\right\}\\ &-\tilde{G}_{jm},\end{aligned}\tag{3.48}$$

where $\tilde{G}_{jm}$ is given by

$$\begin{aligned}\tilde{G}_{jm}=\frac{1}{2}&\left[\left\{\begin{array}{l}\left.\frac{\partial\phi}{\partial x}\right|_{o}+\left.\frac{\partial^2\phi}{\partial x^2}\right|_{o}\Delta x_{jm}+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{o}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{o}\Delta z_{jm}\\ \left.\frac{\partial\phi}{\partial y}\right|_{o}+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{o}\Delta x_{jm}+\left.\frac{\partial^2\phi}{\partial y^2}\right|_{o}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{o}\Delta z_{jm}\\ \left.\frac{\partial\phi}{\partial z}\right|_{o}+\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{o}\Delta x_{jm}+\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{o}\Delta y_{jm}+\left.\frac{\partial^2\phi}{\partial z^2}\right|_{o}\Delta z_{jm}\end{array}\right\}\cdot\left\{\begin{array}{c}\Delta x_{jm}\\ \Delta y_{jm}\\ \Delta z_{jm}\end{array}\right\}+\right.\\ &\left.\left\{\begin{array}{l}\left.\frac{\partial\phi}{\partial x}\right|_{nj}+\left.\frac{\partial^2\phi}{\partial x^2}\right|_{nj}\widetilde{\Delta x}_{jm}+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{nj}\widetilde{\Delta y}_{jm}+\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{nj}\widetilde{\Delta z}_{jm}\\ \left.\frac{\partial\phi}{\partial y}\right|_{nj}+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{nj}\widetilde{\Delta x}_{jm}+\left.\frac{\partial^2\phi}{\partial y^2}\right|_{nj}\widetilde{\Delta y}_{jm}+\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{nj}\widetilde{\Delta z}_{jm}\\ \left.\frac{\partial\phi}{\partial z}\right|_{nj}+\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{nj}\widetilde{\Delta x}_{jm}+\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{nj}\widetilde{\Delta y}_{jm}+\left.\frac{\partial^2\phi}{\partial z^2}\right|_{nj}\widetilde{\Delta z}_{jm}\end{array}\right\}\cdot\left\{\begin{array}{c}\Delta x_{jm}\\ \Delta y_{jm}\\ \Delta z_{jm}\end{array}\right\}\right]^{(curr)}.\end{aligned}\tag{3.49}$$

3.3.3.3 Third Measure of Disagreement

Based on the zero-order expansions for the gradients of the gradients and redefining (3.45) as h_{jm}, one obtains

$$\begin{aligned} h_{jm} = &\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o \Delta x_{jm}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_o \Delta y_{jm}^2 + \left.\frac{\partial^2\phi}{\partial z^2}\right|_o \Delta z_{jm}^2 + \right. \\ &\left. 2\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o \Delta x_{jm}\Delta y_{jm} + 2\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_o \Delta x_{jm}\Delta z_{jm} + 2\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_o \Delta y_{jm}\Delta z_{jm} \right) \\ &- \tilde{H}_{jm}, \end{aligned} \tag{3.50}$$

where $\tilde{H}_{jm}$, whose expansions must be truncated at the second derivatives, is given by

$$\begin{aligned} \tilde{H}_{jm} = \frac{1}{2} &\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o \Delta x_{jm}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_o \Delta y_{jm}^2 + \left.\frac{\partial^2\phi}{\partial z^2}\right|_o \Delta z_{jm}^2 + \right. \\ & 2\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o \Delta x_{jm}\Delta y_{jm} + 2\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_o \Delta x_{jm}\Delta z_{jm} + 2\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_o \Delta y_{jm}\Delta z_{jm} + \\ & \left.\frac{\partial^2\phi}{\partial x^2}\right|_{nj} \Delta x_{jm}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_{nj} \Delta y_{jm}^2 + \left.\frac{\partial^2\phi}{\partial z^2}\right|_{nj} \Delta z_{jm}^2 + \\ & 2\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{nj} \Delta x_{jm}\Delta y_{jm} + 2\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{nj} \Delta x_{jm}\Delta z_{jm} + \\ & \left. 2\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{nj} \Delta y_{jm}\Delta z_{jm} \right)^{(curr)} . \end{aligned} \tag{3.51}$$

3.3.4 Optimization Procedure for the Third Order Scheme

As before for the second-order version, the three measures of disagreement can be rescaled to control the accuracy and stability of the method through the use of the tuning parameter, σ. To do so, the three measures of mismatch: f_{jm}, g_{jm} and h_{jm} , given by (3.47), (3.49) and (3.51), respectively, are rewritten as

$$
\begin{aligned}
f^*_{jm} = \sigma^2 f_{jm} = \sigma^2 \Bigg[& \phi_o + \left.\frac{\partial\phi}{\partial x}\right|_o \Delta x_{jm} + \left.\frac{\partial\phi}{\partial y}\right|_o \Delta y_{jm} + \left.\frac{\partial\phi}{\partial z}\right|_o \Delta z_{jm} + \\
& \frac{1}{2}\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_o \Delta x^2_{jm} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_o \Delta y^2_{jm} + \left.\frac{\partial^2\phi}{\partial z^2}\right|_o \Delta z^2_{jm} \right) + \\
& \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o \Delta x_{jm}\Delta y_{jm} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_o \Delta x_{jm}\Delta z_{jm} + \\
& \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_o \Delta y_{jm}\Delta z_{jm} - \tilde{F}_{jm} \Bigg], \qquad (3.52)
\end{aligned}
$$

$$
\begin{aligned}
g^*_{jm} = \sigma g_{jm} = \sigma \Bigg[& \left(\left.\frac{\partial\phi}{\partial x}\right|_o + \left.\frac{\partial^2\phi}{\partial x^2}\right|_o \Delta x_{jm} + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o \Delta y_{jm} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_o \Delta z_{jm} \right) \Delta x_{jm} + \\
& \left(\left.\frac{\partial\phi}{\partial y}\right|_o + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o \Delta x_{jm} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_o \Delta y_{jm} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_o \Delta z_{jm} \right) \Delta y_{jm} + \\
& \left(\left.\frac{\partial\phi}{\partial z}\right|_o + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_o \Delta x_{jm} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_o \Delta y_{jm} + \left.\frac{\partial^2\phi}{\partial z^2}\right|_o \Delta z_{jm} \right) \Delta z_{jm} \\
& - \tilde{G}_{jm} \Bigg], \qquad (3.53)
\end{aligned}
$$

$$
\begin{aligned}
h^*_{jm} = h_{jm} = \Bigg(& \left.\frac{\partial^2\phi}{\partial x^2}\right|_o \Delta x^2_{jm} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_o \Delta y^2_{jm} + \left.\frac{\partial^2\phi}{\partial z^2}\right|_o \Delta z^2_{jm} + \\
& 2\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o \Delta x_{jm}\Delta y_{jm} + 2\left.\frac{\partial^2\phi}{\partial x\partial z}\right|_o \Delta x_{jm}\Delta z_{jm} + \\
& 2\left.\frac{\partial^2\phi}{\partial y\partial z}\right|_o \Delta y_{jm}\Delta z_{jm} \Bigg) - \tilde{H}_{jm}, \qquad (3.54)
\end{aligned}
$$

where again $\tilde{F}_{jm}$, $\tilde{G}_{jm}$ and $\tilde{H}_{jm}$ are given by Eqs. (3.47), (3.49) and (3.51), respectively.

By using (3.52), (3.53) and (3.54), it is possible to construct the following quadratic functional

$$
F = \sum_{j=1}^{\#neig} \sum_{m=1}^{3} \left({f^*_{jm}}^2 + {g^*_{jm}}^2 + {h^*_{jm}}^2 \right), \qquad (3.55)
$$

which must be minimized in order to find the values of the first and second order derivatives at any given cell centroid. As above, minimization of the function F is straightforward, producing the following five equations (one for each derivative component sought):

$$\begin{aligned}
&\frac{\partial F}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}=0, \frac{\partial F}{\partial\left(\left.\frac{\partial \phi}{\partial y}\right|_o\right)}=0, \frac{\partial F}{\partial\left(\left.\frac{\partial \phi}{\partial z}\right|_o\right)}=0,\\
&\frac{\partial F}{\partial\left(\left.\frac{\partial^2 \phi}{\partial x^2}\right|_o\right)}=0, \frac{\partial F}{\partial\left(\left.\frac{\partial^2 \phi}{\partial y^2}\right|_o\right)}=0, \frac{\partial F}{\partial\left(\left.\frac{\partial^2 \phi}{\partial z^2}\right|_o\right)}=0,\\
&\frac{\partial F}{\partial\left(\left.\frac{\partial^2 \phi}{\partial x \partial y}\right|_o\right)}=0, \frac{\partial F}{\partial\left(\left.\frac{\partial^2 \phi}{\partial x \partial z}\right|_o\right)}=0, \frac{\partial F}{\partial\left(\left.\frac{\partial^2 \phi}{\partial y \partial z}\right|_o\right)}=0.
\end{aligned} \tag{3.56}$$

The first equation in (3.56) can be expressed as

$$\begin{aligned}
\frac{\partial F}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}=2\sum_{j=1}^{\#neig}\sum_{m=1}^{3}\left\{f^*_{jm}\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}+g^*_{jm}\frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}+\right.\\
\left. h^*_{jm}\frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}\right\}=0,
\end{aligned} \tag{3.57}$$

and the other equations are expressed in a similar manner. The internal partial derivatives are given by:

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}=\sigma^2\Delta x_{jm}, \frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}=\sigma\Delta x_{jm}, \frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial x}\right|_o\right)}=0,$$

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial y}\right|_o\right)}=\sigma^2\Delta y_{jm}, \frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial y}\right|_o\right)}=\sigma\Delta y_{jm}, \frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial y}\right|_o\right)}=0,$$

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial z}\right|_o\right)}=\sigma^2\Delta z_{jm}, \frac{\partial g^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial z}\right|_o\right)}=\sigma\Delta z_{jm}, \frac{\partial h^*_{jm}}{\partial\left(\left.\frac{\partial \phi}{\partial z}\right|_o\right)}=0,$$

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x^2}\right|_o\right)}=\frac{\sigma^2\Delta x^2_{jm}}{2},\frac{\partial g^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x^2}\right|_o\right)}=\sigma\Delta x^2_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x^2}\right|_o\right)}=\Delta x^2_{jm}, \tag{3.58}$$

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial y^2}\right|_o\right)}=\frac{\sigma^2\Delta y^2_{jm}}{2},\frac{\partial g^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial y^2}\right|_o\right)}=\sigma\Delta y^2_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial y^2}\right|_o\right)}=\Delta y^2_{jm},$$

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial z^2}\right|_o\right)}=\frac{\sigma^2\Delta z^2_{jm}}{2},\frac{\partial g^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial z^2}\right|_o\right)}=\sigma\Delta z^2_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial z^2}\right|_o\right)}=\Delta z^2_{jm},$$

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x\partial y}\right|_o\right)}=\sigma^2\Delta x_{jm}\Delta y_{jm},\frac{\partial g^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x\partial y}\right|_o\right)}=2\sigma\Delta x_{jm}\Delta y_{jm},$$

$$\frac{\partial h^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x\partial y}\right|_o\right)}=2\Delta x_{jm}\Delta y_{jm},\frac{\partial f^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x\partial z}\right|_o\right)}=\sigma^2\Delta x_{jm}\Delta z_{jm},$$

$$\frac{\partial g^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x\partial z}\right|_o\right)}=2\sigma\Delta x_{jm}\Delta z_{jm},\frac{\partial h^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial x\partial z}\right|_o\right)}=2\Delta x_{jm}\Delta z_{jm},$$

$$\frac{\partial f^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial y\partial z}\right|_o\right)}=\sigma^2\Delta y_{jm}\Delta z_{jm},\frac{\partial g^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial y\partial z}\right|_o\right)}=2\sigma\Delta y_{jm}\Delta z_{jm},$$

$$\frac{\partial h^*_{jm}}{\partial\left(\left.\dfrac{\partial^2\phi}{\partial y\partial z}\right|_o\right)}=2\Delta y_{jm}\Delta z_{jm}.$$

Substituting Eqs. (3.58) into the Eqs. (3.56), dividing all equations by 2, and rearranging terms, the system of equations (3.56) can be rewritten in matrix form as $As = B$, where the components of the matrix of coefficients A and the RHS vector B are given in Appendix B.1.

It is worthwhile to reinforce at this point that the cell neighbors over which the summation is performed in the OGRE third order scheme represent just the first level of face neighbors (e.g., 4 face neighbors for an internal control volume in a tetrahedra-based 3D unstructured mesh). This is in significant contrast to a L-S third order scheme, which would (mathematically) require the summation over a larger stencil made up of at least 9 neighbor cells (in 3D), typically constructed of both first and second level face neighbors. This smaller stencil translates into memory savings in either sequential or parallel calculations. For example, in parallel calculations using a 3rd order scheme, instead of each core having to store two levels of neighbors for each subdomain, OGRE would require just one level. Also, by not using the second level of face neighbors, OGRE avoids (by not increasing the stencil size) additional stiffness in the matrix calculation, a known problematic by-product of the L-S scheme. Higher-order implementations of the OGRE method would similarly require only the immediate face neighbors of each cell to populate the computational stencil.

The global system of equations obtained using the OGRE method can be solved using an iterative procedure. The local system in each cell (e.g. 9x9 for a third order 3D implementation) can be solved using any appropriate technique, for example simple Gaussian Elimination. The derivative values can be sequentially calculated and updated for each cell, thus producing a Gauss-Seidel-type of iterative procedure. For a fixed variable field, the gradient values of the transport scalar will converge toward the values which minimize the objective function. When the OGRE scheme is used within numerical schemes which are themselves iterative in nature, for example implicit finite volume simulations, the gradient field may be updated once per outer (Newton) iteration. Therefore, within each time step (or during a steady-state simulation), the variable field and the gradient field converge simultaneously towards the solution.

For the scalar transport example cases in this study, since the focus is on a steady-state calculation, just one inner-iteration (Newton sub-iteration) is used to update the derivative values, because there is no need to converge the derivatives for a non-converged scalar flow field during a steady-state calculation.

3.4 Boundary Condition Implementation

The previous section describes in detail the procedure to be applied for computing derivatives in interior cells, i.e. cells for which each bounding face is in the interior of the domain. A modification is necessary for boundary cells (i.e. cells for which at least one bounding face lies on the domain boundary). The boundary condition method proposed here first computes values of ϕ and its derivatives at the boundary face centroids. Boundary cells then use a procedure similar to that outlined above, but with matching conditions defined based on projections from the face centroids (instead of from a neighbor centroid).

Similar to what is done for the internal cell calculations, measures of disagreement are defined at each of the face integration points, and the gradients and/or variable values are determined based on a minimization of those measures. The boundary conditions themselves are imposed as constraints in the optimization problem, and a procedure based on

Lagrange multipliers is adopted. Two general types of boundary conditions are addressed: given boundary value (Dirichlet) and given normal gradient at the boundary (Neumann). The former yields a constraint on the value of ϕ at the boundary face, and the latter yields a constraint on the value of $\nabla\phi \cdot \hat{n}$, where $\hat{n}$ is the unit normal vector at the boundary face. The constraints are always applied at the boundary face quadrature (integration) points.

The following analysis is presented based on a tetrahedral-based 3D unstructured mesh, although the methodology can be adopted in a straightforward manner to arbitrary mesh types. Figure 3.2 illustrates in general how the BC procedure is implemented. It shows one (center) cell (denoted cell 0) containing a bounding face (shown shadowed in the figure) which lies on the domain boundary, and two cells (denoted 1 and 2) of the three first-face neighbor cells for this center cell. For the 2nd order scheme, the face centroid denoted by b is the only face integration point, while for the 3rd order scheme, three integration points are needed and denoted as b_m ($m = 1, 2, 3$), which locations are described in [25]. The values of ϕ and its derivatives must be calculated at the boundary location b (face centroid). Using a similar analysis as before (for internal cells), for a 2nd order scheme in 3D, 4 variables (ϕ and its three derivatives) are sought at b, which means that just the information provided by the measures of mismatch derived from the two expansions (ϕ and its gradient) at the center cell 0 plus the BC constraint at the quadrature point (face centroid) are not enough to create a numerically consistent problem, so other cells must be added to the problem (analogous idea works for 3rd order scheme). This extra information comes from expansions (measures of mismatch) from the three first-face-neighbors of the center cell. In Figure 3.2 (that shows the 3rd order variant of the scheme), two (cells 1 and 2) of those three first-face neighbors are shown. For clarity, cell 3 is not shown in the figure. This general procedure is explained in detail in the next two sections (for 2nd and 3rd order schemes).

Two modifications in this procedure occur if a cell has two or three faces lying on domain boundaries. For a cell containing two faces lying on domain boundaries, using a 2nd order scheme, the 4 variables to be sought at face centroid b can be found with the information of the (six) expansions coming from the center cell and its two first-neighbor cells plus the information on the BC constraint, while by using a 3rd order scheme, the 10 variables to be sought at face centroid b can be found with the information of the (nine) expansions coming from the center cell and its two first-neighbor cells plus the (three) BC contraints. For a cell containing three faces lying on domain boundaries, using a 2nd order scheme, the 4 variables to be sought at face centroid b can be found with the information of the (four) expansions coming from the center cell and its first-neighbor cell plus the information on the BC constraint. But a new layer of neighbors must be added in the calculation for the case of a cell containing three faces lying on domain boundaries and the 3rd order scheme is being used. In this special case, 10 variables are sought at face centroid b, but by adding the information of the (three) expansions coming from the center cell, the (three) expansions coming from its only first-neighbor cell, plus the (three) BC contraints, all add up to just 9 data points, which enforces the need of adding in the calculation an extra layer of neighbors (three second-face-neighbors) for the center cell, thus guaranteeing that the system of equations is not under-determined.

3.4.1 BC Implementation for the Second Order Scheme

For the 2nd order scheme the values of ϕ and its gradient are expanded (in general) from the four closest cells (center cell and its three first-face-neighbors) to the face centroid b (which is the only face integration location for the second order scheme) where two measures of disagreement per cell are formed. So, 8 terms (in general) are created (four for the ϕ measure and four for their gradient measure). The measures of disagreement for ϕ and its gradient for the cell k (= 0, 1, 2, 3) are given by:

$$R_{1k} = \sigma \left(\phi_b^{(new)} - \frac{1}{2} \left(\phi_b + \phi_k^*\big|_b \right)^{(current)} \right), \tag{3.59}$$

and

$$R_{2k} = \left(\nabla\phi_b^{(new)} - \frac{1}{2} \left(\nabla\phi_b + \nabla\phi_k^*\big|_b \right)^{(current)} \right) \cdot \vec{r}_{k1}, \tag{3.60}$$

where

$$\vec{r}_{k1} = \vec{X}_b - \vec{X}_k = \Delta x_{k1}\vec{i}_x + \Delta y_{k1}\vec{i}_y + \Delta z_{k1}\vec{i}_z \tag{3.61}$$

(notice here again the inclusion of the tuning parameter σ).

If following the same approach as for the internal cells calculation, the vector used for the dot product in (3.60) should be the connecting vector between the position where the new values are being sought and the face integration points, but since, for 2nd order scheme, the face centroid and the face integration point are the same, it was decided instead to use the connecting vector between the center cell centroid k and the face integration point. The same approach is used for the 3rd order case as well to maintain consistency. Another reason for doing so is that, if the vector chosen for the dot product was the connecting vector between the face centroid and the face integration points (for 3rd order scheme now), the scheme does not converge. The likely cause is that, since the linear combination of those vectors would produce just a plane (always parallel to the boundary face), this would create a geometrical inconsistency in the boundary face calculation, since the measures involving the dot product between the gradient of ϕ and those vectors would discard any component normal to this plane.

The disagreement measures in Eq. (3.59) and (3.60) are squared to create a quadratic functional, which has the form

$$F\left(x^*\right) = \sum_{k=0}^{3} \left(R_{1k}^2 + R_{2k}^2\right), \tag{3.62}$$

where x^* is the solution vector containing the values of ϕ and its three first derivatives at the boundary face centroid b. The additional constraint for a Dirichlet BC is given by:

$$q_b : \phi_b - \phi_{BC} = 0, \tag{3.63}$$

and the additional constraint for a Neumann BC is given by:

$$q_b : \left(\nabla\phi\right)_b \cdot \hat{n} - |\nabla\phi|_{BC} = 0, \tag{3.64}$$

where the vector $\hat{n}$ represents the unit normal vector for the boundary face. The boundary condition values ϕ_{BC} or $|\nabla\phi|_{BC}$ may be constant or vary as a function of position on the boundary. For a zero-flux condition, the Neumann condition is expressed as

$$q_b : \left(\nabla\phi \cdot \hat{n}\right)_b = 0. \tag{3.65}$$

This boundary condition may be applied, for example, at a symmetry plane in finite-volume simulations.

The Lagrangian function is defined in terms of the objective function and the constraint as follows:

$$L\left(x^{*},\lambda\right)=F\left(x^{*}\right)+\lambda q_{b}. \tag{3.66}$$

To minimize the Lagrangian it is necessary to solve the following system of equations:

$$\frac{\partial L}{\partial x_{j}^{*}}=0,\ \ j=1,4, \tag{3.67}$$

$$\frac{\partial L}{\partial \lambda}=0. \tag{3.68}$$

Applying Taylor Series expansions of appropriate order for (3.59) and (3.60), they can be substituted into (3.66) so the system of equations defined by (3.67) – (3.68) can be obtained. After rearranging the terms, one arrives at the system of equations $As=C$, where the matrix of coefficients A for the Dirichlet type of BC is given by

$$A=\begin{bmatrix} \sum\left(2\sigma^{2}\right) & 0 & 0 & 0 & 1 \\ 0 & \sum\left(2\Delta x_{k1}^{2}\right) & \sum\left(2\Delta x_{k1}\Delta y_{k1}\right) & \sum\left(2\Delta x_{k1}\Delta z_{k1}\right) & 0 \\ 0 & \sum\left(2\Delta y_{k1}\Delta x_{k1}\right) & \sum\left(2\Delta y_{k1}^{2}\right) & \sum\left(2\Delta y_{k1}\Delta z_{k1}\right) & 0 \\ 0 & \sum\left(2\Delta z_{k1}\Delta x_{k1}\right) & \sum\left(2\Delta z_{k1}\Delta y_{k1}\right) & \sum\left(2\Delta z_{k1}^{2}\right) & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.69}$$

the expanded solution vector s is now given by

$$s=\begin{bmatrix} \phi_{b} & \left.\frac{\partial \phi}{\partial x}\right|_{b} & \left.\frac{\partial \phi}{\partial y}\right|_{b} & \left.\frac{\partial \phi}{\partial z}\right|_{b} & \lambda \end{bmatrix}^{T}, \tag{3.70}$$

and the RHS vector C is given by

$$C = \left\{ \begin{array}{c} \sum 2\sigma^2 \overline{\phi}_{bk} \\ \sum 2\Delta x_{k1} \overline{\nabla\phi}_{bk} \cdot \vec{r}_{k1} \\ \sum 2\Delta y_{k1} \overline{\nabla\phi}_{bk} \cdot \vec{r}_{k1} \\ \sum 2\Delta z_{k1} \overline{\nabla\phi}_{bk} \cdot \vec{r}_{k1} \\ \phi_{BC} \end{array} \right\} . \tag{3.71}$$

The averaged values on the RHS are given by

$$\overline{\phi}_{bk} = \frac{1}{2} \left(\phi_b + \phi_k + \left.\frac{\partial \phi}{\partial x}\right|_k \Delta x_{k1} + \left.\frac{\partial \phi}{\partial y}\right|_k \Delta y_{k1} + \left.\frac{\partial \phi}{\partial z}\right|_k \Delta z_{k1} \right) \tag{3.72}$$

and

$$\begin{aligned} \overline{\nabla\phi}_{bk} \cdot \vec{r}_{k1} = \frac{1}{2} \Bigg(& \left.\frac{\partial \phi}{\partial x}\right|_b \Delta x_{k1} + \left.\frac{\partial \phi}{\partial y}\right|_b \Delta y_{k1} + \left.\frac{\partial \phi}{\partial z}\right|_b \Delta z_{k1} + \\ & \left.\frac{\partial \phi}{\partial x}\right|_k \Delta x_{k1} + \left.\frac{\partial \phi}{\partial y}\right|_k \Delta y_{k1} + \left.\frac{\partial \phi}{\partial z}\right|_k \Delta z_{k1} \Bigg) \end{aligned} \tag{3.73}$$

and the simplification $\sum = \sum_{k=0}^{1,\ 2\ or\ 3}$ is used.

The following substitution on the fifth row/column of A (since A is symmetric) produces the matrix of coefficients for the Neumann type of BC

$$A(5, 1:4) = A(1:4, 5)^T = \begin{bmatrix} 0 & n_x & n_y & n_z \end{bmatrix}, \tag{3.74}$$

where n_x, n_y and n_z are the components of the normal unit vector for the boundary face. Also, the fifth component in C must be changed such that $C(5) = |\nabla\phi|_{BC}$.

3.4.2 BC Implementation for the Third Order Scheme

For the 3rd order scheme the values of ϕ, the gradient of ϕ and the gradient of the gradient of ϕ are expanded from the four closest cells k (= 0, 1, 2, 3) and from the boundary face centroid to each one of the three face integration points, thus forming three measures of disagreement. In this way, 12 terms are created for each one of the three face integration points. The three measures of disagreement for a given cell k on the boundary face integration point b_m $(m = 1,\ 2,\ 3)$ are given by:

$$R_{1km} = \sigma^2 \left(\phi_b^*\big|_{b_m}^{(new)} - \frac{1}{2}\left(\phi_b^*\big|_{b_m} + \phi_k^*\big|_{b_m} \right)^{(current)} \right), \tag{3.75}$$

$$R_{2km} = \sigma \left(\nabla\phi_b^*\big|_{b_m}^{(new)} - \frac{1}{2}\left(\nabla\phi_b^*\big|_{b_m} + \nabla\phi_k^*\big|_{b_m} \right)^{(current)} \right) \cdot \vec{r}_{km}, \tag{3.76}$$

$$R_{3km} = \left(\left(\nabla\nabla\phi_b^*\big|_{b_m}^{(new)} - \frac{1}{2}\left(\nabla\nabla\phi_b^*\big|_{b_m} + \nabla\nabla\phi_k^*\big|_{b_m} \right)^{(current)} \right) \cdot \vec{r}_{km} \right) \cdot \vec{r}_{km}, \tag{3.77}$$

where the index m here represents boundary face integration point location and

$$\vec{r}_{km} = \vec{X}_{b_m} - \vec{X}_k = \Delta x_{km}\vec{i}_x + \Delta y_{km}\vec{i}_y + \Delta z_{km}\vec{i}_z \tag{3.78}$$

represents the connecting vector between the cell centroid k and the boundary face integration point b_m.

Those 36 terms (when 4 cells are used) are squared to create a quadratic functional, which has the form

$$F\left(x^*\right) = \sum_{m=1}^{3} \sum_{k=0}^{1,2,3\ or\ 4} \left(R_{1km}^2 + R_{2km}^2 + R_{3km}^2 \right), \tag{3.79}$$

where x^* is the solution vector containing the values of ϕ and its first and second derivatives at the boundary face centroid. The additional constraints for a Dirichlet BC are given by:

$$q_m : \phi_b^*\big|_{b_m} - \phi_{bm} = 0, \tag{3.80}$$

and the additional constraints for a Neumann BC are given by:

$$q_m : \left(\nabla\phi \cdot \hat{n}\right)_b^*\big|_{b_m} - \left|\nabla\phi\right|_{bm} = 0, \tag{3.81}$$

Notice that the constraints are always applied at the boundary face integration points. In the previous section, it was applied at the boundary face centroid only because that is the position for the boundary face integration point for a 2nd order scheme.

The Lagrangian function is defined as follows

$$L\left(x^{*},\lambda\right)=F\left(x^{*}\right)+\lambda_{1}q_{1}+\lambda_{2}q_{2}+\lambda_{3}q_{3}. \tag{3.82}$$

To minimize the Lagrangian it is necessary to solve the following system of equations:

$$\frac{\partial L}{\partial x_{j}^{*}}=0,\ \ j=1,10, \tag{3.83}$$

$$\frac{\partial L}{\partial \lambda_{l}}=0,\ \ l=1,3. \tag{3.84}$$

Using third order expansions on (3.75), (3.76) and (3.77), they can be substituted into (3.82) so the system of equations (3.83) – (3.84) can be calculated. After rearranging the terms, one arrives at the system of equations $As=C$, whose components, the matrix of coefficients A and the RHS vector C, for the Dirichlet and Neumann types of BC, are given in the Appendix B.2.

3.5 Results

The OGRE algorithm was implemented as a User Defined Function (UDF) into the commercial flow solver Ansys Fluent v12.0. Three-dimensional versions of both the 2nd and 3rd order schemes were implemented. Results were obtained for several test cases and compared to 2nd and 3rd order other methods (mainly the least-squares method). Test cases were performed with two objectives. The first objective was to check if the OGRE scheme is able to deliver a numerical solution with measured accuracy comparable with the nominal order of the defined scheme for two different cases: a static variable field test case and a finite-volume scalar advection case where a variable flow field is smoothly distributed throughout the domain. Both quantitative and qualitative analyses were performed. The second objective was to determine how effectively the OGRE scheme is able to reconstruct a variable field with a wide spectrum of wavelengths. The result analyzes the spectral behavior of the scheme, which could indicate its potential ability to address problems that require low numerical dissipation and/or energy preservation.

3.5.1 Static Variable Field Test Case

The first test case is straightforward. Given a distribution of a (fixed) arbitrary variable, reconstruct the gradients of the variable at each point in the domain. This test case has relevance to, for example, image processing or data analysis applications. The objective is to show that the OGRE scheme is able to reproduce the gradients to the correct (nominal) order of accuracy. The computational geometry for this case is shown in Figure 3.3. All boundary faces are treated identical to the interior faces by setting them to be of periodic/shadow type. The distribution of scalar ϕ throughout the domain is given by the following analytical function

$$\phi\,(x,\ y,\ z) = cos(4\pi(x-0.5))\ sin(4\pi(y-0.5))\ cos(4\pi(z-0.5)). \tag{3.85}$$

In order to assess the numerical accuracy of this methodology, it is important to first define how the errors are quantified. The error norms in a numerical calculation are the most common metric for accuracy assessment. For 3D formulations, the general form of the error norm can be defined as

$$L_p = \left(\frac{\sum_{i=1}^{N_{cv}} V_i \left| \bar{E}_i \right|^p}{\sum_{i=1}^{N_{cv}} V_i} \right)^{\frac{1}{p}}, \tag{3.86}$$

where

$$\bar{E}_i = \left| \phi_{i,exact} - \phi_{i,calculated} \right|, \tag{3.87}$$

p is the norm index, N_{cv} is the total number of control volumes in the domain, V_i is the volume of the control volume i, and $\bar{E}_i$ is the solution error in the control volume i. While L_1 and L_2 are global norms, L_∞ is a local error indicator since it shows the largest magnitude of the error in the solution domain. It is defined as

$$L_\infty = \max \left| \bar{E}_i \right|. \tag{3.88}$$

To test the numerical order of accuracy of the proposed method, four different unstructured (tetrahedral-based) meshes were created using the commercial mesh generation system Ansys Gambit. The coarsest mesh is denoted Mesh3D 1 and contained 7595 tetrahedral cells. The mesh was created such that opposite domain faces possess matching node/edge positions to permit periodic/shadow BC effectiveness. To create the second mesh (Mesh3D 2), all cell edges on the domain edges were decreased to half of their initial size, resulting in four times as many boundary domain faces/nodes. Mesh3D 2 was then generated, and the number of cells increased by a factor of approximately 8 (as expected). The procedure was repeated until four successively finer meshes were created. The numbers of cells/faces/nodes for these four meshes (and Mesh3D 5 which is used in the spectral resolution test case) are given in Table 3.1. For illustration purposes, Mesh3D 1 (the coarsest mesh) is shown in Figure 3.4.

It is expected that by successively refining the mesh, the error in the solution should follow the relation given by

$$\log(E) \propto -\frac{k}{3}\log(NDOF) \tag{3.89}$$

(see [4] for more details), where E is the error (in some given norm), $NDOF$ is the number of solution unknowns (which for the FV method, it is the total number of control volumes) and k is the numerical accuracy. Or alternatively, between any two different meshes (in 3D) with a number of cells equal to $NDOF_1$ and $NDOF_2$, respectively, the numerical accuracy can be approximated by

$$k = \frac{3\log\left(L_{p1}/L_{p2}\right)}{\log\left(NDOF_2/NDOF_1\right)}, \tag{3.90}$$

where L_{p1} and L_{p2} are the respective error norms (p = 1,2,∞) for the solution on each mesh.

For the test case considered here, the 2nd order OGRE scheme is used with a value $\sigma = 1.0$ and the 3rd order OGRE scheme is used with $\sigma = 1.7$. To assess performance, values for $\dfrac{\partial\phi}{\partial y}$ (2nd order) for $\dfrac{\partial\phi}{\partial y}$ and $\dfrac{\partial^2\phi}{\partial x^2}$ 3rd order) are compared for different schemes. The results for the L_2 error norms of $\dfrac{\partial\phi}{\partial y}$ and $\dfrac{\partial^2\phi}{\partial x^2}$ in comparison with their L-S counterparts are shown in Figures 3.5 and 3.6, respectively. The numerical error accuracy orders for the different schemes, obtained from the results for the two finest meshes, are indicated in the plots, and they compare well with the nominal accuracy of the given schemes. The results for $\frac{\partial\phi}{\partial y}$ using the 2nd order OGRE scheme compare well to the L-S 2nd order results. For the 3rd order OGRE scheme the error in the first derivative is smaller when compared to the L-S results, however the results for $\frac{\partial^2\phi}{\partial x^2}$ show a smaller numerical order of accuracy when compared to L-S. The other first- and second-derivative components showed similar behavior to the ones shown here.

A qualitative examination of the results on the coarsest mesh (Mesh3D 1) when using OGRE and L-S methods for the third-order scheme is provided in Figures 3.7, 3.8 and 3.9, showing the distribution of $\frac{\partial\phi}{\partial y}$ for the OGRE 3rd order scheme (with $\sigma = 1.7$), for exact (analytical) values, and for the L-S 3rd order scheme, respectively. Both the OGRE and L-S schemes yield a close approximation of the analytical result, with the OGRE scheme showing slightly better agreement, at least in the first derivative. The results show that OGRE is able to better capture the extreme (maximum and minimum) values for the first derivatives in a static variable field. To evaluate the differences in performance between the two schemes, it is necessary to investigate results from the sample application (finite-volume solution of advection equation), and these results are presented in the following subsections.

Because the OGRE scheme is implicit, the gradient field must be found through an iterative procedure. It was found that about 26 iterations were sufficient to obtain a gradient solution which presents no more than 10% of difference of the fully converged one for a static variable field. The L-S method, however, is an explicit scheme, and only one iteration is required to obtain the gradients in each cell. For the 3rd order scheme, this represents a tradeoff in terms of computational expense, since the stencil support is reduced in OGRE but multiple iterations are required for gradient convergence.

3.5.2 Simple Scalar Transport Test Case

The first finite-volume test case examines simple advection of a smoothly distributed profile of an arbitrary scalar transport variable ϕ. The geometry and boundary conditions for this case are shown in Figure 3.10. The domain extends one unit length in all directions. A uniform convective velocity in the z-direction ($u_z = 1$) is applied throughout the domain. In this example, the $z = 0$ boundary is a flow (velocity) inlet (Dirichlet-type boundary condition), the $z = 1$ boundary is defined as a flow (pressure) outlet, and all the other four remaining domain boundaries are defined as symmetry conditions (zero-flux Neumann boundary conditions). The inlet profile of ϕ is given by the so-called 3D “Mexican Hat” function:

$$\phi\left(x, y, 0\right) = \left(1 - 200\left(\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2\right)\right) e^{\left(-100\left(\left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2\right)\right)}. \tag{3.91}$$

The inlet profile is shown graphically in Figure 3.11. The exact solution for this test case is simple advection of the profile in the z-direction, i.e. $\phi(x, y, z) = \phi(x, y, 0)$. The same set of four meshes used in the Section 2.5.1 is utilized here, but with the appropriate boundary conditions given in Figure 3.10.

To determine the optimum values to be used for σ in all test cases, the code was run on Mesh3D 1 with different σ values and the results for total number of iterations (needed for convergence at $Residual < 10^{-10}$) and L_2 error norms were obtained. These are plotted in Figures 3.12 and 3.13, for OGRE 2nd and 3rd order schemes, respectively. As expected, there is a small tradeoff in choosing the value for the tuning coefficient σ. The results show that a higher value for the tuning coefficient σ degrades stability (indicating by taking more iterations to obtain convergence) but at the same time improves numerical accuracy. On the other hand, a very small σ value also produces a decay in stability. From the plots, it is apparent that, for the OGRE 2nd order scheme, values ranging from 0.8 to 1.2 seem adequate, and for the OGRE 3rd order scheme, the choice for σ depends on the needs of accuracy and computational resources. A reasonable tradeoff between stability and accuracy is obtained with $\sigma = 1.0$ for OGRE 2nd order scheme and $\sigma = 1.7$ for OGRE 3rd order scheme. These values are used for all of the test cases presented.

The initial intention was to compare OGRE results with those using the standard L-S scheme. Unfortunately, the L-S scheme (2nd and 3rd order), without the use of any slope limiter, was unable to obtain convergence for this test case. The comparison is therefore performed with two pre-defined schemes inside Ansys Fluent (v. 12.0): the L-S cell based 2nd order upwind and 3rd order MUSCL scheme [28]. Fluent's 2nd order upwind scheme is used with the Default (TVD) Slope Limiter in combination with the Cell to Face Limiting Direction option, where the limited value of the reconstruction gradient is determined at cell face centers. Fluent's 3rd order MUSCL scheme blends a central differencing scheme and a second-order upwind scheme, and it does not contain any gradient/flux limiter. The results for L_1 and L_2 error norms for the different meshes using the different schemes are shown in Figures 3.14 and 3.15, respectively, and in Table 3.2. The numerical accuracy orders for the different error norms, obtained from a linear fit of the results on the two finest meshes, are shown in the Table 3.3 and indicated in the plots.

The transport scalar solutions on Mesh3D 2 for $x = 0.5$ cross section, using OGRE 3rd order scheme (with $\sigma = 1.7$) and Fluent's 3rd order MUSCL scheme, are shown in Figures 3.16, 3.17, 3.18 and 3.19. These figures show that the OGRE 3rd order scheme produces a less dissipative distribution of the variable field throughout the domain when compared to the solution obtained using the Fluent 3rd order MUSCL scheme, which clearly seems to produce more numerical dissipation.

The global error norms (L_1 and L_2) listed in Table 3.3 indicate that the new methodology (OGRE) yields a measured numerical order of accuracy equal to or higher than the nominal order of accuracy for both the 2nd and 3rd order schemes, for all L norms. This is also clearly shown in the plots of L_1 and L_2 norms in Figures 3.14 and 3.15. On the other hand, the Fluent L-S 2nd order scheme and 3rd order MUSCL scheme produce a measured numerical order of accuracy much smaller than their nominal order of accuracy. The results therefore demonstrate that there is a clear advantage to using the OGRE (2nd and 3rd order) schemes versus Fluent's L-S 2nd-order L-S (with slope limiter) and Fluent's 3rd order MUSCL schemes, respectively, for this test case. Based on 2D results presented in [4], it is expected that the unlimited L-S results would have demonstrated accuracy orders consistent with their nominal order, however as mentioned above converged results were unable to be obtained. Not only the error order, but the overall error level in the OGRE scheme is also consistently lower, by a significant amount, for all mesh sizes. As a result, the L_2 error using OGRE 3rd-order scheme is about an order of magnitude lower than the equivalent L-S scheme by Mesh3D 2, and two orders of magnitude lower by Mesh3D 5. This highlights one of the primary advantages of the OGRE scheme, namely the use of a stencil comprised only of immediate neighbors, rather than a larger stencil as required by L-S for higher order. As discussed in [4], error is proportional to h^k, where h is the characteristic mesh size and k is the order of accuracy. When the stencil size is increased, the effective mesh size h is likewise increased. The results in Figures 3.14 and 3.15 show the effect of both contributions in the OGRE scheme: higher k and lower h. It is expected that this advantage – smaller effective mesh size due to smaller stencil – will become even more pronounced in contrast to the L-S scheme as the nominal order of the scheme is increased beyond 3.

As discussed in the previous subsection, the OGRE scheme is implicit, and multiple iterations are required to converge the gradient field. This requirement led to an increase in computational expense versus the L-S scheme for the static variable case. In the finite-volume simulations, however, this expense is mitigated. Because the solution algorithm is itself implicit, the gradient field was simply updated once per iteration. Because the convergence rate for the gradients is much more rapid than for the solution of ϕ itself, this was sufficient to allow both the variable and gradient fields to converge together. As a consequence, the overall cost per iteration using OGRE was effectively equal to the cost per iteration using L-S.

Convergence for all cases was defined by Residual $< 10^{-10}$. All OGRE results are fully converged results. None of the Fluent schemes were able to converge to that level, due to the necessity of including a slope limiter. It was observed that these schemes could not obtain residuals smaller than 10^{-5} or 10^{-6}, depending on the mesh and/or the order of the scheme.

3.5.3 Spectral Resolution Test Case

The second transport test case was performed to assess the spectral behavior of the new gradient calculation scheme. The case used the same computational geometry described in Section 2.5.2, with a uniform z-direction velocity field. The inlet and outlet boundary condition types are the same as above, but the other four boundary surfaces are assumed to be periodic. In order to apply 2D Fast Fourier Transform (2D FFT) on the outlet, two approaches could be used. One approach could be to use a fully tetrahedral mesh, where the outlet solution should be interpolated to a structured (2D) solution, and then the 2D FFT could be applied on it. A second approach could be to create a hybrid mesh where both inlet and outlet surface meshed were Cartesian structured (2D). This yields a 3D mesh where two layers of pyramids (at the inlet and at the outlet) are combined with interior tetrahedral elements. The first approach introduces interpolation errors, which would be difficult to distinguish from errors arising from the numerical scheme itself. The second approach slightly decreases mesh quality, but is expected to have little effect on the final result. Therefore the second approach was chosen for the test case.

A new mesh (Mesh3D 5) was created for which the inlet and outlet faces are comprised of 64×64 equally distributed quadrilateral cells, for the purposes of performing Fourier analysis of the inlet and outlet variable profiles in terms of their spectral content. In order to test the new scheme for situations which contain a wide range of scales, an inlet profile for the transport scalar ϕ was synthesized as a sum of Fourier modes. The inlet profile was defined as

$$\phi\left(x,\ y,\ 0\right)=\sum_{\vec{K}=(1,1)}^{(N,N)}\left[a_{\vec{K}}\cos\left(2\pi\vec{K}\cdot\vec{x}\right)+b_{\vec{K}}\sin\left(2\pi\vec{K}\cdot\vec{x}\right)\right]. \tag{3.92}$$

In Eq. (3.92), the phases of the Fourier modes were randomly chosen, and the amplitude for each mode was equal to 1.0, i.e. ($a_{\vec{K}}^2+b_{\vec{K}}^2=1$) and $N=31$ (= Nyquist frequency - 1). As in the previous subsection, the exact solution is given by $\phi(x,y,z)=\phi(x,y,0)$. The simulation was run using each of the different gradient calculation schemes, and the solution profile at the outlet was compared to the profile at the inlet. In both cases, the discrete values $\phi(y)$ were transformed to wavenumber space using the 2D Fast Fourier Transform. Figure 3.20 shows the transformed profile at the outlet for five different cases: 1st order upwind, Fluent's L-S 2nd order and 3rd order MUSCL schemes, and OGRE 2nd (with $\sigma=1.0$) and 3rd (with $\sigma=1.7$) order schemes. The plot clearly shows the lack of effectiveness in resolving higher wavenumber modes of the solution when using low order schemes. The 1st order upwind cannot even reconstruct completely the lowest wavenumber mode. Consistent with previous results, both Fluent schemes yield similar results, and are able to resolve at least 90% of the amplitude of modes up to about wavenumber 2. As in the previous subsection, a significant improvement in accuracy is apparent when using the new scheme.

OGRE 2nd order scheme resolves about 90% of the amplitudes up to about wavenumber 5, and OGRE 3rd order scheme can do the same up to wavenumber 8. This corresponds to resolving variable modes with a spatial period equal to approximately eight characteristic cell sizes. Just as a comparison, in [4], it is shown that OGRE 3rd order scheme (2D) was able to resolve 90% of the amplitudes up to wavenumber 27 (over the total 63), which corresponded to structures of about five characteristic cell sizes. This relative decrease in accuracy when comparing 2D and 3D results is expected since not only there is a natural decay in general mesh quality when going from a triangle-based 2D mesh to a tetrahedral-based 3D mesh, but also since there was the need to introduce two layers of pyramids (at inlet and outlet), the decay in mesh quality was accentuated. Nevertheless, the results clearly show that the OGRE scheme is able to resolve significantly more of the variable spectrum than the other methods, and that the resolution improves as the order of the scheme is increased.

These results not only reinforce those shown in Figures 3.14 and 3.15, they also have important implications for applications requiring high-resolution solution algorithms for a range of spatial scales. For example, in finite-volume CFD applications, these results suggest that the OGRE scheme may provide better numerical accuracy in large-eddy simulations by producing less numerical dissipation.

Table 3.1

Mesh specifications used in all 3D test cases.

Mesh #	# of cells	# of faces	# of nodes
Mesh3D 1	7595	15,872	1,637
Mesh3D 2	61,423	125,532	11,633
Mesh3D 3	425,174	861,124	76,325
Mesh3D 4	3,558,515	7,117,030	614,602
Mesh3D 5	1,233,466	2,475,124	220,330

Table 3.2

Error norms for the 3D simple scalar transport test case on different schemes and meshes.

Mesh -	OGRE	OGRE	FLUENT’s	FLUENT’s
error norm	2nd Order	3rd Order	L-S	3rd Order
	$\sigma = 1.0$	$\sigma = 1.7$	2nd Order	MUSCL
M1 - L_1	0.0198994	0.0113373	0.0220121	0.0217875
M1 - L_2	0.0607814	0.0345558	0.0724993	0.0704919
M1 - L_∞	0.858366	0.532718	1.02486	0.981531
M2 - L_1	0.00629042	0.00146902	0.0101475	0.0115058
M2 - L_2	0.0235183	0.00609128	0.0389089	0.0430283
M2 - L_∞	0.436463	0.124493	0.692294	0.716423
M3 - L_1	0.00142081	0.000154042	0.0035416	0.00508348
M3 - L_2	0.00552696	0.000630339	0.0143198	0.0193745
M3 - L_∞	0.145242	0.0178307	0.340436	0.416208
M4 - L_1	0.000277256	1.742e-05	0.00140122	0.00238814
M4 - L_2	0.00109886	7.04784e-05	0.00553681	0.00909881
M4 - L_∞	0.0336132	0.00212322	0.167604	0.219004

Table 3.3

Numerical orders of accuracy for the error in each scheme between the two finest meshes for the 3D simple scalar transport test case.

error	OGRE	OGRE	FLUENT’s	FLUENT’s
norm	2nd Order	3rd Order	L-S	3rd Order
	$\sigma = 1.0$	$\sigma = 1.7$	2nd Order	MUSCL
L_1	2.31	3.08	1.31	1.07
L_2	2.28	3.09	1.34	1.07
L_∞	2.07	3.00	1.00	0.91

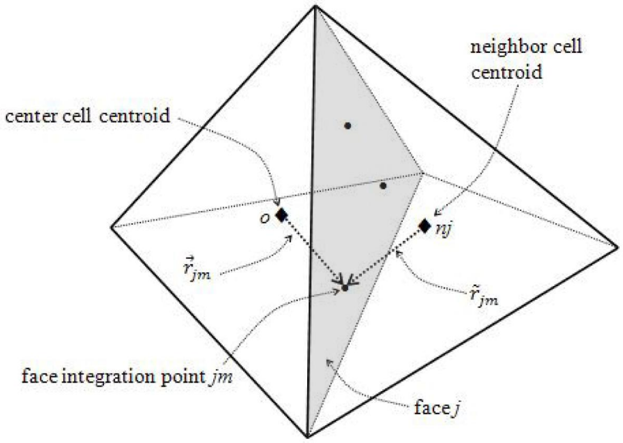

Figure 3.1

3D OGRE implementation for interior cells: center cell centroid, face neighbor centroid and face quadrature (integration) points (three, for the 3rd order case), with respective connecting vectors over generic face quadrature point jm.

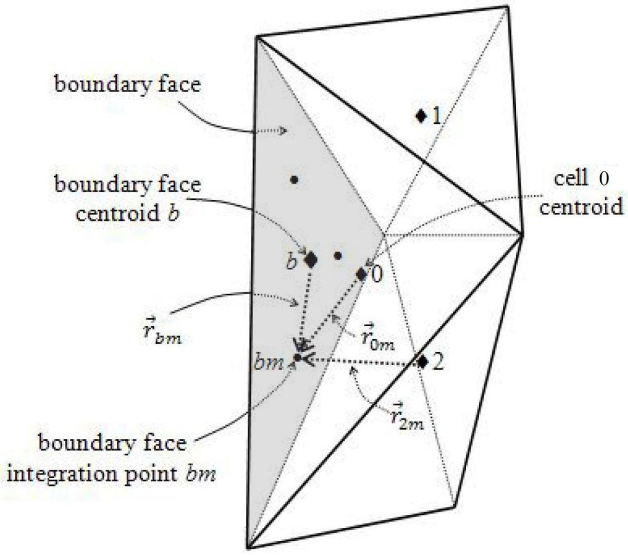

Figure 3.2

3D boundary condition implementation.

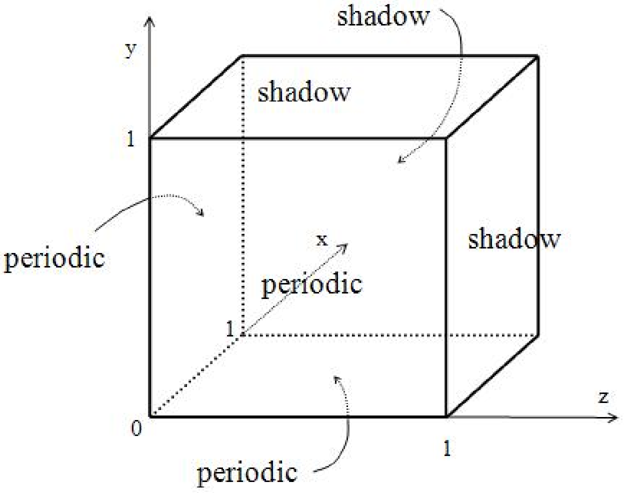

Figure 3.3

3D geometry and boundary conditions for the static variable test case.

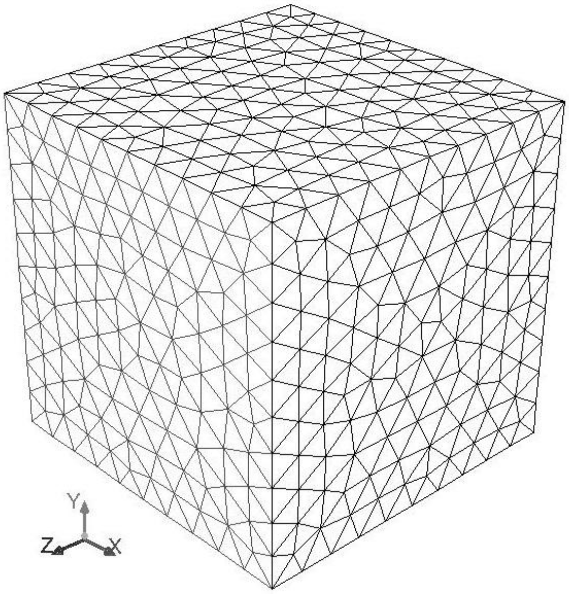

Figure 3.4

Mesh3D 1.

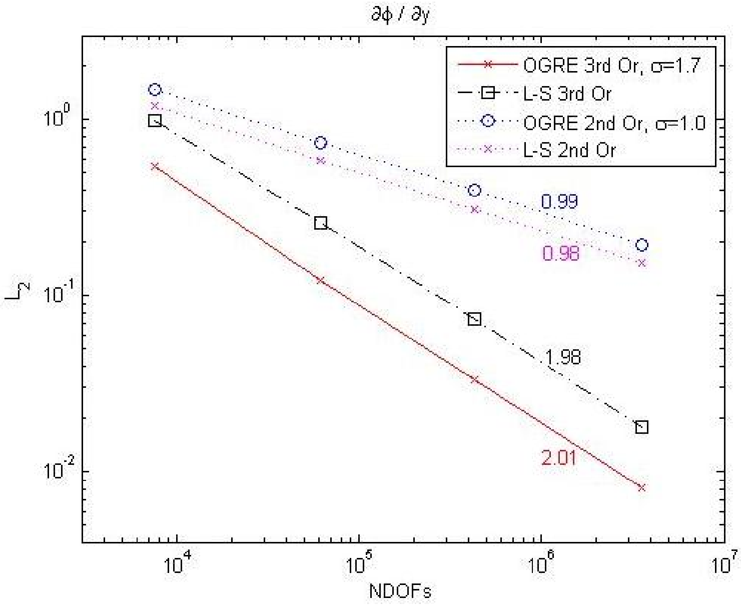

Figure 3.5

L_2 error norm values of $\dfrac{\partial \phi}{\partial y}$ for the 3D static variable test case.

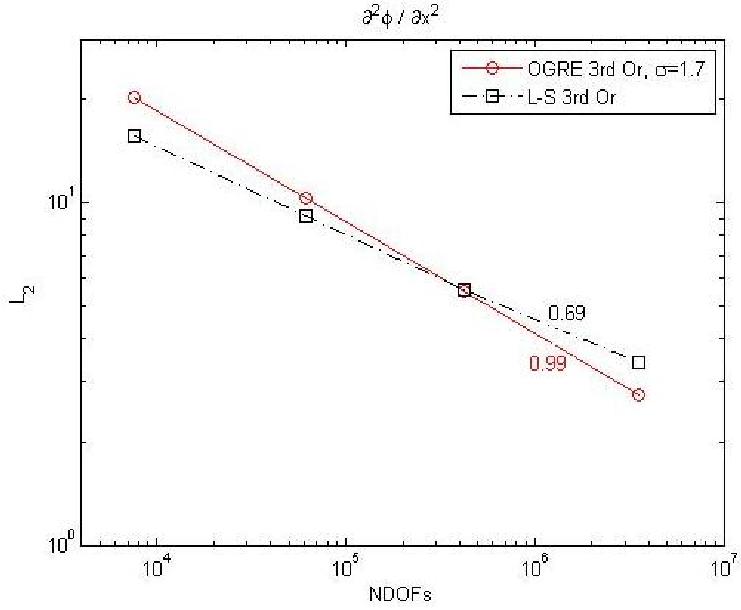

Figure 3.6

L_2 error norm values of $\dfrac{\partial^2 \phi}{\partial x^2}$ for the 3D static variable test case.

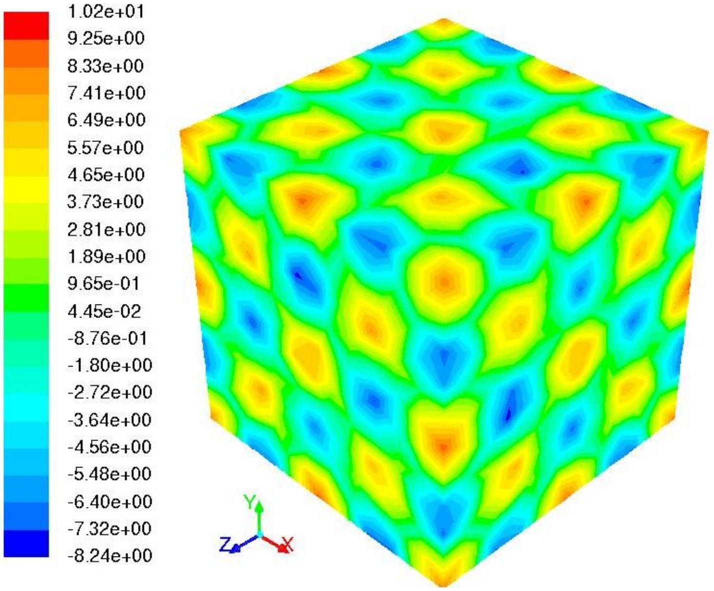

Figure 3.7

OGRE 3rd order solution of $\frac{\partial \phi}{\partial y}$ using σ = 1.7 for the static variable field test case on Mesh3D 1.

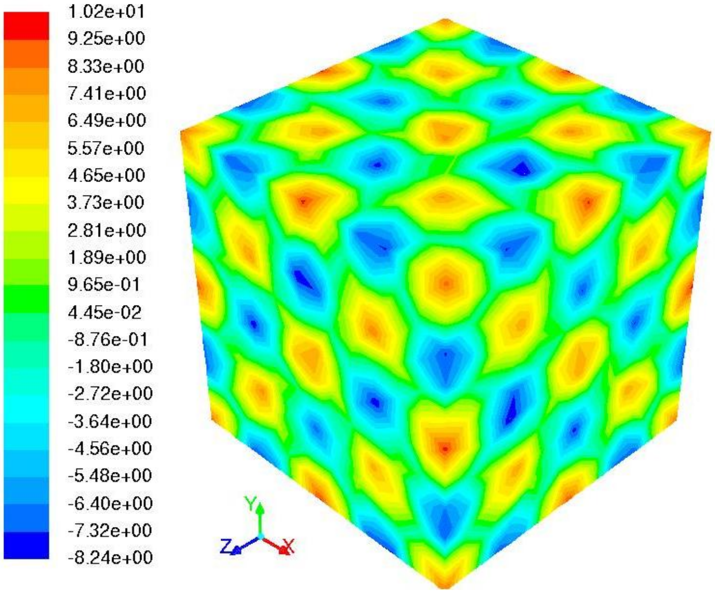

Figure 3.8

Analytical solution of $\frac{\partial \phi}{\partial y}$ for the static variable test case on Mesh3D 1.

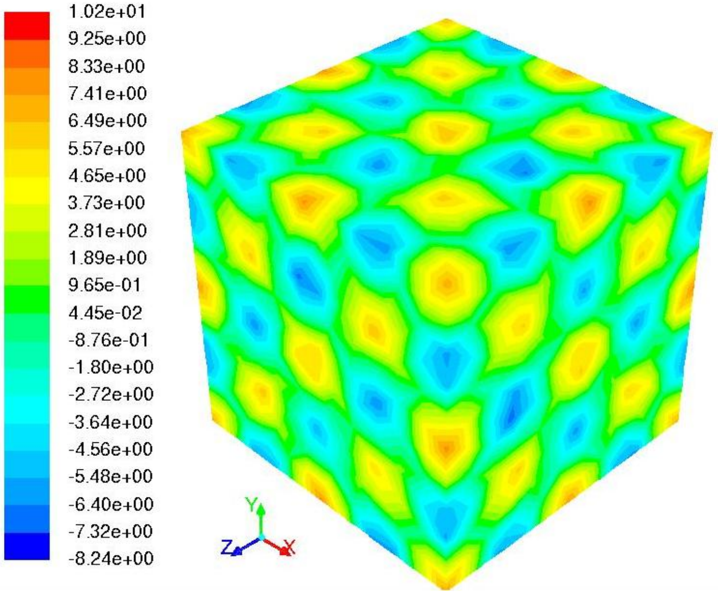

Figure 3.9

L-S 3rd order solution of $\dfrac{\partial \phi}{\partial y}$ for the static variable test case on Mesh3D 1.

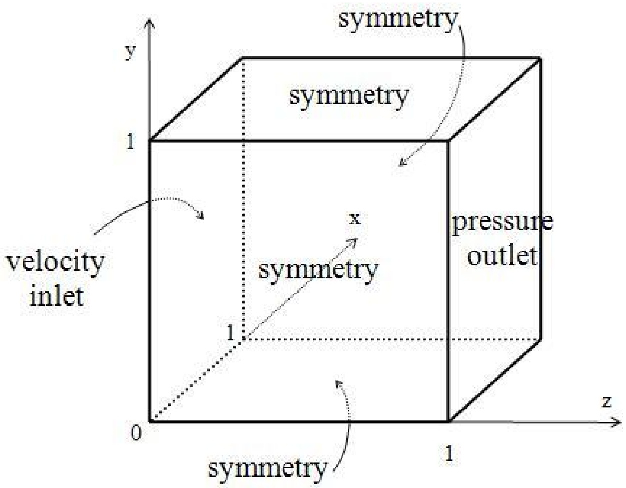

Figure 3.10

3D geometry and boundary conditions for the simple scalar transport test case.

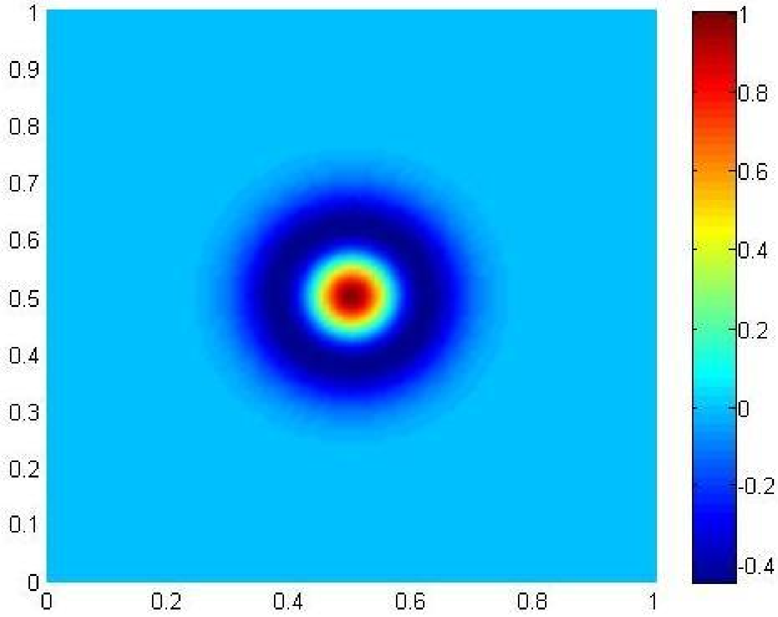

Figure 3.11

3D inlet profile: Mexican Hat function.

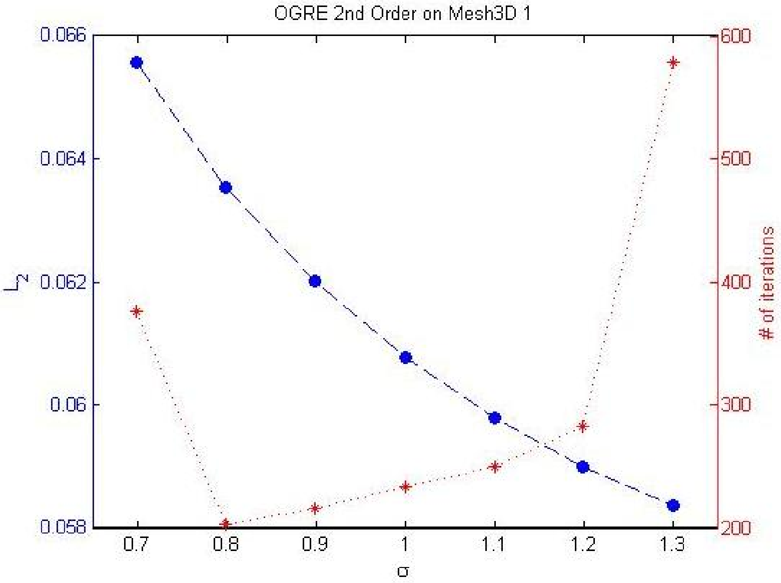

Figure 3.12

σ effect on OGRE 2nd order, Mesh3D 1.

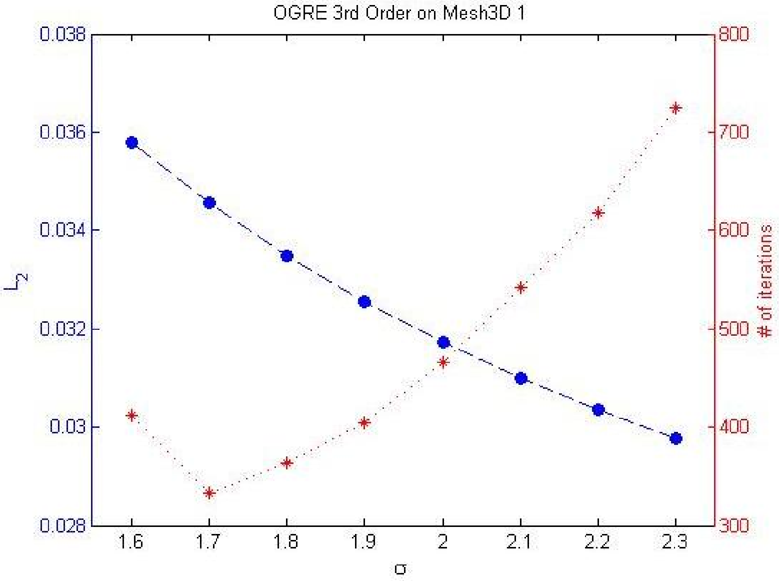

Figure 3.13

σ effect on OGRE 3rd order, Mesh3D 1.

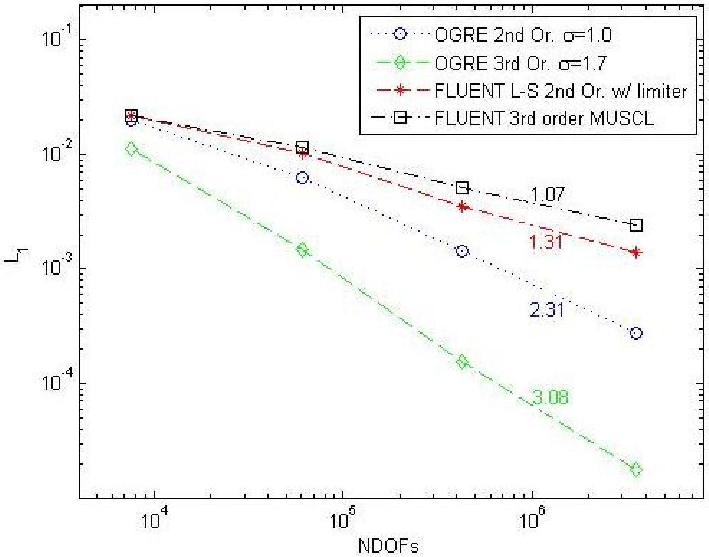

Figure 3.14

L_1 error norms for the 3D simple scalar transport test case.

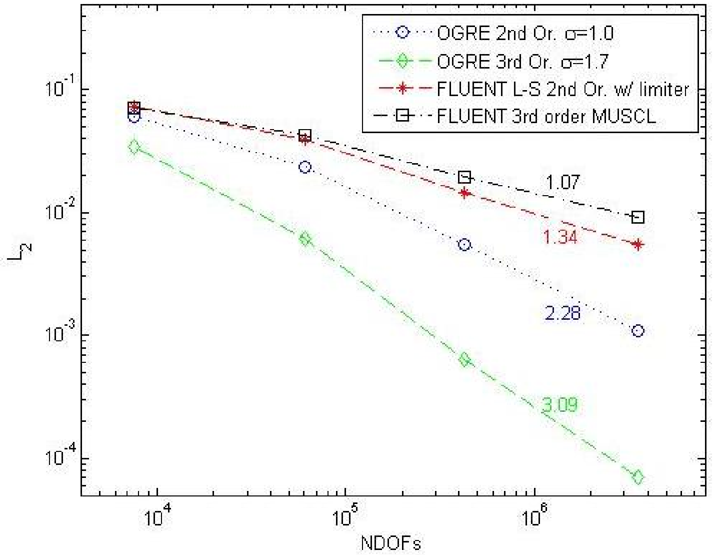

Figure 3.15

L_2 error norms for the 3D simple scalar transport test case.

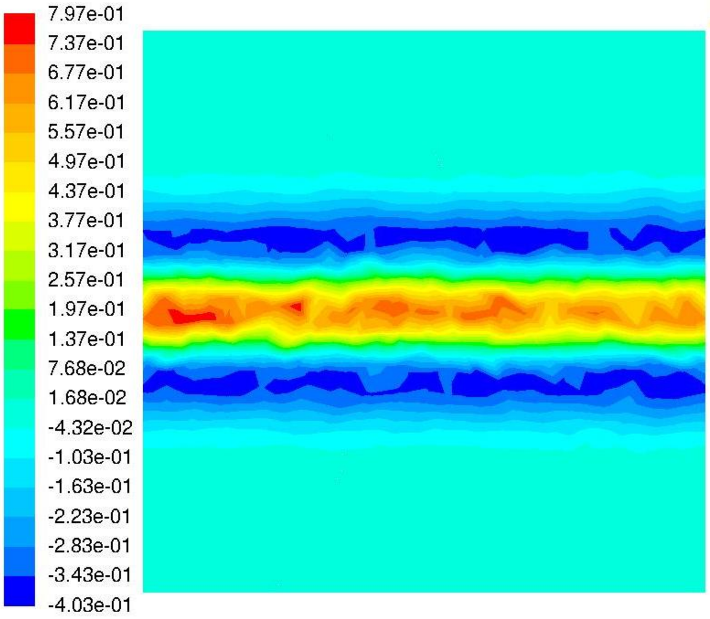

Figure 3.16

ϕ distribution throughout the domain on Mesh3D 2 for $x = 0.5$ cross section using OGRE 3rd order scheme, σ=1.7, for the simple scalar test case.

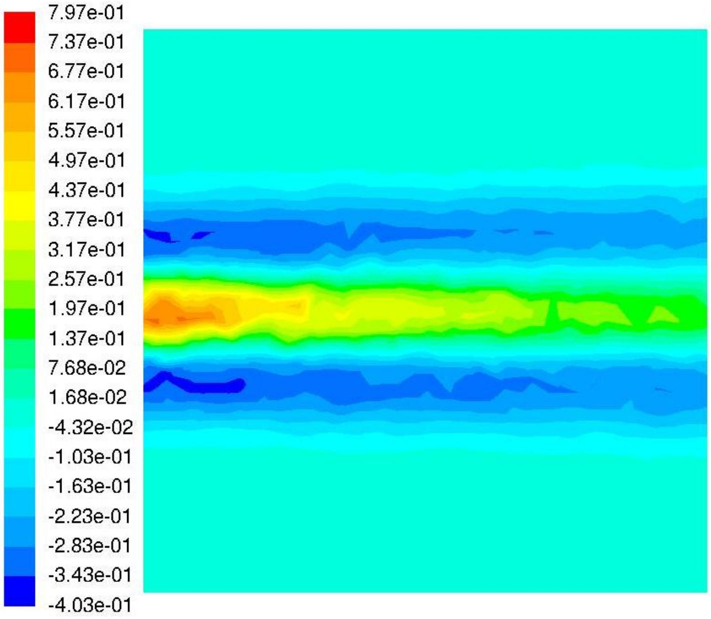

Figure 3.17

ϕ distribution throughout the domain on Mesh3D 2 for $x = 0.5$ cross section using Fluent 3rd order MUSCL scheme, for the simple scalar test case.

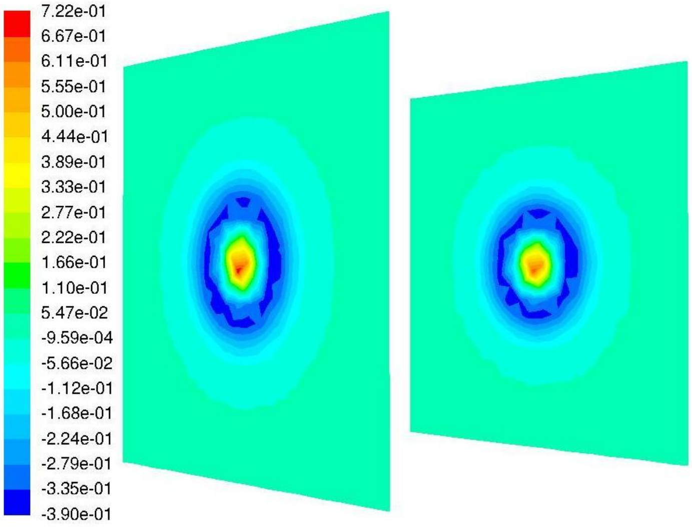

Figure 3.18

ϕ distribution throughout the domain on Mesh3D 2 for $x = 0.5$ cross section using OGRE 3rd order scheme, σ=1.7, for the simple scalar test case – inlet/outlet comparison.

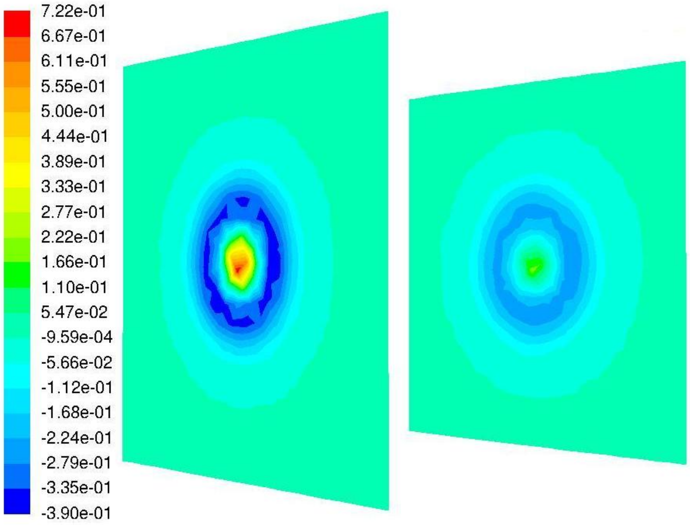

Figure 3.19

ϕ distribution throughout the domain on Mesh3D 2 for $x = 0.5$ cross section using Fluent 3rd order MUSCL scheme, for the simple scalar test case – inlet/outlet comparison.

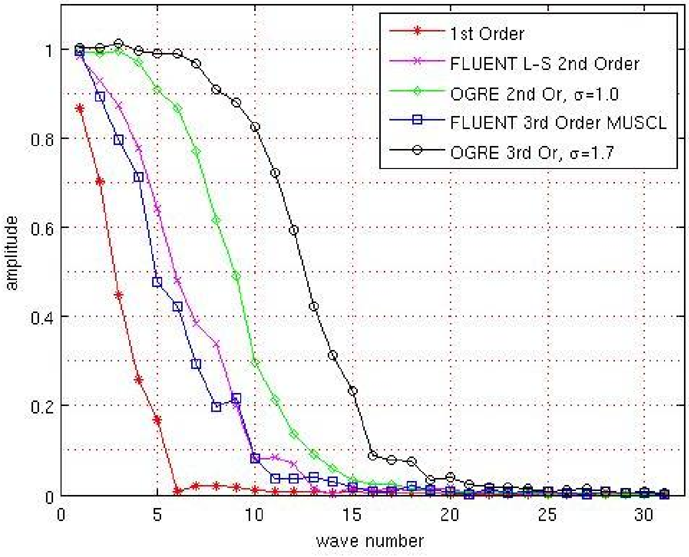

Figure 3.20

2D FFT outlet reconstruction using different schemes for the 3D spectral analysis test case.

CHAPTER 4

CONCLUSIONS

4.1 Contributions

A new scheme (Optimization-based Gradient Reconstruction – OGRE) was presented for calculating gradients to arbitrarily high order on unstructured meshes. The development included both 2nd and 3rd order variants of the scheme for two and three-dimensional meshes. The key aspect of the new scheme is that only first-level neighbors are used in the solution stencil, which minimizes the effective mesh spacing for high-order (greater than one) gradient calculations. This is in contrast to the more commonly used least-squares (L-S) approach. Preliminary results show the proposed method is able to deliver a numerical accuracy equivalent to (or better than) the nominal order of accuracy for both 2nd and 3rd order schemes. The order of accuracy was confirmed for a static variable test case, by examining the order of accuracy of the computed gradients, and for a sample application – finite-volume solution of a scalar advection equation – by examining the order of accuracy of the variable field solution.

The results from finite-volume simulations showed that the tuning parameter (σ) in the new scheme can be adjusted to effect a tradeoff between accuracy and stability/convergence rate. This feature of the scheme may be useful for stabilization on low-quality meshes.

For 2D implementation, in the case of the finite-volume simulation of a simple advection transport problem, the 2nd order variant of the new scheme showed almost identical results to the L-S scheme, and exhibited approximately the same level of computational expense. For the 3rd order variant, however, the new scheme had significantly lower numerical error regardless of grid resolution, as well as a higher measured order of accuracy. As a consequence, the new method was able to reproduce results to the same level of accuracy as the L-S method using a grid with 1/16th the number of cells. The static variable test case results showed the first derivatives calculated using OGRE 3rd order scheme possess a consistently lower numerical error compared to the ones using L-S scheme for all meshes.

For 3D implementation, in the case of the finite-volume simulation of a simple advection transport problem, both variants of the novel scheme obtained full convergence, while the L-S scheme could not obtain full convergence (without using any limiter). When compared to the Fluent L-S 2nd order and 3rd order MUSCL schemes, both variants of the new scheme had significantly lower numerical error regardless of grid resolution, as well as a higher measured order of accuracy. Therefore, the OGRE 3rd order scheme using a grid with about 1/64th the number of cells was able to produce results with smaller error than those produced by the 3rd order MUSCL scheme. As in 2D, the 3D static variable test case results showed the first derivatives calculated using OGRE 3rd order scheme possess a consistently lower numerical error compared to the ones using L-S scheme in either mesh resolution.

Spectral analysis shows that the new scheme possesses a desirable ability to better capture small scales (higher wavenumber modes) of the numerical solution, in comparison with the L-S results in 2D, and in comparison with the Fluent L-S 2nd order and 3rd order MUSCL schemes in 3D. This behavior is presumably due to the use of a smaller stencil in the formulation. The results demonstrate the schemes potential to address problems presenting a wide spectrum of wavelengths, for example unsteady turbulent flows.

The OGRE scheme was shown to produce equivalent 2nd order results as those obtained by the L-S scheme for the 2D test cases presented here. For 3D, 2nd order of accuracy, the OGRE scheme produced more stable results when applied for the simple scalar transport test case, since its equivalent L-S 2nd order of accuracy scheme without any limiter could not obtain convergence. OGRE also produced more effective higher-order (3rd order) numerical results in terms of error accuracy, stability and memory/computational cost (for a given level of required precision) when compared to the Least-Squares scheme in 2D and 3rd order MUSCL scheme in 3D, for the simple scalar transport test cases.

4.2 For Further Research

Future research efforts will include development and implementation of the OGRE scheme for 3D unsteady calculations, and application of the new scheme to more realistic problems, for example Reynolds-averaged Navier-Stokes (RANS) and large-eddy simulations (LES) of turbulent flows.

REFERENCES

[1] R. Abgrall, "On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation," *J. Comput. Phys.*, vol. 114, no. 1, Sept. 1994, pp. 45–58.

[2] T. J. Barth and P. O. Frederickson, "Higher-order solution of the textscEuler equations on unstructured grids using quadratic reconstruction," *AIAA paper 90-0013*, 1990.

[3] T. J. Barth and D. C. Jespersen, "The design and application of upwind schemes on unstructured meshes," *AIAA-89-0366*, 1989.

[4] A. D. Busatto and D. K. Walters, "An optimization-based method for high order gradient calculation on unstructured meshes - 2D implementation," *To appear*, 2012.

[5] A. D. Busatto and D. K. Walters, "An optimization-based method for high order gradient calculation on unstructured meshes - 3D implementation," *To appear*, 2012.

[6] E. M. Collins, *On Mesh Quality Considerations for the Discontinuous Galerkin Method*, doctoral dissertation, Department of Computational Engineering, Mississippi State University, Starkville, Mississippi, 2009.

[7] C. Correa, R. Hero, and K.-L. Ma, "A Comparison of Gradient Estimation Methods for Volume Rendering on Unstructured Meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 3, March 2011, pp. 305–319.

[8] S. Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, 1986, pp. 116 – 125.

[9] L. J. Durlofsky, B. Engquist, and S. Osher, "Triangle based adaptive stencils for the solution of hyperbolic conservation laws," *J. Comput. Phys.*, vol. 98, no. 1, Jan. 1992, pp. 64–73.

[10] J. A. Ekaterinaris, "High-order accurate, low numerical diffusion methods for aerodynamics," *Prog in Aerospace Sci*, vol. 41, 2005, pp. 192–300.

[11] O. Friedrich, "Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids," *J. Comput. Phys.*, vol. 144, no. 1, July 1998, pp. 194–212.

[12] R. E. Harris and Z. J. Wang, "High-order adaptive quadrature-free spectral volume method on unstructured grids," *Computers QQQ Fluids*, vol. 38, no. 10, 2009, pp. 2006 – 2025.

[13] C. Hu and C. W. Shu, "Weighted Essentially Non-Oscillatory Schemes on Triangular Meshes," *J. Comput. Phys*, vol. 150, 1998, pp. 97–127.

[14] K. H. Jones, "A comparison of algorithms used to compute hill slope as a property of the DEM," *Computers QQQ Geosciences*, vol. 24, no. 4, 1998, pp. 315 – 323.

[15] J.-S. Lee, "Refined filtering of image noise using local statistics," *Computer Graphics and Image Processing*, vol. 15, no. 4, 1981, pp. 380 – 389.

[16] K. Michalak and C. Ollivier-Gooch, "Matrix-explicit GMRES for a higher-order accurate inviscid compressible flow solver," 2007, vol. 1, pp. 362–372.

[17] A. Nejat and C. Ollivier-Gooch, "A high-order accurate unstructured GMRES algorithm for inviscid compressible flows," *Proceedings of the Seventeenth AIAA Computational Fluid Dynamics Conference*, June 2005.

[18] A. Nejat and C. Ollivier-Gooch, "A high-order accurate unstructured Newton-Krylov solver for inviscid compressible flows," 2006, vol. 3, pp. 1878–1895.

[19] A. Nejat and C. Ollivier-Gooch, "Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations," *J. Comput. Phys.*, vol. 227, no. 4, Feb. 2008, pp. 2366–2386.

[20] A. Nejat, C. Ollivier-Gooch, and K. Michalak, "Accuracy assessment methodology for a high-order unstructured finite volume solver," 2007, vol. 2, pp. 1071–1091.

[21] C. Ollivier-Gooch, "High-Order ENO Schemes for Unstructured Meshes Based on Least-Squares Reconstruction," *AIAA Paper 97-0540*, 1997, pp. 97–0540.

[22] C. Ollivier-Gooch, "Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction," *J. Comput. Phys.*, vol. 133, no. 1, May 1997, pp. 6–17.

[23] C. Ollivier-Gooch, A. Nejat, and K. Michalak, "On Obtaining High-Order Finite-Volume Solutions to the Euler Equations on Unstructured Meshes," *Proceedings of the Eighteenth AIAA Computational Fluid Dynamics Conference*, vol. 2, no. June, 2007, pp. 1–16.

[24] C. Ollivier-Gooch and M. Van Altena, "A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation," *J. Comput. Phys.*, vol. 181, no. 2, Sept. 2002, pp. 729–752.

[25] J. Reddy, *An Introduction to the Finite Element Method*, McGraw-Hill Series in Mechanical Engineering. McGraw-Hill Higher Education, 2006.

[26] C. W. Shu, "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws," *Advanced numerical approximation of nonlinear hyperbolic equations*, B. Cockburn, C. Johnson, C. W. Shu, E. Tadmor, and A. Quarteroni, eds., Berlin, 1998, vol. 1697 of *Lecture Notes in Mathematics*, pp. 325–432, Springer.

[27] T. Sonar, "On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: Polynomial recovery, accuracy and stencil selection," *Computer Methods in Applied Mechanics and Engineering*, vol. 140, no. 1-2, 1997, pp. 157–181.

[28] B. van Leer, "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method," *Journal of Computational Physics*, vol. 32, no. 1, 1979, pp. 101 – 136.

[29] J. Vassberg, "Expectations for computational fluid dynamics," *International Journal of Computational Fluid Dynamics*, vol. 19, no. 8, 2005, pp. 549–558.

[30] Z. J. Wang, "High-order methods for the Euler and NavierStokes equations on unstructured grids," *Progress in Aerospace Sciences*, vol. 43, no. 1-3, 2007, pp. 1–41.

APPENDIX A

2D OGRE THIRD ORDER SCHEME

A.1 2D Implementation of OGRE Third Order Scheme

The matrix of coefficients A is symmetric and is given by the block matrix:

$$A_{5x5} = \begin{bmatrix} C_{2\times2} & D_{2\times3} \\ \\ D^T_{3\times2} & E_{3\times3} \end{bmatrix}, \tag{A.1}$$

where

$$c_{ik} = \sum\sum \left(\sigma^4 + \sigma^2\right) \Delta x_{i,jm} \Delta x_{k,jm}, \tag{A.2}$$

$$d_{ik} = \sum\sum 2\left(1 - |k-2|\right)\left(\frac{\sigma^4}{2} + \sigma^2\right) \Delta x_{1,jm}^{5-(i+k)} \Delta x_{2,jm}^{-2+(i+k)}, \tag{A.3}$$

$$e_{ik} = \sum\sum 2^{2-(|i-2|+|k-2|)}\left(\frac{\sigma^4}{4} + \sigma^2 + 1\right) \Delta x_{1,jm}^{6-(i+k)} \Delta x_{2,jm}^{-2+(i+k)}, \tag{A.4}$$

the coordinates are given by $(x, y) = (x_1, x_2)$, and the following simplification is used: $\sum\sum = \sum\limits_{j=1}^{\#neig} \sum\limits_{m=1}^{2}$. The components of the *RHS* vector B, containing just current values, are given by the block matrix:

$$B_{5x1} = \begin{Bmatrix} P_{2\times1} \\ \\ Q_{3\times1} \end{Bmatrix}, \tag{A.5}$$

where

$$p_{i1} = \sum\sum \sigma^4 \Delta x_{i,jm}\left(\tilde{F}_{jm} - \phi_o\right) + \sigma^2 \Delta x_{i,jm} \tilde{G}_{jm}, \tag{A.6}$$

$$q_{i1} = \sum\sum 2\left(1 - |i-2|\right) \Delta x_{1,jm}^{3-i} \Delta x_{2,jm}^{-1+i}\left[\frac{\sigma^4}{2}\left(\tilde{F}_{jm} - \phi_o\right) + \sigma^2 \tilde{G}_{jm} + \tilde{H}_{jm}\right], \tag{A.7}$$

and where the averaged values $\tilde{F}_{jm}$, $\tilde{G}_{jm}$ and $\tilde{H}_{jm}$, are given by (2.57), (2.59) and (2.61), respectively. The solution vector s with the values of the derivatives at the center cell centroid is given by:

$$s = \left[\ \left.\frac{\partial\phi}{\partial x}\right|_o \quad \left.\frac{\partial\phi}{\partial y}\right|_o \quad \left.\frac{\partial^2\phi}{\partial x^2}\right|_o \quad \left.\frac{\partial^2\phi}{\partial x \partial y}\right|_o \quad \left.\frac{\partial^2\phi}{\partial y^2}\right|_o\ \right]^T. \tag{A.8}$$

A.2 2D BC Implementation of OGRE Third Order Scheme

The following simplification is used: $\sum\sum = \sum_{k=0}^{2}\sum_{m=1}^{2}$. Since the matrix of coefficients A is symmetric, just the upper diagonal components, for the Dirichlet boundary condition case, are given here:

$$a_{11} = \sum\sum\left(2\sigma^4\right),\ a_{12} = \sum\sum\left(2\sigma^4\Delta x_{bm}\right),\ a_{13} = \sum\sum\left(2\sigma^4\Delta y_{bm}\right),$$

$$a_{14} = \sum\sum\left(\sigma^4\Delta x_{bm}^2\right),\ a_{15} = \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta y_{bm}\right),$$

$$a_{16} = \sum\sum\left(\sigma^4\Delta y_{bm}^2\right),\ a_{17} = 1,\ a_{18} = 1,$$

$$a_{22} = \sum\sum\left(2\sigma^4\Delta x_{bm}^2 + 2\sigma^2\Delta x_{km}^2\right),$$

$$a_{23} = \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta y_{bm} + 2\sigma^2\Delta x_{km}\Delta y_{km}\right),$$

$$a_{24} = \sum\sum\left(\sigma^4\Delta x_{bm}^3 + 2\sigma^2\Delta x_{bm}\Delta x_{km}^2\right),$$

$$a_{25} = \sum\sum\left(2\sigma^4\Delta x_{bm}^2\Delta y_{bm} + 2d\sigma^2\Delta x_{km}\right),$$

$$a_{26} = \sum\sum\left(\sigma^4\Delta x_{bm}\Delta y_{bm}^2 + 2\sigma^2\Delta y_{bm}\Delta x_{km}\Delta y_{km}\right), \tag{A.9}$$

$$a_{27} = \Delta x_{b1},\ a_{28} = \Delta x_{b2},$$

$$a_{33} = \sum\sum\left(2\sigma^4\Delta y_{bm}^2 + 2\sigma^2\Delta y_{km}^2\right),$$

$$a_{34} = \sum\sum\left(\sigma^4\Delta x_{bm}^2\Delta y_{bm} + 2\sigma^2\Delta x_{bm}\Delta x_{km}\Delta y_{km}\right),$$

$$a_{35} = \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta y_{bm}^2 + 2d\sigma^2\Delta y_{km}\right),$$

$$a_{36} = \sum\sum\left(\sigma^4\Delta y_{bm}^3 + 2\sigma^2\Delta y_{bm}\Delta y_{km}^2\right),\ a_{37} = \Delta y_{b1},\ a_{38} = \Delta y_{b2},$$

$$a_{44} = \sum\sum\left(\frac{\sigma^4}{2}\Delta x_{bm}^4 + 2\sigma^2\Delta x_{bm}^2\Delta x_{km}^2 + 2\Delta x_{km}^4\right),$$

$$a_{45} = \sum\sum\left(\sigma^4\Delta x_{bm}^3\Delta y_{bm} + 2d\sigma^2\Delta x_{bm}\Delta x_{km} + 4\Delta x_{km}^3\Delta y_{km}\right),$$

$$a_{46} = \sum\sum\left(\frac{\sigma^4}{2}\Delta x_{bm}^2\Delta y_{bm}^2 + 2\sigma^2\Delta x_{bm}\Delta y_{bm}\Delta x_{km}\Delta y_{km} + 2\Delta x_{km}^2\Delta y_{km}^2\right),$$

$$
\begin{aligned}
&a_{47} = \frac{\Delta x_{b1}^2}{2},\ a_{48} = \frac{\Delta x_{b2}^2}{2},\\
&a_{55} = \sum\sum\left(2\sigma^4\Delta x_{bm}^2\Delta y_{bm}^2 + 2d^2\sigma^2 + 8\Delta x_{km}^2\Delta y_{km}^2\right),\\
&a_{56} = \sum\sum\left(\sigma^4\Delta x_{bm}\Delta y_{bm}^3 + 2d\sigma^2\Delta y_{bm}\Delta y_{km} + 4\Delta x_{km}\Delta y_{km}^3\right),\\
&a_{57} = \Delta x_{b1}\Delta y_{b1},\ a_{58} = \Delta x_{b2}\Delta y_{b2},\\
&a_{66} = \sum\sum\left(\frac{\sigma^4}{2}\Delta y_{bm}^4 + 2\sigma^2\Delta y_{bm}^2\Delta y_{km}^2 + 2\Delta y_{km}^4\right),\ a_{67} = \frac{\Delta y_{b1}^2}{2},\ a_{68} = \frac{\Delta y_{b2}^2}{2},\\
&a_{77} = a_{78} = a_{88} = 0,
\end{aligned}
$$

The components of the RHS vector C for the Dirichlet boundary condition case are given by:

$$
\begin{aligned}
c_1 &= \sum\sum\left(2\sigma^4\overline{\phi}_{bmk}\right),\\
c_2 &= \sum\sum\left(2\sigma^4\Delta x_{bm}\overline{\phi}_{bmk} + 2\sigma^2\Delta x_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right),\\
c_3 &= \sum\sum\left(2\sigma^4\Delta y_{bm}\overline{\phi}_{bmk} + 2\sigma^2\Delta y_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right),\\
c_4 &= \sum\sum\left(\sigma^4\Delta x_{bm}^2\overline{\phi}_{bmk} + 2\sigma^2\Delta x_{bm}\Delta x_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} +\right.\\
&\qquad\qquad \left. 2\Delta x_{km}^2\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right),\\
c_5 &= \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta y_m\overline{\phi}_{bmk} + 2d\sigma^2\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} +\right.\\
&\qquad\qquad \left. 4\Delta x_{km}\Delta y_{km}\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right),\\
c_6 &= \sum\sum\left(\sigma^4\Delta y_{bm}^2\overline{\phi}_{bmk} + 2\sigma^2\Delta y_{bm}\Delta y_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} +\right.\\
&\qquad\qquad \left. 2\Delta y_{km}^2\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right),\\
c_7 &= \phi_{b1},\\
c_8 &= \phi_{b2},
\end{aligned}
\tag{A.10}
$$

where $d = \Delta x_{bm}\Delta y_{km} + \Delta y_{bm}\Delta x_{km}$, the vector connecting the face centroid b and the face integration point b_m is defined as $\vec{r}_{bm} = \vec{X}_{b_m} - \vec{X}_b = \Delta x_{bm}\vec{i}_x + \Delta y_{bm}\vec{i}_y$, and the averaged values on the RHS are given by:

$$\begin{aligned} \overline{\phi}_{bmk} = \frac{1}{2}\Bigg[& \phi_b + \frac{\partial\phi}{\partial x}\bigg|_b \Delta x_{bm} + \frac{\partial\phi}{\partial y}\bigg|_b \Delta y_{bm} + \\ & \frac{1}{2}\left(\frac{\partial^2\phi}{\partial x^2}\bigg|_b \Delta x_{bm}^2 + 2\frac{\partial^2\phi}{\partial x\partial y}\bigg|_b \Delta x_{bm}\Delta y_{bm} + \frac{\partial^2\phi}{\partial y^2}\bigg|_b \Delta y_{bm}^2\right) + \\ & \phi_k + \frac{\partial\phi}{\partial x}\bigg|_k \Delta x_{km} + \frac{\partial\phi}{\partial y}\bigg|_k \Delta y_{km} + \\ & \frac{1}{2}\left(\frac{\partial^2\phi}{\partial x^2}\bigg|_k \Delta x_{km}^2 + 2\frac{\partial^2\phi}{\partial x\partial y}\bigg|_k \Delta x_{km}\Delta y_{km} + \frac{\partial^2\phi}{\partial y^2}\bigg|_k \Delta y_{km}^2\right)\Bigg]^{(current)}. \end{aligned} \tag{A.11}$$

$$\begin{aligned} \overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} = \frac{1}{2}\Bigg[& \left\{\begin{array}{c} \dfrac{\partial\phi}{\partial x}\bigg|_b + \dfrac{\partial^2\phi}{\partial x^2}\bigg|_b \Delta x_{bm} + \dfrac{\partial^2\phi}{\partial x\partial y}\bigg|_b \Delta y_{bm} \\ \dfrac{\partial\phi}{\partial y}\bigg|_b + \dfrac{\partial^2\phi}{\partial x\partial y}\bigg|_b \Delta x_{bm} + \dfrac{\partial^2\phi}{\partial y^2}\bigg|_b \Delta y_{bm} \end{array}\right\}\cdot\left\{\begin{array}{c} \Delta x_{km} \\ \\ \Delta y_{km} \end{array}\right\} + \\ & \left\{\begin{array}{c} \dfrac{\partial\phi}{\partial x}\bigg|_k + \dfrac{\partial^2\phi}{\partial x^2}\bigg|_k \Delta x_{km} + \dfrac{\partial^2\phi}{\partial x\partial y}\bigg|_k \Delta y_{km} \\ \dfrac{\partial\phi}{\partial y}\bigg|_k + \dfrac{\partial^2\phi}{\partial x\partial y}\bigg|_k \Delta x_{km} + \dfrac{\partial^2\phi}{\partial y^2}\bigg|_k \Delta y_{km} \end{array}\right\}\cdot\left\{\begin{array}{c} \Delta x_{km} \\ \\ \Delta y_{km} \end{array}\right\}\Bigg]^{(curr)}. \end{aligned} \tag{A.12}$$

$$\begin{aligned} \left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km} = \frac{1}{2}\Bigg(& \frac{\partial^2\phi}{\partial x^2}\bigg|_b \Delta x_{km}^2 + 2\frac{\partial^2\phi}{\partial x\partial y}\bigg|_b \Delta x_{km}\Delta y_{km} + \frac{\partial^2\phi}{\partial y^2}\bigg|_b \Delta y_{km}^2 + \\ & \frac{\partial^2\phi}{\partial x^2}\bigg|_k \Delta x_{km}^2 + 2\frac{\partial^2\phi}{\partial x\partial y}\bigg|_k \Delta x_{km}\Delta y_{km} + \frac{\partial^2\phi}{\partial y^2}\bigg|_k \Delta y_{km}^2\Bigg)^{(curr)}. \end{aligned} \tag{A.13}$$

For the Neumann boundary condition, the following changes must be performed to represent the new constraints: $c_7 = |\nabla\phi|_{b1}$ and $c_8 = |\nabla\phi|_{b2}$, and the 7th and 8th row/column must be changed for:

$$A(7:8,1:8) = \begin{bmatrix} 0 & n_x & n_y & \Delta x_{b1} n_x & \Delta y_{b1} n_x + \Delta x_{b1} n_y & \Delta y_{b1} n_y & 0 & 0 \\ 0 & n_x & n_y & \Delta x_{b2} n_x & \Delta y_{b2} n_x + \Delta x_{b2} n_y & \Delta y_{b1} n_y & 0 & 0 \end{bmatrix}. \quad \text{(A.14)}$$

The solution vector s is given by:

$$s = \begin{bmatrix} \phi_b & \left.\dfrac{\partial \phi}{\partial x}\right|_b & \left.\dfrac{\partial \phi}{\partial y}\right|_b & \left.\dfrac{\partial^2 \phi}{\partial x^2}\right|_b & \left.\dfrac{\partial^2 \phi}{\partial x \partial y}\right|_b & \left.\dfrac{\partial^2 \phi}{\partial y^2}\right|_b & \lambda_1 & \lambda_2 \end{bmatrix}^T. \quad \text{(A.15)}$$

APPENDIX B

3D OGRE THIRD ORDER SCHEME

B.1 3D Implementation of OGRE Third Order Scheme

The matrix of coefficients A is symmetric and is given by the block matrix:

$$A_{9x9} = \begin{bmatrix} C_{3\times3} & D_{3\times3} & E_{3\times3} \\ D^T_{3\times3} & O_{3\times3} & N_{3\times3} \\ E^T_{3\times3} & N^T_{3\times3} & L_{3\times3} \end{bmatrix}, \tag{B.1}$$

where

$$\begin{aligned}
c_{ik} &= \sum\sum\left(\sigma^4+\sigma^2\right)\Delta x_{jm}^{-1+|i-5/2|+|k-5/2|}\Delta y_{jm}^{2-|i-2|-|k-2|}\Delta z_{jm}^{-1+|i-3/2|+|k-3/2|}, \\
d_{ik} &= \sum\sum\left(\frac{\sigma^4}{2}+\sigma^2\right)\Delta x_{jm}^{-3/2+|i-5/2|+|2k-5|}\Delta y_{jm}^{3-|i-2|-|2k-4|}\Delta z_{jm}^{-3/2+|i-3/2|+|2k-3|}, \\
e_{ik} &= \sum\sum\left(\sigma^4+2\sigma^2\right)\Delta x_{jm}^{-3/2+|i-5/2|+|2k-5|}\Delta y_{jm}^{3-|i-2|-|2k-4|}\Delta z_{jm}^{-3/2+|i-3/2|+|2k-3|}, \\
o_{ik} &= \sum\sum\left(\frac{\sigma^4}{4}+\sigma^2+1\right)\Delta x_{jm}^{-2+|2i-5|+|2k-5|}\Delta y_{jm}^{4-|2i-4|-|2k-4|}\Delta z_{jm}^{-2+|2i-3|+|2k-3|}, \\
n_{ik} &= \sum\sum\left(\frac{\sigma^4}{2}+2\sigma^2+2\right)\Delta x_{jm}^{1/2+|2i-5|-|k-3/2|}\Delta y_{jm}^{2-|2i-4|+|k-2|}\Delta z_{jm}^{1/2+|2i-3|-|k-5/2|}, \\
l_{ik} &= \sum\sum\left(\sigma^4+4\sigma^2+4\right)\Delta x_{jm}^{3-|i-3/2|-|k-3/2|}\Delta y_{jm}^{|i-2|+|k-2|}\Delta z_{jm}^{3-|i-5/2|-|k-5/2|},
\end{aligned} \tag{B.2}$$

and it is used the following simplification: $\sum\sum = \sum_{j=1}^{\#neig}\sum_{m=1}^{3}$. The components of the RHS vector B, containing just current values, are given by the block matrix:

$$B_{9x1} = \begin{Bmatrix} P_{3\times1} \\ Q_{3\times1} \\ W_{3\times1} \end{Bmatrix}, \tag{B.3}$$

where

$$p_{i1} = \sum\sum\Delta x_{i,jm}\left[\sigma^4\left(\tilde{F}_{jm}-\phi_o\right)+\sigma^2\tilde{G}_{jm}\right], \tag{B.4}$$

$$q_{i1} = \sum\sum\Delta x_{i,jm}^2\left[\frac{\sigma^4}{2}\left(\tilde{F}_{jm}-\phi_o\right)+\sigma^2\tilde{G}_{jm}+\tilde{H}_{jm}\right], \tag{B.5}$$

and

$$W_{3x1} = \left\{ \begin{array}{c} \sum\sum \Delta x_{jm}\Delta y_{jm}\left[\sigma^4\left(\tilde{F}_{jm}-\phi_o\right)+2\sigma^2\tilde{G}_{jm}+2\tilde{H}_{jm}\right] \\ \sum\sum \Delta x_{jm}\Delta z_{jm}\left[\sigma^4\left(\tilde{F}_{jm}-\phi_o\right)+2\sigma^2\tilde{G}_{jm}+2\tilde{H}_{jm}\right] \\ \sum\sum \Delta y_{jm}\Delta z_{jm}\left[\sigma^4\left(\tilde{F}_{jm}-\phi_o\right)+2\sigma^2\tilde{G}_{jm}+2\tilde{H}_{jm}\right] \end{array} \right\}, \tag{B.6}$$

and where the averaged values $\tilde{F}_{jm}$, $\tilde{G}_{jm}$ and $\tilde{H}_{jm}$, are given by (2.57), (2.59) and (2.61), respectively. The solution vector s with the values of the derivatives at the center cell centroid is given by:

$$s = \left[\begin{array}{ccccccccc} \left.\frac{\partial\phi}{\partial x}\right|_o & \left.\frac{\partial\phi}{\partial y}\right|_o & \left.\frac{\partial\phi}{\partial z}\right|_o & \left.\frac{\partial^2\phi}{\partial x^2}\right|_o & \left.\frac{\partial^2\phi}{\partial y^2}\right|_o & \left.\frac{\partial^2\phi}{\partial z^2}\right|_o & \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_o & \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_o & \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_o \end{array} \right]^T. \tag{B.7}$$

B.2 3D BC Implementation of OGRE Third Order Scheme

It is used the following simplification: $\sum\sum = \sum_{k=0}^{1,\,2\ or\ 3}\sum_{m=1}^{3}$. Since the matrix of coefficients A is symmetric, just the upper diagonal components, for the inlet boundary condition case, are given here:

$$a_{11} = \sum\sum\left(2\sigma^4\right),\ a_{12} = \sum\sum\left(2\sigma^4\Delta x_{bm}\right),\ a_{13} = \sum\sum\left(2\sigma^4\Delta y_{bm}\right),$$

$$a_{14} = \sum\sum\left(2\sigma^4\Delta z_{bm}\right),\ a_{15} = \sum\sum\left(\sigma^4\Delta x_{bm}^2\right),\ a_{16} = \sum\sum\left(\sigma^4\Delta y_{bm}^2\right),$$

$$a_{17} = \sum\sum\left(\sigma^4\Delta z_{bm}^2\right),\ a_{18} = \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta y_{bm}\right),$$

$$a_{19} = \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta z_{bm}\right),\ a_{1,10} = \sum\sum\left(2\sigma^4\Delta y_{bm}\Delta z_{bm}\right),$$

$$a_{1,11} = 1,\ a_{1,12} = 1,\ a_{1,13} = 1,$$

$$a_{22} = \sum\sum\left(2\sigma^4\Delta x_{bm}^2 + 2\sigma^2\Delta x_{km}^2\right),$$

$$
\begin{aligned}
&a_{23} = \sum\sum \left(2\sigma^4 \Delta x_{bm} \Delta y_{bm} + 2\sigma^2 \Delta x_{km} \Delta y_{km}\right),\\
&a_{24} = \sum\sum \left(2\sigma^4 \Delta x_{bm} \Delta z_{bm} + 2\sigma^2 \Delta x_{km} \Delta z_{km}\right),\\
&a_{25} = \sum\sum \left(\sigma^4 \Delta x_{bm}^3 + 2\sigma^2 \Delta x_{bm} \Delta x_{km}^2\right),\\
&a_{26} = \sum\sum \left(\sigma^4 \Delta x_{bm} \Delta y_{bm}^2 + 2\sigma^2 \Delta y_{bm} \Delta x_{km} \Delta y_{km}\right),\\
&a_{27} = \sum\sum \left(\sigma^4 \Delta x_{bm} \Delta z_{bm}^2 + 2\sigma^2 \Delta z_{bm} \Delta x_{km} \Delta z_{km}\right),\\
&a_{28} = \sum\sum \left(2\sigma^4 \Delta x_{bm}^2 \Delta y_{bm} + 2\sigma^2 d_1 \Delta x_{km}\right),\\
&a_{29} = \sum\sum \left(2\sigma^4 \Delta x_{bm}^2 \Delta z_{bm} + 2\sigma^2 d_2 \Delta x_{km}\right),\\
&a_{2,10} = \sum\sum \left(2\sigma^4 \Delta x_{bm} \Delta y_{bm} \Delta z_{bm} + 2\sigma^2 d_3 \Delta x_{km}\right),\\
&a_{2,11} = \Delta x_{b1},\ a_{2,12} = \Delta x_{b2},\ a_{2,13} = \Delta x_{b3},\\
&a_{33} = \sum\sum \left(2\sigma^4 \Delta y_{bm}^2 + 2\sigma^2 \Delta y_{km}^2\right),\\
&a_{34} = \sum\sum \left(2\sigma^4 \Delta y_{bm} \Delta z_{bm} + 2\sigma^2 \Delta y_{km} \Delta z_{km}\right),\\
&a_{35} = \sum\sum \left(\sigma^4 \Delta x_{bm}^2 \Delta y_{bm} + 2\sigma^2 \Delta x_{bm} \Delta x_{km} \Delta y_{km}\right),\\
&a_{36} = \sum\sum \left(\sigma^4 \Delta y_{bm}^3 + 2\sigma^2 \Delta y_{bm} \Delta y_{km}^2\right),\\
&a_{37} = \sum\sum \left(\sigma^4 \Delta y_{bm} \Delta z_{bm}^2 + 2\sigma^2 \Delta z_{bm} \Delta y_{km} \Delta z_{km}\right),\\
&a_{38} = \sum\sum \left(2\sigma^4 \Delta x_{bm} \Delta y_{bm}^2 + 2\sigma^2 d_1 \Delta y_{km}\right),\\
&a_{39} = \sum\sum \left(2\sigma^4 \Delta x_{bm} \Delta y_{bm} \Delta z_{bm} + 2\sigma^2 d_2 \Delta y_{km}\right),\\
&a_{3,10} = \sum\sum \left(2\sigma^4 \Delta y_{bm}^2 \Delta z_{bm} + 2\sigma^2 d_3 \Delta y_{km}\right),\\
&a_{3,11} = \Delta y_{b1},\ a_{3,12} = \Delta y_{b2},\ a_{3,13} = \Delta y_{b3},\\
&a_{44} = \sum\sum \left(2\sigma^4 \Delta z_{bm}^2 + 2\sigma^2 \Delta z_{km}^2\right),\\
&a_{45} = \sum\sum \left(\sigma^4 \Delta x_{bm}^2 \Delta z_{bm} + 2\sigma^2 \Delta x_{bm} \Delta x_{km} \Delta z_{km}\right),
\end{aligned}
\tag{B.8}
$$

$$a_{46} = \sum\sum\left(\sigma^4 \Delta y_{bm}^2 \Delta z_{bm} + 2\sigma^2 \Delta y_{bm} \Delta y_{km} \Delta z_{km}\right),$$

$$a_{47} = \sum\sum\left(\sigma^4 \Delta z_{bm}^3 + 2\sigma^2 \Delta z_{bm} \Delta z_{km}^2\right),$$

$$a_{48} = \sum\sum\left(2\sigma^4 \Delta x_{bm} \Delta y_{bm} \Delta z_{bm} + 2\sigma^2 d_1 \Delta z_{km}\right),$$

$$a_{49} = \sum\sum\left(2\sigma^4 \Delta x_{bm} \Delta z_{bm}^2 + 2\sigma^2 d_2 \Delta z_{km}\right),$$

$$a_{4,10} = \sum\sum\left(2\sigma^4 \Delta y_{bm} \Delta z_{bm}^2 + 2\sigma^2 d_3 \Delta z_{km}\right),$$

$$a_{4,11} = \Delta z_{b1},\ a_{4,12} = \Delta z_{b2},\ a_{4,13} = \Delta z_{b3},$$

$$a_{55} = \sum\sum\left(\frac{\sigma^4}{2} \Delta x_{bm}^4 + 2\sigma^2 \Delta x_{bm}^2 \Delta x_{km}^2 + 2\Delta x_{km}^4\right),$$

$$a_{56} = \sum\sum\left(\frac{\sigma^4}{2} \Delta x_{bm}^2 \Delta y_{bm}^2 + 2\sigma^2 \Delta x_{bm} \Delta y_{bm} \Delta x_{km} \Delta y_{km} + 2\Delta x_{km}^2 \Delta y_{km}^2\right),$$

$$a_{57} = \sum\sum\left(\frac{\sigma^4}{2} \Delta x_{bm}^2 \Delta z_{bm}^2 + 2\sigma^2 \Delta x_{bm} \Delta z_{bm} \Delta x_{km} \Delta z_{km} + 2\Delta x_{km}^2 \Delta z_{km}^2\right),$$

$$a_{58} = \sum\sum\left(\sigma^4 \Delta x_{bm}^3 \Delta y_{bm} + 2\sigma^2 d_1 \Delta x_{bm} \Delta x_{km} + 4\Delta x_{km}^3 \Delta y_{km}\right),$$

$$a_{59} = \sum\sum\left(\sigma^4 \Delta x_{bm}^3 \Delta z_{bm} + 2\sigma^2 d_2 \Delta x_{bm} \Delta x_{km} + 4\Delta x_{km}^3 \Delta z_{km}\right),$$

$$a_{5,10} = \sum\sum\left(\sigma^4 \Delta x_{bm}^2 \Delta y_{bm} \Delta z_{bm} + 2\sigma^2 d_3 \Delta x_{bm} \Delta x_{km} + 4\Delta x_{km}^2 \Delta y_{km} \Delta z_{km}\right),$$

$$a_{5,11} = \frac{\Delta x_{b1}^2}{2},\ a_{5,12} = \frac{\Delta x_{b2}^2}{2},\ a_{5,13} = \frac{\Delta x_{b3}^2}{2},$$

$$a_{66} = \sum\sum\left(\frac{\sigma^4}{2} \Delta y_{bm}^4 + 2\sigma^2 \Delta y_{bm}^2 \Delta y_{km}^2 + 2\Delta y_{km}^4\right),$$

$$a_{67} = \sum\sum\left(\frac{\sigma^4}{2} \Delta y_{bm}^2 \Delta z_{bm}^2 + 2\sigma^2 \Delta y_{bm} \Delta z_{bm} \Delta y_{km} \Delta z_{km} + 2\Delta y_{km}^2 \Delta z_{km}^2\right),$$

$$a_{68} = \sum\sum\left(\sigma^4 \Delta x_{bm} \Delta y_{bm}^3 + 2\sigma^2 d_1 \Delta y_{bm} \Delta y_{km} + 4\Delta x_{km} \Delta y_{km}^3\right),$$

$$a_{69} = \sum\sum\left(\sigma^4 \Delta x_{bm} \Delta y_{bm}^2 \Delta z_{bm} + 2\sigma^2 d_2 \Delta y_{bm} \Delta y_{km} + 4\Delta x_{km} \Delta y_{km}^2 \Delta z_{km}\right),$$

$$a_{6,10} = \sum\sum\left(\sigma^4 \Delta y_{bm}^3 \Delta z_{bm} + 2\sigma^2 d_3 \Delta y_{bm} \Delta y_{km} + 4\Delta y_{km}^3 \Delta z_{km}\right),$$

$$a_{6,11} = \frac{\Delta y_{b1}^2}{2},\ a_{6,12} = \frac{\Delta y_{b2}^2}{2},\ a_{6,13} = \frac{\Delta y_{b3}^2}{2},$$

$$a_{77} = \sum\sum \left(\frac{\sigma^4}{2}\Delta z_{bm}^4 + 2\sigma^2 \Delta z_{bm}^2 \Delta z_{km}^2 + 2\Delta z_{km}^4 \right),$$

$$a_{78} = \sum\sum \left(\sigma^4 \Delta x_{bm} \Delta y_{bm} \Delta z_{bm}^2 + 2\sigma^2 d_1 \Delta z_{bm} \Delta z_{km} + 4\Delta x_{km} \Delta y_{km} \Delta z_{km}^2 \right),$$

$$a_{79} = \sum\sum \left(\sigma^4 \Delta x_{bm} \Delta z_{bm}^3 + 2\sigma^2 d_2 \Delta z_{bm} \Delta z_{km} + 4\Delta x_{km} \Delta z_{km}^3 \right),$$

$$a_{7,10} = \sum\sum \left(\sigma^4 \Delta y_{bm} \Delta z_{bm}^3 + 2\sigma^2 d_3 \Delta z_{bm} \Delta z_{km} + 4\Delta y_{km} \Delta z_{km}^3 \right),$$

$$a_{7,11} = \frac{\Delta z_{b1}^2}{2},\ a_{7,12} = \frac{\Delta z_{b2}^2}{2},\ a_{7,13} = \frac{\Delta z_{b3}^2}{2},$$

$$a_{88} = \sum\sum \left(2\sigma^4 \Delta x_{bm}^2 \Delta y_{bm}^2 + 2d_1^2\sigma^2 + 8\Delta x_{km}^2 \Delta y_{km}^2 \right),$$

$$a_{89} = \sum\sum \left(2\sigma^4 \Delta x_{bm}^2 \Delta y_{bm} \Delta z_{bm} + 2d_1 d_2 \sigma^2 + 8\Delta x_{km}^2 \Delta y_{km} \Delta z_{km} \right),$$

$$a_{8,10} = \sum\sum \left(2\sigma^4 \Delta x_{bm} \Delta y_{bm}^2 \Delta z_{bm} + 2d_1 d_3 \sigma^2 + 8\Delta x_{km} \Delta y_{km}^2 \Delta z_{km} \right),$$

$$a_{8,11} = \Delta x_{b1} \Delta y_{b1},\ a_{8,12} = \Delta x_{b2} \Delta y_{b2},\ a_{8,13} = \Delta x_{b3} \Delta y_{b3},$$

$$a_{99} = \sum\sum \left(2\sigma^4 \Delta x_{bm}^2 \Delta z_{bm}^2 + 2d_2^2\sigma^2 + 8\Delta x_{km}^2 \Delta z_{km}^2 \right),$$

$$a_{9,10} = \sum\sum \left(2\sigma^4 \Delta x_{bm} \Delta y_{bm} \Delta z_{bm}^2 + 2d_2 d_3 \sigma^2 + 8\Delta x_{km} \Delta y_{km} \Delta z_{km}^2 \right),$$

$$a_{9,11} = \Delta x_{b1} \Delta z_{b1},\ a_{9,12} = \Delta x_{b2} \Delta z_{b2},\ a_{9,13} = \Delta x_{b3} \Delta z_{b3},$$

$$a_{10,10} = \sum\sum \left(2\sigma^4 \Delta y_{bm}^2 \Delta z_{bm}^2 + 2d_3^2\sigma^2 + 8\Delta y_{km}^2 \Delta z_{km}^2 \right),$$

$$a_{10,11} = \Delta y_{b1} \Delta z_{b1},\ a_{10,12} = \Delta y_{b2} \Delta z_{b2},\ a_{10,13} = \Delta y_{b3} \Delta z_{b3},$$

$$a_{11,11} = a_{11,12} = a_{11,13} = a_{12,12} = a_{12,13} = a_{13,13} = 0.$$

The components of the RHS vector C for the Dirichlet boundary condition case are given by:

$$c_1 = \sum\sum \left(2\sigma^4 \overline{\phi}_{bmk} \right),$$

$$c_2 = \sum\sum \left(2\sigma^4 \Delta x_{bm} \overline{\phi}_{bmk} + 2\sigma^2 \Delta x_{km} \overline{\nabla\phi}_{bmk} \cdot \vec{r}_{km} \right),$$

$$
\begin{aligned}
c_3 &= \sum\sum\left(2\sigma^4\Delta y_{bm}\overline{\phi}_{bmk} + 2\sigma^2\Delta y_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right), \\
c_4 &= \sum\sum\left(2\sigma^4\Delta z_{bm}\overline{\phi}_{bmk} + 2\sigma^2\Delta z_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right), \\
c_5 &= \sum\sum\left(\sigma^4\Delta x_{bm}^2\overline{\phi}_{bmk} + 2\sigma^2\Delta x_{bm}\Delta x_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} + \right. \\
&\qquad\left. 2\Delta x_{km}^2\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right), \\
c_6 &= \sum\sum\left(\sigma^4\Delta y_{bm}^2\overline{\phi}_{bmk} + 2\sigma^2\Delta y_{bm}\Delta y_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} + \right. \\
&\qquad\left. 2\Delta y_{km}^2\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right), \\
c_7 &= \sum\sum\left(\sigma^4\Delta z_{bm}^2\overline{\phi}_{bmk} + 2\sigma^2\Delta z_{bm}\Delta z_{km}\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} + \right. \\
&\qquad\left. 2\Delta z_{km}^2\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right), \\
c_8 &= \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta y_m\overline{\phi}_{bmk} + 2\sigma^2 d_1\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} + \right. \\
&\qquad\left. 4\Delta x_{km}\Delta y_{km}\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right), \\
c_9 &= \sum\sum\left(2\sigma^4\Delta x_{bm}\Delta z_m\overline{\phi}_{bmk} + 2\sigma^2 d_2\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} + \right. \\
&\qquad\left. 4\Delta x_{km}\Delta z_{km}\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right), \\
c_{10} &= \sum\sum\left(2\sigma^4\Delta y_{bm}\Delta z_m\overline{\phi}_{bmk} + 2\sigma^2 d_3\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} + \right. \\
&\qquad\left. 4\Delta y_{km}\Delta z_{km}\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km}\right), \\
c_{11} &= \phi_{b1}, \\
c_{12} &= \phi_{b2}, \\
c_{13} &= \phi_{b3},
\end{aligned}
\tag{B.9}
$$

where $d_1 = \Delta x_{bm}\Delta y_{km} + \Delta y_{bm}\Delta x_{km}$, $d_2 = \Delta x_{bm}\Delta z_{km} + \Delta z_{bm}\Delta x_{km}$ and $d_3 = \Delta y_{bm}\Delta z_{km} + \Delta z_{bm}\Delta y_{km}$. The vector connecting the face centroid b and the face integration point b_m is defined as $\vec{r}_{bm} = \vec{X}_{b_m} - \vec{X}_b = \Delta x_{bm}\vec{i}_x + \Delta y_{bm}\vec{i}_y + \Delta z_{bm}\vec{i}_z$, and the averaged values on the RHS vector are given by:

$$
\begin{aligned}
\overline{\phi}_{bmk} = \frac{1}{2}\Bigg[& \phi_b + \left.\frac{\partial\phi}{\partial x}\right|_b \Delta x_{bm} + \left.\frac{\partial\phi}{\partial y}\right|_b \Delta y_{bm} + \left.\frac{\partial\phi}{\partial z}\right|_b \Delta z_{bm} + \\
& \frac{1}{2}\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_b \Delta x_{bm}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_b \Delta y_{bm}^2 + \left.\frac{\partial^2\phi}{\partial z^2}\right|_b \Delta z_{bm}^2 \right) + \\
& \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_b \Delta x_{bm}\Delta y_{bm} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_b \Delta x_{bm}\Delta z_{bm} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_b \Delta y_{bm}\Delta z_{bm} + \\
& \phi_k + \left.\frac{\partial\phi}{\partial x}\right|_k \Delta x_{km} + \left.\frac{\partial\phi}{\partial y}\right|_k \Delta y_{km} + \left.\frac{\partial\phi}{\partial z}\right|_k \Delta z_{km} + \\
& \frac{1}{2}\left(\left.\frac{\partial^2\phi}{\partial x^2}\right|_k \Delta x_{km}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_k \Delta y_{km}^2 + \left.\frac{\partial^2\phi}{\partial z^2}\right|_k \Delta z_{km}^2 \right) + \\
& \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_k \Delta x_{km}\Delta y_{km} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_k \Delta x_{km}\Delta z_{km} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_k \Delta y_{km}\Delta z_{km} \Bigg]^{(curr)} .
\end{aligned}
\tag{B.10}
$$

$$
\begin{aligned}
\overline{\nabla\phi}_{bmk}\cdot\vec{r}_{km} = \frac{1}{2}\Bigg[& \begin{Bmatrix}
\left.\frac{\partial\phi}{\partial x}\right|_b + \left.\frac{\partial^2\phi}{\partial x^2}\right|_b \Delta x_{bm} + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_b \Delta y_{bm} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_b \Delta z_{bm} \\
\left.\frac{\partial\phi}{\partial y}\right|_b + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_b \Delta x_{bm} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_b \Delta y_{bm} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_b \Delta z_{bm} \\
\left.\frac{\partial\phi}{\partial z}\right|_b + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_b \Delta x_{bm} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_b \Delta y_{bm} + \left.\frac{\partial^2\phi}{\partial z^2}\right|_b \Delta z_{bm}
\end{Bmatrix} \cdot \\
& \begin{Bmatrix} \Delta x_{km} \\ \Delta y_{km} \\ \Delta z_{km} \end{Bmatrix} + \\
& \begin{Bmatrix}
\left.\frac{\partial\phi}{\partial x}\right|_k + \left.\frac{\partial^2\phi}{\partial x^2}\right|_k \Delta x_{km} + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_k \Delta y_{km} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_k \Delta z_{km} \\
\left.\frac{\partial\phi}{\partial y}\right|_k + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_k \Delta x_{km} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_k \Delta y_{km} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_k \Delta z_{km} \\
\left.\frac{\partial\phi}{\partial z}\right|_k + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_k \Delta x_{km} + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_k \Delta y_{km} + \left.\frac{\partial^2\phi}{\partial z^2}\right|_k \Delta z_{km}
\end{Bmatrix} \cdot
\end{aligned}
$$

$$\left.\left.\begin{Bmatrix}\Delta x_{km}\\[1em] \Delta y_{km}\\[1em] \Delta z_{km}\end{Bmatrix}\right]\right]^{(curr)} \quad . \tag{B.11}$$

$$\begin{aligned}\left(\overline{\nabla\nabla\phi}_{bmk}\cdot\vec{r}_{km}\right)\cdot\vec{r}_{km} = \frac{1}{2}\Bigg(& \left.\frac{\partial^2\phi}{\partial x^2}\right|_b \Delta x_{km}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_b \Delta y_{km}^2 + \left.\frac{\partial^2\phi}{\partial z^2}\right|_b \Delta z_{km}^2 + \\ & \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_b \Delta x_{km}\Delta y_{km} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_b \Delta x_{km}\Delta z_{km} + \\ & \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_b \Delta y_{km}\Delta z_{km} + \\ & \left.\frac{\partial^2\phi}{\partial x^2}\right|_k \Delta x_{km}^2 + \left.\frac{\partial^2\phi}{\partial y^2}\right|_k \Delta y_{km}^2 + \left.\frac{\partial^2\phi}{\partial z^2}\right|_k \Delta z_{km}^2 + \\ & \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_k \Delta x_{km}\Delta y_{km} + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_k \Delta x_{km}\Delta z_{km} + \\ & \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_k \Delta y_{km}\Delta z_{km}\Bigg)^{(cur.)} \quad . \end{aligned} \tag{B.12}$$

For the Neumann boundary condition, the following changes must be performed to represent the new constraints: $c_{11} = |\nabla\phi|_{b1}$, $c_{12} = |\nabla\phi|_{b2}$ and $c_{13} = |\nabla\phi|_{b3}$, and the 11th, 12th and 13th rows/columns must be changed for:

$$A(11:13,1:10) = \left[\begin{matrix} 0 & n_x & n_y & n_z & \Delta x_{b1}n_x & \Delta y_{b1}n_y & \Delta z_{b1}n_z \\ 0 & n_x & n_y & n_z & \Delta x_{b2}n_x & \Delta y_{b2}n_y & \Delta z_{b2}n_z & \cdots \\ 0 & n_x & n_y & n_z & \Delta x_{b3}n_x & \Delta y_{b3}n_y & \Delta z_{b3}n_z \end{matrix}\right.$$

$$\left.\begin{matrix} & \Delta y_{b1}n_x + \Delta x_{b1}n_y & \Delta z_{b1}n_x + \Delta x_{b1}n_z & \Delta z_{b1}n_y + \Delta y_{b1}n_z \\ \cdots & \Delta y_{b2}n_x + \Delta x_{b2}n_y & \Delta z_{b2}n_x + \Delta x_{b2}n_z & \Delta z_{b2}n_y + \Delta y_{b2}n_z \\ & \Delta y_{b3}n_x + \Delta x_{b3}n_y & \Delta z_{b3}n_x + \Delta x_{b3}n_z & \Delta z_{b3}n_y + \Delta y_{b3}n_z \end{matrix}\right] . \tag{B.13}$$

The solution vector s is given by:

$$s = \Bigg[\; \phi_b \quad \frac{\partial \phi}{\partial x}\bigg|_b \quad \frac{\partial \phi}{\partial y}\bigg|_b \quad \frac{\partial \phi}{\partial z}\bigg|_b \quad \frac{\partial^2 \phi}{\partial x^2}\bigg|_b \quad \frac{\partial^2 \phi}{\partial y^2}\bigg|_b \quad \frac{\partial^2 \phi}{\partial z^2}\bigg|_b \quad ...$$

$$... \quad \frac{\partial^2 \phi}{\partial x \partial y}\bigg|_b \quad \frac{\partial^2 \phi}{\partial x \partial z}\bigg|_b \quad \frac{\partial^2 \phi}{\partial y \partial z}\bigg|_b \quad \lambda_1 \quad \lambda_2 \quad \lambda_3 \; \Bigg]^T . \tag{B.14}$$