Unstructured mesh methods for stratified turbulent flows

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: [https://dspace.lboro.ac.uk/2134/10617](https://dspace.lboro.ac.uk/2134/10617)

Publisher: © Zhao Zhang

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: [https://creativecommons.org/licenses/by-nc-nd/4.0/](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Please cite the published version.
Unstructured Mesh Methods for 
Stratified Turbulent Flows

by

Zhao Zhang

under the supervision of

Dr Joanna Szmelter

Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy of Loughborough University

© by Zhao Zhang 2014

November 26, 2014
Acknowledgement

I would like to thank Dr Joanna Szmelter for all her generous instructions, suggestions and help throughout my PhD. I am fortunate to have her as my supervisor.

I would also like to thank Dr Piotr Smolarkiewicz for his help and most constructive professional guidance aiding my research.
Abstract

Abstract: Developments are reported of unstructured-mesh methods for simulating stratified, turbulent and shear flows. The numerical model employs nonoscillatory forward in-time integrators for anelastic and incompressible flow PDEs, built on Multidimensional Positive Definite Advection Transport Algorithm (MPDATA) and a preconditioned conjugate residual elliptic solver. Finite-volume spatial discretisation adopts an edge-based data structure. Tetrahedral-based and hybrid-based median-dual options for unstructured meshes are developed, enabling flexible spatial resolution. Viscous laminar and detached eddy simulation (DES) flow solvers are developed based on the edge-based NFT MPDATA scheme. The built-in implicit large eddy simulation (ILES) capability of the NFT scheme is also employed and extended to fully unstructured tetrahedral and hybrid meshes. Challenging atmospheric and engineering problems are solved numerically to validate the model and to demonstrate its applications. The numerical problems include simulations of stratified, turbulent and shear flows past obstacles involving complex gravity-wave phenomena in the lee, critical-level laminar-turbulence transitioning and various vortex structures in the wake. Qualitative flow patterns and quantitative data analysis are both presented in the current study.
Contents

1 Introduction 1

2 Governing Equations 6
2.1 Stratified Flows 6
2.2 RANS and Spalart-Allmaras Model 7
2.2.1 RANS for Anelastic Equations 7
2.2.2 Spalart-Allmaras One Equation Model 8
2.3 LES and Dynamic Smagorinsky Model 10
2.3.1 LES for Anelastic Equations 10
2.3.2 Dynamic Smagorinsky Model 11
2.4 DES 13
2.5 ILES 15
2.6 The Function of Absorbers 16

3 Numerical Scheme 19
3.1 Edge Based Finite Volume Spatial Discretisation 19
3.2 MPDATA 20
3.3 Computation of Derivatives 24
3.4 Non-oscillatory Option 25
3.5 NFT Flow Solver 26
3.6 Elliptic Solver 27
3.7 Aspects of Mesh Generation 29

4 Strongly Stratified Flow Past a Steep Isolated Hill 39

5 Stratified Turbulent Shear Flows with Critical Levels 45

6 Stratified Laminar Flows Past a Sphere 59
6.1 The Numerical Experiment 59
6.2 Results 62
1 Introduction

Unstructured-meshes and mesh adaptivity offer flexibility that can provide optimised variable mesh resolution required for improved representation of complex physical processes in stratified turbulent flows. Generally, such flows evince the multiplicity of scales ranging from a fraction of a millimetre where dissipation occurs, to tens of thousands of kilometres where planetary weather and climate take place. For some atmospheric processes, that are still insufficiently understood in spite of their relevance to weather conditions in populated areas, traditional structured mesh resolution may limit realizability, and thus cognition and predictability, attainable with available computational resources. Examples include weather in long winding valleys and mountainous areas, onset and evolution of radiation fog or stratiform clouds, precipitation or extreme events forecasting.

On the other hand, unstructured meshes have become established in the simulation of engineering flows, for example, in aerodynamics of automotive and aeronautical applications, in engine flow simulation, in reservoir modelling, in medical device design, to mention just a few. The popularity of unstructured meshes is primarily due to their ability to reflect complex geometries. They also allow for easy implementation of variable mesh resolution and mesh manipulation techniques.

The current work extends earlier developments aiming at generalisation of the three-dimensional non-oscillatory forward-in-time (NFT) integrators to unstructured meshes. The NFT integrators were first proposed for finite-difference atmospheric models [63], based on the multidimensional positive definite advection transport algorithm (MPDATA). Since its invention in the 1980s by Smolarkiewicz [58, 59], finite-difference MPDATA has gained recognition in the field of atmospheric research. Based on these, a numerical solver for all-scale geophysical flows, the Eulerian-Lagrangian code EULAG, was developed and expanded to a wide range of multiscale multiphysics applications, cf. [53] for a review. Developments of the NFT integrators for unstructured meshes commenced much later. Their roots are in
the development of the edge-based finite-volume MPDATA [64, 65]. Developments subsequent to the finite-volume MPDATA included unstructured mesh finite volume NFT framework for modelling all-speed engineering flows [76, 69], models for simulating idealised hydrostatic dynamics of the planetary atmosphere [77], reduced 2D soundproof models for simulation of nonhydrostatic gravity-wave dynamics [78, 70] and their consequent generalisation to 3D mesoscale modelling of nonhydrostatic dynamics [71]. The current work in Section 4 stems from the latter development and extends its cognitive capabilities by enabling tetrahedral meshing.

The underlying concept of the numerical model used in the current study is presented in detail in [71]. In the model, all dependent variables are co-located, benefiting memory and communication requirements compared to staggered arrangements. This also facilitates implicit representation of buoyant modes. Generally, NFT labels a class of second-order-accurate two-time-level schemes (of the Cauchy-Kowalewski type; cf. section 19 in [81]) for integrating fluid PDEs that are built on nonlinear advection schemes such as MPDATA. The MPDATA schemes control numerical oscillations in the sense of high-resolution methods [54], and this assures the nonlinear stability for the co-located arrangement of dependent variables (cf. appendix A1 in [65]). Furthermore, in the NFT solver, MPDATA provides implicit turbulence modelling capability to the full set of equations. The implicit LES (ILES) properties of MPDATA-based high-Reynolds-number solvers were analysed in detail in the context of structured grids [42, 43, 6, 45, 84]. However, in the context of unstructured meshes, they were addressed only recently [78, 70, 71]. For unstructured meshes ILES capability is especially important as it obviates the need to evaluate viscous stress, which can be a cumbersome task for irregular unstructured meshes with potential issues for stability, smoothness and complexity.

Preceding works on the nonhydrostatic unstructured-mesh NFT models [78, 70, 71] verified the excellent accuracy of the finite-volume approach, using benchmarks from both analytic and laboratory results, and comparing unstructured-mesh results
to the corresponding results obtained with the finite-difference NFT model [53]. The previously studied physical problems representative of mesoscale dynamics included nonhydrostatic mountain waves at weak and strong background stratification (with linear and nonlinear flow responses, respectively), amplification and braking of deep stratospheric gravity waves, low Froude number flows past steep isolated 3D mountain, and evolution of convective planetary boundary layer. Quantitative analysis of the results demonstrated suitability of the approach for accurate simulation of gravity-wave phenomena and thermal convection — two important ingredients of atmospheric mesoscale dynamics. Moreover, a NFT MPDATA unstructured-mesh approach was developed for shallow water equations and was further extended to three-dimensional hydrostatic flows on a rotating sphere [77, 78].

For engineering flows, Manickam [41] applied NFT MPDATA to 2D lid-driven cavity flow for a selection of Reynolds numbers on a Cartesian mesh, and to 2D laminar and turbulent flows past a circular cylinder on an unstructured hybrid mesh. He also developed a 3D incompressible laminar flow solver and demonstrated its performance on a flow past a sphere for a range of Reynolds numbers. The original development of a MPDATA-based solver for high-speed compressible flows was introduced by Smolarkiewicz and Szmelter in [69, 67].

This thesis reports the development of the NFT MPDATA scheme for simulating 3D stratified, turbulent and shear flows on fully unstructured meshes. A dual-mesh generator designed for flexible fully unstructured tetrahedral, prismatic and hybrid primary meshes is developed, enabling a wide range of mesh refinement strategies. The dual mesh construction uses the edge-based data structure and can be applied to arbitrary unstructured meshes. We revisit 3D low Froude number flows past a steep isolated hill to verify the accuracy of tetrahedral discretisation.

We also introduce to the unstructured-mesh NFT modelling the problem of gravity-wave critical-level interaction. This problem is essentially stiff, as the vertical wavelength of a vertically propagating wave diminishes to zero upon approaching
the critical level, so processes acting on small scales become important and even an infinitesimal amplitude wave becomes nonlinear. The problem exposes benefits of unstructured meshes particularly well as it calls for a high degree of refinement in a small portion of the computational domain.

Stratified viscous laminar flows past a sphere are simulated next on fully unstructured meshes, to validate the development and implementation of viscous effects, as well as to further study with flexible resolution offered by unstructured meshes. The computations are conducted on dual meshes of unstructured primary hybrid meshes with tetrahedral and prismatic elements. Comparisons with previous structured-grid results and experimental data validate the accuracy of the NFT MPDATA scheme for stratified viscous laminar flows.

Furthermore, in order to simulate low-Reynolds-number turbulent flows past a 3D obstacle, an unstructured-mesh detached eddy simulation (DES) flow solver based on the NFT MPDATA scheme is developed. The Spallart-Allmaras model [79, 80] is modified, cf. Section 2.4, and implemented into the NFT MPDATA solver. The scheme is validated against results available in literature. The advantage of the unstructured-mesh method is that it can be conveniently applied to engineering flows with complex geometry.

ILES based on NFT MPDATA is employed to study stratified high-Reynolds-number turbulent flows past a sphere and a hemisphere. Qualitative and quantitative comparisons are presented to reveal the difference between the two flows.

Finally, an unstructured mesh generator is developed to operate on fixed reduced Gaussian points provided by the European Centre for Medium-Ranged Weather Forecasts (ECMWF). The Rossby-Haurwitz waves benchmark proposed in [89] is simulated for a mesh sensitivity study. A NFT MPDATA edge-based shallow water global atmospheric model [77] is used in the study. It can be also noted that in [77] a NFT MPDATA unstructured mesh approach was developed for three-dimensional hydrostatic flows on a rotating sphere. The present advancement con-
tributed towards a long-term development aiming at combining the nonhydrostatic NFT unstructured mesh global model with the integrated forecasting system (IFS) at ECMWF.

The remainder of the thesis is organised as follows. The governing equations for stratified flows and different approaches for stratified turbulent flow simulation and modelling are discussed in Section 2. The NFT MPDATA numerical scheme and developments in three dimensional unstructured mesh generation are described in Section 3. Section 4 examines the accuracy of tetrahedral meshing in the context of a strongly stratified flow past a steep isolated hill. Section 5 quantifies the model accuracy for simulations with highly anisotropic and inhomogeneous meshes in the context of critical-level phenomena. Section 6 investigates stratified flows past a sphere at $Re = 200$ for simulations based on a hybrid mesh of tetrahedral and prismatic elements. Section 7 studies turbulent flow past a sphere at $Re = 5000$ to validate the development of unstructured-mesh NFT MPDATA based DES. Section 8 compares stratified turbulent flows past a sphere and a hemisphere employing ILES. In Section 9, an unstructured hybrid mesh generator is developed for flows on a sphere and Rossby-Haurwitz waves are simulated on various hybrid meshes. Section 10 concludes the thesis.
2 Governing Equations

2.1 Stratified Flows

The majority of the numerical solutions presented in this thesis are obtained by integrating the anelastic Lipps-Hemler PDEs [36, 37], in the absence of planetary rotation. Using the suffix notation, the governing equations for the evolution of mass, momentum and potential temperature can be compactly written as

\[
\frac{\partial \rho_0 u_j}{\partial x_j} = 0, \tag{1}
\]

\[
\frac{\partial \rho_0 u_i}{\partial t} + \frac{\partial (\rho_0 u_j u_i)}{\partial x_j} = -\rho_0 \frac{\partial p'}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho_0 g \frac{\theta'}{\theta_0} \delta_{i3}, \tag{2}
\]

\[
\frac{\partial \rho_0 \theta}{\partial t} + \frac{\partial (\rho_0 u_j \theta)}{\partial x_j} = 0, \tag{3}
\]

where \((u_1, u_2, u_3)\) is the flow velocity and \(\theta\) is the potential temperature. Subscripts \(0\) denote the static reference state, while \(e\) denote a hydrostatically balanced ambient state [70]. The \(p'\) is the density-normalized pressure perturbation with respect to ambient state. Let \(p\) be pressure, \(p_e\) be ambient pressure profile and \(\rho_0\) be reference density, then \(p' = \frac{p - p_e}{\rho_0}\). The \(\theta'\) is the potential temperature perturbation with respect to the ambient state: \(\theta' = \theta - \theta_e\). The gravity acceleration is denoted by \(g\) and \(g \rho_0 \frac{\theta'}{\theta_0} \delta_{i3}\) is buoyancy, where \(\delta\) is Kronecker’s delta. The \(\frac{\partial \tau_{ij}}{\partial x_j}\) are viscous forces, where \(\tau_{ij} = 2\mu S_{ij} + \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij}\), with \(\mu\) denoting the dynamic viscosity for linear deformations and \(\lambda\) the second viscosity for the volumetric deformation. The \(S_{ij}\) is the strain tensor given by

\[
S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \tag{4}
\]

If the Stokes hypothesis holds, \(\lambda = -\frac{2}{3} \mu\) is a good approximation [75, 83]. The linear stratification of the ambient flow is defined as \(\theta_e(z) = \theta_0(1 + z N^2 / g)\) where \(z \equiv x_3\), and \(N\) is the Brunt-Väisälä frequency. If Boussinesq approximation is adopted, the reference profile is simplified to be constant \(\theta_0(z) = \text{constant}, \rho_0(z) = \text{constant},\)
and \( \lambda \frac{\partial u_k}{\partial x_k} \) vanishes because of mass continuity (1).

Furthermore, if the flow is incompressible with constant density and isentropic, the governing equations (1–3) can be simplified to

\[
\frac{\partial u_j}{\partial x_j} = 0 , \tag{5}
\]

\[
\frac{\partial u_i}{\partial t} + \frac{\partial (u_j u_i)}{\partial x_j} = -\frac{1}{\rho_0} \frac{\partial p}{\partial x_i} + \nu \frac{\partial S_{ij}}{\partial x_j} , \tag{6}
\]

where \( \nu \) is the kinematic viscosity and \( \nu = \frac{\mu}{\rho_0} \).

### 2.2 RANS and Spalart-Allmaras Model

#### 2.2.1 RANS for Anelastic Equations

The derivation of RANS for incompressible flows is described in [83]. The derivation of RANS for anelastic equations is similar. In RANS, the governing equations are averaged in time and the flow properties are decomposed into mean and fluctuation values. Let \( \phi \) denote any flow property \( u, v, w, \rho \) and \( \theta \), then the time averaging of \( \phi \) is given by

\[
\bar{\phi} = \frac{1}{\Delta t} \int_{t}^{t+\Delta t} \phi d\tau ; \tag{7}
\]

by Reynolds decomposition \( \phi \) can be written as a mean and fluctuation term

\[
\phi = \bar{\phi} + \phi' , \tag{8}
\]

such that

\[
\bar{\phi} = \bar{\phi}, \quad \bar{\phi'} = 0 . \tag{9}
\]
For the convection terms,

\[ \frac{\partial \left( \rho_0 u_j u_i \right)}{\partial x_j} = \frac{\partial \left( \rho_0 \bar{u}_j \bar{u}_i \right)}{\partial x_j} + \frac{\partial \left( \rho_0 u'_j u'_i \right)}{\partial x_j}, \]  

\[ \frac{\partial \left( \rho_0 u_j \theta \right)}{\partial x_j} = \frac{\partial \left( \rho_0 \bar{u}_j \bar{\theta} \right)}{\partial x_j} + \frac{\partial \left( \rho_0 u'_j \theta' \right)}{\partial x_j}, \]  

(10)

(11)

For the isentropic applications considered in this thesis, \( \frac{\partial \left( \rho_0 u'_j u'_i \theta' \right)}{\partial x_j} \) vanishes. Let \( \tau_{ij}^R = -\rho_0 \bar{u}_j \bar{u}'_i \) denote the Reynolds stress, then after time-averaging the governing equations (1–3) become

\[ \frac{\partial \rho_0 \bar{u}_i}{\partial x_i} = 0, \]  

(12)

\[ \frac{\partial \rho_0 \bar{u}_i}{\partial t} + \frac{\partial \left( \rho_0 \bar{u}_j \bar{u}_i \right)}{\partial x_j} = -\rho_0 \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \tau_{ij}^R}{\partial x_j} + \rho_0 g \bar{\theta} \delta_{ij}, \]  

(13)

\[ \frac{\partial \rho_0 \bar{\theta}}{\partial t} + \frac{\partial \left( \rho_0 \bar{u}_j \bar{\theta} \right)}{\partial x_j} = 0. \]  

(14)

For isentropic and incompressible flows with constant density, time-averaging the governing equations (5, 6) gives

\[ \frac{\partial \bar{u}_i}{\partial x_i} = 0, \]  

(15)

\[ \frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \left( \bar{u}_j \bar{u}_i \right)}{\partial x_j} = - \frac{1}{\rho_0} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial \bar{S}_{ij}}{\partial x_j} + \frac{\partial \tau_{ij}^R}{\partial x_j}, \]  

(16)

where \( \tau_{ij}^R = -\bar{u}_j \bar{u}'_i \).

### 2.2.2 Spalart-Allmaras One Equation Model

In Spalart-Allmaras one-equation model [79, 80], utilising Boussinesq assumption, the Reynolds stress is approximated as

\[ \tau_{ij}^R = 2 \mu_t \bar{S}_{ij}, \]  

(17)

\[ \bar{S}_{ij} = \frac{1}{2} \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right), \]  

(18)
where $\mu_t$ is the dynamic eddy viscosity that needs to be modelled.

$$\mu_t = \rho_0 \tilde{\nu} f_{v1} \quad f_{v1} = \frac{x^3}{x^3 + c_{v1}^3} \quad \chi = \frac{\tilde{\nu}}{\nu},$$  \hspace{1cm} (19)$$

where $\tilde{\nu}$ is called the working viscosity and obeys the transport equation

$$\frac{D\tilde{\nu}}{Dt} = c_{b1} \tilde{S} \tilde{\nu} + \frac{1}{\sigma} (\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} (\nabla \tilde{\nu})^2) - c_{w1} f_w \left( \frac{\tilde{\nu}}{d} \right)^2. \hspace{1cm} (20)$$

The right-hand-side terms are the production, diffusion and destruction terms, respectively. Here

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad f_{v2} = 1 - \frac{x}{1 + x f_{v1}}, $$  \hspace{1cm} (21)$$

where $S$ is the magnitude of vorticity; $S = |\nabla \times \mathbf{v}|$ and $\mathbf{v} = (u_1, u_2, u_3)$. $d$ is the distance to the closest wall. $f_w$ is given by

$$f_w = g \left( \frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6}, \quad g = r + c_{w2} (r^6 - r) \quad r = \frac{\tilde{\nu}}{S \kappa^2 d^2}. \hspace{1cm} (22)$$

If $r$ is too large and $f_w$ reaches a constant value, then $r$ can be truncated to around 10. The wall boundary condition is $\tilde{\nu} = 0$, and its free-stream value should be zero, or at least positive. The constants are $c_{b1} = 0.1355$, $\sigma = 2/3$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w1} = c_{b1}/\kappa + (1 + c_{b2})/\sigma$, $c_{w2} = 0.3$, $c_{w3} = 2.0$, $c_{v1} = 7.1$.

The basic idea in the derivation of RANS is that the governing equations are averaged in time, regardless of how fine is the mesh, thus the turbulence model for RANS would not depend on the local grid size. It is also assumed that the eddies are isotropic, so that $\mu_t$ is modelled to be uniform in different directions.
2.3 LES and Dynamic Smagorinsky Model

2.3.1 LES for Anelastic Equations

The formulation of LES for incompressible flows, presented e.g. in [83] applies readily to anelastic equations (1–3). In LES, the governing equations are spatially filtered to separate the effects of large and small eddies. Large eddies behave in a more anisotropic way, while smaller eddies are assumed to be more isotropic and behave in a more universal way. Thus the larger eddies are resolved while the smaller eddies are modelled. For any flow property \( \phi(x, t) \) at position \( x_0 \), the filtering operator [83] is defined as

\[
\tilde{\phi}(x_0, t) = \int_{\Omega} \phi(x, t) G(x_0, x, \Delta) \, dx ,
\]  

(23)

where \( G(x_0, x, \Delta) \) is the spatial filter and \( \Delta \) is the cut-off width. The filter is defined as \( G(x_0, x, \Delta) = 1/\Delta^3 \) if \( |x - x_0| \leq \Delta/2 \), and zero otherwise. The cut-off width is set to be the same order as the grid size [83]. Let the cell volume at \( x_0 \) be \( V_{x_0} \), then \( \Delta = \sqrt[3]{V_{x_0}} \). In finite volume method adopted here, there is only one data point in each computational dual mesh cell and all finer details within this volume are not resolved, so the dual mesh cell size naturally limits the cut-off width.

Applying spatial filtering to the governing equations (1), (2) and (3), for the convection term results in

\[
\rho_0 \tilde{u}_i u_j = \rho_0 \tilde{u}_i \tilde{u}_j + (\rho_0 \tilde{u}_i u_j - \rho_0 \tilde{u}_i \tilde{u}_j) = \rho_0 \tilde{u}_i \tilde{u}_j + \tau_{ij}^S ,
\]  

(24)

where \( \tau_{ij}^S \) is called the sub-grid-scale (SGS) stress. The filtered governing equations

\(^1 \rho_0 \) is a smooth function of \( z \) in the examples considered in this thesis
are written as
\[ \frac{\partial \rho_0 \tilde{u}_j}{\partial x_j} = 0 \] (25)
\[ \frac{\partial \rho_0 \tilde{u}_i}{\partial t} + \frac{\partial (\rho_0 \tilde{u}_j \tilde{u}_i)}{\partial x_j} = -\rho_0 \frac{\partial \tilde{P}'}{\partial x_i} + \frac{\partial \tilde{\tau}_{ij}}{\partial x_j} + \rho_0 g \frac{\tilde{\theta}'}{\theta_0} \delta_{ij} \] (26)
\[ \frac{\partial \rho_0 \tilde{\theta}}{\partial t} + \frac{\partial (\rho_0 \tilde{u}_j \tilde{\theta})}{\partial x_j} = -\frac{\partial (\rho_0 \tilde{u}_i \tilde{\theta} - \rho_0 \tilde{u}_j \tilde{\theta})}{\partial x_j} \] (27)

In Smagorinsky model [56], the SGS stress is modelled as
\[ \tau_{ij}^S - \frac{1}{3} \tau_{kk} \delta_{ij} = -2 \mu_{SGS} \hat{S}_{ij} \] (28)
\[ \hat{S}_{ij} = \frac{1}{2} \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \] (29)
where \( \mu_{SGS} \) is a dynamic viscosity obtained from a SGS model. Let \( \tilde{P}' = \tilde{P}' + \frac{1}{3} \tau_{ii} \delta_{ij} \), then equation (26) can be rewritten as
\[ \frac{\partial \rho_0 \tilde{u}_i}{\partial t} + \frac{\partial (\rho_0 \tilde{u}_j \tilde{u}_i)}{\partial x_j} = -\rho_0 \frac{\partial \tilde{P}''}{\partial x_i} + \frac{\partial \tilde{\tau}_{ij}}{\partial x_j} + 2 \mu_{SGS} \frac{\partial \hat{S}_{ij}}{\partial x_j} + \rho_0 g \frac{\tilde{\theta}'}{\theta_0} \delta_{ij} \] (30)
and for the Smagorinsky model, the dynamic viscosity becomes
\[ \mu_{SGS} = \rho_0 (C_{SGS} \Delta)^2 |\hat{S}_{ij}| = \rho_0 (C_{SGS} \Delta)^2 \sqrt{2 \hat{S}_{ij} \hat{S}_{ij}} \] (31)
where \( C_{SGS} \) is constant for the basic Smagorinsky model. Its value varies for different applications, but it is normally below 0.25 [32, 33].

2.3.2 Dynamic Smagorinsky Model

In the dynamic Smagorinsky model by Germano and Lily [18, 34] \( C_{SGS} \) is not constant. The derivation process in [18, 34] is followed for the governing equations (1–3). The dynamic model is formulated by introducing a test filter in the following
form
\[ \hat{\phi}(x_0, t) = \int_{\Omega} \phi(x, t) G_2(x_0, x, \Delta_2) \, dx. \] (32)

Applying (32) to (25–27) results in
\[ \frac{\partial \rho_0 \hat{u}_j}{\partial x_j} = 0, \] (33)
\[ \frac{\partial \rho_0 \hat{u}_i}{\partial t} + \frac{\partial (\rho_0 \hat{u}_j \hat{u}_i)}{\partial x_j} = -\rho_0 \frac{\partial \hat{p}}{\partial x_i} + \frac{\partial \hat{r}_{ij}}{\partial x_j} - \frac{\partial \tau^{S2}_{ij}}{\partial x_j} + \rho_0 g \delta_{ij}, \] (34)
\[ \frac{\partial \rho_0 \hat{\theta}}{\partial t} + \frac{\partial (\rho_0 \hat{u}_j \hat{\theta})}{\partial x_j} = -\frac{\partial (\rho_0 (\hat{u}_j \hat{\theta} - \hat{u}_j \hat{\theta}))}{\partial x_j}, \] (35)

where the SGS stress becomes
\[ \tau^{S2}_{ij} = \hat{u}_i \hat{u}_j - \hat{\theta}_i \hat{\theta}_j. \] (36)

The intermediate turbulent stress \( L_{ij} \) is defined next as the contribution to \( \tau^{S2}_{ij} \) by the scales between the grid filter and the test filter, such that
\[ L_{ij} = \hat{u}_i \hat{u}_j - \hat{\theta}_i \hat{\theta}_j. \] (37)

Thus
\[ L_{ij} = \tau^{S2}_{ij} - \hat{\tau}^S_{ij}. \] (38)

Let \( |\hat{\delta}_{ij}| = (2 \hat{S}_{kl} \hat{S}_{kl})^{1/2} \), (28) and (31) result in
\[ \tau^S_{ij} - \frac{1}{3} \tau^S_{kk} \delta_{ij} = -2 \rho_0 (C_{SGS} \Delta)^2 |\hat{\delta}_{ij}| \hat{\delta}_{ij}. \] (39)

Similarly,
\[ \tau^{S2}_{ij} - \frac{1}{3} \tau^{S2}_{kk} \delta_{ij} = -2 \rho_0 (C_{SGS} \Delta_2)^2 |\hat{\delta}_{ij}| \hat{\delta}_{ij}. \] (40)
Substituting (39) and (40) into (37) gives

\[ L_{ij} - \frac{1}{3} \delta_{ij} L_{kk} = -2 C_{SGS} M_{ij}, \]  

(41)

where

\[ M_{ij} = \Delta_2 |\hat{\tilde{S}}| \hat{\tilde{S}}_{ij} - \Delta^2 |\tilde{S}| \tilde{S}_{ij}. \]  

(42)

Defining \( Q \) as the square of the error in (41) gives

\[ Q = (L_{ij} - \frac{1}{3} \delta_{ij} L_{kk} + 2 C_{SGS} M_{ij})^2. \]  

(43)

Setting \( \partial Q/\partial C = 0 \) allows \( C_{SGS} \) to be evaluated as

\[ C_{SGS} = -\frac{1}{2} (L_{ij} M_{ij}/M_{ij}^2). \]  

(44)

In contrast to the basic Smagorinsky model, \( C_{SGS} \) obtained from the above equation is not constant, and can lead to more accurate simulation results, especially in rotating or sheared flows and in transitional regimes [18, 34].

### 2.4 DES

Detached eddy simulation (DES) [80] combines the techniques of RANS and LES. The flow is simulated by RANS in regions near the solid wall, while by LES elsewhere, [4]. In Spalart-Allmaras one equation model, there is a production term \( c_{b1} \hat{S} \tilde{\nu} \), and a destruction term \( -c_{w1} f_w (\frac{\tilde{\nu}}{\gamma})^2 \) in the right-hand-side terms of (19). The balance between them leads to

\[ \tilde{\nu} \propto S d^2. \]  

(45)
The kinematic SGS eddy viscosity $\nu_{SGS} = \mu_{SGS}/\rho_0$, therefore, from (31)

$$\nu_{SGS} \propto \tilde{S}_{ij} \Delta^2. \tag{46}$$

Replacing $d$ by a length proportional to $\Delta$ allows to obtain a SGS model of the same order of accuracy as the Smagorinsky model [80]. Thus a new length is defined as

$$\tilde{d} = \min(d, C_{DES} \Delta), \tag{47}$$

giving a turbulence model that acts as a RANS eddy viscosity when $d \leq \Delta$ and as a SGS viscosity when $d > \Delta$. It can be noted that DES also becomes DNS in the limit of an extremely fine mesh and small time step [80].

DES is also called a hybrid RANS-LES approach. The flow variables have a form of time-averaged values in RANS and spatially filtered values in LES. However, since in both RANS and LES the grid size and time interval are both relatively large, $u$, $v$, $w$, $\rho$ and $\theta$ are all averaged values both in time and space [87].

For example, in LES the derivative with respect to time can be evaluated as

$$\frac{\partial \phi(x, t)}{\partial t} = \frac{\phi(x, t + dt) - \phi(x, t)}{dt}. \tag{48}$$

If $dt \to 0$, the approximation is exact. However, in practical calculations, $dt \gg 0$ and (48) results in the derivative of the average value of $\phi(x, t)$ in time. On the other hand, the mesh is not very fine in RANS, therefore $\phi(x, t)$ represents the average value in a computational mesh cell in finite volume method. The governing equations of LES are in the same form as RANS, except that the turbulence models are different [80].
2.5 ILES

Some of the first published successful ILES simulations include [3, 47, 20, 51]. A brief history of ILES is included in [45]. In LES, the SGS stress model is written in an explicit form, while in ILES an implicit SGS model is incorporated in the numerical scheme. The motivation for ILES lies in the modified equation analysis (MEA) [28] of the LES filtered equations. MEA is a method for generating, via Taylor series expansion, a PDE that approximates the numerical scheme [45]. Following [15, 16] the modified equation (ME) for the LES filtered equations (26) is written as

$$\frac{\partial \rho_0 \tilde{u}_i}{\partial t} + \partial (\rho_0 \tilde{u}_j \tilde{u}_i) = -\rho_0 \frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} - \rho_0 g \frac{\tilde{\theta}}{\theta_0} \delta_{i3} + \frac{\partial \tau_{ij}^c}{\partial x_j} + m_i, \quad (49)$$

where $\partial \tau_{ij}^c/\partial x_j$ is the error term associated with the numerical scheme and discretisation, and $m_i$ is the commutation error due to the assumption that $\tilde{\partial \phi}/\tilde{\partial x_i} = \partial \phi/\partial x_i$ for any flow property $\phi$ [16, 5]. A successful LES requires that $|\partial \tau_{ij}^c/\partial x_j| \ll |\partial \tau_{ij}^c/\partial x_j|$, however in practical simulations it is seldom satisfied [16].

In the MEA for ILES, the commutation error terms vanish and there is no explicit SGS model [15]. Equation (49) becomes

$$\frac{\partial \rho_0 \tilde{u}_i}{\partial t} + \partial (\rho_0 \tilde{u}_j \tilde{u}_i) = -\rho_0 \frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho_0 g \frac{\tilde{\theta}}{\theta_0} \delta_{i3} + \frac{\partial \tau_{ij}^c}{\partial x_j}, \quad (50)$$

where $\partial \tau_{ij}^c/\partial x_j$ is the implicit SGS model and the filtering is implicitly provided by the grid size [15].

The success of ILES depends largely on the choice of the numerical scheme and not all schemes are suitable for ILES [45]. Some high-resolution numerical schemes [24] have proven to be suitable for ILES [5]. In this thesis we adopt the NFT MPDATA scheme. A theoretical justification of the scheme’s ILES capability can be found in [44, 45] where it is proved that the implicit SGS stress of MPDATA is the sum of a dissipative term akin to the Smagorinsky model, and a nonlinear term similar to the self-similar Clark model [52]. Thus, the NFT MPDATA based
implicit SGS stress is in the form of a mixed LES model [45].

The connection between the explicit and implicit SGS stress model implies that one can generate a new explicit SGS model from a numerical scheme suitable for ILES, and on the other hand one can also design a new numerical scheme based on an explicit SGS model [45, 27].

2.6 The Function of Absorbers

The derivation in this section follows [73]. The absorbers attenuate the waves in an exponential way and help modify the flow properties $u, v, w$ and $\theta$ to their ambient values. For any property $\phi$, the absorber is in the form

$$\frac{d\phi}{dt} = -\alpha \phi .$$  \hspace{0.5cm} (51)

Suppose $l$ is a spatial coordinate. The absorbers attenuate the waves from $l = l_0$ to the boundary at $l = H$ exponentially. $l = l_0$ can be regarded as the absorber’s inner boundary. Let $\phi = \phi(l, t)$, then

$$\frac{d\phi}{dt} = \frac{d\phi}{dl} \frac{dl}{dt} .$$  \hspace{0.5cm} (52)

Denoting $\frac{dl}{dt} = c$ gives

$$\frac{d\phi}{dt} = -\frac{\alpha}{c} \phi ,$$  \hspace{0.5cm} (53)

where $c$ is a signal propagation speed. Setting $\alpha = \alpha_0 \frac{l - l_0}{H - l_0}$ for $l \in [l_0, H]$ and zero elsewhere leads to

$$\frac{d\phi}{dt} = -\frac{\alpha_0}{c} \frac{l - l_0}{H - l_0} \phi .$$  \hspace{0.5cm} (54)
Let $D = (H - l_0)$ and $\alpha_0 = \frac{1}{\tau}$. The coefficient $D$ is the depth of the absorber-affected area and $\tau$ is the strength of the absorbers. Then

$$\frac{d\phi}{dl} = \frac{l - l_0}{cD\tau} \phi .$$

(55)

Then

$$\frac{1}{\phi} \frac{d\phi}{dl} = -\frac{l - l_0}{cD\tau} .$$

(56)

Integrating from $l_0$ to $l$ and after manipulation gives

$$\phi(l) = \phi(l_0) e^{-\frac{(l-l_0)^2}{2cD\tau}} .$$

(57)

Equation (57) shows that $\phi(l)$ is a probability density function of a normal distribution. Let the standard deviation be denoted as $\sigma$, then

$$\sigma = \sqrt{cD\tau} .$$

(58)

In order to attenuate $\phi(l)$, the depth of the absorber-affected area is assumed to be

$$D = 2\sigma = 2\sqrt{cD\tau} .$$

(59)

Thus

$$\tau = \frac{D}{4c}$$

(60)

For the purpose of implementing absorbers, the governing equations (1) to (3)
are modified to be

\[
\frac{\partial \rho_0 u_j}{\partial x_j} = 0 ,
\]
(61)

\[
\frac{\partial \rho_0 u_i}{\partial t} + \frac{\partial (\rho_0 u_j u_i)}{\partial x_j} = -\rho_0 \frac{\partial p'}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + g \rho_0 \frac{\theta'}{\theta_0} \delta_{i3} - \alpha (u_i - u_{i,e}) ,
\]
(62)

\[
\frac{\partial \rho_0 \theta}{\partial t} + \frac{\partial (\rho_0 u_j \theta)}{\partial x_j} = -\alpha (\theta - \theta_c) ,
\]
(63)

where the attenuation terms (\(\sim \alpha\)) absorb gravity waves in the vicinity of the model’s open boundaries.
3 Numerical Scheme

3.1 Edge Based Finite Volume Spatial Discretisation

The spatial discretisation employed in the present work uses the edge-based median-dual finite volume approach [65] integrating the governing PDE over arbitrarily-shaped cells, with all discrete differential operators evaluated from the Gauss divergence theorem. A schematic of the edge-based data structure for an arbitrary hybrid mesh on a 2D plane is shown in Figure 1. The median-dual finite volume approach constructs the control volume containing the node $i$ by joining the barycentres of polygonal mesh cells encompassing the node $i$ with the midpoints of the edges originating in the node $i$. For illustration, Figure 3 presents a 2D primary triangular mesh and the corresponding dual mesh.

![Figure 1: The edge-based, median-dual approach in 2D. The edge connecting nodes $i$ and $j$ of the primary polygonal mesh pierces, precisely in its middle, the face $S_j$ shared by computational dual cells surrounding nodes $i$ and $j$; open circles represent barycentres of the primary mesh, while solid and dashed lines mark primary and dual meshes, respectively. $r$ and $l$ are barycentres of the primary mesh while $c$ is the midpoint of edge $ij$.](image)

The construction of a dual mesh in three dimensions is similar, yet providing an illustration equivalent to Figure 1 is hardly possible. Instead, Figure 2 shows the
edge $i \rightarrow j$ connecting nodes $i$ and $j$, intersected exactly in the middle by the face $S_j$ of a dual cell containing node $i$. The construction of the face $S_j$ involves first the identification of the polyhedral elements belonging to the primary mesh that share the edge $i \rightarrow j$. Next, the polyhedral barycentres and the centres of polygonal faces, through which neighbouring polyhedra are jointed, are connected with each other and with the mid-point of the edge $i \rightarrow j$, as indicated by dashed lines in Figure 2, to form an umbrella like face of the 3D dual mesh.

![Figure 2: The edge-based median-dual approach in 3D. The edge connecting nodes $i$ and $j$ of the primary mesh pierces (at the edge centre) the face $S_j$ of a computational (dual) cell surrounding node $i$; open circles represent barycentres of polyhedral cells surrounding the edge. Dashed lines mark a fragment of the dual mesh.](image)

3.2 MPDATA

For completeness, the edge-based finite volume MPDATA scheme derived in [65] is briefly described here. All notations follow closely that of [65]. Consider the generic advection equation

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot (\mathbf{v}\phi) ,$$

(64)
where $v = v(x, t)$ is an advection velocity, and $\phi = \phi(x, t)$ is a scalar field. (64) is integrated over the volume of an arbitrary cell while employing the Gauss divergence theorem

\[
\phi_i^{n+1} = \phi_i^n - \frac{\delta t}{V_i} \sum_{j=1}^{l(i)} F_j^\perp S_j ,
\]

where the mean values of $\phi$ of the volume $V_i$ containing the vertex $i$ are assumed to be equal to $\phi_i^n$ and $\phi_i^{n+1}$ at time steps $n$ and $n + 1$, respectively. $F_j^\perp$ is the mean normal flux of $\phi$ through the cell face $S_j$ averaged over temporal increment $\delta t$. $S_j$ also refers to the surface area of the face. $l(i)$ is the number of edges connecting the vertex $i$ and its neighbouring nodes and it is also the number of neighbouring cell volumes for the volume $V_i$. Each edge corresponds to a cell face which is shared between the two neighbouring cells. $F_j^\perp$ is approximated using the first order upwind scheme in the form

\[
F_j^\perp = [v_j^\perp]^+ \phi_i^n + [v_j^\perp]^− \phi_j^n ,
\]

where $v_j^\perp$ is the normal velocity through the face $S_j$.

\[
[v]^+ = 0.5(v+ | v |), \quad [v]^− = 0.5(v− | v |) ,
\]

and $v_j^\perp$ can be evaluated on the face $S_j$. The non-negative/non-positive parts of $v_j^\perp$ corresponds to outflow/inflow from the $i$th cell. The temporal derivative is expressed by expanding the advection equation (64).

\[
\frac{\partial \phi}{\partial t} = -v \cdot \nabla \phi - \phi \nabla \cdot v.
\]
truncation error is determined by expressing \( \phi^n_i \) and \( \phi^n_j \) in (66) using Taylor series expansion in space and time and expressing temporal derivative as spatial derivatives. The expansion is about \( t^{n+1/2} \) and the point \( c \), where the edge intersects the cell face \( S_j \). After substitution and rearrangements the flux (66) can be written as

\[
F^\perp_j = v^\perp_j \phi|^{n+1/2}_c + \text{Error}_{adv},
\]

where the truncation error of the upwind scheme

\[
\text{Error}_{adv} = -0.5 | v^\perp_j | \left( \frac{\partial \phi}{\partial r} \right)^*_{c} (r_j - r_i) + 0.5 v^\perp_j \left( \frac{\partial \phi}{\partial r} \right)^*_{c} (r_i - 2r_c + r_j) + 0.5 \delta t v^\perp_j (\nabla \cdot \phi) |^*_{c}
\]

\[+ 0.5 \delta t v^\perp_j (\phi \nabla \cdot \mathbf{v}) |^*_{c} + 0 (\delta t^2, \delta t \delta r). \tag{70} \]

Here \( r \) is a parametric description of the edge such that \( r(\lambda) = r_i + \lambda (r_j - r_i) \) and \( 0 \leq \lambda \leq 1 \). The asterisk in the equation indicates any time step \( n, n+1/2 \) or \( n+1 \). The choice of the time step for the asterisk has no effect on the order of error. The pseudo velocity for the second upwind scheme can be now defined as \( \tilde{v} = -\frac{1}{\phi} \text{Error} \), and expanded as

\[
\tilde{v}^\perp_j = 0.5 | v^\perp_j | \left( \frac{1}{\phi} \frac{\partial \phi}{\partial r} \right)^*_{c} (r_j - r_i) - 0.5 v^\perp_j \left( \frac{1}{\phi} \frac{\partial \phi}{\partial r} \right)^*_{c} (r_i - 2r_c + r_j) - 0.5 \delta t v^\perp_j (\mathbf{v} \frac{1}{\phi} \nabla \phi) |^*_{c}
\]

\[+0.5 \delta t v^\perp_j (\nabla \cdot \mathbf{v}) |^*_{c}. \tag{71} \]

The terms with the asterisk depends on the result of \( \phi \) from the preceding upwind scheme. In principle, the whole precess of determining the truncation error and compensating it can be repeated to further reduce the truncation error [46]. But in practice the basic MPDATA with one corrective iteration is enough to recover the overall accuracy of time and space centred schemes. The outlined expression above for basic MPDATA can be applied to arbitrary meshes, either structured or unstructured.

The median-dual edge-based data structure finite volume arrangement simplifies the expression (71) for the pseudo velocity. The dual mesh is constructed by joining
the centres of tetrahedra, surfaces and edges surrounding the same vertex. In our edge-based data structure, the point \( c \) where the edge connecting points \( i \) and \( j \) intersects the cell face \( S_j \) is at the middle of the edge. Then the surface-weighted pseudo velocity (71) can be approximately simplified to be

\[
\hat{v}^\perp_j = \left| v^\perp_j \right| \frac{\phi^*_j - \phi^*_i}{\phi^*_j + \phi^*_i + \epsilon} - \frac{\delta t}{2} v^\perp_j \left( \mathbf{v} \cdot \nabla \phi^* + \nabla \cdot \mathbf{v} \right),
\]  

(72)

where \( \phi^* \) denotes the results after the first upwind iteration, and \( \epsilon \) is a small constant introduced to prevent the denominator to be zero when \( \phi^*_j \) and \( \phi^*_i \) both become zero. For each point in the dual mesh, \( \nabla \phi^* \) can be determined using Gauss divergence theorem and \( \overline{\phi^*} \) is the arithmetic average of \( \phi^* \)'s of all the points employed to evaluate \( \nabla \phi^* \) for the point. Herein the infinite gauge option [66] is employed exclusively which simplifies the expression (72) further [76] and evaluates the flux as

\[
\hat{F}^\perp_j = \left| v^\perp_j \right| \frac{\phi^*_j - \phi^*_i}{2} - \frac{\delta t}{2} v^\perp_j (\mathbf{v} \cdot \nabla \phi^*)_c.
\]  

(73)

The basic MPDATA consists of two upwind iterations in a time step. The first iteration is a generic upwind scheme and the second one is a upwind scheme with a pseudo-velocity to compensate the truncation error of the first iteration. A flow chart of the implementation of MPDATA is listed below

1. Compute the upwind fluxes \( F^\perp_j \) according to Equations (66) and (67).
2. Update \( \phi \) using Equation (65) with the fluxes obtained in Step 1
3. Calculate the pseudo-velocity according to Equation (72).
4. Calculate the normal flux as in Step 1 but using values of \( \phi \) and \( \mathbf{v} \) obtained in Steps 2 and 3.
5. Update \( \phi \) by reusing the upwind scheme (65) but with the pseudo-flux obtained in the previous step.
The consistency, stability and accuracy of finite volume MPDATA is discussed in detail in [65, 66] and references therein. The upwind scheme is consistent, conditionally stable and first-order accurate [59]. When the temporal and spatial increments tend to zero, the pseudo velocity (71) also tend to zero, therefore the consistency of the upwind scheme leads to that of MPDATA. In MPDATA, the corrective step compensates the first order leading error of the first upwind step, with the uncompensated error remaining second order.

3.3 Computation of Derivatives

The edge-based data structure provides a convenient way of designing discrete differential operators. For a differentiable vector field $H$, the Gauss divergence theorem

$$\int_{\Omega} \nabla \cdot H = \int_{\partial \Omega} H \cdot n$$

applied over the control volume $V_i$ surrounding vertex $i$ leads to

$$\nabla_i \cdot H = \frac{1}{V_i} \sum_{j=1}^{l(i)} H_j \cdot S_j ,$$

where $l(i)$ denotes the number of edges connecting the vertex $i$. The left-hand-side term is interpreted as the mean value of the derivative within the control volume $V_i$ and $H_j \cdot S_j$ is interpreted as the mean normal component of $H$ on cell face $S_j$.

Partial derivatives $\frac{\partial \phi}{\partial x}$, $\frac{\partial \phi}{\partial y}$ and $\frac{\partial \phi}{\partial z}$ can also be calculated according to the Gauss Divergence Theorem using the edge-based data structure.

$$\left( \frac{\partial \phi}{\partial x} \right)_i = \frac{1}{V_i} \sum_{j=1}^{l(i)} \phi_{\text{average}} S_x |_j ,$$

$$\left( \frac{\partial \phi}{\partial y} \right)_i = \frac{1}{V_i} \sum_{j=1}^{l(i)} \phi_{\text{average}} S_y |_j ,$$

$$\left( \frac{\partial \phi}{\partial z} \right)_i = \frac{1}{V_i} \sum_{j=1}^{l(i)} \phi_{\text{average}} S_z |_j ,$$
where \( i \) denotes the \( i' \)th point and \( V_i \) the volume around point \( i \). \( \phi_{\text{average}} \) denotes the average value of \( \phi \) on the edge. If the two points of the edge are \( i \) and \( j \) then we simply assume that

\[
\phi_{\text{average}} = \frac{\phi_i + \phi_j}{2}.
\] (79)

### 3.4 Non-oscillatory Option

MPDATA scheme is positive definite, or in other words, preserves the sign, but it does not assure the monotonicity of the transported variable [65]. The solutions is not bounded by the local extreme values, which are the maximum and minimum values of the transported variable among neighbouring mesh points. In other words, the solutions are not free of spurious extreme values. The reason is that the pseudo velocity is not necessarily solenoidal, even for a solenoidal flow [9]. MPDATA can be made monotone by using the treatment similar to a Flux Corrected Transport (FCT) scheme [92], to limit the pseudo velocity. A detailed discussion can be found in [61]. The FCT-limited anti-diffusive pseudo velocity, following notation in [65], is

\[
\hat{\hat{v}}_j^\perp = [\hat{v}_j^\perp]^+(\min(1, \beta_\uparrow^i, \beta_\downarrow^j)[sgn(\phi_n^i)]^+ + \min(1, \beta_\uparrow^i, \beta_\downarrow^j)[sgn(-\phi_n^i)]^+) + [\hat{v}_j^\perp]^-(\min(1, \beta_\uparrow^i, \beta_\downarrow^j)[sgn(\phi_n^j)]^+ + \min(1, \beta_\uparrow^i, \beta_\downarrow^j)[sgn(-\phi_n^j)]^+),
\] (80)

where for all \( i \), the limiting coefficients \( \beta_\uparrow^i \) and \( \beta_\downarrow^i \) are

\[
\beta_\uparrow^i = \frac{\phi_i^{\text{MAX}} - \phi_i^*}{\frac{\mu}{V} \sum_{j=1}^{l(i)} [F_j^*]^- + \epsilon}, \quad \beta_\downarrow^i = \frac{\phi_i^* - \phi_i^{\text{MIN}}}{\frac{\mu}{V} \sum_{j=1}^{l(i)} [F_j^*]^+ + \epsilon},
\] (82)

and the limiters \( \phi_i^{\text{MAX}} \) and \( \phi_i^{\text{MIN}} \) are defined as

\[
\phi_i^{\text{MAX}} = \max_{j=1}^{l(i)}(\phi_n^i, \phi_n^j, \phi_n^*, \phi_n^*),
\] (83)

\[
\phi_i^{\text{MIN}} = \min_{j=1}^{l(i)}(\phi_n^i, \phi_n^j, \phi_n^*, \phi_n^*).
\] (84)
3.5 NFT Flow Solver

The NFT method were introduced for structured grid solvers in the early nineties [62]. They have been further extended to unstructured meshes for anelastic atmospheric models for the first time in [78]. Any of the governing equations can be represented by a generic equation as

$$\frac{\partial \phi}{\partial t} + \text{div}(\mathbf{v}\phi) = R,$$

where the value of a fluid property per unit volume is denoted as $\phi$, $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ and $R = R(\mathbf{x}, t)$. $R$ combines all forces or sources and density is included in $\phi$. The homogeneous counterpart of (85) is (64). The NFT flow solver for (85) can be built on a two-time-level non-oscillatory advection scheme. Following [65], extending the truncation error analysis of the advection equation (64) to (85) gives

$$F_j^{n+1} = v_j \phi_{c}^{n+1/2} + \text{Error}_{adv} - 0.5 \delta t v_j R_{c}^{n+1/2} .$$

The difference between (69) and (86) is that in the latter one there is an error which couples velocity and forces/sources. This forcing-related error can be compensated by advecting the sum of the transported variable $\phi$ and the forces/sources, which is important for preserving second-order accuracy and stability of the numerical scheme (Section 3.3 in [63]). The resulting NFT flow solver template can be written in the following form

$$\phi_i^{n+1} = A_i(\phi^n + 0.5 \delta t R^n, \mathbf{v}^{n+1/2}) + 0.5 \delta t R_{i}^{n+1} \equiv \hat{\phi}_i + 0.5 \delta t R_{i}^{n+1} ,$$

where the transport operator $A$ denotes a NFT advection algorithm such as MPDATA which is used in this work. In (87), $R_i$ is integrated in time along a trajectory using $\int_{n}^{n+1} R_i = \frac{R_i^n + R_i^{n+1}}{2} = 0.5 \delta t R_i^n + 0.5 \delta t R_i^{n+1}$. The first term $0.5 \delta t R_i^n$ of time step $n$ is advected with the transporting variable $\phi$ while the second term $R_i^{n+1}$ of
time step \( n + 1 \) is calculated explicitly or implicitly. In general, the NFT solver is determined by the choice of the non-oscillatory advection scheme \( A \) and the method of evaluating \( R_i^{n+1} \). The computation method of \( R_i^{n+1} \) depends on the form of PDE we solve and the choice of dependent variables.

### 3.6 Elliptic Solver

The execution of the NFT template solver (87) for the governing equations (1–3) follows the description in [78]. Firstly, the advective velocity at \( t + 0.5\delta t \) is estimated by a linear extrapolation in time. The term \( \phi^n + 0.5\delta t R^n \) is advected for momentum components and potential temperature, resulting in respective \( \hat{\phi}_i \) counterparts in (87). The advection also updates the potential temperature in (3), for adiabatic dynamics considered in the present study. In consequence, the buoyancy term in (2) can be calculated explicitly, and the only unknown term on the right-hand-side of (2) is the pressure perturbation gradient. Applying divergence operator on both sides of (2) and employing mass continuity (1) results in the Poisson equation for pressure perturbation \( p' \).

This section recalls material based on the preconditioned conjugate residual scheme exposed in [70]. All notation from [70] is retained. The computation of a general second-order elliptic partial differential equation proceeds as follows

\[
\sum_{i=1}^{M} \frac{\partial}{\partial x_i} \left( \sum_{j=1}^{M} C_{i,j} \frac{\partial \phi}{\partial x_j} + D_i \phi \right) - A \phi = R ,
\]

(88)

with coefficients \( A, C_{i,j}, D_i, R \) and periodic, Dirichlet, or Neumann boundary conditions. Equation (88) can be compactly written as

\[
L(\phi) - R = 0 .
\]

(89)

The preconditioned conjugate residual scheme is employed to solve Equation (89). Equation (89) is augmented with a pseudo-time dependence to help explain an
iterative variational Krylov solver. In lieu of Equation (89), the following equation is solved

\[
\frac{\partial^k P(\phi)}{\partial \tau^k} + \frac{1}{T_{k-1}(\tau)} \frac{\partial^{k-1} P(\phi)}{\partial \tau^{k-1}} + \ldots + \frac{1}{T_1(\tau)} \frac{\partial P(\phi)}{\partial \tau} = L(\phi) - R.
\] (90)

The residual error is defined to be

\[
r = L(\phi) - R.
\] (91)

A pre-conditioner \( P \) helps accelerate the convergence of the scheme and \( P \) can be (in principle) any linear operator such that \( LP^{-1} \) is negative definite. The scheme is written as follows [70]

1. For any initial guess \( \phi^0 \), set \( r^0 = L(\phi^0) - R, p^0 = P^{-1}(r^0) \).

2. For \( n = 0, 1, 2 \ldots \) until convergence do

3. For \( v = 0, \ldots, k - 1 \) do

4. \( \beta = -\frac{r^v L(p^v)}{L(p^v) L(p^v)} \)

5. \( \phi^{v+1} = \phi^v + \beta p^v \)

6. \( r^{v+1} = r^v + \beta^n L(p^v) \)

7. exit if \( |r^{v+1}| \leq \epsilon \),

8. \( q = P^{-1}(r^{v+1}) \)

9. \( L(q) = \sum_{i=1}^{M} \frac{\partial}{\partial x_i} \left( \sum_{j=1}^{M} C_{i,j} \frac{\partial q}{\partial x_j} + D_i q \right) - Aq \)

10. \( \forall l0, v, \alpha_l = -\frac{L(q) L(p^l)}{L(p^v) L(p^v)} \)

11. \( p^{v+1} = q + \sum_{l=0}^{v} \alpha_l p^l \)

12. \( L(p^{v+1}) = L(q) + \sum_{l=0}^{v} \alpha_l L(p^l) \)
13. end do
14. reset $\phi^k, r^k, p^k$ and $L(p^k)$ to be $\phi^0, r^0, p^0$ and $L(p^0)$
15. end do

### 3.7 Aspects of Mesh Generation

In this section, we will discuss the details regarding dual mesh construction. Primary meshes are stored in an element-based basic data structure which includes a list of mesh points, a list of elements\(^2\) and a list of faces on each boundary of the computational domain. Any data structured beyond the basic data structure is called a derived data structure [38]. The corresponding dual meshes are stored in an edge-based data structure to achieve discretisation flexibility and low memory allocation. The edge-based data structure consists of a list of mesh points with the cell volume enclosing each point, a list of internal and boundary edges, a list of orientated dual-mesh cell face area associated with each edge, a list of orientated boundary face areas and their associated unit normal vectors. The process of converting the three dimensional element-based primary mesh into the edge-based dual mesh is not straightforward, but involves complex manipulation of different derived data structures, e.g. elements surrounding points, points surrounding points, elements surrounding elements, etc. For instance, a dual mesh generation process for primary tetrahedral meshes is outlined below.

1. The primary-mesh elements surrounding each point are identified. The selected searching algorithms follow closely principles described in [38]. For the reader’s convenience, the notation from [38] is also used in this section. The points of each element are stored in a matrix $\text{inpoel}(nelem, mx)$ where $nelem$ is the number of elements in the primary mesh and $mx$ is the biggest possible number of nodes for an element. In the case of primary tetrahedral mesh,

\(^2\)The list contains the nodes of every element for recording connectivity
**3 NUMERICAL SCHEME**

$m_x = 4$. However, the number of elements surrounding a point can fluctuate widely across an unstructured mesh and using a matrix for storage can lead to a considerable waste of computer memory, especially as the total number of elements can be high, e.g. more than 4 million in the primary mesh in Section 6. A much more efficient way is available with *linked lists*, which consists of two one-dimensional arrays. The two arrays can be written as [38]

$$esup_1(mesup); esup_2(npoin + 1)$$

where $mesup$ is the total number of elements surrounding points. Let $nesup(i)$ be the number of elements surrounding point $i$, then $mesup = \sum_{i=1}^{nElem} nesup(i)$. $esup_1(nesup)$ stores all the elements surrounding each point and $esup_2(npoin + 1)$ the locations in the linked list of the first and last elements surrounding each point. For an arbitrary point $i$, the elements surrounding it are stored in $esup_1$ from location $esup_2(i) + 1$ to location $esup_2(i + 1)$. The algorithm for building $esup_1(mesup)$ and $esup_2(npoin + 1)$ is presented in page 12 in [38]. The idea is that each element is recognised as a surrounding element of all its nodes. Therefore we execute a loop over all elements and store them in $esup_1$ as their nodes’ surrounding elements.

2. The points surrounding each point are established, based on elements surrounding each point. The number of immediate neighbouring points of each point can also vary across an unstructured mesh, so linked lists are used to store this information. The lists can be denoted by

$$psup_1(mpsup), psup_2(npoin + 1)$$

where $psup_1$ stores the points surrounding each point and $psup_2$ the locations of the starting and ending points. The algorithm for constructing $psup_1$ and $psup_2$ is presented in page 14 in [38]. The data structure of elements sur-
rounding points are employed. For a point $i$, all its surrounding elements are searched to find all nodes of the elements. These nodes include point $i$ and all its immediate neighbours. Then, point $i$ itself and its repetitive neighbours are removed from this set of points, with the help of an array for marking points counted more than once. The algorithm for points surrounding each point is different from that for elements surrounding each point. We store points and count locations at the same time for points surrounding each point. However, for elements surrounding each point, the loop is over the elements while $esup2$ is based on points, therefore locations need to be calculated before storing elements in corresponding locations.

3. The edges are defined from the data structure of immediate neighbouring points surrounding each point. The edges can only be formed between neighbouring points. The method for building edges is shown in Appendix A.1. The key idea is to avoid storing an edge more than once.

4. The elements surrounding each element are identified with the help of the data structure of elements surrounding each point. There is one neighbouring element on each face of a primary-mesh element. The data structure is a $nelem \times 4$ matrix for a primary tetrahedral mesh, and the building process is presented in page 16 in [38]. The basic idea is that two neighbouring elements share a common face. For each element, a loop over all elements could be run to find neighbouring elements. However, this method is too computationally expensive for a mesh of millions of elements. Thus employing the data structure of elements surrounding each point, we can search over all elements surrounding the nodes of the element instead. Then repetitive elements are removed.

5. Next, the data structure of all edges originated from each point is established, in linked lists, to accelerate the process in the next step. For each point,
all edges are searched to match the point with the edges’ endpoints. The searching process can be accelerated by splitting the domain into sub-domains and only searching through the edges in the sub-domain to which the given point belongs.

6. Finally a data structure for elements surrounding each edge, in the form of linked lists, needs to be provided. The elements surrounding each edge are ordered in the clockwise or counter-clockwise direction. The building process is presented in Appendix A.2. For each element, a search over all edges is needed to recognise the edges of the element. This process is time-consuming for a fine mesh. In order to accelerate the process, a data structure for all edges from each node is established when building the edges, and is employed here. Thus, the search loop is now over all edges from the edge’s endpoints, as shown in Appendix A.3. After recognising the edge in the first element, the next element is one of the first element’s neighbours. The process continues until the next element returns to the first one.

Having defined the mesh in planar geometry, all geometric elements such as cell volume, cell face area, and normals are evaluated from vector calculus. All dependent variables are co-located in the nodes. In 2D, the dual-mesh polygonal cell can be split into small triangles. For example in Figure 1, triangle $icl$ is one of the small triangles constituting the dual mesh cell of point $i$. Thus, the area of the dual mesh cell can be calculated by summing up the areas of all small triangles. For triangle $icl$, the vector from point $i$ to point $c$ is $\vec{ic} = < x_c - x_i, y_c - y_i >$ and the vector from point $i$ to point $l$ is $\vec{il} = < x_l - x_i, y_l - y_i >$. The area of the triangle is given by

$$Area = |(x_c - x_i)(y_l - y_i) - (x_l - x_i)(y_c - y_i)| .$$ (92)
3 NUMERICAL SCHEME

Figure 3: Two dimensional dual mesh construction. The left mesh is the primary triangular finite element mesh and the right one is the corresponding dual finite volume mesh.

The faces of the dual-mesh cell are associated with edges. In Figure 1, edge $ij$ points from $i$ to $j$, $l$ and $r$ are on the edge’s left and right-hand-side, respectively. The dual-mesh cell face for edge $ij$ is composed of segments $rc$ and $cl$. Then the projections on $x \equiv constant$ and $y \equiv constant$ are denoted as $S_x$ and $S_y$, respectively, such that

$$S_x = y_l - y_r ,$$

$$S_y = x_r - x_l .$$

Figure 4: This figure illustrates a small tetrahedron 4123 inside a tetrahedral element 4576 of a primary mesh.

Figure 4 illustrates the 3D dual mesh construction based on a primary tetrahedral mesh. The big tetrahedral element 4576 of the primary mesh can be split into a series of small tetrahedra. An example of such tetrahedra is tetrahedron 4123.
This decomposition method applies to any type of 3D convex shape. Point 1 is the midpoint of the edge, 3 and 2 are barycentres of the triangular face 456 and the tetrahedral element 4576. All small tetrahedra surrounding point 4 are agglomerated to build the dual-mesh cell enclosing it.

![Figure 5: This figure illustrates the parallelepiped built on the three vectors](image)

The volume of tetrahedron 4123 is one sixth that of the parallelepiped built from vectors $\vec{12}$, $\vec{13}$ and $\vec{14}$, as shown in Figure 5. Denoting the coordinate of point $i$ as $(x_i, y_i, z_i)$ and the vector from point $j$ to $i$ as $r_{ij}$, the volume of the small tetrahedron 4123 is $\frac{VOLUME}{6}$ and $VOLUME$ is the volume of the parallelepiped given by

$$VOLUME = (r_{21} \times r_{31}) \cdot r_{41} = \det \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{bmatrix}.$$  \hspace{1cm} (95)

It can be noted that the volume of the parallelepiped can also be interpreted as

$$VOLUME = (x_4 - x_1) \cdot S_x + (y_4 - y_1) \cdot S_y + (z_4 - z_1) \cdot S_z,$$  \hspace{1cm} (96)

where $S_x$, $S_y$ and $S_z$ are projections of the parallelogram surface area created by vectors $r_{21}$ and $r_{31}$. The area is $|r_{21} \times r_{31}|$. So $S_x$ is the projection of the area $|r_{21} \times r_{31}|$ on the plane $x \equiv constant$. By analogy, $S_y$ and $S_z$ are the projections of
the same area on $y \equiv \text{constant}$ and $z \equiv \text{constant}$, respectively. In particular, $S_x$, $S_y$ and $S_z$ can be expanded as

\begin{align*}
S_x &= (y_2 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1), \quad (97) \\
S_y &= (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1), \quad (98) \\
S_z &= (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1). \quad (99)
\end{align*}

The triangle built on vectors $r_{21}$ and $r_{31}$ is shown in blue in Figure 4, and its projections are therefore $\frac{S_x}{2}$, $\frac{S_y}{2}$ and $\frac{S_z}{2}$. The blue triangle is part of the umbrella-like dual-mesh face shown in Figure 2. Thus the projections of the dual-mesh face around edge 45 is the sum of that for all blue triangles. The algorithm for calculating volume and projections in 3D is presented in Appendix A.4.

The boundary of a 3D primary tetrahedral mesh is a surface mesh consisting of triangles. The boundary of the corresponding dual mesh is composed of polygons. These polygons are called boundary faces. The surface mesh on each boundary can be treated as a 2D primary mesh and areas of the boundary faces are calculated similarly as 2D dual mesh. Then projections of boundary faces can be derived from their areas and normal vectors. However, for some particular computational domains, a more straightforward method is possible.

For any mesh point $i$, any 3D dual-mesh cell enclosing it is unique and can be denoted as $\text{cell}(i)$. $\text{cell}(i)$ is a closed shape, therefore the sum of projections should be zero,

\begin{align*}
\sum_{\text{all faces of } \text{cell}(i)} S_x &= 0, \quad (100) \\
\sum_{\text{all faces of } \text{cell}(i)} S_y &= 0, \quad (101) \\
\sum_{\text{all faces of } \text{cell}(i)} S_z &= 0. \quad (102)
\end{align*}
For any mesh point \( i \) at the boundary of the domain, the boundary face enclosing it is also unique, and can be denoted as \( bface(i) \). Thus the projections of \( bface(i) \) can be calculated as

\[
S_{x}^{bface(i)} = - \sum_{\text{all other faces of } cell(i)} S_x , \tag{103}
\]
\[
S_{y}^{bface(i)} = - \sum_{\text{all other faces of } cell(i)} S_y , \tag{104}
\]
\[
S_{z}^{bface(i)} = - \sum_{\text{all other faces of } cell(i)} S_z . \tag{105}
\]

If point \( i \) is at a corner of the computational domain, then it is shared by more than one boundary faces. Thus the above equations become

\[
\sum_{\text{all boundary faces of } cell(i)} S_x = - \sum_{\text{all other faces of } cell(i)} S_x , \tag{106}
\]
\[
\sum_{\text{all boundary faces of } cell(i)} S_y = - \sum_{\text{all other faces of } cell(i)} S_y , \tag{107}
\]
\[
\sum_{\text{all boundary faces of } cell(i)} S_z = - \sum_{\text{all other faces of } cell(i)} S_z . \tag{108}
\]

Then \( S_x \), \( S_y \) and \( S_z \) for each boundary face can be obtained with help of its unit normal vector.

Each boundary face is accompanied by a unit normal vector that points to the inside of the computational domain. For a cuboid geometry, the unit normal vectors of the six surfaces are straightforward. For boundary surface which is defined by a function \( z = f(x, y) \), the normal vectors are calculated numerically. Let the derivatives be evaluated as

\[
\frac{df}{dx} = \frac{f(x + dx, y) - f(x - dx, y)}{2dx}, \tag{109}
\]
\[
\frac{df}{dy} = \frac{f(x, y + dy) - f(x, y - dy)}{2dy}. \tag{110}
\]
then the unit normal vector \((n_x, n_y, n_z)\) is

\[
\begin{align*}
n_x &= -\frac{df}{dx} \cdot \frac{1}{\sqrt{(\frac{df}{dx})^2 + (\frac{df}{dy})^2 + 1}}, \\
n_y &= -\frac{df}{dy} \cdot \frac{1}{\sqrt{(\frac{df}{dx})^2 + (\frac{df}{dy})^2 + 1}}, \\
n_z &= \frac{1}{\sqrt{(\frac{df}{dx})^2 + (\frac{df}{dy})^2 + 1}}.
\end{align*}
\] (111) (112) (113)

A preliminary check following the dual mesh construction is necessary. The volume of any dual-mesh cell should be positive. Let \(vol(i)\) be the volume of point \(i\). For any point \(i\),

\[
vol(i) > 0. 
\] (114)

The sum of the volume of all dual-mesh cell should be the volume of the computational domain. Let \(npoin\) denote the total number of points, \(vol_{domain}\) be the volume of the computational domain and \(C\) be a tolerance value, e.g. \(10^{-5}\). Then we have

\[
|\sum_{i=1}^{npoin} vol(i) - vol_{domain}| < C. 
\] (115)

The sum of projections for each dual-mesh cell should be zero. So for each dual-mesh we have

\[
\sum_{all faces} S_x < C, 
\] (116)

\[
\sum_{all faces} S_y < C, 
\] (117)

\[
\sum_{all faces} S_z < C. 
\] (118)

This method of dual mesh construction can be generalised to any type of primary meshes. Unstructured primary meshes employed in the current work are triangular
or hybrid meshes in 2D, and tetrahedral, prismatic or hybrid meshes in 3D. Triangular meshes are generated via the advancing frontier technique, while tetrahedral mesh generation is based on a Delaunay triangulation principle. Prismatic mesh offer the option of refinement near the wall or boundary. In Sections 4 and 5, primary tetrahedral and prismatic meshes are generated in a cuboid geometry followed by a vertical mesh movement akin to the Gal-Chen transformation [85]. This allows the mesh to conform the vertical location of the shape of terrain. In Section 6, 7 and 8, 3D hybrid primary meshes are employed, consisting of prismatic layers near the sphere to resolve boundary layer flows, and tetrahedral elements elsewhere with varying spatial resolution. In Section 9, a mesh generator development is described for building primary bespoke meshes, which are 2D hybrid meshes consisting of triangles and rectangles, based on fixed reduced Gaussian points consistent with supported points used by the spectral method in the Integrated Forecast System in ECMWF.
4 Strongly Stratified Flow Past a Steep Isolated Hill

In contrast to modelling engineering flows, where simulations on tetrahedral and related meshes have achieved a high degree of maturity, their applicability to atmospheric flows is still being explored. While prismatic meshes are ideally suited for global models — for which the (relatively) thin atmosphere imposes stringent constraints on the design of numerical models — the unstructured tetrahedral discretisation can benefit small- and mesoscale models. For example, applications involving terrain so complex as caves and canyons cannot be easily resolved with continuous mappings. Moreover, as the tetrahedral discretisation provides means for accommodating irregular interfaces, it provides new avenues for study of cloud physics. On the other hand, atmospheric flow simulations pose new challenges for tetrahedral discretisation, because the underlying hydrostatic balance and physics (e.g., rainfall and radiation) are predominantly ordered in the vertical direction. Here, we probe the accuracy of tetrahedral discretisation against prismatic and Cartesian meshes, in the context of strongly stratified flow past a steep 3D mountain. For this purpose, we adopt the canonical problem of a low-Froude number flow past an axially-symmetric hill [29, 12, 60], simulated in [71] on dual meshes derived from structured and prismatic primary meshes.

The cosine hill defined as

\[
h(x, y) = \begin{cases} 
  h_0 \cos^2(\pi r/2L) & \text{if } r = ((x - x_0)^2 + (y - y_0)^2)^{1/2} \leq L \\
  0 & \text{otherwise ,}
\end{cases}
\]

where the half-width \( L = 3000 \) m and height \( h_0 = 1500 \) m, is centred at the bottom of the computational domain. The domain size is \( 5L \times 4L \times 2L \) in \( x \), \( y \) and \( z \) directions, respectively. Two primary meshes consisting of tetrahedral elements were generated in the domain. The first one (not shown) using a uniform background point spacing
of $\delta x = \delta y = \delta z = 120$ m giving 1945090 of the total number of points, and the second one with the varying point spacing. Figure 6 shows two cross-sections of the primary mesh that consists of 337510 points and uses varying resolution ranging from 450 m at the boundaries to about 100 m in the hill’s vicinity. The number of points in the varying resolution tetrahedral mesh is not only substantially lower than in the constant resolution tetrahedral mesh but also lower than in the prismatic (refined in the horizontal only) and Cartesian meshes used for this problem in [71]. In order to produce solutions of matching quality, those meshes consisted of 692533 and 1121812 computational points respectively.

The governing equations (1–3) assume Boussinesq limit, with the constant potential temperature of the reference state $\Theta_o = 300K$ and the ambient state charac-
Figure 7: $Fr = 1/3$ flow solution after two advective time scales $T = L/U$. Contours of vertical velocity in central $xz$ cross section $y = 0$ (top) and in the $xy$ cross-section at $z = h_0/3$ (bottom). The contour interval is 0.5 ms$^{-1}$, and positive/negative contours are presented with solid/dashed lines; the zero contours are not displayed.

With the specified hill geometry, the ambient conditions result in a low Froude number, $Fr = U/Nh_0 = 1/3$, flow. Because $h_0/L \sim O(1)$, the problem is essentially nonhydrostatic and can be compared to experimental results given in [29]. All results are shown after two advective time scales $T = L/U$ ($t = 1200s$) when the main features of the solution are already established. The initial condition is provided by the solution of the potential flow problem, with a gradient of the potential perturbation imposed on the ambient wind. While the boundary conditions are rigid in $x$, $y$ and $z$, the gravity-wave absorbers near the upper and lateral streamwise boundaries
attenuate the solution toward ambient profiles with absorbing coefficient increasing linearly from zero at the distance $L/2$ from the boundary to $150^{-1}$ s$^{-1}$ at the boundary. A description of absorbers can be found in Section 2.6.

Figure 8: As in the bottom panel of Figure 7 but at $z = (5/3) h_0$ horizontal cross-section.

The flow patterns displayed in figures 7 and 8 are computed for the varying resolution mesh, and show key features of a low Froude number flow, including the characteristic separation and reversal of the lower upwind stream, and the formation of intense vertically-oriented vortices on the lee side of the hill [29, 60, 12], with the flow aloft transitioning to the linear gravity wave response [57]. In Figure 7, in the central $xz$ cross-section at $y = 0$, a turbulent wake is formed in the lee-side of the hill and characteristic gravity waves response is visible above the wake. This result matches closely the reference solution obtained on structured and prismatic meshes (upper panels in Figs. 6 and 8 in [71]). Figure 7, in $xy$ cross-section at $z = (1/3) h_0 = 500$ m, presents a pair of eddies behind the hill, showing the intrinsic three-dimensionality of the lee-side flow. In brief, the reason of this flow structure is that the incoming flow up to $z_c \approx (1 - Fr) h_0$ — the so called dividing streamline [29] — is forced to deflect and split around to the hill as it lacks sufficient kinetic
energy to go over the hill. Above the dividing streamline the hill is sufficiently low for the flow to go over, thus resulting the characteristic gravity response aloft. Figure 8 shows the horizontal structure of the gravity wave for the $xy$ cross-section at $z = (5/3)h_0 = 2500$ m. The corresponding solutions for the regular tetrahedral mesh are not shown, because their departures from the results in Figs. 7 and 8 are insignificant. The two horizontal flow patterns also compare well with the reference results in [71].

The statistics of the results are presented in Tables 1 and 2 for the tetrahedral meshes with constant and varying point resolution, respectively. In the tables, the units of the three velocity components are in $\text{ms}^{-1}$ and the potential temperature in K. The statistics of $\Theta'$ are converted to the vertical displacements of isentropes using a crude approximation

$$\eta_z \approx -\Theta' \frac{g}{N^2 \Theta_o} \approx -333 \Theta' \text{ m}.$$  \hfill (120)

The statistics of dependent-variable fluctuations about the ambient state are evaluated over the entire computational domain. The irregularity of the mesh is taken into account when calculating the averages and standard deviations of the fluctuations. With $\vartheta_i$ denoting the cell volume surrounding mesh point $i$, the average values $\overline{\psi'}$ are calculated according to

$$\overline{\psi'} = \frac{\sum_{i=1}^{n} (\psi'_i \vartheta_i)}{\sum_{i=1}^{n} \vartheta_i},$$ \hfill (121)

and the corresponding standard deviations $\sigma_{\psi'}$ are calculated as

$$\sigma_{\psi'} = \sqrt{\frac{\sum_{i=1}^{n} ((\psi'_i - \overline{\psi'})^2 \vartheta_i)}{\sum_{i=1}^{n} \vartheta_i}},$$ \hfill (122)

where $n$ denotes the total number of nodes in the domain. The statistics for the two
tetrahedral meshes as well as those provided for the refined prismatic mesh (Table 2 in [71]) are in close agreement.\(^3\)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Maximum</th>
<th>Minimum</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u')</td>
<td>3.51</td>
<td>-11.39</td>
<td>0.06</td>
<td>0.76</td>
</tr>
<tr>
<td>(v')</td>
<td>8.04</td>
<td>-8.48</td>
<td>(4.7 \times 10^{-6})</td>
<td>0.47</td>
</tr>
<tr>
<td>(w')</td>
<td>5.35</td>
<td>-4.80</td>
<td>(-1.02 \times 10^{-4})</td>
<td>0.36</td>
</tr>
<tr>
<td>(\Theta')</td>
<td>2.85</td>
<td>-1.48</td>
<td>(-4.7 \times 10^{-3})</td>
<td>0.20</td>
</tr>
<tr>
<td>(\eta')</td>
<td>-949</td>
<td>493</td>
<td>1.57</td>
<td>66.6</td>
</tr>
</tbody>
</table>

Table 1: Fluctuations’ statistics on a tetrahedral mesh with constant resolution

<table>
<thead>
<tr>
<th>Variable</th>
<th>Maximum</th>
<th>Minimum</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u')</td>
<td>3.42</td>
<td>-11.33</td>
<td>0.06</td>
<td>0.78</td>
</tr>
<tr>
<td>(v')</td>
<td>8.42</td>
<td>-8.26</td>
<td>(1.93 \times 10^{-6})</td>
<td>0.48</td>
</tr>
<tr>
<td>(w')</td>
<td>5.56</td>
<td>-4.76</td>
<td>(-6.40 \times 10^{-5})</td>
<td>0.36</td>
</tr>
<tr>
<td>(\Theta')</td>
<td>2.93</td>
<td>-1.52</td>
<td>(-6.99 \times 10^{-3})</td>
<td>0.20</td>
</tr>
<tr>
<td>(\eta')</td>
<td>-976</td>
<td>506</td>
<td>2.33</td>
<td>66.6</td>
</tr>
</tbody>
</table>

Table 2: Fluctuation’s statistics on a tetrahedral mesh with varying resolution

\(^3\)The \(u_e = 5 \text{ ms}^{-1}\) was not subtracted from the values of \(u\) given in [71].
5 Stratified Turbulent Shear Flows with Critical Levels

Section 4 examined the efficacy of variable-resolution flexible meshes for simulating orographically forced gravity-wave response in canonical media with uniform wind and constant stratification. However, in the atmosphere, wind and stratification generally vary with height, and this can profoundly affect the response. This is particularly evident for ambient flows with critical levels, identified by the horizontal wave phase speed equal to the ambient velocity component in the direction of the wave propagation. In steady mountain waves like those discussed in Section 4 the phase speed is zero, so the critical level is where the ambient wind vanishes. The influence of the critical levels on the gravity waves propagation is widely discussed in the literature; see \[2, 8, 48, 21, 7, 86\] and references therein. Theoretically, as the wave packet approaches a critical level the group velocity and the vertical wavelength tend to zero, while the horizontal velocity grows unboundedly, and the effective wave absorption occurs below the critical level (i.e., within the critical layer). This singularity of the linear predictions indicates that processes acting on small scales become important in critical layers; in particular, the nonlinear steepening and overturning of the waves. In 3D, this is a challenge for structured grids, which require uniformly high resolution to gain insights into the morphology of critical layers.

Here, the anelastic equations 1 to 3 are integrated on highly anisotropic and inhomogeneous mesh with high resolution concentrated in a small portion of the computational domain. The geometry of the problem and flow conditions are selected to reproduce the numerical experiment defined in \[21\]. The domain is \(100 \times 100 \times 3\) km\(^3\), with an axially symmetric mountain

\[ h(r) = h_0 \left(1 + \frac{r^2}{a^2}\right)^{-3/2}, \quad r \equiv \sqrt{x^2 + y^2}, \tag{123} \]
centred at the bottom of the domain. The ambient velocity profile with a reverse linear shear is defined as

\[ U(z) = U_0 \left(1 - \frac{z}{z_c}\right), \]  

(124)

where \( z_c \) denotes the altitude of the critical level. Free-slip impermeable boundary conditions are applied at the vertical and spanwise lateral boundaries. At the streamwise lateral boundaries the ambient profile is assumed. The absorbing layers are employed in the vicinity of the upper and streamwise lateral boundaries with respective thickness of 1 and 10 km. The inverse timescale of the absorbers is \( 150^{-1}\text{s}^{-1} \) at the boundaries.

For this problem we employ a primary unstructured mesh built from distorted prisms with triangular bases. This is because the computational domain is relatively thin in the vertical, and the use of prismatic mesh allows a considerably finer resolution in the vertical direction. To construct such a mesh, firstly, a planar triangular mesh is generated that defines the \( x \) and \( y \) coordinates for all nodes. Secondly, the layers of prisms are stack in the vertical, such that the vertical position of the node is specified as

\[ \tilde{z}_{i,k} = \tilde{z}_{i,k-1} + \delta \tilde{z}_k. \]  

(125)

Here, \( i = 1, 15059 \) numbers the nodes in each prismatic layer, \( k = 1, 70 \) numbers the layers, \( \tilde{z}_{i,k} \) is the height of the layer with respect to the flat bottom, \( \delta \tilde{z}_k \) is the increment of the \( k \)th layer, with \( \tilde{z}_{i,0} = 0 \) and \( \delta \tilde{z}_1 = 0 \). Finally, the layers are elevated according to the shape of the mountain by displacing \( z \) coordinate of the mesh to mimic the terrain-following coordinate transformation \([17, 85]\)

\[ z_{i,k} = \tilde{z}_{i,k} \left(1 - \frac{h_i}{H}\right) + h_i, \]  

(126)
where \( h_i \) is the height of the mountain at the \( i \)th node in each layer and \( H = 3 \) km is the top of the model domain. The horizontal grid size is approximately 150 m in the vicinity of the hill and gradually grows larger to be around 3.5 km near the boundary of the domain. To ensure adequate geometry representation, a finer point resolution is applied where the mountain slope is higher. The vertical increment \((\delta \tilde{z}_k)\) grows gradually from 30 m close to or below the critical level to 100 m near the top of the domain. Figures 9 and 10 show, respectively, a representative primary surface mesh for the lateral and top surfaces of the model the domain, and a fragment of the bottom-surface mesh conforming to the hill.

Figure 9: The surface mesh of the prismatic volumetric mesh.

Following [21], the ground-level wind speed \( U_0 = 10 \) ms\(^{-1}\), the buoyancy frequency \( N = 0.01 \) s\(^{-1}\), the half-width of the hill \( a = 5000 \) m, and the position of the critical level \( z_c = 1000 \) m. Consequently, the ratio of the horizontal to the vertical wave number of the hydrostatic mountain wave forced by the hill is \( U_0/Na = 0.2 \) and the Richardson number \( Ri = (Nz_c/U_0)^2 = 1 \). The ambient shear flow is thus hydrostatic and stable. Four key benchmarks for different mountain heights have
Figure 10: A zoomed fragment of the horizontal triangular surface mesh at the bottom of the domain.

Table 3: Parameters of the selected numerical experiments, LS2-LS5 after citecritical.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Ri</th>
<th>( \hat{h} )</th>
<th>( D/D_0(6) )</th>
<th>( D/D_0(10) )</th>
<th>( D/D_0(18) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS2 (linear)</td>
<td>1</td>
<td>0.05</td>
<td>0.923</td>
<td>0.954</td>
<td>0.997</td>
</tr>
<tr>
<td>LS3 (nonlinear)</td>
<td>1</td>
<td>0.1</td>
<td>0.983</td>
<td>1.02</td>
<td>1.160</td>
</tr>
<tr>
<td>LS4 (nonlinear)</td>
<td>1</td>
<td>0.2</td>
<td>1.04</td>
<td>1.15</td>
<td>1.26</td>
</tr>
<tr>
<td>LS5 (nonlinear)</td>
<td>1</td>
<td>0.3</td>
<td>1.09</td>
<td>1.22</td>
<td>1.23</td>
</tr>
</tbody>
</table>

been selected from those studied in [21]: their LS2 for a linear case; and LS3, LS4 and LS5 for nonlinear cases. The term "linear" or "nonlinear" is read from Figure 8 in [21]. The magnitudes of a nondimensional mountain height \( \hat{h} = h_0 N/U_0 \) (viz. the inverse Froude number) are specified by varying \( h_0 \). All four experiments summarised in Table 3 were run until the dimensionless time \( T = tU_0/a = 18 \). The mountain wave drag \( D(T) \) is calculated numerically as \( D = -\int_{\Gamma} p' n_x dS \), where \( \Gamma \) denotes the mountain surface, and \( n_x \) is the \( x \) component of the unit vector normal to \( \Gamma \). The numerical drag \( D \), listed in the table at \( T = 6, 10 \) and 18, is normalised by the analytic hydrostatic drag for uniform ambient flow, \( D_0 = \frac{2}{\pi} \rho_0 N U_0 a h_0^2 \) [57].

Figure 11 shows the histories of normalised drag, corresponding to those in Figure
Figure 11: Normalised drag history. The linear case is shown in solid line, while nonlinear cases are in dashed lines.

9 of [21] for the LS2-LS5 experiments. It should be noted that in [21] all experiments were initialised by growing the mountain gradually over the first $T = 8$ time units; whereas in the present study, the initial conditions assume ambient wind with a potential perturbation imposed to satisfy the mass continuity equation 1 together with the boundary conditions on the fixed full-size mountain. In consequence, our results are substantially advanced in the dimensionless time compared to [21], and the details of this advancement depend on the hill size. These are of minor importance for the linear case that approaches a steady state, but are more consequential for the nonlinear cases that depart steadily from the linear results. Overall, the drag histories in Figure 11 reproduce the dependence of the drag on the amplitude of the lower-boundary forcing reported in [21].

Figure 12 shows the isentropes in the $y = 0$ vertical plane at $T = 6$ for LS2-LS5 experiments. These solutions compare well with Figure 13(a) and Figs. 16(a), (c), (d) at $T = 18$ in [21]. As anticipated, they show that there is no significant propagation of the wave energy throughout the critical level. Furthermore, for sufficiently large amplitude of incoming waves nonlinearity dominates viscous (ILES) processes in the layer. This results in the wave steepening and overturning beneath the critical layer. However, the flow below the critical layer remains essentially decoupled from the flow aloft. Figure 13 highlights the dependence of the solution on the magnitude
Figure 12: Isentropes at T=6 in $y = 0$ vertical plane for experiments LS2 (left, top), LS3 (right, top), LS4 (left, bottom) and LS5 (right, bottom).

of the lower boundary forcing and time; it corresponds to Figure 12 with the results displayed at the later time $T = 18$. Comparing the corresponding panels in Figs. 13 and 12 shows that the flow stability diminishes with increasing amplitude of the mountain wave. Furthermore, it demonstrates that given a sufficiently long time and low viscosity even the ”linear” wave overturns right beneath the critical layer. The top two panels in Figure 13 show solutions qualitatively similar to their earlier forms, with the one on the right transitioning to turbulence but still decoupled from the flow aloft. This is corroborated in the upper panels in Figure 14 and Figure 15 that complements these two panels with the display of streamlines that capture the characteristic Kelvin’s ”cat’s eye” circulation [48] in LS2 run but becoming irregular in LS3 experiment at $T = 18$. In contrast, the two bottom panels of Figure 13 already evince massive wave breaking with turbulent flow in the lee and evanescent perturbations, excited by convective eddies, penetrating through the critical layer. The evolution of LS5 in streamlines is illustrated in the bottom panels in Figure 14 and Figure 15. Figure 16 shows the change of velocity vectors and isolines of
streamwise velocity component perturbation for LS5. Notably, all these results may be viewed as a 3D paraphrases of the 2D flow discussed in section 5 of [69].

Figure 13: As in Figure 12 but at T=18.

Figure 14: Streamline corresponding to the isentrope display in Figure 12.
Figure 15: Streamline corresponding to the isentrope display in Figure 13.

Figure 17 shows the isentropic surface at $T = 6$ and $T = 18$ for the run LS2, roughly corresponding to the first layer below $z = 1$ km in the top-left panel in Figure 12 and 13, respectively. It can be clearly seen that in Figure 17 as the deflection of the isentropic surface is steepened vertically, it also grows horizontally. The corresponding vorticity pattern is illustrated in Figure 18, where a pair of counter rotating eddies resulting from wave-mean flow interaction can be observed. These flow features are similar to those reported in Figure 14 in [21].

Figure 19 shows evolving isentropic surfaces, defined by their undisturbed height $z = 0.94z_c$, at the dimensionless times $T = 6$ and 18 for the LS5 experiment. Figure 20 presents an enlargement of Figure 19. At $T = 6$, the top panel of Figure 20 roughly corresponds to the first isentrope beneath $z = 1$ km in the bottom-right panel of Figure 12. It supplements the latter with a highlight of the 3D structure of the overturning mountain wave. At $T = 18$, the lower panel of Figure 20 evinces an abundance of fine scale features reflecting convective eddies in the turbulent wake forming in the lee. This nature of the solution substantiates two important aspects
Figure 16: Contour plots of stream-wise velocity component perturbation $u - u_e$ on $y = 0$ plane with velocity vectors for LS5 at $T = 6$ (top) and $T = 18$ (bottom). Negative values are presented in dashed lines.
Figure 17: Isentropic surface with undisturbed height $z = 0.94z_c$ for the run LS2 at $T = 6$ (top) and $T = 18$ (bottom).
Figure 18: Vertical vorticity isolines for the run LS2 at $T = 6$ on $x = a/4$ plane (top) and on $z = 0.94z_c$ plane (bottom)
Figure 19: Isentropic surface with undisturbed height $z = 0.94z_c$ at $T = 6$ (top) and $T = 18$ (bottom) for the run LS5.
Figure 20: A fragment of the isentropic surface with undisturbed height $z = 0.94z_c$ at $T = 6$ (top) and $T = 18$ (bottom) for the run LS5.
of this work. First, it attests to the utility of the flexible meshing admitting highly
anisotropic and inhomogeneous varying resolution in regions of interest. Second, it
corroborates the utility of ILES approach assuring nonlinearly stable solutions on
variable meshes without the need for explicit subgrid-scale (SGS) models. Note-
worthy, high resolution is required to capture intermittent turbulence in the lee of
the mountain, but variable resolution complicates explicit SGS modelling already at
the theoretical level as it invalidates the commutativity of SGS filtering and finite-
volume differencing — one of the key assumptions underlying basic LES models.
In contrast to explicit LES, ILES naturally selects the variable size of the filter
and imposes no dissipative stability constraints controlled by the square of the local
resolution, cf. Section 2.5.
6 Stratified Laminar Flows Past a Sphere

An important test case for engineering and atmospheric flows is flow past a sphere. Lofquist and Purtell [39] measured the drag on a sphere moving horizontally through stratified flows experimentally over a wide range of Reynolds and Froude numbers, and provided quantitative data for corresponding drag coefficients. Lin, et al. [35] studied stratified flows past a sphere experimentally and visualized a rich range of characteristic flow phenomena for Froude number $Fr \in [0.005, 20]$ and Reynolds number $Re \in [5, 10000]$. Hanazaki [23] investigated stratified flows past a sphere numerically at $Re = 200$. His results are obtained on a structured grid that extends in the radial direction. Although high resolution can be achieved near the sphere surface, the mesh in the wake is relatively coarse. In this section, we use an unstructured mesh to obtain high resolution both close to the sphere surface and in the wake. This leads to a more accurate drag calculation and flow pattern in the wake. Furthermore, the use of unstructured meshes in general also allows for a more convenient application of the numerical method to engineering flows with complex geometries.

6.1 The Numerical Experiment

The computation domain for this numerical experiment is a $20 \times 20 \times 20$ box. A sphere of diameter $D = 2r = 1$ is located in the centre of the domain. A tetrahedral mesh is built in the domain except within one diameter distance to the sphere surface. Prismatic layers are built on the basis of the triangular surfaces of the tetrahedral mesh to ensure higher resolution for simulating the boundary layer flow. A one-dimensional stretching function is introduced to define the prismatic layers such that mesh points are clustered in radial direction close to the sphere. If the number of prismatic layers is $n$ and the $i$’th prismatic layer is considered, then the
stretching function [13] is written as

\[ s = P\eta + (1 - P) \left( 1 - \frac{\tanh[Q(1 - \eta)]}{\tanh Q} \right), \] (127)

where \( P \) and \( Q \) are parameters to control the position of the layers. \( P \) defines the slope of the distribution and \( Q \) controls the departure from linearity. In the current study, we use \( P = 1.72, Q = 2.00 \) and \( n = 24 \). \( \eta \) controls the order of layers according to \( \eta = i/n \). Let \( A \) denote the points on the triangular faces where prismatic layers are built on, and \( E \) the points on the last layer, then the coordinates of the point on \( i \)'th layer are calculated as

\[ x_i = x_A + s(x_E - x_A), \] (128)
\[ y_i = y_A + s(y_E - y_A), \] (129)
\[ z_i = z_A + s(z_E - z_A). \] (130)

Figure 21 shows the \( y = 0 \) cross-section of the primary hybrid mesh in the computational domain and the prismatic layers around the sphere. The governing equations are solved on the corresponding dual finite volume mesh. Table 4 shows the statistics of the primary hybrid mesh. The prismatic layer closest to the sphere is located at a distance of 0.0096 from the sphere surface. The depth of the boundary layer on the sphere surface when \( Re = 200 \) for homogeneous flow is estimated by \( D/Re^{0.5} = 1/200^{0.5} \approx 0.07 \) [23]. It is calculated that in the boundary layer there are 8 points in the direction normal to the sphere surface.

<table>
<thead>
<tr>
<th>Number of Points</th>
<th>Number of Elements</th>
<th>Number of Prismatic Layers</th>
<th>Shortest Distance to the Sphere</th>
<th>Average Edge Length on the Sphere</th>
</tr>
</thead>
<tbody>
<tr>
<td>770331</td>
<td>4137195</td>
<td>24</td>
<td>0.009</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 4: Statistics of the mesh for stratified flows past a sphere

The governing equations adopt the anelastic system (1) to (3) with Boussinesq approximation. The integration method is the NFT MPDATA scheme (87). A free-stream velocity \( \mathbf{v}_e = (U,0,0) = (1,0,0) \) is applied at the inlet and outlet.
boundaries. A non-slip boundary condition is applied on the sphere surface, while free-slip condition is applied on all other boundaries. Absorbers are applied on all boundaries except the sphere surface. The absorbing coefficient increases linearly from zero at the distance \( r \) from the boundary to \( 150^{-1} \text{s}^{-1} \) at the boundary. The initial condition is the potential flow solution.

The governing equations are solved subject to the initial and boundary conditions for a fixed Reynolds number and a range of Froude numbers. The kinetic viscosity is set to be 0.005 so that the Reynolds number is fixed such that \( Re = \frac{UD}{\nu} = 200 \). Since \( U \) and \( r \) are both constant, the Froude number \( Fr = \frac{U}{Nr} \) is changed by varying \( N \). In the experiments \( Fr \in [0.25, 200] \).

Figure 21: (left) The \( y = 0 \) cross-section of the primary mesh for stratified laminar flow past a sphere. It is a hybrid mesh comprising prismatic layers near the sphere and tetrahedral elements elsewhere. Prismatic layers are built on the basis of the triangular faces of the tetrahedra and finer resolution is applied close to and behind the sphere. Computations are on the dual finite volume mesh. (right) Enlargement showing details of the prismatic layers near the sphere. Note that due to limited plotting capability, the prismatic cells in the primary hybrid mesh had to be changed to tetrahedra for this display.
6.2 Results

6.2.1 Flow Patterns

The computation is continued until almost steady state, which requires about $T = \frac{Ut}{D} = 30$ where $t$ is the physical time. This implies that the flow has travelled a distance of $30D$. Figures 23–29 illustrate the flow patterns for a range of Froude numbers, using contour, streamline and velocity vector plots on the central vertical ($y = 0$) and horizontal ($z = 0$) cross-sections. They compare well with Figure 3 in [23]. The lee-wave suppression of separation, reported in [39], is observed by measuring the separation angles in velocity vector plots. The separation angle is between the line connecting the upstream stagnation point ($-0.5, 0.0, 0.0$) with the centre of the sphere, and the line connecting the centre with the lee-side stagnation point. The values of horizontal and vertical separation angles observed in Figures 23–29 compare well with Figure 31, 32 in [40] and Figure 6 in [39]. The separation angles for different cases are observed in the original velocity vector plots without interpolating. For example, Figure 22 shows the enlarged velocity vector plots near the sphere on $y = 0$ plane for the non-stratified case, where the separation angle is around $95^\circ$. In Figures 23 to 29 the velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.

Figure 23 represents a limit when the flow is non-stratified. Hereafter the non-stratified case is labelled as homogeneous. An axisymmetric standing eddy exists behind the sphere and the flow near the central line ($y = z = 0$) is advected back towards the sphere. The contour plots, streamline patterns, and the velocity vector plots on both $y = 0$ and $z = 0$ planes are almost identical. There are separations on both planes, which can be clearly seen on the two velocity plots. The separation angles on both planes are around $95^\circ$. Since the eddy is axisymmetric, the lee-side flow is divided into four identical regions by the $y = 0$ and $z = 0$ planes.

Figure 24 shows the flow patterns for Froude number $Fr = 200$. The stratification is very weak and the flow patterns shown in Figure 24 are almost the same as
those in Figure 23 for homogeneous flow. The separation angle is also close to 120°. The dynamic pressure distribution around the sphere is illustrated in Figure 35 (a). The dynamic pressure $p_d$ is equal to the total pressure subtracted by the pressure due to hydrostatic balance. Since the ambient state in the governing equations (1–3) is hydrostatically balanced, $p_d$ can be evaluated as $p_d = p - p_e = \rho_0 p'$. The solid line for the dynamic pressure distribution in $y = 0$ plane and the dashed line for the distribution in $z = 0$ plane overlap with each other. This is consistent with the fact that the vertical velocity contour, streamline pattern and velocity vector plots on both planes are almost identical.

Figure 25 illustrates the case $Fr = 2$. Still, almost all flow particles near the central line $y = z = 0$ climb vertically over the sphere. However, compared to $Fr = 200$, the influence of stratification is much more significant. The lee-side axisymmetric standing eddy no longer exists. There is a small-scale longitudinal symmetric lee-side eddy and lee-side waves with very large wavelength in the vertical cross-section.
The flow is vertically stratified and the lee-wave is induced by the buoyancy force. In the horizontal plane, the amplitude of the lee-side eddy is larger and there is no wave motion. In addition, there is adverse flow behind the sphere because of the remaining lee-side vortex. The separation line in the vertical plane moves backward and the position of the minimum pressure moves downstream compared to $Fr = 200$, c.f. Figure 35 (b). The vertical separation angle is about 105° and the horizontal separation angle is around 95°.

Figure 26 shows the flow patterns for $Fr = 1$. The standing eddy collapsed completely both vertically and horizontally. Compared to $Fr = 2$, the length of the vertical lee-waves on $y = 0$ plane is reduced due to stronger stratification; and there is still no wave motion on $z = 0$ plane. The flow near the central line is advected away from the sphere surface and there is no separation both vertically and horizontally. The flow is up-down symmetric as the stratification is linear. At this Froude number, the amplitude of the lee-wave is the greatest, as can be seen from the contour and velocity plots on $y = 0$ plane. The amplitude of the lee-wave is attenuated along the flow direction due to dissipation.

As the Froude number decreases to 0.7, c.f. Figure 27, the lee-wave still remains but is shorter than the one for $Fr = 1$. The wavelength is reduced further as stratification is increased. The eddies behind the sphere appear again on both $y = 0$ and $z = 0$ planes, as can be seen near position $(1.4, 0, 0.15)$ in Figure 27 (c). There is an overturning eddy motion as well as recirculation under the first crest of the lee-wave. After the second crest, the lee-wave gradually disappears due to dissipation. In addition, there is separation again both vertically and horizontally, because of the adverse pressure gradient. The adverse pressure gradient on the vertical plane is the result of the interaction between the wave motion and overturning vortex behind the sphere, while that on the horizontal plane is induced by the vortex only. The vertical and horizontal separation angles are about 145° and 115°, respectively.

As the Froude number decreases to 0.5, c.f. Figure 28, the overturning eddy
Figure 23: Contour, streamline and vector plots of velocity \( \mathbf{v} = (u, v, w) \) for stratified flow past a sphere. The flow is non-stratified and the Reynolds number is 200. The interval between two neighbouring contours is 0.1. (a) the contour plot of vertical velocity component \( w \) at \( y = 0 \) plane. (b) the contour plot of velocity component \( v \) at \( z = 0 \) plane. (c) the streamline pattern at \( y = 0 \) plane. (d) the streamline pattern at \( z = 0 \) plane. (e) velocity vector plot at \( y = 0 \) plane. (f) velocity vector plot at \( z = 0 \) plane. The velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.
Figure 24: Contour, streamline and vector plots of velocity $\mathbf{v} = (u, v, w)$ for stratified flow past a sphere. The Froude number is 200 and the Reynolds number is 200. The interval between two neighbouring contours is 0.1. (a) is the contour plot of vertical velocity component $w$ at $y = 0$ plane. (b) is the contour plot of velocity component $v$ at $z = 0$ plane. (c) the streamline pattern at $y = 0$ plane. (d) the streamline pattern at $z = 0$ plane. (e) velocity vector plot at $y = 0$ plane. (f) velocity vector plot at $z = 0$ plane. The velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.
Figure 25: Contour, streamline and vector plots of velocity \( \mathbf{v} = (u, v, w) \) for stratified flow past a sphere. The Froude number is 2 and the Reynolds number is 200. The interval between two neighbouring contours is 0.1. (a) is the contour plot of vertical velocity component \( w \) at \( y = 0 \) plane. (b) is the contour plot of velocity component \( v \) at \( z = 0 \) plane. (c) the streamline pattern at \( y = 0 \) plane. (d) the streamline pattern at \( z = 0 \) plane. (e) velocity vector plot at \( y = 0 \) plane. (f) velocity vector plot at \( z = 0 \) plane. The velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.
Figure 26: Contour, streamline and vector plots of velocity $\mathbf{v} = (u, v, w)$ for stratified flow past a sphere. The Froude number is 1 and the Reynolds number is 200. The interval between two neighbouring contours is 0.1. (a) is the contour plot of vertical velocity component $w$ at $y = 0$ plane. (b) is the contour plot of velocity component $v$ at $z = 0$ plane. (c) the streamline pattern at $y = 0$ plane. (d) the streamline pattern at $z = 0$ plane. (e) velocity vector plot at $y = 0$ plane. (f) velocity vector plot at $z = 0$ plane. The velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.
Figure 27: Contour, streamline and vector plots of velocity $\mathbf{v} = (u, v, w)$ for stratified flow past a sphere. The Froude number is 0.7 and the Reynolds number is 200. The interval between two neighbouring contours is 0.1. (a) is the contour plot of vertical velocity component $w$ at $y = 0$ plane. (b) is the contour plot of velocity component $v$ at $z = 0$ plane. (c) the streamline pattern at $y = 0$ plane. (d) the streamline pattern at $z = 0$ plane. (e) velocity vector plot at $y = 0$ plane. (f) velocity vector plot at $z = 0$ plane. The velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.
Figure 28: Contour, streamline and vector plots of velocity $\mathbf{v} = (u, v, w)$ for stratified flow past a sphere. The Froude number is 0.5 and the Reynolds number is 200. The interval between two neighbouring contours is 0.1. (a) is the contour plot of vertical velocity component $w$ at $y = 0$ plane. (b) is the contour plot of velocity component $v$ at $z = 0$ plane. (c) the streamline pattern at $y = 0$ plane. (d) the streamline pattern at $z = 0$ plane. (e) velocity vector plot at $y = 0$ plane. (f) velocity vector plot at $z = 0$ plane. The velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.
Figure 29: Contour, streamline and vector plots of velocity $\mathbf{v} = (u, v, w)$ for stratified flow past a sphere. The Froude number is 0.25 and the Reynolds number is 200. The interval between two neighbouring contours is 0.1. (a) is the contour plot of vertical velocity component $w$ at $y = 0$ plane. (b) is the contour plot of velocity component $v$ at $z = 0$ plane. (c) the streamline pattern at $y = 0$ plane. (d) the streamline pattern at $z = 0$ plane. (e) velocity vector plot at $y = 0$ plane. (f) velocity vector plot at $z = 0$ plane. The velocity vector plots are shown after interpolated onto a relatively coarse Cartesian mesh.
motion under the first crest of the lee-wave grow larger while the amplitude of
the lee-waves become smaller. The vortex in the wake on $z = 0$ plane becomes
a two-dimensional double vortex and the flow near the stream-wise centre line is
advected backwards. This flow pattern is termed 'lee-wave instability' [35]. The
flow particles with sufficient kinetic energy go over the sphere and descend further
because of a negative buoyancy force. This leads to the narrowing of the separation
region vertically. Consequently, the horizontal separation angle ($\approx 120^\circ$) is smaller
than the vertical one ($\approx 135^\circ$).

At Froude number $Fr = 0.25$, illustrated in Figure 29, the vertical wave motion
become weaker, while the horizontal eddies become larger. The reason is that, as
stratification becomes stronger, fewer flow particles have sufficient kinetic energy to
overcome the potential energy required to reach the top of the sphere, and more
flow particles tend to go around the sphere instead. Thus, the flow behave in a
more two-dimensional way and become more horizontal. The separation line moves
upstream both vertically and horizontally. The horizontal separation angle becomes
about $75^\circ$, while the vertical one is around $125^\circ$.

Figure 30 shows the out-of-plane (normal to the plane shown) vorticity com-
ponent contour plots for the homogeneous flow case. The contours on the central
vertical and horizontal planes are identical and compare well with Figure 6 in [23].
The position of maximum out-of-plane vorticity is upstream of the separation point,
near the sphere surface. If the flow is stratified with $Fr = 0.5$, as in Figure 31,
the maximum out-of-plane contour position moves downstream along with the sep-
aration point both vertically and horizontally, compared to the homogeneous flow
case. The 3D structure of the vortices is illustrated in Figure 32. The region with
greater vorticity magnitude is closer to the sphere. Moreover, it is clear that the
isentropic surfaces are axisymmetric in the homogeneous flow case, but are up-down
symmetric in the stratified case. Another effect of stratification is that more vortices
tend to pass the sphere vertically, reaching further downstream.
Figure 30: Out-of-plane vorticity component contour plots for the homogeneous flow case and $Re = 200$ on $y = 0$ (top) and $z = 0$ planes (bottom).
Figure 31: Out-of-plane vorticity component contour plots for $Fr = 0.5$ and $Re = 200$ on $y = 0$ (top) and $z = 0$ planes (bottom).
Figure 32: 3D vorticity magnitude iso-surfaces for the homogeneous case (left-hand-side panels) and $Fr = 0.5$ (right-hand-side panels). In both cases, $Re = 200$.
6.2.2 Drag Coefficients

The drag coefficient $C_d$ is defined as

$$C_d = \frac{F_d}{\frac{1}{2} \rho_0 AU^2}, \quad (131)$$

where $F_d$ denotes the drag force, $\rho_0$ is density, $A = \pi r^2$ is the cross-section area normal to the stream-wise direction and $U$ is the velocity of the obstacle relative to the fluid. For viscous flow past a sphere, $F_d$ comprises the form drag $F_p$ and the frictional drag $F_f$, thus

$$F_d = F_p + F_f, \quad (132)$$

and we can calculate $C_d$ as the sum of form drag coefficient $C_p$ and friction drag coefficient $C_f$.

$$C_p = \frac{F_p}{\frac{1}{2} \rho_0 AU^2}, \quad (133)$$

$$C_f = \frac{F_f}{\frac{1}{2} \rho_0 AU^2}, \quad (134)$$

where $F_p$ is the total pressure force exerted on the obstacle in the direction of the fluid, and is calculated as an integral over the sphere surface. Let $\mathbf{n} = (n_x, n_y, n_z)$ be the unit vector normal to the sphere surface. $F_p$ is calculated as

$$F_p = \int_S (p n_x) dS. \quad (135)$$

$F_f$ is the sum of viscous stress exerted on the obstacle in the direction of the fluid and is calculated as a integral over the sphere surface according to

$$F_f = \int_S -\mu \left( 2 \frac{\partial u}{\partial x} n_x + \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) n_y + \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) n_z \right) dS. \quad (136)$$
In order to separate the influence of stratification we define $\Delta C_d$ as

$$\Delta C_d = C_d(Re, 1/Fr) - C_d(Re, 0).$$  \hspace{1cm} (137)

The change of $\Delta C_d$ for different Froude numbers is presented in Figure 33 as a solid line. The dashed line is a previous numerical solution [23] by Hanazaki and the open circles are experimental results [39]. The two numerical solutions are both for $Re = 200$ while the experimental results are obtained for Reynolds number ranging from 100 to 10000, thus the comparison is under the assumption that the influence of Reynolds number on $\Delta C_d$ is limited. For $1/Fr \in [0.1, 0.5]$, $\Delta C_d$ is close to zero and stay almost unchanged, implying that the drag coefficient is almost unaffected by very weak stratification.

For $1/Fr \in [0.5, 2]$, $\Delta C_d$ increases considerably. At $1/Fr = 1.0$, the slope of the dashed line for Hanazaki’s $\Delta C_d$ solution decreases sharply and then increases again. However, the slope of the solid line for our result remains almost constant till close to $1/Fr = 2$, followed by a sharp decrease. The difference between our results and the experimental results is probably due to the difference in Reynolds number. In fact, most experimental data points in [39] for $1/Fr \in [0.5, 2]$ are obtained with $Re > 1000$, but our numerical results are obtained with $Re = 200$. The various flow patterns for different values of Reynolds number as well as Froude number are summarised in [35]. For example, at $Fr = 0.9$ ($1/Fr = 1.11$), as Reynolds number changes from 200 to 5000, the flow pattern experiences different phases: non-axisymmetric attached vortex ($200 \leq Re \leq 700$), symmetric vortex shedding ($700 \leq Re \leq 1300$) and non-symmetric vortex shedding ($1300 \leq 5000$). Moreover, as $1/Fr$ increases, the experimental values of $\Delta C_d$ become on more scattered. The effects of stratification, or the value of Froude number, on drag coefficient for different flow patterns are likely to be different. Thus, our solution is within the tolerance of experimental results.

For $1/Fr \in [2, 9]$, the slope of our $\Delta C_d$ line decreases gently and for $1/Fr > 3$
Figure 33: Change of $\Delta C_d$ as a function of the inverse Froude number $1/\text{Fr}$. Hanazaki’s result [23] is presented in dashed lines and the NFT result is shown in solid lines. The open circles are experimental results [39].

The slope becomes negative. The reason may be the amplification of the horizontal lee-side eddies as more fluid particles lack the kinetic energy to pass the sphere vertically. Hanazaki’s result fails to capture a reasonable value of $\Delta C_d$ for $1/\text{Fr} > 2$ when the flow is highly stratified. In contrast, our result shows good agreement with the experimental data. Figure 34 shows the change of $\Delta C_d$ as the sum of $\Delta C_p$ and $\Delta C_f$. Values of $\Delta C_f$ are relatively small compared to $\Delta C_p$, therefore $C_d$ is dominated by $C_p$. Compared with Figure 10 in [23], the values of $\Delta C_f$ in our study are close to Hanazaki’s results, and the difference in $\Delta C_d$ comes mainly from $C_p$.

The difference in pressure can also be shown when comparing our dynamic pressure plot around the sphere in Figure 35 with Figure 3 in [23]. For Froude number $\text{Fr} = 200, 2$ and $1$, the distribution plots in our study and Hanazaki’s compare well with each other, and this is also where the values of $\Delta C_d$ in the two studies agree. For Froude number $\text{Fr} = 0.7, 0.5$ and $0.25$, the distribution plots in the two studies show considerable departure from each other, and this is also where the values of
$\Delta C_d$ disagree.

![Graph showing change of $\Delta C_d$, $\Delta C_p$, and $\Delta C_f$ as $1/Fr$ increases from 0.1 to 9.0](image-url)

Figure 34: Change of $\Delta C_d$ (solid line), $\Delta C_p$ (dashed line) and $\Delta C_f$ (dotted line) as $1/Fr$ increases from 0.1 to 9.0.
Figure 35: Dynamic pressure distribution on the sphere surface in $y = 0$ (solid lines) and $z = 0$ (dashed lines) planes for different Froude numbers $Fr$ while $Re = 200$. The dynamic pressure $C_D$ is equal to the total pressure subtracted by the hydrostatic pressure around the sphere. The angle is measured from the upstream stagnation point at $(x, y, z) = (-0.5, 0, 0)$. 
Incompressible Turbulent Flow Past a Sphere

Achenbach [1] studied vortex shedding experimentally from spheres and provided data for Strouhal number and drag coefficient for \( Re \in [6000, 3 \times 10^5] \). Constantinescu and Squires [4] investigated flow past a sphere at \( Re = 10000 \), comparing and validating the results obtained by LES with dynamic Smagorinsky SGS model [18], [34], and DES based on a modified Spalart-Allmaras model [79, 80]. Hassanzadeh et al. [25] studied flow structures around a sphere at \( Re = 5000 \) using LES with dynamic Smagorinsky model. Tsoutsanis et al. [82] simulated turbulent flow past a sphere at \( Re = 10000 \) to validate weighted-essentially-non-oscillatory schemes on a unstructured mesh.

In this section, our development in Section 7 for DES using a modified Spalart-Allmaras model, based on the unstructured-mesh finite volume NFT MPDATA scheme, is validated by a simulation of incompressible turbulent flow past a sphere. DES is equivalent to LES for separated flows and becomes RANS in attached region. It is suitable for simulating massively separated flows of engineering interest [4].

### 7.1 The Numerical Experiment

For the non-stratified flow past a sphere considered here, the computational domain is, the same as in Section 6, \( 20 \times 20 \times 20 \) with a sphere of diameter \( d = 1 \) placed in the center of the domain. The governing equations are (5) and (6) for incompressible flows with constant density. DES, which is based on the modified Spalart-Allmaras model discussed in section 2.4, is implemented in the present study. Non-slip boundary condition is imposed on the sphere surface. The upstream and downstream boundary conditions are set to impose free-stream normal velocity component \( U_0 = 1 \). Free-slip boundary condition is applied on all other boundaries. The kinematic viscosity is \( \nu = 0.0002 \), giving the Reynolds number \( Re = U_0 d / \nu = 5000 \).

The primary mesh is the same as in Section 6 but with a refined prismatic mesh to obtain higher resolution in the vicinity of the sphere. In the present study, the
number of prismatic layers is \( n = 25 \) and the parameters in the stretching function are \( P = 1.82 \) and \( Q = 2.0 \). Some useful information of the primary hybrid mesh is given in Table 5. The first layer of points off the sphere surface is located at \( r^+ = 0.98 \). \( r^+ \) is the wall unit given by

\[
r^+ = u_\tau r / \nu ,
\]

where \( r \) is the distance to the sphere surface, and \( u_\tau = 0.04 U_0 \) is an estimate of the friction velocity [4]. There are 5 points within ten wall units in the direction normal to the sphere surface.

<table>
<thead>
<tr>
<th>Number of Points</th>
<th>Number of Elements</th>
<th>Number of Prismatic Layers</th>
<th>Shortest Distance to the Sphere</th>
<th>Average Edge Length on the Sphere</th>
</tr>
</thead>
<tbody>
<tr>
<td>777506</td>
<td>4635316</td>
<td>25</td>
<td>0.0049</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 5: Statistics of the mesh for incompressible turbulent flow past a sphere

The computational mesh is the corresponding dual finite volume mesh. The experiment is run until \( T = 30 \) and \( T \) is the time for the flow to travel the distance of a diameter of the sphere. The time increment \( \Delta t \) is 0.00025, so that the Courant number \( C = u \Delta t / \Delta r \) is always smaller than 0.15, where \( \Delta r \) is the smallest cell size in the mesh.

7.2 Results

The time variation of the stream-wise drag coefficient is shown in Figure 36. The value of the corresponding mean drag coefficient is approximately 0.5285. The time development of drag coefficient is different from that in Figure 3 in [25]. Table 6 compares the mean drag coefficient \( C_d \) calculated in the present numerical study with previous experimental and numerical studies. \( C_d \) is averaged in \( T \in [15, 30] \) in the present work. It indicates that the present calculation compares well with previous studies and shows better agreement with Lofquist’s experimental result than the previous LES study by Hassanzadeh, et al. [25]. Previous results for \( Re = 10000 \) is
shown for extra information. $Re = 5000$ is chosen for comparison as it requires less computational cost.

![Time variation of drag coefficient](image)

Figure 36: The time variation of drag coefficient for DES of incompressible turbulent flow past a sphere at $Re = 5000$. The mean drag coefficient is approximately 0.5285.

<table>
<thead>
<tr>
<th>Reynolds number</th>
<th>Authors</th>
<th>Method</th>
<th>$C_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>Constantinescu and Squires</td>
<td>LES</td>
<td>0.393</td>
</tr>
<tr>
<td>10000</td>
<td>Constantinescu and Squires</td>
<td>DES [4]</td>
<td>0.440</td>
</tr>
<tr>
<td>10000</td>
<td>Achenbach</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>Lofquist and Purtell [39]</td>
<td></td>
<td>0.52</td>
</tr>
<tr>
<td>5000</td>
<td>Hassanzadeh, <em>et al.</em> [25]</td>
<td>LES</td>
<td>0.4683</td>
</tr>
<tr>
<td>5000</td>
<td>Present study</td>
<td>DES</td>
<td>0.5285</td>
</tr>
</tbody>
</table>

Table 6: Comparison of mean drag coefficient obtained via different methods

Figure 37 shows the instantaneous distribution of dynamic pressure around the sphere in the central vertical ($y = 0$) and horizontal ($z = 0$) cross-sections. Unlike the laminar flow cases in Section 6, the distribution in $[0^\circ, 180^\circ]$ and $[180^\circ, 360^\circ]$ is not symmetric, due to the randomness of turbulent flow. From around $70^\circ$ to $290^\circ$ the vertical and horizontal distributions are different and asymmetric. This is consistent with velocity vector images in Figure 38 showing that the separation is around $70^\circ$. The flow patterns in the central vertical and horizontal cross-sections
in Figure 38 are almost identical for the incompressible turbulent flow considered in the current work.

Figure 37: The distribution of dynamic pressure around the sphere. The solid line is the distribution in the central vertical ($y = 0$) cross-section while the dashes line is in the central horizontal ($z = 0$) cross-section. The angle is measured from the upstream stagnation point ($-0.5, 0, 0$).

Figure 38: Instantaneous velocity vector plots for DES of incompressible turbulent flow past a sphere when $Re = 5000$ in $y = 0$ (left) and $z = 0$ (right) planes.

Figure 39 shows that instantaneous flow patterns in the central vertical and horizontal planes are different. The reason is that the wake is highly turbulent and
larger eddies are more anisotropic. Figure 40 shows the instantaneous out-of-plane vorticity component contours in the detached shear layer. Vortex tubes of a range of scales are visible behind the sphere. The shear layer is laminar until separation, where transition occurs and the shear layers develop to be turbulent due to Kelvin-Helmholtz instability. Figure 4 in [4] shows the vortex tubes behind the sphere for \( Re = 10000 \), obtained via DES and LES. A comparison between Figure 4 in [4] and Figure 40 shows that the scales of the vortex tubes are generally larger at \( Re = 10000 \) than at \( Re = 5000 \).

![Figure 39: Contour plots of instantaneous velocity component \( w \) in \( y = 0 \) (top) and \( v \) in \( z = 0 \) (bottom) planes at \( T = 30 \) for DES of incompressible turbulent flow past a sphere with \( Re = 5000 \). There are 13 contour levels from -0.7 to 0.6 and negative values are in dashed lines. The wake is turbulent and the instantaneous vertical velocity contour plots in the central vertical and horizontal planes are different.

Figure 41 shows the instantaneous iso-surfaces of vorticity magnitude at \( T = 30 \). It is clearly seen that wake is turbulent and three dimensional. The maximum
Figure 40: Out-of-plane instantaneous vorticity component contour plots in $y = 0$ (top) and $z = 0$ (bottom) planes for DES of incompressible turbulent flow past a sphere with $Re = 5000$. There are 20 contour levels from -10 to 10.
vorticity magnitude is obtained slightly upstream in the vicinity of the sphere. As the flow moves downstream and away from the sphere, the vorticity magnitude decreases because of turbulence dissipation.

Figure 41: Instantaneous iso-surfaces of vorticity magnitudes at $T = 30$ for DES of incompressible turbulent flow past a sphere with $Re = 5000$. 
8 Stratified Turbulent Flows Past a Sphere and a Hemisphere

Stratified flows past an obstacle are relevant to geophysical, environmental and engineering applications. Okamoto [49] studied turbulent shear flow past a hemisphere-cylinder (a sphere whose lower half is replaced by a circular cylinder) experimentally, and compared the results with those for a sphere and a cylinder. Kim and Choi [30] simulated laminar flow past a hemisphere up to $Re = 5300$ numerically and found that, unlike steady flows, significant differences exist between unsteady flows past a hemisphere and a sphere. In Section 7 we investigated turbulent flows past a sphere for $Re = 5000$. Here we consider stratified flows with very high Reynolds number past a sphere as well as a hemisphere, for $Re > 10^7$. For such high Reynolds number, properties of turbulence are assumed to be independent on the actual value of $Re$.

8.1 The Numerical Experiment

The governing equations are (1–3) and viscous terms are neglected in the presented calculations. Implicit LES is employed for the current simulation. Free-slip boundary condition is applied on all but the upstream and downstream boundaries. It is shown e.g. in [60] that a realistic and physical solution can be obtained with a free-slip boundary condition on the surface of the obstacle [11]. This also precludes the difficulty of resolving or modelling the boundary layer associated with the non-slip condition. It is proved in [14] that the boundary layer becomes narrower as the Reynolds number increases.

For the numerical experiment of flow past a sphere, the computational domain and mesh are the same as those in Section 6. For flow past a hemisphere, the computational domain and mesh are half of those in Section 6, as shown in Figure 42. For both cases, the free-stream velocity is $v_e = (U, 0, 0) = (1, 0, 0)$; the Brunt-Väisälä frequency $N$ is $6s^{-1}$ such that the Froude number $Fr = \frac{U}{N_f} = 1/3$. In the
hemisphere test case, the flow is prevented from penetrating the central horizontal plane \((z = 0)\).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure42.png}
\caption{The upper panel shows the \(y = 0\) cross-section of the hybrid mesh for ILES of stratified turbulent flow past a hemisphere. The enlargement, in the lower panel, shows details of the prismatic layers.}
\end{figure}

\section{8.2 Results}

Both the sphere and hemisphere test cases are run until \(T = 15\). Figure 43 shows the change of drag coefficient with time for both cases. Overall the two lines agree well with each other. However, the drag coefficient for the hemisphere has much stronger oscillations before \(T = 3\). The mean drag coefficients for both cases are both around 1.2.

Figure 44 illustrates the flow patterns in the central vertical \((y = 0)\) plane of the
Figure 43: The history of drag coefficients for ILES of stratified turbulent flow past a sphere (solid black line) and a hemisphere (dashed red line), $Re \rightarrow \infty$, $Fr = 1/3$

sphere and hemisphere cases at $T = 2$. In both cases, there are similar lee-waves and vortices behind the obstacles. The vertical velocity contours' flow pattern for the sphere case is up-down symmetric, so we only compare the upper half of the top panel with the bottom panel. For illustration, the line $y = z = 0$ in the two panels are both called the central line. In both cases, the direction of the flow changes under the first lee-wave crest because of the overturning motions and the flow near the central line is advected back towards the obstacle. Furthermore, there are separation in both cases, and both separation angles are around $152^\circ$. However, the flow patterns in the two panels are not identical. In the hemisphere case, the first group of positive contours behind the hemisphere spreads over a larger region and extends further downwards than that in the sphere case, implying that the flow past the hemisphere has a stronger upward motion. The reason is that in the sphere case, the flow can go through the central horizontal ($z = 0$) plane, cf. Figure 45, and the flow structure is influenced by the interaction between the flow above and below the plane. However, in the hemisphere case, the flow in blocked by the
impenetrable bottom \((z = 0)\) boundary. In Figure 45, the wake contains a two-dimensional double vortex and the flow near the central line is advected backwards. The horizontal separation angle is about \(107^\circ\), which is different from its vertical counterpart. The vertical and horizontal separations are both induced by the adverse pressure gradient in the immediate lee of the sphere [40]. Both wave motion and the overturning vortices may contribute to the change of pressure distribution in the lee[23].

Figure 44: The vertical velocity contour plots on \(y = 0\) plane for ILES of stratified flow past a sphere (top) and hemisphere (bottom) at \(T = 2\). The velocity vectors are plotted after being interpolated onto a coarse Cartesian mesh, while the contours are from the original finite volume mesh. Negative values are presented as dashed lines. The contour values are from \(-1.1\) to \(1.1\). The interval for all contour plots in this section is \(0.1\), and zero contour lines are not shown.
Figure 45: The vertical velocity contour plots on the central horizontal plane \((z = 0)\) for ILES of stratified turbulent flows past a sphere at \(T = 2\).

Figure 46 shows the vertical velocity contour in \(z = r/3\) plane for the sphere and hemisphere cases. They are approximately identical. Let half-width \(L_h\) be the width of the obstacle at half of its height. The half-width of the hemisphere is \(L_h = \sqrt{3}r\). The wakes extend to around \(1.44L_h\) and \(0.35L_h\) from the sphere surface along the direction of the flow and in the direction normal to it, respectively. However, for flow past an isolated mountain, the wake extends to around \(L\) and \(0.5L\) in the respective to the directions of the flow, where \(L\) is the hill’s half-width, cf. Figure 7. Thus at this altitude, the wake extends further along the stream-wise direction, but shorter normal to the stream-wise direction. As the cross-section move upward to \(z = 5r/3\) the flow patterns, for the sphere and hemisphere cases are still approximately the same, cf. Figure 47, and are in similar shape as that of the mountain case in Figure 8. Compared with Figure 8, the larger positive contour in Figure 47 spreads further along the stream-wise direction, but shorter normal to the stream-wise direction.

Figure 48 shows the flow patterns at \(T = 6\) for the sphere and hemisphere cases, and again they are similar to each other. The lee-side waves propagate downstream further. There are overturning motions under the first crest of lee-waves and reverse
Figure 46: The vertical velocity contour plots in $z = \frac{1}{2}r$ plane for ILES of stratified turbulent flows past a sphere (top) and hemisphere (bottom) at $T = 2$. 
Figure 47: The vertical velocity contour plots on $z = \frac{5}{2}r$ plane for ILES of stratified turbulent flows past a sphere (top) and hemisphere (bottom) at $T = 2$. 
flows near the central lines. The difference between the two cases is similar to that for $T = 2$. There is a small group of positive contours under the first crest in the hemisphere case, but there is no such phenomenon in the sphere case. This implies that, as before, there is a stronger motion upwards behind the obstacle in the hemisphere case than the sphere case. This distinctive difference still exists at $T = 15$, cf. Figure 49. However, there are no adverse flows near the central lines and the wakes develop to be fully turbulent at $T = 15$. The velocity vectors plots shown in Figure 50, illustrate the different flow patterns near the sphere and the hemisphere at $T = 15$.

Figure 48: The vertical velocity contour plots on $y = 0$ plane for ILES of stratified turbulent flows past a sphere (top) and hemisphere (bottom) at $T = 6$.

Figure 51 compares the out-of-plane vorticity component contours on $y = 0$
Figure 49: The vertical velocity contour plots on $y = 0$ plane for ILES of stratified turbulent flows past a sphere (top) and hemisphere (bottom) at $T = 15$. 
Figure 50: The velocity vector plots on $y = 0$ plane for ILES of stratified turbulent flows past a sphere (top) and hemisphere (bottom) at $T = 15$
plane of the sphere and hemisphere cases. The vorticity patterns are similar for both geometries except near the central horizontal plane. Figure 52 illustrates the out-of-plane vorticity component contours in $z = 0$ plane for the sphere case. It clearly shows the shedding of vortices and implies the interaction between the flows on both sides of the $z = 0$ plane. However, there is no out-of-plane vorticity component on $z = 0$ plane in the hemisphere case, which is likely to be the reason for the different flow patterns in Figures 44, 48 and 49 for the two cases. The 3D structure of vorticity magnitudes is shown in Figure 53. Vortices of higher magnitudes are closer to the sphere and on the lee-side of the sphere or hemisphere. This is different from Figure 41 for $Re = 5000$ and Figure 32 for $Re = 200$, where vortices of maximum magnitudes are on the wind-side of the sphere.

Figure 54 shows the dynamic pressure distribution in $y = 0$ and $z = 0$ planes for both the sphere and hemisphere cases at $T = 2, 6$ and 15. The vertical distribution is considerably different from its horizontal counterpart because of stratification. The pressure distributions for the sphere and hemisphere cases are similar for angles from $0^\circ$ to about $120^\circ$, but different for angles bigger than $120^\circ$ at $T = 2$ and 6. This is in accordance with the difference in flow patterns. When the flow in the wake becomes fully turbulent at $T = 15$, the vertical distribution for both cases become similar to each other, while the difference between the horizontal distribution for both cases is still apparent.
Figure 51: Out-of-plane instantaneous vorticity component contours on $y = 0$ plane for ILES of stratified turbulent flows past a sphere (top) and a hemisphere (bottom); $Re \to \infty, Fr = \frac{1}{3}$
Figure 52: Out-of-plane instantaneous vorticity component contours on $z = 0$ plane for ILES of stratified turbulent flows past a sphere; $Re \to \infty$, $Fr = \frac{1}{3}$.

Figure 53: Vorticity magnitude iso-surfaces for ILES of stratified turbulent flows past a sphere (left panels) and a hemisphere (right panels); $Re \to \infty$, $Fr = \frac{1}{3}$. (a, b) is for vorticity magnitude 10, (c, d) is for magnitude 20 and (e, f) is for magnitude 30. They are shown separately for a clearer display.
Figure 54: The dynamic pressure distribution around the sphere on $y = 0$ plane (solid line) and $z = 0$ plane (dashed line) for ILES of stratified turbulent flows past a sphere (left) and hemisphere (right) at $T = 2$. The angles are measured from the upstream stagnation point.
9 Shallow Water Model Mesh Sensitivity Research

The current work is an extension of the work in [77] to explore further the accuracy of the edge-based NFT MPDATA flow solver of modelling the Rossby-Haurwitz (RH) wave [89] and the sensitivity of the solutions to details of meshes. Hybrid meshes of triangular and rectangular elements are generated on fixed reduced Gaussian points provided by ECMWF, to initiate the combining of the nonhydrostatic NFT solver with the IFS at ECMWF.

The RH waves are analytic solutions of the solenoidal non-linear barotropic vorticity equation on the sphere [77], and they propagate zonally with the same shape [26]. The simulation of the RH wave number 4 benchmark using the shallow water equations can be employed to test the accuracy of an algorithm for keeping the non-linear balance of wave form solutions [77]. All notations and descriptions of solution method follows closely material in [77]. The shallow-water equations on a sphere are written as

\[
\frac{\partial GD}{\partial t} + \nabla \cdot (Gv^* D) = 0 ,
\]

\[
\frac{GQ_x}{\partial t} + \nabla \cdot (Gv^* Q_x) = G\left(-\frac{g}{h_x} D \frac{\partial H}{\partial x} + fQ_y - \frac{1}{GD} \frac{\partial h_x}{\partial y} Q_x Q_y \right) ,
\]

\[
\frac{GQ_y}{\partial t} + \nabla \cdot (Gv^* Q_y) = G\left(-\frac{g}{h_y} D \frac{\partial H}{\partial y} + fQ_x - \frac{1}{GD} \frac{\partial h_x}{\partial y} Q_x^2 \right) ,
\]

where \( D \) is the depth of the shallow water and \( H \) is the height of the water surface. In the absence of orography or bathymetry, \( H = D \). \( G \) is the Jacobian determinant of the spherical coordinate transformation \((\lambda, \phi)\) with \( \lambda, \phi \) and \( r \) denoting the longitude, latitude angles and the spherical radius of the sphere, respectively. Therefore, \( G = h_x h_y \) and \( h_x = r \cos \phi, \ h_y = r \). Besides, \( v^* = (u^*, v^*) = (\dot{\lambda}, \dot{\phi}) \) and the momentum vector \( Q = Dv \) where \( v \) velocity is related to \((u^*, v^*)\)

\[
v = (u, v) = (h_x \dot{\lambda}, h_y \dot{\phi}) = (h_x u^*, h_y v^*) ,
\]
and
\[ \nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right) = \left( \frac{\partial}{\partial \lambda}, \frac{\partial}{\partial \phi} \right). \] (144)

The shallow water equations (142) can be written as the transport equation (85) with Jacobian $G$ in the form
\[ \frac{\partial G\psi}{\partial t} + \nabla \cdot (G\mathbf{v}^*\psi) = GR, \] (145)
where $R$ combines all right-hand-side forces. After (145) is rewritten in the following form
\[ \frac{\partial G\psi}{\partial t} + \nabla \cdot (\mathbf{V}\psi) = GR, \] (146)
where $\mathbf{V} = G\mathbf{v}^*$, the edge-based NFT MPDATA scheme (87) is employed. The resulting explicit model is distinctly different to the anelastic and incompressible flow models described earlier in Section 2. Its full derivation and description are provided in [77]. In this work, a class of mesh generators for the reduced Gaussian points distribution was developed. The number of points can vary depending on a prescribed mesh density. For a particular mesh density, all primary meshes of various shape combinations are based on the same mesh points, therefore the impact of mesh connectivity is revealed. The mesh points are projected from a sphere surface to a rectangular domain. Denoting the radius of the sphere as $r$, the corresponding rectangular domain is $2\pi r \times \pi r$. There are no points on the poles, so for the purpose of mesh generation, the domain $2\pi r \times (\pi r - 2\Delta_1)$ is considered, where $\Delta_1$ is the distance between the top or bottom boundary and the pole. The mesh points are allocated on latitudes only. On each latitude, the points are distributed evenly according to reduced Gaussian point rules of IFS. Next, the points are mirrored with respect to the equator. For the points distribution obtained here, the number
Figure 55: Periodic boundaries for RH wave test case (after [77]). The left panel shows the dual mesh construction at the left boundary for periodic boundary condition on the computational domain. The right panel shows the physical position of the boundary points on the sphere.

of mesh points is 89094. Since the latitudes become shorter near the poles, more points are allocated closer to the equator than the poles. The mesh points are up-down symmetric. The difference between the primary meshes studied here is mesh connectivity. The number of points’ layers is even and while the number of element’s layers is odd. The computational meshes are dual finite volume meshes generated based on the primary meshes, following the procedures detailed in Section 3.

Periodic boundary conditions are imposed at the left and right boundaries. The points on the two boundaries are in pairs, so no special treatment is needed for the position of the points. For every point on the left boundary there is a matching point on the right boundary and vice versa. In fact, each pair of points on the left and right boundaries of the rectangular domain are at the same position on the sphere, so the volumes associated with them are joined together to form a complete finite volume. Figure 55 shows the dual mesh construction for periodic boundary condition and the physical position of the boundary points. All fluxes and derivatives for periodic points are agglomerated analogously for such a complete finite volume.

The dual-mesh cells are extended to the latitudes of poles. In the final layer, 18 points are placed on a latitude about 0.037\(r\) from the pole. It is stated in [77] that at least 8 points are required in the polar region to avoid over stretched volumes. The flow from one of such points on the sphere towards the pole will pass the pole and reach the corresponding point on the other side. On the rectangular computational
Figure 56: The left panel (after [77]) is the node at the top latitude and its corresponding point at the other side of the pole on the computational domain. The right panel shows the physical domain viewed from above the pole.

domain, the corresponding point of \((x_i, \frac{\pi}{2} - \Delta_1)\) is located at \((x_i + \pi r, \frac{\pi}{2} - \Delta_1)\) This condition is satisfied as the points on the top boundary is evenly distributed and the number of points on the top boundary is odd. Figure 56 shows the node at the top latitude and its corresponding point at the other side of the pole. For special treatment of the boundary conditions see [77].

To aid visualisation, seven coarser primary meshes having 6166 mesh points only are shown in Figure 57 and Figure 58 to illustrate the methods for the primary mesh generation. In mesh A, if the number of points of two neighbouring latitudes is the same, rectangular cells are used, else triangular cells are used. The middle layer itself is up-down symmetric and all other layers on different sides of the equator are up-down symmetric. In mesh B, based on mesh A, the connectivity of two neighbouring triangular cell layers is modified to improve the isotropic quality of the mesh. Mesh C is based on mesh B and the only difference is that the rectangular layers are re-meshed to be triangular except for the part in the middle of the domain. Next, in mesh D all the other rectangular layers are replaced by triangles except for the middle element layer. In mesh E, the whole mesh is triangular via Delaunay triangulation but the mesh is no longer symmetric. Mesh F is constructed by creating points on the equator, using mesh D as a basis, and re-meshing the two middle element layers by triangles.

Figures 59 and 60 show the numerical solution of RH wave number 4 after five
Figure 57: RH wave test case coarse primary meshes; types A, B and C
Figure 58: RH wave test case coarse primary meshes; types D, E and F
Figure 59: RH wave after 5 simulation days on finer primary meshes
Figure 60: RH wave after 5 simulation days on finer primary meshes
Figure 61: RH wave after 14 simulation days on finer primary meshes
Figure 62: RH wave after 14 simulation days on finer primary meshes
simulated days for different meshes of the types listed above, however, the computational meshes are finer and use 89094 points. Stable solutions are obtained on Meshes B, C D and F and they compare well with Figure 6 in [77], obtained on an isotropic triangular mesh, and Figure 2 in [63], obtained on a Cartesian grid. Solutions on mesh A and E are stable and symmetric except for the polar regions. Figures 61 and 62 show results after 14 days of simulation. Mesh B gives the most stable result after 14 days, which shows good agreement with Figure 10 in [77], obtained on an isotropic mesh. Mesh C, D and F are all generated by replacing a proportion of rectangular mesh in mesh B by triangles, which, as the results indicate, leads to less isotropic meshes and less stable solutions.

Thus, more stable solutions of the RH wave number 4 benchmark are obtained on more isotropic primary meshes, which is also illustrated in [77]. The current study advocates an approach of generating most isotropic primary meshes based on fixed reduced Gaussian points. This finding is consistent with the limitation that Gauss divergence theorem, with the procedures for derivatives applied in this work.
10 Conclusions

A numerical study has been summarised that explores unstructured-mesh methods for simulating stratified, turbulent and shear flows past complex terrain or an obstacle. The governing anelastic and incompressible flow PDEs were integrated using the NFT MPDATA scheme. Developments for modelling stratified viscous laminar flows and DES have also been reported. The unstructured tetrahedral and hybrid based finite volume discretisation with the efficient edge-based data structure has been explored and developed, enabling the extension of the numerical scheme to fully unstructured meshes. The performance and potential of these methods were illustrated and quantified with challenging numerical simulations.

In particular, the successful validation of the NFT MPDATA model has been further extended to meshes constructed from the irregular tetrahedral primary grids. The presented results for intricate strongly stratified flow past a steep isolated hill [29] confirmed that the approach performs well for arbitrary shaped meshes. Furthermore, the obtained results matched the accuracy attainable on Cartesian grids and prismatic unstructured (in the horizontal) meshes with a similar spatial resolution. Special attention has been given to simulations of complex anisotropic and inhomogeneous flows with a range of scales emerging intermittently in the course of the simulations. Numerical study of such flows can clearly benefit from highly anisotropic and inhomogeneous meshes conforming to the nature of the flow.

Our simulations of sheared stratified 3D orographic flows with critical-level [21] documented good accuracy offered by, while illustrating cognitive benefits of, prismatic unstructured-mesh discretisation with high degree of the mesh variability. The process of laminar-turbulent transition and the interaction between critical layer and terrain-induced vortices were better revealed by higher spatial resolution.

Stratified flows with $Re = 200$ and a range of Froude numbers has been investigated based on a fully unstructured primary hybrid mesh, built from tetrahedral an prismatic meshes. Thanks to the flexibility of mesh resolution offered by un-
structured meshes, a much finer mesh was generated in the wake than Hanazaki. Flow patterns in cross-sections, 3D flow structure and quantitative results were discussed. Our results showed higher agreement with experimental data than in proceeding Hanazaki’s result [23] obtained on a structured grid.

DES based on the NFT MPDATA scheme has been developed and validated through simulation of incompressible turbulent flow past a sphere at $Re = 5000$. The qualitative and quantitative results were assessed, and compared well with data available in literature. The advantage of unstructured-mesh DES is that it allows a convenient application to engineering flows with complex geometries.

High-Reynolds-number stratified flows past a sphere or hemisphere were analysed using ILES properties of the NFT MPDATA scheme. The flow patterns of the two cases in the wake were different. The difference was quantified by the dynamic pressure distribution on the surface of the sphere and the hemisphere.

Mesh generators for hybrid meshes composed of triangular and rectangular elements were developed. Meshes generated from fixed reduced Gaussian points were used in a mesh sensitivity study. The simulation of Rossby-Haurwitz waves suggested that the results could be improved by using a more isotropic mesh with more rectangular elements. The mesh generators and successful validation of the NFT MPDATA scheme contributed to the development towards a massively-parallel framework for unstructured-mesh based finite-volume simulation of global atmospheric dynamics [88].
A Algorithms for Dual Mesh Construction

A.1 Edges

In the algorithm below, the two endpoints of each edge are stored in a $2 \times n_{edge}$ matrix

\[
inpoed(2,n_{edge})
\]

where $n_{edge}$ denotes the total number of edges in the mesh and the two end points are stored such that for any edge $i_{edge}$ we have

\[
inpoed(1,i_{edge}) < inpoed(2,i_{edge})
\]

$inpoed$ is built immediately from $psup1$ and $psup2$. $psup3$ is a help-array that marks repetitive edges.

Algorithm

!Initialize
\[psup3 = 0\] !psup3 is set to be a one-dimensional help array of the same length as $psup1$

\[do \ ip = 1, npoin \ !loop \ over \ all \ points\]
\[do \ ip = psup2(ipoin) + 1, psup2(ipoin + 1) \ !locations \ of \ all \ neighbouring \ points\]
\[jpoin = psup1(ip) \ !point \ number\]
\[jpoinfl = psup3(ip) \ !check \ marker\]
\[if(jpoinfl.ne.0)\] goto 301
!mark in case we calculate one edge twice
\[do \ jp = psup2(jpoin) + 1, psup2(jpoin + 1) \ !points \ surrounding \ neighbours\]
\[kp = psup1(jp)\]
\[if(kp.eq.ipoin)\] then !this is an edge
!mark in the corresponding location in $p_{sup3}$ for $jpoine$  

$$p_{sup3}(jp) = 1$$

$$i1 = \min(i_{poine}, jpoine)$$  !determine the start point

$$i2 = i_{poine} + jpoine - \min(i_{poine}, jpoine)$$

$$nedge = nedge + 1$$  
!store edges’ nodes in inpoed

$$inpoed(1, nedge) = i1$$

$$inpoed(2, nedge) = i2$$

endif

endo

301 continue

endo

endo

A.2 Elements Surrounding Each Edge

In this algorithm for elements surrounding each edge, five linked lists are used

$$iround1(nedge \ast ngx) \quad iround2(nedge) \quad iround3(nedge)$$

$$iround4(nedge \ast ngx) \quad iround5(nedge)$$

Their meanings are

1. $iround1(nedge \ast ngx)$ is used to store the elements around edges. $ngx$ is assumed to be the average number of elements around edges. $nedge$ is the total number of edges.

2. $iround2(nedge + 1)$ is for storing the locations of the starting and ending elements around edges in $iround1$. The elements surrounding the $j^{th}$ edge
in \textit{iround2} are stored in \textit{iround1} from location \textit{iround2}(j) + 1 to location \textit{iround2}(j + 1).

3. \textit{iround3(nedge)} relates edge numbers in \textit{iround2} to the global edge numbers. For the \textit{j}'th edge in \textit{iround2}, \textit{iround3}(j) is the global edge number.

4. \textit{iround4(nedge*ngx)} is for storing the forward directions or the searching path using element faces for road signs. The dimension of \textit{iround4} is the same as that of \textit{iround1}. For an element \textit{iround1}(i), \textit{iround4}(i) is the element’s face where the next searching element \textit{iround1}(i + 1) is located on.

5. \textit{iround5(nedge)} is used to store the edges at the boundary of the computational domain. For elements surrounding an internal edge the last element’s next neighbouring element is the first element. If the next neighbouring element of an element is zero, then the edge is at the domain boundary.

\begin{figure}[h]
    \centering
    \includegraphics[width=0.5\textwidth]{tetrahedron.png}
    \caption{An example of a tetrahedron}
\end{figure}

In Step 1, \textit{iface2} is the face for searching the neighbouring element and \textit{iedge} is the edge number. \textit{ipoi1} and \textit{ipoi2} are the endpoints of an edge. Three lists are used to remember which region the points and edges are located in. \textit{irgnp}(ip) is the region number the point \textit{ip} belongs to. The edges in region \textit{irgn} are stored in \textit{irgne}(icr(irgn) + 1 : icr(irgn + 1)).

There are two faces on the element attached to the edge. We need to specify one of them to be the forward direction face according to the order determined by \textit{lpoa},
$lpoed$ and $ilfe$. $lpoed$ relates edge nodes to element nodes. It is a $2 \times 6$ matrix since there are 6 edges in a tetrahedron and each edge has two end points.

$$lpoed = \begin{bmatrix} 1 & 2 & 3 & 1 & 2 & 3 \\ 2 & 3 & 1 & 4 & 4 & 4 \end{bmatrix}$$

where the columns correspond to the edges of the tetrahedral element in Figure 63. The $i^{th}$ node of the $j^{th}$ edge of the element is the $lpoed(i, j)^{th}$ node of the element. There are two neighbouring faces for each edge of the element. This information is given by $ilfe$ which is a $2 \times 6$ matrix since there are 6 edges in a tetrahedron and each edge has two neighbouring faces.

$$ilfe = \begin{bmatrix} 4 & 4 & 2 & 3 & 1 & 2 \\ 3 & 1 & 4 & 2 & 3 & 1 \end{bmatrix}$$

where the columns correspond to the element edges. The $i^{th}$ neighbouring face of the $j^{th}$ edge of the element is the $ilfe(i, j)^{th}$ face of the element. The edge and element numbers of the element are the same as that in $lpoed$ and $lpofa$, respectively.

The next step, Step 2, is for searching the elements surrounding edges. If the edge is not at the boundary, the criterion for stopping searching an edge is that the next element is the first element so that all elements around an edge are stored. If the edge is at the boundary, the elements surrounding the edge need to be searched in both directions until the next neighbouring element of an element is zero. In Step 2, the $boundaryedge$ subroutine is activated when the edge is recognised to be at the boundary. This subroutine is the second search for the same edge. The search starts from the same first element but in a different direction.

**Algorithm**

**STEP 1:** Find out the edge number and the correct face in which direction to search the next element

```plaintext
do ielem = 1, nelem !loop over all elements
```
do  iedel = 1, 6  
! loop over all edges of the element

! make sure choose right-hand face 'iface2' under right-hand system

ipoi2 = inpoel(lpoed(2, iedel), ielem)  ! second node number
ipoi1 = inpoel(lpoed(1, iedel), ielem)  ! first node number

iface2 = 0  ! forward direction initialisation

irno1 = irgnp(ipoi1)
irno2 = irgnp(ipoi2)
if(irno1 == irno2) then
    ns1 = icr(irno1) + 1
    ns2 = icr(irno1 + 1)
else
    ns1 = icr(nrgn) + 1
    ns2 = icr(nrgn + 1)
endif

do  nsc = ns1, ns2 ! loop over all edges in the region

ied = irgne(nsc)
i1 = inpoed(1, ied)  ! first node number
i2 = inpoed(2, ied)  ! second node number
if(i2.eq.ipoi2.and.i1.eq.ipoi1) then
    i1c = i1
    i2c = i2
    irhf = 2  ! right-hand face is second
    ilhf = 1
    iface2 = ilfe(irhf, iedel)
    iedge = ied
endif
if(i1.eq.ipoi2.and.i2.eq.ipoi1) then
    i1c = i1
\[ i2c = i2 \]
\[ \text{irhf} = 1 \] !right-hand face is first
\[ \text{ilhf} = 2 \]
\[ \text{iface2} = \text{ilfe} (\text{irhf}, \text{iedel}) \]
\[ \text{iedge} = \text{ied} \]
\[ \text{endif} \]
\[ \text{enddo} \]

\textit{STEP 2} Search the elements around edges

\texttt{do ielem = 1, nelem} \hspace{1em} !loop over all elements
\texttt{do iedel = 1, 6} \hspace{1em} !loop over all edges of the element

Use the algorithm in \textit{STEP 1} to find \texttt{iedge} and \texttt{iface2}

\texttt{!iflag} is a help array to avoid searching one edge twice

\texttt{if(iflag(iedge).ne.0)goto 30}
\texttt{nnlicz = nnlicz + 1}
\texttt{nlicz = 1}
\texttt{iround1(nnlicz) = ielem} \hspace{1em} !round1 the surrounding order of elements
\texttt{ielfirst = ielem}
\texttt{iel1 = ielem}
\texttt{iel2 = iesuel(iface2, ielem)} \hspace{1em} !the element's co-edge neighbor on \texttt{iface2}
\texttt{if(iel2.eq.0)then} \hspace{1em} !iel1 is a boundary element
\texttt{call boundaryedge}
\texttt{iround4(nnlicz) = iface2}
\texttt{iround5(iedge) = 1 goto 20} \hspace{1em} \texttt{endif}
\texttt{10 continue}
\texttt{iround4(nnlicz) = 0} \hspace{1em} !means the element is not at the boundary
\texttt{nnlicz = nnlicz + 1}
\texttt{nlicz = nlicz + 1}
\texttt{iround1(nnlicz) = iel2}
 inf0 = 0
 do  inf = 1, 4
  ielcheck = iesuel(inf, iel2) ! neighbour’s 4 neighbours
  if(ielcheck.eq.iel1)inf0 = inf ! find corresponding iface2
 enddo

 ! find the corresponding edge node number and the face in the forward direction
 do  id = 1, 6
  kipoi2 = inpoel(lpoed(2, id), iel2)
  kipoi1 = inpoel(lpoed(1, id), iel2)
  if(kipoi2.eq.i2c.and.kipoi1.eq.i1c)id0 = id
  if(kipoi1.eq.i2c.and.kipoi2.eq.i1c)id0 = id
 enddo

 ifex1 = ilfe(1, id0) Two faces of the edge
 ifex2 = ilfe(2, id0)
 if(ifex1.eq.inf0) then
  iface2 = ifex2 ! find next co-edge neighbour
 endif
 if(ifex2.eq.inf0) then
  iface2 = ifex1
 endif

 iel1 = iel2
 iel2 = iesuel(iface2, iel2)
 if(iel2.ne.iel1.first.and.iel2.ne.0) goto 10
 if(iel2.eq.0) then Boundary element
  call boudaryedge
  iaround4(nnlicz) = iface2 ! means jump to the first element and remember the boundary face
  iaround5(idenav) = 1
A.3 Elements Surrounding Each Edge Acceleration

In the following algorithm, two linked lists are used to accelerate the algorithm in Appendix A.2. They are

\[ \text{inpoe1}(npoin + 1), \text{inpoe2}(npoin \cdot nedpp) \]

and the edges emanating from the point \( ipoin \) are stored in \( \text{inpoe2} \), from position \( \text{inpoe1}(ipoin) + 1 \) to position \( \text{inpoe1}(ipoin + 1) \). The edges emanating from the point \( ipoin \) are those edges whose first node is \( ipoin \).

Algorithm

\begin{align*}
\text{do } ielem &= 1, nelem \quad \text{!Loop over all elements} \\
\text{do } iedel &= 1, nedel \quad \text{!Loop over all element edges} \\
ipoi1 &= \text{inpoe1}(lpoed(1, iedel), ielem) \\
ipoi2 &= \text{inpoe1}(lpoed(2, iedel), ielem) \\
ipmin &= \min(ipoi1, ipoi2)
\end{align*}
\[ \text{ipmax} = \max(\text{ipoi}_1, \text{ipoi}_2) \]

!Loop over the edges emanating from ipmin

do iedn = inpoel(ipmin) + 1, inpoel(ipmin + 1)
ied = inpoel(iedn)
if(inpoed(2, ied).eq.ipmax) then
iedge = ied
i1c = inpoed(1, iedge)
i2c = inpoed(2, iedge)
endif
enddo
!Then use Step 2 in the algorithm in Appendix A.2.

A.4 Volume and Projection

The following algorithm corresponds to section 3.1.

Algorithm

do ielicz = 1, nedge    !loop over all edges
ie = iround3(ielicz)    !edge number
i1 = inpoed(1, ie)      !edge nodes number
i2 = inpoed(2, ie)
nl = iround2(ielicz) + 1 !start element number
nu = iround2(ielicz + 1) !end element number
nnn = nu - nl + 1       !number of elements around this edge
nooo = 0
do is = nl, nu          !loop over the elements around the edge
nooo = nooo + 1         !counter
iel1 = iround1(is) !the first surrounding element
ii1 = inpoel(1, iel1)   !four nodes
ii2 = inpoel(2, iel1)
ii3 = inpoel(3, iel1)
ii4 = inpoel(4, iel1)
!boundary element
if(irlond4(is).ne.0) then
    iface2 = irlond4(is)
do nfn = 1, 3
    ifn(nfn) = inpoel(lpofa(nfn, iface2), iel1) !face nodes number
endo
!mid point of the face
x3 = (coord(1, ifn(1)) + coord(1, ifn(2)) + coord(1, ifn(3)))/3.0
y3 = (coord(2, ifn(1)) + coord(2, ifn(2)) + coord(2, ifn(3)))/3.0
z3 = (coord(3, ifn(1)) + coord(3, ifn(2)) + coord(3, ifn(3)))/3.0
!find edge and element mid point
x4 = (coord(1, i1) + coord(1, i2)) * 0.5 !for edge
y4 = (coord(2, i1) + coord(2, i2)) * 0.5
z4 = (coord(3, i1) + coord(3, i2)) * 0.5
x2 = (coord(1, ii1) + coord(1, ii2) + coord(1, ii3) + coord(1, ii4)) * 0.25
y2 = (coord(2, ii1) + coord(2, ii2) + coord(2, ii3) + coord(2, ii4)) * 0.25
z2 = (coord(3, ii1) + coord(3, ii2) + coord(3, ii3) + coord(3, ii4)) * 0.25
!edge nodes
x1 = coord(1, i1)
y1 = coord(2, i1)
z1 = coord(3, i1)
x5 = coord(1, i2)
y5 = coord(2, i2)
z5 = coord(3, i2)
!calculate volume
call cvolume
goto 10
endif

!Continue with internal elements
if(nooo.ne.nnn) then !find the second element
iel2 = iround1(is + 1)
endif

if(nooo.eq.nnn) then
iel2 = iround1(nl)
endif

jj1 = inpoel(1, iel2) !the four nodes of the next element
jj2 = inpoel(2, iel2)
jj3 = inpoel(3, iel2)
jj4 = inpoel(4, iel2)

!find edge and element mid point
x4 = (coord(1, i1) + coord(1, i2)) * 0.5 !for edge
y4 = (coord(2, i1) + coord(2, i2)) * 0.5
z4 = (coord(3, i1) + coord(3, i2)) * 0.5
x2 = (coord(1, ii1) + coord(1, ii2) + coord(1, ii3) + coord(1, ii4)) * 0.25
y2 = (coord(2, ii1) + coord(2, ii2) + coord(2, ii3) + coord(2, ii4)) * 0.25
z2 = (coord(3, ii1) + coord(3, ii2) + coord(3, ii3) + coord(3, ii4)) * 0.25

!find the face nodes between the two elements
npp(1) = ii1
npp(2) = ii2
npp(3) = ii3
npp(4) = ii4
npp2(1) = jj1
npp2(2) = jj2
npp2(3) = jj3
npp2(4) = jj4
iiic = 0

do icount = 1, 4

do iicount = 1, 4

if (npp(icount).eq.npp2(iicount)) then

iiic = iiic + 1

endif

enddo

enddo

! determine the mid point of the face

x3 = (coord(1, ifn(1)) + coord(1, ifn(2)) + coord(1, ifn(3)))/3.0
y3 = (coord(2, ifn(1)) + coord(2, ifn(2)) + coord(2, ifn(3)))/3.0
z3 = (coord(3, ifn(1)) + coord(3, ifn(2)) + coord(3, ifn(3)))/3.0

! edge nodes

x1 = coord(1, i1)
y1 = coord(2, i1)
z1 = coord(3, i1)
x5 = coord(1, i2)
y5 = coord(2, i2)
z5 = coord(3, i2)

! calculate volume

call cvolume

call cvolume

! calculate the neighboring volume in the next element

x2 = (coord(1, jj1) + coord(1, jj2) + coord(1, jj3) + coord(1, jj4)) * 0.25
y2 = (coord(2, jj1) + coord(2, jj2) + coord(2, jj3) + coord(2, jj4)) * 0.25
z2 = (coord(3, jj1) + coord(3, jj2) + coord(3, jj3) + coord(3, jj4)) * 0.25

call cvolume

10 continue
SUBROUTINE cvolume

!build vector coordinates using point coordinates

\[ x_{25} = x_2 - x_5 \]
\[ x_{35} = x_3 - x_5 \]
\[ x_{15} = x_1 - x_5 \]
\[ y_{25} = y_2 - y_5 \]
\[ y_{35} = y_3 - y_5 \]
\[ y_{15} = y_1 - y_5 \]
\[ z_{25} = z_2 - z_5 \]
\[ z_{35} = z_3 - z_5 \]
\[ z_{15} = z_1 - z_5 \]

\[ s_{x5} = y_{25} \cdot z_{35} - z_{25} \cdot y_{35} \]
\[ s_{y5} = z_{25} \cdot x_{35} - x_{25} \cdot z_{35} \]
\[ s_{z5} = x_{25} \cdot y_{35} - y_{25} \cdot x_{35} \]

\[ \text{volume}_i2 = (x_{15} \cdot s_{x5} + y_{15} \cdot s_{y5} + z_{15} \cdot s_{z5}) \]
\[ x_{24} = x_2 - x_4 \]
\[ x_{34} = x_3 - x_4 \]
\[ x_{14} = x_1 - x_4 \]
\[ y_{24} = y_2 - y_4 \]
\[ y_{34} = y_3 - y_4 \]
\[ y_{14} = y_1 - y_4 \]
\[ z_{24} = z_2 - z_4 \]
\[ z_{34} = z_3 - z_4 \]
\[ z_{14} = z_1 - z_4 \]
\[ s_{x4} = y_{24} \cdot z_{34} - z_{24} \cdot y_{34} \]
\( sy4 = z24 * x34 - x24 * z34 \)
\( sz4 = x24 * y34 - y24 * x34 \)

\( \text{volume}i1 = (x14 * sx4 + y14 * sy4 + z14 * sz4) \)

\( \text{coef} = 1 \)

\( \text{if} (\text{volume}i1 < 0) \text{then} \)
\( \text{coef} = -1. \)
\( \text{endif} \)

\( \text{vol}(i1) = \text{vol}(i1) + \text{volume}i1 * \text{coef} / 6. \)
\( \text{coef} = 1. \)

\( \text{if} (\text{volume}i2 < 0) \text{then} \)
\( \text{coef} = -1. \)
\( \text{endif} \)

\( \text{vol}(i2) = \text{vol}(i2) + \text{volume}i2 * \text{coef} / 6. \)
\( sn(ie, 1) = sn(ie, 1) + sx4 * 0.5 * \text{coef} \)
\( sn(ie, 2) = sn(ie, 2) + sy4 * 0.5 * \text{coef} \)
\( sn(ie, 3) = sn(ie, 3) + sz4 * 0.5 * \text{coef} \)

\( \text{return} \)
\( \text{end} \)
B References

References


REFERENCES


REFERENCES


[34] D.K. Lilly, A proposed modification of the Germano sub-grid scale closure method, Physics of Fluids, A 4 (1992), 633-635


[38] R. Löhner, Applied CFD Techniques, John Wiley & Sons Ltd. (2001), p37


REFERENCES


REFERENCES


REFERENCES


REFERENCES


