

Multi-resolution lattice Green's function method for high Reynolds number external flows

Thesis by
Ke Yu

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended May 6, 2021

© 2021

Ke Yu

ORCID: 0000-0003-0157-4471

All rights reserved

ACKNOWLEDGEMENTS

I have always considered myself as extremely lucky. The past years have been amazing because I was surrounded by the best parents, teachers, and friends. Rarely in my life did I achieve anything by myself, and because of that I am forever in debt.

First and foremost, I would like to thank my supervisor, Tim Colonius. I could never wish for a better advisor and I was really lucky to join the right group. Thank you for your guidance, patience, and encouragement throughout my PhD. Thanks for teaching me how to do research rigorously, and to set high standards for myself. You were always available whenever I needed your help, and at the same time you trusted me with a lot of freedom. Thanks for teaching me how to be confident in my research and myself, and for that I am always immensely grateful.

I want to offer special thanks to a few professors here at Caltech. I would like to thank Prof. Pullin for being my first-year advisor and for helping me with my research in the following years. You are basically my second advisor. I also would like to thank Prof. Tropp for putting together so many wonderful ACM courses. I would like to thank Prof. Meiron for the help in research and for running GALCIT. I would also like to thank Prof. Blanquart for taking time and effort to take care of every student in Gates-Thomas. With you guys here, I know future students are in good hands.

I want to thank the computational flow physics group — Tim, Oliver, Georgios, Gianmarco, Benedikt, Lennart, Mauro, Shunxiang, Spencer, Kevin, Andres and Andre, Kazuki, Phillippe, Ethan, Marcus, Jean, Liam, Omar (in age order). Because of you guys I felt like going to the office everyday. I would like to highlight my collaborators here, Benedikt and Gianmarco, this thesis is mine as well as yours.

I want to thank my family. You have given me infinite amount of love, support, and trust. Thanks for always being there for me.

I have been blessed with many good friends here at Caltech. I will always remember the countless coffee breaks during the day and the hugs goodbyes at night. I would like to express my gratitude to beloved fellows — Danilo, Florian, Guillaume, Simon, Kostas, Daniel, Fabien, Joel, Richard, Marcus, Karena, Zoila, Ke No.2, Yu, Jingling, De, Tomo, Shiyu, and so many more (in age order). I would also like to thank some of my dear friends that have helped me tremendously — Yaqian, Tianjie, Yamin, YY, YYC, Xinpeng, Yifan, Huiyu, Cheng, Tianhan, Yuanxun, YiYing, Yuanwen.

Whether we are physically in the same place or not, I will always bring our memories with me.

ABSTRACT

This work expands the state-of-the-art computational fluid dynamics (CFD) methods for simulating three-dimensional, turbulent, external flows by further developing the immersed boundary (IB) Lattice Green's function (LGF) method. The original IB-LGF method applies an exact far-field boundary condition using fundamental solutions on regular Cartesian grids and allows active computational cells to be restricted to vortical flow regions in an adaptive fashion as the flow evolves. The combination of spatial adaptivity and regular Cartesian structure leads to superior efficiency, scalability, and robustness, but necessitates uniform grid spacing. However, the scale separation associated with thin boundary layers and turbulence at higher Reynolds numbers favors a more flexible distribution of elements/cells, which is achieved in this thesis by developing a multi-resolution LGF approach that permits block-wise grid refinement while maintaining the important properties of the original scheme. We further show that the multi-resolution LGF method can be fruitfully combined with the IB method to simulate external flows around complex geometries at high Reynolds numbers. This novel multi-resolution IB-LGF scheme retains good efficiency, parallel scaling as well as robustness (conservation and stability properties). DNS of bluff and streamlined bodies at Reynolds numbers $O(10^4)$ are conducted and the new multi-resolution scheme is shown to reduce the total number of computational cells up to 99.87%.

We also extended this method to large-eddy simulation (LES) with the stretched-vortex sub-grid-scale model. In validating the LES implementation, we considered an isolated spherical region of turbulence in free space. The initial condition is spherically windowed, isotropic homogeneous incompressible turbulence. We study the spectrum and statistics of the decaying turbulence and compare the results with decaying isotropic turbulence, including cases representing different low wavenumber behavior of the energy spectrum (i.e. k^2 versus k^4). At late times the turbulent sphere expands with both mean radius and integral scale showing similar time-wise growth exponents. The low wavenumber behavior has little effect on the inertial scales, and we find that decay rates follow Saffman (1967) predictions in both cases, at least until about 400 initial eddy turnover times. The boundary of the spherical region develops intermittency and features ejections of vortex rings. These are shown to occur at the integral scale of the initial turbulence field and are hypothesized to occur due to a local imbalance of impulse on this scale.

PUBLISHED CONTENT AND CONTRIBUTIONS

- Yu, Ke, Benedikt Dorschner, and Tim Colonius (n.d.). “Multi-resolution lattice Green’s function method for high Reynolds number external flows”. In preparation.
- K.Y. proposed the idea, implemented the algorithm together with B.D., performed the numerical simulations, and analyzed the results.
- Yu, K., T. Colonius, D.I. Pullin, and G. Winckelmans (2021). “Dynamics and decay of a spherical region of turbulence in free space”. In: *J. Fluid Mech.* 907 (2021), A19. DOI: [10.1017/jfm.2020.818](https://doi.org/10.1017/jfm.2020.818).
- K.Y. proposed the idea, implemented the LES model, conducted the theoretical analysis, performed the DNS and LES of the spherical region of turbulence, and analyzed the results.
- Dorschner, B., K. Yu, G. Mengaldo, and T. Colonius (2020). “A fast multi-resolution lattice Green’s function method for elliptic difference equations”. In: *J. Comput. Phys.* (2020), p. 109270. DOI: <https://doi.org/10.1016/j.jcp.2020.109270>.
- K.Y. together with B.D. and G.M proposed the idea. K.Y. with B.D. implemented the algorithm.
- Yu, Ke, Benedikt Dorschner, and Tim Colonius (2020). “Multi-resolution lattice Green’s function method for incompressible flows”. In: (2020). Submitted.
- K.Y. proposed the idea, implemented the algorithm together with B.D., performed the numerical simulation and analyzed the results.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	v
Published Content and Contributions	vi
Table of Contents	vi
List of Illustrations	ix
List of Tables	xiii
Chapter I: Introduction	1
1.1 Motivation	1
1.2 Contributions and outline	4
Chapter II: Multi-resolution lattice Green's function method	6
2.1 Introduction	6
2.2 Navier-Stokes LGF solution on a uniform grid	7
2.3 Navier-Stokes LGF on an AMR grid	11
2.4 Adaptivity	23
2.5 Implementation	27
2.6 Verification	28
2.7 Collision of vortex rings	29
2.8 Concluding remarks	32
Chapter III: Immersed boundaries	35
3.1 Introduction	35
3.2 Immersed boundary method for external flows using LGF on a uni- form grid	36
3.3 IB on the AMR grid	39
3.4 Adaptivity	41
3.5 Fast linear solver for the IB formulation (Eq. (3.14b))	41
3.6 Flow around a sphere	43
3.7 Flow around a delta wing at an angle of attach of 20°	49
3.8 Concluding remarks	52
Chapter IV: Large-eddy simulation	53
4.1 Introduction	53
4.2 Stretched vortex sub-grid stress (SGS) model	54
4.3 Problem setup	56
4.4 Qualitative evolution	57
4.5 Quantitative evolution	60
4.6 Concluding remarks	63
Chapter V: Dynamics and decay of a spherical region of turbulence	65
5.1 Introduction	65
5.2 Initial spectrum and low wavenumber limit	66
5.3 Long-term statistics and low wavenumber behavior	68

5.4 Vortex ring ejections	71
5.5 Concluding remarks	75
Chapter VI: Summary and outlook	78
6.1 Summary	78
6.2 Outlook	79
Appendix A: Efficiency and parallel performance of AMR-LGF poisson solver	81
Appendix B: Energy spectrum of inhomogeneous turbulent flows	83
B.1 General definitions	83
B.2 Homogeneous and locally homogeneous turbulence	84
Appendix C: Spectrum: analytical and computational details	87
Bibliography	89

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
2.1 (a) Base unit of a finite-volume staggered grid. (b) 1-D diagram for the composite grid (dotted line) and the corresponding AMR grid (solid line with corresponding vertices in circle)	13
2.2 1-D diagram for the commutative interpolation with the Laplacian to avoid constructing a fine grid on level $k + 1$: (a) applying an interpolation $P_{k \rightarrow k+1}$ first and L_{k+1}^{-1} second; (b) applying L_k^{-1} first and $\bar{P}_{k \rightarrow k+1}$ second.	16
2.3 2D diagram for the AMR grid (blue) and the extended correction region (gray) at each refinement level.	18
2.4 Convergence of solutions on the finest grid w.r.t. to the refinement levels from 0 to 3 with a criterion $\alpha = 1/6$ of (a) the numerical solution Ψ and (b) left applying the discrete Laplacian to the numerical solution, $L\Psi$. For each plot three cases are considered: without the correction (red); with correction but without an extended region (blue); and with correction and with an extended region (black). Across all three cases the mesh topology is kept the same.	19
2.5 L_∞ velocity convergence of the NS-AMR-LGF solver wrt (a) the base level grid resolution, and (b) the finest level grid resolution, for $N_l = 0, 1, 2, 3$ respectively.	29
2.6 Evolution of the vortex ring collision at $Re = 7500$ from the case $N_l = 1$. From left to right, vorticity isosurfaces $ \omega R^2/\Gamma = [0.5 - 7.5]$ are given for $t\Gamma/R^2 = 0, 4.5, 9.0, 15.1, 19.9$, and 25.2	31
2.7 Evolution of the kinetic energy $\mathcal{K}(t)$ and enstrophy $\mathcal{E}(t)$ for the thin vortex ring collision at $Re = 7500$ for maximum refinement levels $N_l = 0, 1$, and 2 . Note that the $N_l = 0$ corresponds to the case that uses uniform finest cells for the whole adaptive domain (the most number of cells).	32
2.8 Mesh topology over the cross-section for $N_l = 0, 1$, and 2 . The top row shows the flow evolution at $ \omega R^2/\Gamma = [0.5 - 7.5]$ are given for $t\Gamma/R^2 = 0, 4.5, 9.0, 15.1$, and 19.9 . The second to the fourth row show the corresponding mesh topology for $N_l = 0, 1, 2$. For all cases, coarse mesh to fine mesh are shown from light gray to dark gray.	32

2.9	Flow visualization and mesh topology at $t \Gamma / R^2 = 19.9$ for (a) the non-AMR case ($N_l = 0$), and (b) the AMR case ($N_l = 1$). The computational blocks for the finest grid are shown in transparent boxes and a quarter of coarse grid for $N_l = 1$ is shown in green.	33
3.1	Diagram for the static mesh refinement for region near the immersed surface and adaptive mesh refinement with different levels for areas away from the bluff body.	42
3.2	Example of using Fibonacci lattice to distribute IB points on the surface of a sphere. 500 points are shown in this case, whereas up to 3×10^5 points are applied in the simulations to be discussed.	45
3.3	ℓ_∞ and ℓ_2 convergence of the velocity field for flow around a sphere at $Re = 100$. The black, blue and red curves correspond to to different but fixed base resolution. Each curve presents different levels of refinement with the adaptation criteria discussed in the text.	46
3.4	Vorticity of flow around a sphere at $Re = 3700$ at $tU_0/D \sim 30$. Vorticity contour level $\omega D/U_0 = [1, 30]$. Grid topology over the center plane is plotted for 0-2 levels of refinement from light to dark gray.	48
3.5	Comparison of the vorticity field near the sphere at $Re = 3700$ and $Re = 10,000$ for $x/D < 4$. Grid topology over the center plane is shown from light to dark gray.	48
3.6	Delta wing setup (He, An, et al., 2019): (a) Experimental setup ; (b) CAD model; (c) immersed boundary mesh.	50
3.7	(a) Vorticity visualization at $t^* = t U_0/D = 1.5, 4.0, 10.0$ for the delta wing at the angle of attack $\alpha = 20^\circ$. (b) Mesh topology at $t^* = 4.0$. Grid level from 0 to 2 is shown from light to dark gray.	51
4.1	Vorticity magnitude in a cross-section through the center at $t/t_\ell = 0$ and $t/t_\ell = 8.6$, corresponding to the case DNS_0 from table 4.1. Black contour lines indicate the spatially adaptive computational domain. The smallest adaptivity unit is a block of 16^3 computational cells.	55
4.2	Energy spectrum of (1) DNS_0 (—) and (2) a DNS calculation at 3/2 times the resolution as that in the case DNS_0 (- - -), at $t/t_\ell = 1.3$	58
4.3	Vorticity magnitude iso-surface of (a) DNS_0 and (b) LES_0 at $t/t_\ell = 0, 1.72, 4.02, 8.60, 17.47$ from top to bottom, where t_ℓ is the large eddy turnover time of the initial IHT field.	59

4.4	Decay of the kinetic energy $\mathcal{E}(t)$ for different simulations: DNS_0 (—); LES_0 resolved kinetic energy (⋯⋯); LES_0 total kinetic energy (- - -); an under-resolved DNS (- - -). The initial kinetic energy in DNS_0 is used to normalize all simulations.	61
4.5	Total energy spectrum at (a) $t/t_\ell = 0$, (b) $t/t_\ell = 4.02$ and (c) $t/t_\ell = 17.47$ for DNS_0 (—) and LES_0 (- - -). The same guide line for $k^{-5/3}$ as in figure 5.1 is also given (⋯⋯).	62
4.6	Spherical-shell spectrum at different radii and times. The left column (a-c) are results from DNS_0 at $t/t_\ell = 0, 4.02, 17.47$ respectively. The right column (d-f) are results from LES_0 at the same times. For each figure, the gradation in color corresponds to radii $r/B = (0.32, 0.48, 0.64, 0.80, 0.96)$ from darkest to lightest shade. An equivalent guide line for $k^{-5/3}$ as in figure 5.1 is given (⋯⋯).	64
5.1	Energy spectrum of (1) the initial condition of simulation DNS_0. (—), where the low wavenumber limit (left of the dotted region) is calculated through an expansion method (Appendix C); (2) DNS_0 with the initial impulse cancelled using Eq. (5.7) (- - -); (3) the corresponding IHT field multiplied by the ratio between the spherical region volume $\frac{4}{3}\pi R^3$ and the cubic domain volume B^3 (- - -); (4) a guide line for $1.6\varepsilon^{2/3}k^{-5/3}$ scaled with the same ratio, where ε is the dissipation rate in the original IHT field (⋯⋯); (4) a slope of k^2 (- - -) and k^4 (- - -) for the low wavenumber limit.	69
5.2	Long-term evolution of (a) LES_0 and (b) LES_IC2 at $t/t_\ell = 0, 4, 17, 66, 143, 263$ from top to bottom.	70
5.3	Long-term evolution of (a) the kinetic energy decay compared with asymptotic behavior of Saffman IHT $\mathcal{E}(t) \sim t^{-6/5}$ (⋯⋯) and Batchelor IHT $\mathcal{E}(t) \sim t^{-10/7}$ (- - -); (b) the integral scale growth for case LES_0 (—) and LES_IC2 (- - -) up to $t/t_\ell = 400$ compared with asymptotic behavior of Saffman IHT $\ell \sim t^{2/5}$ (⋯⋯) and Batchelor $\ell \sim t^{2/7}$ (- - -).	72
5.4	Long-term evolution of the total spectrum for (a) LES_0 and (—) (b) LES_IC2 (—), up to $t/t_\ell = 500$ with $\Delta t/t_\ell = 15$ between each line. In figure (a) guide lines for $k^{-5/3}$ (⋯⋯) and k^2 (- - -) are given. In figure (b) guide lines for $k^{-5/3}$ (⋯⋯) and k^4 (- - -) are given.	73
5.5	Long-term evolution of the normalized Loitsyansky integral $I(t)/I(0)$ of LES_IC2 (k^4 type), up to $t/t_\ell = 500$	74

5.6	Trajectory of the center of the turbulence cloud from $t/t_\ell = 0$ to 400 for LES_0 (—) and LES_IC2 (—) with every marker separted by $\Delta t/t_\ell = 30$. For LES_0 the direction of the initial impulse is indicated with the arrow at the end of the trajectory. For LES_IC2 the impulse is zero.	74
5.7	Comparison of the long-term mean radius \bar{r} for $p = 2$ between LES_0 (—) and LES_IC2 (- - -) up to $t/t_\ell \sim 400$. Results are normalized with their initial mean radii \bar{r}_o . Guide lines for $t^{2/5}$ (.....) and $t^{2/7}$ (- - -) are also given.	75
5.8	Long-term turbulence cloud evolution with vortex ring ejections for cases defined in table 4.1. First row: LES_D1, LES_0, LES_D2; second row: LES_R2, LES_R1, LES_0; third row: LES_B1, LES_0, LES_B2.	76
5.9	Maximum Gaussian weighted impulse over \mathbf{x} , $\max_{\mathbf{x}} I(\mathbf{x}; \zeta)$ as a function of ‘width’ ζ for LES_0 at $t = 0$ (—).	77
A.1	Computational rates and parallel performance.	82

LIST OF TABLES

<i>Number</i>	<i>Page</i>
2.1 Coefficients of the HERK scheme (Brasey and Hairer, 1993).	11
2.2 Number of computational cells for $N_l=0, 1$ and 2 , compared with a static rectangular domain of minimum bounding box over the time horizon $t \Gamma / R^2 = [0 - 20]$	33
3.1 Mean flow drag coefficient of a sphere at Reynolds numbers 3700 and $10,000$. Results from LC17 - Liska and Colonius (2017), RB11 - Rodriguez, Borell, et al. (2011), RB13 - Rodriguez, Lehmkuhl, et al. (2013), YK06 - Yun, Kim, and Choi (2006), A72 - Achenbach (1972), CS03 - Constantinescu and Squires (2003).	49
3.2 Comparison of the number of computational cells used in the AMR-IB-LGF scheme, compared with a spatially adaptive computational domain with uniform fine grid, and a rectangular bounding domain with uniform fine grid, together with the percentage in the reduction of grid points.	49
3.3 Lift and drag coefficients from simulation at $Re = 10^4$ and experiment at $Re = 10^5$ by He, Le Provost, and Williams (2018).	51
4.1 Simulation parameters. The spectrum type refers to the leading non-zero order in the low wavenumber limit which is discussed in detail in Chapter 5.	57
4.2 Summary of the simulation parameters used in the pseudo-spectral code and the resulting IHT fields. ℓ is the integral scale, η is the Kolmogorov length scale, and k_{\max} is the maximum wavenumber.	57

Chapter 1

INTRODUCTION

1.1 Motivation

In an external flow system, the fluid medium moves around an immersed body and the dynamics of the fluid are subject to the far-field boundary condition. Such systems are ubiquitous, including systems from low Reynolds number flying insects to high Reynolds number airfoils and wind-turbines. Advances in the area of computational methods during the past decade highlight the importance of using high-fidelity algorithms and data from simulations to provide a test bed for characterizing and understanding complex phenomena presented in external flows. However, simulations of high Reynolds number external flows are challenging due to the far-field boundary conditions, the complex geometries associated with the immersed body, and the wide range of physical scales associated with thin boundary layers and turbulence. To date, the use of computational techniques are still largely restricted to low to moderate Reynolds number flows. This research seeks to fill this gap by providing high-fidelity, efficient, scalable computational tools to predict the detailed flow fields and forces associated with the complex geometries and unsteady aerodynamics.

The unbounded physical domain present in external flows may numerically be truncated with artificial inflow/outflow boundary conditions (Lackner, 1976; James, 1977; Tsynkov, 1998; Colonius, 2004; McCorquodale et al., 2007), or truncated based on the compact vorticity field, together with a free-space Green's function that satisfies the exact far-field boundary condition. The Green's function can be based on the discretized equations, i.e., the lattice Green's function (LGF) (Glasser and Zucker, 1977; Delves and Joyce, 2001; Gillman and Martinsson, 2010; Gillman and Martinsson, 2014; Liska and Colonius, 2014; Liska and Colonius, 2016), or the continuous ones, as is typically done in vortex methods (e.g., Leonard, 1980; Chatelain and Koumoutsakos, 2010). In this work we focus on the LGF method which is reviewed in detail in section 2.2.2. In the LGF approach, the grid need only be defined where the vorticity induced by the surface is nonzero. This results in a much more compact computational domain compared to typical CFD meshes used for external flows. The computational domain can also be adaptive based

on a threshold value of the vorticity. In other words, the grid can be dynamically switched on and off in different regions of space as needed to most efficiently resolve the vorticity generated by the surface. Additionally, when combined with mimetic finite-difference/finite-volume methods, the LGF method is advantageous as the resulting schemes are discretely conservative with provable stability bounds (Liska and Colonius, 2016). Efficient, parallel solutions of the discrete convolution in three spatial dimensions can be achieved by adapting variants of the fast multipole method to yield linear complexity (Liska and Colonius, 2014).

The complex geometries associated with the immersed bodies are often dealt with body-fitted meshes in the CFD (e.g. Hirt, Amsden, and Cook, 1974; Tezduyar, Behr, et al., 1992). The flexibility in the nodal distribution often allows thin boundary layers to be resolved more efficiently and thus favors high Reynolds number flows. However, moving and deforming complex geometries often require re-meshing at each time step and other sophisticated procedures to maintain the accuracy (Mittal and Iaccarino, 2005; Tezduyar, Sathe, et al., 2006). Additionally, techniques that use irregular grids often require more general numerical solvers, which are less efficient compared to those designed for grids with higher regularity. Furthermore, the associated discretization often does not share the same conservation, commutativity, orthogonality, and symmetry properties.

In this work we focus on an alternative approach for solving complex geometries, the immersed boundary (IB) method. The IB method, which was originally developed by Peskin (1972), solves PDEs on Eulerian grids with immersed geometries. The surfaces of the geometries are emulated with a set of discrete IB points, and the no-slip boundary conditions are imposed through interpolating the underlying fields at those points, which circumvents the need to generate body-fitted mesh and still allows the usage of efficient numerical solvers for regular grids. The current work uses the distributed Lagrange multiplier (DLM) method developed by Colonius and Taira (2008) where the no-slip boundary conditions are imposed exactly. More about the DLM method is discussed in Chapter 3.

While the combination of the LGF and IB methods is highly efficient (Liska and Colonius, 2017), it is still limited to moderate Reynolds numbers since they fall short in solving the scale separation associated with thin boundary layers and turbulence present in high Reynolds number flows. The former, the LGF method, does not readily permit the static or adaptive mesh refinement (SMR and AMR) since the definition of the LGF is only based on regular grids. While several multi-resolution

schemes based on FMM and multigrid have been proposed for the Poisson equation that arises in incompressible flow (e.g. Lashuk et al., 2012; Ying, Biros, and Zorin, 2004), these methods are likewise not straightforward to be combined with the LGF. On the other hand, the majority of IB methods have only been employed for low to moderate Reynolds number flows. One disadvantage of many IB methods, including ours, is that only first-order accuracy is achieved near the surface. More importantly, since the IB methods are based on Eulerian grids compared to methods where body-fitted meshes are used, it is more difficult to cluster cells near the surface and thus less efficient to resolve the thin boundary layers that exist for high Reynolds number flows.

The present research aims to overcome these issues and bring the advantages of both the LGF technique and the IB method to external flows at high Reynolds numbers. We extend the current methodology in three ways. First, we developed an efficient AMR framework that is fully compatible with, and exploits the existing strengths of the fast LGF method. Second, we extend the AMR-LGF framework to the DLM-IB method. We show that by employing the fast AMR-LGF solver, the IB method can efficiently simulate external flows with complex geometries at relatively high Reynolds numbers. Third, we explored the possibilities of combining the LGF method with a state-of-the-art LES model which is discussed below.

Despite progress over several decades, numerical simulations of unsteady external aerodynamics remains a roadblock for engineering prediction. Direct numerical simulations (DNS), where all scales in the flow are resolved explicitly on the mesh, require a number of computational cells to scale as at least $\text{Re}^{2.25}$. Typical engineering problems such as airplanes are often at Reynolds numbers of $\mathcal{O}(10^6)$ or higher where DNS approaches become prohibitive. Among all turbulence modeling approaches, the Reynolds-averaged Navier-Stokes (RANS) equations and large-eddy simulation (LES) are the most widely used. In RANS, the Navier–Stokes equations are time-averaged and the turbulence characteristics are fully modelled. For this reason RANS is the least computationally demanding but often is less accurate in predicting flows with separations and when fluctuating forces are important in applications such as flows around airfoils at angles of attack. On the other hand, in LES, the Navier–Stokes equations are filtered in space and only the unresolved small turbulence scales up to a cutoff are modeled. Because of that, LES is able to capture the large flow fluctuations and has been shown a very successful modeling approach for flows in regions away from the immersed surface (Lesieur and Metais,

1996; Meneveau and Katz, 2000; Sagaut, 2006).

However, the cascade of turbulence length scales as assumed in LES does not hold in the same sense near the surface. One approach would be to resolve fully the inner and outer boundary layers (wall-resolved LES), where the number of computational grids required would still scale as $Re^{13/7}$ (Choi and Moin, 2012). Note that those estimations are based on body-fitted meshes and meshes with cubic cells such as the Cartesian grids would require even higher numbers of grid points. To reach Reynolds numbers greater than 10^5 , different approaches are needed. One popular attempt is the detached eddy simulation (DES), where RANS is used for attached and near-wall flows and LES is used elsewhere. However a major weakness of DES is its response to grid resolution. In some situations, DES on a given grid is less accurate than RANS on the same grid, or DES on a coarser grid (Spalart, 2009). The other approach is the wall-modeled LES (WMLES), where the near-wall eddies are considered sub-grid scale and their effects are modeled (usually through imposing a slip velocity) instead of being fully resolved. Good results using WMLES have been reported for a number of smooth and rough-wall attached flows (Inoue, Pullin, and Marusic, 2013; Saito and Pullin, 2012; Saito and Pullin, 2014) and separated flows (Cheng and Pullin, 2015). The WMLES can be made fully compatible with the IB method where slip velocities can be specified.

1.2 Contributions and outline

In this thesis, we propose three steps to further develop the original IB-LGF method towards external flows at higher Reynolds numbers: a multi-resolution LGF framework for solving viscous, incompressible flows on unbounded domains, an IB extension to this method for solving external flows around immersed bodies, and a combination of the LGF technique with a state-of-the-art LES model.

In Chapter 2, we develop a novel AMR framework to enhance the LGF approach. In this framework, the AMR grid is considered as a subset of a composite grid that is constructed from a series of unbounded uniform staggered Cartesian grid of differing resolution. The LGF technique is formally applied on all levels. We show that by considering the commutativities between interpolation, coarsening, and the LGF, this process can be evaluated efficiently without numerically constructing the composite grid, while the solutions still enjoy the desired properties. We then extend the AMR-LGF framework to the incompressible Navier-Stokes equations by combining it with a half-explicit Runge-Kutta scheme (NS-AMR-LGF-HERK).

We show both spatial adaptivity and refinement adaptivity can be applied to this approach, and demonstrate its capabilities by simulating the collision of thin-cored vortex rings at $Re = 7500$.

In Chapter 3, we show that the IB method can readily be included in the NS-AMR-LGF-HERK scheme to solve external flows around complex geometries at high Reynolds numbers. The resulting AMR-IB-LGF scheme is demonstrated with DNS of the flow around a sphere at $Re = 3700$ and $10,000$, and a delta wing at $Re = 10,000$. With these two examples we show that the multi-resolution scheme is able to achieve significant computational saving through the spatial adaptivity as well as the mesh refinement for the boundary layers and regions with high vorticity.

In Chapter 4, the LGF method is coupled with an specific LES sub-grid-scale model, the stretched-vortex of Chung and Pullin (2009a). This state-of-the-art LES model was validated in the LGF framework by developing a novel turbulent flow in free-space — the spherical cloud of turbulence. We perform DNS and LES of this flow and the results from LES are compared with DNS in detail. In future work, the LES modeling can be extended to include the virtual wall model to further alleviate computational expense in performing LES at higher Reynolds numbers (i.e to enable WMLES).

Several features of the evolution of such a spherical region of turbulence are also of theoretical interest. In Chapter 5, address some fundamental questions in the theory of (nearly) isotropic homogeneous turbulence, namely the evolution of the low-wavenumber (Batchelor- or Saffman-type) turbulence.

Lastly, in Chapter 6, we conclude the current work and present areas for future research.

*Chapter 2*MULTI-RESOLUTION LATTICE GREEN'S FUNCTION
METHOD**2.1 Introduction**

Numerical simulations of high Reynolds numbers, incompressible flows on unbounded domains are challenging due to the wide range of physical scales and unbounded computational domain. The scale separation associated with the boundary layers and turbulence favors a flexible distribution of elements/cells with refinement in regions of high gradients. For unstructured or structured body-fitted meshes, many techniques are available to achieve this clustering of elements, whereas for immersed boundary methods, the most natural way to do this is through static or adaptive mesh refinement (AMR) in Berger and Olinger (1984), Berger and Colella (1989a), MacNeice et al. (2000), and Nissen et al. (2015).

In this chapter we focus on the LGF method for the unbounded domain. Efficient, parallel solutions of the discrete convolution in three spatial dimensions can be achieved by adapting variants of the fast multipole method (Liska and Colonius, 2014), which we refer to as the fast LGF (FLGF) method. As discussed in chapter 1, an important disadvantage is that the LGF does not readily permit the static or adaptive local refinement required to efficiently simulate high Reynolds number flows. In our recent work (Dorschner et al., 2020), we proposed a multi-resolution extension of the FLGF method (AMR-LGF) that enables block-structured mesh refinement while retaining the efficiency of the FLGF technique. In this chapter, we propose an ansatz for AMR that reinterprets and improves this algorithm, and we further extend the technique to solve the incompressible Navier-Stokes equations. We consider the AMR grid as a restriction from an ambient composite grid that is constructed from a series of infinite lattices of differing resolution. Solutions to the Poisson equation are formally solved on every level of the composite mesh using the LGF, before being restricted back to the AMR grid. We then construct commutative interpolation operators that obviate the need for explicitly computing most of the composite grid. In applying the scheme to the full Navier-Stokes equations, we limit our attention to unbounded flows without immersed surfaces, but the algorithms we propose are compatible with the previous IBLGF method and will be combined in

the next chapter.

The chapter is organized as follows. In section 2.2, the FLGF based scheme for solving incompressible flows on unbounded uniform grids is briefly reviewed. In section 2.3, the concept of a composite grid is introduced. The previous AMR-LGF method (Dorschner et al., 2020) is recast in this framework and an extended source correction is proposed. In section 2.3.5 we extend this framework to the LGF for an integrating factor to exactly advance the viscous terms when combined with a half-explicit Runge-Kutta scheme, and in 2.3.6 we construct the remaining operators for the multi-resolution Navier-Stokes solver for incompressible external flows. Section 2.4 discusses how both spatial and refinement adaptivity can be achieved and section 2.5 summarizes the implementation. Lastly the numerical results are given in section 2.6 and 2.7.

2.2 Navier-Stokes LGF solution on a uniform grid

2.2.1 Discretization on unbounded uniform grid

The Navier-Stokes LGF algorithm developed by Liska and Colonius (2016) is briefly reviewed in this section. This algorithm solves the incompressible viscous Navier-Stokes equations subject to the exact far-field boundary conditions. In a non-dimensional form, the equations are given by

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}, \quad (2.1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1b)$$

$$\mathbf{u}(\mathbf{x}, t) \rightarrow \mathbf{0}, \quad p(\mathbf{x}, t) \rightarrow p_\infty \quad \text{as} \quad |\mathbf{x}| \rightarrow \infty, \quad (2.1c)$$

where \mathbf{u} is the velocity field, p is the pressure, and Re is the Reynolds number.

Eq. (2.1) are formally discretized on an unbounded staggered, uniform Cartesian grid (lattice) of single resolution. A base unit of this grid is shown in Fig. 2.1a: its cell (C) and vertices (\mathcal{V}) discretize scalar quantities, and its positive faces (\mathcal{F}) and edges (\mathcal{E}) store vector quantities. We use \mathbb{R}^Q to denote the grid function spaces with values defined on $Q \in \{C, \mathcal{F}, \mathcal{E}, \mathcal{V}\}$. The two principal discrete quantities to be solved for are the velocity and the pressure; we denote their corresponding grid functions (on the infinite lattice) as $\mathbf{u} \in \mathbb{R}^{\mathcal{F}}$ and $p \in \mathbb{R}^C$, respectively.

The grid function space is equipped with the following differential operators: the discrete gradient $\mathbf{G} : \mathbb{R}^C \rightarrow \mathbb{R}^{\mathcal{F}}$, the discrete divergence $\mathbf{D} : \mathbb{R}^{\mathcal{F}} \rightarrow \mathbb{R}^C$, the discrete curl $\mathbf{C} : \mathbb{R}^{\mathcal{F}} \rightarrow \mathbb{R}^{\mathcal{E}}$ and $\bar{\mathbf{C}} : \mathbb{R}^{\mathcal{E}} \rightarrow \mathbb{R}^{\mathcal{F}}$, and the discrete Laplacian $\mathbf{L}_Q : \mathbb{R}^Q \rightarrow \mathbb{R}^Q$.

This discretization is of second-order accuracy and yields conservative, mimetic and commutative properties. These properties are extensively exploited in this algorithm (Liska and Colonius, 2016). For instance, one has the following mimetic properties,

$$\mathbf{D} = -\mathbf{G}^\dagger, \quad \bar{\mathbf{C}} = \mathbf{C}^\dagger, \quad \mathbf{L}_C = -\mathbf{G}^\dagger \mathbf{G}, \quad (2.2)$$

and the following commutativity properties,

$$\mathbf{H}_{\mathcal{F}} \mathbf{G} = \mathbf{G} \mathbf{H}_C, \quad \mathbf{D} \mathbf{H}_{\mathcal{F}} = \mathbf{H}_C \mathbf{D}, \quad (2.3)$$

where \mathbf{H}_Q is the integrating factor operator to be introduced in section 2.2.3.

Using these differential operators, Eq. (2.1) are discretized in space as

$$\frac{d\mathbf{u}}{dt} + \mathbf{N}(\mathbf{u} + \mathbf{u}_\infty) = -\mathbf{G}\mathbf{p} + \frac{1}{\text{Re}} \mathbf{L}^{\mathcal{F}} \mathbf{u}, \quad \mathbf{D}\mathbf{u} = 0, \quad (2.4)$$

where $\mathbf{N}(\mathbf{u} + \mathbf{u}_\infty) \approx \boldsymbol{\omega} \times (\mathbf{u} + \mathbf{u}_\infty)$ is the corresponding discretized non-linear term with $\boldsymbol{\omega} = \mathbf{C}\mathbf{u}$ being the vorticity. We use a second-order kinetic-energy preserving discretization of this term as reported in Liska and Colonius (2017).

2.2.2 Fast LGF algorithm

Retaining the formally infinite grid, we employ the LGF for the corresponding discrete Poisson problem,

$$\mathbf{L}_Q \phi = \mathbf{f}, \quad \lim_{n \rightarrow \infty} \phi(\mathbf{n}) = 0, \quad (2.5)$$

$$\implies \phi = \mathbf{L}_Q^{-1} \mathbf{f} = \mathbf{G}_Q * \mathbf{f}, \quad (2.6)$$

where $\mathbf{f}(\mathbf{n}), \mathbf{s}(\mathbf{n}) \in \mathbb{R}^Q$, and \mathbf{n} denotes the trio of integers associated with the infinite lattice. \mathbf{G}_Q is the LGF which incorporates exact far-field boundary condition and $*$ denotes the discrete convolution.

Given a source term $\mathbf{f}(\mathbf{n})$ the solution can in principle be evaluated anywhere on the infinite lattice, but for a source with finite support, we only need do so at those lattice positions that are required to advance the solution. This is accomplished in practice by thresholding the source of the Poisson equation, which is in turn proportional to the vorticity field. Furthermore, this allows the solution to be spatially adaptive, as the active lattice points can be adjusted at each time-step. This process is described in detail in Liska and Colonius (2017), and summarized later in section 2.4.1.

To accelerate the evaluation of Eq. (2.6), a variant of the fast multipole method (FMM) is applied. Specifically we employ an FMM-based fast summation technique for a 3-D uniform Cartesian grid (Liska and Colonius, 2014) yields linear complexity and good parallel efficiency.

2.2.3 Integrating factor for the viscous term

Similar to the LGF, an integrating factor (IF) is defined as the solution operator to the discrete heat equation on an unbounded uniform Cartesian grid

$$\frac{df}{dt} = \kappa L_Q f, \quad f(\mathbf{n}, \tau) = f_\tau(\mathbf{n}), \quad (2.7)$$

$$\implies f(\mathbf{n}, t) = \left[H_Q \left(\frac{\kappa(t - \tau)}{(\Delta x)^2} \right) f_\tau \right] (\mathbf{n}, t), \quad t \geq \tau, \quad (2.8)$$

where $f \in \mathbb{R}^Q$, $\kappa > 0$ is a constant, $f_\tau(\mathbf{n})$ is a known source field, and H^Q is the integrating factor operator. The IF is a convolution with an exponentially decaying kernel, whereas the LGF kernel, $G_Q(\mathbf{n})$ in Eq. (2.6) decays as $1/|\mathbf{n}|$. The FLGF algorithm can be applied directly to this kernel (Liska and Colonius, 2016).

2.2.4 Half-explicit Runge-Kutta scheme

With the IF technique permitting an exact time integration of the viscous term, the remaining terms are discretized in time using a half-explicit Runge-Kutta (HERK) method (Hairer, Lubich, and Roche, 2006). HERK schemes exactly enforce algebraic constraints (in this case the divergence-free constraint), while using an explicit RK method to advance the differential equations. More traditionally, split methods are needed so that the viscous terms are integrated implicitly. With the IF technique the usage of split methods are obviated, which is discussed here.

By applying the IF operator $H_{\mathcal{F}}$ to Eq. (2.4) we obtain

$$\frac{dv}{dt} = -H_{\mathcal{F}} N \left((H_{\mathcal{F}})^{-1} v + u_\infty \right) - H_{\mathcal{F}} G p, \quad D(H_C)^{-1} v = 0, \quad (2.9)$$

where $v = H_{\mathcal{F}} u \in \mathbb{R}^{\mathcal{F}}$. Because the integrating factor $H_{\mathcal{F}}$ commutes with the gradient and the divergence operators G and D by Eq. (2.3), Eq. (2.9) is simplified to

$$\frac{dv}{dt} = -H_{\mathcal{F}} N \left(H_{\mathcal{F}}^{-1} v + u_\infty \right) - G H_C p, \quad Dv = 0. \quad (2.10)$$

Eq. (2.10) is integrated in time using the HERK scheme, which breaks down the total time integration into N_k subproblems $\cup_{k=0}^{N_k} [t_k, t_{k+1}]$. For each subproblem $[t_k, t_{k+1}]$, the HERK scheme takes an input of the velocity field $u(t_k)$, and output the velocity field $u(t_{k+1})$, which is defined as one timestep. The HERK scheme further breaks a timestep into many stages. At each stage it requires solving a system of equations of the following form,

$$\begin{bmatrix} (H_{\mathcal{F}}^i)^{-1} & G \\ D & 0 \end{bmatrix} \begin{bmatrix} u^i \\ p^i \end{bmatrix} = \begin{bmatrix} r^i \\ 0 \end{bmatrix}, \quad (2.11)$$

where $\mathbf{u} = \mathbf{H}_{\mathcal{F}}^{-1}\mathbf{v} \in \mathbb{R}^{\mathcal{F}}$, \mathbf{r} is a known right-hand side, and the superscript refers to quantities evaluated at the i -th stage of the HERK scheme. Eq. (2.11) can be solved using a block-LU decomposition:

$$\mathbf{u}^* = \mathbf{H}_{\mathcal{F}}^i \mathbf{r}^i, \quad \mathbf{D}\mathbf{H}_{\mathcal{F}}^i \mathbf{G}\mathbf{p}^i = \mathbf{D}\mathbf{u}^*, \quad \mathbf{u}^i = \mathbf{u}^* - \mathbf{H}_{\mathcal{F}}^i \mathbf{G}\mathbf{p}^i. \quad (2.12)$$

Again using the commutativity by Eq. (2.3) and the mimetic properties by Eq. (2.2), Eq. (2.12) is simplified to

$$\mathbf{L}_C \mathbf{p}^i = \mathbf{D}\mathbf{r}^i, \quad (2.13)$$

$$\mathbf{u}^i = \mathbf{H}_{\mathcal{F}}^i (\mathbf{r}^i - \mathbf{G}\mathbf{p}^i). \quad (2.14)$$

This simplified form involves the discrete Poisson equation, which is then solved with the FLGF technique discussed in section 2.2.2.

On the right-hand side of Eq. (2.11), \mathbf{r}^i is constructed using the information from the previous stages and the non-linear term at the current stage

$$\mathbf{r}^i = \mathbf{q}^i + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{i,j} \mathbf{w}^{i,j} + \mathbf{g}^i, \quad (2.15)$$

where Δt is the time-step length, \mathbf{g}^i is related to the nonlinear term given by

$$\mathbf{g}^i = -\tilde{a}_{i,i} \Delta t \mathbf{N} \left(\mathbf{u}^{i-1} + \mathbf{u}_{\infty} \left(t^{i-1} \right) \right), \quad t^i = t + \tilde{c}_i \Delta t, \quad (2.16)$$

and \mathbf{q}^i and $\mathbf{w}^{i,j}$ are recursively computed for $i > 1$ and $j < i$ using

$$\mathbf{q}^i = \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{q}^{i-1}, \quad \mathbf{q}^1 = \mathbf{u}^0 \quad (2.17)$$

$$\mathbf{w}^{i,j} = \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{w}^{i-1,j}, \quad \mathbf{w}^{i,i} = (\tilde{a}_{i,i} \Delta t)^{-1} (\mathbf{g}^i - \mathbf{G}\mathbf{p}^i), \quad (2.18)$$

with c_i and $\tilde{a}_{i,j}$ being the coefficients of a HERK scheme, and \mathbf{u}^0 being the velocity field at the beginning of the time-marching.

The current implementation uses a HERK scheme introduced by Brasey and Hairer (1993) with the coefficients given in Table (2.1), which corresponds to the Scheme B discussed in Liska and Colonius (2016). This scheme was chosen because it offers the highest order of accuracy for both the solution variable, velocity \mathbf{u} (third-order) and the constraint variable, pressure \mathbf{p} (second-order) among all three schemes considered in Liska and Colonius (2016).

0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	0	0
1	-1	2	0
	0	$\frac{3}{4}$	$\frac{1}{4}$

Table 2.1: Coefficients of the HERK scheme (Brasey and Hairer, 1993).

2.3 Navier-Stokes LGF on an AMR grid

As discussed in section 2.2, the mimetic properties of the differential operators and the commutativity between the differential operators and the LGF/IF operators are crucial for the algorithm (see the simplification of Eq. (2.9) and Eq. (2.12)). Furthermore, the LGF is only defined on a regular grid and the regularity of the uniform grid also in turn enables an efficient evaluation of the fast LGF algorithm (Liska and Colonius, 2014). However, an irregular grid does not possess those features. In this section we propose a novel AMR technique that preserves the desired properties.

2.3.1 Spatial discretization on an AMR grid

In this section an AMR grid used for the discretization of Eq. (2.1) is constructed in two steps. First, we define a series of uniform unbounded staggered Cartesian grids. Each grid is of the form introduced in section 2.2 but with different resolution $\{\mathbb{R}_k^Q\}_k$, where $k \in \mathbb{Z}^+$ refers to the resolution or grid level. We use the convention that \mathbb{R}_0^Q is the coarsest level and grid \mathbb{R}_{k+1}^Q is generated by evenly dividing every grid unit \mathbb{R}_k^Q into 2^d new units with d being the dimension of Eq. (2.1), and denote N_l as the maximum number of levels ($0 \leq k < N_l$).

We refer to the collection of uniform, unbounded grids as the *composite* grid, which is defined as a tensor product of the series of grids

$$\overline{\mathbb{R}^Q} := \otimes_{k=0}^{N_l} \mathbb{R}_k^Q. \quad (2.19)$$

We equip this new tensor space with an inner product that is induced from each \mathbb{R}_k^Q .

In the second step, an AMR grid is constructed as a subspace of the composite grid. More specifically, we define an AMR grid through a restriction operator. For each level k , a restriction operator $\Gamma_k^Q: \mathbb{R}_k^Q \rightarrow \mathbb{R}_k^Q$ is a linear functional defined as

$$\left(\Gamma_k^Q f\right)(\mathbf{n}) = \begin{cases} f(\mathbf{n}) & \text{for } \mathbf{n} \in \Omega_k \\ 0 & \text{otherwise} \end{cases} \quad (2.20)$$

for all $f \in \mathbb{R}_k^Q$, where $\{\Omega_k\}_k$ are regions that partition the whole space \mathbb{R}^3 , i.e.,

$$\Omega_l \cap \Omega_q = \delta_{lq} \Omega_l, \quad \cup_{q=0}^{N_l} \Omega_q = \mathbb{R}^3. \quad (2.21)$$

For simplicity we write Γ_k^Q as Γ_k since Q can be determined by the context. A full restriction operator for the composite grid, $\Gamma: \overline{\mathbb{R}^Q} \rightarrow \overline{\mathbb{R}^Q}$ is then defined by the tensor product

$$\Gamma^Q := \otimes_{k=0}^{N_l} \Gamma_k^Q. \quad (2.22)$$

We again simplify the notation Γ^Q to be Γ . Finally an AMR grid $\hat{\mathbb{R}}^Q \subset \overline{\mathbb{R}^Q}$ is defined as the image of the full restriction operator Γ , i.e.,

$$\hat{f} := \Gamma f \in \hat{\mathbb{R}}^Q, \quad \forall f \in \overline{\mathbb{R}^Q}. \quad (2.23)$$

A diagram for the composite grid and the AMR grid is shown in Fig. 2.1b.

2.3.2 Interpolation/coarsening operators

So far we have defined a composite grid function space $\overline{\mathbb{R}^Q}$ which consists of N_l unbounded uniform Cartesian grids, and an AMR grid $\hat{\mathbb{R}}^Q$ as a subspace of $\overline{\mathbb{R}^Q}$. By definition, the full restriction operator projects functions in $\overline{\mathbb{R}^Q}$ to the AMR grid $\hat{\mathbb{R}}^Q$. Numerically we only store information on the AMR grid. Assuming every part of the AMR grid is approximating the same continuous function, then the information on the composite grid can be approximated using interpolations and coarsening. In this section we introduce the interpolation/coarsening operators that fulfill this idea.

First, we denote an interpolation/coarsening operator between any two levels l and q as $P_{l \rightarrow q}^Q$ ($P_{l \rightarrow q}^Q$ is an interpolation when $l < q$, it is coarsening when $l > q$, and it is the identity when $l = q$). We construct $P_{l \rightarrow q}^Q$ as compositions of the P-operators between consecutive levels. For example, an interpolation operator between level l and q ($l < q$) is given by

$$P_{l \rightarrow q}^Q = P_{q-1 \rightarrow q}^Q \cdots P_{l+1 \rightarrow l+2}^Q P_{l \rightarrow l+1}^Q, \quad \text{for } 0 \leq l < q < N_l. \quad (2.24)$$

This construction will be shown favorable for the numerical efficiency in section 2.3.4. Since the AMR grid is defined by the regions $\{\Omega_k\}_k$ that partition the space, then for a given level, the information can be estimated anywhere using the P-operators. More specifically, given $\hat{f} = \otimes_k \hat{f}_k \in \hat{\mathbb{R}}^Q \subset \overline{\mathbb{R}^Q}$ on the AMR grid, the information on level k of the composite grid can be estimated by

$$f_k = \sum_{i=0}^{N_l} P_{i \rightarrow k}^Q \hat{f}_i + O(h^{N_p}) := P_k^Q \hat{f} + O(h^{N_p}), \quad (2.25)$$

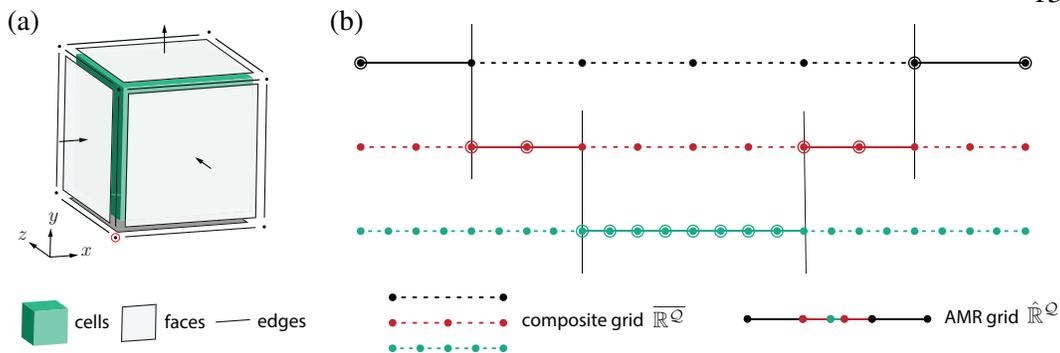


Figure 2.1: (a) Base unit of a finite-volume staggered grid. (b) 1-D diagram for the composite grid (dotted line) and the corresponding AMR grid (solid line with corresponding vertices in circle)

where the error convergence rate, N_p , is determined by the specific choice of interpolant. This is discussed in more detail below. In the second equality in Eq. (2.25), we defined another interpolation/coarsening operator between the AMR grid and the uniform grid on level k , $P_k^Q : \hat{\mathbb{R}}^Q \rightarrow \mathbb{R}_k^Q$. Similarly, a full P-operator between the AMR grid and the composite grid $P^Q : \hat{\mathbb{R}}^Q \rightarrow \overline{\mathbb{R}}^Q$ is defined by

$$P^Q = \otimes_{k=0}^{N_I} P_k^Q. \quad (2.26)$$

Together with the restriction operator Γ , one has the approximation relation between the AMR grid and the composite grid

$$\hat{f} = \Gamma f, \quad f \approx P^Q \hat{f}, \quad (2.27)$$

where $f \in \overline{\mathbb{R}}^Q$ and $\hat{f} \in \hat{\mathbb{R}}^Q$.

The general idea of our AMR technique is that, we consider the information on the AMR grid as being restricted from the ambient composite grid. At every time-step we can formally ‘recover’ the information on the composite grid from the AMR grid using the P-operator. Then, the information on every level of the composite grid is formally marched in time even if the specific operations need not be performed on the entire composite grid. At the end of the time-step, the solution on the composite grid is again restricted to the AMR grid through the restriction Γ . The most important feature of this process is that it need only be carried out on those portions of the composite grid that are needed to advance the AMR grid. Thus, while in principle the solution is defined on every grid level, only the subspace defined by the AMR grid is required in practice. We discuss how this is done in the next sections.

2.3.3 Differential operators

Differential operators are simply constructed for the composite grid by

$$\mathbf{A} = \otimes_{k=0}^{N_l} \mathbf{A}_k, \quad (2.28)$$

where $\mathbf{A}_k \in \{\mathbf{G}_k, \mathbf{D}_k, \mathbf{C}_k, \mathbf{L}_k\}$ are the corresponding discrete differential operators on grid \mathbb{R}_k defined in section 2.2.1. This construction also preserves the second-order accuracy, conservation properties, mimetic properties, and commutativity (for the composite grid) since for every level they are the same as the native ones defined in section 2.2.1. We note that differential operators need not be constructed directly for the AMR grid.

2.3.4 Fast LGF algorithm on the AMR grid

Before discussing the full algorithm for the Navier-Stokes equations, we provide details for applying the fast LGF/IF algorithms on the AMR grid by using the techniques derived in the proceeding section. The resulting algorithm is essentially the same as the aforementioned AMR-LGF algorithm (Dorschner et al., 2020). Here we use the composite-grid ansatz introduced above to reinterpret the algorithm and provide a more complete correction term than the one derived previously.

To solve the Poisson equation on the AMR grid, we use the information on the AMR grid to reconstruct the field on the composite grid, where the Poisson equation is hypothetically solved on every level through the FLGF technique, and the solution is then restricted back to the AMR grid. However, computationally one only has access to the AMR grid. To efficiently evaluate the aforementioned process, we consider the commutativity between the interpolation and the LGF convolution: instead of interpolating the information from a coarse grid to a fine grid and then applying the LGF convolution, we seek to apply the LGF convolution on the coarse grid first and then interpolate the solution to the fine grid. In other words, we seek a commutative P-operator $\overline{\mathbf{P}}_{k \rightarrow l}^Q$ that satisfies

$$\mathbf{L}_l^{-1} \mathbf{P}_{k \rightarrow l}^Q = \overline{\mathbf{P}}_{k \rightarrow l}^Q \mathbf{L}_k^{-1}, \quad (2.29)$$

where $0 \leq k, l < N_l$ denote to two distinct levels in the composite grid, and $\mathbf{L}_k, \mathbf{L}_l$ are the corresponding Laplacian operators. Solving for $\overline{\mathbf{P}}_{k \rightarrow l}^Q$ yields

$$\overline{\mathbf{P}}_{k \rightarrow l}^Q = \mathbf{L}_l^{-1} \mathbf{P}_{k \rightarrow l}^Q \mathbf{L}_k, \quad (2.30)$$

which suggests that $\bar{P}_{k \rightarrow l}^Q$ takes the form of a convolution. Note that $\bar{P}_{k \rightarrow l}^Q$ holds a similar composition relation as Eq. (2.24)

$$\bar{P}_{k \rightarrow l}^Q = \bar{P}_{q \rightarrow l}^Q \bar{P}_{k \rightarrow q}^Q. \quad (2.31)$$

A useful form of $\bar{P}_{k \rightarrow l}^Q$ from Eq. (2.30) is derived by considering $\bar{P}_{k \rightarrow l}^Q$ as the original interpolation $P_{k \rightarrow l}^Q$ with a correction

$$\begin{aligned} \bar{P}_{k \rightarrow l}^Q &= L_l^{-1} P_{k \rightarrow l}^Q L_k \\ &= P_{k \rightarrow l}^Q + L_l^{-1} (P_{k \rightarrow l}^Q L_k - L_l P_{k \rightarrow l}^Q) \\ &:= P_{k \rightarrow l}^Q + L_l^{-1} S_{k \rightarrow l}^Q, \end{aligned} \quad (2.32)$$

where the correction is in the form of a source term, given by applying the operator $S_{k \rightarrow l}^Q$ to the solution field on level k (Eq. (2.29)). One important property is that the correction source $S_{k \rightarrow l}^Q L_k^{-1}$ yields a faster decay than the original LGF kernel. For example, the correction term constructed from Lagrange polynomial interpolations is shown to oscillate and decay as $|\mathbf{n}|^{-4}$ (independent of the order of the chosen polynomial), whereas the LGF only decays as $|\mathbf{n}|^{-1}$.¹

The Poisson equation for the composite grid $\overline{\mathbb{R}^Q}$ subject to the far-field boundary condition is given by

$$L\phi = f, \quad \lim_{|\mathbf{n}| \rightarrow \infty} \phi(\mathbf{n}) = 0, \quad (2.33)$$

where $f, \phi \in \overline{\mathbb{R}^Q}$ and L is the Laplacian for the composite grid defined in section 2.3.3. Approximating the source term on the whole composite grid f using the AMR grid by Eq. (2.27) and solving for ϕ shows

$$\begin{aligned} \phi(\mathbf{n}) &= L^{-1} f \approx L^{-1} (P^Q \hat{f}) \\ &= \otimes_{k=0}^{N_l} L_k^{-1} \left(\sum_{i=0}^{N_l} P_{i \rightarrow k}^Q \hat{f}_i \right) \\ &= \otimes_{k=0}^{N_l} \left[\left(\sum_{i=0}^{k-1} \bar{P}_{i \rightarrow k}^Q L_i^{-1} \hat{f}_i \right) + L_k^{-1} \left(\sum_{i=k}^{N_l} P_{i \rightarrow k}^Q \hat{f}_i \right) \right] \\ &:= \otimes_{k=0}^{N_l} [\phi_1^k(\mathbf{n}) + \phi_2^k(\mathbf{n})] \end{aligned} \quad (2.34)$$

$$\hat{\phi}(\mathbf{n}) = \Gamma \phi(\mathbf{n}) \quad (2.35)$$

¹As an example, the source correction constructed from an interpolation of simple averaging decays to about 3×10^{-5} at 20 cells away from the center of the interpolation.

where $L^{-1} = \otimes_k L_k^{-1}$ is the LGF for the composite grid, $\hat{f} \in \hat{\mathbb{R}}$ lives only on the AMR grid, $\phi_1^k(\mathbf{n})$ is the partial solution corresponding to the source field from coarser levels, and $\phi_2^k(\mathbf{n})$ is the partial solution corresponding to the source contribution from all finer levels as well as level k itself. Eq. (2.34) uses the commutative \bar{P}^Q operator for the interpolations in $\phi_1^k(\mathbf{n})$. As shown by the diagram Fig. 2.2, the commutative construction avoids interpolating to a fine grid while yielding the same results. Eq. (2.34, 2.35) are effectively the AMR-LGF algorithm introduced

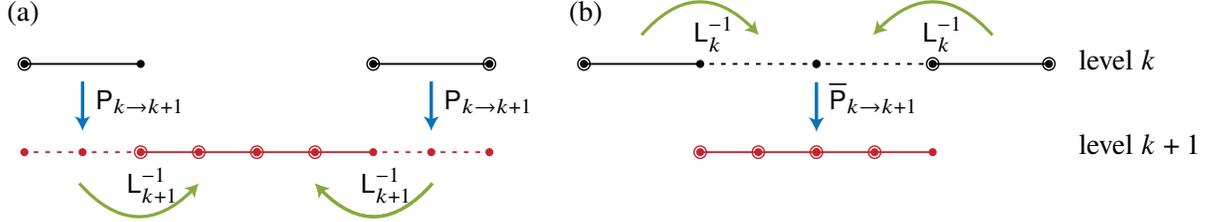


Figure 2.2: 1-D diagram for the commutative interpolation with the Laplacian to avoid constructing a fine grid on level $k+1$: (a) applying an interpolation $P_{k \to k+1}$ first and L_{k+1}^{-1} second; (b) applying L_k^{-1} first and $\bar{P}_{k \to k+1}$ second.

in Dorschner et al. (2020).

To summarize, the revised AMR-LGF algorithm is:

1. From fine to coarse levels, evaluate the source term $\sum_{i=k}^{N_l} P_{i \rightarrow k}^Q \hat{f}_i$ in $\phi_2^k(\mathbf{n})$ through coarsening. Because P-operators are defined as compositions of consecutive levels (Eq. (2.24)), this term is calculated cumulatively by

$$\sum_{i=k+1}^{N_l} P_{i \rightarrow k}^Q \hat{f}_i = P_{k+1 \rightarrow k}^Q \left(\sum_{i=k+2}^{N_l} P_{i \rightarrow k+1}^Q \hat{f}_i + \hat{f}_{k+1} \right). \quad (2.36)$$

2. $\phi_1^k(\mathbf{n})$ are also evaluated cumulatively but from coarse to fine levels

$$\begin{aligned} \phi_1^k(\mathbf{n}) &= \sum_{i=0}^{k-1} \bar{P}_{i \rightarrow k}^Q L_i^{-1} \hat{f}_i \\ &= \bar{P}_{k-1 \rightarrow k}^Q \left[L_{k-1}^{-1} \hat{f}_{k-1} + \sum_{i=0}^{k-2} \bar{P}_{i \rightarrow k-1}^Q L_i^{-1} \hat{f}_i \right] \\ &:= \bar{P}_{k-1 \rightarrow k}^Q \psi_{k-1} \end{aligned} \quad (2.37)$$

Using the source correction introduced in Eq. (2.32), Eq. (2.37) becomes

$$\phi_1^k(\mathbf{n}) = P_{k-1 \rightarrow k}^Q \psi_{k-1} + L_k^{-1} S_{k-1 \rightarrow k}^Q \psi_{k-1}. \quad (2.38)$$

This form avoids specific evaluation of the non-local \bar{P} -operators, i.e., source terms can be combined with $\phi_2^k(\mathbf{n})$ and Eq. (2.34) can be directly evaluated as

$$\phi(\mathbf{n}) = \otimes_{k=0}^{N_l} \left[P_{k-1 \rightarrow k}^Q \psi_{k-1} + L_k^{-1} \left(S_{k-1 \rightarrow k}^Q \psi_{k-1} + \sum_{i=k}^{N_l} P_{i \rightarrow k}^Q \hat{f}_i \right) \right], \quad (2.39)$$

where ψ_{k-1} is evaluated accumulatively from coarse to fine levels and L_k^{-1} is evaluated using the FLGF algorithm for the uniform grid. We define the combined source in Eq. (2.39) as S_k given by

$$S_k = S_{k-1 \rightarrow k}^Q \psi_{k-1} + \sum_{i=k}^{N_l} P_{i \rightarrow k}^Q \hat{f}_i, \quad (2.40)$$

which is utilized in the refinement indicator function to be introduced in section 2.4.2.

3. Lastly, the restriction operator Γ is applied by limiting the region needed for the interpolation and the region used in the FLGF algorithm. With the construction of the composite grid, AMR grid, P-operators, the restriction operator Γ , as well as the differential operators, the AMR-LGF algorithm in a nutshell is

$$\left[\Gamma L^{-1} P^Q \right] \hat{f}. \quad (2.41)$$

The approach developed here clarifies that the correction procedure developed in Dorschner et al. (2020) is associated with the non-commutativity between P-operators and the LGF operators, and it yields a more precise form of the correction as an additive source term. Though the source correction $S_{k-1 \rightarrow k}^Q \psi_{k-1}$ decays rapidly, it is non-local and requires an extended region up to a cut-off distance depicted in Fig. 2.3. More specifically the AMR grid is defined with a set of physical domains $\{\Omega_k\}_k$ (Eq. (2.21)), and the corresponding extended correction regions on level k are given by

$$\Omega_k^E = \{ \mathbf{x} : |\mathbf{x} - \mathbf{y}| \leq N_E \Delta x_k, \mathbf{y} \in \Omega_k, \mathbf{x} \notin \Omega_k \}, \quad (2.42)$$

where Δx_k is the cell width on grid level k and N_E controls the cut-off distance of the extended region.

To test the new LGF-AMR algorithm, including the extended source correction, we use the same test case as Dorschner et al. (2020) and solve a manufactured vorticity-streamfunction equation

$$\nabla^2 \Psi = -\omega, \quad (2.43)$$

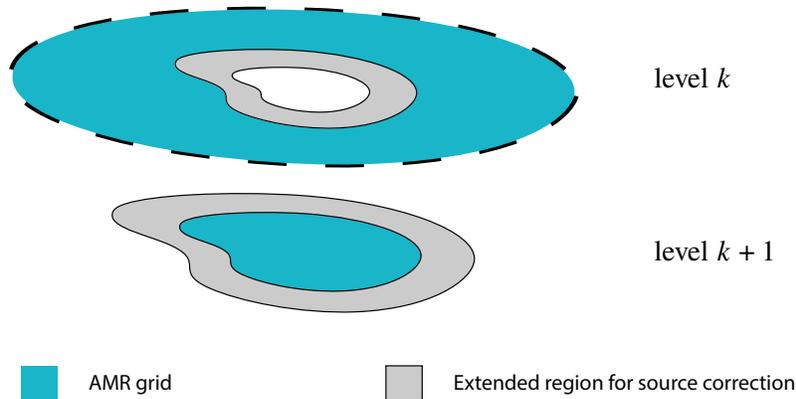


Figure 2.3: 2D diagram for the AMR grid (blue) and the extended correction region (gray) at each refinement level.

with the solution Ψ given by

$$\Psi(r, z) = f\left(\frac{\sqrt{(r-R)^2 + z^2}}{R}\right) \mathbf{e}_\theta, \quad f(t) = \begin{cases} c_1 \exp\left(-\frac{c_2}{1-t^2}\right) & \text{if } |t| < 1 \\ 0 & \text{otherwise} \end{cases}. \quad (2.44)$$

For this test we let $c_1 = 10^3$, $c_2 = 10$ and $R = 0.125$. For every level, an extended correction region of a cut-off parameter $N_E = 14$ is added, which corresponds to a relative source correction cut-off about 10^{-4} . We use the following criterion that a region on level k is refined if

$$\omega^k(\mathbf{x}) > \alpha^{L_R - k} \omega_{\max}, \quad \forall \mathbf{x} \in \Omega_k, \quad (2.45)$$

where L_R is the maximum number of refinement and $\alpha = 1/6$ is used.

Fig. 2.4a compares the L_∞ error of the solutions on the finest grid level for an increasing number of refinement levels, and Fig. 2.4b shows the error after left applying the discrete forward Laplacian L^Q to the numerical solutions. Three cases are considered: (1) without source correction (2) with correction but without an extended correction region and (3) with correction and with an extended correction region. For all tests the mesh topology is kept constant during the run. It can be seen that the proposed extended source correction not only improves the accuracy of the solution but also helps ensure the consistency with the discrete forward Laplacian. The computation rate and parallel efficiency are reported in Appendix A for the same test case up to 10^3 number of cores. Note that this result is tested with the correction region off. On the every level, the extended source correction region is only added to the ‘source’ in the FMM technique (section 2.5) but not the ‘target’, so the asymptotic computation rate is not affected.

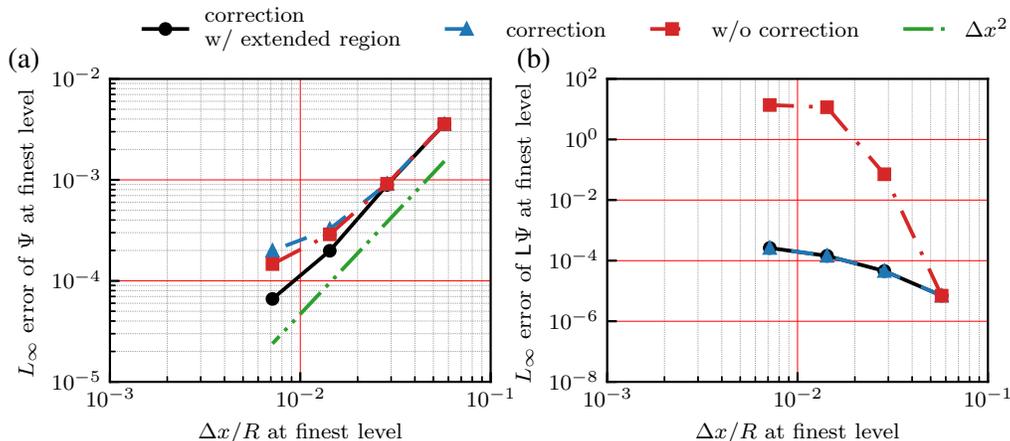


Figure 2.4: Convergence of solutions on the finest grid w.r.t. to the refinement levels from 0 to 3 with a criterion $\alpha = 1/6$ of (a) the numerical solution Ψ and (b) left applying the discrete Laplacian to the numerical solution, $L\Psi$. For each plot three cases are considered: without the correction (red); with correction but without an extended region (blue); and with correction and with an extended region (black). Across all three cases the mesh topology is kept the same.

2.3.5 Fast IF algorithm on the AMR grid

Similar to the LGF for the Laplacian, the IF for the composite grid is constructed as $H^Q = \otimes_k H_k^Q$ with H_k^Q being the integrating factor for each level defined by Eq. (2.8). Similar to Eq. (2.41), the fast AMR-IF algorithm is

$$\hat{f}(\mathbf{n}, t) \approx \left[\Gamma H^Q P^Q \right] \hat{f}_\tau(\mathbf{n}), \quad (2.46)$$

where $\hat{f}, \hat{f}_\tau \in \hat{\mathbb{R}}^Q$ live on the AMR grid. Note that the kernel of H^Q decays exponentially, which simplifies the implementation as one only needs to apply the interpolation/coarsening P -operator to an extended region and then apply the IF convolution before finally restricting the solution back to the AMR grid. Numerically the extended region used for the AMR-IF is the same as the one used for the AMR-LGF shown in Fig. (2.3), and the same cut-off parameter N_E for the extended region is used, which corresponds to a relative error less than 10^{-10} due to the exponential decay.

2.3.6 Navier-Stokes AMR-LGF-HERK algorithm

We now gather the elements developed in the preceding sections to construct the AMR technique for the full Navier-Stokes equations. As discussed previously, the main steps are to (a) provide an algorithm to formally recover the flow field everywhere on the composite grid and time-marching every level, and (b) restrict the solution back to the AMR grid such that only a small subset of the composite

grid need actually be computed. To achieve this we combine the half-explicit Runge-Kutta scheme discussed in section 2.2.4 with the AMR-LGF and AMR-IF algorithms derived in section 2.3.4 and section 2.3.5.

As we have constructed the differential operators and the LGF/IF operators on every level of the composite grid using the operators native to each unbounded uniform staggered grid, the mimetic properties and the commutativity are preserved. Thus Eq. (2.4-2.14) are formally the same, but with the corresponding operators and variables referring to the those for the composite grid. For example, the LGF and IF operators for the composite grid is given by

$$\mathbf{L}^{-1} = \otimes_{k=0}^{N_l} \mathbf{L}_k^{-1}, \quad \mathbf{H}_{\mathcal{F}}^i = \otimes_{k=0}^{N_l} \mathbf{H}_{\mathcal{F}} \left(\frac{(\tilde{c}_i - \tilde{c}_{i-1}) \Delta t}{(\Delta x_k)^2 \text{Re}} \right), \quad (2.47)$$

where the IF operator for the composite grid depends on the stage of the HERK scheme i , and the level of the composite grid k . Like the AMR-LGF algorithm described in section 2.3.4, the NS-LGF-AMR-HERK algorithm tries to approximate the right-hand-sides of each update equation on the *composite grid*, and restrict the solution back to the AMR grid in a way such that the full composite grid is never built, but rather only the regions that are required by the AMR grid.

The process of evaluating the equivalent form of the system of equations Eq. (2.11) for the composite grid can be broken down as follows.

1. Eq. (2.11) for the composite grid is also solved using the block-LU decomposition. Similarly because of the commutativity between the *composite grid* differential operators and the IF operator, Eq. (2.13, 2.14) can formally be solved with the composite grid LGF and IF given by Eq. (2.47). Again, in this framework we approximate only the right-hand sides using the information on the AMR grid

$$\mathbf{D}r^i = \mathbf{P}^C \widehat{\mathbf{D}r^i} + \epsilon_D, \quad (2.48)$$

$$r^i - \mathbf{G}p^i = \mathbf{P}^{\mathcal{F}} \left(\widehat{r^i} - \widehat{\mathbf{G}p^i} \right) + \epsilon_u, \quad (2.49)$$

where $\widehat{\mathbf{D}r^i} = \Gamma \mathbf{D}r^i$ and $\widehat{r^i} - \widehat{\mathbf{G}p^i} = \Gamma (r^i - \mathbf{G}p^i)$ are the restricted fields on the AMR grid, \mathbf{P}^C and $\mathbf{P}^{\mathcal{F}}$ are the interpolation and coarsening operators, and the associated approximation error terms ϵ_D and ϵ_u are

$$\epsilon_D = \mathbf{D}r^i - \mathbf{P}^C \widehat{\mathbf{D}r^i}, \quad \epsilon_u = (r^i - \mathbf{G}p^i) - \left[\mathbf{P}^{\mathcal{F}} \left(\widehat{r^i} - \widehat{\mathbf{G}p^i} \right) \right]. \quad (2.50)$$

These errors are controlled by the order of the interpolation/coarsening and the local grid resolution, which are discussed in more detail in the next section.

2. Notice that the solution to the pressure Poisson equation is only used in the form $\widehat{\mathbf{Gp}}^i$ and with Eq. (2.13) the solution after the restriction is

$$\widehat{\mathbf{Gp}}^i = \Gamma \mathbf{Gp}^i = [\Gamma \mathbf{GL}_C^{-1} \mathbf{P}^C] \widehat{\mathbf{Dr}}^i + \widetilde{\epsilon}_D, \quad \widetilde{\epsilon}_D = \Gamma \mathbf{GL}_C^{-1} \epsilon_D. \quad (2.51)$$

Here the AMR-LGF algorithm given by Eq. (2.41) is used. Note that because of the gradient operator \mathbf{G} before the restriction, solutions from the AMR-LGF are restricted to a grid which is one extra cell larger along the boundary on every level of the AMR grid.

3. After solving the pressure gradient $\widehat{\mathbf{Gp}}^i$, by Eq. (2.14), the updated velocity field at stage i after being restricted back to the AMR grid can be expressed as

$$\hat{\mathbf{u}}^i = \left[\Gamma \mathbf{H}_{\mathcal{F}}^i \mathbf{P}^{\mathcal{F}} \right] \left(\hat{\mathbf{r}}^i - \widehat{\mathbf{Gp}}^i \right) + \widetilde{\epsilon}_u, \quad \widetilde{\epsilon}_u = \Gamma \mathbf{H}_{\mathcal{F}}^i \epsilon_u. \quad (2.52)$$

Here the AMR-IF algorithm given by Eq. (2.46) is applied.

The aforementioned process requires right-hand sides $\hat{\mathbf{r}}^i$ and $\widehat{\mathbf{Dr}}^i$. The evaluation of $\widehat{\mathbf{Dr}}^i$ on the AMR grid is essentially the same as $\hat{\mathbf{r}}^i$, except that a restriction that is one extra cell larger along the boundary on every level of the AMR grid is used due to the divergence operator. $\hat{\mathbf{r}}^i$ is recursively defined using the solutions from previous stages by Eq. (2.15) and the process of updating $\hat{\mathbf{r}}^i$ using the idea of the composite grid is described as follows.

1. \mathbf{q}^i and $\mathbf{w}^{i,j}$ on the required portion of the composite grid are approximated using the AMR solutions from previous stages,

$$\mathbf{q}^i = \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{q}^{i-1} = \mathbf{H}_{\mathcal{F}}^{i-1} \left(\mathbf{P}^{\mathcal{F}} \hat{\mathbf{q}}^{i-1} + \epsilon_q \right), \quad (2.53)$$

$$\mathbf{w}^{i,j} = \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{w}^{i-1,j} = \mathbf{H}_{\mathcal{F}}^{i-1} \left(\mathbf{P}^{\mathcal{F}} \hat{\mathbf{w}}^{i-1,j} + \epsilon_w \right), \quad (2.54)$$

where ϵ_q and ϵ_w are the interpolation/coarsening errors. The solutions on the AMR grid are given by

$$\hat{\mathbf{q}}^i = \left[\Gamma \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{P}^{\mathcal{F}} \right] \hat{\mathbf{q}}^{i-1} + \widetilde{\epsilon}_q, \quad \widetilde{\epsilon}_q = \Gamma \mathbf{H}_{\mathcal{F}}^i \epsilon_q, \quad (2.55)$$

$$\hat{\mathbf{w}}^{i,j} = \left[\Gamma \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{P}^{\mathcal{F}} \right] \hat{\mathbf{w}}^{i-1,j} + \widetilde{\epsilon}_w, \quad \widetilde{\epsilon}_w = \Gamma \mathbf{H}_{\mathcal{F}}^i \epsilon_w. \quad (2.56)$$

Here the AMR-IF algorithm given by Eq. (2.46) is again used.

2. Similarly the nonlinear term $\widehat{\mathbf{g}}^i$ on the AMR grid is given by

$$\widehat{\mathbf{g}}^i = -\tilde{a}_{i,i} \Delta t \Gamma \left[\mathbf{P}^{\mathcal{F}} \mathbf{N} \left(\mathbf{u}^{i-1} + \mathbf{u}_{\infty} \left(t^{i-1} \right) \right) + \epsilon_{\mathbf{N}} \right], \quad (2.57)$$

with

$$\epsilon_{\mathbf{N}} = \mathbf{N} \left(\mathbf{u}^{i-1} + \mathbf{u}_{\infty} \left(t^{i-1} \right) \right) - \mathbf{P}^{\mathcal{F}} \mathbf{N} \left(\hat{\mathbf{u}}^{i-1} + \hat{\mathbf{u}}_{\infty} \left(t^{i-1} \right) \right), \quad (2.58)$$

which contains two sources: an interpolation/coarsening error and an aliasing error.

In summary, the NS equations are formally discretized on the composite grid and time integrated using the HERK scheme. At each stage of the RK time integration, the system of equations Eq. (2.11) for the composite grid is solved using the block-LU decomposition. By approximating the right-hand sides using the information from the AMR grid, the resulting algorithm applies the AMR-LGF algorithm for the pressure gradient $\widehat{\mathbf{Gp}}^i$, and uses the AMR-IF algorithm for the updated velocity $\hat{\mathbf{u}}^i$, and the intermediate fields $\hat{\mathbf{q}}^i$ and $\hat{\mathbf{w}}^{i,j}$.

2.3.7 Approximation errors

The four interpolation/coarsening errors $\epsilon_{\mathbf{u}}$, $\epsilon_{\mathbf{D}}$, $\epsilon_{\mathbf{q}}$ and $\epsilon_{\mathbf{w}}$ control the difference between the composite-grid and AMR-grid solutions. For the composite grid, the method inherits the second-order convergence properties associated with the existing FLGF-HERK algorithm; these were characterized and measured in previous work (Liska and Colonius, 2016). The additional errors associated with AMR have a local truncation error of order $O(\Delta x_k^2)$. This error will vanish subjected to global refinement of all levels together. We empirically demonstrate the convergence in section 2.6 by considering the evolution of a vortex ring.

Some additional observations can be made about the truncation errors. By Eqs. (2.48) to (2.58), the approximation errors can propagate from coarse to fine levels through the LGF and IF convolutions \mathbf{L}_C^{-1} and $\mathbf{H}_{\mathcal{F}}$ to produce $\tilde{\epsilon}_{\mathbf{u}}$, $\tilde{\epsilon}_{\mathbf{D}}$, $\tilde{\epsilon}_{\mathbf{q}}$ and $\tilde{\epsilon}_{\mathbf{w}}$. Meanwhile $\epsilon_{\mathbf{N}}$ poses another source of the error due to non-linearity (aliasing). As \mathbf{GL}_C^{-1} and $\mathbf{H}_{\mathcal{F}}$ are bounded operators, the corresponding errors $\tilde{\epsilon}_{\mathbf{u}}$ and $\tilde{\epsilon}_{\mathbf{D}}$ are well-behaved (the constant in the error term is finite).

Note that Eqs. (2.48) and (2.49) simultaneously approximate the terms r^i and $\mathbf{D}r^i$ using $\mathbf{P}^{\mathcal{F}}$ and $\mathbf{P}^{\mathcal{C}}$. A more consistent approach would be to only approximate r^i using interpolation/coarsening and evaluate $\mathbf{D}r^i$ accordingly. This would require a

commutative P-operator with the divergence operator

$$DP^{\mathcal{F}} = P^C D, \quad (2.59)$$

which could then be implemented using a similar correction step as in Eq. (2.38). Unfortunately, this approach requires a construction of a divergence operator for the AMR grid—in our simplified approach operators need only be defined on the composite grid, and we have thus presently opted for the former, simpler approach (by tolerating the additional error term).

A final point regarding these errors is that they are, in principle, not different from those associated with any AMR scheme. The spirit of AMR is to adapt the mesh according to the (measured) smoothness of the solution. In other words, the AMR scheme will attempt to minimize all the errors discussed in this section, subject to being balanced by the overall truncation error. This is achieved by the adaptation strategies described in the next section.

2.4 Adaptivity

We have thus far described the algorithm for solving the incompressible Navier-Stokes equations on an AMR grid by introducing approximations to the (theoretical) solution on a composite grid associated with interpolation/coarsening operators between grid levels, and appropriate restriction operators. The AMR grid involves a collection of regions $\{\Omega_k\}_k$, where k denotes the level of refinement. The collection $\{\Omega_k\}_k$ still contains at least one unbounded region (i.e. the coarsest grid). In this section we will discuss how to *adaptively* truncate the coarsest grid level and strategies for determining the adaptive truncation and adaptive refinement.

2.4.1 Adaptive truncation

The vortical regions in external flows are associated with the source term in either the pressure Poisson equation or the vorticity-streamfunction equation. A truncation of the computation domain is plausible since the vorticity field is compact, and it suffices to assume only the base level (coarsest grid) Ω_0 is infinite. The spatial truncation for the computational domain Ω_0 is adapted from Liska and Colonius (2016), where a formally unbounded staggered uniform grid of a single resolution was also truncated. We refer to Liska and Colonius (2016) for a more detailed explanation. Here we only provide a brief summary of the truncation algorithm and discuss how to combine it with the AMR technique.

Two types of convolutions are performed in the current algorithm, namely the LGF,

L_C^{-1} for the pressure Poisson equation and the IF, $H_{\mathcal{F}}$ for the velocity field. To ensure the solution in the region of interest is both accurate and minimal in extent, the corresponding source terms in these convolutions need to be restricted to regions where they have magnitude greater than a tunable threshold value.

The source term in the LGF convolution introduced in section § 2.3.6 is given by Dr^i . This term is approximately the divergence of the nonlinear term (Lamb vector), which is in turn proportional to the compact vorticity field. Given a threshold ϵ^* , a truncation of the base level domain Ω_0^{supp} for the source term Dr is defined as

$$\Omega_0^{\text{supp}} := \left\{ \mathbf{x} \in \mathbb{R}^3 : \frac{|Dr(\mathbf{x})|}{\|Dr\|_{\infty}} \leq \epsilon^* \right\}. \quad (2.60)$$

Ω_0^{supp} determines the domain needed to yield an accurate solution to the Poisson equation.

The source term in the IF convolution is the velocity field which however yields a much slower decay. To accurately and efficiently evaluate the IF convolution we make use of the following two properties: the kernel of the IF decays exponentially and the velocity field can be recovered from the vorticity. More specifically, to evaluate the solution of the IF convolution in the region of interest Ω_0^{soln} , only the velocity field in an extended region Ω_0^{xsoln} is needed, which is defined by

$$\Omega_0^{\text{xsoln}} := \{ \mathbf{x} \in \mathbb{R}^3 : |\mathbf{x} - \mathbf{y}| < d_{\text{IF}}, \quad \mathbf{y} \in \Omega_0^{\text{soln}} \}, \quad (2.61)$$

where d_{IF} is a cut-off distance for the exponentially decaying kernel. The velocity in the extended region is recovered using the discrete vorticity-streamfunction relation

$$\mathbf{u} = -\mathbf{C}^{\dagger} L_{\mathcal{E}}^{-1} \boldsymbol{\omega}, \quad (2.62)$$

where $\boldsymbol{\omega}$ is the discrete vorticity, $L_{\mathcal{E}}$ is the Laplacian for the edge quantities, \mathbf{C}^{\dagger} is the discrete curl for the edge quantities defined in section 2.2.1, and \mathbf{u} is the discrete velocity. We refer to this process as the velocity refresh. The velocity refresh is only needed for the base level and the vorticity is calculated from the coarsened velocity field from the AMR grid given by

$$\boldsymbol{\omega}_0 = \mathbf{CP}_0^{\mathcal{C}} \hat{\mathbf{u}}, \quad (2.63)$$

where $\hat{\mathbf{u}}$ is the velocity on the AMR grid and \mathbf{P}_0 is the coarsening operator defined by Eq. (2.25). The evaluation of Eq. (2.62) uses the FLGF algorithm for a single level, introduced in section 2.2.2. We also require $\Omega_0^{\text{supp}} \subset \Omega_0^{\text{soln}} \subset \Omega_0^{\text{xsoln}}$. The

computational domain of the base level AMR grid Ω_0 is truncated such that $\Omega_0^{\text{xsoln}} = \cup_{k=0}^{N_l} \Omega_k$.

Lastly we note that the velocity refresh need not be performed at every time-step. The frequency of the refresh depends on the decay of the IF kernel, and whether the base level mesh topology is updated. More details about the decay of IF kernel can be found in Liska and Colonius (2016).

2.4.2 Adaptive refinement

Adaptive refinement is achieved by updating the restriction operator to Γ' as the solution progresses. The updated velocity field u' on the new AMR grid is related to the original velocity field u by interpolation/coarsening

$$u' = \Gamma' P^{\mathcal{F}} \Gamma u. \quad (2.64)$$

To yield an accurate solution, the resolution of the different parts of the AMR grid needs to reflect the different scales in the flow, which are a-priori unknown.

The choice of indicator function used to invoke refinement/derefinement has been discussed in previous work on AMR. For instance, Berger and Colella (1989b) applies a Richardson extrapolation by comparing the time-marched solutions on both the coarse and fine mesh. The identification of vortical structures in the flow often relies on the usage of the gradient, the curvature and magnitude of the vorticity in the indicator functions (Almgren et al., 1998; Sussman et al., 1999; Popinet, 2003; Sitaraman et al., 2010), while the gradient of the density is often used to detect the existence of shocks (Almgren et al., 1998; Quirk, 1996; Papoutsakis et al., 2018). A combination of different indicators can also be used. For example, Kamkar et al. (2011) uses the Q-criterion with the Richardson extrapolation and Shenoy, Smith, and Park (2014) uses the vorticity field, the non-linear term, and the Q-criterion together.

Our NS-LGF-AMR-HERK scheme is mainly based on two algorithms: the AMR-LGF algorithm for the pressure Poisson equation and the AMR-IF algorithm for the velocity field involved in the viscous term. Both algorithms make use of the fundamental solutions defined on uniform grids by hypothetically interpolating/coarsening the source fields to the composite grid and solving on each level independently. These interpolations/coarsening provide information about the truncation error and can therefore be used as an adaptation indicator.

The current implementation uses a refinement indicator function that focuses solely on the source term involved in the AMR-LGF (Poisson) algorithm for the following two reasons. First the kernel of the IF yields an exponential decay, which results in a more localized error, whereas the LGF kernel decays much slower and the LGF convolution can instantly propagate the error to the whole flow field. Secondly the source term in the IF convolution, i.e., the velocity is smoother compared with the source term in the Poisson equation which is proportional to the vorticity field, and therefore it suffices to resolve the vorticity on the AMR grid.

More specifically, we propose to use the combined source term S_k from Eq. (2.40) as the criterion. Note that S_k is defined on the AMR grid as well as the extended source correction region introduced in section 2.3.4. At time t , the AMR mesh at grid point \mathbf{n} on level k , or on level $k - 1$ with an overlapping extended correction region on level k is refined when

$$S_k(\mathbf{n}, t) > \alpha^{N_l - k} S_{\max}(t), \quad (2.65)$$

where $0 < \alpha < 1$ is a constant, N_l is the prescribed maximum refinement levels, and $S_{\max}(t)$ is a quantity that renders the criterion dimensionless. To make the refinement as efficient as possible, this quantity should monitor when the prescribed maximum resolution is most limited during the time horizon $[0, t]$ for an on-going simulation. The current implementation uses the following form,

$$S_{\max}(t) = \max_{\tau < t, B_n \in B} \text{RMS}_{\mathbf{n} \in B_n} [S_k(\mathbf{n}, \tau)], \quad (2.66)$$

where RMS refers to the root mean square, B_n denotes a block of computational cells, and B denotes the union of cell blocks that partition the AMR grid. More details are discussed in section 2.5. We choose the maximum rms of the combined source term over all cell blocks because it estimates the least resolved region represented by the block where the maximum is reached, and as a statistical quantity it is less affected by numerical noise. Similarly, a region on level k is coarsened when

$$S_k(\mathbf{n}, t) < \beta \alpha^{N_l - k} S_{\max}(t), \quad (2.67)$$

where $0 < \beta < 1$ is a constant to avoid constant changes in the refinement levels due to small oscillations around the refinement criterion.

The combined source term S_k consists of two parts: the source term in the pressure Poisson equation Dr^i defined in section 2.3.6, and the source correction term defined in Eq. (2.32). The former approximates the divergence of the non-linear term which

is proportional to the vorticity field. The later is related to the difference between the partial solutions (corresponding to the partial source from all coarser grids) on the coarse grid and the fine grid which is associated with the Richardson extrapolation. Using the combined source term has the benefits of an automatic incorporation of both the vorticity criterion and a Richardson extrapolation process using only one non-dimensionalization parameter without additional numerical expense.

2.5 Implementation

In this section the implementation of the Navier-Stokes LGF-AMR-HERK algorithm and the parallelization are briefly summarized. The development of the solver is based on Dorschner et al. (2020) and the same data structure is adopted here. The solver is written in C++ and uses MPI for parallel communications. The code uses a block-structured computational grid (i.e. the smallest unit for grid addition/removal and refinement/derefinement is a block of N_b^3 computational cells) and the blocks are also used in the refinement criterion Eq. (2.66). The current implementation uses $N_b = N_E = 14$. Those blocks are further organized using a tree structure (octree in 3D), where every node (octant) maps to a block. The reason for using the octree is that the AMR-LGF algorithm used in the NS-LGF-AMR-HERK scheme applies the FMM algorithm on each level of the computational grid, and each FMM operation uses the hierarchical subtree structure for the calculating the far-field interaction (Dorschner et al., 2020). Different from Dorschner et al. (2020), where each leaf corresponds to a cubic domain in physical space, the current solver does not have this requirement since the extended correction regions can overlap with the physical space.

As in Dorschner et al. (2020), a server-client model is used for the parallelization, where the server has the tree information but does not store any data, whereas each client only stores a part of the octree with its data. At the beginning of a simulation, the server guesses a mesh topology according to the given initial condition, anticipates the load, and distributes the whole tree to the clients. While the work is mainly done by the clients during the run, the server receives the adaptivity requests (spatial addition/removal and local refinement/derefinement) from its clients, finds a new compatible mesh topology (smooth in the transition of refinement levels), calculates a new load distribution, and sends the adaptivity instructions back to the clients to transfer the data. The server is also responsible for identifying the subtree used in each FMM calculation during the AMR-LGF algorithm.

The LGF kernel decays geometrically and is identical regardless of the grid level. Numerically, we store the exact values for near points, and use asymptotic expansions for points far away (Liska and Colonius, 2014; Dorschner et al., 2020). The IF kernel used in the AMR, on the other hand, depends on not only the stage of the HERK scheme, but also the grid level k as suggested by Eq. (2.47). However since it decays exponentially, one can still numerically calculate and store the exact values needed for all stages and levels during the initialization.

Since the IF kernel decays faster than the LGF, it requires fewer neighbor contributions. Numerically, one evaluation of the AMR-IF algorithm uses about 10% of the execution time compared with the AMR-LGF algorithm. The three stage HERK scheme considered here requires the application of the AMR-LGF algorithm at each stage, using around 60% of the total execution time. The HERK scheme on the other hand applies the AMR-IF algorithm to vector fields 5 times, contributing another 30% of the total execution time. One additional factor that affects the solver speed is the ‘velocity refresh’ procedure introduced in section 2.4.1 which requires solving the vector Poisson Eq. (2.62) for the base level only. However the ‘velocity refresh’ need not be performed at each time-step (Liska and Colonius, 2016). For the numerical tests considered in the following sections, we observe an additional contribution to the execution time of less than 15%.

2.6 Verification

The Navier-Stokes AMR-LGF-HERK algorithm introduced in section 2.3.6 is verified by considering a fat-cored vortex ring with an initial vorticity distribution of the form

$$\omega_\theta(r, z) = \begin{cases} \alpha \frac{\Gamma}{R^2} \exp(-4s^2/(R^2 - s^2)) & \text{if } s \leq R \\ 0 & \text{otherwise} \end{cases}, \quad \omega_z(r, z) = 0, \quad (2.68)$$

where R is the radius of the vortex ring, $s^2 = z^2 + (r - R)^2$, and we let $\alpha \simeq 0.54857674$ so the total circulation integrates to the parameter Γ . The Reynolds number is set to $\text{Re} = \Gamma/\nu = 1000$, and the initial velocity field is calculated using the discrete vorticity-streamfunction relation Eq. (2.62). The convergence study is performed by considering a series of runs with different levels of refinement and grid resolution: We vary the grid resolution on the base level from $\Delta x_{\text{base}}/\Delta x_0 = 2^{-3}, \dots, 2^0$, where $\Delta x_0 = R/4.7$ is a constant, and the grids are repeatedly refined using a refinement criterion $\alpha = 1/4$ (see section 2.5) to reach the same finest resolution $\Delta x_{\text{fine}}/\Delta x_0 = 2^{-3}$. The grid topology is kept constant during the test and all cases

are performed with $\Delta t/\Delta x_{\text{base}} = 0.35 \times 2^{-N_l}$ up to 128 time-steps for the finest case. Finally the reference solution is performed using a uniform grid at double of the finest test resolution.

The L_∞ convergence in the velocity field for all ten cases are shown in Fig. 2.5 in terms of both the base level and the finest level grid resolution. The errors are calculated by interpolating the reference solution onto the coarse grids using simple (2nd-order) averaging of nearest points. The plots show second-order convergence with the grid resolution for a fixed refinement level. Furthermore, with the aforementioned refinement criterion, adding a refinement level would produce a solution with comparable accuracy as refining the whole AMR grid at once. This verifies the efficacy of the AMR, i.e. we achieve better computational efficiency through local refinement. The computation saving in the spatial adaptivity and the nodal distribution will be further discussed in section 2.7.

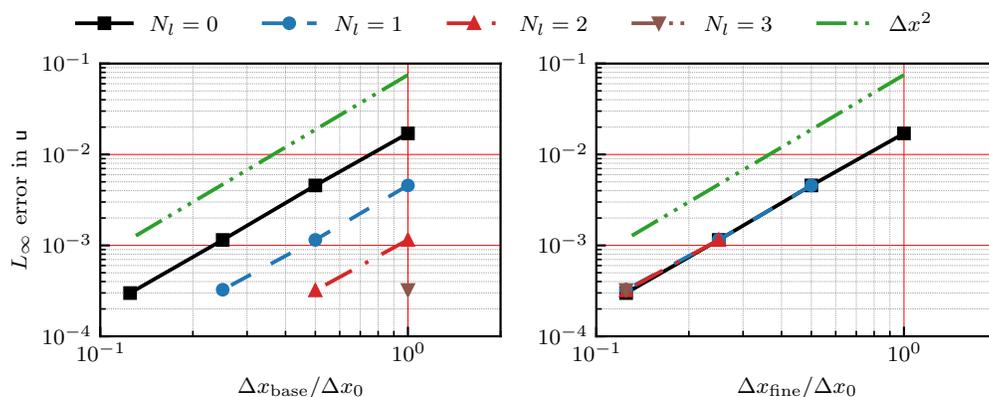


Figure 2.5: L_∞ velocity convergence of the NS-AMR-LGF solver wrt (a) the base level grid resolution, and (b) the finest level grid resolution, for $N_l = 0, 1, 2, 3$ respectively.

2.7 Collision of vortex rings

Vortex ring collisions are readily created in experiments (Oshima, 1978; Lim and Nickels, 1992). One notable feature of the vortex ring collision is that smaller secondary flow structures can develop at low Reynolds numbers, and turbulence cloud can form almost instantaneously at high Reynolds numbers, creating a wide range of length scales through a complex process of instabilities and vortex interaction (McKeown et al., 2018). The simulation of a vortex ring collision not only requires the numerics to be able to accurately and efficiently capture the fast and irregular changes in the flow configuration (e.g. the vortex ring radii can grow over 6 times during the expansion in the test case to be discussed), but at the same time also

challenges the AMR scheme to add minimum noise to the flow as the transition is sensitive to perturbations.

In this section we use the collision of two thin vortex rings at high Reynolds number to demonstrate the Navier-Stokes AMR-LGF-HERK solver with its spatial and refinement adaptivity. We assume that each vortex ring has an initial vorticity profile

$$\omega_\theta(r, z) = \frac{\Gamma}{\pi\delta^2} \exp\left(-\frac{z^2 + (r - R)^2}{\delta^2}\right), \quad \omega_z(r, z) = 0, \quad (2.69)$$

where we set $\text{Re} = \Gamma/\nu = 7500$, and $\delta/R = 0.2$ controls the width of the vortex ring. The initial distance between the two vortex rings is set to $R + \delta$ to mitigate the initial interaction. We consider three cases with the maximum refinement levels $N_l = 0, 1, 2$ respectively, and keep the resolution on *the finest level* constant with $\delta/\Delta x_{N_l} = 16$. i.e., the $N_l = 0$ case is using the finest grids everywhere with the maximum number of computational cells. Case with $N_l = 1$ and 2 are using decreasingly fewer numbers of cells with AMR. Across all three cases the ratio $\Delta t/\Delta x_{N_l} = 0.35$ is held constant. Both the spatial computational domain and the refinement regions are allowed to fully adapt. We use a spatial adaptive truncation threshold $\epsilon^* = 10^{-4}$ defined in Eq. (2.60) and a refinement criterion $\alpha = 1/4$ and $\beta = 0.75$ defined in Eq. (2.65, 2.67). Perturbations are added to the initial conditions of each vortex ring to accelerate the transition. The initial perturbation follows the recipe in Shariff, Verzicco, and Orlandi (1994), where the radii of the two vortex rings are independently perturbed with Fourier modes of uniform magnitude and random phases. For this test, perturbations are added to the first 32 modes with a magnitude of 3×10^{-4} relative to the unperturbed vortex ring diameter $2R$.

The evolution of the vortex ring collision taken from the case $N_l = 1$ is shown in Fig. 2.6. As the rings collide, they expand rapidly about the impact center plane, leaving a pair of thin vortex sheets trailing the leading vortices. The leading vortex pair becomes narrower, and as the vortex sheets are stretched, they eventually tear off, producing two disjoint circles. Both the Crow instability and the elliptic instability develop with the expansion which can be clearly observed around $t \Gamma/R^2 \sim 15$. The symmetry is broken, and finally the vortex ring pair transitions into turbulence.

To quantitatively compare the AMR cases with the run $N_l = 0$, we apply two statistical measures, namely the kinetic energy $\mathcal{K}(t)$ and the enstrophy $\mathcal{E}(t)$ given

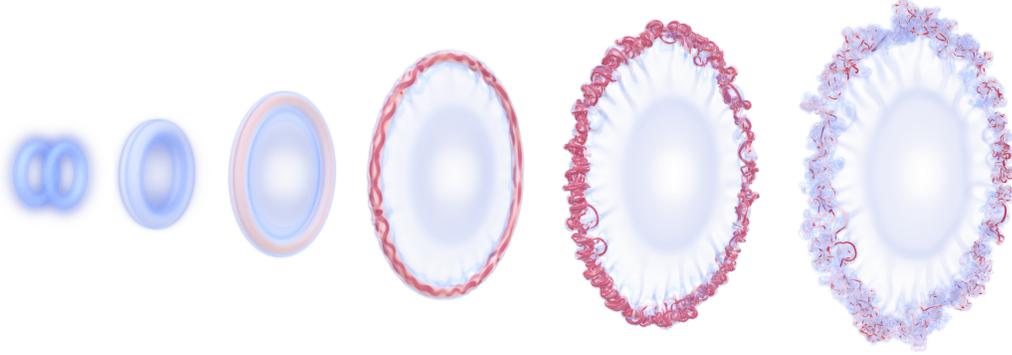


Figure 2.6: Evolution of the vortex ring collision at $\text{Re} = 7500$ from the case $N_l = 1$. From left to right, vorticity isosurfaces $|\omega R^2/\Gamma| = [0.5 - 7.5]$ are given for $t\Gamma/R^2 = 0, 4.5, 9.0, 15.1, 19.9,$ and 25.2 .

by

$$\mathcal{K}(t) = \int_{\mathbb{R}^3} \mathbf{u} \cdot (\mathbf{x} \times \boldsymbol{\omega}) d\mathbf{x}, \quad \mathcal{E}(t) = \frac{1}{2} \int_{\mathbb{R}^3} |\boldsymbol{\omega}|^2 d\mathbf{x}. \quad (2.70)$$

As shown in Fig. 2.7, the expansion of the vortex ring pair is accompanied with a decay in the kinetic energy and a growth in the enstrophy. The turbulence transition starts around around $t\Gamma/R^2 \sim 15$ with an acceleration in enstrophy growth. The enstrophy reaches a maximum at $t\Gamma/R^2 \sim 20$, which corresponds to the fifth flow visualization in Fig 2.6. We see the results with varying numbers of refinement levels ($N_l = 1, 2$) agree well with the uniform grid simulation ($N_l = 0$) and predict the transition time accurately. As transitional flows are very sensitive to the noise, it suggests the extra numerical perturbation from the AMR scheme is lower than the initial perturbation.

Fig. 2.8 compares the mesh topology across the three cases for the time period ($t\Gamma/R^2 = [0 - 20]$) where the top row shows the side view of the same vorticity isosurfaces as in Fig. 2.6, and the second to the fourth rows show the mesh topology over a horizontal cross-section about the center. The flow fields and the mesh topology in 3D for the non-AMR case ($N_l = 0$) and the AMR simulation ($N_l = 1$) at $t\Gamma/R^2 = 19.9$ are shown in Fig. 2.9, where we see the evolution compares well even after the entire transition period, and under the proposed criterion the refinement regions accurately surround the leading vortices while a lower resolution grid is used for the remaining regions.

The computational saving for all three cases is reported in Table 3.2, where the numbers of cells used in the NS-AMR-LGF-HERK scheme over the same time

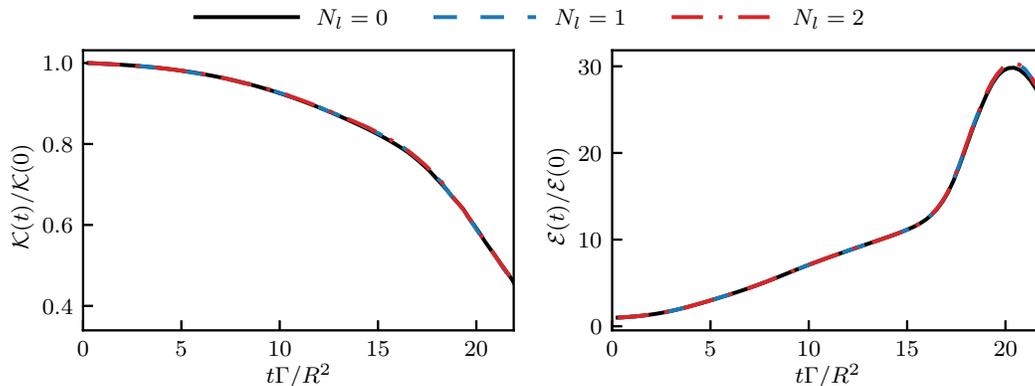


Figure 2.7: Evolution of the kinetic energy $\mathcal{K}(t)$ and enstrophy $\mathcal{E}(t)$ for the thin vortex ring collision at $\text{Re} = 7500$ for maximum refinement levels $N_l = 0, 1$, and 2. Note that the $N_l = 0$ corresponds to the case that uses uniform finest cells for the whole adaptive domain (the most number of cells).

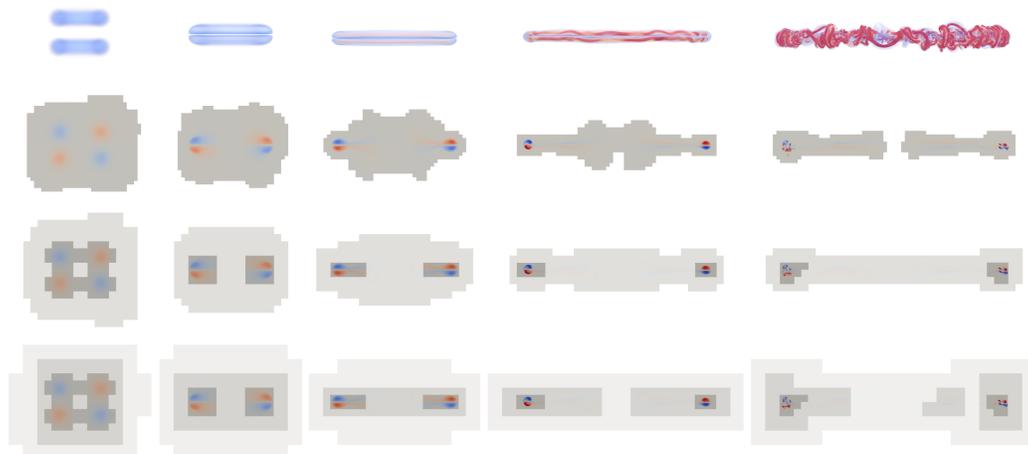


Figure 2.8: Mesh topology over the cross-section for $N_l = 0, 1$, and 2. The top row shows the flow evolution at $|\omega R^2/\Gamma| = [0.5 - 7.5]$ are given for $t\Gamma/R^2 = 0, 4.5, 9.0, 15.1$, and 19.9. The second to the fourth row show the corresponding mesh topology for $N_l = 0, 1, 2$. For all cases, coarse mesh to fine mesh are shown from light gray to dark gray.

horizon are compared with a static rectangular domain of a minimum bounding box with the same finest resolution everywhere. It can be seen that by involving the spatial adaptivity with the AMR, under the proposed criterion a factor of $10 \sim 20$ of reduction in computational cells is achieved.

2.8 Concluding remarks

We proposed an AMR technique to enhance the LGF approach for solving viscous, incompressible flows on unbounded domains. We consider the AMR grid as a subset of a composite grid that is constructed from a series of unbounded uniform

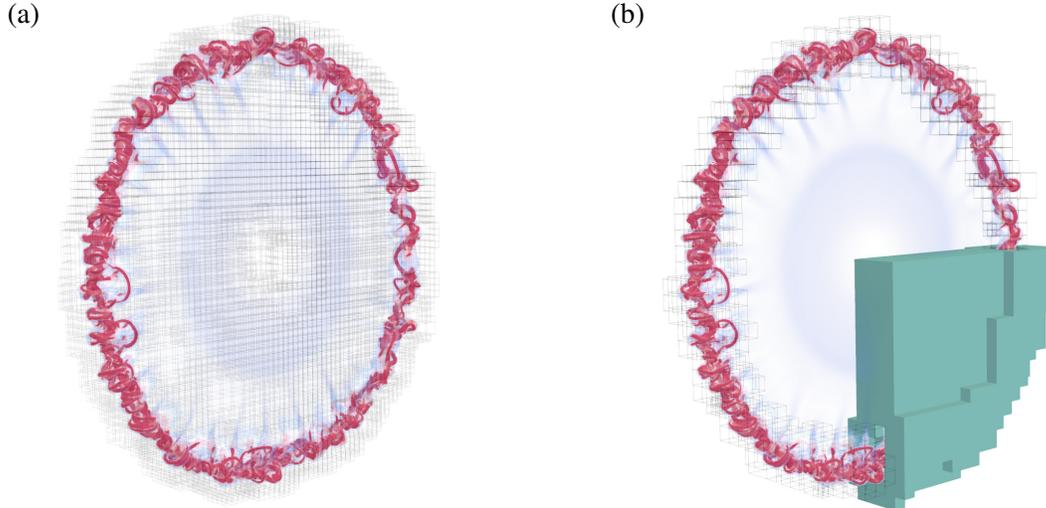


Figure 2.9: Flow visualization and mesh topology at $t\Gamma/R^2 = 19.9$ for (a) the non-AMR case ($N_l = 0$), and (b) the AMR case ($N_l = 1$). The computational blocks for the finest grid are shown in transparent boxes and a quarter of coarse grid for $N_l = 1$ is shown in green.

$t\Gamma/R^2$	0	4.5	9.0	15.1	19.9
$N_l = 0$	13.0%	12.9%	13.9%	17.0%	24.0%
$N_l = 1$	5.2%	4.7%	5.8%	6.9%	9.9%
$N_l = 2$	4.3%	4.5%	4.5%	5.4%	9.7%

Table 2.2: Number of computational cells for $N_l=0, 1$ and 2 , compared with a static rectangular domain of minimum bounding box over the time horizon $t\Gamma/R^2 = [0 - 20]$.

staggered Cartesian grid of differing resolution. Differential and LGF/IF operators are constructed for the composite grid and preserve the mimetic properties and the commutativity of the original IBLGF scheme. Interpolation/coarsening P-operators are defined through composition and their commutation with the aforementioned operators was studied. Based on this analysis, we refined the original AMR-LGF algorithm (Dorschner et al., 2020) for solving the 3-D Poisson equation subject to free-space boundary conditions. The P-operators are applied to formally recover the information on the whole composite grid from the AMR grid, where the Poisson equation is hypothetically solved on every level using the FLGF technique, before the solution is restricted back to the AMR grid. We also showed that this hypothetical process can be evaluated efficiently by commuting the interpolation and the LGF convolution, resulting in an extended source correction and improved accuracy of the AMR-LGF technique.

The Navier-Stokes equations were then discretized on this composite grid using a

second-order FV scheme, and we extended the AMR technique to incorporate the IF technique for the viscous term, and a Runge-Kutta (HERK) scheme for the resulting differential-algebraic equations. An incompressible Navier-Stokes method is then designed where we formally advance the flow fields on all levels of the composite grid using information from the AMR grid, but restricting the resulting computation back to the AMR grid obviates the need for all but a small subset of the composite grid.

Because the LGF represents the solution to the Poisson equation as a convolution of its source term, i.e. the divergence of the Lamb vector in the momentum equation, we construct an efficient and accurate refinement criterion that naturally tracks the associated truncation errors that are associated with interpolation and coarsening of the sources on different grid levels.

The Navier-Stokes solver was verified to give second-order accuracy through a refinement study of a fat-cored vortex ring. We also demonstrated the capabilities and performance by simulating the collision of thin-cored vortex rings at $Re = 7500$. We showed the AMR simulations agree well with the simulation using a uniform grid for the entire laminar and transitional period, while providing significant reductions in computational cells.

Lastly we note that the computational saving from the AMR depends on the scale separation present in the physics. In this chapter we restrict our attention to flows in the free space without bluff bodies. The combination of the AMR framework and the IB method for external flows with immersed surfaces is introduced in the next chapter.

IMMERSED BOUNDARIES

3.1 Introduction

In this chapter we further extend the AMR-LGF framework introduced in Chapter 2 to simulate external flows around complex immersed geometries at high Reynolds numbers via the IB method. We show that the adaptivity permits the thin boundary layers to be resolved more efficiently and at the same time retains the desired properties of the LGF method for flow simulations on otherwise unbounded domains.

As discussed in the Chapter 1, we focus on the specific type of IB technique termed the distributed Lagrange multiplier (DLM) or projection method (Colonius and Taira, 2008). As in other IB techniques, the DLM method solves the flow on an Eulerian grid and treats the immersed surfaces in a Lagrangian framework. The immersed surfaces are discretized on a set of IB points and the no-slip boundary condition is imposed by interpolating the flow field at those IB points. More specifically in the DLM method, the forces associated with the IB points are considered as Lagrange multipliers and the no-slip boundary condition is satisfied exactly. Apart from the benefit of an exact boundary condition, the DLM method also has the advantage that the modified incompressible Navier–Stokes equations can be solved using a projection method with an efficient numerical procedure that is analogous to the traditional fractional-step method (but free of splitting errors).

In the continuous setting, the IB method introduces an extra singular forcing term to the Navier-Stokes equations which are given by,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} + \int_{\mathcal{B}(t)} \mathbf{f}(\xi, t) \delta(\mathbf{X}(\xi, t) - \mathbf{x}) d\xi, \quad (3.1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (3.1b)$$

$$\int_{\mathbb{R}^3} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(\xi, t)) d\mathbf{x} = \mathbf{u}_{\mathcal{B}}(\xi, t), \quad (3.1c)$$

where \mathbf{u} is the velocity field, p is the pressure, $\delta(\mathbf{X}(\xi, t) - \mathbf{x})$ is the singular delta function which is later regularized with the discrete delta function, and \mathbf{f} is the force along the surface which is parameterized by $\mathcal{B}(t)$. For external flows, the velocity \mathbf{u} and the pressure p are subject to far-field boundary condition $\mathbf{u}(\mathbf{x}, t) \rightarrow$

$\mathbf{0}$, $p(\mathbf{x}, t) \rightarrow p_\infty$ as $|\mathbf{x}| \rightarrow \infty$, and the unknown forces \mathbf{f} are treated as Lagrange multipliers such that the no-slip constraints are satisfied.

The chapter discusses how to discretize and solve the aforementioned Eq. (3.1) using a multi-resolution grid and the LGF method. In section 3.2, the original IB-LGF method for grid of single uniform resolution is briefly reviewed. In section 3.3, we discuss how to equip the Navier-Stokes FLGF-AMR-HERK algorithm introduced in section 2.3.6 with the IB method. In section 3.4, we discuss an important advantage of the AMR-IB-LGF method that spatial truncation and refinement adaptivity can be applied to a minimum spatial domain that only contain vortical flow areas to admit accurate solutions. Section 3.5 discusses the numerical procedure to solve the associated forcing equation. The combined AMR-IB-LGF-HERK algorithm is validated and tested in section 3.6 and 3.7, where flow around a sphere and a delta-wing at low and high Reynolds numbers are considered.

3.2 Immersed boundary method for external flows using LGF on a uniform grid

In this section we briefly review the IB-LGF algorithm developed by Liska and Colonius (2017), which will be extended to the multi-resolution framework in the next section.

3.2.1 Discretization on unbounded uniform grid

As mentioned above, the IB-LGF method uses two discretizations. The flow field is discretized on a formally unbounded uniform, staggered Cartesian grid of single resolution. We use \mathbb{R}^Q to represent the corresponding grid function spaces with values defined on $Q \in \{C, \mathcal{F}, \mathcal{E}, \mathcal{V}\}$, where centers (C) and vertices (\mathcal{V}) store scalar quantities, and faces (\mathcal{F}) and edges (\mathcal{E}) store vector quantities. The immersed surface \mathcal{B} , on the other hand, is discretized using a set of IB points that are distributed on the immersed surface ‘evenly’, the concept of which will be made clear later.

The resulting discrete system of equations is given by

$$\frac{du}{dt} + N(u, t) = -Gp + \frac{1}{\text{Re}}L_{\mathcal{F}}u + R(t)f \quad (3.2a)$$

$$Du = 0, \quad (3.2b)$$

$$E(t)u = u_{\mathcal{B}}(t), \quad (3.2c)$$

where $u \in \mathbb{R}^{\mathcal{F}}$ is the discrete velocity field, G , D are the gradient and divergence operators, $L_{\mathcal{F}}$ is the Laplacian for $\mathbb{R}^{\mathcal{F}}$, $N(u, t)$ approximates the non-linear term, and

$u_{\mathcal{B}}$ is the prescribed velocity at the body surfaces.

Note that the delta functions in Eq. (3.1b), which represent the singular surface forces along the immersed surfaces, are regularized with the discrete delta functions in $R(t)$ and $E(t)$, where $R(t)$ smears the discrete point IB forces to the staggered Cartesian grid, whereas $E(t)$ interpolates the field information from the flow grids back to the IB point. They are defined as

$$[Eu](i, t) = (\Delta x)^3 \sum_{\mathbf{n} \in \mathbb{Z}^3} u(\mathbf{n}, t) \delta_{\Delta x}(\mathbf{x}_{\mathcal{F}}(\mathbf{n}) - \mathbf{X}(\xi_i, t)), \quad (3.3)$$

$$[Rf](\mathbf{n}, t) = \sum_i f(i, t) \delta_{\Delta x}(\mathbf{x}_{\mathcal{F}}(\mathbf{n}) - \mathbf{X}(\xi_i, t)), \quad (3.4)$$

where i refers to the i -th IB point with coordinate $\mathbf{X}(\xi_i, t)$, $\delta_{\Delta x}$ is the discrete delta function for a uniform grid with spacing Δx , and $\mathbf{x}_{\mathcal{F}}(\mathbf{n})$ is the grid coordinate for cell \mathbf{n} .

The system of equations can be made self-adjoint if one defines the smearing R as the dual operator of the interpolation E up to a scaling, i.e.,

$$E f = R^\dagger \hat{f}, \quad (3.5)$$

where $f = (\Delta x)^3 \hat{f}$ with Δx being the cell width of the underlying fluid grid. This definition allows the usage of more efficient numerical solvers designed for self-adjoint systems such as conjugate-gradient method. More about the fast linear solver is discussed in section 3.4.

We require the IB points to be ‘evenly’ distributed, meaning that the distance between nearest IB points $s_{\mathcal{B}}$ need to approximately equal to the grid spacing of the underlying Cartesian mesh Δx . Numerically we often require a mean ratio of $s_{\mathcal{B}}/\Delta x = 1 \sim 2$. The idea is that having IB points too close to each other will result in an ill-conditioned system of Eq. (3.2), and when the distance between the nearest IB points is much larger than the width of the regularization defined by the discrete delta functions Eq. (3.3), the immersed surface becomes permeable, which in turn leads to inaccurate solutions.

For rigid immersed-body motion, the solutions can also be facilitated by writing the above equations in an accelerating reference frame (moving with the body), but with a change of dependent variable to the velocity in the inertial reference frame. We refer the readers to (Liska and Colonius, 2017) for more details for this variation.

3.2.2 LGF, integrating factor and commutative properties

The LGF technique and the analytical integrating factor (IF) technique for a uniform unbounded grid of single resolution are essentially the same as introduced in Chapter 2, section 2.2.3. The LGF technique solves the discrete Poisson problem of the form

$$L_Q \phi = f, \quad \lim_{n \rightarrow \infty} \phi(\mathbf{n}) = 0, \quad (3.6)$$

$$\implies \phi = L_Q^{-1} f = G_Q * f, \quad (3.7)$$

where $f, s \in \mathbb{R}^Q$, G_Q is the LGF and $*$ denotes the discrete convolution, and an integrating factor (IF) is defined as the solution operator to the discrete heat equation on an unbounded uniform Cartesian grid

$$\frac{df}{dt} = \kappa L_Q f, \quad f(\mathbf{n}, \tau) = f_\tau(\mathbf{n}), \quad (3.8)$$

$$\implies f(\mathbf{n}, t) = \left[H_Q \left(\frac{\kappa(t - \tau)}{(\Delta x)^2} \right) f_\tau \right] (\mathbf{n}, t), \quad t \geq \tau, \quad (3.9)$$

where $f \in \mathbb{R}^Q$, $\kappa > 0$ is a constant, $f_\tau(\mathbf{n})$ is a known source field, and H^Q is the integrating factor operator. LGF and IF operators share the following mimetic and commutative properties with the differential operators:

$$D = -G^\dagger, \quad \bar{C} = C^\dagger, \quad L_C = -G^\dagger G, \quad (3.10)$$

$$H_{\mathcal{F}} G = G H_C, \quad D H_{\mathcal{F}} = H_C D, \quad L_{\mathcal{F}} G = G L_C, \quad D L_{\mathcal{F}} = L_C D \quad (3.11)$$

3.2.3 Time integration with integrating factor technique and half-explicit Runge-Kutta scheme

The DLM method treat the immersed boundary points along the surface as Lagrange multipliers so the no-slip boundary condition is maintained. The immersed boundary Eq. (3.2c), together with the divergence free condition Eq. (3.2b), add an algebraic constraint to the differential equation. To integrate the resulting DAE system of index 2 (Eq. (3.2)) in time, we apply a half-explicit Runge-Kutta (HERK) scheme.

Similarly as in Chapter 2 section 2.2.4, we begin by applying the IF operator $H_{\mathcal{F}}$ to Eq. (3.2) to yield

$$\frac{dv}{dt} + H_{\mathcal{F}} N \left(H_{\mathcal{F}}^{-1} v, t \right) = -Gp - H_{\mathcal{F}} E^\dagger(t) \hat{f}, \quad (3.12a)$$

$$Dv = 0, \quad (3.12b)$$

$$E(t) H_{\mathcal{F}}^{-1} v = u_\Gamma, \quad (3.12c)$$

where $\mathbf{v} = \mathbf{H}_{\mathcal{F}}\mathbf{u}$. The commutative properties Eq. (3.11) have already been used to simplify the above expressions.

Eqs. (3.12) are integrated using the same HERK scheme as in section 2.2.4. We refer to section 2.2.4 and Liska and Colonius, 2017 for more details. The main difference is that, due to the immersed boundary constraint, one will have an extended system of equations at each stage of every timestep of the following form:

$$\begin{bmatrix} \left(\mathbf{H}_{\mathcal{F}}^i\right)^{-1} & \mathbf{G} & \mathbf{E}^\dagger \\ \mathbf{G}^\dagger & \mathbf{0} & \mathbf{0} \\ \mathbf{E} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^i \\ \mathbf{p}^i \\ \mathbf{f}^i \end{bmatrix} = \begin{bmatrix} \mathbf{r}^i \\ \mathbf{0} \\ \mathbf{u}^i \end{bmatrix}, \quad (3.13)$$

where i refers to the i -th stage of each timestep, and \mathbf{r}^i is a known right-hand side. Eq. 3.13 then can be solved through block LU decomposition. Again, using the mimetic properties and the commutativities for simplification, Eq. (3.13) can be solved in the following steps:

$$\mathbf{L}_C \mathbf{p}^* = \mathbf{D} \mathbf{r}^i \quad (3.14a)$$

$$\mathbf{S}^i \mathbf{f}^i = \mathbf{E} \mathbf{H}_{\mathcal{F}}^i [\mathbf{r}^i - \mathbf{G} \mathbf{p}^*] - \mathbf{u}_c^i \quad (3.14b)$$

$$\mathbf{p}^i = \mathbf{p}^* - \mathbf{L}_C^{-1} \mathbf{D} \mathbf{E}^\dagger \mathbf{f}^i \quad (3.14c)$$

$$\mathbf{u}^i = \mathbf{H}_{\mathcal{F}}^i [\mathbf{r}^i - \mathbf{G} \mathbf{p}^i - \mathbf{E}^\dagger \mathbf{f}^i], \quad (3.14d)$$

where \mathbf{S} is related to the Schur complement

$$\mathbf{S}^i = \mathbf{E} \mathbf{H}^i \left(\mathbf{I}_{\mathcal{F}} - \mathbf{G} \mathbf{L}_C^{-1} \mathbf{D} \right) \mathbf{E}^\dagger. \quad (3.14e)$$

It is worth mentioning that Eq. (3.14) is in the form of a fractional step method but it is an exact block LU decomposition and thus there is no splitting error associated.

3.3 IB on the AMR grid

In order to implement the IB in the IB-LGF-HERK algorithm, we use the differential operators for the composite grid, $\overline{\mathbb{R}^Q}$, defined in section 2.3.3 to formulate Eq. (3.12-3.14). Likewise, we define the IB interpolation operator \mathbf{E} for the composite grid as the tensor product

$$\mathbf{E} = \otimes_{k=0}^{N_l} \mathbf{E}_k, \quad (3.15)$$

where \mathbf{E}_k is the corresponding operators for grid level k . This definition allows a more flexible treatment of the IB surface discretization. However, in the current

implementation, we assume that all IB points and their associated regions, denoted as $\Omega^{\mathcal{B}}$, are embedded in the same (finest) grid level Ω^{N_l} . i.e., $\Omega^{\mathcal{B}} \subset \Omega^{N_l}$. The definition of the associated region $\Omega^{\mathcal{B}}$ will be made clear later, and as will be shown, this assumption obviates the need to solve the forcing equation at each grid level since the forces on the finest level can be used to approximate the contribution to all other levels through the P-operators.

Again, we seek to approximate the right-hand side of Eq. (3.14) for the composite grid using the information from the AMR grid. Notice that the IB formula Eq. (3.12-3.14) are only different from Eq. (2.4-2.14) introduced in section 2.3.6 by Eq. (3.14b) and (3.14c), where the former solves for the IB forces and the later adds the extra contribution to the pressure field. In this section we may therefore limit the discussion to how those two equations are handled under the proposed framework.

The process of evaluating Eq. (3.14b, 3.14c) for the composite grid is broken down as:

1. The intermediate pressure p^* and its gradient Gp^* are evaluated in the same fashion as in Eq. (2.51), where the AMR-LGF algorithm is applied, i.e.,

$$\widehat{Gp^*} = \Gamma Gp^* \quad (3.16)$$

$$\approx [\Gamma GL_C^{-1} P^C] \widehat{Dr^i}. \quad (3.17)$$

2. The right-hand side of Eq. (3.14b) is approximated similarly to Eq. (2.52). The gradient of intermediate pressure field Gp^* is used as the source term in the AMR-FIF algorithm, which is then interpolated back to the IB points through operator E,

$$EH_{\mathcal{F}}^i [r^i - Gp^*] \approx E \left[\Gamma H_{\mathcal{F}}^i P^{\mathcal{F}} \right] \left(\widehat{r^i} - \widehat{Gp^*} \right), \quad (3.18)$$

3. Eq. (3.14b) in theory can be solved on every level using the linear operator $S^i = EH^i \left(I_{\mathcal{F}} - GL_C^{-1} D \right) E^\dagger$ defined for the composite grid. However, in the current implementation, the solutions are simplified under the assumption that all IB points and the associated regions $\Omega^{\mathcal{B}}$ are embedded in the same (finest) AMR level only, and thus Eq. (3.14b) needs only be solved once,

$$S^{i\mathcal{F}} \approx E \left[\Gamma H_{\mathcal{F}}^i P^{\mathcal{F}} \right] \left(\widehat{r^i} - \widehat{Gp^*} \right) - u_c^i. \quad (3.19)$$

The IB associated region $\Omega^{\mathcal{B}}$ is defined as the minimum support to evaluate S^i for all i , based on the distribution of the IB points, and the associated width

of the IB regularization (discrete delta function), and the minimum region required for evaluation of the operator S^i .

4. Having assumed all IB points are embedded in the finest grid with $\Omega^B \subset \Omega^{N_i}$, the forces on the finest level from Eq. (3.19) are coarsened to approximate the contribution to the pressure on the whole AMR grid.

$$\widehat{p}^i = \widehat{p}^* - [\Gamma G L_C^{-1} P^C] \widehat{DE}^\dagger f^i. \quad (3.20)$$

5. After solving the pressure gradient \widehat{Gp}^i , by Eq. (2.14), the updated velocity field at stage i after being restricted back to the AMR grid can be expressed as

$$\widehat{u}^i = [\Gamma H_{\mathcal{F}}^i P^{\mathcal{F}}] \left(\widehat{r}^i - \widehat{Gp}^i \right) + \widetilde{\epsilon}_u, \quad \widetilde{\epsilon}_u = \Gamma H_{\mathcal{F}}^i \epsilon_u. \quad (3.21)$$

Here the FIF-AMR algorithm given by Eq. (2.46) is applied.

3.4 Adaptivity

In this section we discuss factors associated with the IB that impact the adaptivity criteria previously discussed in section 2.4. As before, the computational domain only needs to encompass the vortical flow regions to yield an exact solution to the equations on an unbounded domain (as it is associated with the source term in the pressure Poisson equation). Thus the vorticity threshold for the outer domain truncation is left unaltered.

Regarding the mesh refinement, the AMR strategy must be changed owing to the restriction of the IB points to the finest mesh level. A diagram of the setup is shown in Fig. 3.1, where static mesh refinement (SMR) is applied to the near-surface area Ω^B defined by Eq. (3.19), and AMR is applied to the flow away from the immersed body. For AMR, both the spatial adaptivity and the refinement adaptivity can be applied based on previously defined parameters ϵ^* and α (see section 2.4).

The adaptivity criteria developed in the last chapter are appropriate for DNS. In future work, these adaptivity criteria should be revisited when LES is invoked, both as a sub-grid-scale stress model and as a near-surface wall model.

3.5 Fast linear solver for the IB formulation (Eq. (3.14b))

While Eq. (3.12-3.14) can be mostly solved similar to Eq. (2.4-2.14) using the fast AMR-LGF method introduced in section 2.3.6, a separate conjugate gradient (CG) method is applied to solve the IB formulation (Eq. (3.14b)). In this section the AMR-LGF-based CG solver is discussed in detail.

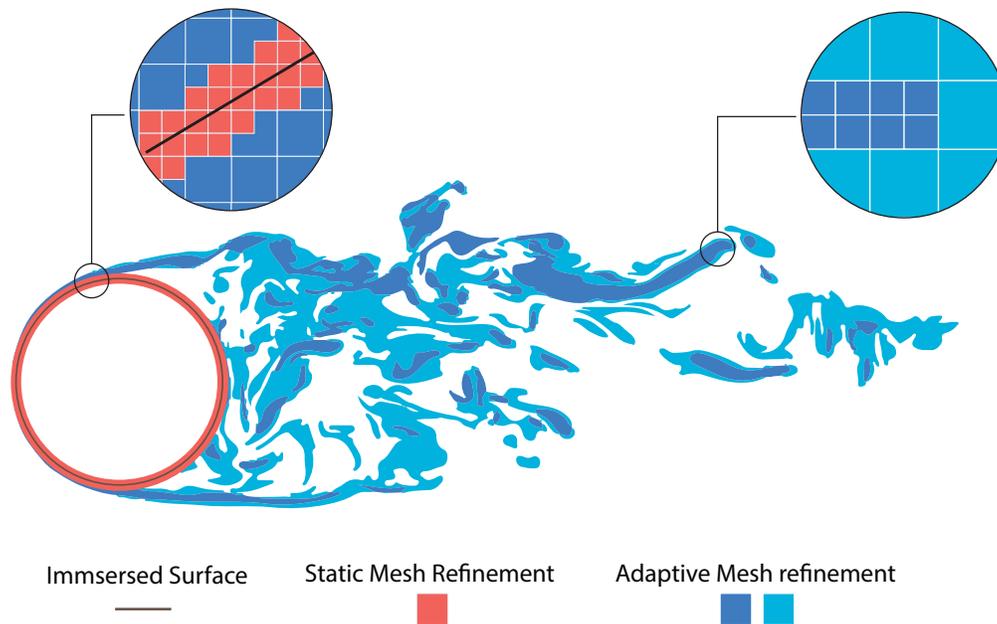


Figure 3.1: Diagram for the static mesh refinement for region near the immersed surface and adaptive mesh refinement with different levels for areas away from the bluff body.

A large number of IB points are often required to discretize the immersed surface, and this in turn necessitates a linear solver with good performance and parallel efficiency.

The development of the linear solver is simplified by the positive definiteness of the system of equations Eq. (3.13) and its associated Schur complement, Eq. (3.14b). This permits efficient direct solvers based on Cholesky decomposition, and efficient iterative solvers based on the CG algorithm. For rigid immersed body motions, the Cholesky decomposition has the advantage that the decomposition need only be computed once, since the same system is solved repeatedly. However, this requires operations and memory usage that scale as $\mathcal{O}(N_{\mathcal{B}}^3)$, where $N_{\mathcal{B}}$ is the number of IB points. On the other hand, the iterative CG solver only requires a memory usage of $\mathcal{O}(N_{\mathcal{B}})$, which allows the IB method to be applied to much larger systems.

For laminar boundary layers near the IB, the boundary layer thickness scales as $\text{Re}^{-1/2}$, which implies that the total number of IB points would scale with Re^1 . As for turbulent boundary layers, Choi and Moin (2012) estimated that for a flat plate at $\text{Re} > 10^6$, the number of grid points required for the near-surface region would scale as $\text{Re}^{13/7}$ and Re^1 for wall-resolving LES and wall-modeled LES respectively. In their example of a delta wing, Choi and Moin (2012) showed that for wall-modeled LES, $\mathcal{O}(10^7 - 10^{10})$ number of grid points are needed at Reynolds numbers from

10^6 to 10^9 . For the current IB method, it would require approximately $O(10^5 - 10^7)$ number of IB points. In the example of the DNS of flow around a sphere at $Re = 10,000$ to be discussed in the next section, a total number of 3.5×10^5 number of IB points are involved in the discretization of the surface, which makes the application of direct solvers prohibitive even in a parallel fashion.

Based on these considerations, we implemented a CG solver for Eq. (3.14b). The performance of the iterative CG solver largely depends on the speed of evaluating the left-hand side operator S^i , which involves nested evaluations of L_C^{-1} and $H_{\mathcal{F}}$ via the FLGF method. As all IB points are embedded in the finest AMR grid level that can be snugly wrapped around the IB surface, the associated source and target points for the FLGF are limited to a region that scales as the surface area of the IB rather than the volume, which makes the nested FLGF CG solver for the IB much faster than the FLGF that is associated with the volumetric sources. Two other factors also affect the performance of the CG solver. The first is the initial guess of the solution. For the current time-marching scheme, solutions from previous timesteps provide a good starting point. The second factor is the number of iterations involved in the CG solver, which depends on both the condition number of operator S^i and the numerical tolerance. For the former, we only see a slow increase of the condition number with the number of IB points as reported by Liska and Colonius (2017). As for the later, due to the noisy nature of the solutions to Eq. (3.14b) (Goza et al., 2016), it is often preferable to use a relatively less stringent error tolerance.

The performance of the CG solver can still be improved: For example, the CG algorithm can often benefit from preconditioning, which is not applied in the current scope; So far we only require the smearing operator R to be the adjoint of the interpolation E without specifying any exact form of the regularization defined by the delta functions, but in theory every IB point can use different regularization, and the those choices would affect the operator S^i and then affect the performance of the CG algorithm. We retain those optimization techniques to future work.

3.6 Flow around a sphere

In this section we consider the flow around a sphere at $Re = U_0 D / \nu = 100, 3700$ and $10,000$, where D is the diameter of the sphere, U_0 is the free stream velocity and ν denotes the kinematic viscosity. The low Reynolds number case at $Re = 100$ is used to verify the AMR-IB-LGF-HERK algorithm with a convergence study, and the higher Reynolds number cases are used to illustrate the capabilities and

computational savings of the proposed AMR scheme.

The IB method requires a set of discrete points that are approximately evenly distributed on the surface: having IB points too close to each other will result in Eq. 3.14b to be ill-conditioned, and when the distance between the nearest IB points is larger than the width of the regularization defined by the discrete delta function, the immersed surface becomes permeable. The discrete delta function applied in this work is adopted from Yang et al. (2009), which has a support of 3 grid points in each direction. To uniformly locate IB points on the surface of a sphere we use the Fibonacci lattice approach developed by Swinbank and James Purser (2006), with the ratio between the averaged spacing among nearest IB points $\Delta s_{\mathcal{B}}$ and the cell width of the underlying Eulerian grid Δx kept constant at $s/\Delta x = 1.5$. This approach has the benefits that the surface points are areally uniform and has approximately isotropic resolution. An example of using 500 points to discretize the surface of a sphere is shown in Fig. 3.2, whereas up to 3×10^5 points are applied in the simulations to be discussed. For all cases considered, flow is initialized starting from rest and increasing the freestream velocity $U_{\infty}(t)$ until the Reynolds number reaches desired value at $tU_{\infty}/D = 1/2$. i.e.,

$$U_{\infty}(t) = \begin{cases} \frac{2U_0^2}{D}t & \text{for } 0 \leq \frac{tU}{D} \leq \frac{1}{2} \\ U_0 & \text{for } \frac{1}{2} < \frac{tU}{D} \end{cases} . \quad (3.22)$$

For all cases to be discussed, we keep the refinement criterion $\alpha = 0.25, \beta = 0.75$ and the base level adaptivity threshold $\epsilon^* = 10^{-5}$ defined in section 2.4. The choices of those values were validated in section 2.7. Note that as compared to the IB-LGF method for a uniform grid of single resolution, the AMR-IB-LGF method allows a smaller base level truncation threshold since the application of AMR allows a more efficient coverage of larger spatial domain.

3.6.1 Verification

We begin by performing a simulation of the flow around a sphere at $\text{Re} = 100$. At this Reynolds number the flow approaches a steady state and thus serves as a verification for the convergence of the velocity. Different from the validation study discussed in section 2.6, where the convergence was conducted by varying the grid resolution with the topology of the AMR grid fixed, here we choose three base level resolution, and keep refining the grid with proposed criteria to reach the same finest resolution. As the smallest unit of the refinement is a computational block of a fixed number of cells, different base resolution with the same levels of refinement will

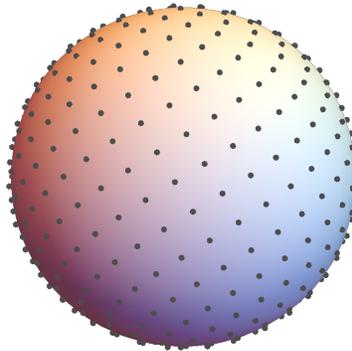


Figure 3.2: Example of using Fibonacci lattice to distribute IB points on the surface of a sphere. 500 points are shown in this case, whereas up to 3×10^5 points are applied in the simulations to be discussed.

result in different grid topology, and thus it also serves as a test of the aforementioned refinement criteria.

More specifically we choose three base level resolution to be $\Delta x_{\text{base}}/\Delta x_0 = 2^{-2}, \dots, 2^0$ with $x_0/D = 0.125$ being a constant. For each base resolution we refine the AMR grid to the same finest resolution $\Delta x_{\text{fine}}/\Delta x_0 = 2^{-3}$ and all cases are simulated for 256 timesteps with the largest CFL number smaller than 0.5. Results are compared to a reference solution at $\Delta x_{\text{ref}} = 0.5\Delta x_{\text{fine}}$ with the AMR turned off. The ℓ_∞ and ℓ_2 convergence of the velocity field is shown in Fig. 3.3, where black, blue, and red curves correspond to the chosen base resolution, and each point on a given curve represents a different level of refinement. We see a first-order convergence in the ℓ_∞ norm and a slightly faster convergence in the ℓ_2 norm. This result is expected since the regularization of the singular forces in the IB method results in a smooth velocity field across the immersed surface, whereas the exact velocity field yields a discontinuous first derivative. A posterior examination shows that the errors are mostly concentrated near the boundary of the geometry and this type of behavior was reported in other studies of IB methods such as Goza et al., 2016.

3.6.2 Flow around spheres at high Reynolds numbers

We further demonstrate the algorithm with a DNS of the flow around a sphere at Reynolds numbers $\text{Re} = 3700$ and a preliminary DNS at $\text{Re} = 10,000$. The simulations are conducted by first running a coarser simulation up to $t^* = tU_0/D \approx 20$ and then gradually increasing the levels of refinement in a manner that is to be described in detail.

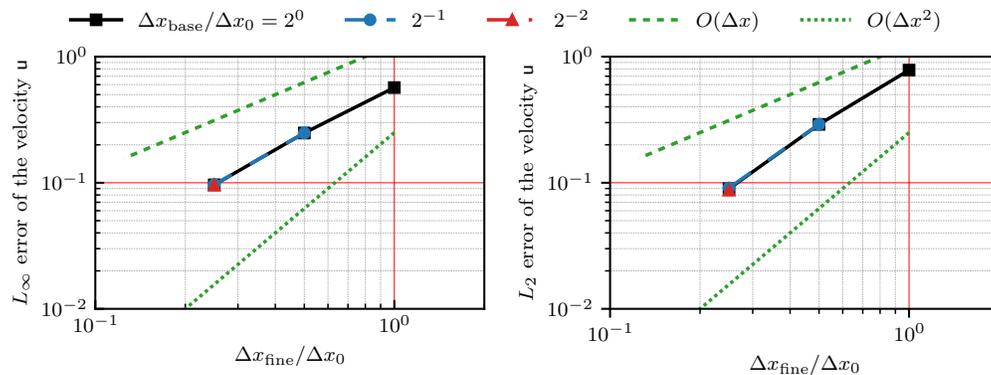


Figure 3.3: ℓ_∞ and ℓ_2 convergence of the velocity field for flow around a sphere at $Re = 100$. The black, blue and red curves correspond to different but fixed base resolution. Each curve presents different levels of refinement with the adaptation criteria discussed in the text.

For the simulation at $Re = 3700$, the simulation is performed from $t^* = 20$ to 30 with 2 levels of refinement. The finest resolution for the near-surface region is at $\Delta x_2/D = 4.3 \times 10^{-3}$ at 2 levels of refinement. Liska and Colonius (2017) showed that with identical grid resolution the boundary layer is resolved, and Yun, Kim, and Choi (2006) performed an IB/LES study of the same case with a minimum near-wall resolution of $9 \times 10^{-3}D$. For flow away from the surface an adaptive mesh up to 1 level of refinement is used which is of resolution $\Delta x_1/D = 8.6 \times 10^{-3}$. Rodriguez, Borell, et al. (2011) previously showed that the minimum Kolmogorov length scale present in the flow is $\eta/D = 1.34 \times 10^{-2}$ for the same Reynolds number. Since in the current DNS one has $\Delta x_1/\eta < 0.6$, the turbulence is expected to be fully resolved.

For the simulation at $Re = 10,000$, the $Re = 3700$ case is further refined after $t^* \approx 30$ to 35 with an additional level of refinement to both the near-surface region (SMR) and the wake (AMR), up to resolution $\Delta x_3/D = 2.15 \times 10^{-3}$ and $\Delta x_2/D = 4.3 \times 10^{-3}$ respectively. We expect the flow near the surface to be well-resolved based on the $Re^{-1/2}$ scaling of the thickness of the laminar boundary layer. As for the wake, the ratio between the Kolmogorov scale and the grid resolution is estimated to peak around $\Delta x_2/\eta \approx 0.7$ estimated from the minimum length scale present at $Re = 3700$. Rodriguez, Lehmkuhl, et al. (2013) performed a DNS of $Re = 10,000$ using unstructured meshes where the resolution is validated through a comparison with another simulation at coarser grid resolution as well as studies by Moin and Mahesh (1998) and Pope (2001). They showed that the same ratio peaks at $\eta/\Delta x = 2.5$ with a mean value $\bar{\eta}/\Delta x = 0.91$. The current DNS is at a much lower peak and mean ratio, and thus we expect the flow in the wake region to be

full-resolved. With this proposed grid setup near the body, a total of 350×10^3 IB points are solved in the discretization of the surface which is a magnitude higher than the highest number of IB points reported in Liska and Colonius (2017).

The vorticity and grid topology are visualized for $Re = 3700$ at $t^* \approx 30$ in Fig. 3.4. Rodriguez, Borell, et al. (2011) showed that the smallest turbulence scale occurs at $x/D < 3$ and one can see that under the chosen adaptivity criteria, the algorithm successfully detects the intensive turbulence region and applies grid refinement to the corresponding space. At the same time the algorithm automatically uses a coarser grid for the remaining areas defined by the truncation threshold $\epsilon^* = 10^{-5}$ so computational saving is achieved.

A closer comparison between the $Re = 3700$ and $Re = 10,000$ vorticity fields is given in Fig. 3.5. For both cases we observe complex vortex structures behind the sphere and large-scale vortex shedding downstream. The flow field at $Re = 10,000$ clearly yields a richer range of scales, and we observe azimuthal instabilities as the vortex sheet develops wrinkles on the suction side of the sphere, whereas at $Re = 3700$ it remains as a smooth surface. Secondly, we see the Kelvin–Helmholtz (KH) type of instabilities developing at the outer rim of the wake just downstream of the sphere. These waves grow rapidly and facilitate a more rapid transition to turbulence in the wake. This KH type of vortex roll-up for the sphere at similar Reynolds numbers was previously shown in experimental studies such as Werle (1981) and Jang and Lee (2007), and our results are similar to the $Re = 10,000$ flow observed by Rodriguez, Lehmkuhl, et al. (2013), where they showed that the transition occurs at about $x/D = 1 \sim 1.2$ from the separation point, in contrast to $x/D = 1.8 \sim 2.6$ in the $Re = 3700$ case. In addition, the flow visualization indicates that there are more elongated and finer vortex structures in the wake region at $Re = 10,000$ compared to the $Re = 3700$ case.

We report the mean drag coefficient for $Re = 3700$ and preliminary results for $Re = 10,000$ in Table 3.1. For $Re = 3700$, we see a significant agreement with Rodriguez, Borell, et al. (2011) and Liska and Colonius (2017) where the difference is approximately 0.3% with the former and 0.8% with the later. For $Re = 10,000$, there is a discrepancy about 5% with the experimental study by Achenbach (1972) or the DNS by Rodriguez, Lehmkuhl, et al. (2013), and about 3.5% with the detached eddy simulation (DES) study by Constantinescu and Squires (2003) or the LES study by Yun, Kim, and Choi (2006). Note that a longer simulation for the $Re = 10,000$ case might be needed to have a more accurate mean drag coefficient at this Reynolds

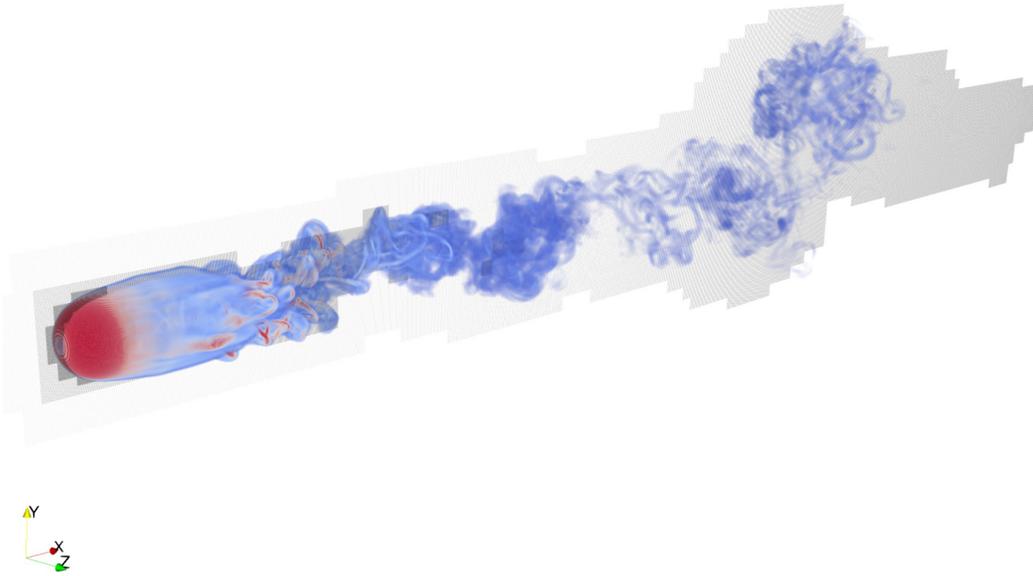


Figure 3.4: Vorticity of flow around a sphere at $Re = 3700$ at $tU_0/D \sim 30$. Vorticity contour level $\omega D/U_0 = [1, 30]$. Grid topology over the center plane is plotted for 0-2 levels of refinement from light to dark gray.

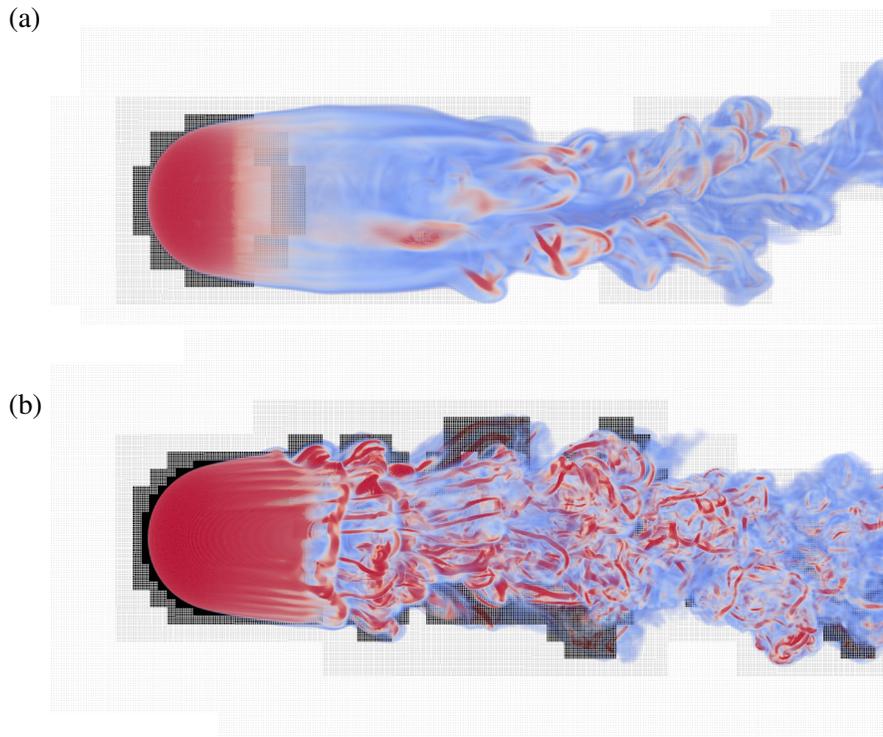


Figure 3.5: Comparison of the vorticity field near the sphere at $Re = 3700$ and $Re = 10,000$ for $x/D < 4$. Grid topology over the center plane is shown from light to dark gray.

number.

Lastly we report the the computational saving from AMR-IB-LGF scheme in Table 3.2, where the number of computational cells used in the current scheme is compared to both a spatial adaptive domain with uniform fine grid, and a rectangular bounding domain with uniform fine grid. For $Re = 3700$, the AMR-IB-LGF scheme is saving 98% of the computational cells compared to the former, and 99.2% compared to the later. For $Re = 10,000$, we see the AMR-IB-LGF scheme saves 99.5% and 99.87% of computational cells for the two cases respectively. We expect even greater computational savings when flows at higher Reynolds number are simulated with a wider range of scale separation.

		Re	$\overline{C_D}$
Present	DNS	3700	0.392
LC17	DNS	3700	0.389
RB11	DNS	3700	0.393
YK06	LES	3700	0.355
Present	DNS	10,000	0.38
RB13	DNS	10,000	0.403
A72	Exp.	10,000	0.40
YK06	LES	10,000	0.393
CS03	DES	10,000	0.393

Table 3.1: Mean flow drag coefficient of a sphere at Reynolds numbers 3700 and 10,000. Results from LC17 - Liska and Colonius (2017), RB11 - Rodriguez, Borell, et al. (2011), RB13 - Rodriguez, Lehmkuhl, et al. (2013), YK06 - Yun, Kim, and Choi (2006), A72 - Achenbach (1972), CS03 - Constantinescu and Squires (2003).

Name	AMR	Spatial adaptive domain	Bounding rectangle
Sphere - 3700	0.9×10^8	4.4×10^9 (98.0%)	1.2×10^{10} (99.2%)
Sphere - 10,000	1.9×10^8	3.8×10^{10} (99.5%)	1.5×10^{11} (99.87%)
Delta wing - 10,000	1.1×10^8	2.2×10^9 (95.0%)	3.7×10^9 (97.5%)

Table 3.2: Comparison of the number of computational cells used in the AMR-IB-LGF scheme, compared with a spatially adaptive computational domain with uniform fine grid, and a rectangular bounding domain with uniform fine grid, together with the percentage in the reduction of grid points.

3.7 Flow around a delta wing at an angle of attach of 20°

In this section we demonstrate the capability of the AMR-IB-LGF-HERK scheme in solving external flows around complex geometries at high Reynolds numbers. The

immersed geometry considered is an aspect ratio 1.74 delta wing designed by He, An, et al. (2019). The experimental setup, the delta wing, and the mesh used are shown in Fig. 3.6.

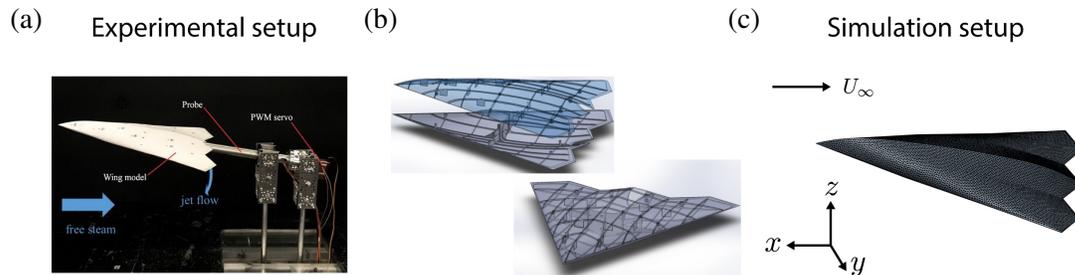


Figure 3.6: Delta wing setup (He, An, et al., 2019): (a) Experimental setup ; (b) CAD model; (c) immersed boundary mesh.

To generate the surface IB points for the delta wing we apply a meshing technique developed by Persson (2005). This technique utilizes a user-defined level set function of the desired geometry that maps a spatial point to a real number that is negative when the point is inside the geometry, positive when the point is outside the geometry and zero on the surface. This algorithm first generates a random unstructured mesh for the surface and iteratively improve the mesh towards an equal nodal distribution by solving for a force equilibrium in the element edges, and projecting the mesh nodes back to the surface of the geometry using the level set function (Persson and Strang, 2004). The level set function for the delta wing was generated by partitioning the total surface into a number of smooth parts and use b-splines for each part individually.

A DNS at an angle of attack $\alpha = 20^\circ$ is performed at $Re = 10,000$ based on the the constant freestream velocity, U_0 and the chord length, c . A maximum of 2 levels of refinement were utilized, with the finest grid resolution of $\Delta x_2 = 3.1 \times 10^{-3}$. The resolution is determined by a low Reynolds number study at $Re = 1000$ and the scaling of the laminar boundary layer thickness of $Re^{-1/2}$, which is also comparable to the sphere case discussed above. Similarly, we use a refinement adaptivity parameter $\alpha = 0.25$ and a base level truncation threshold $\epsilon = 10^{-5}$.

The flow evolution at $tU_0/c = 1.5, 4.0, 10.0$ is shown in Fig. 3.7. The lift and drag coefficients at are compared with the experimental study by He, Le Provost, and Williams (2018) at a higher $Re = 10^5$ in Table 3.3, where the discrepancies in the lift/drag coefficients may be due to the difference in the Reynolds numbers.

While much remains to be analyzed about this flow field, the present results serve to illustrate the flexibility of the IB approach along with the ability of the developed numerics to perform high fidelity simulations for complex geometries. Lastly, we report the computational saving from the AMR scheme compared both a spatially adaptive computational domain with uniform fine grid and a rectangular bounding domain with uniform fine grid in Table 3.2 where we see the current scheme is reducing the number of computational cells up to 97.5%.

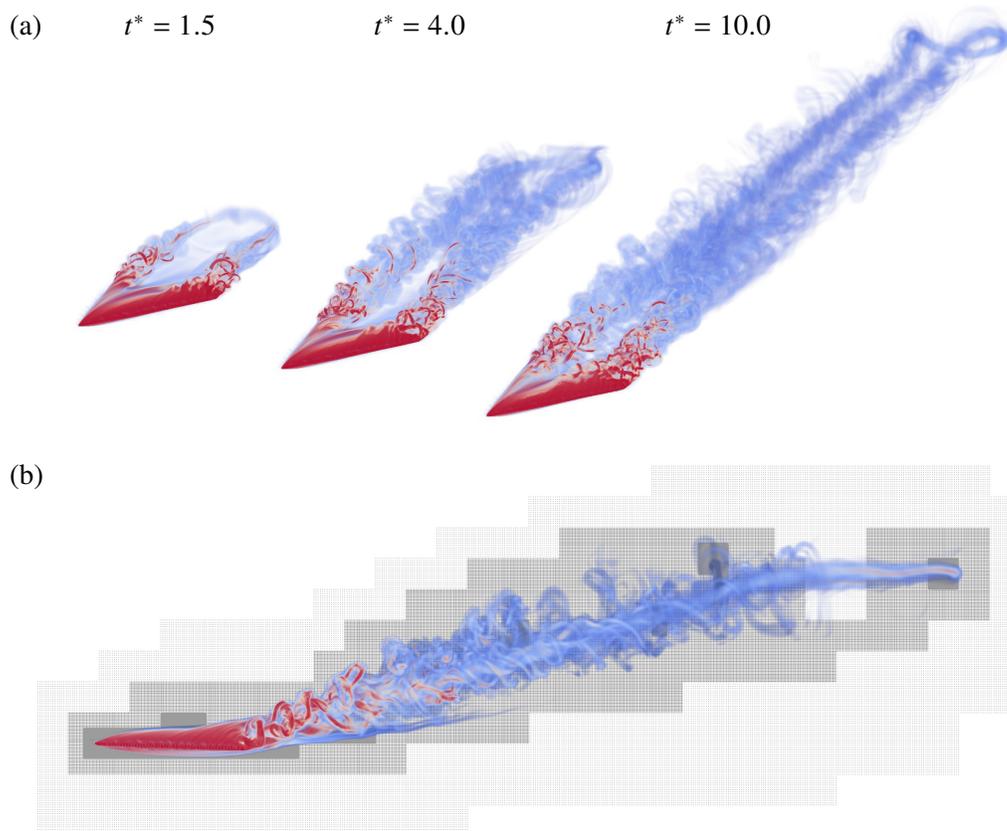


Figure 3.7: (a) Vorticity visualization at $t^* = t U_0/D = 1.5, 4.0, 10.0$ for the delta wing at the angle of attack $\alpha = 20^\circ$. (b) Mesh topology at $t^* = 4.0$. Grid level from 0 to 2 is shown from light to dark gray.

	Re	C_L	C_D	C_L/C_D
Experiment	10^5	0.95	0.34	2.8
DNS	10^4	0.64	0.22	2.3

Table 3.3: Lift and drag coefficients from simulation at $\text{Re} = 10^4$ and experiment at $\text{Re} = 10^5$ by He, Le Provost, and Williams (2018).

3.8 Concluding remarks

In this chapter we extended the AMR-LGF-HERK scheme developed in Chapter 2 to include complex immersed surfaces via the DLM IB method. We restricted the IB to be embedded in the finest mesh level and developed a fast, iterative solver based on the CG method for solving the system of equations for the surface forces. We verified the solutions for steady flow around a sphere at $Re = 100$, and demonstrated the capabilities of the algorithm through DNS of flow around a sphere at $Re = 3700$ and $10,000$, and a delta wing at $Re = 10,000$. We showed that this algorithm is capable of performing state-of-the-art DNS in an efficient manner. For both cases, the adaptive grids reduced the total number of cells by 99.87% and 97.5% respectively compared to a rectangular bounding domain with uniform resolution.

LARGE-EDDY SIMULATION

4.1 Introduction

In this chapter we implement and validate the LES in the LGF framework. In past work, LES validations have involved canonical flows in wall-bounded or periodic domains. While flows over immersed surfaces have been investigated with LES, they are not optimal in terms of validation of the SGS model, since they bring in additional complications such as thin boundary layers and the necessity of wall modeling to reach high Reynolds numbers. In this chapter, we manufacture a turbulent flow evolving in free space and report on both DNS and LES of this flow. Apart from thoroughly validating the LES implementation, the flow we constructed—nearly homogeneous turbulence evolving in a compact, spherical region, turned out to be theoretically interesting for a number of reasons which are also discussed in the next chapter.

In free space (without immersed boundaries) and without artificial forcing, there are no mechanisms to sustain turbulence, and it will decay in time. Of possible initial flow fields, two of relevance here are a collection of one or more vortex rings and a random initial condition. Initial conditions comprising a collection of vortex rings are readily created in both experiments and simulations. Recent experiments have studied the generation of turbulence through vortex-ring collisions (Matsuzawa et al., 2019) and have evaluated hypothesized mechanisms of turbulence self-sustenance (McKeown et al., 2018). The resulting turbulence occurs through a complex process of instabilities, vortex interaction, and reconnection (Lim and Nickels, 1992). Simulations are computationally intensive since distinct turbulent and laminar regions will occur, and the associated vortex-ring Reynolds numbers must be sufficiently high for transition to occur. In contrast, a random initial condition is computationally simple and turbulent Reynolds numbers can easily be reached (even in DNS), but a disadvantage is that there will be an initial transient period that, while governed by the Navier-Stokes equations, will not be associated with physical turbulence.

In this chapter, we manufacture an initial condition by first generating isotropic homogeneous turbulence (IHT) in a periodic domain, and initializing a free-space

cloud of turbulence by tiling the periodic IHT field in space and windowing it with an indicator function that falls to zero outside a spherical region of radius R , which is later varied compared to the initial scales in the IHT. Figure 4.1 shows the setup for the problem at hand. Initially only a spherical region in free space is filled with turbulence. The spherical region then starts to deform, evolve, and decay. This method utilizes fixed local cell size but is able to spatially adapt with the vortical areas by adding or removing blocks of computation cells (black contour lines in figure 4.1). All simulations here are conducted with a spatial adaptive threshold ϵ_{supp} equal to 10^{-5} defined in Liska and Colonius (2016). Because of the spatial adaptivity, the total number of computation cells varies through one simulation. The code is fully parallelized and the DNS simulation uses around a maximum of 2×10^9 computation cells running on over 1,500 cores.

This chapter is organized as follows. In section 4.2, the sub-grid stress model used in the LES is briefly reviewed. In section 4.3, the initial condition and the parameters for the simulations involved are introduced. In section 4.4, DNS and LES results are used to visualize the evolution of the turbulence field. In § 4.5, we use statistical measures to characterize the decay and show LES calculations agree well with the DNS in these measures. Lastly we conclude this chapter in section 4.6.

4.2 Stretched vortex sub-grid stress (SGS) model

The LES solutions we report rely on the stretched vortex SGS model (SVM) (Chung and Pullin, 2009b). The LES solves the filter-averaged Navier-Stokes equations

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \widetilde{\mathbf{u} \cdot \nabla \mathbf{u}} = -\nabla \tilde{P} + \nu \nabla^2 \tilde{\mathbf{u}} - \nabla \cdot \tilde{T}, \quad (4.1)$$

$$\nabla \cdot \tilde{\mathbf{u}} = 0, \quad (4.2)$$

where $\tilde{T} = \widetilde{\mathbf{u} \otimes \mathbf{u}} - \tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}$ is the SGS tensor. The model assumes that the subgrid motions for a single computational cell are dominated by small vortices in a direction \mathbf{e}^v that is aligned with the principle eigenvector of the resolved strain rate tensor. Then the SGS tensor is given by

$$\tilde{T}_{ij} = K \left(\delta_{ij} - e_i^v e_j^v \right), \quad (4.3)$$

$$K = \int_{k_c}^{\infty} E(k) dk = \mathcal{K}'_0 \Gamma[-1/3, \kappa_c^2] / 2, \quad (4.4)$$

where K is the subgrid kinetic energy, $k_c = \pi/\Delta_x = \pi/\Delta_y = \pi/\Delta_z$ with $\Delta_{(\cdot)}$ being the cell length, and Γ is the incomplete gamma function. The second equality

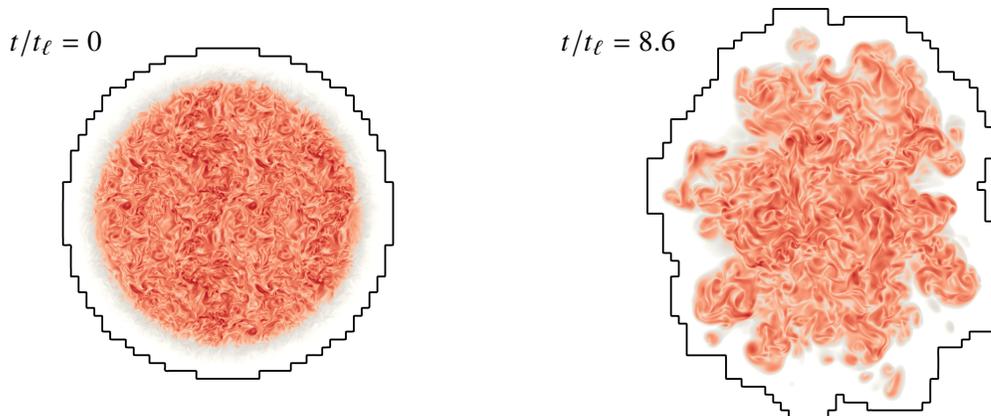


Figure 4.1: Vorticity magnitude in a cross-section through the center at $t/t_\ell = 0$ and $t/t_\ell = 8.6$, corresponding to the case DNS_0 from table 4.1. Black contour lines indicate the spatially adaptive computational domain. The smallest adaptivity unit is a block of 16^3 computational cells.

in equation (4.4) assumes that SGS vortices are of the stretched-spiral type with spectrum (Lundgren, 1982)

$$E(k) = \mathcal{K}_0 \epsilon^{2/3} k^{-5/3} \exp[-2k^2 \nu / (3|\bar{a}|)], \quad (4.5)$$

where ν is the fluid viscosity

$$\bar{a} = e_i^\nu e_j^\nu \tilde{S}_{ij}, \quad \mathcal{K}'_0 = \mathcal{K}_0 \epsilon^{2/3} \lambda_\nu^{2/3}, \quad \lambda_\nu = (2\nu/3|\bar{a}|)^{1/2}, \quad \kappa_c = k_c \lambda_\nu, \quad (4.6)$$

and \tilde{S}_{ij} is the resolved strain rate tensor. Finally the constant \mathcal{K}'_0 in equation (4.4) is determined by matching the resolved second-order velocity structure function with the prediction from the energy spectrum given by Eq. (4.5). Details regarding the efficient evaluation of the aforementioned SGS stress can be found in Voelkl, Pullin, and Chan (2000) and Chung and Pullin (2009b).

The SVM is structure based and not of the eddy-viscosity type. All model parameters are calculated dynamically using only local information from the resolved-scale field surrounding the grid cell or point where sub-grid stresses are calculated. The SVM keeps track of the actual fluid viscosity and also the subgrid kinetic energy, and will automatically become subdominant to real viscous stresses when the flow is locally resolved. It has proven robustness and has been successfully used for studies of decaying turbulence (Misra and Pullin, 1997), and wall-resolved LES of channel flow (Voelkl, Pullin, and Chan, 2000; Chung and Pullin, 2010), bluff-body flows (Cheng, Pullin, Samtaney, et al., 2017; Cheng, Pullin, and Samtaney, 2018) and Taylor-Couette flow (Cheng, Pullin, and Samtaney, 2020).

4.3 Problem setup

4.3.1 Initial condition

The initial condition is generated by spherically windowing a turbulence field from a separate IHT computation with periodic boundary conditions. This field is then tiled in all directions to fill the free space and the velocity field is multiplied by a smooth window function of the form

$$\Phi(r) = \frac{1}{2} \left[1 - \tanh \left(\frac{2(r - R)}{\sigma} \right) \right], \quad (4.7)$$

where R is the radius of the sphere and σ is the width of the transition, whose impact on the results will be assessed. The forced periodic IHT field is generated using a simple 3D pseudo-spectral code and we define the domain size to be B^3 . A low wavenumber forcing method is applied (Huang, 1994). The forcing is restricted to modes with wavenumbers $|\mathbf{k}| < 2.5$ and the magnitude of the forcing is chosen to keep the energy input rate constant, which would equal to the dissipation rate ϵ after the forced turbulence becomes stationary. To make sure all IHT flows are fully resolved, ϵ is determined such that $\eta k_{\max} \sim 1.5$, where $k_{\max} = N_s/2$ is the maximum wavenumber and $\eta = (\nu^3/\epsilon)^{1/4}$ is the Kolmogorov length scale with ν being the viscosity. We also confirmed the isotropy of the IHT field by verifying that $\frac{E_{ii}(k)}{E(k)} - \frac{1}{3} \approx 0$, $i = 1, 2, 3$.

Figure 4.1 visualizes the initial vorticity field in a cross-section through the center. This corresponds to the case DNS_0 defined in Table 4.1. Note that black contour lines in figure 4.1 are the spatially adaptive computational domain which encompasses the initial vorticity field as discussed in §2.4. For LES, the IHT field is spectrally filtered before tiling and windowing. More about initial conditions for LES is discussed in § 4.4.2 when results from LES are presented.

Once an IHT field and resolution are selected (which give an initial turbulence Reynolds number, Re_λ), two non-dimensional parameters characterize the initial condition: B/R and σ/R . Table 4.1 summarizes the parameters for all runs studied in this chapter and the next chapter. The simulation parameters used in the pseudo-spectral code to generate the IHT fields and their statistical characteristics are given in Table 4.2.

4.3.2 Resolution

For DNS_0, the IHT field used to generate the initial condition has $\text{Re}_\lambda = 122.4$ and uses a computational domain of 256^3 in the pseudo-spectral code with $\eta k_{\max} > 1.5$

Name	Re_λ	σ/R	B/R	Spectrum Type
DNS_0	122.4	0.10	1.0	2
LES_0	122.4	0.10	1.0	2
LES_IC2	122.4	0.10	1.0	4
LES_D1	122.4	0.05	1.0	2
LES_D2	122.4	0.20	1.0	2
LES_B1	122.4	0.10	0.5	2
LES_B2	122.4	0.10	2.0	2
LES_R1	76.9	0.10	1.0	2
LES_R2	45.0	0.10	1.0	2

Table 4.1: Simulation parameters. The spectrum type refers to the leading non-zero order in the low wavenumber limit which is discussed in detail in Chapter 5.

Run Name	Re_λ	ℓ/B	ηk_{\max}	Resolution
LES_R1	76.9	0.17	1.52	128^3
LES_R2	45.0	0.19	1.58	64^3
All others	122.4	0.16	1.53	256^3

Table 4.2: Summary of the simulation parameters used in the pseudo-spectral code and the resulting IHT fields. ℓ is the integral scale, η is the Kolmogorov length scale, and k_{\max} is the maximum wavenumber.

to ensure that it is fully resolved. The same resolution (same number of points used for every length scale B) is used in the LGF solver for the turbulence cloud. To guarantee this resolution is also sufficient for the finite volume solver, another DNS simulation of $3/2$ times the resolution is performed up to 1.3 initial large eddy turnover time. The difference in the total kinetic energy is about 0.23% and the maximum relative difference in the spectra for all wavenumber kR is about 1% which is shown in figure 4.2.

4.4 Qualitative evolution

4.4.1 DNS

First we perform DNS of the spherical cloud of turbulence corresponding to the case ‘DNS_0’ in table 4.1. The flow evolution is shown in figure 4.3a. Instantaneous vorticity magnitude iso-surfaces at $t/t_\ell = 0, 1.7, 4.0, 8.6, 17.5$ are given, where t_ℓ is the large eddy turnover time of the original IHT field. The iso-surface of the lowest vorticity magnitude represents the TNTI. This interface is sufficiently thin (Mathew and Basu, 2002) that using a lower minimum vorticity magnitude would not affect the boundary envelope noticeably. At $t/t_\ell = 0$ the turbulence

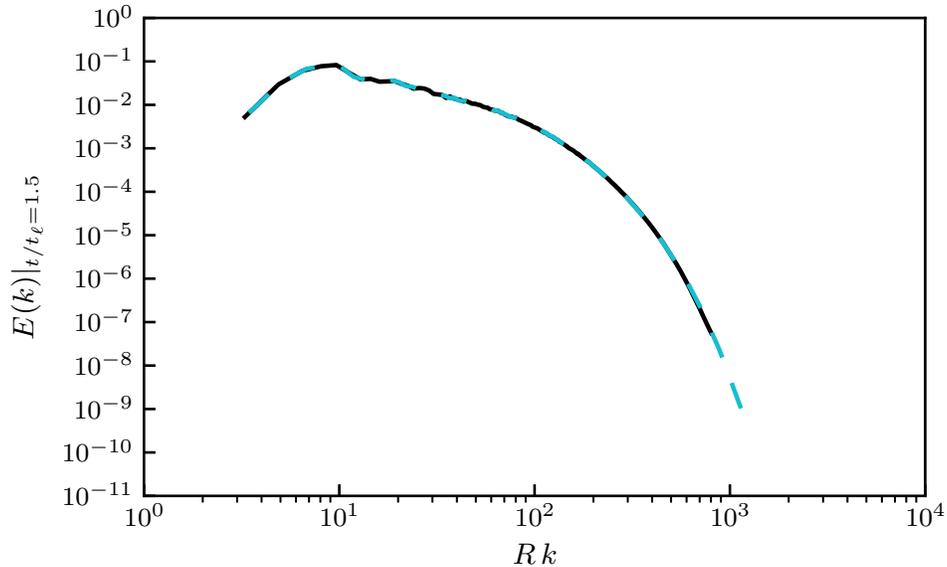


Figure 4.2: Energy spectrum of (1) DNS_0 (—) and (2) a DNS calculation at 3/2 times the resolution as that in the case DNS_0 (---), at $t/t_\ell = 1.3$.

is contained within a spherical region defined by the window function. As the turbulence evolves, the transition region is mixed with the turbulence inside and becomes gradually indistinguishable around $t/t_\ell \sim 1.5$. At $t/t_\ell \sim 4.0$ more fine features have developed near the boundary while the general spherical shape is still maintained. Around $t/t_\ell \sim 8.6$, small features start to merge and create protrusions. Meanwhile the general shape has also become more ellipsoidal. The DNS flow evolution is simulated up to $t/t_\ell = 17.5$. From $t/t_\ell = 8.6$ to 17.5 the cloud of turbulence becomes more irregular, and finer scales are less evident as the turbulence decays.

4.4.2 Comparison between DNS and LES

DNS is applied to study the more active early stage evolution of a turbulence cloud but it is computationally expensive to reach late times. To study the long-term behavior we turn to LES calculations of the same setup. To ensure LES calculations are able to accurately capture the abiding features of the flow, we qualitatively compare the evolution for DNS and LES of the same case.

The initial condition for the LES run is created in the following way: first, the same IHT field from case DNS_0 is spectrally filtered from 256^3 to 32^3 , keeping only $(1/8)^3$ of its original spectrum; second, the same recipe (tiling and spherical windowing) is used with the filtered turbulence field to create a spherical region of

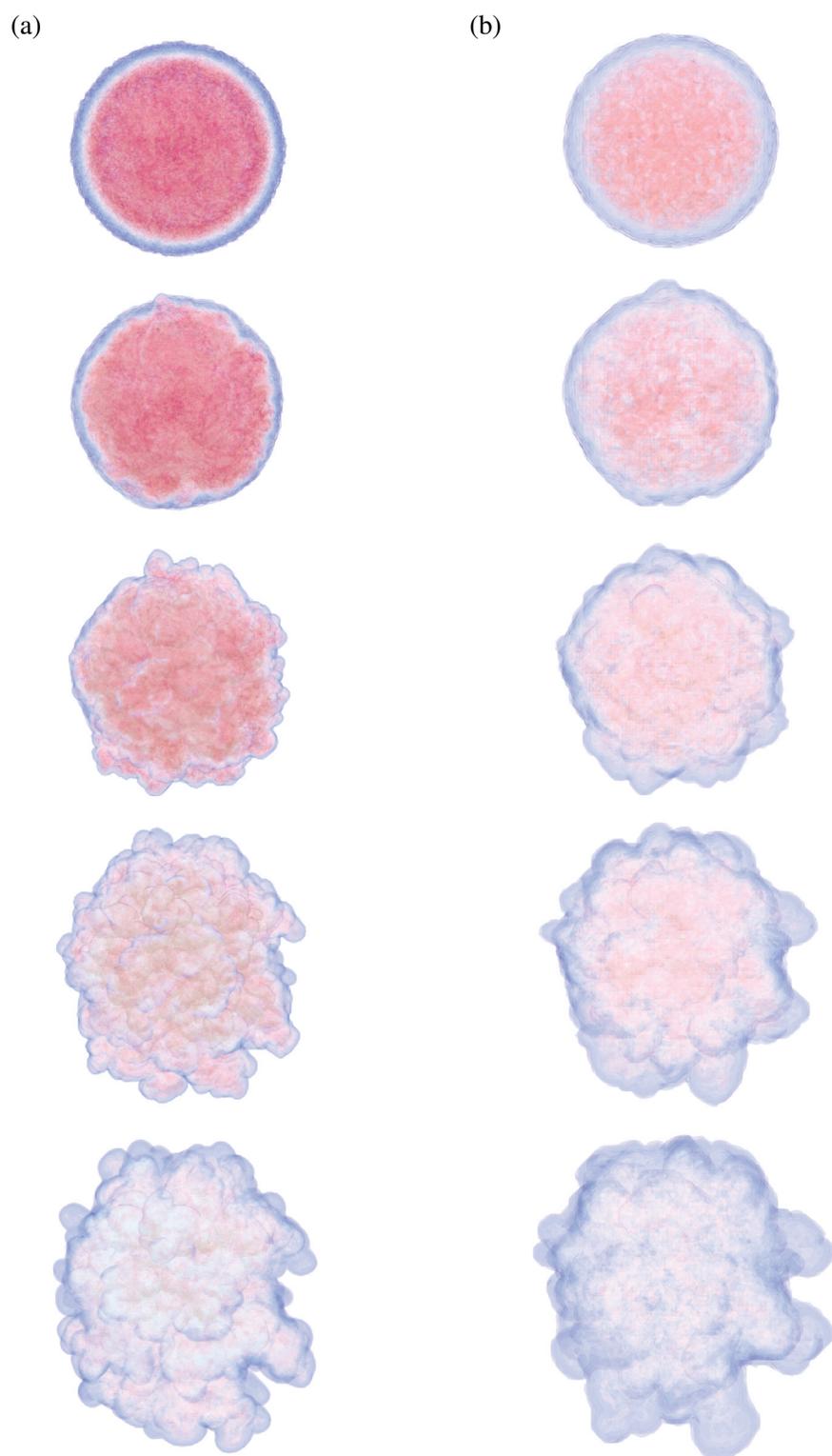


Figure 4.3: Vorticity magnitude iso-surface of (a) DNS_0 and (b) LES_0 at $t/t_\ell = 0, 1.72, 4.02, 8.60, 17.47$ from top to bottom, where t_ℓ is the large eddy turnover time of the initial IHT field.

under-resolved turbulence. This field is then given to the LGF finite-volume solver with the SGS model turned on. This simulation corresponds to the case ‘LES_0’ in table 4.1.

Figure 4.3 also compares DNS_0 (4.3a) and LES_0 (4.3b) at $t/t_\ell = 0, 1.72, 4.02, 8.60, 17.47$. To ensure that the difference in grid resolution between DNS and LES would not affect the visualization, all iso-surfaces are re-sampled to the same grid. The LES captures the general shape and most of the large-scale features such as the radius, the ellipticity, the sizes and locations of the protrusions. On the other hand, some small-scale features near the boundary are missed. We also noticed that the vorticity is less intense in the LES run (the crimson regions) especially towards the early stage. All of these differences are to be expected, as LES is designed to capture the statistical properties of the turbulence (and specifically their influence on the largest scales). Nevertheless, over the time range displayed in figure 4.3 there is little decorrelation of the large scales in DNS and LES originating from the same initial condition. To further quantify the comparison, in § 4.5, four statistical measures are introduced and applied to both cases DNS_0 and LES_0.

4.5 Quantitative evolution

4.5.1 Statistical measures for DNS and LES

In this section we quantify the initial evolution of the cloud of turbulence using statistical measures. DNS results are compared with LES during the initial decay period up to about $t/t_\ell \simeq 20$.

Firstly the kinetic energy $\mathcal{E}(t)$ decay is studied. Results from three simulations are compared in figure 4.4: (1) DNS_0; (2) LES_0 and (3) an under-resolved DNS (the same setup as LES_0 but with SGS model turned off). DNS_0 should be regarded as the most accurate case among all three and its value at $t = 0$ is used to normalize all results. For LES_0 we show both the kinetic energy resolved by the grid and a ‘total kinetic energy’ which is the sum of the resolved energy and the estimated subgrid energy predicted by the SGS model. The total kinetic energy in LES_0 compares well with DNS_0. The initial resolved energy in LES_0 is smaller than that in DNS_0 owing to the spectral filtering process discussed in § 4.3.1. On the other hand the under-resolved DNS shows evident energy pile-up due to the lack of the SGS model. After $t/t_\ell \simeq 8$, the resolution of case LES_0 is high enough to resolve all flow scales due to the decay and it is effectively ‘DNS’ after this point.

Secondly, the total energy spectrum, which will be discussed in detail in Chapter 5,

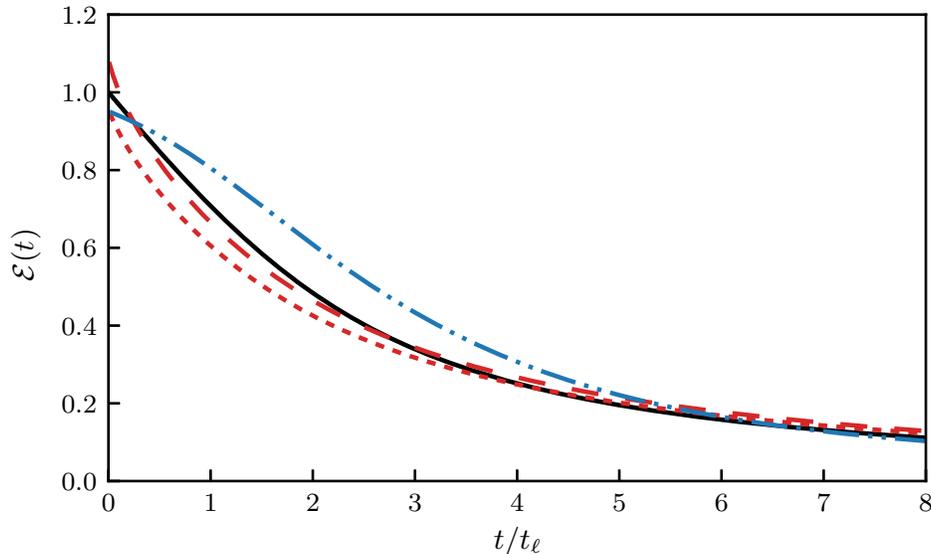


Figure 4.4: Decay of the kinetic energy $\mathcal{E}(t)$ for different simulations: DNS_0 (—); LES_0 resolved kinetic energy (.....); LES_0 total kinetic energy (---); an under-resolved DNS (-.-). The initial kinetic energy in DNS_0 is used to normalize all simulations.

§ 5.2 (Eq. (5.2)) is applied. Results for DNS_0 and LES_0 are compared in figure 4.5 at three time instants $t/t_\ell = 0, 4.02, 17.47$, corresponding to the visualizations in figure 4.3. The agreement is significant and LES seems to only suppress the high wavenumbers slightly. For both cases, the initial condition features about a decade of inertial-range turbulence with a $k^{-5/3}$ spectrum; as expected, this region shrinks (from the high wavenumbers) as the turbulence decays.

The third statistical measure we consider is the energy spectrum on a spherical shell. The flow is designed to be homogeneous in the azimuthal and polar directions but is only so in the radial direction deep within the cloud. As shown in § 4.4.1 visually the cloud also undergoes radial growth over time. This non-uniformity in the radial direction suggests that one should further characterize the energy spectrum as a function of the radius r and time t . A special spectrum defined on a spherical shell of a given radius is applied (Lombardini, Pullin, and Meiron, 2014). Similar to the total energy spectrum $E(k)$, it seeks a relationship between the energy and the wavenumber, where the wavenumber on a spherical shell is defined using the spherical harmonics. The spherical shell wavenumber and the classical wavenumber defined using Fourier transform can be related via the Laplace operator. This relation also connects the spherical shell spectrum to the classical energy spectrum.

Following the detailed derivation given by Lombardini, Pullin, and Meiron (2014),

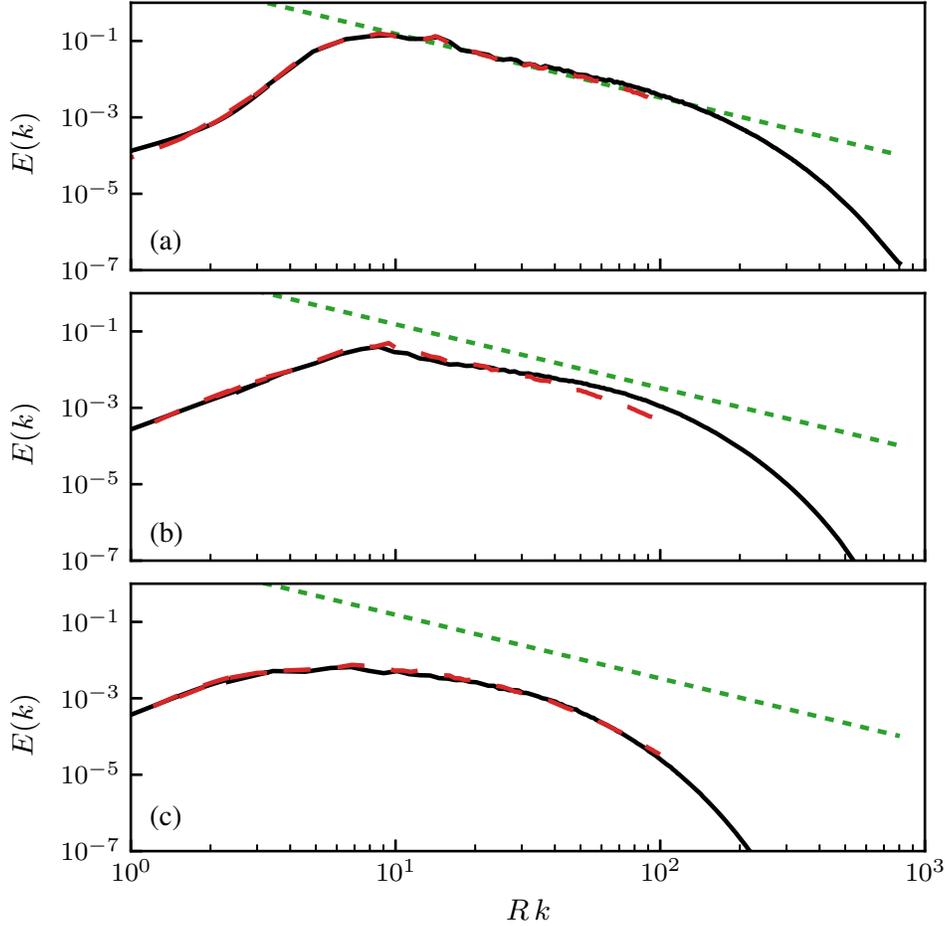


Figure 4.5: Total energy spectrum at (a) $t/t_\ell = 0$, (b) $t/t_\ell = 4.02$ and (c) $t/t_\ell = 17.47$ for DNS_0 (—) and LES_0 (- - -). The same guide line for $k^{-5/3}$ as in figure 5.1 is also given (.....).

we acquire the shell spectrum for a field $f_r(\theta, \phi)$ defined for a given radius r by expanding the field using spherical harmonics:

$$f_r(\theta, \phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_{\ell m} Y_{\ell m}(\theta, \phi), \quad (4.8)$$

where

$$Y_{\ell m}(\theta, \phi) = \begin{cases} N_{(\ell, m)} P_{\ell}^m(\cos \theta) \cos(m\phi) & m \geq 0 \\ N_{(\ell, |m|)} P_{\ell}^{|m|}(\cos \theta) \sin(|m|\phi) & m < 0 \end{cases}, \quad (4.9)$$

with P_{ℓ}^m being the associated Legendre polynomials, $N_{(\ell, m)}$ being the normalization constant and ℓ being the equivalent wavenumber. The wavenumber ℓ is then related to the classic wavenumber k defined through the Fourier transform by,

$$k^2 = \ell(\ell + 1)/r^2. \quad (4.10)$$

Assuming a power law for the energy spectrum $E(k) \sim k^{-\alpha}$ one has the following relationship between the energy spectrum and the shell spectrum,

$$E(k) \sim k^{-\alpha} \sim \ell C_\ell, \quad (4.11)$$

where

$$C_\ell = \frac{1}{2\ell + 1} \sum_{m=-\ell}^{\ell} f_{\ell m}^2. \quad (4.12)$$

Eq. (4.11) suggests that one can understand the shell spectrum C_ℓ in a similar way as the classical energy spectrum $E(k)$.

The shell spectra for DNS_0 and LES_0 are shown in figure 4.6. Results for various radii ($r/B = 0.32, 0.48, 0.64, 0.80, 0.96$) and time instants ($t/t_\ell = 0, 4.02, 17.47$) are given. At $t/t_\ell = 0$ (figure 4.6: a, d) all 5 curves collapse together as expected since they represent the original IHT field. The energy decays over time, but the dependence of the shell spectrum on the radius r is weak. It seems that the boundary does not have a strong effect on the turbulence decay. This evidence further supports the assumption of local homogeneity that underpins our definition of the total spectra used above, as discussed in the appendix, at least up through the times considered here.

4.6 Concluding remarks

In this chapter we introduced a new flow representing an isolated spherical region of turbulence evolving in free space and validated the stretched-vortex SGS (Chung and Pullin, 2009a) LES model in the LGF framework. This flow is created by tiling a periodic IHT field in space and windowing it with an indicator function which falls to zero outside a spherical region. We performed both the DNS and the LES of this flow at $Re_\lambda = 122.4$ and the results are compared. We showed the LES is able to capture most of the large features in the evolution of the turbulence boundary, and we demonstrated that the stretched-vortex SGS model predicts the total kinetic energy decay, the total energy spectrum, and the shell spectrum well, although at much lower resolution. This study indicates that the LES can further enhance the capability of the previously introduced AMR-IB-LGF algorithm for simulating external flows at even higher Reynolds numbers and lays the foundation for using the LES to study the long-term evolution of this flow itself which is discussed in detail in the next chapter.

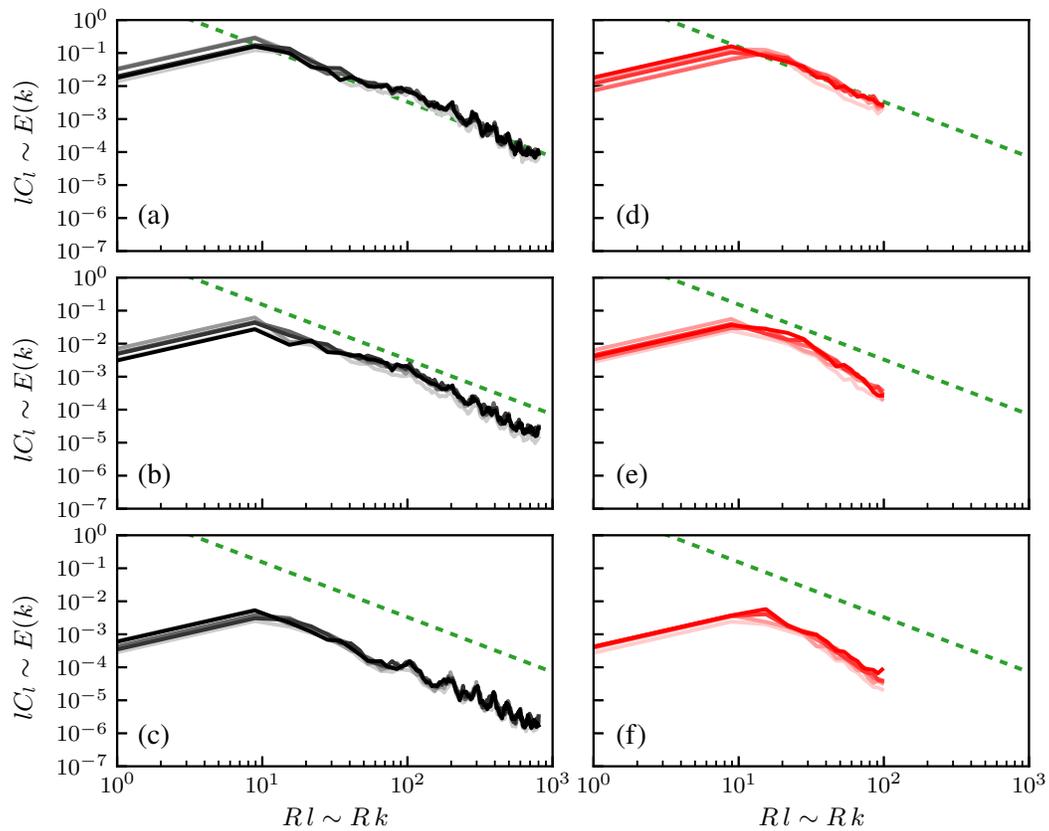


Figure 4.6: Spherical-shell spectrum at different radii and times. The left column (a-c) are results from DNS_0 at $t/t_\ell = 0, 4.02, 17.47$ respectively. The right column (d-f) are results from LES_0 at the same times. For each figure, the gradation in color corresponds to radii $r/B = (0.32, 0.48, 0.64, 0.80, 0.96)$ from darkest to lightest shade. An equivalent guide line for $k^{-5/3}$ as in figure 5.1 is given (.....).

*Chapter 5***DYNAMICS AND DECAY OF A SPHERICAL REGION OF
TURBULENCE****5.1 Introduction**

In this chapter we further address some of theoretical interest related to the dynamics of the spherical region of turbulence proposed in chapter 4. In IHT, the evolution of the largest scales is governed by the initial conditions, or, in the case of forced IHT, by the forcing scheme. For example, spectra with low wavenumber that asymptotes as k^2 (Saffman, 1967) and k^4 (Batchelor and Proudman, 1956) can be contrived (Chasnov, 1995; Ishida, Davidson, and Kaneda, 2006; Davidson, 2010). Indeed, the same is true for the spherical turbulence cloud, where, unlike IHT, the largest scales can subsequently grow and, as we will show, different behavior is obtained. The spherical region of turbulence also exemplifies the localized turbulence region introduced in Phillips (1956), where the final viscous stage of the evolution was studied theoretically.

A second motivation concerns the emergence of coherent structures. IHT is devoid of large-scale instabilities (typically associated with shear, buoyancy, or other imposed forces) that give rise to important classes of coherent structures. However, as we show, the same is not the case for the spherical cloud — we observe the formation of coherent vortex rings being ejected near the cloud edge.

A related issue is the interaction of the turbulent flow with the outer irrotational fluid at the turbulent/nonturbulent interface (TNTI). In recent work, TNTIs have been experimentally and numerically studied in shear layers and in numerically-constructed shear-free interfaces (Wolf et al., 2013; Silva et al., 2013; Da Silva, Taveira, and Borrell, 2014). The spherical cloud of turbulence also exhibits a TNTI, and may prove a useful source of data for further study, though in the present work we do not investigate its behavior in detail.

First in section 5.2 of this chapter, we discuss the theories regarding the energy spectrum and its low wavenumber limit for turbulence fields with localized vorticity. In section 5.3, we study the long-term evolution of a turbulence cloud through LES and in section 5.4 we discuss one distinctive feature in its evolution, the ejection of vortex rings, and we conjecture about the relationship between the initial condition

and the scale of the ejections. A brief summary of the main conclusions is given in section 5.5.

5.2 Initial spectrum and low wavenumber limit

As we only expect the turbulence cloud to remain homogeneous deep within the sphere, ambiguities arise in interpreting the energy spectrum: it can be viewed as the expectation of a random process, or merely as the Fourier transform of a deterministic function. In order to reach the broadest conclusions possible (i.e. ones not limited to the specific initial condition), we show in Appendix B that by invoking local homogeneity deep within the spherical region, we can estimate the total spectrum through a single realization of this flow. The estimated spectrum approximates the true one in the limit of large R/ℓ , which may only be barely reached in our simulations, but in principle could be improved upon in future. Thus we take

$$\begin{aligned}\tilde{E}(k) &= \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \mathbf{u}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}') e^{-ik \cdot (\mathbf{x}' - \mathbf{x})} d\mathbf{x} d\mathbf{x}' \\ &= \frac{1}{16\pi^3} |\mathcal{F}\{\mathbf{u}\}|^2 = \frac{1}{16\pi^3} \frac{1}{|k|^2} |\mathcal{F}\{\boldsymbol{\omega}\}|^2 \\ &= \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{1}{|k|^2} \boldsymbol{\omega}(\mathbf{x}) \cdot \boldsymbol{\omega}(\mathbf{x}') e^{-ik \cdot (\mathbf{x}' - \mathbf{x})} d\mathbf{x} d\mathbf{x}',\end{aligned}\quad (5.1)$$

$$E(k) = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\sin(k|\mathbf{x}' - \mathbf{x}|)}{k|\mathbf{x}' - \mathbf{x}|} \boldsymbol{\omega}(\mathbf{x}') \cdot \boldsymbol{\omega}(\mathbf{x}) d\mathbf{x} d\mathbf{x}',\quad (5.2)$$

where we expressed the spectrum in terms of the vorticity field (Phillips, 1956; Leonard, 1985; Winckelmans and Leonard, 1993; Winckelmans, 1995). The spherical symmetry of the problem is used in the last step where the 3-D energy spectrum $\tilde{E}(k)$ is integrated over a spherical shell to produce a scalar spectrum $E(k)$. Expanding $E(k)$ for the low wavenumber, the odd powers vanish, giving

$$E(k) = \frac{k^2}{4\pi^2} L + \frac{k^4}{24\pi^2} I + O(k^6),\quad (5.3)$$

where

$$L = -\frac{1}{6} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} |\mathbf{x}' - \mathbf{x}|^2 \boldsymbol{\omega}(\mathbf{x}') \cdot \boldsymbol{\omega}(\mathbf{x}) d\mathbf{x}' d\mathbf{x} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \mathbf{u}(\mathbf{x}') \cdot \mathbf{u}(\mathbf{x}) d\mathbf{x}' d\mathbf{x}\quad (5.4)$$

is the Saffman integral and

$$I = \frac{1}{20} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} |\mathbf{x}' - \mathbf{x}|^4 \boldsymbol{\omega}(\mathbf{x}') \cdot \boldsymbol{\omega}(\mathbf{x}) d\mathbf{x}' d\mathbf{x} = - \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} |\mathbf{x}' - \mathbf{x}|^2 \mathbf{u}(\mathbf{x}') \cdot \mathbf{u}(\mathbf{x}) d\mathbf{x}' d\mathbf{x}\quad (5.5)$$

is the Loitsyansky integral (Loitsyansky, 1939). Note that even though Eq. (5.1-5.3) are well defined for the flows with finite energy, using the velocity forms from Eq. (5.4, 5.5) requires certain decay rates of the velocity field. Thus we have used the vorticity formula for the calculation of the low wavenumber spectra. Details of the expansion and the calculation method are given in Appendix C.

The Saffman integral L is related to the total momentum impulse which is an invariant of the motion and remains constant for all time. When $L \neq 0$ the cloud will exhibit a small-wavenumber, k^2 Saffman limit (Saffman, 1967), whereas when $L = 0$ the spectrum is of the k^4 Batchelor type (Batchelor and Proudman, 1956). The ramifications of a Saffman or Batchelor spectrum have been widely explored in IHT, but less so in other (inhomogeneous) flows. As we discuss below, the low wavenumber spectrum can be used to derive asymptotic energy decay and integral-scale growth rates, which can be compared to those obtained for the spherical cloud.

In order to investigate this issue, we develop a procedure by which we control the value of L in the initial condition. Two factors contribute to the linear momentum. Firstly the IHT field generated from the pseudo-spectral code is continuously divergence-free but not necessarily discrete divergence-free as required by the finite volume FLGF scheme. Secondly, the windowing process will introduce extra non-solenoidality, i.e., given a divergence-free velocity field \mathbf{u} and a scalar window function $\Phi(r)$, $\Delta \cdot (\Phi(r)\mathbf{u}(\mathbf{x})) \neq 0$ in general. Both of these non-solenoidal components are projected out at the very first time step and this projection will introduce an impulse. The result of this impulse as mentioned in Batchelor (1967) is a $1/|\mathbf{x}|^3$ decaying velocity field of the form

$$\lim_{x \rightarrow \infty} \mathbf{u}(\mathbf{x}) = \frac{1}{8\pi} \nabla \left[\nabla \left(\frac{1}{|\mathbf{x}|} \right) \cdot \int_{\mathbb{R}^3} \mathbf{x}' \times \boldsymbol{\omega}(\mathbf{x}') d\mathbf{x}' \right]. \quad (5.6)$$

This suggests a way to cancel the impulse in order to have a k^4 type spectrum: one can add a vortex ring with an opposite impulse to the initial velocity field. More specifically, we add a Stokes vortex ring with velocity (Kambe and Oshima, 1975; Cantwell, 1986)

$$\mathbf{u}(r, \theta) = \frac{1}{(2\pi)^{3/2} \zeta^3} \left[W\left(\frac{r}{\zeta}; 1\right) \boldsymbol{\gamma} - W\left(\frac{r}{\zeta}; 3\right) \boldsymbol{\gamma} \cos \theta \hat{\mathbf{e}}_r \right], \quad (5.7)$$

where $\boldsymbol{\gamma}$ is the impulse of the vortex ring, θ is the angle between $\boldsymbol{\gamma}$ and the unit vector \mathbf{e}_r , ζ controls the size of the ring and

$$W(\rho; b) = e^{-\rho^2/2} - \frac{b}{\rho^3} \left[\sqrt{\frac{\pi}{2}} \operatorname{erf}\left(\frac{\rho}{\sqrt{2}}\right) - \rho e^{-\rho^2/2} \right]. \quad (5.8)$$

We chose $\zeta/R = 0.19$ and performed an LES computation with this cancellation, referred to as case LES_IC2 in table 4.1. Except for LES_IC2, all other cases are conducted without the cancellation.

While our method of manipulating the initial condition in order to cancel the finite impulse is arbitrary, it is effective in the sense that the added vortex ring quickly interacts with the turbulence. This was verified by monitoring the difference between simulations initialized with and without the cancellation. The results showed that in less than one initial large-eddy turnover time, the difference field was decorrelated with the added vortex ring. Thus we conclude that the two simulations can be regarded as representing (different random realizations of) locally homogeneous turbulence that differ significantly only in their low wavenumber spectrum.

Figure 5.1 shows the resulting initial energy spectrum $E_0(k)$ of a spherical region of turbulence field corresponding to the condition of DNS_0, superposed on another spectrum where Eq. (5.7) was used to cancel the impulse (i.e. the initial condition, after filter, for the case ‘LES_IC2’ in table 4.1). Also plotted is the energy spectrum of the original IHT field scaled by the ratio between the volume of the sphere and the cubic domain size B^3 . We see that the $k^{-5/3}$ portion of the spectrum from IHT is retained in the spherical cloud, whereas the low wavenumber behavior is controlled by the resulting impulse (or its absence).

5.3 Long-term statistics and low wavenumber behavior

The long-term evolution of a turbulence cloud is studied through LES. Two cases LES_0 and LES_IC2 from table 4.1 are simulated up to $t/t_\ell = 400$ where LES_0 has a k^2 -type initial spectrum while LES_IC2 has a k^4 -type. The evolution is visualized in figure 5.2. The spread of the cloud is similar in both cases, but details of the large-scale structures are different.

Figure 5.3a shows the long-term evolution of the kinetic energy decay. In the case of IHT, Saffman (1967) predicts an asymptotic decay rate of $t^{-6/5}$ for the k^2 turbulence and Kolmogorov (1962) predicts a decay rate of $t^{-10/7}$ for the k^4 spectrum (both guide lines are indicated in the figure). However in both the case of LES_0 and LES_IC2, the decay is similar and closer to the Saffman scaling. Figure 5.3b shows the evolution of the integral scale over time for LES_0 and LES_IC2, compared to the theoretical asymptotic growth rate for Saffman IHT ($t^{2/5}$) and Batchelor IHT ($t^{2/7}$). As in the energy decay, both cases are closer to the Saffman type.

This apparent discrepancy with the theory can be clarified by examining the long-

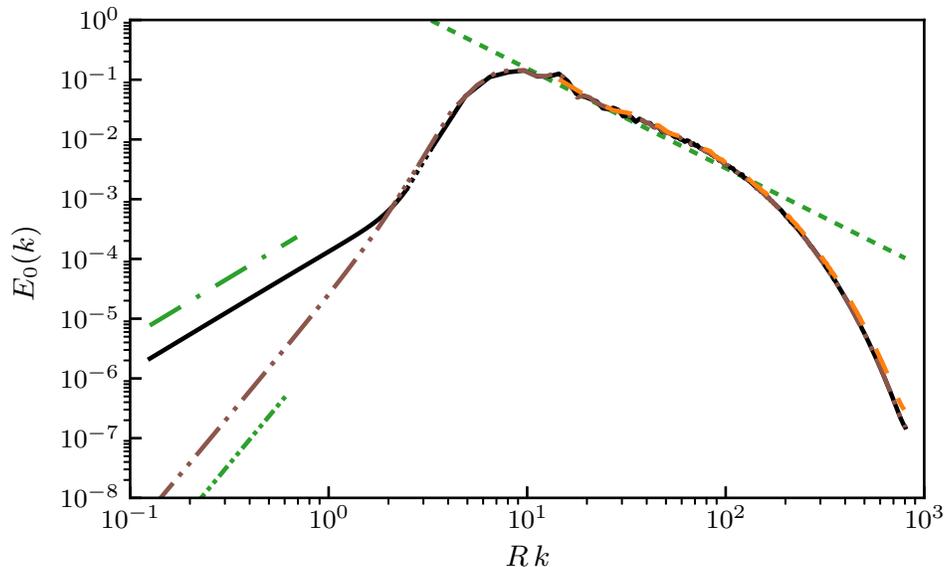


Figure 5.1: Energy spectrum of (1) the initial condition of simulation DNS_0. (—), where the low wavenumber limit (left of the dotted region) is calculated through an expansion method (Appendix C); (2) DNS_0 with the initial impulse cancelled using Eq. (5.7) (---); (3) the corresponding IHT field multiplied by the ratio between the spherical region volume $\frac{4}{3}\pi R^3$ and the cubic domain volume B^3 (- - -); (4) a guide line for $1.6\varepsilon^{2/3}k^{-5/3}$ scaled with the same ratio, where ε is the dissipation rate in the original IHT field (.....); (4) a slope of k^2 (- · - ·) and k^4 (- · · -) for the low wavenumber limit.

term decay of the total energy spectrum depicted in figure 5.4, which shows results for both LES_0 (k^2) and LES_IC2 (k^4) cases. The k^4 spectrum is similar to that reported in Ishida, Davidson, and Kaneda (2006). As expected the coefficient of the limiting k^2 spectrum for LES_0 is invariant, but the coefficient of the k^4 term for LES_IC2 is increasing over time first rapidly up to about $t/t_\ell \simeq 20$, and then more slowly, as shown in figure 5.5. The coefficient of the k^4 term is proportional to the Loitsyansky integral I given by Eq. (5.5) which is assumed constant in the theory under the assumption that remote points be statistically independent (Loitsyansky, 1939).

Superposition of the energy spectra for cases LES_0 and LES_IC2 shows they are similar for $kR > 1$, corresponding to a wavelength $\lambda = 2\pi R$. Apart from the very largest scales, which cannot be seen in visualizations like figure 5.2, the two simulations are otherwise statistically similar. The weak vortex ring and vortex ring dipole associated with the k^2 and k^4 terms would only become evident as $t \rightarrow \infty$, after which all turbulence will decay. However at the same time, the properties are entirely predictable from their initial conditions. Indeed, figure 5.6 shows that the entire cloud of LES_0 meanders in space but eventually attains a trajectory that is

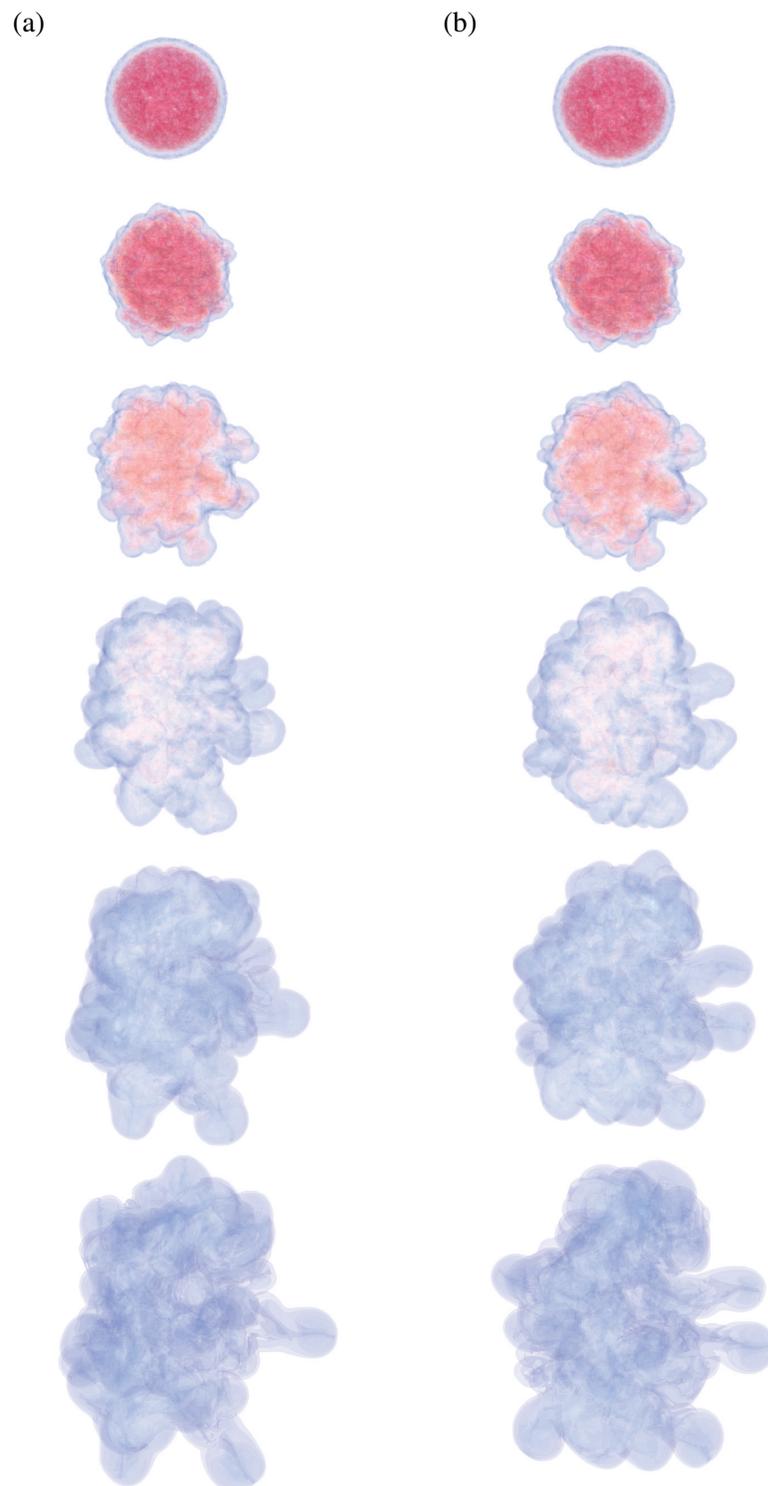


Figure 5.2: Long-term evolution of (a) LES_0 and (b) LES_IC2 at $t/t_e = 0, 4, 17, 66, 143, 263$ from top to bottom.

associated with the initial impulse.

The predictability of the long-term evolution from the initial condition argues against any universality of the very largest scales of the spherical cloud of turbulence. While we expect the wavenumber spectrum for $kR > 1$ is approximately universal, the low wavenumber behavior is always an artifact of initial and boundary conditions.

While this lack of universality is perhaps unsurprising, the veracity of the Saffman-type decay-rate predictions even in the absence of a k^2 spectrum is interesting. Consider the process by which the initial k^4 spectrum is created for the case LES_IC2, whereby a weak vortex ring is added to offset the initial impulse associated with windowing the IHT field. While we superposed this ring at the center of our cloud, we could have cancelled the impulse by adding a ring at any position, even one very far from the cloud. Over the timescale simulated here, the results would be identical to those of the k^2 cloud, and one would have to go to even lower values of kR in order to see the ultimate k^4 behavior.

Lastly we consider the radial growth of the turbulence cloud over time. Because the cloud does not hold its sphericity we define the radius by a statistical moment

$$\bar{r} = \left(\frac{\int u^2 |\mathbf{x} - \mathbf{x}_c|^p \, d\mathbf{x}}{\int u^2 \, d\mathbf{x}} \right)^{1/p}, \quad (5.9)$$

where u is the velocity magnitude and \mathbf{x}_c is the center of the turbulence cloud, defined using

$$\mathbf{x}_c = \frac{\int \mathbf{x} u^2 \, d\mathbf{x}}{\int u^2 \, d\mathbf{x}}. \quad (5.10)$$

A definition of the center is necessary because, as discussed in § 4.3.1, the final stage of a turbulence cloud is a large vortex ring drifting in the direction of the impulse. Also $p \leq 2$ is needed for \bar{r} to exist as the velocity field $u(\mathbf{x}) \sim 1/|\mathbf{x}|^3$ as $|\mathbf{x}| \rightarrow \infty$. Here we only consider the case when $p = 2$ for simplicity. The results between LES_0 and LES_IC2 are shown in figure 5.7. The mean radius growth in time is almost the same for both cases, and approaches a power-law behavior that is similar to the growth of the integral scale.

5.4 Vortex ring ejections

One of the most distinctive features in the late-stage evolution of a turbulence cloud (figure 5.2) is that it ejects vortex rings of roughly the same size from its boundary.

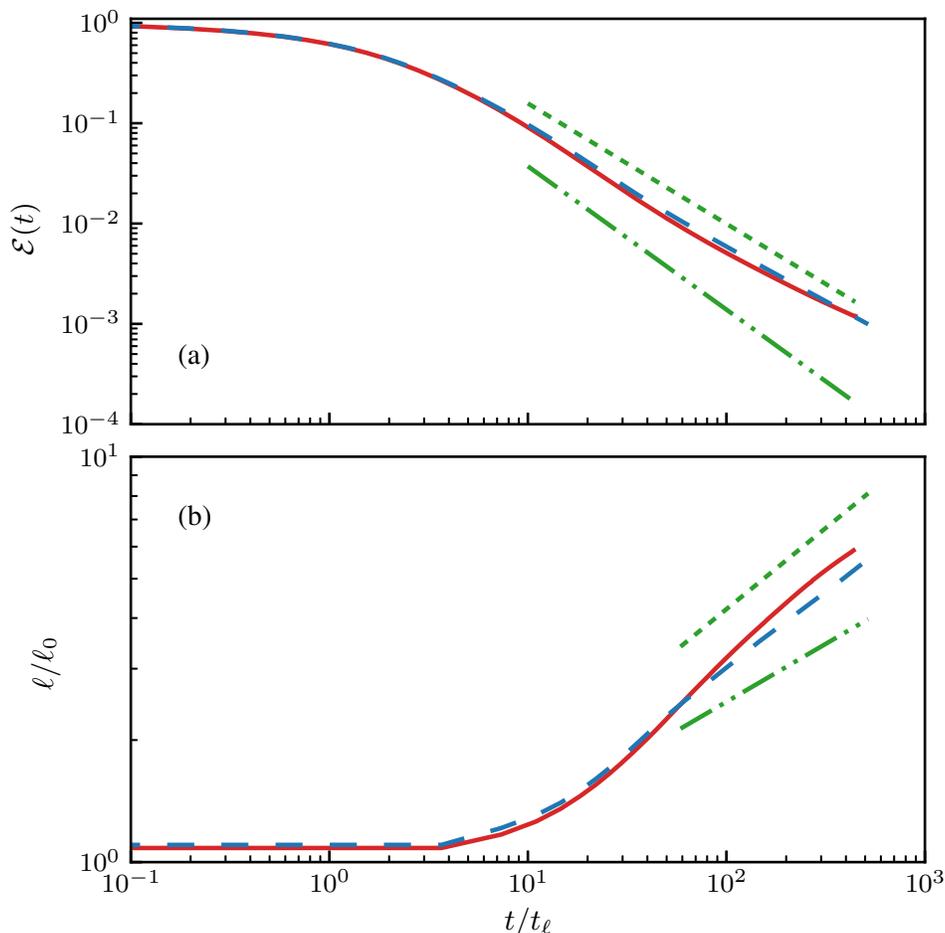


Figure 5.3: Long-term evolution of (a) the kinetic energy decay compared with asymptotic behavior of Saffman IHT $\mathcal{E}(t) \sim t^{-6/5}$ (.....) and Batchelor IHT $\mathcal{E}(t) \sim t^{-10/7}$ (-.-.); (b) the integral scale growth for case LES_0 (—) and LES_IC2 (- - -) up to $t/t_\ell = 400$ compared with asymptotic behavior of Saffman IHT $\ell \sim t^{2/5}$ (.....) and Batchelor $\ell \sim t^{2/7}$ (-.-.).

In this section, we investigate the relation between the size of the vortex rings and properties associated with the turbulence.

We create LES simulations which independently vary the three independent non-dimensional parameters that control the initial conditions. Long-term evolution for all three pairs at $t = 260$ are provided in Figure 5.8. The first parameter is the width of the transition region associated with the windowing function, σ/R , which is varied from [0.05,0.1,0.2] in three cases [LES_D1, LES_0, LES_D2]. We see that the width of the transition region has little influence on the number or scale of the ejections. Next, we consider varying the microscale Reynolds number Re_λ , which is varied from [45.0, 76.9, 122.4] in three cases [LES_R2, LES_R1, LES_0]. Again, though each cloud has a different range of scales present, the vortex ejections

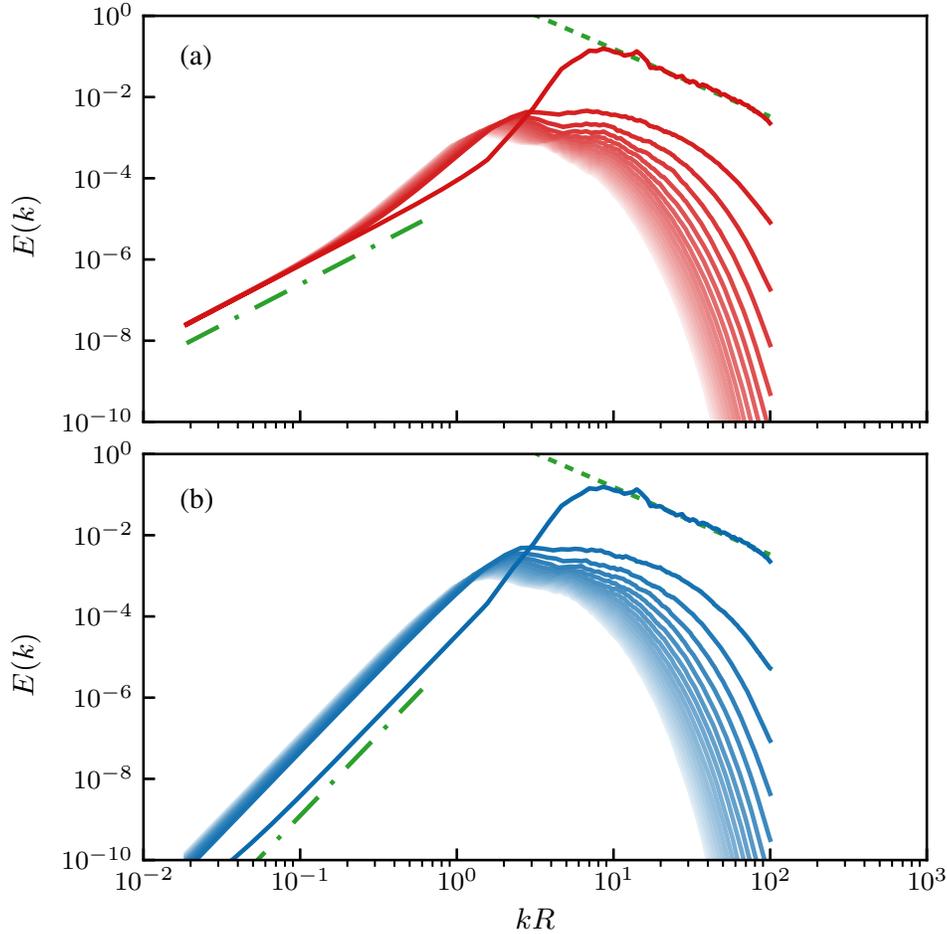


Figure 5.4: Long-term evolution of the total spectrum for (a) LES_0 and (—) (b) LES_IC2 (—), up to $t/t_\ell = 500$ with $\Delta t/t_\ell = 15$ between each line. In figure (a) guide lines for $k^{-5/3}$ (.....) and k^2 (-.-.-) are given. In figure (b) guide lines for $k^{-5/3}$ (.....) and k^4 (-.-.-) are given.

occur again at roughly the same scale. Finally, we vary initial integral scale ℓ/R , by changing the size of the initial periodic box to the sphere radius, B/R over the range $[0.5, 1.0, 2.0]$ for cases [LES_B1, LES_0, LES_B2]. Quite evidently, the size of the ejections is halved for case LES_B1 and doubled for case LES_B2, compared to the baseline LES_0. Therefore we conjecture that the vortex rings are generated by the integral-scale structures in the original IHT field.

We hypothesize that the ejections occur due to a local imbalance of impulse associated with the IHT field. Consider a Gaussian weighted impulse centered at point \mathbf{x} , with a ‘width’ ς

$$\mathbf{I}(\mathbf{x}; \varsigma) = \int_{\mathbb{R}^3} e^{-\frac{|\mathbf{x}-\mathbf{x}'|^2}{2\varsigma^2}} \mathbf{u}(\mathbf{x}') d\mathbf{x}'. \quad (5.11)$$

Figure 5.9 shows the maximum impulse $\mathbf{I}(\mathbf{x}; \varsigma)$ over \mathbf{x} , as a function of the width

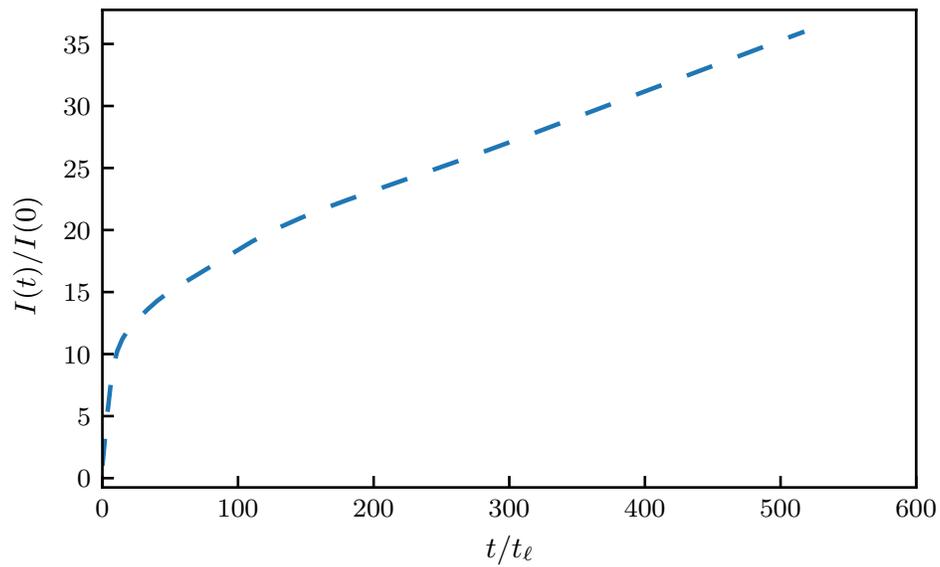


Figure 5.5: Long-term evolution of the normalized Loitsyansky integral $I(t)/I(0)$ of LES_IC2 (k^4 type), up to $t/t_\ell = 500$.

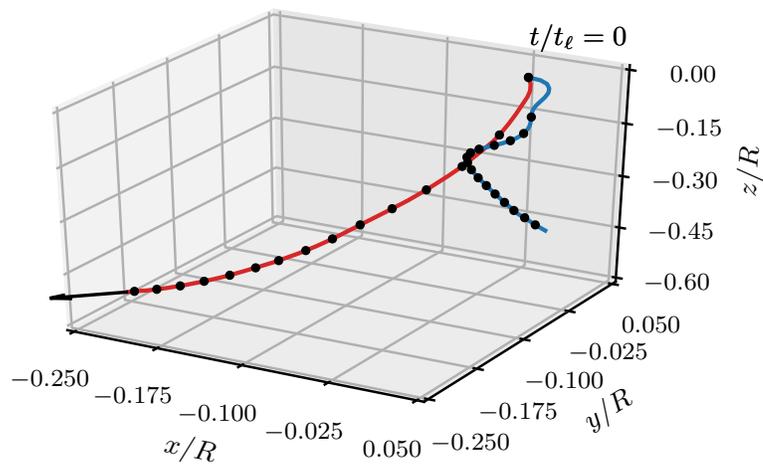


Figure 5.6: Trajectory of the center of the turbulence cloud from $t/t_\ell = 0$ to 400 for LES_0 (—) and LES_IC2 (—) with every marker separated by $\Delta t/t_\ell = 30$. For LES_0 the direction of the initial impulse is indicated with the arrow at the end of the trajectory. For LES_IC2 the impulse is zero.

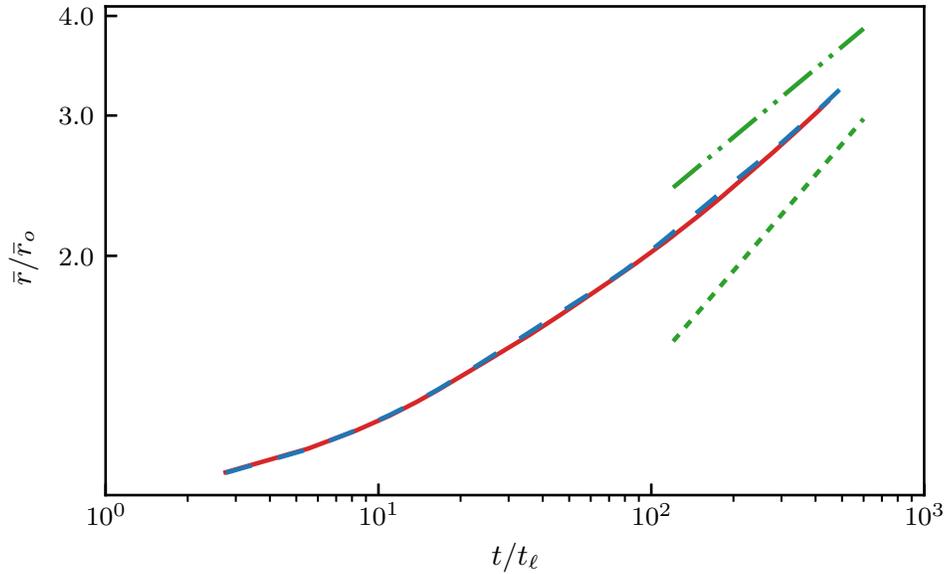


Figure 5.7: Comparison of the long-term mean radius \bar{r} for $p = 2$ between LES_0 (—) and LES_IC2 (---) up to $t/t_\ell \sim 400$. Results are normalized with their initial mean radii \bar{r}_0 . Guide lines for $t^{2/5}$ (.....) and $t^{2/7}$ (-.-.-) are also given.

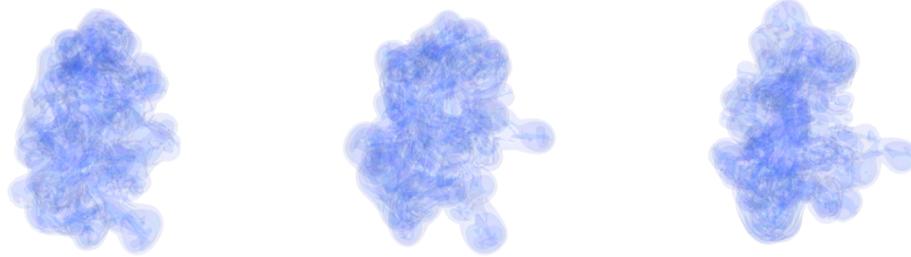
ς/ℓ_o at $t/t_\ell = 0$, where ℓ_o is the initial integral scale. The maximum Gaussian weighted impulse reaches its maximum when ς/ℓ_o is around 1.8.

For points deep within the cloud, imbalance of the locally filtered impulse would simply result in complicated local vortex dynamics. However, near the edge of the cloud, this imbalance, when pointed outwards, would eject vorticity out of the cloud. In some sense, this process is universal as the scale is a property of the IHT field itself, and the net imbalance would create ejections near the edge of any region of IHT. This result also agrees with studies of TNTIs. It was discussed in Townsend (1980) that, while a wide range of turbulence scales affect the evolution of the turbulence boundary, the largest distortion at the TNTI is from the largest eddies in the turbulence.

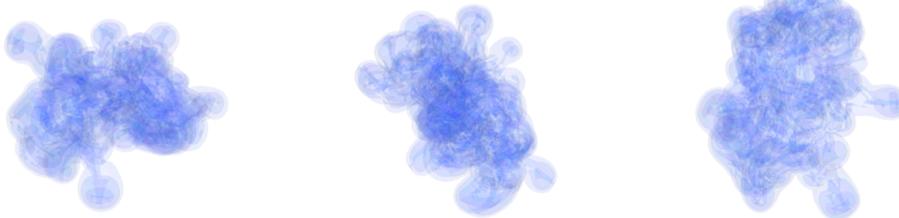
5.5 Concluding remarks

The flow exhibits aspects of both homogeneous turbulence, deep within the sphere, as well as inhomogeneous turbulence near the TNTI. For strictly homogeneous turbulence, a spectrum of either the Saffman k^2 type or the Batchelor k^4 type determines the kinetic energy decay rate and the the integral scale growth rate. For spherical region of turbulence we showed that both types of initial conditions can also be created. For the cloud, we confirm, by comparing spectra on spherical shells of different radii from the initial center, that the turbulence remains locally

$$\sigma/R = [0.05, 0.1, 0.2]$$



$$\text{Re}_\lambda = [45.0, 76.9, 122.4]$$



$$B/R = [0.05, 0.1, 0.2]$$

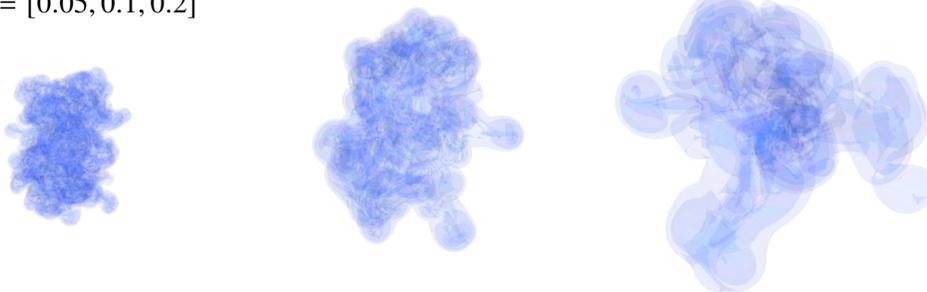


Figure 5.8: Long-term turbulence cloud evolution with vortex ring ejections for cases defined in table 4.1. First row: LES_D1, LES_0, LES_D2; second row: LES_R2, LES_R1, LES_0; third row: LES_B1, LES_0, LES_B2.

homogeneous deep within the cloud. However, the resulting long-term decay of the kinetic energy and the growth of the integral scale are similar in both cases, and closer to the predictions of the Saffman theory. This may be related to an observed growth in the Loitsyansky integral, but which is assumed constant in the Batchelor characterization of the turbulence. At least through about 400 eddy turnover times, there is little difference in the shape of the respective spectra between the two cases for k values near the inertial scale, and it appears that the integral scale is relatively unaffected by the behavior at very low k , whether k^2 or k^4 . In any event, the spectrum at these wavenumbers is controlled by the initial conditions and may not be universal. Finally, we defined a mean radius of the turbulence cloud in terms of its velocity moments, and showed the turbulence gives rise to a similar growth of radius as of the integral scale.

The spherical region of turbulence is bounded by a TNTI that evolves into distinct

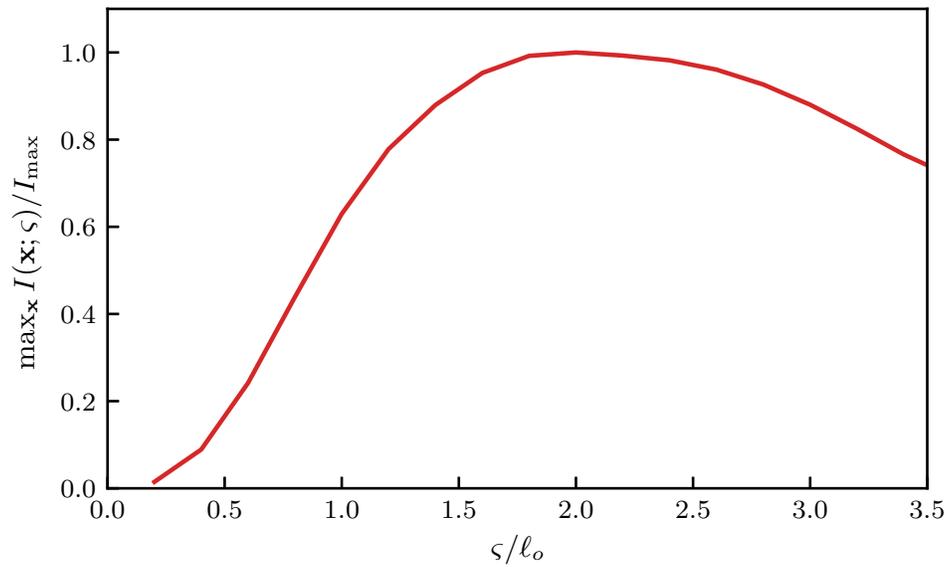


Figure 5.9: Maximum Gaussian weighted impulse over \mathbf{x} , $\max_{\mathbf{x}} I(\mathbf{x}; \varsigma)$ as a function of ‘width’ ς for LES_0 at $t = 0$ (—).

large-scale features. By varying each of the three independent nondimensional parameters controlling the cloud, we find that the structures are related to the (initial) integral scale of the IHT field. The TNTI features include vortex rings that are ejected from the cloud. We hypothesize that this evolution is associated with an imbalance in specific impulse over the integral scale, which, near the TNTI, gives rise to the vortex rings.

SUMMARY AND OUTLOOK

6.1 Summary

In this thesis, we expanded the state-of-the-art in simulating three-dimensional Navier-Stokes equations for unsteady, viscous, external flows. This new CFD technique aims to address the three main challenges present in simulating high Reynolds number external flows: the unbounded domain; the complex immersed geometries; and the scale separation associated with the thin boundary layers and turbulence. The original IBLGF method by Liska and Colonius (2017) applies the exact free-space boundary conditions and allows for adaptive computational domains with active grid cells comprising only vortical flow regions which is highly efficient, scalable, and robust. However the necessity of a regular Cartesian grid is problematic for resolving thin boundary layers at high Reynolds numbers as compared to body-fitted meshes where a more flexible nodal distribution is allowed. In this work we solved this issue by further developing a multi-resolution framework that extends the original method.

We began by proposing a novel multi-resolution framework for the LGF method for solving Poisson equation on unbounded domains. The new formulation of the AMR-LGF technique is based on the concept of composite grid, which is defined as a series of infinite lattices of differing resolution, and we consider the AMR grid as a subset of the ambient composite grid. The LGFs are formally applied to every level of the composite grid with right-hand sides approximated using the information on the AMR grid. By considering the (non-)commutativities between the interpolation/coarsening and the LGFs, we show that this hypothetical process can be evaluated efficiently. In this way, we retain the original infinite-space mimetic differential operators whose commutative and conservation properties lead to the efficiency and robustness of the original method. The resulting method was also demonstrated to retain good parallel scaling with little penalty associated with the AMR overhead. The AMR-LGF method is then applied to solve the Navier-Stokes equations subject to far-field boundary conditions by incorporating an analytical integrating factor technique and an appropriate Runge-Kutta scheme for the time-marching. The resulting AMR-LGF-HERK scheme is demonstrated

with the collision of vortex rings at $Re = 7500$.

In Chapter 3, we further extended the scheme to be coupled with the IB method to solve external flows with complex geometries. We showed that the idea of the composite grid can readily be applied to the immersed boundary method and the resulting IB Navier-Stokes formulation can be solved efficiently using the AMR-LGF method developed, as well as the spatial and refinement adaptivity discussed in Chapter 2. The new scheme was validated and demonstrated with DNS of flow around a sphere at $Re = 3700$ and $10,000$, where the spatial and refinement adaptivity is shown to save 99.2% and 99.87% of the computational cells respectively, with up to 350×10^3 number of IB points in the $Re = 10,000$ case. We also demonstrated that the scheme is capable of simulating flows around complex geometries at high Reynolds numbers by considering a DNS of a delta wing at $Re = 10,000$, where 97.5% of saving in the computational cells was realized.

To push the aforementioned numerical method to even higher Reynolds numbers, we explored in Chapter 4 the possibility of combining LES sub-grid-scale model specifically the stretched-vortex of Chung and Pullin, 2009a, with the LGF method for the turbulence simulations. In order to validate the LES, we designed a novel turbulent flow in free-space—the spherical cloud of turbulence and performed DNS and LES of this flow at $Re_\lambda = 122.4$. The LES was shown to capture most of the large features in the boundary evolution and agreed well with DNS in three statistical measures. This study indicates that the LES modeling will further enhance the AMR-IB-LGF method in simulating flows at higher Reynolds numbers. In addition, the turbulent dynamics proved interesting and allowed us to address some fundamental questions in the theory of (nearly) isotropic homogeneous turbulence, namely the evolution of the low-wavenumber (Batchelor- or Saffman-type) turbulence, which was discussed in detail in Chapter 5.

6.2 Outlook

The AMR-IB-LGF framework is capable of performing state-of-the-art CFD simulations, yet there remain several issues to address to reach external flow problems of relevance in myriad scientific and engineering applications. Luckily, emerging advances in the CFD community provide new directions to further extend the current method.

In Chapter 4 and 5 we showed that the LES modeling can be fruitfully combined with the LGF method for the simulation of turbulence evolution in free space. This LES

modeling can readily be applied to the AMR-IB-LGF method for wall-resolved LES. In addition, the IB method adopted in the current method is also compatible with virtual-wall model techniques, and we expect the AMR-IB-LGF algorithm together with wall-modeled LES to be able to solve a broader range of fluid problems.

The AMR-IB-LGF method is based on the AMR-LGF method which is developed from the fast LGF method for uniform grid. The superior efficiency, as well as the principle disadvantage of the original method stems from the regularity of the underlying discretization. The AMR-LGF algorithm unlocks greater flexibility from the FLGF method as now only locally regular grids are required. We expect the AMR-LGF method to be applicable to a wide range of physics and scientific problems that involves kernel convolution with the need of flexible nodal distribution such as quantum fluids, or 3D incompressible flows with one homogeneous direction. Both of these are being actively studied in the group. Furthermore, since this method only requires local regularity in its discretization, it does not rely upon any specific numerical scheme. For instance, the current solver is built on the finite-volume scheme but it could be easily adapted to the finite-element method.

Advances in hardware architecture also bring new opportunities into the future developments of the AMR-IB-LGF method. As mentioned, this method develops from the FLGF method which is based on the FMM algorithm and block-wise FFT operations. For the AMR-IB-LGF solver, often we see over 95% of the total computational time spent on the FFT calculations. GPUs have shown to be a more suitable hardware for large-scale parallel FFTs compared to CPUs, which makes the current method an ideal fit for GPU or heterogeneous CPU/GPU architectures.

The first-order convergence of the DLM-IB method adopted in the current algorithm often requires a finer mesh to resolve the boundary layer, which can be inefficient at high Reynolds numbers. However it is not the only option for solving PDEs with irregular boundaries on an Eulerian grid. For instance, cut-cell and sharp interface methods can extend the IB treatment to second order. Preliminary results also indicate that when solving external flows with immersed bodies of finite volume, it is possible to modify the symmetric smearing and interpolation operators used in the current IB method to be asymmetric to yield more accurate solutions on the outside the surface with less accurate velocity solution inside.

We believe that the the AMR-IB-LGF method will provide ample opportunities for discovery of new flow phenomena and provide a good foundation for better, emerging CFD techniques.

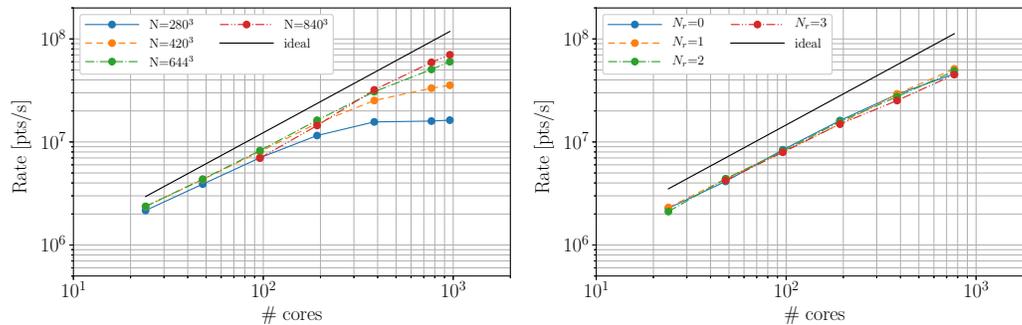
*Appendix A***EFFICIENCY AND PARALLEL PERFORMANCE OF AMR-LGF
POISSON SOLVER**

To demonstrate the efficiency and parallel performance of the proposed block-refined algorithm, we consider the same test case as in section 2.3.4 and report its efficiency and parallel performance. We also refer to Liska and Colonius (2014) for performance investigations and the parallel implementation strategy of the uniform solver, which is similar to the one employed here. However, for completeness we also report the computational rates here.

The solver was written in a C++ framework and uses MPI for parallel communications as well as FFTW for fast Fourier transforms. The code implements an octree data structure, where each leaf corresponds to a cubic domain in physical space. A server-client model is used for load balancing of the octree, where the server stores the full octree but does not allocate any data. The clients on the other hand store one or multiple sub-trees including the corresponding data. For load balancing, the anticipated load (mainly the load of fast Fourier transforms) is computed for each octant and the leaf octants are sorted according to their Morton code for each level. Finally, the sorted array of leaf octants is split into chunks with almost equal loads, which are then assigned to each processor. Subsequently, the parents are assigned to the processor with the minimum load in a recursive fashion. Note that the FMMs are sequential in terms of the level due to the correction term, which depends on previous levels and thus necessitates level-wise balancing to avoid an imbalance of load on a particular level. All communication patterns between clients are established using the server and communication costs are almost fully hidden using non-blocking MPI calls.

Note that by far most time is spent in the level interactions ($\sim 99\%$) of the algorithm and in particular the level convolution within each FMM. The time to construct or traverse the octree data structure is negligible due to the block-wise nature of the algorithm, which also allows SIMD vectorization of the Fourier transforms and the Hadamard product for additional speed.

Figure A.1a shows the strong scaling for the computational rates of our implementation for various domain sizes N . The parallel efficiencies are in line with the implementation of Liska and Colonius (2014) as well as other kernel-independent



(a) Computational rates for a uniform mesh of size N . (b) Computational rates for $N = 644^3$ and an increasing number of refinement levels.

Figure A.1: Computational rates and parallel performance.

FMM solvers and thus verifies our implementation. In addition, in figure A.1b, the dependence of the computational rate with the number of refinement levels is plotted for the case of $N = 644^3$. As stated earlier, the complexity scales linearly with the number of levels and we thus expect that the computational rates are independent of the refinement levels. This is confirmed in A.1b. Note however that the parallel efficiency is limited to the efficiency of the individual FMMs.

Appendix B

ENERGY SPECTRUM OF INHOMOGENEOUS TURBULENT FLOWS

Ambiguities arise in interpreting the (spatial) energy spectrum of inhomogeneous turbulent flows. The term ‘spectrum’ itself can be used in two different ways. For a function, the spectrum can refer to the magnitude of its Fourier transform and gives information about the scales present in the function. For a random process, on the other hand, the spectrum represents a statistical statement about how energy is distributed amongst scales *on average*. Turbulence is generally thought to be random in the sense (e.g. Pope, 2001) that any realization (e.g. specific initial condition) cannot be predicted with certainty from any other; only through an average of a sufficient number of realizations of the random process can we make statements about the likely properties of any. In what follows, we interpret the term spectrum in this latter sense. We discuss approximations we make in order to estimate the spectrum for the spherical region of turbulence under the approximation that the turbulence is locally homogeneous deep within the sphere.

B.1 General definitions

The *spatial* energy spectrum for an inhomogeneous flow can be formulated from the two-point velocity covariance tensor (written at some moment in time, and we suppress the temporal dependence in what follows)

$$R_{ij}(\mathbf{x}, \mathbf{r}) = \mathbb{E} [u_i(\mathbf{x})u_j(\mathbf{x} + \mathbf{r})], \quad (\text{B.1})$$

where \mathbb{E} is the expectation (ensemble average). Note that the result depends on both the position in the flow \mathbf{x} and the separation vector \mathbf{r} between the observations. The Fourier transform (in the generalized sense) of R_{ij} over the separation vector gives the cross-spectral density tensor

$$S_{ij}(\mathbf{x}, \mathbf{k}) = \frac{1}{8\pi^3} \int_{\mathbb{R}^3} R_{ij}(\mathbf{x}, \mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}} d\mathbf{r}, \quad (\text{B.2})$$

where the integral is over the separation vector.

The resulting local kinetic energy spectrum (per unit volume) is

$$\widehat{E}(\mathbf{x}, \mathbf{k}) = \frac{1}{2} S_{ii}(\mathbf{x}, \mathbf{k}). \quad (\text{B.3})$$

We can define a total kinetic energy spectrum of a domain Ω by integration,

$$\tilde{E}(\mathbf{k}) = \int_{\Omega} \widehat{E}(\mathbf{x}, \mathbf{k}) \, d\mathbf{x}, \quad (\text{B.4})$$

and, by Parseval, the kinetic energy in the entire flow is $\int \tilde{E}(\mathbf{k}) \, d\mathbf{k}$. For turbulence that is homogeneous in one or more directions, discussed in more detail below, the integral diverges and the total energy is not defined. However, in that case the local kinetic energy is also uniform in the homogeneous directions, and it is sufficient to speak of the energy per unit volume.

For future reference, using linearity of the expectation operator, we may also write

$$\tilde{E}(\mathbf{k}) = \frac{1}{16\pi^3} \mathbb{E} \left[\int_{\mathbb{R}^3} \int_{\mathbb{R}^3} u_i(\mathbf{x}) u_i(\mathbf{x} + \mathbf{r}) e^{-i\mathbf{k} \cdot \mathbf{r}} \, d\mathbf{x} \, d\mathbf{r} \right]. \quad (\text{B.5})$$

This form of the spectrum is often written without the expectation operator, but it then refers to the spectrum of a deterministic velocity field rather than that of an underlying random process. Evaluating it with the expectation requires an ensemble of realizations for the general case of inhomogeneous turbulence.

B.2 Homogeneous and locally homogeneous turbulence

Depending on additional hypotheses on the structure of the turbulence, different averaging procedures can be employed to determine a spatial or temporal spectrum in place of the ensemble average over realizations. For example, if the turbulence is hypothesized as ergodic-stationary, then the ensemble average can be replaced by a sufficiently long time average over a single realization. Likewise, if the turbulence is hypothesized as ergodic-homogeneous, then a spatial average over any or all homogeneous directions can be used. For example, for the fully (all 3 directions) homogeneous case, we may write

$$R_{ij} = R_{ij}(\mathbf{r}) = \lim_{V \rightarrow \mathbb{R}^3} \frac{1}{V} \left(\int_V u_i(\mathbf{x}) u_j(\mathbf{x} + \mathbf{r}) \, d\mathbf{x} \right). \quad (\text{B.6})$$

The resulting cross-spectral density and local energy spectrum will likewise only be functions of the separation or wavenumber vectors, respectively, i.e. $S_{ij} = S_{ij}(r)$, $\tilde{E} = \tilde{E}(\mathbf{k})$. Recall that total energy spectrum $\tilde{E}(\mathbf{k})$ is infinite (undefined) in this case, since the integral over all space diverges.

Provided that the turbulence is *locally* homogeneous (or homogeneous plus isotropic) over a lengthscale L such that $L \gg l$ (l the integral scale) then the volume averaging

can be performed locally. Define a region Ω centered about \mathbf{x} with scale $\Omega \sim L^3$, and define

$$\bar{R}_{ij}(\mathbf{x}, \mathbf{r}) = \frac{1}{|\Omega(\mathbf{x})|} \left(\int_{\Omega(\mathbf{x})} u_i(\mathbf{x}') u_j(\mathbf{x}' + \mathbf{r}) d\mathbf{x}' \right). \quad (\text{B.7})$$

We expect \bar{R}_{ij} to be a constant over the region of homogeneity (except close to its edge).

B.2.1 Estimating $\tilde{E}(\mathbf{k})$ for the spherical cloud of turbulence

We hypothesize that the turbulence is locally homogeneous over a region deep within the sphere of turbulence. To apply this concept to the spherical cloud, we begin by breaking up the volume in equation (B.5) into 3 parts: an inner region ($\Omega_{<R}$) deep in the sphere where we will assume local homogeneity, a transition region ($\Omega_{\sim R}$) near the turbulent/irrotational interface, and an outer, irrotational region ($\Omega_{>R}$)

$$\tilde{E}(\mathbf{k}) = \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \left[\int_{\Omega_{<R}} + \int_{\Omega_{\sim R}} + \int_{\Omega_{>R}} \right] \mathbb{E} [u_i(\mathbf{x}) u_i(\mathbf{x} + \mathbf{r})] e^{-i\mathbf{k} \cdot \mathbf{r}} d\mathbf{x} d\mathbf{r}. \quad (\text{B.8})$$

With local homogeneity over $\Omega_{<R}$, we insert equation (B.7), which is constant with \mathbf{x} over this region, into the first integral, and obtain

$$\begin{aligned} \tilde{E}(\mathbf{k}) &= \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \int_{\Omega_{<R}} u_i(\mathbf{x}) u_i(\mathbf{x} + \mathbf{r}) e^{-i\mathbf{k} \cdot \mathbf{r}} d\mathbf{x} d\mathbf{r} \\ &+ \frac{1}{16\pi^3} \left[\int_{\Omega_{\sim R}} + \int_{\Omega_{>R}} \right] \mathbb{E} [u_i(\mathbf{x}) u_i(\mathbf{x} + \mathbf{r})] e^{-i\mathbf{k} \cdot \mathbf{r}} d\mathbf{x} d\mathbf{r} \\ &= \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} u_i(\mathbf{x}) u_i(\mathbf{x} + \mathbf{r}) e^{-i\mathbf{k} \cdot \mathbf{r}} d\mathbf{x} d\mathbf{r} + E'_{\Omega_{\sim R}} + E'_{\Omega_{>R}}(\mathbf{k}), \end{aligned} \quad (\text{B.9})$$

where the remainder terms are of the form of a *difference* between the ensemble average and one realization, i.e.

$$\tilde{E}'(\mathbf{k})_{\Omega} = \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \int_{\Omega} (\mathbb{E} [u_i(\mathbf{x}) u_i(\mathbf{x} + \mathbf{r})] - u_i(\mathbf{x}) u_i(\mathbf{x} + \mathbf{r})) e^{-i\mathbf{k} \cdot \mathbf{r}} d\mathbf{x} d\mathbf{r}. \quad (\text{B.10})$$

In breaking up the integral in this way, we highlight that we can integrate the velocity field from a single simulation over free space and obtain the correct ensemble-averaged spectrum up to a statistical error associated only with the difference between one realization and the ensemble average only over the transition and outer regions.

Regarding the transition region, the contribution to the overall energy scales with the volume of this region, $4\pi\sigma R^2$, where σ is the width of the transition region. By making the initial sphere large compared to σ and the correlation length (integral

scale), $\ell(x)$, this error can, at least in principle, be made indefinitely small compared to the first term.

Regarding the outer region, the irrotational velocity field decays at least as fast as $|\mathbf{x}|^{-3}$ when the initial impulse is nonzero. For wavenumber k not too small, we expect this to only produce a small contribution to the total energy spectrum. However, as $k \rightarrow 0$, this term will eventually *dominate* the spectrum, and the behavior at low k will be $O(k^2)$ provided the initial impulse is nonzero. Indeed, we therefore do not expect the low wavenumber spectrum to be universal as it depends on how much impulse there is in the initial condition, which, as described in § 4.3.1, is arbitrary and can be contrived, with little effect on the resulting turbulence, to have any value (including zero). When the impulse is zero, the resulting k^4 spectrum may be universal over a broader range of low wavenumbers, but is still contrived as $k \rightarrow 0$.

This discussion has strong but unsurprising implications about whether the low wavenumber spectrum of *any* turbulent flow can be considered to be universal. An alternative interpretation is that it is associated with the initial/boundary conditions and can be arbitrarily manipulated independently of the turbulence behavior at smaller scales. In any event, it is clear that, in the present simulations, the low wavenumber behavior is wholly controlled by the (arbitrary) initial condition. At sufficiently long time, after the turbulence has substantially decayed, the error terms above will eventually dominate the spectrum, resulting in a (in the case of finite impulse), a fat vortex “puff” whose properties are solely related to the initial condition.

If the error terms are neglected, equation (B.9) is identical to equation (B.5), but without the expectation. It has been used before to express the energy spectra of deterministic velocity fields but its equivalence (to within the error) to the ensemble-averaged spectrum of a random process, under local homogeneity, has not to our knowledge been reported elsewhere. It is interesting that the error vanishes like the ratio of the volume of inhomogeneous turbulence to volume of homogeneous turbulence.

Appendix C

SPECTRUM: ANALYTICAL AND COMPUTATIONAL DETAILS

For $kR \gtrsim 1$, equation (5.2) is evaluated using the three-dimensional vorticity form

$$\tilde{E}(\mathbf{k}) = \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \mathbf{u}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}') e^{-i\mathbf{k} \cdot (\mathbf{x}' - \mathbf{x})} d\mathbf{x} d\mathbf{x}' \quad (\text{C.1})$$

$$\begin{aligned} &= \frac{1}{16\pi^3} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{1}{|\mathbf{k}|^2} \boldsymbol{\omega}(\mathbf{x}) \cdot \boldsymbol{\omega}(\mathbf{x}') e^{-i\mathbf{k} \cdot (\mathbf{x}' - \mathbf{x})} d\mathbf{x} d\mathbf{x}' \\ &= \frac{1}{16\pi^3 |\mathbf{k}|^2} \mathcal{F}\{\boldsymbol{\omega}(\mathbf{x})\} \cdot \overline{\mathcal{F}\{\boldsymbol{\omega}(\mathbf{x})\}}, \end{aligned} \quad (\text{C.2})$$

$$E(k) = \int_{S_k} \tilde{E}(\mathbf{k}) dS_k, \quad (\text{C.3})$$

where $\mathcal{F}\{\cdot\}$ denotes the Fourier transform and S_k denotes a spherical shell of radius k . Fast Fourier transform is used to efficiently evaluate (C.2) and zero-padding is applied to attenuate the effect from the spurious periodicity.

For $kR \lesssim 1$, formula (C.2) must be evaluated carefully to avoid numerical singularity. For these values, we expand the integral in a Taylor series about $k = 0$ to obtain

$$E(k) = \frac{1}{4\pi^2} \sum_{i=1}^{\infty} (-1)^i \frac{1}{(2i+1)!} k^{2i} \int_{\mathbb{R}^3} |\mathbf{r}|^{2i} \mathcal{F}^{-1} \left\{ \mathcal{F}\{\boldsymbol{\omega}\} \cdot \overline{\mathcal{F}\{\boldsymbol{\omega}\}} \right\}(\mathbf{r}) d\mathbf{r}, \quad (\text{C.4})$$

which already uses the relation that $\int \boldsymbol{\omega}(\mathbf{x}) d\mathbf{x} = 0$. For all results presented here, up to 10 terms are used to yield accurate spectrum for the low wavenumber limit.

A more common form in terms of the velocity field for the low wavenumber limit can be derived by expanding equation (C.1)

$$E(k) = \frac{1}{4\pi^2} \sum_{i=0}^{\infty} (-1)^i \frac{1}{(2i+1)!} k^{2i+2} \int_{\mathbb{R}^3} |\mathbf{r}|^{2i} \mathcal{F}^{-1} \left\{ \mathcal{F}\{\mathbf{u}\} \cdot \overline{\mathcal{F}\{\mathbf{u}\}} \right\}(\mathbf{r}) d\mathbf{r}. \quad (\text{C.5})$$

Comparing the corresponding terms in equation (C.1) and (C.2) gives another relation

$$\int_{\mathbb{R}^3} \int_{\mathbb{R}^3} r^{2p+2} \boldsymbol{\omega}(\mathbf{x}) \cdot \boldsymbol{\omega}(\mathbf{x} + \mathbf{r}) d\mathbf{x} d\mathbf{r} = -(2p+2)(2p+3) \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} r^{2p} \mathbf{u}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x} + \mathbf{r}) d\mathbf{x} d\mathbf{r}. \quad (\text{C.6})$$

Lastly one can show the k^2 term in equation (C.4) is related to the total vorticity impulse \mathbf{J}_ω through

$$\int_{\mathbb{R}^3} \int_{\mathbb{R}^3} r^2 \boldsymbol{\omega}(\mathbf{x}) \cdot \boldsymbol{\omega}(\mathbf{x} + \mathbf{r}) \, d\mathbf{x} \, d\mathbf{r} = 4|\mathbf{J}_\omega|^2, \quad (\text{C.7})$$

where

$$\mathbf{J}_\omega = \frac{1}{2} \int_{\mathbb{R}^3} \mathbf{x} \times \boldsymbol{\omega}(\mathbf{x}) \, d\mathbf{x}. \quad (\text{C.8})$$

BIBLIOGRAPHY

- Achenbach, E. (1972). “Experiments on the flow past spheres at very high Reynolds numbers”. In: *J. Fluid Mech* 54.3 (1972), pp. 565–575.
- Almgren, A. S., J. B. Bell, P. Colella, L. Howell, and M. Welcome (1998). “A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier–Stokes Equations”. en. In: *J. Comput. Phys.* 142.1 (1998), pp. 1–46.
- Batchelor, G. K. (1967). *An introduction to fluid dynamics*. Cambridge University Press, 1967.
- Batchelor, G. K. and I. Proudman (1956). “The large-scale structure of homogeneous turbulence”. In: *Phil. Trans. A* 248 (1956), pp. 369–405.
- Berger, M. J. and P. Colella (1989a). “Local adaptive mesh refinement for shock hydrodynamics”. en. In: *J. Comput. Phys.* 82.1 (May 1989), pp. 64–84. ISSN: 00219991.
- (1989b). “Local adaptive mesh refinement for shock hydrodynamics”. In: *J. Comput. Phys.* 82.1 (1989), pp. 64–84.
- Berger, M. J. and J. E. Olinger (1984). “Adaptive mesh refinement for hyperbolic partial differential equations”. In: *J. Comput. Phys.* 53.3 (1984), pp. 484–512.
- Brasey, V. and E. Hairer (1993). “Half-Explicit Runge–Kutta Methods for Differential-Algebraic Systems of Index 2”. en. In: *J. Numer. Anal.* 30.2 (1993), pp. 538–552.
- Cantwell, B. J. (1986). “Viscous starting jets”. In: *J. Fluid Mech.* 173 (1986), pp. 159–189.
- Chasnov, J. R. (1995). “The decay of axisymmetric homogeneous turbulence”. In: *Phys. Fluids* 7 (1995), pp. 600–605.
- Chatelain, Philippe and Petros Koumoutsakos (2010). “A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions”. en. In: *J. Comput. Phys.* 229.7 (Apr. 2010), pp. 2425–2431. ISSN: 0021-9991.
- Cheng, W., D.I. Pullin, and R. Samtaney (2018). “Large-eddy simulation of flow over a grooved cylinder up to transcritical Reynolds numbers”. In: *J. Fluid Mech.* 835 (2018), pp. 327–362.
- (2020). “Large-eddy simulation and modelling of Taylor–Couette flow”. In: *J. Fluid Mech* 890 (2020), Art–No A17.
- Cheng, W., D.I. Pullin, R. Samtaney, W. Zhang, and W. Gao (2017). “Large-eddy simulation of flow over a cylinder with Re_D from 3.9×10^3 to 8.5×10^5 : a skin-friction perspective”. In: *J. Fluid Mech.* 820 (2017), pp. 121–158.

- Cheng, Wan and R. Pullin D.I. and Samtaney (2015). “Large-eddy simulation of a separation/reattachment bubble in a turbulent-boundary-layer subjected to a prescribed upper-boundary, vertical-velocity profile”. In: *Proceedings of Ninth International Symposium on Turbulence and Shear Flow Phenomena (TSFP-9)*. 2015.
- Choi, H. and P. Moin (2012). “Grid-point requirements for large eddy simulation: Chapman’s estimates revisited”. In: *Phys. Fluids* 24.1 (2012), p. 011702.
- Chung, D and DI Pullin (2009a). “Large-eddy simulation and wall modelling of turbulent channel flow”. In: *J. Fluid Mech* 631 (2009), pp. 281–309.
- (2009b). “Large-eddy simulation and wall modelling of turbulent channel flow”. In: *J. Fluid Mech.* 631 (2009), pp. 281–309.
- (2010). “Direct numerical simulation and large-eddy simulation of stationary buoyancy-driven turbulence”. In: *J. Fluid Mech* 643 (2010), pp. 279–308.
- Colonus, T. (2004). “Modeling Artificial Boundary Conditions for Compressible Flow”. In: *Annu. Rev. Fluid Mech.* 36.1 (2004), pp. 315–345.
- Colonus, T. and K. Taira (2008). “A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions”. In: *Comput. Method Appl. Mech. Eng.* 197.25 (2008), pp. 2131–2146.
- Constantinescu, G. S. and K. D. Squires (2003). “LES and DES investigations of turbulent flow over a sphere at $Re=10,000$ ”. In: *Flow, Turbulence and Combustion* 70.1 (2003), pp. 267–298.
- Da Silva, C. B., R. R. Taveira, and G. Borrell (2014). “Characteristics of the turbulent/nonturbulent interface in boundary layers, jets and shear-free turbulence”. In: *J. Phys.: Conf. Ser.* 506 (2014), p. 012015.
- Davidson, P. A. (2010). “On the decay of Saffman turbulence subject to rotation, stratification or an imposed magnetic field”. In: *J. Fluid Mech.* 663 (2010), pp. 268–292.
- Delves, R. T. and G. S. Joyce (2001). “On the Green Function for the Anisotropic Simple Cubic Lattice”. en. In: *Annu. Phys.s* 291.1 (July 2001), pp. 71–133. ISSN: 0003-4916.
- Dorschner, B., K. Yu, G. Mengaldo, and T. Colonius (2020). “A fast multi-resolution lattice Green’s function method for elliptic difference equations”. In: *J. Comput. Phys.* (2020), p. 109270. DOI: <https://doi.org/10.1016/j.jcp.2020.109270>.
- Gillman, A. and P.G. Martinsson (2010). “Fast and accurate numerical methods for solving elliptic difference equations defined on lattices”. en. In: *J. Comput. Phys.* 229.24 (Dec. 2010), pp. 9026–9041. ISSN: 00219991.

- Gillman, A. and P.G. Martinsson (2014). “A fast solver for Poisson problems on infinite regular lattices”. en. In: *J. Comput. Appl. Math.* 258 (Mar. 2014), pp. 42–56. ISSN: 03770427.
- Glasser, M. L. and I. J. Zucker (1977). “Extended Watson integrals for the cubic lattices”. en. In: *Proc. Natl. Acad. Sci.* 74.5 (May 1977), pp. 1800–1801. ISSN: 0027-8424, 1091-6490.
- Goza, A., S. Liska, B. Morley, and T. Colonius (2016). “Accurate computation of surface stresses and forces with immersed boundary methods”. en. In: *J. Comput. Phys.* 321 (Sept. 2016), pp. 860–873. ISSN: 0021-9991.
- Hairer, E., C. Lubich, and M. Roche (2006). *The numerical solution of differential-algebraic systems by Runge-Kutta methods*. Vol. 1409. Springer, 2006.
- He, X., X. An, D. R. Williams, and M. Le Provost (2019). “Pressure Feedback Active Flow Control of Unsteady Roll Moment on a UCAS Delta Wing”. en. In: *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2019.
- He, X., M. Le Provost, and D. R. Williams (2018). “Dynamic Active Flow Control of the Roll Moment on a Generic UCAS Wing”. In: *2018 AIAA Aerospace Sciences Meeting*. 2018.
- Hirt, C. W., A. A. Amsden, and J. L. Cook (1974). “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”. In: *J. Comput. Phys.* 14.3 (1974), pp. 227–253.
- Huang, M. (1994). “Theoretical and computational studies of isotropic homogeneous turbulence”. PhD thesis. California Institute of Technology, 1994.
- Inoue, M., Z. Pullin D.I. and Harun, and I. Marusic (2013). “LES of the adverse-pressure gradient turbulent boundary layer”. In: *International Journal of Heat and Fluid Flow* 44 (2013), pp. 293–300.
- Ishida, T., P. A. Davidson, and Y. Kaneda (2006). “On the decay of isotropic turbulence”. In: *J. Fluid Mech.* 564 (2006), pp. 455–475.
- James, R. A. (1977). “The solution of poisson’s equation for isolated source distributions”. en. In: *J. Comput. Phys.* 25.2 (Oct. 1977), pp. 71–93. ISSN: 00219991.
- Jang Y, I. and S. J. Lee (2007). “Visualization of turbulent flow around a sphere at subcritical Reynolds numbers”. In: *J. Vis.* 10.4 (2007), pp. 359–366.
- Kambe, T. and Y. Oshima (1975). “Generation and decay of viscous vortex rings”. In: *J. Phys. Soc. Japan* 38 (1975), pp. 271–280.
- Kamkar, S., A. Jameson, A. Wissink, and V. Sankaran (2011). “Automated Off-Body Cartesian Mesh Adaption for Rotorcraft Simulations”. en. In: *49th AIAA Aerospace Sciences Meeting*. Orlando, Florida: AIAA, Jan. 2011.

- Kolmogorov, A. N. (1962). “A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high Reynolds number”. In: *J. Fluid Mech.* 13 (1962), pp. 82–85.
- Lackner, K. (1976). “Computation of ideal MHD equilibria”. en. In: *Comput. Phys. Comm.* 12.1 (1976), pp. 33–44. ISSN: 00104655.
- Lashuk, I., A. Chandramowlishwaran, H. Langston, T. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin, and G. Biros (2012). “A Massively Parallel Adaptive Fast Multipole Method on Heterogeneous Architectures”. In: *Commun. ACM* 55.5 (May 2012), pp. 101–109. ISSN: 0001-0782.
- Leonard, A (1980). “Vortex methods for flow simulation”. en. In: *J. Comput. Phys.* 37.3 (Oct. 1980), pp. 289–335. ISSN: 0021-9991.
- (1985). “Computing three-dimensional incompressible flows with vortex elements”. In: *Ann. Rev Fluid Mech.* 17 (1985), pp. 523–559.
- Lesieur, M. and O. Metais (1996). “New trends in large-eddy simulations of turbulence”. In: *Annu. Rev. Fluid Mech.* 28.1 (1996), pp. 45–82.
- Lim, T. T. and T. B. Nickels (1992). “Instability and reconnection in the head-on collision of two vortex rings”. In: *Nature* 357 (1992), pp. 225–227.
- Liska, S. and T. Colonius (2016). “A fast lattice Green’s function method for solving viscous incompressible flows on unbounded domains”. In: *J. Comput. Phys.* 316 (2016), pp. 360–384.
- (2017). “A fast immersed boundary method for external incompressible viscous flows using lattice Green’s functions”. en. In: *J. Comput. Phys.* 331 (Feb. 2017), pp. 257–279. ISSN: 00219991.
- Liska, Sebastian and T. Colonius (2014). “A parallel fast multipole method for elliptic difference equations”. In: *J. Comput. Phys.* 278 (2014), pp. 76–91.
- Loitsyansky, L. G. (1939). “Some basic laws of isotropic turbulence”. In: *Trudy Tsent. Aero.-Gidrodin. Inst.* 440 (1939), pp. 3–23.
- Lombardini, M., D.I. Pullin, and D. Meiron (2014). “Turbulent mixing driven by spherical implosions. Part 2. Turbulence statistics”. In: *J. Fluid Mech.* 748 (2014), pp. 113–142.
- Lundgren, T. S. (1982). “Strained spiral vortex model for turbulent fine structure”. In: *The Phys. Fluids* 25 (1982), pp. 2193–2203.
- MacNeice, Peter, Kevin M. Olson, Clark Mobarry, Rosalinda de Fainchtein, and Charles Packer (2000). “PARAMESH: A parallel adaptive mesh refinement community toolkit”. en. In: *Comput. Phys. Comm.* 126.3 (Apr. 2000), pp. 330–354. ISSN: 0010-4655.
- Mathew, J. and A. J. Basu (2002). “Some characteristics of entrainment at a cylindrical turbulence boundary”. In: *Phys. Fluids* 14 (2002), pp. 2065–2072.

- Matsuzawa, T., N. Mitchell, S. Perrard, and W. Irvine (2019). “Realization of Confined Turbulence Through Multiple Vortex Ring Collisions”. In: *APS Meeting Abstracts*. 2019.
- McCorquodale, P., P. Colella, G. T. Balls, and S. B. Baden (2007). “A local corrections algorithm for solving Poisson’s equation in three dimensions”. en. In: *Commun. Appl. Math. Comput. Sci.* 2.1 (Aug. 2007), pp. 57–81. ISSN: 2157-5452, 1559-3940.
- McKeown, R., R. Ostilla-Mónico, A. Pumir, M. P. Brenner, and S. M. Rubinstein (2018). “Cascade leading to the emergence of small structures in vortex ring collisions”. In: *Phys. Rev. Fluids* 3 (2018), p. 124702.
- Meneveau, C. and J. Katz (2000). “Scale-invariance and turbulence models for large-eddy simulation”. In: *Annu. Rev. Fluid Mech.* 32.1 (2000), pp. 1–32.
- Misra, A. and D.I. Pullin (1997). “A vortex-based subgrid stress model for large-eddy simulation”. In: *Phys. Fluids* 9.8 (1997), pp. 2443–2454.
- Mittal, Rajat and Gianluca Iaccarino (2005). “Immersed boundary methods”. In: *Annu. Rev. Fluid Mech.* 37 (2005), pp. 239–261.
- Moin, P. and K. Mahesh (1998). “Direct numerical simulation: a tool in turbulence research”. In: *Annu. Rev. Fluid Mech.* 30.1 (1998), pp. 539–578.
- Nissen, Anna, Katharina Kormann, Magnus Grandin, and Kristoffer Virta (2015). “Stable Difference Methods for Block-Oriented Adaptive Grids”. en. In: *J. Sci. Comput.* 65.2 (Nov. 2015), pp. 486–511. ISSN: 0885-7474, 1573-7691.
- Oshima, Yuko (1978). “Head-on collision of two vortex rings”. In: *J. Phys. Soc. Japan* 44 (1978), pp. 328–331.
- Papoutsakis, Andreas, Sergei S. Sazhin, Steven Begg, Ionut Danaila, and Francky Luddens (2018). “An efficient Adaptive Mesh Refinement (AMR) algorithm for the Discontinuous Galerkin method: Applications for the computation of compressible two-phase flows”. en. In: *J. Comput. Phys.* 363 (June 2018), pp. 399–427.
- Persson, P. and G. Strang (2004). “A simple mesh generator in MATLAB”. In: *SIAM review* 46.2 (2004), pp. 329–345.
- Persson, Per-Olof (2005). “Mesh generation for implicit geometries”. PhD thesis. Massachusetts Institute of Technology, 2005.
- Peskin, C. S. (1972). “Flow patterns around heart valves: a numerical method”. In: *J. Comput. Phys.* 10.2 (1972), pp. 252–271.
- Phillips, O. M. (1956). “The final period of decay of non-homogeneous turbulence”. In: *Proc. Camb. Phil. Soc.* Vol. 52. 1. 1956, p. 135.
- Pope, S. B. (2001). *Turbulent Flows*. Cambridge University Press, 2001.

- Popinet, S. (2003). “Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries”. In: *J. Comput. Phys.* 190.2 (2003), pp. 572–600.
- Quirk, J. J. (1996). “A parallel adaptive grid algorithm for computational shock hydrodynamics”. en. In: *Appl. Numer. Math.* 20.4 (Apr. 1996), pp. 427–453.
- Rodriguez, I., R. Borell, O. Lehmkuhl, C. D. P. Segarra, and A. Oliva (2011). “Direct numerical simulation of the flow over a sphere at $Re = 3700$ ”. In: *J. Fluid Mech* 679 (2011), pp. 263–287.
- Rodriguez, I., O. Lehmkuhl, R. Borrell, and A. Oliva (2013). “Flow dynamics in the turbulent wake of a sphere at sub-critical Reynolds numbers”. In: *Computers & Fluids* 80 (2013), pp. 233–243.
- Saffman, P. G. (1967). “The large-scale structure of homogeneous turbulence”. In: *J. Fluid Mech.* 27 (1967), pp. 581–593.
- Sagaut, P. (2006). *Large eddy simulation for incompressible flows: an introduction*. Springer Science & Business Media, 2006.
- Saito, N. and M. Pullin D.I. and Inoue (2012). “Large eddy simulation of smooth-wall, transitional and fully rough-wall channel flow”. In: *Phys. Fluids* 24.7 (2012), p. 075103.
- Saito, Namiko and D.I. Pullin (2014). “Large eddy simulation of smooth–rough–smooth transitions in turbulent channel flows”. In: *Int. J. Heat Mass Transf.* 78 (2014), pp. 707–720.
- Shariff, Karim, Roberto Verzicco, and Paolo Orlandi (1994). “A numerical study of three-dimensional vortex ring instabilities: viscous corrections and early nonlinear stage”. In: *J. Fluid Mech.* 279 (1994), pp. 351–375.
- Shenoy, R., M. J. Smith, and M. A. Park (2014). “Unstructured Overset Mesh Adaptation with Turbulence Modeling for Unsteady Aerodynamic Interactions”. en. In: *J. Aircr.* 51.1 (Jan. 2014), pp. 161–174.
- Silva, C. M. de, J. Philip, K. Chauhan, C. Meneveau, and I. Marusic (2013). “Multiscale geometry and scaling of the turbulent-nonturbulent interface in high Reynolds number boundary layers”. In: *Phys. Rev. Lett.* 111 (2013), p. 044501.
- Sitaraman, J., M. Floros, A. Wissink, and M. Potsdam (2010). “Parallel domain connectivity algorithm for unsteady flow computations using overlapping and adaptive grids”. en. In: *J. Comput. Phys.* 229.12 (2010), pp. 4703–4723.
- Spalart, P. R. (2009). “Detached-eddy simulation”. In: *Annu. Rev. Fluid Mech.* 41 (2009), pp. 181–202.
- Sussman, Mark, Ann S Almgren, John B Bell, Phillip Colella, Louis H Howell, and Michael L Welcome (1999). “An adaptive level set approach for incompressible two-phase flows”. In: *J. Comput. Phys.* 148.1 (1999), pp. 81–124.

- Swinbank, R. and R. James Purser (2006). “Fibonacci grids: A novel approach to global modelling”. In: *Quarterly Journal of the Royal Meteorological Society* 132.619 (2006), pp. 1769–1793.
- Tezduyar, T. E., M. Behr, S Mittal, and J. Liou (1992). “A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders”. In: *Comput. Method Appl. Mech. Eng.* 94.3 (1992), pp. 353–371.
- Tezduyar, T. E., S. Sathe, R. Keedy, and K. Stein (2006). “Space–time finite element techniques for computation of fluid–structure interactions”. In: *Comput. Method Appl. Mech. Eng.* 195.17-18 (2006), pp. 2002–2027.
- Townsend, A. A. R. (1980). *The structure of turbulent shear flow*. Cambridge University Press, 1980.
- Tsynkov, S. V. (1998). “Numerical solution of problems on unbounded domains. A review”. In: *Appl. Numer. Math.* 27.4 (1998), pp. 465–532.
- Voelkl, T., D.I. Pullin, and D. C. Chan (2000). “A physical-space version of the stretched-vortex subgrid-stress model for large-eddy simulation”. In: *Phys. Fluids* 12 (2000), pp. 1810–1825.
- Werle, H. (1981). “Transition and separation: Visualizations in the ONERA water tunnel”. In: *In its La Rech. Aerospaciale* 1980.1980-5 (1981), pp. 35–49.
- Winckelmans, G. S. (1995). “Some progress in large-eddy simulation using the 3-D vortex particle method”. In: *CTR Annual Research Briefs* (1995), pp. 391–415.
- Winckelmans, G. S. and A. Leonard (1993). “Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows”. In: *J. Comput. Phys.* 109 (1993), pp. 247–273.
- Wolf, M., M. Holzner, B. Lüthi, D. Krug, W. Kinzelbach, and A. Tsinober (2013). “Effects of mean shear on the local turbulent entrainment process”. In: *J. Fluid Mech.* 731 (2013), pp. 95–116.
- Yang, X., X. Zhang, Z. Li, and G. He (2009). “A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations”. In: *J. Comput. Phys.* 228.20 (2009), pp. 7821–7836.
- Ying, L., G. Biros, and D. Zorin (2004). “A kernel-independent adaptive fast multipole algorithm in two and three dimensions”. en. In: *J. Comput. Phys.* 196.2 (May 2004), pp. 591–626. ISSN: 00219991.
- Yun, G., D. Kim, and H. Choi (2006). “Vortical structures behind a sphere at subcritical Reynolds numbers”. In: *Phys. Fluids* 18.1 (2006), p. 015102.