

UNIVERSITY OF CALIFORNIA, SAN DIEGO

The Interaction of Near-wall Turbulence with Compliant Tensegrity Fabrics:
Modeling, Simulation, and Optimization

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Engineering Sciences (Mechanical Engineering)

by

Haoxiang Luo

Committee in charge:

Professor Thomas R. Bewley, Chair
Professor Randolph E. Bank
Professor J. William Helton
Professor Constantine Pozrikidis
Professor Robert E. Skelton

2004

UMI Number: 3144307

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 3144307

Copyright 2004 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

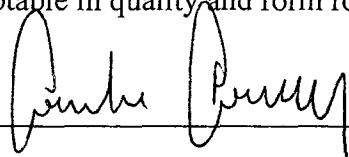
ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Copyright

Haoxiang Luo, 2004

All rights reserved.

The dissertation of Haoxiang Luo is approved, and it is acceptable in quality and form for publication on microfilm:



J. William Helton



R. S. W. L.



Chair

University of California, San Diego

2004

But seek first his kingdom and his righteousness, and all these things will be given to you as well.

Matthew 6:33

TABLE OF CONTENTS

Signature Page	iii
Table of Contents	v
Preface	viii
Acknowledgements	xii
Vita	xiii
Abstract	xiv
 I Modeling and simulation of the interaction of near-wall turbulence with compliant tensegrity fabrics	 1
Abstract	2
1. Introduction	3
1.1. The compliant surface problem	3
1.2. Tensegrity fabrics: a new class of compliant surfaces	4
1.3. DNS models for channel flows with moving boundaries	5
2. Governing equation	7
2.1. Domain transformation	7
2.2. Choice of dependent variables	10
2.3. Boundary conditions	12
3. Numerical algorithm	12
3.1. Spatial discretization	13
3.2. Temporal discretization	16
3.3. Pressure equation	17
4. Code validation	19
4.1. Laminar flow through a sinusoidal channel	19
4.2. Three-dimensional laminar flow over a bump	20
4.3. Moving-boundary test	24
4.4. Turbulence over a stationary wavy wall	26
5. The compliant surface model: tensegrity fabric	31
5.1. Introduction	31
5.2. Dynamics of the tensegrity structure	34
5.3. Interconnection of the multiple-bar system	36
5.4. Tendon forces in the interconnected system	38
5.5. Periodic boundary condition for the infinite tensegrity fabric	38
6. Simulation of a turbulent flow over tensegrity fabrics	40
6.1. Nondimensionalization of the tensegrity parameters	40
6.2. Representation of the flow/structure interface	40
6.3. Simulation parameters	41
7. Results	42
7.1. Case II and III	43
7.2. Case VII and VIII	56
7.3. Dependence on domain size	56

7.4. Visualization and statistics of the other cases	63
8. Concluding remarks	63
A. List of the terms in Eq. (I.7)	69
B. Details of solving Eq. I.11	71
C. Boundary conditions on the additional pressure update equation solved at the beginning of each timestep	73
References	75
 II On the contravariant form of the Navier-Stokes equations in time-dependent curvilinear coordinate systems	79
Abstract	80
1. Introduction	80
2. Derivation of the Navier-Stokes equations in moving coordinate systems	82
2.1. Equations of motion in a fixed coordinate system	82
2.2. The intrinsic derivative	84
2.3. Equations of motion in a moving coordinate system	87
2.4. Example: Uniform flow in a rotating coordinate system	88
2.5. Example: Stagnation point flow in a material coordinate system	88
3. Derivation by direct transformation	89
4. Application: flow in a channel with moving walls	90
4.1. Numerical algorithm	92
4.2. Laminar steady flow in channels with sinusoidal walls	93
4.3. Moving boundary simulation	94
5. Conclusions	98
A. Numerical implementation	101
A.1. Temporal discretization	102
A.2. Pressure equation	104
References	105
 III Lattices for derivative-free optimization	107
Abstract	108
1. Introduction	108
2. Favorable properties for packings used by GPS algorithms	111
2.1. Enumeration	111
2.2. Refinement	111
2.3. Uniformity	112
2.4. Neighborhood	112
2.5. Objective of paper	112
3. Notation	113
4. Comparison of the cubic and hexagonal lattices for 2D optimization	115
4.1. Enumeration	116
4.2. Refinement	116
4.3. Uniformity	116
4.4. Neighborhood	119
4.5. Summary of the 2D case	119
5. Comparison of the cubic, FCC, and BCC lattices for 3D optimization	119
5.1. Enumeration	120
5.2. Refinement	122

5.3. Uniformity	122
5.4. Neighborhood	123
5.5. Summary of the 3D case	124
6. Extending the approach to higher dimensions	124
6.1. Exhaustive search strategy	125
6.2. “Force”-based search strategy	126
6.3. Incorporating previous function evaluations	127
7. Comparison of the cubic, checkerboard, and staggered lattices for optimization in n dimensions	128
7.1. Enumeration	128
7.2. Refinement	129
7.3. Uniformity	131
7.4. Neighborhood	132
7.5. Summary of the n -dimensional case	133
8. Convergence tests of GPS optimizations using the alternative lattices	134
9. Concluding remarks	135
A. Tables of optimized stencils of minimal positive bases	139
References	143
Appendix Simulation of near-wall turbulence over a moving wall using the immersed boundary method	
1. Immersed boundary method	144
2. DNS flow model	145
3. Numerical scheme	146
4. Code validation	147
References	148
	152

PREFACE

Substantial energy shortages and rising fuel costs have been driving factors making the development of new technologies for reducing the drag and improving the efficiency of aircraft, ships, and land vehicles more important today than ever before. As near-wall turbulence is responsible for a large fraction of the drag on the surface of such transportation systems, a very significant amount of effort has been directed towards the control and manipulation of boundary layer flow systems, using both passive and active and both open- and closed-loop strategies, in order to both delay laminar-to-turbulent transition and attenuate the intensity of fully-developed turbulence. Recent breakthroughs in the understanding of the dynamics of the coherent structures dominating near-wall turbulence, coupled with tractable extensions of relevant control and optimization theories to PDE systems, rapid growth in computational capabilities, and the technological development of effective devices for small-scale actuation and sensing of flows, have led to significant progress on these challenging problems, though much work remains to be done.

The theoretical problem of the active feedback control of small-scale perturbations in wall-bounded turbulent flow systems using actuation and sensing at the walls has enjoyed great advancements in the last 5 years. However, practical implementation of such turbulence control strategies requires the development of *durable* small-scale actuators and sensors, which are yet to be produced in large numbers.

Passive approaches, on the other hand, require no actuators, sensors, or external energy input, and thus appear to be easier to implement. Passive strategies currently under investigation include polymer additives, riblets, large eddy break-up devices (LEBUs), and compliant surfaces or coatings.

As a passive approach to boundary layer control, compliant surfaces have captured the interest of many researchers for almost half a century, perhaps due to their original inspiration from nature (e.g., dolphin skin). In these investigations, the utility of compliant surfaces for transition delay has been well established, and the use of compliant surfaces for noise mitigation has also been explored. However, the problem of using compliant surfaces to mitigate the intensity of fully-developed turbulence has proved to be a much more elusive target, and thus forms the focus of the present investigation. Leveraging direct numerical simulation techniques developed in the last two decades, this thesis focuses on three issues related to this problem, as summarized below.

Chapter 1 - Modeling and simulation of the interaction of near-wall turbulence

with compliant tensegrity fabrics

To avoid the use of expensive finite-element models for the surface deformations, most researchers have focused their attention primarily on simple spring-supported plates to represent the compliant surface. Though convenient for theoretical analyses, numerical studies have shown that this model has little, if any, effect on the statistics of near-wall turbulence. In contrast, our work focuses on a completely different type of compliant surface model than has been considered previously. By so doing, we hope to find a region in the parameter space defining the surface compliance characteristics that has previously been unreachable by simple plate-spring models. We thus concentrate our attention on an intriguing structural paradigm called tensegrity, the dynamics, optimization, and control of which has been characterized thoroughly by Skelton *et al.* at UC San Diego. Our surface model, which we call a *tensegrity fabric*, is based on this paradigm.

In this chapter, we study the careful numerical characterization of the turbulent flow / tensegrity structure interaction. The turbulence is modeled with direct numerical simulations, as no other computational tools are currently available to capture the precise effects of boundary conditions on a near-wall turbulent flow. An efficient pseudospectral flow solver with a 3D time-dependent coordinate transformation is developed to calculate accurately the chaotic, multiscale behavior of the near-wall turbulent flow system with compliant walls.

We perform a number of simulations of the turbulent flow / tensegrity structure interaction using this solver. Three material parameters that parameterize the surface compliance properties are varied: specifically, the density, stiffness, and damping of the members of the tensegrity fabric, which are all (in these initial tests) taken to have the same material properties. A significant influence of the compliant surface on the statistics of the near-wall turbulence is reported.

Chapter II - On the contravariant form of the Navier-Stokes equations in time-dependent curvilinear coordinate systems

When expressing the Navier-Stokes equation for the flow in moving coordinates, either a contravariant formulation or a Cartesian formulation may be used. We have considered both, and have found that the former needs to be treated very carefully for the time differentiation of the momentum vector to be handled correctly, which some of the most significant published literature on the problem of computing turbulent flows over moving walls and compliant surfaces has failed to do. This chapter draws attention to, and corrects, this error, deriving the correct contravariant

formulation for general moving curvilinear coordinate systems. However, the Cartesian formulation for simulations of the flow over compliant surfaces, as implemented in chapter I, is found to be more efficient computationally, and is thus preferred by our group for this problem.

Chapter III - Lattices for derivative-free optimization

Although drag is increased and turbulence is intensified by the compliant surface in the simulations presented in chapter I, the substantial flexibility of the tensegrity fabric design has not yet been exploited. In the present chapter, we develop an efficient algorithm to optimize within the large parameter space defining the compliance properties of this structure. In this optimization problem, the cost function (which is based on the statistics of a long-time computation of the flow) is both non-differentiable and extremely expensive to evaluate. For these reasons, we have focused on refining and implementing a recently-developed direct search method, called the surrogate management framework (SMF), which is both well suited for such optimization problems and easily parallelized. To enhance the efficiency of the SMF, which is typically based on a Cartesian mesh over the parameter space, we investigate the use of n -dimensional extrapolations of the body-centered cubic (BCC) and face-centered cubic (FCC) crystalline structures as the underlying lattice over the parameter space of interest during the optimization. These lattices allow the trial points to be distributed more uniformly in the parameter space. The new lattices are tested in randomly-generated optimization problems, and are shown to lead to substantial performance improvements in the optimization algorithm.

Appendix - Simulation of near-wall turbulence over a moving wall using the immersed boundary method

To accommodate the interface motion of the flow, we also investigated the potential use of the immersed boundary method in addition to the coordinate transformation method described in chapter I. In the immersed boundary method, the flow domain is extended to immerse the irregular interface, a Cartesian grid is used over this extended domain, and a fictitious force is applied to the “flow” outside the physical part of the flow domain in order to bring the flow to rest at the location of the irregular interface defining the actual flow boundary. In comparison with the coordinate transformation method, the immersed boundary method is fairly simple, as it does not require reconstruction of grid at each time step. However, when applied in a pseudospectral flow simulation

code, as used in the present work, the immersed boundary method was found to facilitate accurately only very small wall deformations. Larger deformations (greater than the wall-normal grid spacing at the wall) typically triggered Gibbs phenomenon, exciting small-scale fluctuations that grew and eventually destabilized the nonlinear simulation. Thus, though this method appears to be useful for the purpose of linear stability analysis, it was eventually abandoned for the purpose of nonlinear flow simulations in present work when wall deformations are relatively large.

Future work

An efficient, parallelized SMF optimization code has been developed with each cost function evaluation being the calculation of the time-averaged drag of a turbulent-flow passing over a compliant tensegrity fabric with the parameters specified for that iteration. The work left (over the next year or two) is simply to obtain a very large allocation of supercomputer time and actually run this optimization code in search of a set of surface compliance parameters that might provide significant drag reduction.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Professor Tom Bewley for his guidance during my studies. His ambition, passion and enthusiasm on research have been encouraging and inspiring. I also want to thank Professor Skelton and his students, Jean-Paul Pinaud, Milenko Masic, Jeongheon Han and Waileung Chan, who collaborated with us and made the compliant surface model possible. Many thanks for the valuable advice from Professor Pozrikidis, as well.

I am grateful to the post-docs who worked in the lab, Peter Blossey, Bartosz Protas, and especially Scott Miller. They have generously offered assistance on many computer administration tasks. I would also like to acknowledge Professor Banerjee and Dr. De Angelis for providing data, and Dr. Marsden for her collaboration on optimization.

In particular, I want to thank my colleague Laura Cerviño. She and I have worked side-by-side, watched over each other, taken all the same classes, and discussed homework together.

My beautiful wife, Heidi, is the most important one on this list. I certainly couldn't have done this work without her love and support.

Finally, I gratefully acknowledge the Dynamics and Control Directorate of the AFOSR, the ONR Young Investigator Program, and the Mechanical and Aerospace Engineering Department for their financial support during the course of this investigation.

The text of Chapter II, in full, is a reprint of the material as it appears in "On the contravariant form of the Navier-Stokes equations in time-dependent curvilinear coordinate systems," *J. Comput. Phys.*, **199** (1), 355–375. The text of Appendix, in part, is a reprint of the material as it appears in "Design, modeling, and optimization of tensegrity compliant surface for reduction of drag induced by the turbulent flow," *SPIE Paper*, 5049-57. The dissertation author was the primary author in these publications.

VITA

Education:

- 1996 B.A., Tsinghua University, Beijing
1999 M.S., Tsinghua University, Beijing
2004 Doctor of Philosophy, University of California, San Diego

Publications & Presentations

- Luo, H. & Bewley, T. R. 2004 On the contravariant form of the Navier-Stokes equation in time-dependent curvilinear coordinate systems. *J. Comput. Phys.* **199** (1), 355-375.
- Bewley, T. R. & Luo, H. 2004 Lattices for derivative-free optimization. Part 1: Cubic (Z^n), Hexagonal (A_2), FCC (D_3), BCC (D_3^*), Checkerboard (D_n), and Staggered (D_n^*). To be submitted to *SIAM J. Optim.*
- Luo, H. & Bewley, T. R. 2004 Modeling and simulation of the interaction of near-wall turbulence with compliant tensegrity fabrics. To be submitted to *J. Fluid Mech.*
- Luo, H. & Bewley, T. R. 2003 Design, modeling, and optimization of tensegrity compliant surface for reduction of drag induced by the turbulent flow. *Smart structures and materials 2003: modeling, signal processing, and control (SPIE proceedings series)*, 3-6 March, San Diego, 5049-57.
- Luo, H & Bewley, T.R. (2003) Optimization of compliant surfaces for reduction of turbulence-induced drag. The APS Division of Fluid Dynamics 56th Annual Meeting, Nov. 23-25, East Rutherford, NJ.
- Luo, H & Bewley, T.R. (2002) Numerical simulation of the turbulent flow over a tensegrity fabric. The APS Division of Fluid Dynamics 55th Annual Meeting, Nov. 24-26, Dallas.
- Luo, H & Bewley, T.R. (2001) On the utility of the global TKE equation for turbulence control. The APS Division of Fluid Dynamics 54th Annual Meeting, Nov. 18-20, San Diego.

Awards & Honors

- The Dissertation Fellowship for Spring quarter, MAE Dept. , UC San Diego, 2004.
- Fellowship from Helmholtz Institute for Supercomputational Physics for the program *First summer school: Tools to Simulate Turbulence on Supercomputers*, Potsdam, Germany, summer 2001.
- First level of Gaotian scholarship for graduate students, Tsinghua University, 1997
- First level of scholarship for Excellent Students, Tsinghua University, 1995.

Others

- Teaching assistant for *Thermodynamics*, MAE Dept. , UC San Diego, winter 2000.

ABSTRACT OF THE DISSERTATION

The Interaction of Near-wall Turbulence with Compliant Tensegrity Fabrics:
Modeling, Simulation, and Optimization

by

Haoxiang Luo

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2004

Professor Thomas R. Bewley, Chair

This work focuses on the modeling, simulation, and optimization of a novel type of compliant surface, a *tensegrity fabric*, for the possible reduction of the drag caused by an overlying turbulent flow.

The spatially-periodic tensegrity fabric is modeled with an extension of the tensegrity dynamics software developed by Skelton *et al.*, who also designed the plate-class tensegrity structures used in this work. To account for the skin friction and pressure forces on the tensegrity structure, a simple tessellation is used.

The spatially-periodic turbulent flow is modeled with direct numerical simulation. To account for the effect of the interface motion on the flow, a 3D time-dependent coordinate transformation is adopted to map the deformed flow domain to a regular domain. When formulating the Navier-Stokes equation governing the flow in moving coordinates, both the contravariant form and the Cartesian form are considered. The former needs special care in the time differentiation of the momentum vector, and is discussed separately in the tensor framework. The latter is computationally less expensive and is thus used in the bulk of our simulations.

A significant influence of the compliant surface on the statistics of the near-wall turbulence is found in simulations at $Re_\tau = 150$, which show that, when the structure's stiffness and damping are low, the interface forms streamwise-traveling waves which significantly increase both drag and turbulent kinetic energy.

To exploit the (yet unexplored) large design flexibility of the tensegrity fabric, we have significantly refined a recently-developed direct search method, called the surrogate management framework (SMF), which is suitable for static optimization problems such as the present, in which

the cost function is both non-differentiable and expensive to evaluate. Our refinements of the SMF focus on the use of n -dimensional extrapolations of the body-centered cubic (BCC) and face-centered cubic (FCC) crystalline structures as the underlying lattice during the optimization, rather than the default Cartesian mesh. These lattices both cover the parameter space and distribute the vectors of the minimal positive basis more uniformly than the Cartesian mesh, thus provide significantly improved convergence when used as the underlying lattice in pattern search algorithms.

Chapter I

Modeling and simulation of the interaction of near-wall turbulence with compliant tensegrity fabrics

This chapter will be submitted as

Luo, H. & Bewley, T. R. 2004 Modeling and simulation of the interaction of near-wall turbulence with compliant tensegrity fabrics. To *J. Fluid Mech.*

Abstract

This paper presents a new class of compliant surfaces, dubbed *tensegrity fabrics*, for the problem of reducing the drag induced by near-wall turbulent flows. The substructure upon which this compliant surface is built is based on the “tensegrity” structural paradigm, and is formed as a stable pretensioned network of compressive members (“bars”) interconnected by tensile members (“tendons”). Compared with existing compliant surface studies, most of which are based on spring-supported plates or membranes, tensegrity fabrics appear to be better configured to respond to the shear stress fluctuations (in addition to the pressure fluctuations) generated by near-wall turbulence. As a result, once the several parameters affecting the compliance characteristics of the structure are tuned appropriately, the tensegrity fabric might exhibit an improved capacity for dampening the fluctuations of near-wall turbulence, thereby reducing drag.

In the present flow simulations, a 3D time-dependent coordinate transformation is used to account for the motion of the channel walls, and the Cartesian components of the velocity are used as the flow variables. For the spatial discretization, a dealiased pseudospectral scheme is used in the homogeneous directions and a second-order finite difference scheme is used in the wall-normal direction. The code is first validated with several benchmark results that are available in the published literature for flows past both stationary and nonstationary walls. Direct numerical simulations of turbulent flows at $Re_\tau = 150$ over the compliant tensegrity fabric are then presented. It is found that, when the stiffness, mass, and damping of the members of the tensegrity fabric are selected appropriately, the near-wall statistics of the turbulence are altered significantly. The flow/structure interface is found to form streamwise-travelling waves reminiscent of those found at air-water interfaces, but traveling at a faster phase velocity. Under certain conditions, the coupled flow/structure system is found to resonate, exhibiting a synchronized, almost sinusoidal interfacial motion with relatively long streamwise correlation. Future work will perform an optimization of the several parameters affecting the compliance of the tensegrity structure in an attempt to exploit the extensive flexibility of this design, and its significant influence on the statistics of the turbulent flow, in search of a fabric with drag-reducing characteristics.

I.1 Introduction

I.1.1 The compliant surface problem

Dominated by so-called “coherent structures” (that is, distinctive vortices with characteristic statistics that evolve in a chaotic fashion), near-wall turbulence is responsible for significant drag penalties in many flows of engineering relevance. Many ideas have been explored in various attempts to attenuate turbulence near walls to improve system efficiency. Among them, the use of compliant surfaces is one of the most attractive, as this approach requires no control inputs and is quite simple in concept: the structure is allowed to flex in response to the fluctuations of the near-wall turbulence, thereby allowing the energy of the turbulent fluctuations to be transmitted into the structure, where it may be damped out. By reducing the intensity of the fluctuations of near-wall turbulence in this manner (if this effect can be realized), presumably the turbulence-induced drag might also be reduced.

However, perhaps due largely to a lack of theoretical insight into the choice of an appropriate compliant material, most experiments to date have failed to establish the hypothesis that turbulence-induced drag can in fact be reduced by compliant surfaces. Comprehensive reviews and comments about the long history of related experiments may be found in Bushnell, Hefner & Ash (1977), Carpenter & Garrad (1985) and Gad-el-Hak (1986, 1987, 1996). Despite several unsuccessful experimental trials, two recent exceptions in the literature are worth noting: Lee, Fisher & Schwarz (1993) observed a significant reduction of turbulent intensity in their experiments of boundary layers over compliant surfaces, and Choi *et al.* (1997) claimed up to 7% drag reduction and up to 5% reduction of turbulent intensity across almost the entire turbulent boundary layer in their experiments, apparently due to the effect of a compliant surface.

Though results are mixed in the fully turbulent regime, compliant surfaces have a well-established capability to delay laminar-to-turbulent transition. This has been studied analytically by many investigators using linear stability theory (Benjamin, 1960; Landahl, 1962; Carpenter & Garrad, 1985; Carpenter & Morris, 1990; Davies & Carpenter, 1997), and has been confirmed in experiments (Daniel *et al.*, 1987; Gaster, 1988). Some tend to believe that the efficient swimming capability of the bottle-nosed dolphin is due, at least in part, to the transition delay caused by its compliant skin (see, e.g., Carpenter, Davies & Lucey, 2000). Unfortunately, the linear stability theory that has shed so much light on the compliant surface problem in the transitional regime fails to provide much useful guidance in the turbulent regime, where we must instead resort to other

tools, such as numerical simulation.

As computers continue to become more powerful and numerical simulation tools continue to become more efficient and accurate, we can begin to address the compliant surface problem in the turbulent regime numerically. Prior work in this area is mostly quite recent. Endo & Himeno (2002) performed a direct numerical simulation of turbulence over a compliant surface and reported approximately 2 to 3% drag reduction. However, their result was soon challenged by Xu, Rempfer & Lumley (2003) for the reason of insufficient averaging time. In the simulations of Xu *et al.*, no drag reduction was found; in fact, wall compliance had no statistically significant effect on the turbulent boundary layer whatsoever. In both papers, the rms extent and velocity of the wall motion, $y_{w,rms}$ and $v_{w,rms}$, are quite small. In Endo & Himeno, $y_{w,rms}^+$ is about 0.008 and $v_{w,rms}^+$ is about 0.025. In Xu *et al.*, we estimate from their figures that $y_{w,rms}^+$ is about 0.05 and $v_{w,rms}^+$ is about 0.01. It is, indeed, improbable that such small wall motions can have a significant influence on the statistics of the turbulence, as the length scales of the energetic motions of the coherent structures are much larger. The models used for the compliant surface in both of these papers are generalizations of the spring-supported thin plate model proposed by Carpenter & Garrad (1985). Governed by a simple linear PDE, this type of model is convenient for theoretical analyses in laminar flows. However, this surface model responds only to the normal load (that is, pressure fluctuations), not to the tangential load (skin friction fluctuations). As skin-friction fluctuations are related to the first-order terms of the Taylor series expansion of the velocity fluctuations near the wall, whereas pressure fluctuations are related to higher-order terms (see, e.g., Bewley & Protas, 2004), one might hypothesize that the surface response to the skin friction fluctuations of the flow should at least be comparable, in some averaged sense, to the surface response to the pressure fluctuations of the flow. In other words, a compliant surface, such as the tensegrity fabric, that can respond to both kinds of loads might present certain advantages. Motivated by this hypothesis, we have focused on tensegrity fabrics exclusively in this work.

I.1.2 Tensegrity fabrics: a new class of compliant surfaces

As mentioned in the abstract, the truss paradigm known as *tensegrity* is a stable pre-tensioned collection of structural members always either under tension (“tendons”) or compression (“bars”). Such structures are often particularly mass efficient for bearing loads. Further, no individual structural member ever experiences bending moments. The same structural paradigm appears to form the molecular foundation for spider fibers (Ingber, 1997, 1998), which is nature’s strongest

material per unit mass (several times stronger per unit mass than steel). In contrast with the spring-supported plate model, the tensegrity fabric is an inherently discrete structure with a relatively involved topology. However, the calculation of the dynamics of the present tensegrity structure is straightforward when framed appropriately, as discussed in detail in §I.5.

Note that, with this work, we are *not* proposing tensegrity as a new model for viscoelastic materials such as a rubber coating. Instead, we are exploring the possible development of an altogether new class of compliant surfaces. If our computations show that this type of compliant surface holds promise for exhibiting drag reducing capabilities, we will then explore the manufacturability of appropriately tuned tensegrity fabrics via extensions of existing textile technologies, incorporating compressive elements into the weave.

I.1.3 DNS models for channel flows with moving boundaries

This work performs a direct numerical simulation of an incompressible turbulent flow in a channel with deformable walls. A time-dependent coordinate transformation is applied to map the deformed channel into a regular domain. Similar simulations that compute the dynamics of incompressible channel flows with moving boundaries can be found in Rosenfeld & Kwak (1991), Carlson, Berkooz & Lumley (1995), Mito & Kasagi (1998), and Fulgosi *et al.* (2003); the rest of this section will review such related simulation codes. A new flow solver was developed from scratch in this work that is substantially different from these several existing codes in one manner or another, thus allowing us to tune the numerical code to be maximally efficient and accurate for the particular problem of interest here.

Rosenfeld & Kwak (1991) presented a solution method for the incompressible Navier-Stokes equations in generalized moving coordinates. Their equations are in a contravariant form derived in the discrete setting using a finite volume discretization. Their method is appropriate for any general moving flow domain, with one of the validation cases being the laminar flow in a non-periodic, two-dimensional channel with an oscillating bump on the wall. In the turbulent channel flow solver by Mito & Kasagi (1998), spanwise wall deformation is applied as a simple active open-loop control strategy to reduce drag. They also formulated their equations in a contravariant form obtained by casting the Cartesian equations into curvilinear coordinates. In their work, a finite difference scheme is used for all spatial derivatives.

In the present work, we employ a pseudospectral method to compute spatial derivatives in the homogeneous directions of the (spatially-periodic) channel flow, which is substantially more

accurate at the resolution used than the finite difference or finite volume methods used by more general flow solvers.

There are other pseudospectral codes for flows with moving interfaces in literature. Carlson *et al.* (1995) developed a pseudospectral code to simulate the flow in a channel with a three-dimensional moving wall, and later applied the code to study the drag of emerging obstacles on the boundary (see Carlson & Lumley, 1996*a,b*). Note that the contravariant formulation of the governing equations used in these works, which is quite involved, is slightly flawed, due to some missing terms in the temporal differentiation of the momentum vector in moving curvilinear coordinates (for further discussion, see Luo & Bewley, 2004).

In comparison, the present formulation is more straightforward. The coordinate transformation in both our work and that of Carlson *et al.* (1995) is nonorthogonal, as (for reasons of simplicity) the grid is deformed in the y direction only. Due to this nonorthogonal coordinate transformation, the contravariant formulation of the Navier-Stokes equation, as used by Carlson *et al.*, is quite involved and expensive to solve. Thus, the present work uses a simpler Cartesian formulation of the governing equation, as described in detail in §2.

Fulgosi *et al.* (2003) performed a pseudospectral direct numerical simulation of turbulence in a sheared air-water flow with a deformable interface. In their work, the two-phase flow is confined to a channel, and the two time-varying subdomains containing each phase are mapped into rectangular domains, on which the transformed Navier-Stokes equation is solved using pseudospectral techniques.

In both Carlson *et al.* (1995) and Fulgosi *et al.* (2003), the momentum and pressure equations are solved using Fourier/Chebyshev/Fourier spatial discretization techniques in the streamwise/wall-normal/spanwise directions. Thus, most of the terms in both formulations must be treated explicitly, except for the directly invertible part of the Laplacian operator. In the present work, we choose a finite difference method to compute the derivatives in the wall-normal direction and solve the momentum equation in physical space. Although this approach is less accurate than a Chebyshev approach, it allows us to treat most of the terms in the governing equation containing wall-normal derivatives implicitly, even when these terms contain non-constant coefficients related to the geometry. Thus, the method used in the present work has a less restrictive CFL constraint on the time step than explicit (Chebyshev-based) codes, at the cost of a requiring increased grid resolution in the wall-normal direction to achieve a desired degree of accuracy.

To accommodate the interface motion of the flow, we also investigated the potential use

of the immersed boundary method (Luo & Bewley, 2003). In the immersed boundary method, the flow domain is extended to immerse the irregular interface, a Cartesian grid is used over the entire extended domain, and a fictitious force is applied to the “flow” outside the physical part of the flow domain in order to bring the flow to rest at the location of the irregular interface defining the actual flow boundary. In comparison with the coordinate transformation method, the immersed boundary method is fairly simple, as the numerical grid remains fixed even as the walls flex. However, when applied in a pseudospectral flow simulation code, as used in the present work, the immersed boundary method was found to facilitate accurately only very small wall deformations. Larger deformations (greater than the wall-normal grid spacing at the wall) typically triggered Gibbs phenomenon, exciting small-scale fluctuations that grew and eventually destabilized the nonlinear simulation. Thus, though this method appears to be useful for the purpose of linear stability analysis, it was eventually abandoned for the purpose of nonlinear flow simulations in present work, in which wall deformations are greater than the wall-normal grid spacing.

I.2 Governing equation

I.2.1 Domain transformation

Defining $\eta_u(x_1, x_3, t)$ and $\eta_l(x_1, x_3, t)$ as the upper and lower wall displacements in the wall-normal direction from the respective nominal positions of the walls ($x_2 = \pm 1$), and defining $\eta_1 = (\eta_u - \eta_l)/2$ and $\eta_0 = (\eta_u + \eta_l)/2$, the following time-dependent coordinate transformation may be used to map the irregular physical domain into a rectangular domain

$$\begin{cases} x_1 = \xi_1 \\ x_2 = \xi_2(1 + \eta_1) + \eta_0 \\ x_3 = \xi_3 \\ t = t, \end{cases} \quad (\text{I.1})$$

where x_i denotes the Cartesian coordinates and ξ_i denotes the curvilinear coordinates. This domain transformation is illustrated in Figure I.1. Note that the code developed in this work allows both walls to deform, as seen in some of the code validation tests reported in §4, though only the lower wall is deformed in the main compliant surface simulations reported in §7, as illustrated in Figure I.1. Note also that we use the variables $\{x, y, z\}$ and $\{x_1, x_2, x_3\}$ interchangeably to denote the stream-wise, wall-normal, and spanwise directions. The nominal domain size is $L_x \times L_y \times L_z$, where L_x and

L_z are identified in the sections to follow and $L_y = 2$.

The incompressible Navier-Stokes equation that governs the flow system is

$$\begin{aligned} \frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} - \delta_{i1} P_x, \\ \frac{\partial u_j}{\partial x_j} &= 0, \end{aligned} \quad (\text{I.2})$$

where ρ is the (constant) density, p is the hydrodynamic pressure, and ν is the kinetic viscosity. Distances x_i are normalized by the half-width of the channel h , velocities u_i are normalized by the mean friction velocity u_τ of the corresponding turbulent channel flow with solid walls, and time t is normalized by h/u_τ . (Note that, where explicitly specified, the scaling is different for some of the code validation tests in §4.) The spatially-uniform pressure gradient in the x_1 direction, P_x , is adjusted in time to maintain constant mass flux in the physical domain. The Jacobian of the spatial transformation and its determinant are

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \\ 0 & 0 & 1 \end{pmatrix}, \quad J = \left| \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right| = \frac{\partial x_2}{\partial \xi_2}.$$

Note that J is independent of ξ_2 . The inverse spatial transformation matrix is

$$\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{J} \frac{\partial x_2}{\partial \xi_1} & \frac{1}{J} & -\frac{1}{J} \frac{\partial x_2}{\partial \xi_3} \\ 0 & 0 & 1 \end{pmatrix}.$$

Defining the nontrivial elements of the transformation as

$$\varphi_i = \frac{\partial \xi_2}{\partial x_i}, \quad \varphi_\tau = \frac{\partial \xi_2}{\partial t},$$

we may apply the substitutions

$$\begin{aligned} \frac{\partial}{\partial t} &\rightarrow \frac{\partial}{\partial t} + \varphi_\tau \frac{\partial}{\partial \xi_2}, \\ \frac{\partial}{\partial x_i} &\rightarrow \frac{\partial}{\partial \xi_i} + \varphi_i \frac{\partial}{\partial \xi_2} \quad \text{for } i = 1, 3, \\ \frac{\partial}{\partial x_i} &\rightarrow \varphi_2 \frac{\partial}{\partial \xi_i} \quad \text{for } i = 2, \end{aligned} \quad (\text{I.3})$$

to express the derivatives in (A.2) in terms of the new coordinates.

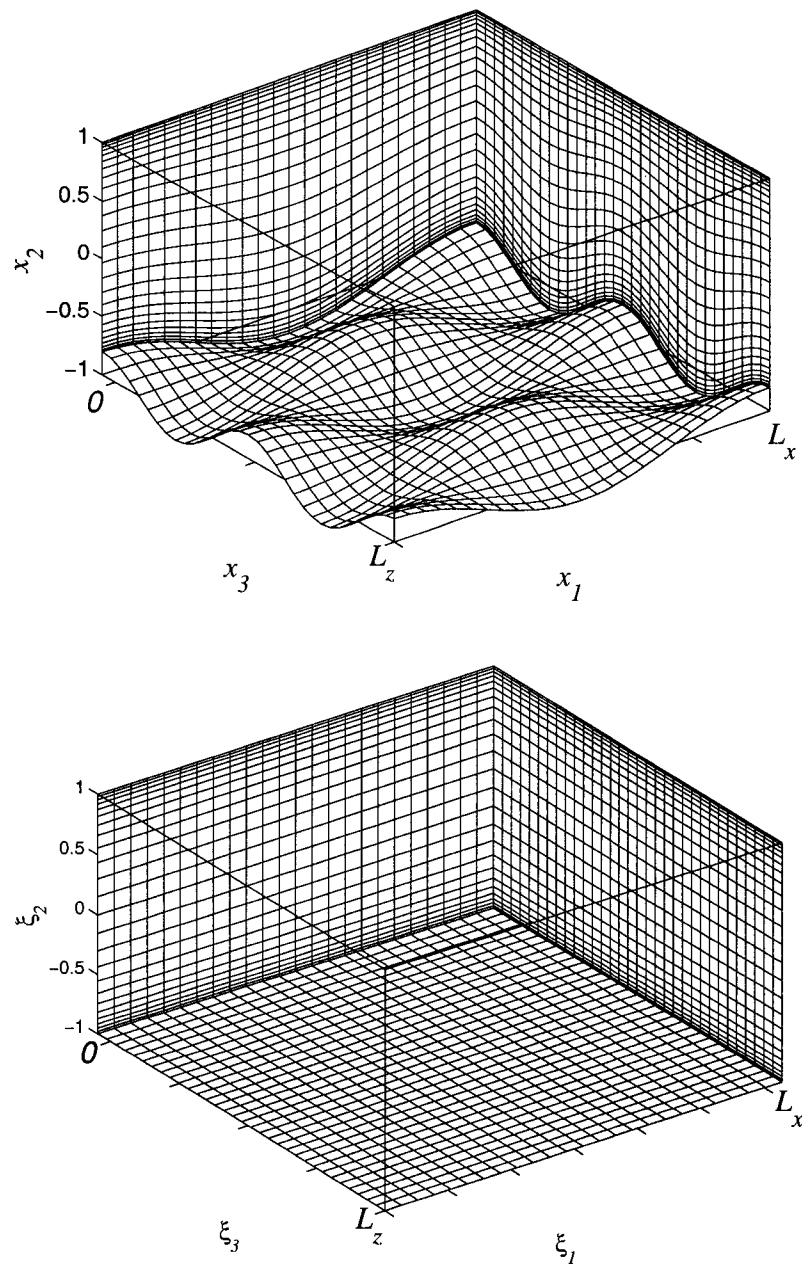


Figure I.1: Domain transformation.

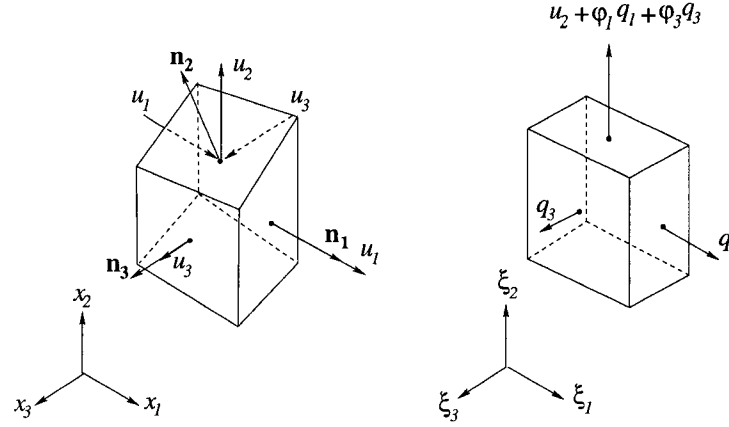


Figure I.2: Cell transformation.

I.2.2 Choice of dependent variables

Velocity vectors in a curvilinear coordinate system may be defined as Cartesian vectors, whose bases are associated with the original ($\mathbf{x}$) coordinates, or contravariant vectors, whose bases are associated with the deformed ($\boldsymbol{\xi}$) coordinates. In either case, multiplying the Navier-Stokes equation by the transformation Jacobian determinant J results in a governing equation in a strong conservation form which is favorable (for the purpose of momentum conservation) in numerical codes. However, using the contravariant form generally involves several additional terms to achieve the correct expression of both the temporal and spatial derivatives in a moving coordinate system; for further discussion of this approach, see Luo & Bewley (2004). The contravariant formulation thus renders the governing equation more involved and expensive to solve even in a relatively simple flow such as the present, in which the transformation only affects the wall-normal coordinate. In this work, therefore, we choose to represent the velocity vectors in Cartesian form, thereby simplifying the computation significantly.

To understand our choice of primitive variables, consider the problem of mass conservation from finite volume point of view. Figure I.2 illustrates the transformation of a single computational cell. In the transformation used in this work, the vertical grid lines are not deformed by the transformation. As a result, among 6 faces of the cell indicated, only top face and bottom face have their surface normals inclined from the corresponding cartesian basis vectors. Horizontal flow (u_1 or u_3 , in the x_1 or x_3 direction) will cause fluid to pass through both of these faces. When considering the conservation of mass in this cell, we may assume that all of its faces are stationary, as the effects

of the cell's motion are taken into account in the geometrical conservation law (see Rosenfeld & Kwak, 1991), which is satisfied by the analytical coordinate transformation (II.34). Thus, in the case of the top face (whose surface normal is $\mathbf{n}_2$), the volume flux out of the cell is

$$\begin{aligned}\mathbf{u} \cdot \mathbf{n}_2 &= \mathbf{u} \cdot \left(\frac{\partial \mathbf{x}}{\partial \xi_3} d\xi_3 \times \frac{\partial \mathbf{x}}{\partial \xi_1} d\xi_1 \right) = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \cdot \begin{pmatrix} -\frac{\partial x_2}{\partial \xi_1} \\ 1 \\ -\frac{\partial x_2}{\partial \xi_3} \end{pmatrix} d\xi_3 d\xi_1 = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \cdot \begin{pmatrix} J\varphi_1 \\ 1 \\ J\varphi_3 \end{pmatrix} d\xi_3 d\xi_1 \\ &= (\varphi_1 Ju_1 + u_2 + \varphi_3 Ju_3) d\xi_3 d\xi_1,\end{aligned}$$

where $\times$ denotes the cross product. Similarly, the volume flux through the positive x_1 and x_3 faces are

$$\begin{aligned}\mathbf{u} \cdot \mathbf{n}_1 &= \mathbf{u} \cdot \left(\frac{\partial \mathbf{x}}{\partial \xi_2} d\xi_2 \times \frac{\partial \mathbf{x}}{\partial \xi_3} d\xi_3 \right) = \frac{\partial x_2}{\partial \xi_2} u_1 d\xi_2 d\xi_3 = Ju_1 d\xi_2 d\xi_3, \\ \mathbf{u} \cdot \mathbf{n}_3 &= \mathbf{u} \cdot \left(\frac{\partial \mathbf{x}}{\partial \xi_1} d\xi_1 \times \frac{\partial \mathbf{x}}{\partial \xi_2} d\xi_2 \right) = \frac{\partial x_2}{\partial \xi_2} u_3 d\xi_1 d\xi_2 = Ju_3 d\xi_1 d\xi_2.\end{aligned}$$

To summarize, the volume flux through these three faces are Ju_1 in the ξ_1 direction, $u_2 + \varphi_1 Ju_1 + \varphi_3 Ju_3$ in the ξ_2 direction, and Ju_3 in the ξ_3 direction; the volume flux through the opposite three faces are analogous. Note that these three components form a vector which equals to the contravariant velocity vector in the curvilinear coordinates multiplied by the volume dilatation factor J . To avoid using such a contravariant vector for the reason of simplicity as described previously, and additionally avoiding the repeated application of this volume dilatation factor in the numerical code, we define the following Cartesian vector

$$\begin{aligned}q_1 &= Ju_1 \\ q_2 &= u_2 \\ q_3 &= Ju_3\end{aligned}\tag{I.4}$$

and the modified pressure

$$\tilde{p} = Jp/\rho\tag{I.5}$$

as the primitive variables in our numerical code. As a consequence, the q_1 - and q_3 -momentum equations are represented in strong conservation form due to their weighting by J . Requiring the net mass flux into each cell be zero, we may write the continuity equation for this system as

$$D(q_i) = \frac{\partial q_1}{\partial \xi_1} + \frac{\partial (\varphi_1 q_1 + q_2 + \varphi_3 q_3)}{\partial \xi_2} + \frac{\partial q_3}{\partial \xi_3} = 0,\tag{I.6}$$

where $D(\cdot)$ denotes the divergence operator.

With the aforementioned chain rules (I.3), the momentum equation in the new coordinates may be written as

$$\frac{\partial q_i}{\partial t} + T_i(q_i) + N_i(q_j) = -G_i(\tilde{p}) + \nu L_i(q_i) - JP_x \delta_{i1}, \quad (\text{I.7})$$

where $T_i(q_i)$ is the term associated with the motion of the coordinates, $N_i(q_j)$ is the convection term, $G_i(\tilde{p})$ is the pressure term, and $L_i(q_i)$ is the viscous term. All four of these terms are written out in full in Appendix I.A.

To maintain a constant bulk velocity U_{bulk} , the necessary spatially-uniform streamwise pressure gradient P_x is computed by integrating the u_1 momentum equation over the entire physical domain,

$$\begin{aligned} U_{bulk} &= \frac{1}{2L_x L_z} \int_0^{L_x} \int_0^{L_z} \int_{-1+\eta_l}^{1+\eta_u} u_1 dx_2 dx_3 dx_1 = \text{constant} \\ \Rightarrow P_x &= \frac{1}{2 \iint J d\xi_1 d\xi_3} \iint \left[-\varphi_1 \tilde{p} + \nu \varphi_1 \frac{\partial \varphi_1 q_1}{\partial \xi_2} + \nu \varphi_2^2 \frac{\partial q_1}{\partial \xi_2} + \nu \varphi_3 \frac{\partial \varphi_3 q_1}{\partial \xi_2} \right]_{\xi_2=-1}^{\xi_2=1} d\xi_1 d\xi_3. \end{aligned} \quad (\text{I.8})$$

I.2.3 Boundary conditions

Periodic boundary conditions are assumed in the streamwise direction (ξ_1) and spanwise direction (ξ_3) for the dependent variables, $\{\mathbf{q}, \tilde{p}\}$, and the wall deformation functions, $\{\eta_u, \eta_l\}$. For most of the cases (unless explicitly specified otherwise), no-slip and no-penetration boundary conditions are assumed at the two walls,

$$\begin{aligned} q_1 &= q_3 = 0, & \text{at } \xi_2 &= \pm 1, \\ q_2 &= \frac{\partial \eta_u}{\partial t}, & \text{at } \xi_2 &= +1, \\ q_2 &= \frac{\partial \eta_l}{\partial t}, & \text{at } \xi_2 &= -1. \end{aligned}$$

I.3 Numerical algorithm

The numerical algorithm used for solving the unsteady incompressible Navier-Stokes equation in this work is based closely on that in Bewley, Moin & Temam (2001), in which the flow in a rectangular channel is controlled by unsteady wall-normal blowing and suction. As in that work, a hybrid pseudospectral / finite-difference method is used for the spatial discretization, and a mixed Crank-Nicolson (CN) / 3rd-order Runge-Kutta (RK) method is used for the temporal discretization. Several additional considerations are necessary in present work, however, as the governing equation is considerably more involved due to the coordinate transformation that accounts

for the moving boundaries. In addition to the flow variables, the geometry-related time-varying coefficients, φ_i and φ_τ , need to be spatially discretized. Further, not all terms involving derivatives in the wall-normal (ξ_2) direction can be treated implicitly, as was possible in Bewley *et al.* (2001). For example, the terms with cross derivatives in the wall-normal direction and in one of the homogeneous directions (ξ_1 or ξ_3) must be treated explicitly. In addition, since the various Fourier modes can not be fully decoupled in the evaluation of the Laplacian, the pressure equation needs to be solved iteratively, subject to a boundary condition that is derived by imposing the incompressibility condition at the (possibly moving) walls.

I.3.1 Spatial discretization

The grid is chosen to be evenly spaced and non-staggered in the streamwise direction (ξ_1) and the spanwise direction (ξ_3) so that spectral techniques may be used to compute spatial derivatives in these directions. In the wall-normal direction (ξ_2), the grid is staggered and stretched using a hyperbolic tangent function, facilitating the grid refinement necessary to resolve the near-wall region accurately (see Figure I.1). The variables q_2 , φ_i , and φ_τ are discretized on the family of gridpoints $j = 0, 1, 2, \dots, NY$, where $j = 0$ corresponds to the lower wall and $j = NY$ corresponds to the upper wall. The variables q_1 , q_3 , and $\bar{p}$, on the other hand, are discretized on the family of gridpoints $j = \frac{1}{2}, 1 + \frac{1}{2}, 2 + \frac{1}{2}, \dots, NY - \frac{1}{2}$, where the gridpoint $j = n - \frac{1}{2}$ is midway between the gridpoints $j = n - 1$ and $j = n$, i.e.,

$$\xi_{2,n-\frac{1}{2}} = \frac{1}{2} (\xi_{2,n} + \xi_{2,n-1}).$$

Correspondingly, two grid spacing families are defined as

$$\begin{aligned} \Delta \xi_{2,n-\frac{1}{2}} &= \xi_{2,n} - \xi_{2,n-1}, \\ \Delta \xi_{2,n} &= \xi_{2,n+\frac{1}{2}} - \xi_{2,n-\frac{1}{2}} = \frac{1}{2} (\Delta \xi_{2,n+\frac{1}{2}} + \Delta \xi_{2,n-\frac{1}{2}}). \end{aligned}$$

Second-order accurate linear interpolation is used to interpolate a variable a discretized on the n family of gridpoints onto the $n - \frac{1}{2}$ family,

$$\bar{a}_{n-\frac{1}{2}} = \frac{1}{2} (a_n + a_{n-1}),$$

and to interpolate a variable b discretized on the $n - \frac{1}{2}$ family onto the n family,

$$\bar{b}_n = \frac{1}{2\Delta \xi_{2,n}} (\Delta \xi_{2,n-\frac{1}{2}} b_{2,n+\frac{1}{2}} + \Delta \xi_{2,n+\frac{1}{2}} b_{2,n-\frac{1}{2}}).$$

With the arrangement of the discrete variables on the numerical grid as described above, the three components of the momentum equation are enforced at the gridpoints at which the corresponding velocity components are discretized, and the continuity equation is solved on the cells centered at the pressure points. We use the notation $\frac{\delta}{\delta \xi_i}$ to denote a numerical approximation of differentiation in the ξ_i direction. All derivatives in the ξ_1 and ξ_3 directions, with corresponding wavenumbers k_x and k_z , are computed in Fourier space, i.e., $\frac{\delta \hat{a}}{\delta \xi_1} = ik_x \hat{a}$ and $\frac{\delta \hat{a}}{\delta \xi_3} = ik_z \hat{a}$, where $i = \sqrt{-1}$, the hat (^) indicates the Fourier transform in the appropriate direction(s), and the subscript s is used to emphasize that the derivative is computed spectrally. The derivatives in the ξ_2 direction are computed with quasi-second-order accurate centered finite difference formulae which approach second order accuracy as the grid is refined when the stretching function is held constant. Specifically, the terms in the equation (I.7) are discretized as follows, where the rotation indices are $i = 1, 3$ and $j = 1, 3$ and summation notation is implied:

- The terms associated with the motion of the coordinates:

$$T_{i,n-\frac{1}{2}} = \frac{(\varphi_\tau \bar{q}_i)_n - (\varphi_\tau \bar{q}_i)_{n-1}}{\Delta \xi_{2,n-\frac{1}{2}}},$$

$$T_{2,n} = \varphi_{\tau,n} \frac{q_{2,n+1} - q_{2,n-1}}{2 \Delta \xi_{2,n}}.$$

- The convection terms:

$$N_{i,n-\frac{1}{2}} = \frac{\delta_s(\varphi_2 q_i q_j)_{n-\frac{1}{2}}}{\delta \xi_j} + \varphi_2 \frac{(\varphi_j \bar{q}_i \bar{q}_j)_n - (\varphi_j \bar{q}_i \bar{q}_j)_{n-1}}{\Delta \xi_{2,n-\frac{1}{2}}} + \varphi_2 \frac{(\bar{q}_i q_2)_n - (\bar{q}_i q_2)_{n-1}}{\Delta \xi_{2,n-\frac{1}{2}}},$$

$$N_{2,n} = \frac{\delta_s(\varphi_2 q_2 \bar{q}_j)_n}{\delta \xi_j} + \varphi_2 \varphi_{j,n} \frac{(\bar{q}_2 q_j)_{n+\frac{1}{2}} - (\bar{q}_2 q_j)_{n-\frac{1}{2}}}{\Delta \xi_{2,n}} + \varphi_2 \frac{(\bar{q}_2 \bar{q}_2)_{n+\frac{1}{2}} - (\bar{q}_2 \bar{q}_2)_{n-\frac{1}{2}}}{\Delta \xi_{2,n}}.$$

- The pressure gradient terms:

$$G_{i,n-\frac{1}{2}} = \frac{\delta_s \bar{p}_{n-\frac{1}{2}}}{\delta \xi_i} + \frac{(\varphi_i \bar{p})_n - (\varphi_i \bar{p})_{n-1}}{\Delta \xi_{2,n-\frac{1}{2}}},$$

$$G_{2,n} = \varphi_2^2 \frac{\bar{p}_{n+\frac{1}{2}} - \bar{p}_{n-\frac{1}{2}}}{\Delta \xi_{2,n}}.$$

- The viscous terms:

$$\begin{aligned}
L_{i,n-\frac{1}{2}} &= \frac{\delta_s}{\delta \xi_j} \left[\frac{\delta_s q_{i,n-\frac{1}{2}}}{\delta \xi_j} + \frac{(\varphi_j \bar{q}_i)_n - (\varphi_j \bar{q}_i)_{n-1}}{\Delta \xi_{2,n-\frac{1}{2}}} \right] \\
&+ \frac{1}{\Delta \xi_{2,n-\frac{1}{2}}} \left[\varphi_{j,n} \frac{\delta_s \bar{q}_{i,n}}{\delta \xi_j} - \varphi_{j,n-1} \frac{\delta_s \bar{q}_{i,n-1}}{\delta \xi_j} \right] \\
&+ \frac{1}{\Delta \xi_{2,n-\frac{1}{2}}} \left[\varphi_{j,n} \frac{(\bar{\varphi}_j q_i)_{n+\frac{1}{2}} - (\bar{\varphi}_j q_i)_{n-\frac{1}{2}}}{\Delta \xi_{2,n}} - \varphi_{j,n-1} \frac{(\bar{\varphi}_j q_i)_{n-\frac{1}{2}} - (\bar{\varphi}_j q_i)_{n-\frac{3}{2}}}{\Delta \xi_{2,n-1}} \right] \\
&+ \frac{\varphi_j^2}{\Delta \xi_{2,n-\frac{1}{2}}} \left[\frac{q_{i,n+\frac{1}{2}} - q_{i,n-\frac{1}{2}}}{\Delta \xi_{2,n}} - \frac{q_{i,n-\frac{1}{2}} - q_{i,n-\frac{3}{2}}}{\Delta \xi_{2,n-1}} \right], \\
L_{2,n} &= \frac{\delta_s}{\delta \xi_j} \left[\frac{\delta_s q_{2,n}}{\delta \xi_j} + \varphi_{j,n} \frac{q_{2,n+1} - q_{2,n-1}}{2 \Delta \xi_{2,n}} \right] \\
&+ \frac{\varphi_{j,n}}{\Delta \xi_{2,n}} \left[\frac{\delta_s \bar{q}_{2,n+\frac{1}{2}}}{\delta \xi_j} - \frac{\delta_s \bar{q}_{2,n-\frac{1}{2}}}{\delta \xi_j} \right] \\
&+ \frac{\varphi_{j,n}}{\Delta \xi_{2,n}} \left[\bar{\varphi}_{j,n+\frac{1}{2}} \frac{q_{2,n+1} - q_{2,n}}{\Delta \xi_{2,n+\frac{1}{2}}} - \bar{\varphi}_{j,n-\frac{1}{2}} \frac{q_{2,n} - q_{2,n-1}}{\Delta \xi_{2,n-\frac{1}{2}}} \right] \\
&+ \frac{\varphi_j^2}{\Delta \xi_{2,n}} \left[\frac{q_{2,n+1} - q_{2,n}}{\Delta \xi_{2,n+\frac{1}{2}}} - \frac{q_{2,n} - q_{2,n-1}}{\Delta \xi_{2,n-\frac{1}{2}}} \right].
\end{aligned}$$

Note that, in the expression for $L_{i,n-\frac{1}{2}}$ (for $i = 1, 3$), one-sided finite differences are substituted to approximate the first order derivatives at each wall.

The discretization of the continuity equation (II.26) is

$$D(q_i)_{n-\frac{1}{2}} = \frac{\delta_s q_{1,n-\frac{1}{2}}}{\delta \xi_1} + \frac{(\varphi_1 \bar{q}_1 + q_2 + \varphi_3 \bar{q}_3)_n - (\varphi_1 \bar{q}_1 + q_2 + \varphi_3 \bar{q}_3)_{n-1}}{\Delta \xi_{2,n-\frac{1}{2}}} + \frac{\delta_s q_{3,n-\frac{1}{2}}}{\delta \xi_3}. \quad (\text{I.9})$$

For consistency, the discrete version of the Laplacian operator (I.27) is obtained by taking the discrete divergence of the discretized pressure gradient. That is,

$$\begin{aligned}
\mathcal{L}(\bar{p})_{n-\frac{1}{2}} &= D(G_i)_{n-\frac{1}{2}} \\
&= \frac{\delta_s G_{1,n-\frac{1}{2}}}{\delta \xi_1} + \frac{(\varphi_1 \bar{G}_1 + G_2 + \varphi_3 \bar{G}_3)_n - (\varphi_1 \bar{G}_1 + G_2 + \varphi_3 \bar{G}_3)_{n-1}}{\Delta \xi_{2,n-\frac{1}{2}}} + \frac{\delta_s G_{3,n-\frac{1}{2}}}{\delta \xi_3}. \quad (\text{I.10})
\end{aligned}$$

Note that, although the continuous form of the Laplacian operator in the Poisson equation for the pressure is the same as the continuous form of the Laplacian operator in the streamwise and spanwise momentum equations, $L_1(\cdot)$ and $L_3(\cdot)$, their discretizations as defined above are different. The discretized Laplacian operator in the pressure equation involves 5 pressure nodes in the wall-normal direction, whereas the discretized Laplacian operator in the momentum equation involves only 3 nodes.

I.3.2 Temporal discretization

A low storage 3rd-order Runge-Kutta (RK) scheme is used to advance the flow from time step m to the next step $m + 1$. At each RK substep k ($k = 1, 2$, or 3), a fractional-step method (Akselvoll & Moin, 1995; Bewley *et al.*, 2001) is used to march the flow and project the flow field onto a divergence-free subspace. In this paper, we use superscript k to indicate the intermediate RK time level, not the full time step. Therefore, $\mathbf{q}^{k=0}$ corresponds to $\mathbf{q}$ at time step m , and $\mathbf{q}^{k=3}$ corresponds to $\mathbf{q}$ at the next time step $m + 1$.

In the fractional-step method used, on each RK step an intermediate flow field, $\mathbf{q}^*$, which is non-divergence-free, is first solved with the pressure accounted for explicitly. Other terms are split into two groups — one group is computed explicitly, the other implicitly. This is done to relax the time step restriction related to the stability of explicit codes while maintaining accuracy. Let the operator A_i represent the terms treated explicitly and B_i represent the terms treated implicitly in the Navier-Stokes equation, and let Δt denote the time interval for a full timestep (that is, 3 RK substeps). The first step of the fractional-step method at each RK substep may then be written

$$\begin{aligned} \frac{q_i^* - q_i^{k-1}}{\Delta t} = & \beta_k \left(B_i^k(q_j^*) + B_i^{k-1}(q_j^{k-1}) \right) + \gamma_k A_i^{k-1}(q_j^{k-1}) + \zeta_k A_i^{k-2}(q_j^{k-2}) \\ & + 2\beta_k \left(-G_i^k(\bar{p}^{k-1}) - \delta_{1i} J^k P_x \right), \end{aligned} \quad (\text{I.11})$$

where β_k , γ_k , and ζ_k are the RK coefficients, and the operators are given in detail in Appendix I.B. Note that when $k = 1$, $\mathbf{q}^*$ is computed using current flow field information $\mathbf{q}^{k=0}$ only, since $\zeta_1 = 0$; we thus refer to this as a self-starting scheme.

Some nonlinear terms in (I.11), while involving derivatives in the wall-normal direction only (such as $\frac{\delta \varphi_3 \varphi_2 q_1 q_3}{\delta \xi_2}$ in the q_1 -momentum equation, $\varphi_1 \frac{\delta \varphi_2 q_1 q_2}{\delta \xi_2}$ and $\varphi_3 \frac{\delta \varphi_2 q_2 q_3}{\delta \xi_2}$ in the q_2 -momentum equation, and $\frac{\delta \varphi_1 \varphi_2 q_1 q_3}{\delta \xi_2}$ in the q_3 -momentum equation), are nevertheless treated explicitly. This is done to decouple the three momentum equations, which are first solved for q_2^* , then for q_1^* , and finally for q_3^* . If the wall deformation is small (i.e., φ_1 and φ_3 are near zero), explicit treatment of these terms will not affect significantly the time step restriction. Other nonlinear wall-normal derivatives may be linearized (subject to second order error) according to:

$$q_j^* q_j^* \approx 2q_j^* q_j^{k-1} - q_j^{k-1} q_j^{k-1}.$$

Further details of solving (I.11) may be found in Appendix I.B.

Since the intermediate velocity field, $\mathbf{q}^*$, is a second order approximation of the solenoidal velocity field $\mathbf{q}^k$ (Akselvoll & Moin, 1995), we may use the same boundary conditions for both $\mathbf{q}^*$

and $\mathbf{q}^k$, simplifying the numerical code. These boundary conditions are obtained by simple linear interpolation onto the RK substep k of the boundary conditions at two consecutive time steps.

The second step of the fractional-step algorithm at each RK substep is to solve a projection equation for the pressure update $\phi = \tilde{p}^k - \tilde{p}^{k-1}$ based on the divergence of the intermediate field:

$$\mathcal{L}^k(\phi) = \frac{1}{2\beta_k \Delta t} D^k(q_i^*),$$

where the discrete form of the Laplacian operator is given by (I.10). This projection is used to make the velocity field divergence free. The Poisson equation is solved in Fourier space. Note that the product of the non-constant coefficients φ_i and the pressure update ϕ in physical space produces convolution sums in the Fourier space, coupling the equation at all wavenumber pairs and making the Fourier transform of the Laplacian difficult to invert. We thus split the Laplacian operator into two parts. The terms with the various Fourier modes decoupled are treated implicitly, and the rest are treated explicitly. The equation is then solved iteratively,

$$\tilde{\mathcal{L}}(\phi^s) = \frac{\delta}{\delta \xi_i} \left(\frac{\delta \phi^s}{\delta \xi_i} \right) = RHS^{s-1} = \frac{1}{2\beta_k \Delta t} D^k(q_i^*) - (\mathcal{L}^k - \tilde{\mathcal{L}})(\phi^{s-1}), \quad (\text{I.12})$$

where s is the iteration index. The equation converges quickly when the wall deformation is small. In present work, it typically takes 10 iterations or less for the accuracy to be satisfactory. After ϕ converges, the flow field and pressure are updated by

$$q_i^k = q_i^* - 2\beta_k \Delta t G_i^k(\phi) \quad (\text{I.13})$$

and

$$\tilde{p}^k = \tilde{p}^{k-1} + \phi. \quad (\text{I.14})$$

I.3.3 Pressure equation

In present problem, the boundary velocity is unsteady. At the beginning of each timestep, $k = 0$, the desired boundary condition on q_2 of the end of the time step is either prescribed (as in some of the test problems in §I.4) or obtained by marching the evolution equation for the compliant surface over a single timestep (based on the current skin friction and pressure forces applied by the flow), and is then linearly interpolated onto the three RK substeps to obtain the boundary conditions for each substep. Thus, the time rate of change of the fluid velocity at the wall (that is, the acceleration of the wall), $\frac{\partial \mathbf{q}}{\partial t}|_{wall}$, which is approximated with a simple two-point finite difference, is held constant for RK substeps $k = 2$ and $k = 3$, but changes at $k = 1$. Consequently, there is

a change in the boundary condition for the pressure at the beginning of each timestep. In order for $\mathbf{q}^*$ to be an accurate approximation of $\mathbf{q}^{k=1}$, we update the pressure at the beginning of each timestep to accommodate this change in the boundary condition on the pressure, before marching over first RK substep. To do this, we could solve a full pressure Poisson equation at the beginning of each timestep using a consistent discretization (that is, imposing the discrete divergence operator on the discrete momentum equation, as described in Luo & Bewley (2004)). However, for the current problem with a 3D time-dependent coordinate transformation, such a Poisson equation has a cumbersome right-hand-side forcing term which is expensive to compute. Thus, we instead solve an additional projection equation, subject to zero forcing on the interior but a non-homogeneous boundary condition, and perform an additional pressure update at the beginning of each timestep.

Orszag, Israeli & Deville (1986) and Gresho & Sani (1987) have shown that the Neumann boundary condition implied by projection of the momentum equation onto the surface normal vector $\mathbf{n}_2$, i.e.,

$$\nabla p \cdot \mathbf{n}_2 = \rho \left(-\frac{\partial \mathbf{u}}{\partial t} - \mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u} \right) \cdot \mathbf{n}_2, \quad (\text{I.15})$$

is an appropriate boundary condition for the pressure equation in this type of problem.

To apply this boundary condition, we first derive the discrete form of (I.15) for $\nabla p \cdot \mathbf{n}_2$ by imposing the discrete divergence operator on the discrete momentum equation at the boundaries and applying the incompressibility condition. We then calculate the change of $\nabla p \cdot \mathbf{n}_2$ at the boundaries induced by a change in the wall velocity (which happens at the beginning of each timestep), and use this change to determine the boundary condition on an additional pressure update equation (a Laplace equation with inhomogeneous boundary conditions) solved at the beginning of each timestep. (As the velocity field is already divergence free at this point, the RHS forcing on the interior of this equation is zero.) As this approach is both new and a significant factor in the accuracy of the resulting code, though it is somewhat involved, the relevant equations are described in detail in Appendix I.C.

I.4 Code validation

We now apply the code discussed in the previous section to reproduce several existing results in the published literature for both laminar and turbulent channels flows with both stationary and moving walls in order to validate the accuracy of the new code.

I.4.1 Laminar flow through a sinusoidal channel

In the first test, we consider the laminar flow in a two-dimensional symmetric channel whose walls are sinusoidally deformed in the streamwise direction, that is,

$$\eta_l = -\eta_u = \varepsilon \cos(x_1),$$

where $\varepsilon \ll 1$ is the deformation amplitude. With the domain transformation used by the present code, this flow is solved in a fixed curvilinear coordinate system. This problem has been solved analytically for Reynolds numbers far above that of creeping flow in Tsangaris & Leiter (1984) using a perturbation method to derive an asymptotic solution via a series expansion in ε of the stream function, truncated to the first-order terms. This result was reproduced numerically in Luo & Bewley (2004) using a 2D contravariant formulation of the Navier-Stokes equation, showing fairly close agreement with the asymptotic solution of Tsangaris & Leiter for deformation amplitudes up to $\varepsilon = 0.2$ and Reynolds numbers up to $Re = 400$.

In the present section, we attempt to reproduce the asymptotic flow solution result using the new code (based on a 3D Cartesian formulation), at both low Reynolds numbers ($Re \leq 400$) and small deformation amplitude ($\varepsilon = 0.1$), for which the asymptotic solution of Tsangaris & Leiter (1984) and the numerical solution of Luo & Bewley (2004) match almost exactly.

The flow parameters tested are listed in Table I.1, which are the same as those in the first test case of Luo & Bewley (2004), except that we now use a 3D grid (albeit very coarse in the spanwise direction). Note that, in this subsection, the Reynolds number is based on the centerline velocity U_c and channel half width h , that is, $Re = U_c h / \nu$, and all velocities are normalized by U_c and distances normalized by h . Figure I.4.1 shows the comparisons of the velocity profiles for u_1 and u_2 . It can be seen that the numerical simulations agree very well with the asymptotic results when Re is small, with very slight discrepancies becoming evident as Re is increased. Further, the discrepancies with the asymptotic result are almost identical to those exhibited by the numerical solution computed in Luo & Bewley (2004) using a (completely different) 2D contravariant formulation, indicating that

Flow parameters	Domain size	No. Fourier modes (dealiased collocation points)
$Re = 1, 10, 75, 200, 400 (\frac{U_c h}{\nu})$	$L_x = 2\pi$	$N_x = 32(48)$
$\epsilon = 0.1$	$L_y = 2$	$N_y = 64$
	$L_z = \pi$	$N_z = 4(6)$

Table I.1: Simulation parameters for the laminar flows through a sinusoidal channel.

these discrepancies are in fact due primarily to the breaking down of the validity of the asymptotic solution, not an inaccuracy of the numerical solution.

For small Reynolds numbers, the flows are dominated by viscous effects, and the velocity profiles are smooth, almost parabolic curves, as shown in Figure I.4.1. For $Re = 1$, the vertical velocity is near zero at the throat of the channel ($x_1 = 0$), indicating that the flow in the converging section of the (periodic in x_1) channel loses almost all of its vertical momentum by the time it reaches the nozzle throat. As the Reynolds number is increased, inertial effects become more significant, and not all of the vertical momentum is lost by the time the flow reaches the throat, leading to slightly negative values of u_2 at $x_1 = 0$ (for $x_2 > 0$) as shown. For larger Re , the extremum of the u_2 -profiles shift toward the wall due to heightened inertial effects. Further, in the diverging parts of the channel, the u_1 -profiles are less steep near the walls, indicating that separation is imminent; see Luo & Bewley (2004) for additional plots of this flow in the separated regime.

I.4.2 Three-dimensional laminar flow over a bump

In the second test, we consider a channel flow passing over a 3D Gaussian bump on the center of the lower wall, as depicted in Figure I.4. A similar flow was originally simulated by Mason & Morton (1987) using a Blasius boundary layer as the mean flow and a bump defined using a cosine function. Carlson *et al.* (1995) introduced a slightly modified test flow to validate their pseudospectral channel flow code. In both Carlson *et al.* (1995) and the present tests, the undisturbed flow profile is taken to be a laminar channel flow, and the bump is defined by a Gaussian function, which is a bit smoother than the cosine function used by Mason & Morton (1987). The bump is defined by

$$\eta_l(x_1, x_3) = \epsilon \exp \left[-\frac{(x_1 - \frac{L_x}{2})^2 + (x_3 - \frac{L_z}{2})^2}{\sigma^2} \right].$$

The parameters used for the simulation are listed in Table I.2. The flow is initialized at $t = 0$ as laminar channel flow with the bump absent. For $t > 0$, the bump starts to grow in amplitude until it reaches its final shape at $t = 5h/U_c$, during which time fluid is extracted over the upper wall

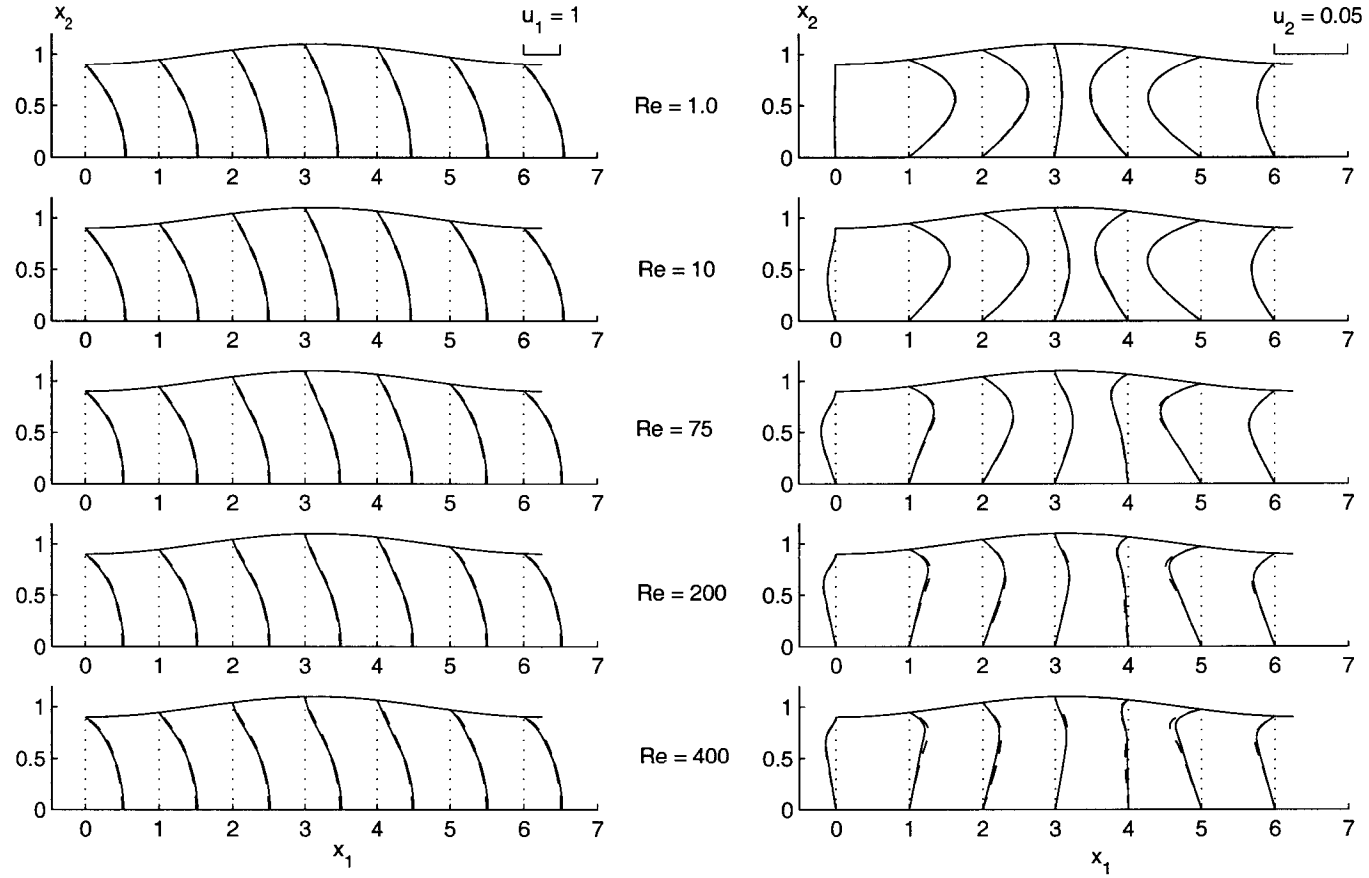


Figure I.3: Velocity profiles (normalized by U_c) at various cross sections ($x_1 = 0, 1, 2, 3, 4, 5, 6$) of the channel for steady flow with $\varepsilon = 0.1$ and various Reynolds number. Left: u_1 component; right: u_2 component. Solid: numerical results (using the present 3D Cartesian formulation); dashed: perturbation analysis.

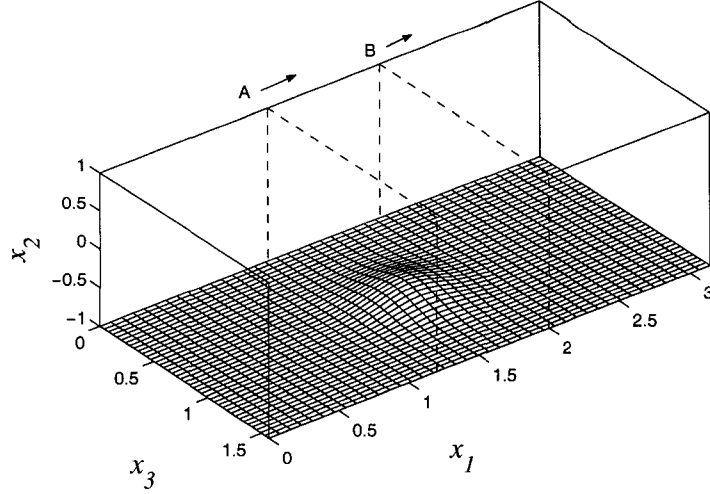


Figure I.4: Domain of the laminar flow over a 3D Gaussian bump.

Flow parameters	Domain size	No. Fourier modes (dealiased collocation points)
$Re = 2083 (\frac{U_c h}{\nu})$	$L_x = \pi$	$N_x = 64 (96)$
$\varepsilon = 0.12$	$L_y = 2$	$N_y = 95$
$\sigma = 0.18$	$L_z = \frac{\pi}{2}$	$N_z = 32 (48)$

Table I.2: Simulation parameters for the laminar flow over a 3D obstacle.

at the appropriate rate to maintain incompressibility. All visualizations reported in this section are taken at $t = 30h/U_c$.

Two cross flow planes located at $x_1 = 1.28$ and $x_1 = 1.87$ (see Figure I.4), where the bump height is diminished to about 0.07ε , are chosen to study the flow behavior upstream and downstream of the bump. The flow patterns and streamwise vorticity in these two planes are shown in Figures I.5 and I.6. On the upstream side, the approaching flow is diverted to the two sides and lifted by the bump; on the downstream side, the flow converges to fill the region behind the bump. Qualitatively, the streamwise vorticity distributions in these two planes are almost identical to those reported by Carlson *et al.* and Mason & Morton. Normalized by the local length scale ε and the local velocity U_l , defined as the mean velocity of the laminar flow that would exist at a distance ε away from the wall in the absence of the bump and given in laminar channel flow by

$$U_l = \frac{U_c}{\varepsilon} \int_{-1}^{-1+\varepsilon} (1 + x_2^2) dx_2 \approx 0.12U_c,$$

the maxima/minima of the streamwise vorticity are located at the wall and given by $\pm 0.99U_l/\varepsilon$ in

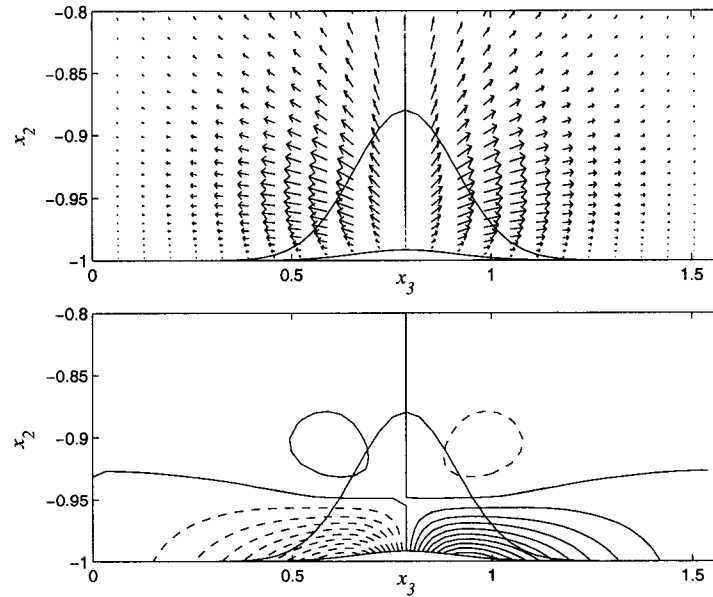


Figure I.5: The velocity vectors (up) and the streamwise vorticity contours (down) in plane A (upstream of the bump). Positive/negative values of the vorticity are denoted by solid/dashed lines. An outline of the maximum extent of the bump (situated further downstream) is also indicated (solid).

plane A (upstream) and $\pm 0.55U_l/\epsilon$ in plane B (downstream). These values are only slightly different from the corresponding values of $\pm 0.91U_l/\epsilon$ (upstream) and $\pm 0.61U_l/\epsilon$ (downstream) obtained by Carlson *et al.*, who did not specify the exact locations of the analysis planes they reported. The corresponding results reported by Mason & Morton, who considered a perturbed Blasius boundary layer flow instead of a perturbed laminar channel flow, are $\pm 0.88U_l/\epsilon$ (upstream) and $\pm 0.64U_l/\epsilon$ (downstream).

The pressure disturbance field on the lower wall of the channel is projected onto a horizontal plane and depicted in Figure I.7. The pressure on the leading side of the bump reaches a maximum of $0.46\rho U_l^2$, and drops to a minimum value of $-0.97\rho U_l^2$ near the crest as flow accelerates to divert around the bump. The egg-shaped positive contours and bean-shaped negative contours on the obstacle are qualitatively similar to those reported by Mason & Morton, where the maximum value of the wall pressure is $0.60\rho U_l^2$. The pressure contours far from the obstacle in Figure I.7 are different from those reported in Mason & Morton, as periodic boundary conditions are used in the horizontal directions in the present work, but not in Mason & Morton (1987). The streamlines in the streamwise symmetry plane in Figure I.7 indicate that there is a region of flow

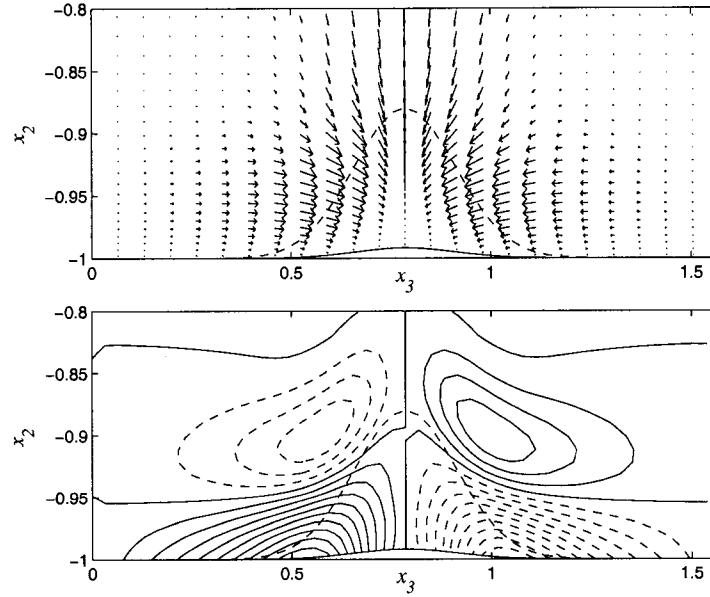


Figure I.6: The velocity vectors (up) and the streamwise vorticity contours (down) in plane B (downstream of the bump). Positive/negative values of the vorticity are denoted by solid/dashed lines. An outline of the maximum extent of the bump is also indicated (dashed).

separation behind the bump, as also observed in Mason & Morton.

I.4.3 Moving-boundary test

In the third test, we consider a (2D) flow with a moving boundary. As the present code is written to accommodate wall compliance, it is necessary to validate this code in test flows with moving boundaries. Unfortunately, there are very few fundamental test flows of this sort available in the literature for comparison. In previous work by our group, Luo & Bewley (2004) performed simulations of a laminar flow through a two-dimensional channel with an oscillating Gaussian bump using the (involved) contravariant formulation of the Navier-Stokes equation which is completely different from the present code (which is formulated with Cartesian vectors). We will use this result to validate the present code in the moving-boundary case.

Consider the laminar flow in a 2D channel with an oscillating Gaussian bump whose motion is prescribed by

$$\eta_l(x_1, t) = \varepsilon \sin(\omega t) \exp \left[-\frac{(x_1 - \frac{L_x}{2})^2}{\sigma^2} \right],$$

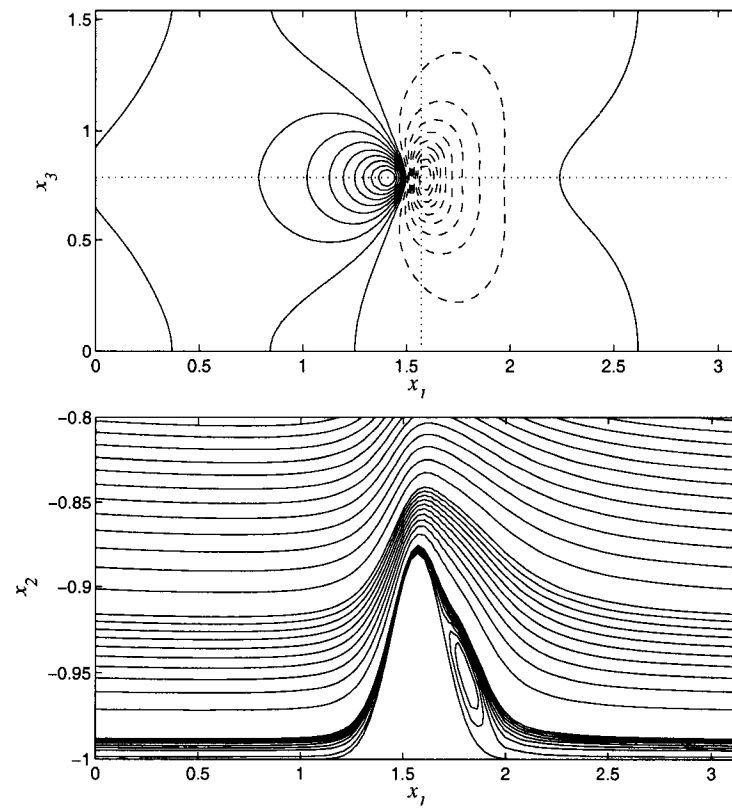


Figure I.7: Up: surface pressure disturbance field projected vertically on to a horizontal plane where negative values are denoted by dashed lines; down: the streamlines in the streamwise center plane.

Flow parameters	Domain size	No. Fourier modes (dealiased collocation points)
$Re = 200(\frac{U_c h}{\nu})$	$L_x = \pi$	$N_x = 42(64)$
$\varepsilon = 0.1$	$L_y = 2$	$N_y = 84$
$\sigma = 0.2$	$L_z = \pi$	$N_z = 4(6)$

Table I.3: Simulation parameters for the laminar flow over an oscillating bump.

where ε is the maximum amplitude of the bump, ω is the oscillation frequency, and σ is a constant parameterizing the length of the bump. The bump is uniform in the spanwise direction, so the laminar flow is two dimensional. In order to maintain incompressible flow, the upper wall is assumed to be penetrable and the normal velocity of the flow at the upper wall is taken to be identical to the vertical velocity of the lower wall. The simulation parameters are listed in Table I.3.

Two simulation results are presented, one with slow wall motion, $\omega = 0.5$ and the other with faster wall motion, $\omega = 4$. In Figures II.6 and II.7, the instantaneous streamlines and pressure are compared with the results from Luo & Bewley (2004). As seen in Figures II.6 and II.7, in both simulations, the agreement between the two codes is excellent. Several symmetries of the present (3D) code in the moving boundary case were also checked by aligning this (2D) flow and bump in various directions. For further discussion of the physics of this flow, see Luo & Bewley (2004).

I.4.4 Turbulence over a stationary wavy wall

In the fourth and final test, we consider the problem of flow over a wavy wall in the turbulent regime. Turbulence over a wavy wall has been studied both theoretically and experimentally in many investigations; see, e.g., De Angelis *et al.* (1997), Ohta *et al.* (1998), Zilker *et al.* (1977) and Zilker & Hanratty (1979). Due to the existence of a periodic variation of the streamwise pressure gradient in such flows, the flow in this geometry displays certain characteristics that are not found in plane channel flow. Flow separation regions may be formed behind the crests if the amplitude-to-wavelength ratio is large enough. In this test, two simulation results are presented, one for which this ratio is moderate, causing intermittent separation regions, as was studied by Ohta *et al.* (1998), and the other for which this ratio is large, causing stationary separation regions, as studied by De Angelis *et al.* (1997).

In the first simulation, the geometry considered has a flat top wall and a rippled bottom consisting of one wave, with $\eta_l(x_1) = -\varepsilon \cos(2\pi x/L_x)$ where ε is the wave amplitude. Parameters used in the test are given in Table I.4. Note that the friction velocity u_τ used for the definition of Re_τ

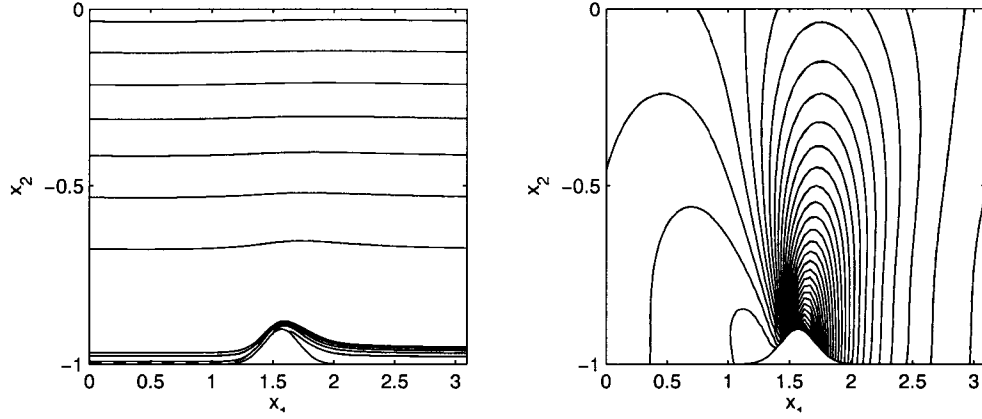


Figure I.8: Flow at $Re = 200$ passing over an oscillating Gaussian bump uniform in the span-wise direction with the oscillation frequency $\omega = 0.5$ at $t = 3.0h/U_c$. Left: instantaneous streamlines ψ ; right: pressure contours. Solid: 3D simulation; dashed: results from Luo & Bewley (2004). Quantification of error: $\|\psi_{error}\|_2 / \|\psi\|_2 = 1.4e - 04$; $\max_{\Omega}(\psi_{error}) / \|\psi\|_2 = 2.2e - 04$; $\|p_{error}\|_2 / \|p\|_2 = 2.2e - 03$; $\max_{\Omega}(p_{error}) / \|p\|_2 = 8.7e - 03$.

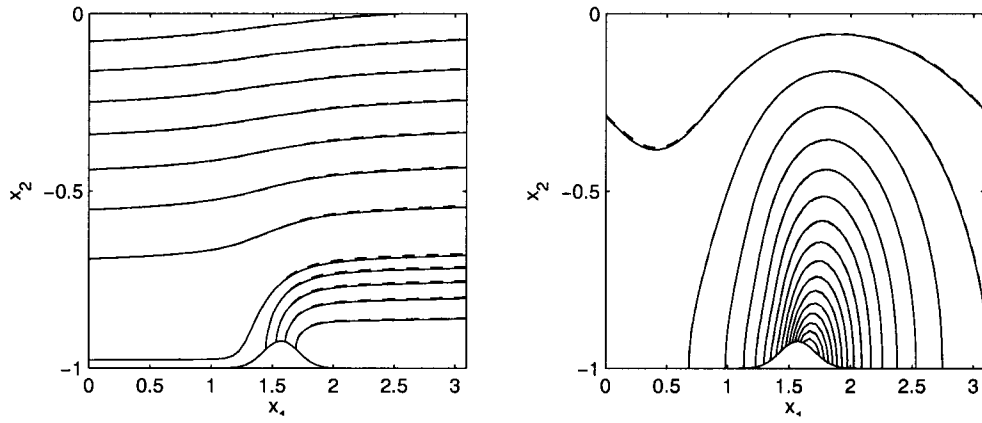


Figure I.9: Flow at $Re = 200$ passing over an oscillating Gaussian bump uniform in the span-wise direction with the oscillation frequency $\omega = 4$ at $t = 1.8h/U_c$. Left: instantaneous streamlines ψ ; right: pressure contours. Solid: 3D simulation; dashed: results from Luo & Bewley (2004). Quantification of error: $\|\psi_{error}\|_2 / \|\psi\|_2 = 3.8e - 04$; $\max_{\Omega}(\psi_{error}) / \|\psi\|_2 = 1.5e - 03$; $\|p_{error}\|_2 / \|p\|_2 = 4.2e - 03$; $\max_{\Omega}(p_{error}) / \|p\|_2 = 2.5e - 02$.

Flow parameters	Domain size	No. Fourier modes (dealiased collocation points)
$Re_\tau = 150$	$L_x = 2.5\pi$	$N_x = 64(96)$
$\varepsilon = 0.1$	$L_y = 2$	$N_y = 94$
	$L_z = 1.25\pi$	$N_z = 64(96)$

Table I.4: Simulation parameters for the turbulent flow in a wavy channel with a moderate slope.

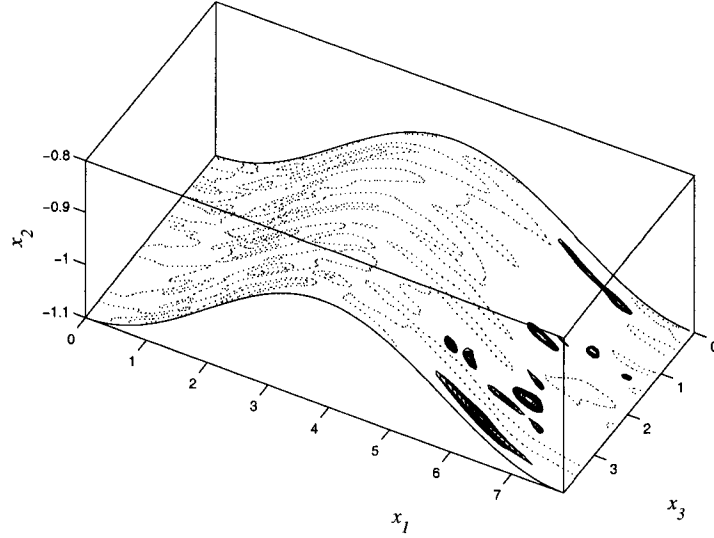


Figure I.10: Instantaneous contours of shear stress on the wavy wall. Dashed lines indicate positive values, and solid, dense lines indicate negative values where the flow separates.

in this test is defined with the averaged total drag on both walls, including the pressure drag caused by wall curvature. The amplitude-to-wavelength ratio is $\varepsilon/L_x = 0.013$. The grid resolution in each direction, in units of ν/u_τ , is $\Delta\xi_1^+ = 12.3$, $\Delta\xi_2^+ = 0.71 \sim 5.9$, and $\Delta\xi_3^+ = 6.1$. The time step is 0.002 normalized by h/u_τ , or 0.3 normalized by ν/u_τ^2 , resulting in a CFL number of about 0.6.

Figure I.10 shows the instantaneous contour plot of the streamwise shear stress on the wavy wall. The negative shear stress contours (solid lines) indicate the regions where flow separates from the wall. In animations of this simulation, these separation regions appear intermittently, and are located randomly in the spanwise direction but mostly behind the wavy crest in the streamwise direction.

The spanwise averaged skin friction of the flat (upper) wall τ_f , the skin friction of the wavy (lower) wall τ_w , and the form drag on the wavy (lower) wall p_{wx} are plotted in Figure I.11, where the form drag is caused by pressure forces and is calculated by $p_{wx} = -\rho\phi_1\tilde{p}$. These quanti-

Flow parameters	Domain size	No. Fourier modes (dealiased collocation points)
$Re_\tau = 171$	$L_x = 2\pi$	$N_x = 84 (128)$
$\varepsilon = 0.052$	$L_y = 2$	$N_y = 128$
	$L_z = \pi$	$N_z = 84 (128)$

Table I.5: Simulation parameters for the turbulent flow in a wavy channel with a larger slope.

ties agree with the results of Ohta *et al.* (1998) well by visual inspection. In both the present results and Ohta *et al.* (1998), the time-averaged shear stress on the wavy wall, τ_w , is always positive along the channel, which implies that the time-averaged flow does not separate from the wall, although there is often instantaneous local flow detachment, as shown in Figure I.10.

Figure I.11 shows the spanwise-averaged and time-averaged pressure coefficient C_p (averaged pressure normalized by ρu_τ^2), the non-dimensional pressure gradient β (normalized by $\rho u_\tau^3/\nu$), and the local friction velocity $u_{\tau,f}^+ = \sqrt{\tau_f/\rho}/u_\tau$ on the flat wall. Comparing these quantities with Figures 6 of Ohta *et al.*, only $u_{\tau,f}^+$ shows minor visible differences. From Figure 5 of Ohta *et al.*, the integral of the drag over the flat and wavy walls, $T_f = \int_0^{L_x} \tau_f dx / \rho u_\tau^2 L_x$ and $T_w = \int_0^{L_x} \tau_w dx / \rho u_\tau^2 L_x$ respectively, are both slightly less than 1, with additional form drag, $D_{form} = \int_0^{L_x} p_{wx} dx / \rho u_\tau^2 L_x$, on the wavy wall as the rest portion of the averaged overall drag. This is confirmed in our work, where $T_f \approx 0.92$, $T_w \approx 0.92$ and $D_{form} \approx 0.16$. (Note that the average value of $(u_{\tau,f}^+)^2$ over the channel in Figure 6 of Ohta *et al.* seems to be greater than 1, which appears to be inconsistent with their reported value of T_f .)

Our second simulation is compared to case S2 in De Angelis *et al.* (1997), where the amplitude-to-wavelength ratio is $\varepsilon/\lambda = 0.05$. The simulation domain contains 6 periods of wall oscillations. Table I.5 lists the simulation parameters. The Reynolds number $Re_\tau = 171$ is based on the mean shear velocity u_τ , which is defined using the total drag in the channel. To be consistent with De Angelis *et al.*, we also use a constant mean pressure gradient which maintains the corresponding regular channel flow at the same Reynolds number to drive the flow. The grid resolution in each direction, is $\Delta\xi_1^+ = 8.4$, $\Delta\xi_2^+ = 0.58 \sim 5.0$, and $\Delta\xi_3^+ = 4.2$. The time step is $\Delta t^+ = 0.26$ and the corresponding CFL number is about 0.7.

Figure I.12 shows the respective comparison of profiles of pressure and shear stress on the wavy (lower) wall, which show that our results match those of De Angelis *et al.* very well. Negative shear stress at the valley of each wave indicates consistent flow separation in that region. Small discrepancies are unavoidable since in De Angelis *et al.* the upper wall is assumed to be a slip wall and thus their channel (after appropriate scaling) is not exactly equivalent to the the lower

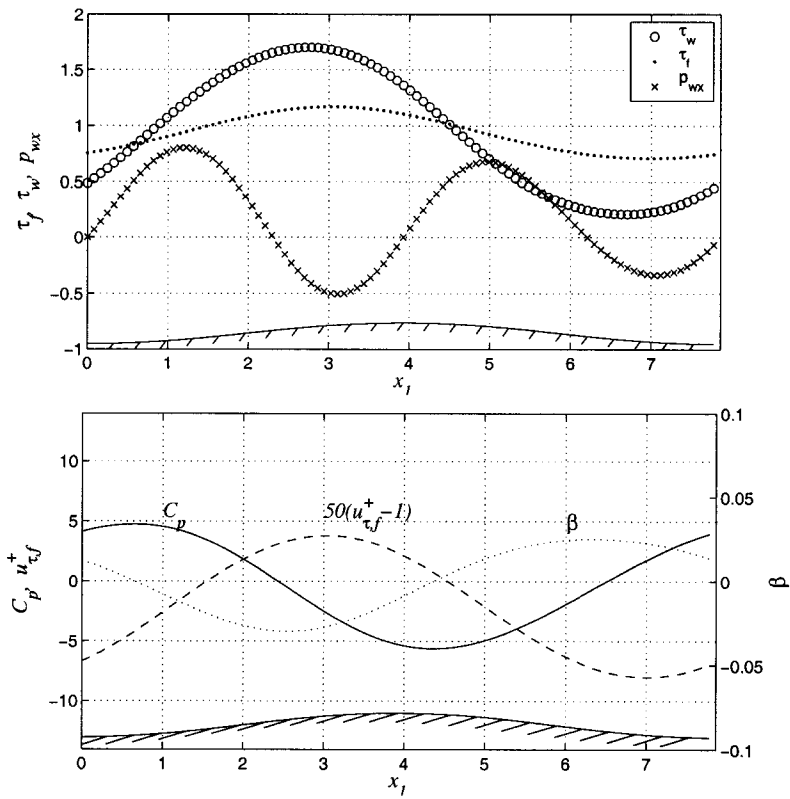


Figure I.11: Up: spanwise averaged components of drag on the walls. τ_f : friction stress on the flat wall; τ_w : friction stress on the wavy wall; p_{wx} : form drag on the wavy wall. Down: friction velocity $u_{\tau,f}^+$, pressure coefficient C_p and pressure gradient coefficient β along the flat wall.

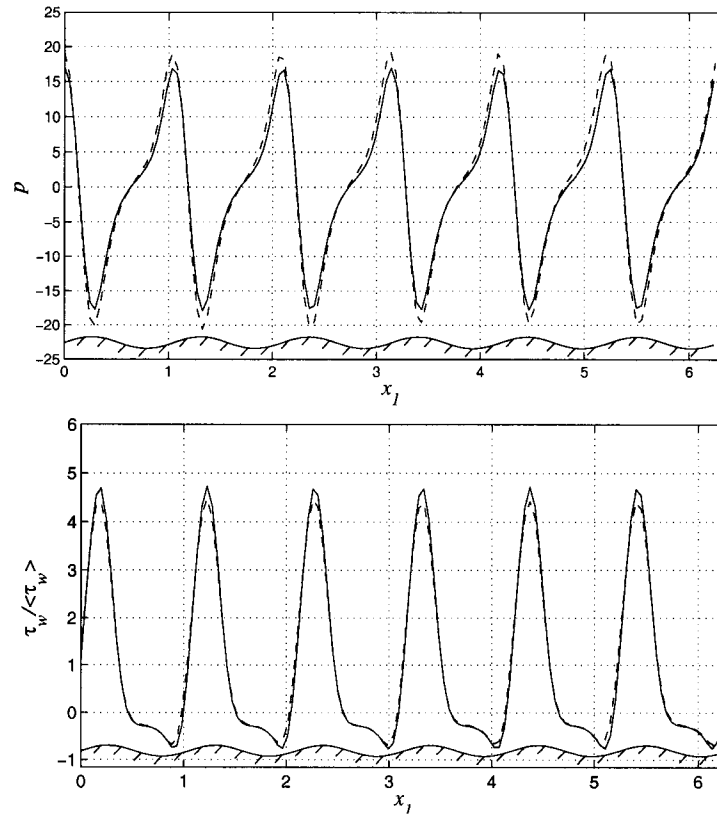


Figure I.12: Spanwise averaged pressure (up) and shear stress (down) on the wavy wall with the amplitude-to-wavelength ratio of 0.05. Solid: our simulation; dashed: simulation from De Angelis *et al.* (1997).

half channel in our work, as our upper wall is a no-slip flat wall and the mean shear stress at the center plane is not exactly zero.

I.5 The compliant surface model: tensegrity fabric

I.5.1 Introduction

The compliant surface model we consider is based on a special structural paradigm known as *tensegrity*, as mentioned previously. Invented by Buckminster Fuller and often realized in creative artwork, tensegrity structures have been extensively characterized in the work of Skelton *et al.* (2001). This and related work have characterized the geometry, statics, dynamics, controllability, deployability, and many other interesting characteristics of these interesting structures. One partic-

ular class of tensegrity of interest in the present work is the *plate-class* tensegrity structure, which is formed by joining stable tensegrity unit cells together so that they tile the plane (Masic & Skelton, 2002). Figure I.13 shows the top views of different possible unit cells interconnected to form a plate by attaching certain nodes of a cell to the tendons of its adjacent cells and conversely, attaching certain nodes of the adjacent cells to its tendons. Further details about the geometry and equilibrium conditions of such structures can be found in Masic & Skelton (2002). With slight modifications, we extended the four-bar plate class tensegrity structures shown in Figure I.13 to form our compliant surface model, as described below.

The tensegrity fabric, which is our compliant surface model, is formed by simply extending periodic pattern of a plate-class tensegrity infinitely in plane and fixing its bottom nodes on ground. In present research, we choose to study 4-bar-unit configuration only since in this paradigm the two periodic directions are perpendicular, which is consistent to the canonical rectangular flow domain. The geometry of a fabric unit cell at equilibrium is illustrated in Figure I.14. Both the fixed base and flexible top of the cell consist of four nodes which form two identical squares in parallel planes. The two squares are twisted such that the angle between them is $\alpha = \frac{\pi}{4}$. There are four horizontal tendons connecting the top nodes, and another four vertical tendons connecting the top nodes and base nodes. The geometry of the cell may be determined by the radius of the squares, r , and the distance between the squares, h_c . Other parameters can be derived from r and h_c . For example, the bar length is $l_b = \sqrt{h_c^2 + (2 + \sqrt{2})r^2}$, the edge of the squares is $a = \sqrt{2}r$, and the vertical tendon length at equilibrium is $l_v = \sqrt{h_c^2 + (2 - \sqrt{2})r^2}$. To maintain equilibrium, the bar force f_b , the vertical tendon force f_v , and the horizontal tendon force f_h satisfy the condition $\|f_b\| : \|f_v\| : \|f_h\| = l_b : l_v : r$. Note that the direction of each force is parallel to the corresponding structural member, and that f_b is compressive force and f_v and f_h are tensile forces.

Figure I.15 shows geometry of the two neighboring unit cells connected to each other. Node 1 on the top of cell B is attached to tendon T_1 of cell A , and node 2 on the top of cell A is attached to tendon T_2 of cell B . The overlapping portion of the two tendons, T_1 and T_2 , is such that the base edges of the two cells are overlapping as well. As a matter of fact, in a 4-bar plate-class tensegrity where the bottom tendons are present, the overlapping structural elements at the base are attached in the same manner as those on the top. In this 4-bar unit configuration, the ratio of overlapping portion, a_0 , to the total length of T_1 , a , is a constant, $a_0/a = 2 - \sqrt{2}$. In the same way, a unit cell is connected to four neighbor cells in a tensegrity fabric, and each cell would stay in equilibrium and preserve its original geometry. When two tendons overlap at some portions,

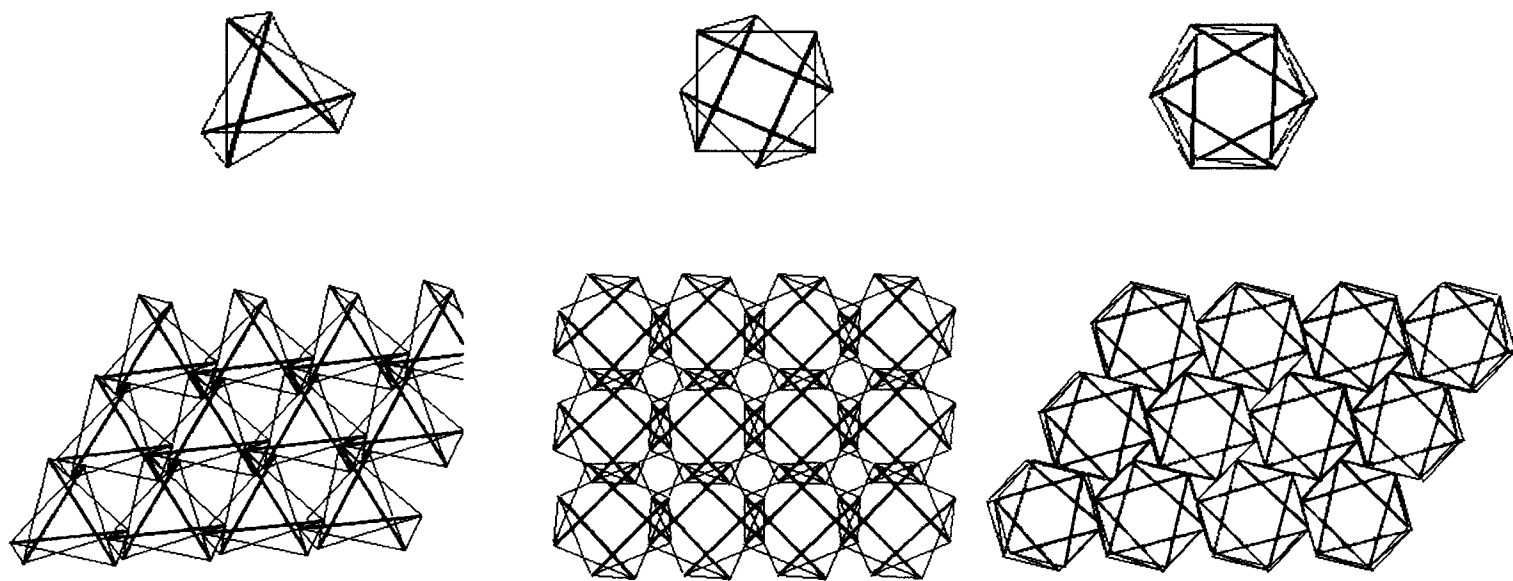


Figure I.13: tensegrity cells and their corresponding plate-class structure. Left: 3-bar-unit configuraton; middle: 4-bar-unit configuraton; right: 6-bar-unit configuraton.

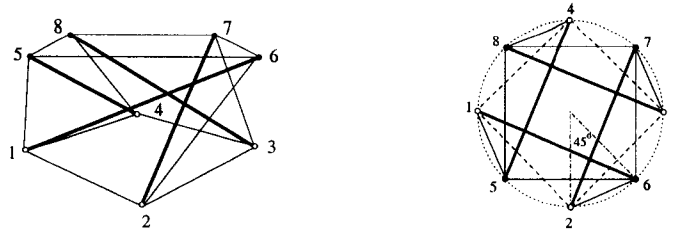


Figure I.14: A 4-bar-unit tensegrity cell. Left: an oblique/perspective view; right: top view.

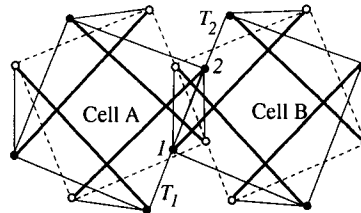


Figure I.15: Connection of two unit cells. The bottom is indicated with dashed lines.

the overlapping portions may reduce to a new tendon with the tension equal to the sum of the two tendons' forces.

By designing the fabric in such a manner, failures of individual members do not compromise the integrity of the entire tensegrity fabric, but simply lead to a modest deformation of the nearby cells, thus providing robustness in the overall system.

I.5.2 Dynamics of the tensegrity structure

Prof. Skelton's group have done extensive study on dynamics of general type of tensegrity structures. We now slightly modify their work and extend it to the tensegrity fabric.

In order to derive the equation for dynamics of the entire interconnected tensegrity fabric, we first consider a single bar of length l_b and linear density ρ_b with one node subject to force $\mathbf{f}$ (Figure I.16) and the other node fixed. The two orientation angles, θ and ϕ are used to describe the position of the bar $\mathbf{q} = \begin{bmatrix} \theta \\ \phi \end{bmatrix}$. The inertia about the fixed node is $I_o = \frac{1}{3}\rho_b l_b^3$. By the angular momentum conservation law, we may easily derive the bar's dynamics,

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = H(\mathbf{q})\mathbf{f}, \quad (\text{I.16})$$

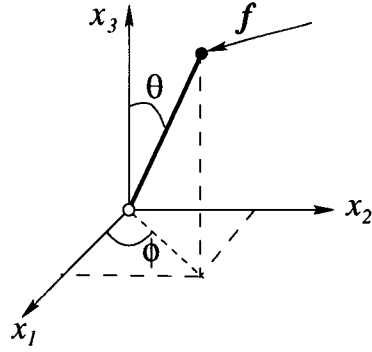


Figure I.16: Dynamics of a bar with one node fixed.

where M is the inertia matrix given by

$$M(\mathbf{q}) = \begin{bmatrix} I_o & 0 \\ 0 & I_o \sin^2 \theta \end{bmatrix}, \quad (\text{I.17})$$

$C(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis and Centripetal matrix with $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ given by

$$C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \begin{bmatrix} -\frac{1}{2}I_o \sin(2\theta) \dot{\phi}^2 \\ I_o \sin(2\theta) \dot{\phi} \dot{\theta} \end{bmatrix}, \quad (\text{I.18})$$

and the matrix H is given by

$$H(\mathbf{q}) = l_b \begin{bmatrix} \cos \theta \cos \phi & \cos \theta \sin \phi & -\sin \theta \\ -\sin \theta \sin \phi & \sin \theta \cos \phi & 0 \end{bmatrix}. \quad (\text{I.19})$$

Consider now a multiple-bar system consisting of N_b bars. We now add the subscript i to all of the quantities associated with the i 'th bar, then combine the equations for all of the bars and rewrite them in a compact form. To start, we make the following definitions:

$$\begin{aligned} \mathbf{q} &= [\mathbf{q}_1^T, \dots, \mathbf{q}_i^T, \dots, \mathbf{q}_{N_b}^T]^T \\ \mathbf{M}(\mathbf{q}) &= \text{BlockDiag}[M_1, \dots, M_i, \dots, M_{N_b}] \\ \mathbf{f} &= [\mathbf{f}_1^T, \dots, \mathbf{f}_i^T, \dots, \mathbf{f}_{N_b}^T]^T \\ \mathbf{H}(\mathbf{q}) &= \text{BlockDiag}[H_1, \dots, H_i, \dots, H_{N_b}] \\ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} &= [(C(\mathbf{q}_1, \dot{\mathbf{q}}_1)\dot{\mathbf{q}}_1)^T, \dots, (C(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i)^T, \dots, (C(\mathbf{q}_{N_b}, \dot{\mathbf{q}}_{N_b})\dot{\mathbf{q}}_{N_b})^T]^T. \end{aligned} \quad (\text{I.20})$$

Note that, from here forward, $\mathbf{q}$, $\mathbf{f}$ are redefined for the entire multiple-bar system, as indicated above. We may now express the dynamics of the multiple-bar system in the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{H}(\mathbf{q})\mathbf{f}. \quad (\text{I.21})$$

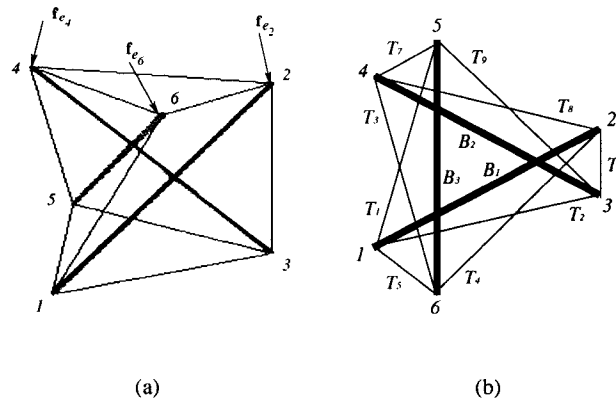


Figure I.17: A 3-bar tensegrity unit cell. (a) Oblique view (with node 6 closest to viewer and node 5 farthest from viewer), (b) top view (with nodes 2, 4, & 6 closest to viewer and nodes 1, 3, & 5 farthest from viewer).

I.5.3 Interconnection of the multiple-bar system

In a tensegrity structure, the bars are interconnected by tendons. In the dynamical equation (I.21), this interconnection appears in the calculation of the nodal force vector $\mathbf{f}$ as

$$\mathbf{f} = \mathbf{S}\mathbf{f}_t + \mathbf{f}_e, \quad (\text{I.22})$$

where $\mathbf{S}$ is the connectivity matrix, $\mathbf{f}_t$ is the tendon force vector, and $\mathbf{f}_e$ is the external force vector acting on the nodes (in our case, generated by the flow). The connectivity matrix $\mathbf{S}$, whose elements consist only of 1's, -1's, and 0's, defines to which two nodes each tendon is connected.

To show what the connectivity of the structure means and how the nodal force is computed from the tendon force based on the connectivity, we now illustrate with a simple 3-bar unit tensegrity cell whose nodes are free of constraint, as depicted in Figure I.17. In a tensegrity fabric, the principle of connectivity for the free nodes and the way to use connectivity to calculate forces on free nodes are exactly the same as in the example. There are 3 bars (B_1 to B_3), 6 nodes, and 9 tendons (T_1 to T_9) in this cell structure. Denoting N_b as number of bars, N_d as number of nodes, and N_t as number of tendons, we have $N_b = 3$, $N_d = 6$, and $N_t = 9$ in this example. Note that in a typical tensegrity system, we require that there are no bar-to-bar connections, so the number of nodes is exactly twice the number of the bars. Assume that there are external force vectors $\mathbf{f}_{e2}$, $\mathbf{f}_{e4}$, $\mathbf{f}_{e6}$ imparting on the top nodes 2, 4, 6, respectively. We denote $\mathbf{p}_i$ as the position, $\mathbf{v}_i$ as the velocity of the i 'th node, and define the i 'th tendon vector $\mathbf{l}_i$ to be the vector from the starting node to the ending node of tendon

T_i , as shown in the following table:

Tendon	Starting node	Ending node	Tendon vector
T_1	5	1	$\mathbf{l}_1 = \mathbf{p}_1 - \mathbf{p}_5$
T_2	1	3	$\mathbf{l}_2 = \mathbf{p}_3 - \mathbf{p}_1$
T_3	4	6	$\mathbf{l}_3 = \mathbf{p}_6 - \mathbf{p}_4$
T_4	6	2	$\mathbf{l}_4 = \mathbf{p}_2 - \mathbf{p}_6$
T_5	1	6	$\mathbf{l}_5 = \mathbf{p}_6 - \mathbf{p}_1$
T_6	3	2	$\mathbf{l}_6 = \mathbf{p}_2 - \mathbf{p}_3$
T_7	5	4	$\mathbf{l}_7 = \mathbf{p}_4 - \mathbf{p}_5$
T_8	4	2	$\mathbf{l}_8 = \mathbf{p}_2 - \mathbf{p}_4$
T_9	3	5	$\mathbf{l}_9 = \mathbf{p}_5 - \mathbf{p}_3$.

The associated connectivity matrix $\mathbf{S}$ has dimension $3N_d \times 3N_t$. In block matrix form, the connectivity matrix $\mathbf{S}$ of the 3-bar unit and the tendon force vector $\mathbf{f}_t$ are

$$\mathbf{S} = \begin{bmatrix} -I_3 & I_3 & \mathbf{0} & \mathbf{0} & I_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -I_3 & \mathbf{0} & -I_3 & \mathbf{0} & -I_3 & \mathbf{0} \\ \mathbf{0} & -I_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_3 & \mathbf{0} & \mathbf{0} & I_3 \\ \mathbf{0} & \mathbf{0} & I_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & -I_3 & I_3 & \mathbf{0} \\ I_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_3 & \mathbf{0} & -I_3 \\ \mathbf{0} & \mathbf{0} & -I_3 & I_3 & -I_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{f}_t = \begin{bmatrix} \mathbf{f}_{t_1} \\ \mathbf{f}_{t_2} \\ \mathbf{f}_{t_3} \\ \mathbf{f}_{t_4} \\ \mathbf{f}_{t_5} \\ \mathbf{f}_{t_6} \\ \mathbf{f}_{t_7} \\ \mathbf{f}_{t_8} \\ \mathbf{f}_{t_9} \end{bmatrix},$$

where I_3 is a 3 by 3 Identity matrix. In the above matrix, the i 'th block column is associated with the i 'th tendon in a prescribed manner. For example, the first tendon T_1 starts from node 5 and ends at node 1, so in the first block column of $\mathbf{S}$, we put $-I_3$ at the first block row and I_3 at the fifth block row. The general rule is that if the i 'th tendon T_i starts from node j and ends at node k , then in the i 'th block column of $\mathbf{S}$, the k 'th block row element is $-I_3$ and j 'th block row element is I_3 . With $\mathbf{f}_{t_i}$

defined as the tension force vector of the i 'th tendon, we may now compute the nodal force as

$$\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \\ \mathbf{f}_5 \\ \mathbf{f}_6 \end{bmatrix} = \begin{bmatrix} -I_3 & I_3 & 0 & 0 & I_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I_3 & 0 & -I_3 & 0 & -I_3 & 0 \\ 0 & -I_3 & 0 & 0 & 0 & I_3 & 0 & 0 & I_3 \\ 0 & 0 & I_3 & 0 & 0 & 0 & -I_3 & I_3 & 0 \\ I_3 & 0 & 0 & 0 & 0 & 0 & I_3 & 0 & -I_3 \\ 0 & 0 & -I_3 & I_3 & -I_3 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f}_{t_1} \\ \mathbf{f}_{t_2} \\ \mathbf{f}_{t_3} \\ \mathbf{f}_{t_4} \\ \mathbf{f}_{t_5} \\ \mathbf{f}_{t_6} \\ \mathbf{f}_{t_7} \\ \mathbf{f}_{t_8} \\ \mathbf{f}_{t_9} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{f}_{e_2} \\ 0 \\ \mathbf{f}_{e_4} \\ 0 \\ \mathbf{f}_{e_6} \end{bmatrix}.$$

I.5.4 Tendon forces in the interconnected system

The tension force in each tendon is based on Hooke's law, with linear damping incorporated which is proportional to the rate of change of the tendon vector. Define a tendon's length to be $l_i = \|\mathbf{l}_i\|$. When a tendon's length is less than its rest length, the tendon is considered slack which does not produce any force. The force vector of the i 'th tendon is thus modeled with

$$\mathbf{f}_{t_i} = \begin{cases} \frac{\kappa_i}{l_{0_i}} (l_i - l_{0_i}) \frac{\mathbf{l}_i}{l_i} - \zeta_i \dot{\mathbf{l}}_i, & \text{if } l_i \geq l_{0_i} \\ \mathbf{0}, & \text{if } l_i < l_{0_i} \end{cases}. \quad (\text{I.23})$$

where κ_i is the linear stiffness of the i 'th tendon per inverse unit rest length, l_{0_i} is its rest length, and ζ_i is its damping coefficient.

I.5.5 Periodic boundary condition for the infinite tensegrity fabric

The tensegrity fabric with the recurrent pattern exhibits homogeneous material properties from macroscopic view although each composing unit cell is heterogeneous. A well-adopted way of studying an incompressible homogeneous material with infinite dimension is to crop the material to a finite patch and assume periodic condition for the boundaries, which has been applied to the overlying flow. Therefore, to be consistent, it is necessary to extend the assumption to the tensegrity fabric. However, it is not as obvious to specify periodic condition for such an anomalous discrete structure as for a continuum system. Thanks to the repetitive pattern of the structure, the periodic specification may be stipulated for the individual structural members at the boundaries. To show the procedure, we take the example of one-dimensional periodic array of the 4-bar unit cells connected

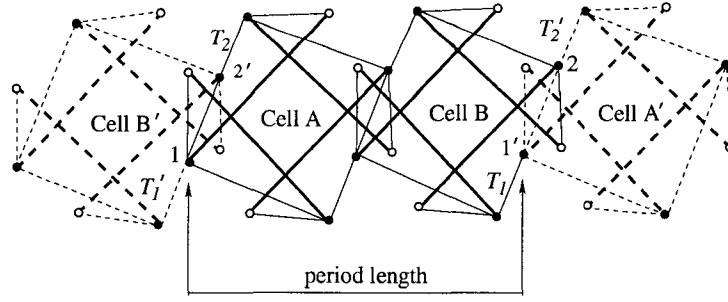


Figure I.18: Illustration of periodic boundary condition for a 2-cell fabric patch. The ghost elements are drawn in dashed lines.

together as seen in Figure I.18. Assume that the finite patch consists of only two mutually attached cells, A and B. With the periodic connection assumption, the ghost cells of A and B, A' and B', are connected to the right and left end, respectively. The attachment nodes are marked with 1, 2' and 1', 2 in the figure.

In practice, however, we don't need to generate the two ghost cells when specifying the geometry of the periodic patch. We only need to specify the nodes in the ghost cells that are attached to the patch, which we call ghost nodes. In present example, the ghost nodes are node 1' and 2' which are associated to node 1 and 2, respectively. Note that tendon $\overline{12'}$ and tendon $\overline{1'2}$ are identical, and we specify only one of them in the patch. One can easily extend the method to a two-dimensional periodic structural array.

To compute the dynamics of a finite tensegrity fabric with periodic boundary conditions, the influence of the ghost tendons to the boundary bars needs to be taken into account. At the same time, the position and velocity of the ghost node i need to be updated to match its associated node i' as the structure evolves in time. That is,

$$\mathbf{p}_{i'} = \mathbf{p}_i + \mathbf{L}, \quad \dot{\mathbf{p}}_{i'} = \dot{\mathbf{p}}_i, \quad (\text{I.24})$$

where $\mathbf{L}$ is the periodic vector of the tensegrity fabric. When computing the force imparting on node 2, the forces generated by tendons T_2 and $\overline{12'}$ which impart on the ghost node 2' are transmitted onto node 2. Similarly, when computing the force on node 1, the force of tendon T_1 on the ghost node 1' is transmitted onto it. Therefore, the nodal forces on a tensegrity fabric may be computed in three steps:

1. $\mathbf{f} = \mathbf{S}\mathbf{f}_i$ for all nodes including the ghost nodes.

2. $\mathbf{f}_i + \mathbf{f}_{i'} \rightarrow \mathbf{f}_i$ for node i which has a corresponding ghost node.
3. $\mathbf{f} + \mathbf{f}_e \rightarrow \mathbf{f}$ to account for the external nodal forces.

I.6 Simulation of a turbulent flow over tensegrity fabrics

I.6.1 Nondimensionalization of the tensegrity parameters

Once the topology, geometry, and prestress of a tensegrity fabric are defined, the structure's characteristics are governed by the material properties of its unit cell, which include mass per unit length of each bar ρ_{b_i} , stiffness κ_j and damping ς_j of each tendon. These parameters are normalized by the metrics used by the flow system. The procedure may be done by simply considering the linear acceleration equation of a free (ungrounded) bar subject only to the pressure disturbance from the flow,

$$\rho_{b_i} l_{b_i} \ddot{\mathbf{p}}_{c_i} = \sum_j \left[\frac{\kappa_j}{l_{0_i}} (l_i - l_{0_i}) \frac{\mathbf{l}_i}{l_i} - \varsigma_j \dot{\mathbf{l}}_i \right] - p_w A_0 \mathbf{e}_2, \quad (\text{I.25})$$

where l_{b_i} is the length of the bar, $\mathbf{p}_{c_i}$ is the center of mass, p_w is the hydrodynamic pressure disturbance generated by the overlying flow, and A_0 is the interface area associated to the top node of the bar on which the external pressure is lumped.

If we use L for the length scale, T for the time scale, it can be easily seen the following dimension similarities are valid

$$\rho_{b_i} \sim p_w T^2, \quad \kappa_j \sim \frac{\rho_{b_i} L}{T^2}, \quad \varsigma_j \sim \frac{\rho_{b_i} L}{T}.$$

Since in the present flow system, L is normalized by the half channel width h , p_w is normalized by ρu_τ^2 , T is normalized by h/u_τ , then the final normalization is

$$\rho_{b_i} \sim \rho h^2, \quad \kappa_j \sim \rho h u_\tau^2, \quad \varsigma_j \sim \rho h^2 u_\tau. \quad (\text{I.26})$$

I.6.2 Representation of the flow/structure interface

Geometrical configuration of the flow/structure system is shown in the Figure I.19. All the bars in the tensegrity fabric are assumed to be grounded. Discretized with the third-order Runge-Kutta scheme, (I.21), which governs dynamics of the compliant surface, is coupled with (I.7), which governs dynamics of the flow, and is marched in time to simulate the turbulence/structure interaction. Two techniques have been identified to form the flow/structure interface between the (continuous) flow above and the (discrete) structure below. The first is to simply stretch a massless,

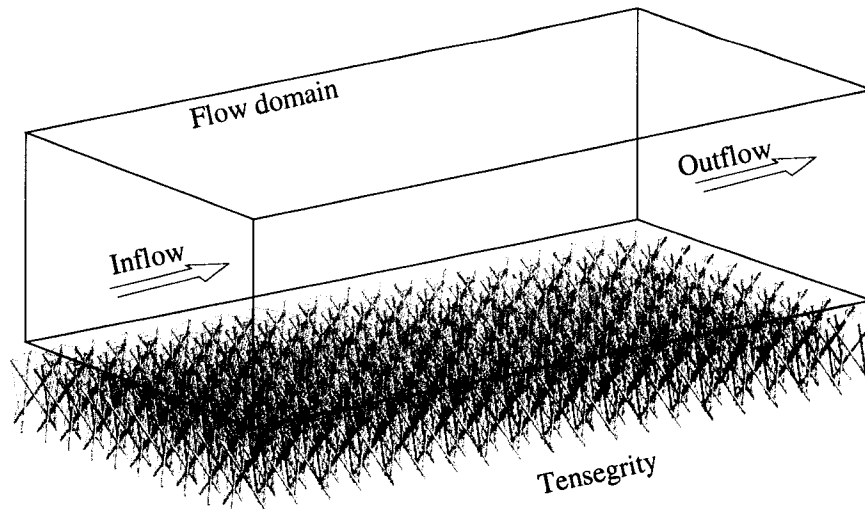


Figure I.19: Illustration of the computational domain for a turbulent flow over a tensegrity-based compliant surface.

tensionless and non-penetrable membrane over the top of the tensegrity structure, transmitting the force generated by the flow to the top nodes of the structure. Another option includes the attachment of small mechanical "scale" to each surface node of the tensegrity structure, mimicking the scales on a shark's skin. No-slip and no-penetration are assumed for the boundary condition on the interface in either case. In our simulation model, we assume that the force from the flow is transmitted to the top nodes of the structure in a simple fashion approximating the latter technique. The surface is tessellated into small patches surrounding each node. The friction and pressure forces induced by the flow are then integrated over each patch and lumped to the associated node of the tensegrity structure. Since the bar nodes of the tensegrity structure do not coincide with the grid points in the flow model, the vertical displacements and velocities of the nodes are interpolated onto the uniform $x_1 - x_3$ grid so they could be fed back to the flow system as the boundary condition.

I.6.3 Simulation parameters

In the flow/structure simulations, all velocities are normalized by u_τ which is the viscous velocity from the corresponding regular channel flow with the same bulk mass flux. The Reynolds number for the flow is $Re_\tau = u_\tau h / \nu = 150$. (This corresponds to the Reynolds number based on the mean centerline velocity of the regular channel flow of $Re = U_c h / \nu = 2663$, and to the Reynolds number based on the bulk velocity of $Re_b = U_{bulk} h / \nu = 2280$.) The size of the computational domain

is $5.625\pi \times 2 \times \pi$. That is, in wall units, the domain length and width are $L_x^+ \approx 2651$ and $L_z^+ \approx 471$. The number of Fourier modes used is $144 \times 94 \times 52$ in the ξ_1 , ξ_2 , and ξ_3 directions respectively (i.e., $216 \times 94 \times 78$ dealiased collocation points). The tensegrity fabric used in the simulations consists of 45 unit cells in length and 8 cells in width, each of which has edge length of $0.36h$ and height of $0.51h$ (the height is taken to be twice of the radius of the cells). Tendons are pre-stretched so they are about $2 \sim 3$ times as long as their rest length.

For preliminary studies, we assume that all structural members have uniform material properties, i.e., $\rho_{b_i} = \rho_b$, $\kappa_i = \kappa$ and $\varsigma_i = \varsigma$.

I.7 Results

A fully developed turbulent flow in the regular channel is used as the initial conditions for all the compliant channel flows. Transient process is ignored, and flow statistics is taken from long enough time intervals. Table I.6 shows the selected flow statistics for some various combinations of tensegrity parameters, along with the statistics of the flow in the regular channel with the same bulk velocity. The statistics includes the total drag D_w on the compliant wall, drag on the flat wall D_f , root-mean-square (RMS) of the compliant wall pressure $p_{w,rms}$, RMS wall velocity $v_{w,rms}$, maximum wall deformation at a time instance $y_{w,max}^+$, RMS of wall deformation $y_{w,rms}^+$, power done on the flow by wall pressure $p_w v_w$, form drag on the compliant wall D_{form} and the turbulent kinetic energy (TKE) in the channel.

When the structure is stiff and has high damping, e.g., case I where $\rho_b = 0.08$, $\kappa = 0.10$ and $\varsigma = 0.05$, the compliant surface barely changes the flow statistics. In case I, the wall deformation and the wall motion are both very small, so the flow acts as if the interface were a solid flat wall. When the structural damping is reduced, as in case II and III where $\varsigma = 0.035$ and $\varsigma = 0.03$, respectively, the interface starts to move more and the flow statistics is changed significantly. The total drag on the interface is increased by 5% in case II and 17% in case III, and the TKE is increased by 3% and 12% respectively. In next section, we will show that these changes are caused by the resonant vibration of the compliant structure and we will characterize the vibration mode.

In case IV and V, we keep the damping $\varsigma = 0.05$, but reduce the tendon stiffness κ to 0.06 and 0.03, respectively. The flow statistics is slightly modified. The wall deformation grows as the structure gets softer. However, the wall does not move much faster than in case I due to high system damping. In case VI, we reduce the bar density ρ_b to 0.04. Not surprisingly, the

potential influences of external disturbances and its own restoring forces on this lighter structure are counteracted by the increased effect of internal damping, so the flow is not affected much by the surface compliance.

Case VII and VIII have both the stiffness and damping lowered. In case VII where $\kappa = 0.04$ and $\varsigma = 0.035$, the instantaneous wall deformation reaches 5 viscous units, and in case VIII $\kappa = 0.04$ and $\varsigma = 0.03$, it reaches 9 viscous units. The flow statistics is greatly affected by the surface compliance. The total drag on the interface is increased by 6% in case VII and 22% in case VIII, and the TKE is increased by 7% and 23% respectively. We distinguish these two cases from case II and III because, although case VII and VIII have larger wall deformations, but the motions of the structure are less synchronized than they are in case II and III. We will discuss in detail in §I.7.2.

I.7.1 Case II and III

Interestingly, all the cases in the present work where the flow statistics is significantly modified by the wall compliance show that, the deformation of the wall is dominated by spanwise ridges that travel in the streamwise direction, reminiscent of the interface of air-water flow in Fulgosi *et al.* (2003). The wavy interface in Fulgosi *et al.* travels at a much slower phase speed than in the present work and has no significant influence on the air flow. This section investigates more details of the flow/structure interaction in case II and III where the wall seems to be resonant under excitation of the flow disturbances, causing strong flow oscillations.

Figure I.20 visualizes the shape and velocity of the interface at an instant moment for case III. The wall deformation is approximately sinusoidal in the streamwise direction, and the wall velocity has about 90° phase shift ahead of the wall, indicating that the wave is traveling in the same direction as the flow. Case II has the similar wall deformation with a smaller interfacial wave amplitude.

The wavelength, λ , of the interfacial motion may be seen from the statistics of streamwise two-point correlation function¹ of the wall displacement, $R(y_w, y_w)$, which is plotted in Figure I.21(a). The location of first valley point may be treated as the half wavelength. Therefore, for both case II and III we have $\lambda^+ = 530$. The correlation functions also show that the interface in case III has more regular sinusoidal shape since its autocorrelation decays more slowly than that in case II. In the streamwise spectra of wall deformation (Figure I.21(b)), the corresponding vibration mode for the two cases, $k_x = 1.78$, is much stronger than other modes. The hump in the high spectra

¹The spatial correlation function of two quantities, f and g , is normalized, given by $R(f, g) = \frac{\langle f(\mathbf{x})g(\mathbf{x}+\mathbf{r}) \rangle}{\sqrt{\langle ff \rangle \langle gg \rangle}}$.

case	ρ_b	κ	ζ	D_w	D_f	$p_{w,rms}$	$v_{w,rms}$	$y_{w,max}^+$	$y_{w,rms}^+$	$p_w v_w$	D_{form}	TKE
I	0.08	0.10	0.05	1.00	0.99	1.5	0.007	0.45	0.12	-0.004	4.7×10^{-4}	3.51
II	0.08	0.10	0.035	1.05	1.01	3.1	0.036	1.30	0.78	-0.084	0.014	3.64
III	0.08	0.10	0.03	1.17	1.03	5.0	0.069	2.38	1.00	-0.267	0.045	3.95
IV	0.08	0.06	0.05	1.01	1.02	1.6	0.008	0.63	0.18	-0.005	6.0×10^{-4}	3.56
V	0.08	0.03	0.05	1.02	1.00	1.6	0.008	1.14	0.32	-0.005	9.6×10^{-4}	3.55
VI	0.04	0.10	0.05	1.01	1.01	1.6	0.009	0.47	0.12	-0.007	7.8×10^{-4}	3.55
VII	0.08	0.04	0.035	1.06	1.01	3.3	0.034	2.66	0.93	-0.070	0.013	3.75
VIII	0.08	0.04	0.03	1.22	1.05	5.9	0.078	5.34	1.94	-0.327	0.065	4.33
regular channel				1	1	1.45	0	0	0	0	0	3.52

Table I.6: Flow statistics for different tensegrity parameters.

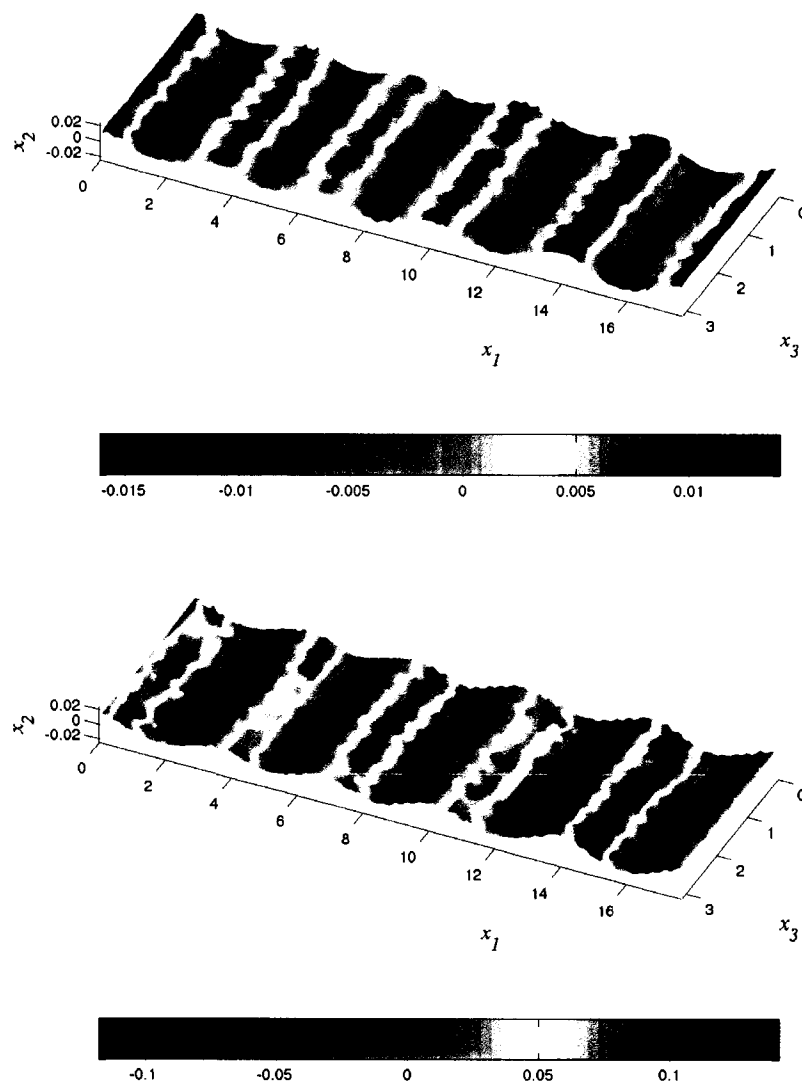


Figure I.20: Visualization of the flow/structure interface in case III. Up: wall displacement from its nominal position $x_2 = -1$; down: wall velocity.

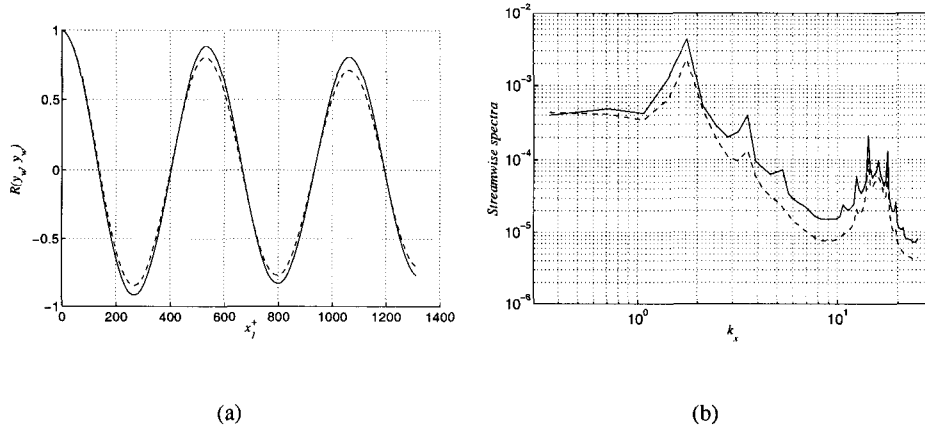


Figure I.21: Spatial characteristics of the wall deformation y_w for case II (dashed) and case III (solid). (a) streamwise autocorrelation; (b) streamwise spectra in log-log scale.

area represents response of the wall to characteristic disturbances of the continuous length scales in near-wall turbulence. Spikes on the hump coincide with the geometry lengths in the tensegrity cells, implying that they are caused by the discrete nature of the structure. Compare to case III, case II has weaker spectra.

The time correlation function of a fixed point on the wall, $Z(t)$, and the time spectra of wall oscillations of the two cases, are shown in Figure I.22. From them we may estimate the period and frequency of resonance, which are $t_p = 0.6$, and $f_{freq} = 1.6$, respectively. Therefore the phase traveling speed of the wave is $c/u_\tau = \lambda/t_p \approx 6$.

In all cases, pressure is an order of magnitude higher than shear stress disturbances and is the dominant excitation force that deforms the wall. Therefore, we focus on the relation of wall motion and pressure. Figure I.23 shows the correlation of wall pressure and wall deformation, $R(p_w, y_w)$, and the correlation of wall pressure and wall velocity, $R(p_w, v_w)$. Autocorrelations of y_w , v_w are provided as references. $R(p_w, y_w)$ is shifted about 100° to the right of $R(y_w, y_w)$, meaning that pressure peaks at the upstream side of the wave crests. $R(p_w, v_w)$ is shifted about 180° from $R(v_w, v_w)$, meaning that pressure is in opposite phase with the wall velocity and $\langle p_w v_w \rangle$ is negative, so that kinetic energy is transmitted from the flow to the wall structure.

Instantaneous pressure disturbances on the wavy interface are visualized in Figure I.24(a). Different from the turbulence over a stationary wavy wall where the pressure peaks at the upstream side of the wave but reaches to minimum near the wave crests, the pressure in present case has min-

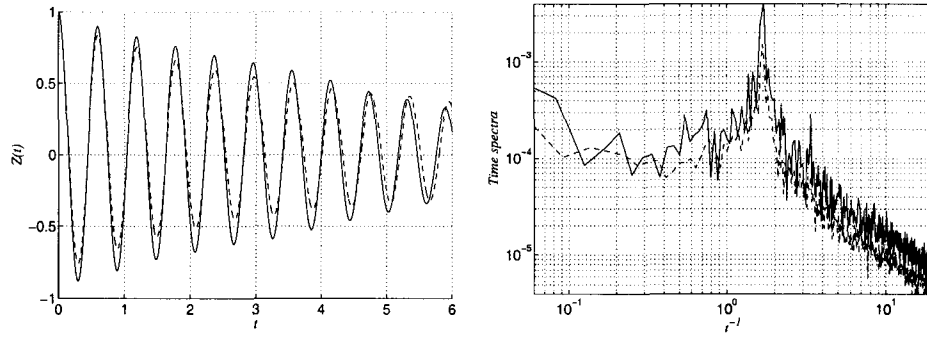


Figure I.22: Temporal characteristics of the wall deformation y_w for case II (dashed) and case III (solid). Left: time autocorrelation; right: time spectra in log-log scale.

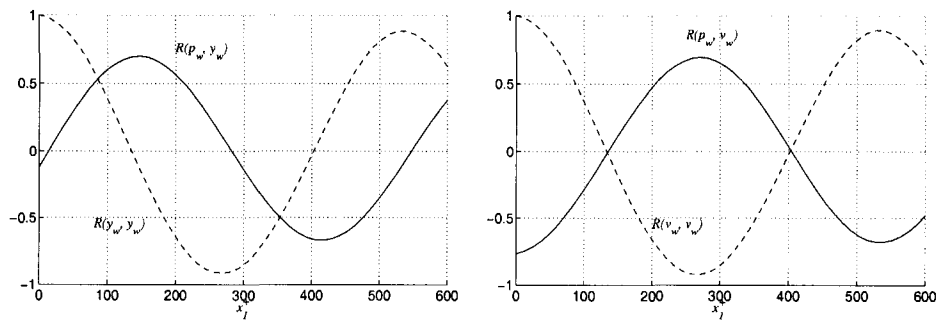


Figure I.23: Streamwise correlations of the wall deformation, velocity and pressure for Case III.

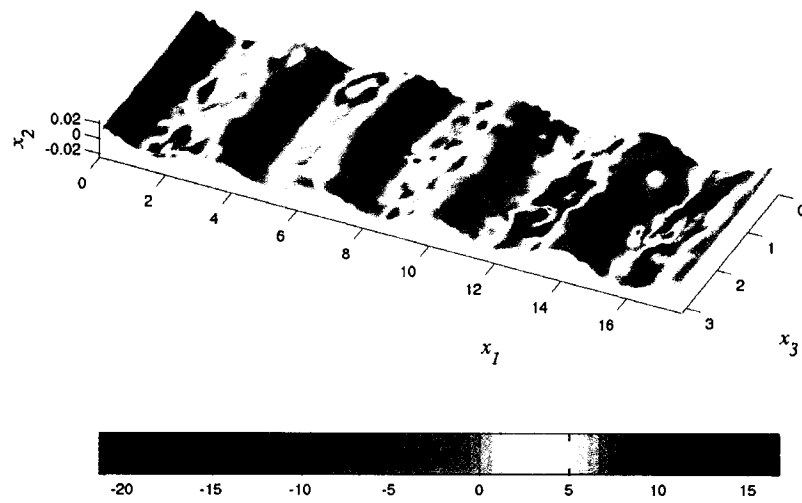
imum at the downstream side of the wave crests, causing downstream wall to elevate. In addition, adverse pressure gradient slows down the flow and therefore reduces the shear stress at downstream side, and sometimes even causes flow separation (Figure I.24(b)). Deformation of the compliant wall has the wave slope $y_w k_x = 0.027$. This is much smaller compared to deformation of the stationary wavy walls discussed in §I.4.4 and comparable to deformation of air-water interface in Fulgosi *et al.* (2003) where it is shown that, at a shear Reynolds number $Re_\tau = 171$, the slow interface motion has only slight effects on the air flow near the interface. In our case, changes to the flow are mainly caused by the resonant motion of the interface.

Near-wall quasi-streamwise vortices are stronger due to the interfacial motion. Figure I.25 visualizes the discriminant² field in the near-wall region where the isosurface level is 10^{-4} . The deformation of wall is indicated by the color bar in the figure. It can be seen that the vortices are more generated above the valleys of the wall.

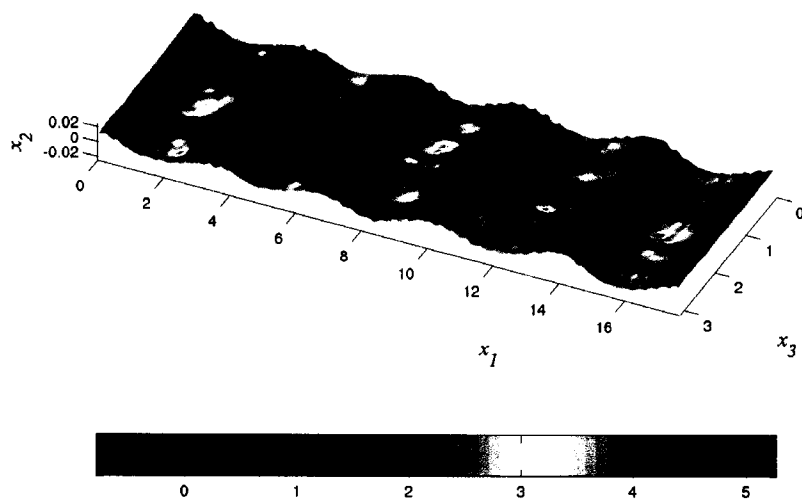
Profiles of root-mean-square (RMS) values of velocity fluctuations, Reynolds stress, and mean velocity of case III are given in Figure I.26, together with the corresponding profiles in regular channel turbulence with the same bulk flow. Note that the lower wall is a moving rough surface, statistics is taken from the planes where ξ_2 is constant. When the wall deformation is small, these planes are approximately parallel to the wall. Near the lower wall, the flow disturbances are increased dramatically due to the wall compliance. $u_{2,rms}$ and $u_{3,rms}$ are more than 20% higher, and $u_{1,rms}$ is slightly higher, than in the regular channel case. The Reynolds stress, $-\langle u_1 u_2 \rangle$, is much larger and therefore produces more turbulent kinetic energy. Near the flat wall, the flow disturbances are barely modified, indicating that compliance of the lower wall has little influence on the upper wall. However, the profile of the mean velocity, $\langle u_1 \rangle$, is distorted on both sides, with lower values on the lower wall side, but higher values on the upper wall side. This means that the flow at the lower part of the channel is significantly slowed down by the high drag on the compliant wall, and that the flow at the upper part is speed up due to the constant total bulk flow constraint. Therefore, it is expected that the skin friction on the upper wall is higher than that in the regular channel case.

Two-point correlation functions of the velocity fluctuations $R(u_i, u_i)$ near the wall ($x_2^+ = 5.6$) region and near the channel center region ($x_2^+ = 118$) for case III are shown in Figure I.27 and Figure I.28, respectively. Their Fourier transform, the energy spectra E_{ii} , are shown in Figure I.29

²A scalar quantity derived from the velocity gradient tensor $A_{ij} = \partial u_i / \partial x_j^+$ which in present work is defined in wall units, the discriminant D of the flow characterizes the local topology of the flow observed in a frame moving with the fluids, and provides a handy identification technique for location of “vortex-type” structures in the turbulent flow. Define the second and third invariants of A to be $Q \triangleq \{[tr(A)]^2 - tr(A^2)\}/2$ and $R \triangleq -det(A)$, and the discriminant is given by $D = (27/4)R^2 + Q^3$. Details may be found in Perry & Chong (1987) and Blackburn, Mansour & Cantwell (1996).

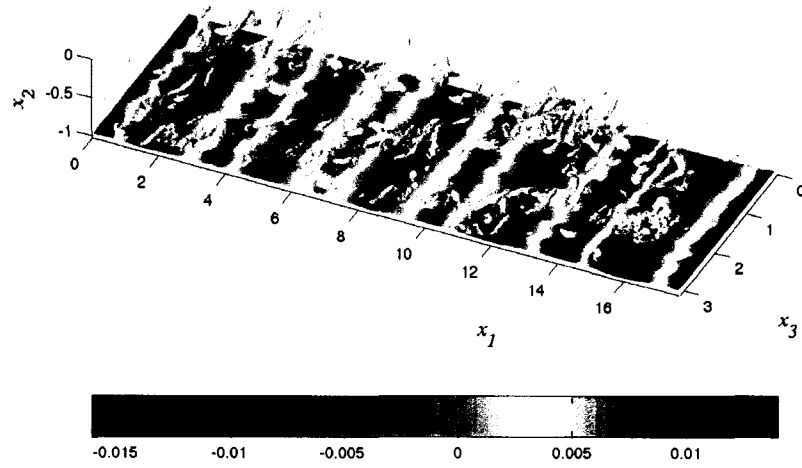


(a) wall pressure

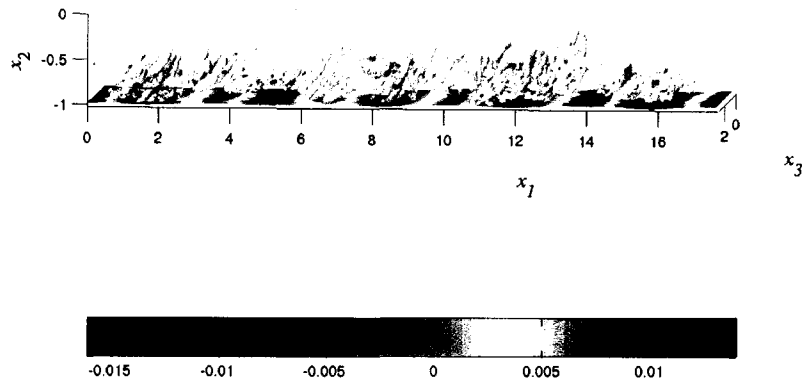


(b) shear stress

Figure I.24: Instantaneous pressure and streamwise shear stress on the interface in case III.



(a) view 1



(b) view 2

Figure I.25: The discriminant of the flow in case III at the isosurface level 10^{-4} . The color bar indicates the amplitude of the interface deformation.

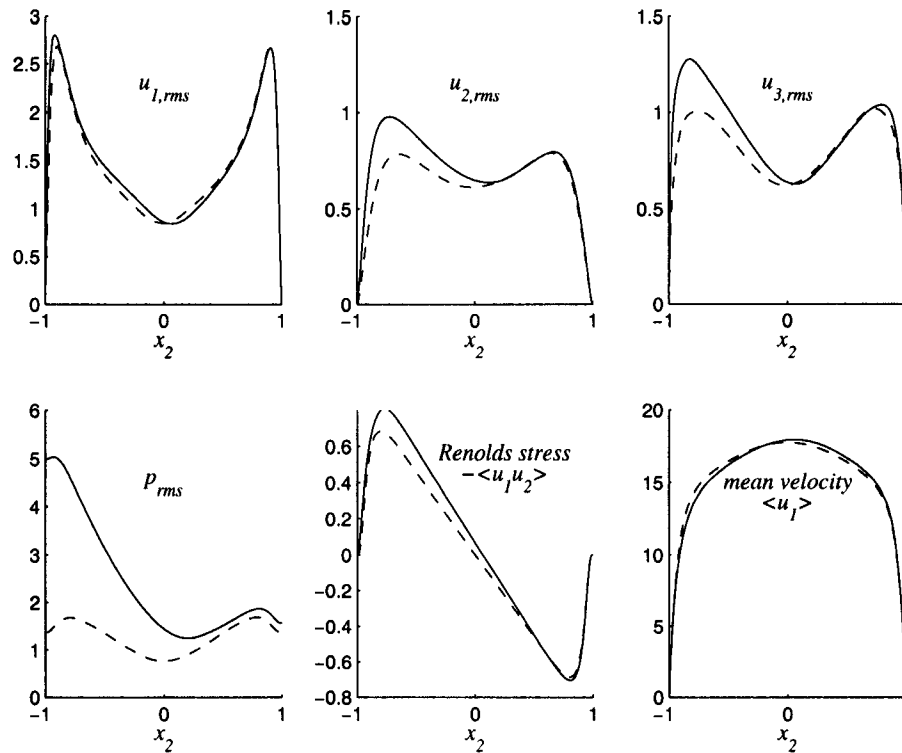


Figure I.26: Profiles of the flow field for case III. Dashed: statistics from the regular channel flow; solid: statistics from the compliant channel.

and Figure I.30, respectively. Near the wall region, the correlations of the velocity fluctuations are clearly modified. $R(u_i, u_i)$ in the streamwise direction monotonically decreases near the wall in the regular channel case, but fluctuates above and below zero due to the sinusoidal wall deformation in case III. The fluctuation mode is indicated by the spikes in their spectra plots, which is the same as the vibration mode of the compliant structure. The spanwise correlations near the wall are greatly lifted by the vertical boundary motion which is nearly uniform in the x_3 direction. Correspondingly, they have more power for all k_z modes as shown in the energy spectra. However, the characteristic length scale of quasi-streamwise vortices seems to be unaffected by the wall motion, since these spanwise correlation functions still maintain their basic shape. In contrast, the wall motion has less influence on the correlations near the channel center, where all the spanwise modes have slightly more power than in the regular channel case and the high streamwise modes are nearly unchanged.

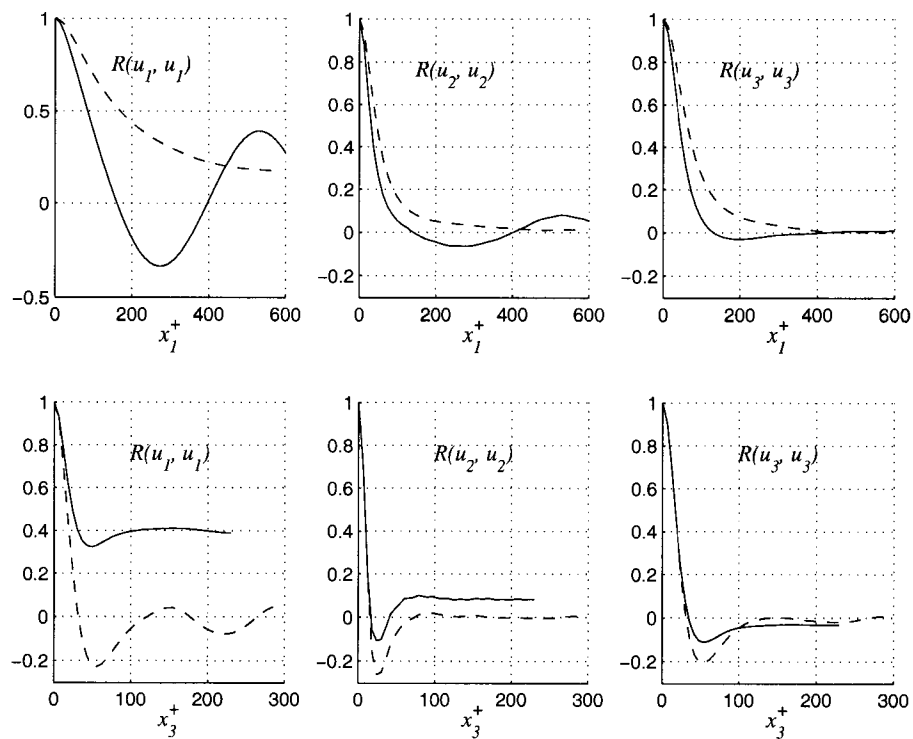


Figure I.27: Streamwise (upper row) and spanwise (down row) two-point correlations of velocities at $x_2^+ = 5.6$ for case III. Dashed: statistics from the regular channel flow; solid: statistics from the compliant channel.

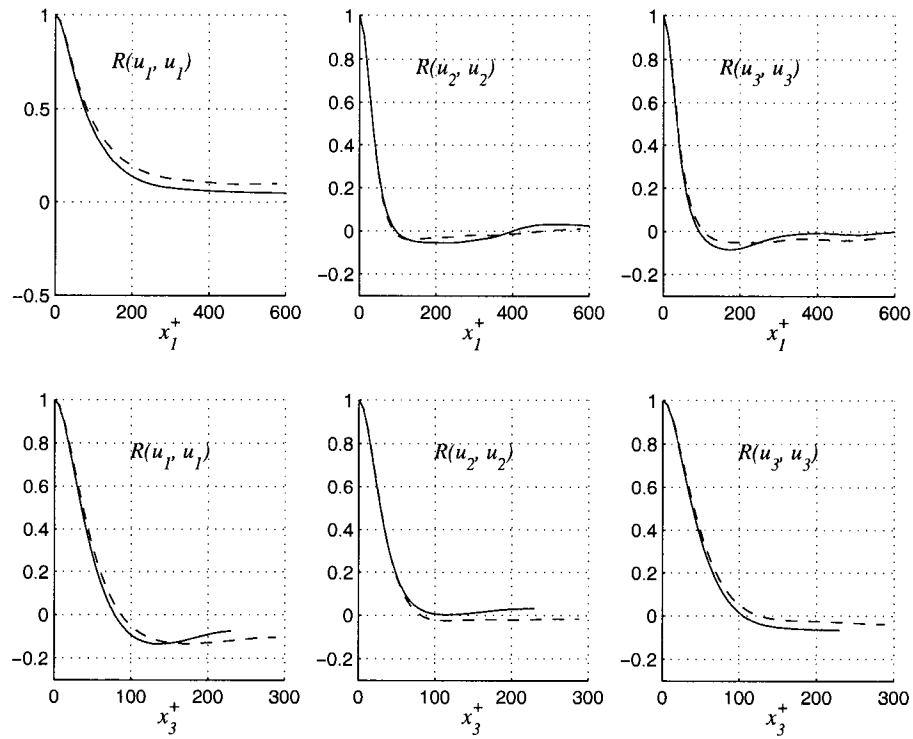


Figure I.28: Streamwise (upper row) and spanwise (down row) two-point correlations of velocities at $x_2^+ = 118$ for case III. Dashed: statistics from the regular channel flow; solid: statistics from the compliant channel.

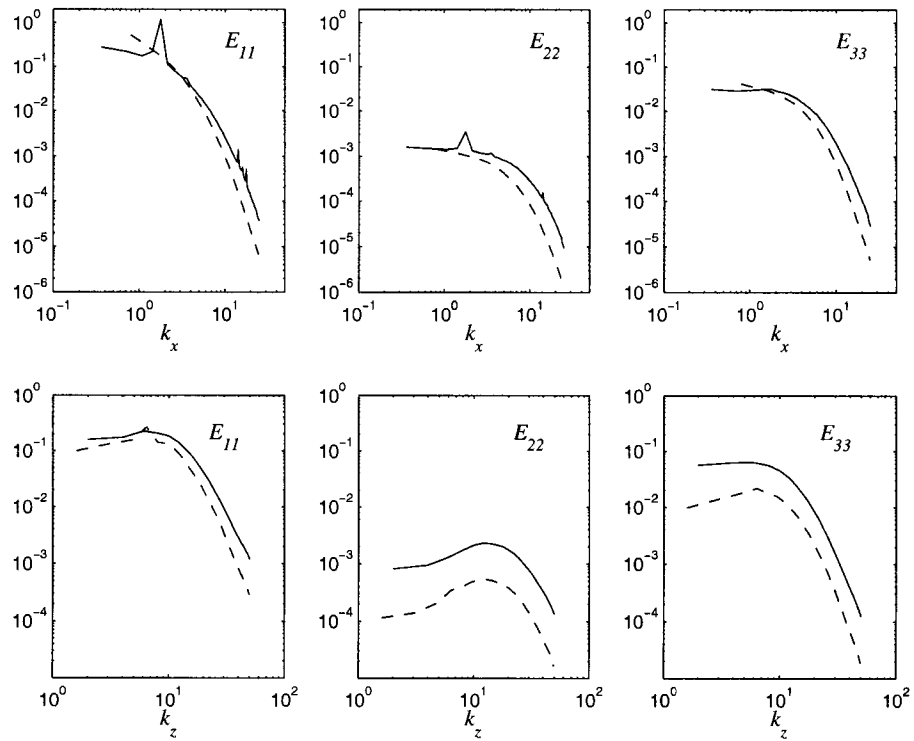


Figure I.29: Streamwise (upper row) and spanwise (down row) one-dimensional spectra at $x_2^+ = 5.6$ for case III. Dashed: statistics from the regular channel flow; solid: statistics from the compliant channel.

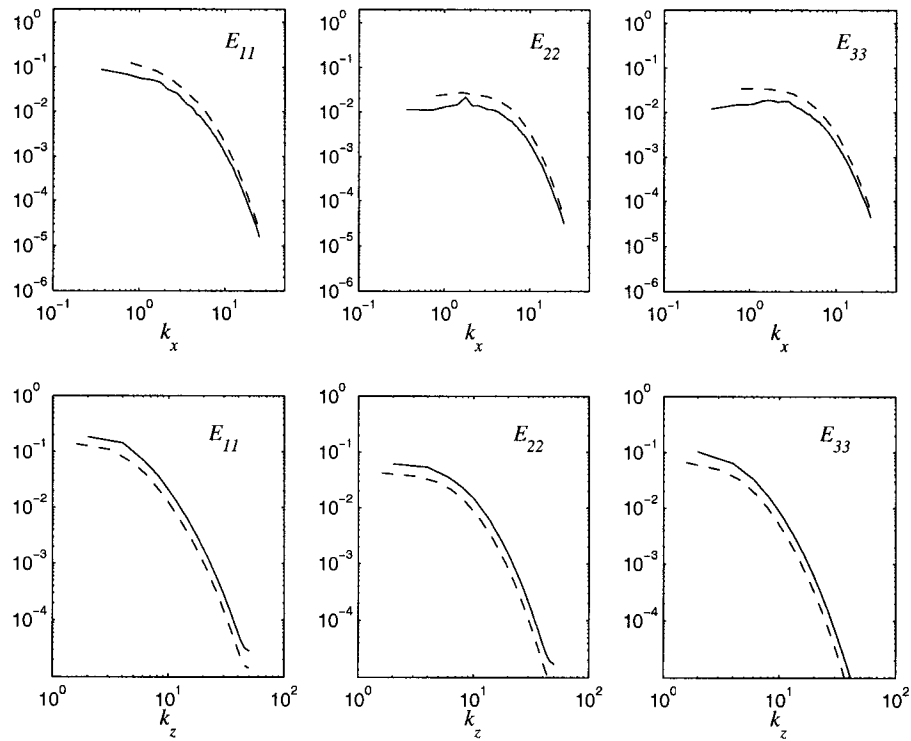


Figure I.30: Streamwise (upper row) and spanwise (down row) one-dimensional spectra at $x_2^+ = 118$ for case III. Dashed: statistics from the regular channel flow; solid: statistics from the compliant channel.

I.7.2 Case VII and VIII

In case VII and VIII where the wall structure is soft and damping is low, the interface forms large streamwise traveling ridges. The wall pressure disturbances, wall velocity, wall shear stress and the discriminant field in the lower channel for case VIII are visualized in Figure I.31 and Figure I.32. In these two cases, the wall deformation has less regular sinusoidal shape than in case II and III. The distribution of the pressure and shear stress disturbances is similar to that in case II and III. The pressure and shear stress are high at the upstream side of the ridges and low at the downstream side. The flow detaches the wall at the downstream side and re-attaches near the bottom of the valleys. The vortex-type motion of the flow in case VIII, indicated by the discriminant isosurfaces, is mainly focused above the valleys and is stronger than that in case III.

From the two-point correlation functions $R(y_w, y_w)$ and their spectra (Figure I.33), the dominant wave mode for both case VII and VIII is $k_x = 1.35$ and the wavelength is $\lambda^+ = 700$, but case VIII has larger wall deformation. From the time correlation functions $Z(t)$ (Figure I.34), the oscillation period of the wall is 0.94 for both cases. Thus the phase traveling speed is $c/u_\tau \approx 5$, slower than that in case II and III.

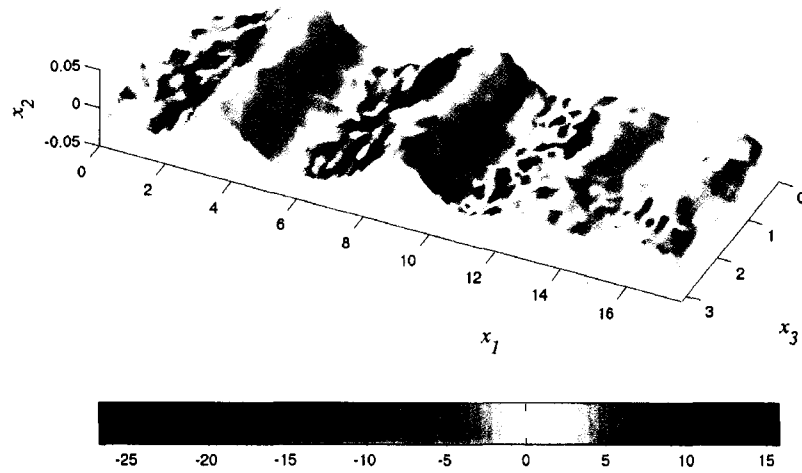
In case VIII, the slope of the wave is about $y_w k_x \approx 0.05$. The form drag in the two cases is 0.013 and 0.065 respectively, both less than one-third of their own total drag increase, so the skin-friction increase is still the main factor for the drag rise.

Profiles of RMS velocity fluctuations, Reynolds stress $-\langle u_1 u_2 \rangle$, and mean velocity $-\langle u_1 \rangle$ of case VIII are given in Figure I.26, together with the corresponding quantities in the regular channel turbulence with the same bulk flow. Case VII has similar profiles but the effects of the compliant surface is less dramatic. Compare to case III, case VIII has larger velocity and pressure fluctuations, and higher Reynolds stress near the compliant wall, and the mean velocity profile is more skewed due to higher drag on the lower wall (interface).

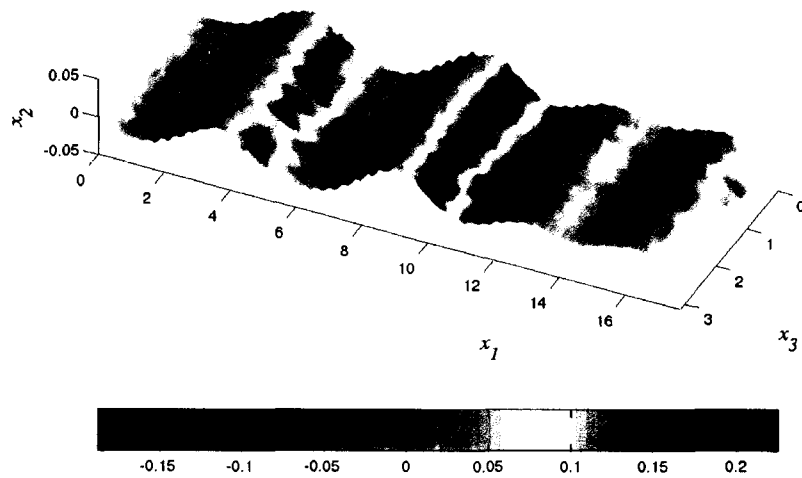
The deformed tensegrity fabrics in case III and case VIII are visualized in Figure I.36. The flow/structure interface displacements are about 3% of the fabric height in case III and about 10% of the fabric height in case VIII.

I.7.3 Dependence on domain size

When studying an infinite but homogeneous system such as turbulence over a compliant surface in the present problem, it facilitates numerical analysis and therefore is a well-adopted

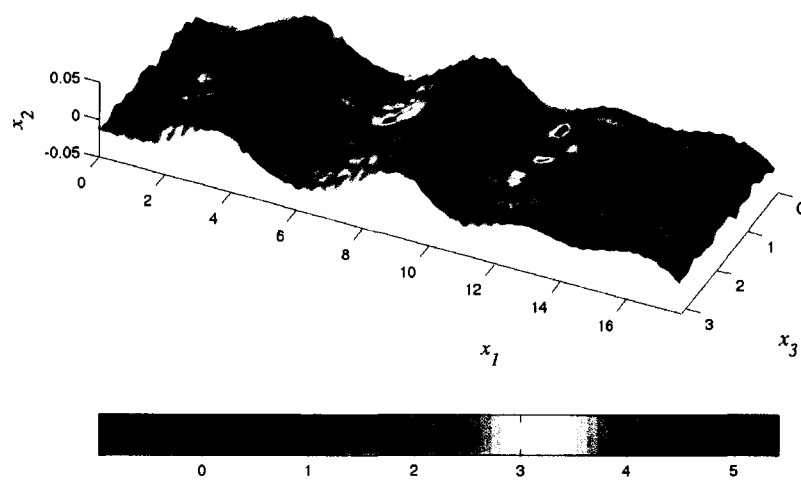


(a) wall pressure

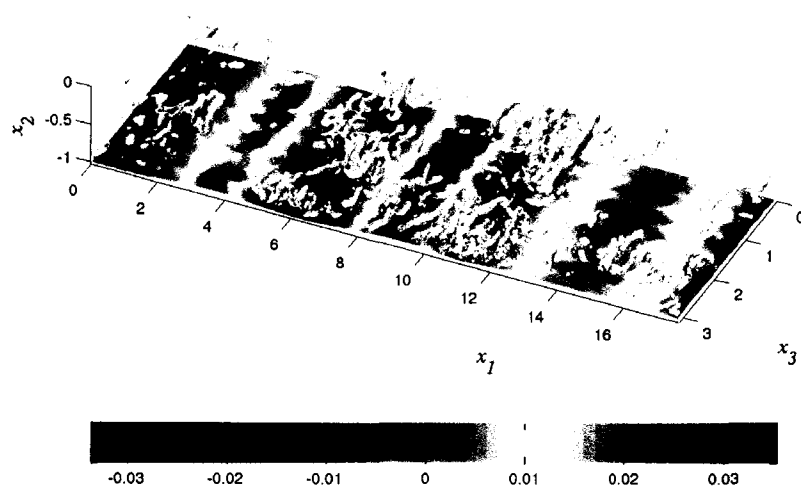


(b) wall velocity

Figure I.31: Visualization of the flow for case VIII. The interface is deformed from its nominal position $x_2 = -1$.



(a) shear stress



(b) discriminant

Figure I.32: Visualization of the flow for case VIII. The interface is deformed from its nominal position $x_2 = -1$. In (d) the isosurface level is 10^{-4} and the color represents of wall deformation.

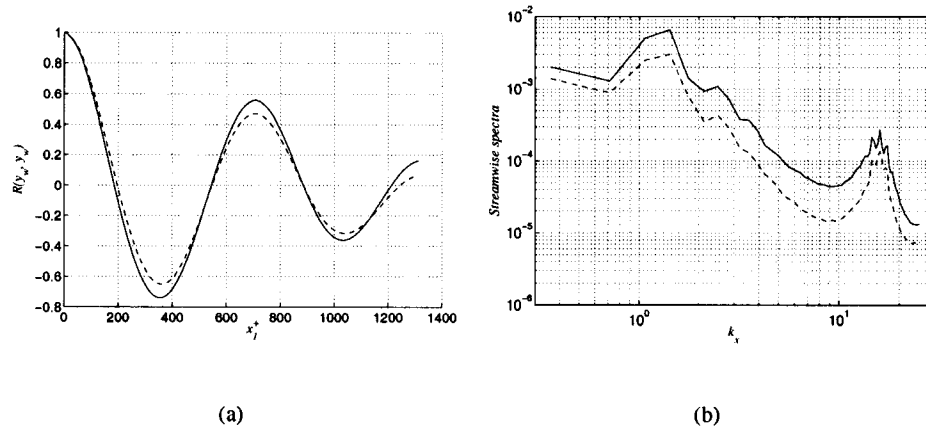


Figure I.33: Spatial characteristics of the wall deformation y_w for case VII and VIII. Left: streamwise autocorrelation; right: streamwise spectra in log-log scale. Dashed: case VII; solid: case VIII.

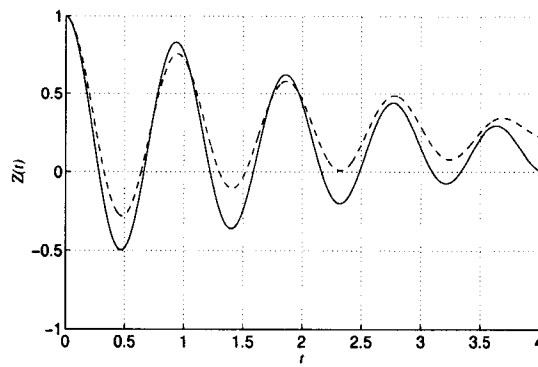


Figure I.34: Time correlation of the wall deformation. Dashed: case VII; solid: case VIII.

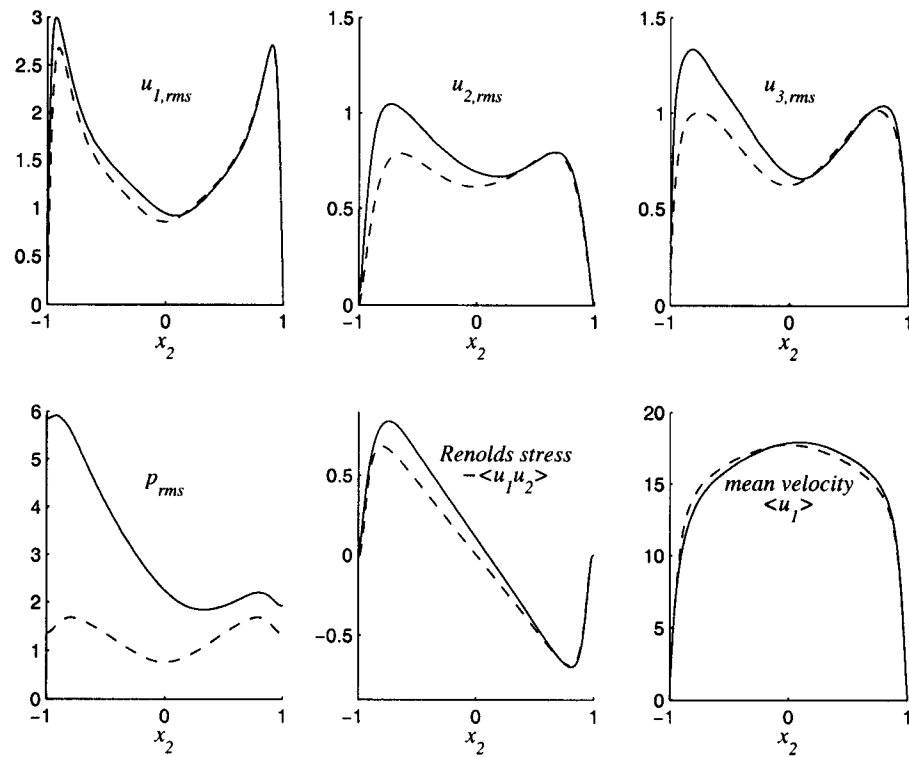
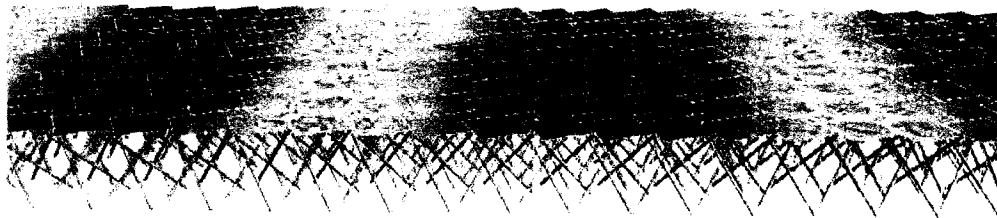
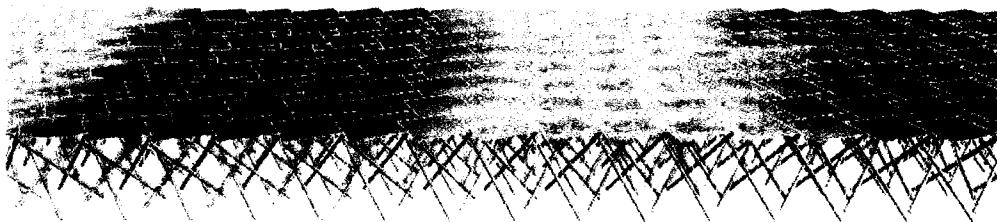


Figure I.35: Profiles of the flow field for case VIII. Dashed: statistics from the regular channel flow; solid: statistics from the compliant channel.



(a) case III



(b) case VIII

Figure I.36: Visualization of the deformed tensegrity fabrics under flow disturbances. Red color represents high elevation of the surface, and blue color represents low elevation. The flow goes from left to right.

L_x	D_w	D_f	$p_{w,rms}$	$v_{w,rms}$	$y_{w,max}^+$	TKE
2.5π	1.74	1.11	9.1	0.19	6.5	5.86
3.75π	1.14	1.02	4.3	0.06	2.0	3.91
5.625π	1.17	1.03	5.0	0.07	2.4	3.95
7.5π	1.17	1.02	5.0	0.07	2.6	3.97

Table I.7: Comparison of statistics from the flow over the compliant surface of case III with various channel sizes.

procedure to cut out a finite block and impose periodic boundary condition. However, one needs to be careful that the finite block has to be large enough so that the artificial periodicity does not change the fundamental physics of the system. In this paper, we have tested four different channel lengths, $L_x = 2.5\pi$, 3.75π , 5.625π , and 7.5π , for in the streamwise direction the compelling flow phenomena are observed. Case III where the compliant structure is resonant under flow perturbations is used for the test. The number of tensegrity unit cells in the x_1 direction is 20, 30, 45, and 60, respectively. Grid spacing is kept the same in these tests to resolve fine scales of the flow. If the walls were rigid and flat, all these channel lengths would be large enough to contain the characteristic scales in present turbulent flow and able to provide sufficient data for accurate statistics. However, without a priori knowledge of length scales of the flow in presence of a compliant surface, it is necessary to ask the question and check if the domain size is sufficiently large.

Streamwise traveling waves of the flow/structure interface exist for all the tests. However, the interface deforms more and oscillates faster when the channel length is small. Selected flow statistics are compared in table I.7. When $L_x = 2.5\pi$, the wave amplitude reaches $y_{w,max}^+ = 6.5$, and the RMS of wall velocity fluctuations is $v_{w,rms} = 0.19$. The form drag is significantly increased and the total drag on the interface is about 74% higher than the regular channel. As the channel length gets larger, the flow statistics is convergent. The explanation for this is that disturbances are spatially correlated more when periodicity is imposed at the edge of a shorter box. In some situations the close-correlated disturbances may incite each other, causing waves to grow as in the case where $L_x = 2.5\pi$.

Streamwise correlation of the interfacial wave and its spectra for all the tests are plotted in Figure I.37. When $L_x = 2.5\pi$ ($L_x^+ = 1178$), the wavelength is $\lambda^+ = 590$, about half of channel length, and the wave spectra are much higher than in the other tests. The three longer boxes contain about 3, 5 and 6 waves, respectively. Their streamwise spectra are very close. The relative error of the wavelength between cases of $L_x = 5.625\pi$ and of $L_x = 7.5\pi$ is less than 4%.

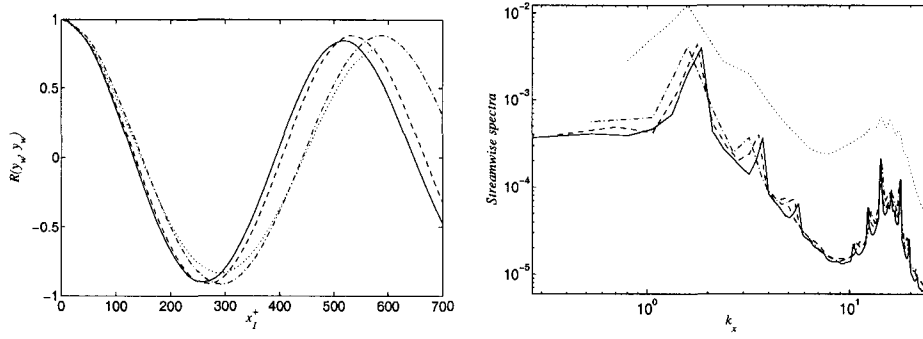


Figure I.37: Comparison of the statistics of the wall deformation y_w in case III for various box lengths. Left: streamwise autocorrelation; right: streamwise spectra in log-log scale. Dotted: $L_x = 2.5\pi$; dash-dotted: $L_x = 3.75\pi$; dashed: $L_x = 5.625\pi$; solid: $L_x = 7.5\pi$. Note that $L_x = 5.625\pi$ appears to be sufficient to obtain approximate convergence of these statistics.

The profiles of RMS velocity fluctuations for the four box lengths are shown in Figure I.38. It can be seen that, except for $L_x = 2.5\pi$, all the other three cases have very close results. Therefore, we conclude that $L_x = 5.625\pi$ is long enough to capture both the interfacial features and the flow field characteristics. Further enlargement of the channel length would lead to expensive computational cost.

I.7.4 Visualization and statistics of the other cases

For completeness, the flow/structure interface displacements in other cases than III and VIII are visualized in Figure I.39 to Figure I.41, and their RMS profiles of velocity fluctuations are given in Figure I.42. Except case II and VII which are discussed in previous sections, in the rest of the cases, I, IV, V, and VI, the wall compliance only produces small interface deformations and has no significant influence on the flow statistics.

I.8 Concluding remarks

The purpose of this paper was to present a new type of structure, tensegrity fabric, as the fundamental paradigm of the compliant coating, to model the dynamical interaction of a turbulent flow of a moderate Reynolds number and the tensegrity fabric, and to investigate the effects of the compliant surface on near-wall turbulence. The final goal of this research is to find the appropriate

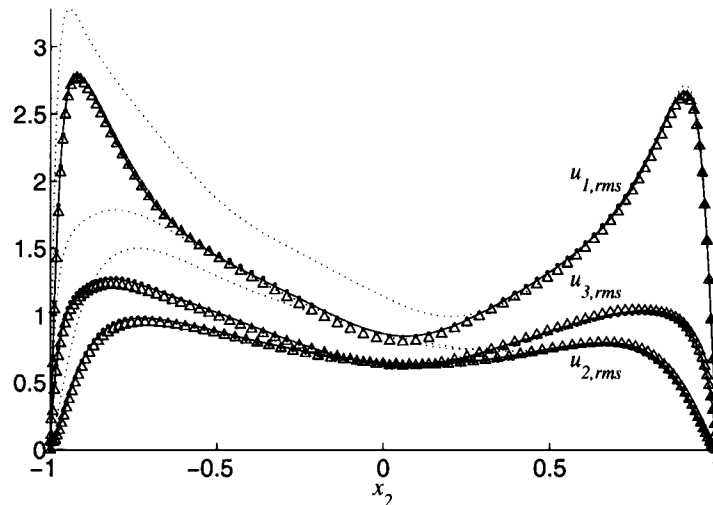
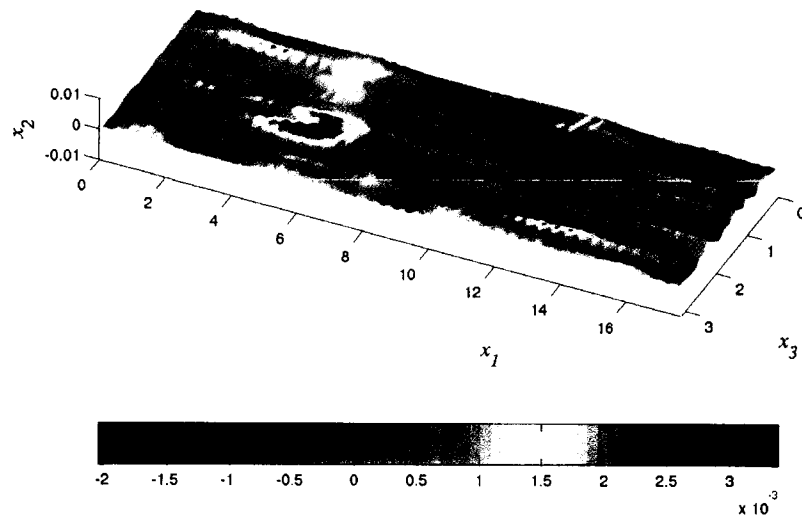


Figure I.38: Comparison of RMS velocity disturbances in case III for various box lengths. Dotted: $L_x = 2.5\pi$; triangle: $L_x = 3.75\pi$; solid: $L_x = 5.625\pi$; point: $L_x = 7.5\pi$. Note that $L_x = 3.75\pi$ appears to be sufficient to obtain approximate convergence of these statistics.

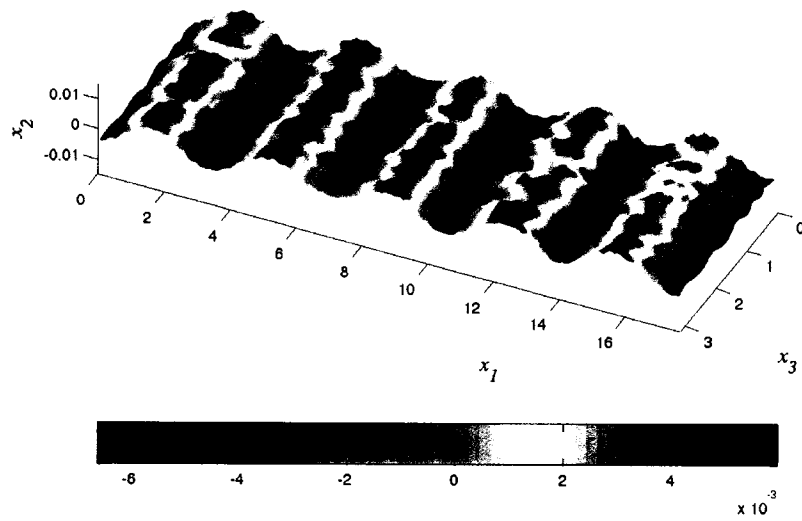
properties of surface compliance that can reduce turbulence-induced drag.

Unlike the spring/plate surface model to which a turbulent flow is not sensitive, the tensegrity-based compliant surface may have significant effects on the near-wall turbulence. This paper performs preliminary study of three main material parameters, the structure's density, stiffness, and damping, that represent the surface's properties. Simulations show that, when the structure's stiffness and damping are low, the interface forms streamwise traveling waves, resembling an air-water interface but convecting with a much faster phase speed. For some combinations of the tensegrity parameters, the compliant wall may be resonant under excitation of flow disturbances. The wavy motion of the interface causes large increase of drag on the wall and turbulent kinetic energy of the flow, and shear stress accounts for major portion of the drag increase.

High sensitivity of the turbulent flow to surface compliance is important for the wall to interact with the flow in a favorable way, although current results seem to point to the contrary to the aim of the research. Besides the spanwise aligned deformations of the flow/structure interface, we have also found that the interface may form streamwise aligned ridges as seen in case V where the wall deformation amplitude and motion are still too small to have significant effect on the flow. This could be promising since it has been found that streamwise aligned wall deformations, such as riblets of proper size (Choi *et al.*, 1993), may have drag reduction effect, and that active spanwise

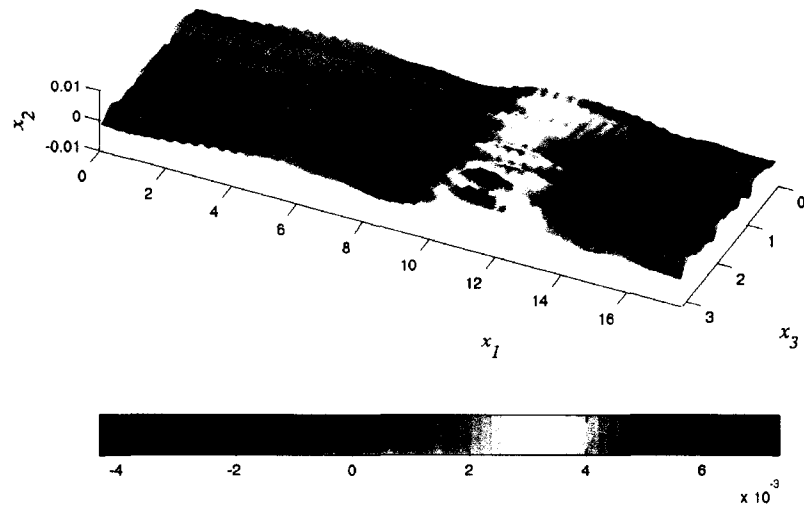


(a) case I

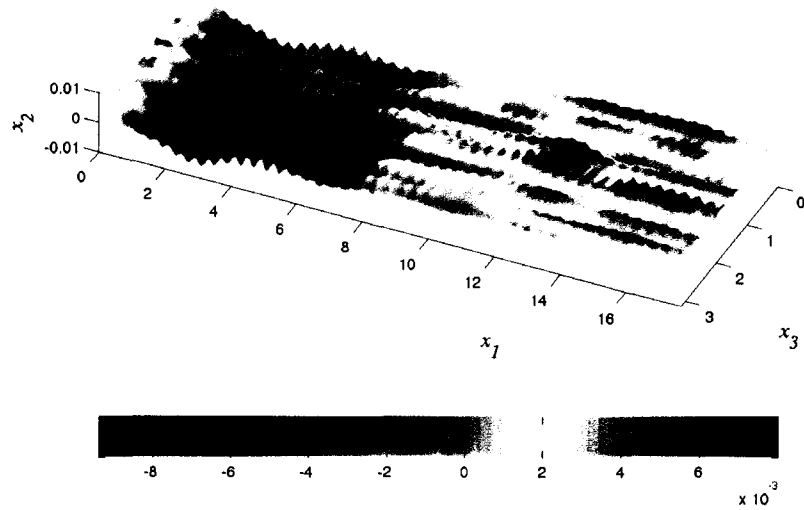


(b) case II

Figure I.39: Visualization of the deformation of the flow/structure interface from the nominal position $x_2 = -1$ for case I and II.

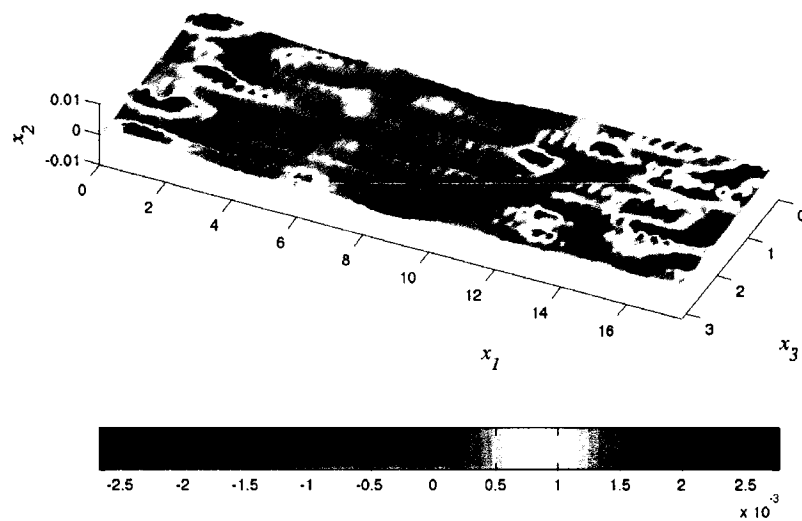


(a) case IV

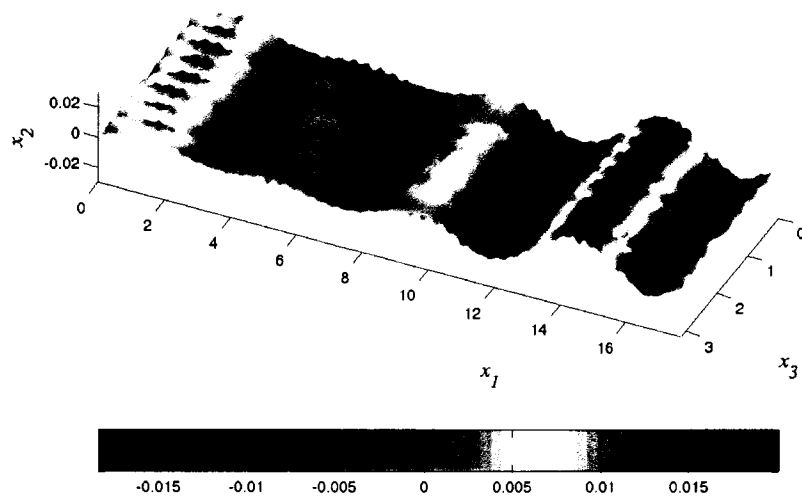


(b) case V

Figure I.40: Visualization of the deformation of the flow/structure interface from the nominal position $x_2 = -1$ for case IV and V.

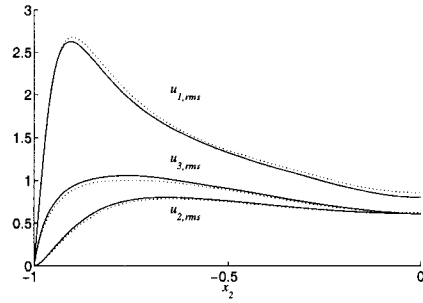


(a) case VI

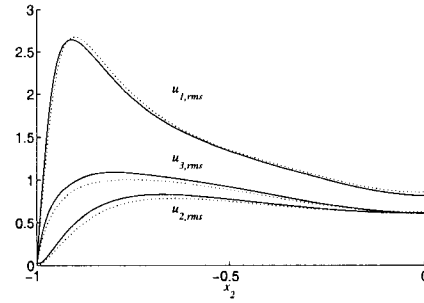


(b) case VII

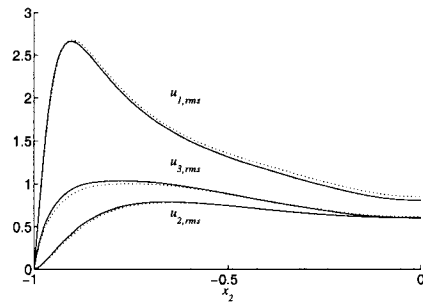
Figure I.41: Visualization of the deformation of the flow/structure interface from the nominal position $x_2 = -1$ for case VI and VII.



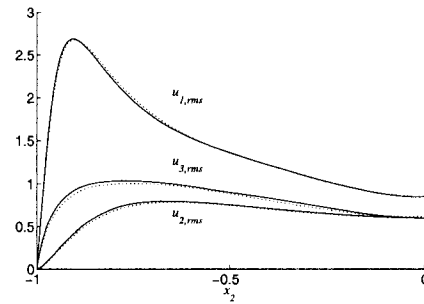
(a) case I



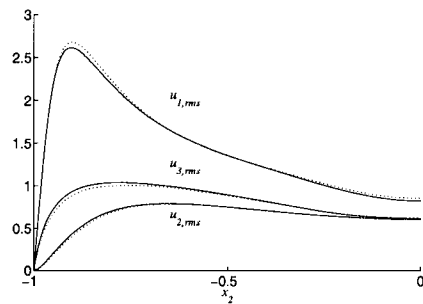
(b) case II



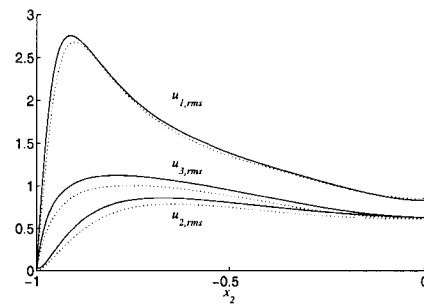
(c) case IV



(d) case V



(e) case VI



(f) case VII

Figure I.42: RMS profiles of velocity fluctuations for the cases other than III and VIII. The corresponding plot for case III is shown in Figure I.26 and for case VIII is shown in Figure I.35. Dotted: statistics from channel flow with solid walls; solid: statistics from the channel flow with compliant walls.

oscillating wall (Jung *et al.*, 1992) may bring similar benefit. In addition, the tensegrity fabric has more design flexibilities that we have not been able to explore yet, e.g., the tensegrity cell's geometry, material properties of each structural elements in a cell. Considering obscure physics behind, our future work is to optimize the tensegrity fabric in order to find its favorable properties.

The software that simulates the dynamics of the flow/structure system couples a Fortran90 code that solves the turbulent flow with a moving boundary using a time-dependent coordinate transformation, and a C++ code that calculates dynamics of a large array of structural elements which are highly unstructured in the computer memory. The current simulation contains 7×10^5 grid points and a structure of 1440 bars and 3653 tendons. A typical run costs about 20 hours on an IBM P655+ (8-way) node. Therefore, an efficient optimization strategy is essential to success.

I.A List of the terms in Eq. (I.7)

- $T_i(q_i)$:

$$\begin{aligned} T_1(q_1) &= \frac{\partial \varphi_\tau q_1}{\partial \xi_2}, \\ T_2(q_2) &= \varphi_\tau \frac{\partial q_2}{\partial \xi_2}, \\ T_3(q_3) &= \frac{\partial \varphi_\tau q_3}{\partial \xi_2}. \end{aligned}$$

- $N_i(q_j)$:

$$\begin{aligned} N_1(q_j) &= \frac{\partial \varphi_2 q_1 q_1}{\partial \xi_1} + \frac{\partial \varphi_1 \varphi_2 q_1 q_1}{\partial \xi_2} + \frac{\partial \varphi_2 q_1 q_2}{\partial \xi_2} + \frac{\partial \varphi_2 q_1 q_3}{\partial \xi_3} + \frac{\partial \varphi_3 \varphi_2 q_1 q_3}{\partial \xi_2}, \\ N_2(q_j) &= \frac{\partial \varphi_2 q_1 q_2}{\partial \xi_1} + \varphi_1 \frac{\partial \varphi_2 q_1 q_2}{\partial \xi_2} + \varphi_2 \frac{\partial q_2 q_2}{\partial \xi_2} + \frac{\partial \varphi_2 q_2 q_3}{\partial \xi_3} + \varphi_3 \frac{\partial \varphi_2 q_2 q_3}{\partial \xi_2}, \\ N_3(q_j) &= \frac{\partial \varphi_2 q_3 q_1}{\partial \xi_1} + \frac{\partial \varphi_1 \varphi_2 q_3 q_1}{\partial \xi_2} + \frac{\partial \varphi_2 q_3 q_2}{\partial \xi_2} + \frac{\partial \varphi_2 q_3 q_3}{\partial \xi_3} + \frac{\partial \varphi_3 \varphi_2 q_3 q_3}{\partial \xi_2}. \end{aligned}$$

- $G_i(\bar{p})$:

$$\begin{aligned} G_1(\bar{p}) &= \frac{\partial \bar{p}}{\partial \xi_1} + \frac{\partial \varphi_1 \bar{p}}{\partial \xi_2}, \\ G_2(\bar{p}) &= \varphi_2^2 \frac{\partial \bar{p}}{\partial \xi_2}, \\ G_3(\bar{p}) &= \frac{\partial \bar{p}}{\partial \xi_3} + \frac{\partial \varphi_3 \bar{p}}{\partial \xi_2}. \end{aligned}$$

• $L_i(q_i)$:

$$\begin{aligned} L_1(q_1) &= \left(\frac{\partial}{\partial \xi_1} + \frac{\partial \varphi_{1\cdot}}{\partial \xi_2} \right)^2 q_1 + \varphi_2^2 \frac{\partial^2}{\partial \xi_2^2} q_1 + \left(\frac{\partial}{\partial \xi_3} + \frac{\partial \varphi_{3\cdot}}{\partial \xi_2} \right)^2 q_1, \\ L_2(q_2) &= \left(\frac{\partial}{\partial \xi_1} + \varphi_1 \frac{\partial}{\partial \xi_2} \right)^2 q_2 + \varphi_2^2 \frac{\partial^2}{\partial \xi_2^2} q_2 + \left(\frac{\partial}{\partial \xi_3} + \varphi_3 \frac{\partial}{\partial \xi_2} \right)^2 q_2, \\ L_3(q_3) &= \left(\frac{\partial}{\partial \xi_1} + \frac{\partial \varphi_{1\cdot}}{\partial \xi_2} \right)^2 q_3 + \varphi_2^2 \frac{\partial^2}{\partial \xi_2^2} q_3 + \left(\frac{\partial}{\partial \xi_3} + \frac{\partial \varphi_{3\cdot}}{\partial \xi_2} \right)^2 q_3. \end{aligned}$$

Note that $\tilde{p} = Jp/\rho$ is the modified pressure.

The continuous version of the Laplacian operator $\mathcal{L}(\tilde{p})$ for solving the pressure equation has the same form as $L_1(\cdot)$ and $L_3(\cdot)$,

$$\mathcal{L}(\tilde{p}) = D[G_i(\tilde{p})] = \left(\frac{\partial}{\partial \xi_1} + \frac{\partial \varphi_{1\cdot}}{\partial \xi_2} \right)^2 \tilde{p} + \varphi_2^2 \frac{\partial^2 \tilde{p}}{\partial \xi_2^2} + \left(\frac{\partial}{\partial \xi_3} + \frac{\partial \varphi_{3\cdot}}{\partial \xi_2} \right)^2 \tilde{p}. \quad (\text{I.27})$$

We distinguish $\mathcal{L}(\cdot)$ from $L_i(\cdot)$ because they are discretized differently as seen in section I.3.1.

I.B Details of solving Eq. I.11

The explicit operators in Eq. I.11 are listed as follows:

$$\begin{aligned}
A_1(q_j) &= v \frac{\delta}{\delta \xi_1} \left(\frac{\delta q_1}{\delta \xi_1} + \frac{\delta \varphi_1 q_1}{\delta \xi_2} \right) + v \frac{\delta \varphi_1}{\delta \xi_2} \left(\frac{\delta q_1}{\delta \xi_1} \right) \\
&\quad + v \frac{\delta}{\delta \xi_3} \left(\frac{\delta q_1}{\delta \xi_3} + \frac{\delta \varphi_3 q_1}{\delta \xi_2} \right) + v \frac{\delta \varphi_3}{\delta \xi_2} \left(\frac{\delta q_1}{\delta \xi_3} \right) \\
&\quad - \frac{\delta \varphi_2 q_1 q_1}{\delta \xi_1} - \frac{\delta \varphi_2 q_1 q_3}{\delta \xi_3} - \frac{\delta \varphi_3 \varphi_2 q_1 q_3}{\delta \xi_2}, \\
A_2(q_j) &= v \frac{\delta}{\delta \xi_1} \left(\frac{\delta q_2}{\delta \xi_1} + \varphi_1 \frac{\delta q_2}{\delta \xi_2} \right) + v \varphi_1 \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_2}{\delta \xi_1} \right) \\
&\quad + v \frac{\delta}{\delta \xi_3} \left(\frac{\delta q_2}{\delta \xi_3} + \varphi_3 \frac{\delta q_2}{\delta \xi_2} \right) + v \varphi_3 \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_2}{\delta \xi_3} \right) \\
&\quad - \frac{\delta \varphi_2 q_1 q_2}{\delta \xi_1} - \frac{\delta \varphi_2 q_3 q_2}{\delta \xi_3} - \varphi_1 \frac{\delta \varphi_2 q_1 q_2}{\delta \xi_2} - \varphi_3 \frac{\delta \varphi_2 q_3 q_2}{\delta \xi_2}, \\
A_3(q_j) &= v \frac{\delta}{\delta \xi_1} \left(\frac{\delta q_3}{\delta \xi_1} + \frac{\delta \varphi_1 q_3}{\delta \xi_2} \right) + v \frac{\delta \varphi_1}{\delta \xi_2} \left(\frac{\delta q_3}{\delta \xi_1} \right) \\
&\quad + v \frac{\delta}{\delta \xi_3} \left(\frac{\delta q_3}{\delta \xi_3} + \frac{\delta \varphi_3 q_3}{\delta \xi_2} \right) + v \frac{\delta \varphi_3}{\delta \xi_2} \left(\frac{\delta q_3}{\delta \xi_3} \right) \\
&\quad - \frac{\delta \varphi_2 q_1 q_3}{\delta \xi_1} - \frac{\delta \varphi_2 q_3 q_3}{\delta \xi_3} - \frac{\delta \varphi_1 \varphi_2 q_1 q_3}{\delta \xi_2}, \\
B_1(q_j) &= v \frac{\delta \varphi_1}{\delta \xi_2} \left(\frac{\delta \varphi_1 q_1}{\delta \xi_2} \right) + v \varphi_2^2 \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_1}{\delta \xi_2} \right) + v \frac{\delta \varphi_3}{\delta \xi_2} \left(\frac{\delta \varphi_3 q_1}{\delta \xi_2} \right) \\
&\quad - \frac{\delta \varphi_1 \varphi_2 q_1 q_1}{\delta \xi_2} - \frac{\delta \varphi_2 q_1 q_2}{\delta \xi_2} - \frac{\delta \varphi_\tau q_1}{\delta \xi_2}, \\
B_2(q_j) &= v \varphi_1 \frac{\delta}{\delta \xi_2} \left(\varphi_1 \frac{\delta q_2}{\delta \xi_2} \right) + v \varphi_2^2 \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_2}{\delta \xi_2} \right) + v \varphi_3 \frac{\delta}{\delta \xi_2} \left(\varphi_3 \frac{\delta q_2}{\delta \xi_2} \right) \\
&\quad - \varphi_2 \frac{\delta q_2 q_2}{\delta \xi_2} - \varphi_\tau \frac{\delta q_2}{\delta \xi_2}, \\
B_3(q_j) &= v \frac{\delta \varphi_1}{\delta \xi_2} \left(\frac{\delta \varphi_1 q_3}{\delta \xi_2} \right) + v \varphi_2^2 \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_3}{\delta \xi_2} \right) + v \frac{\delta \varphi_3}{\delta \xi_2} \left(\frac{\delta \varphi_3 q_3}{\delta \xi_2} \right) \\
&\quad - \frac{\delta \varphi_2 q_3 q_2}{\delta \xi_2} - \frac{\delta \varphi_3 \varphi_2 q_3 q_3}{\delta \xi_2} - \frac{\delta \varphi_\tau q_3}{\delta \xi_2}.
\end{aligned}$$

The Runge-Kutta coefficients used in the present computations are:

$$\begin{aligned}
\beta_1 &= \frac{4}{15}, & \beta_2 &= \frac{1}{15}, & \beta_3 &= \frac{1}{6}, \\
\gamma_1 &= \frac{8}{15}, & \gamma_2 &= \frac{5}{12}, & \gamma_3 &= \frac{3}{4}, \\
\zeta_1 &= 0, & \zeta_2 &= -\frac{17}{60}, & \zeta_3 &= -\frac{5}{12}.
\end{aligned}$$

Re-organize the linearized equation (I.11) so that its left-hand-side contains only terms computed implicitly and its right-hand-side contains only terms computed explicitly,

$$\{1 + \beta_k \Delta t M_2 - \nu \beta_k \Delta t Q_2\} q_2^* = R_2, \quad (\text{I.28})$$

$$\{1 + \beta_k \Delta t M_1 - \nu \beta_k \Delta t Q_1\} q_1^* = R_1, \quad (\text{I.29})$$

$$\{1 + \beta_k \Delta t M_3 - \nu \beta_k \Delta t Q_3\} q_3^* = R_3, \quad (\text{I.30})$$

which are solved for the intermediate field $\mathbf{q}^*$ in the order of q_2^* , q_1^* and q_3^* . M_i and Q_i are the tridiagonal systems given by

$$\begin{aligned} M_1 &= \varphi_2^k \left(2 \frac{\delta \varphi_1^k q_1^{k-1}}{\delta \xi_2} + \frac{\delta q_2^*}{\delta \xi_2} \right) + \frac{\delta \varphi_\tau^k}{\delta \xi_2}, \\ M_2 &= 2 \varphi_2^k \frac{\delta q_2^{k-1}}{\delta \xi_2} + \varphi_\tau^k \frac{\delta}{\delta \xi_2}, \\ M_3 &= \varphi_2^k \left(\frac{\delta \varphi_1^k q_1^*}{\delta \xi_2} + \frac{\delta q_2^*}{\delta \xi_2} + 2 \frac{\delta \varphi_3^k q_3^{k-1}}{\delta \xi_2} \right) + \frac{\delta \varphi_\tau^k}{\delta \xi_2}, \end{aligned}$$

and

$$\begin{aligned} Q_1 &= \frac{\delta \varphi_1^k}{\delta \xi_2} \left(\frac{\delta \varphi_1^k}{\delta \xi_2} \right) + (\varphi_2^k)^2 \frac{\delta}{\delta \xi_2} \left(\frac{\delta}{\delta \xi_2} \right) + \frac{\delta \varphi_3^k}{\delta \xi_2} \left(\frac{\delta \varphi_3^k}{\delta \xi_2} \right), \\ Q_2 &= \varphi_1^k \frac{\delta}{\delta \xi_2} \left(\varphi_1^k \frac{\delta}{\delta \xi_2} \right) + (\varphi_2^k)^2 \frac{\delta}{\delta \xi_2} \left(\frac{\delta}{\delta \xi_2} \right) + \varphi_3^k \frac{\delta}{\delta \xi_2} \left(\varphi_3^k \frac{\delta}{\delta \xi_2} \right), \\ Q_3 &= \frac{\delta \varphi_1^k}{\delta \xi_2} \left(\frac{\delta \varphi_1^k}{\delta \xi_2} \right) + (\varphi_2^k)^2 \frac{\delta}{\delta \xi_2} \left(\frac{\delta}{\delta \xi_2} \right) + \frac{\delta \varphi_3^k}{\delta \xi_2} \left(\frac{\delta \varphi_3^k}{\delta \xi_2} \right). \end{aligned}$$

The right hand side terms, R_i , are given by

$$\begin{aligned} R_1 &= q_1^{k-1} + \gamma_k \Delta t A_1^{k-1} (q_j^{k-1}) + \zeta_k \Delta t A_1^{k-2} (q_j^{k-2}) - 2\beta_k \Delta t \left(G_1^k (\tilde{p}^{k-1}) + J^k P_x \right) \\ &\quad + \nu \beta_k \Delta t \left[\frac{\delta \varphi_1^{k-1}}{\delta \xi_2} \left(\frac{\delta \varphi_1^{k-1} q_1^{k-1}}{\delta \xi_2} \right) + (\varphi_2^k)^{k-1} \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_1^{k-1}}{\delta \xi_2} \right) + \frac{\delta \varphi_3^{k-1}}{\delta \xi_2} \left(\frac{\delta \varphi_3^{k-1} q_1^{k-1}}{\delta \xi_2} \right) \right] \\ &\quad + \beta_k \Delta t \left[\frac{\delta \left(\varphi_2^k \varphi_1^k - \varphi_2^{k-1} \varphi_1^{k-1} \right) q_1^{k-1} q_1^{k-1}}{\delta \xi_2} - \varphi_2^{k-1} \frac{\delta q_1^{k-1} q_2^{k-1}}{\delta \xi_2} - \frac{\delta \varphi_\tau^{k-1} q_1^{k-1}}{\delta \xi_2} \right], \end{aligned}$$

$$\begin{aligned} R_2 &= q_2^{k-1} + \gamma_k \Delta t A_2^{k-1} (q_j^{k-1}) + \zeta_k \Delta t A_2^{k-2} (q_j^{k-2}) - 2\beta_k \Delta t G_2^k (\tilde{p}^{k-1}) \\ &\quad + \nu \beta_k \Delta t \left[\varphi_1^{k-1} \frac{\delta}{\delta \xi_2} \left(\varphi_1^{k-1} \frac{\delta q_2^{k-1}}{\delta \xi_2} \right) + (\varphi_2^k)^2 \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_2^{k-1}}{\delta \xi_2} \right) + \varphi_3^{k-1} \frac{\delta}{\delta \xi_2} \left(\varphi_3^{k-1} \frac{\delta q_2^{k-1}}{\delta \xi_2} \right) \right] \\ &\quad + \beta_k \Delta t \left[(\varphi_2^k - \varphi_2^{k-1}) \frac{\delta q_2^{k-1} q_2^{k-1}}{\delta \xi_2} - \varphi_\tau^{k-1} \frac{\delta q_2^{k-1}}{\delta \xi_2} \right], \end{aligned}$$

and

$$\begin{aligned}
 R_3 = & q_3^{k-1} + \gamma_k \Delta t A_3^{k-1} (q_j^{k-1}) + \zeta_k \Delta t A_3^{k-2} (q_j^{k-2}) - 2\beta_k \Delta t G_3^k (\tilde{p}^{k-1}) \\
 & + \nu \beta_k \Delta t \left[\frac{\delta \varphi_1^{k-1}}{\delta \xi_2} \left(\frac{\delta \varphi_1^{k-1} q_3^{k-1}}{\delta \xi_2} \right) + (\varphi_2^2)^{k-1} \frac{\delta}{\delta \xi_2} \left(\frac{\delta q_3^{k-1}}{\delta \xi_2} \right) + \frac{\delta \varphi_3^{k-1}}{\delta \xi_2} \left(\frac{\delta \varphi_3^{k-1} q_3^{k-1}}{\delta \xi_2} \right) \right] \\
 & + \beta_k \Delta t \left[-\varphi_2^{k-1} \frac{\delta q_2^{k-1} q_3^{k-1}}{\delta \xi_2} + \frac{\delta \left(\varphi_2^k \varphi_3^k - \varphi_2^{k-1} \varphi_3^{k-1} \right) q_3^{k-1} q_3^{k-1}}{\delta \xi_2} - \frac{\delta \varphi_1^{k-1} q_3^{k-1}}{\delta \xi_2} \right].
 \end{aligned}$$

I.C Boundary conditions on the additional pressure update equation solved at the beginning of each timestep

As described in §I.3.3, an additional pressure update is performed at the beginning of each timestep to account for the change in the wall velocity. We now derive the boundary conditions on this additional pressure update equation, demonstrating the procedure for the lower wall only. We begin by approximating the mixed RK/CN march over the first RK substep with an explicit Euler scheme, in which the discrete momentum equation for the first RK substep may be written

$$\begin{cases} q_i^* = q_i^{k-1} + 2\beta_k \Delta t \left[-T_i^{k-1} - N_i^{k-1} + \nu L_i^{k-1} - G_i^{k-1} (\tilde{p}_{\text{updated}}^{k-1}) - J^{k-1} P_x^{k-1} \delta_{il} \right] & \text{on interior,} \\ q_i^* = q_i^k \text{ (specified)} & \text{on walls,} \end{cases} \quad (\text{I.31})$$

where $k = 1$. We then determine the pressure $\tilde{p}_{\text{updated}}^{k-1}$ that is necessary to make q_i^* divergence free in this approximation, and finally proceed with the mixed RK/CN march over the three RK substeps with the standard fractional-step algorithm. Note that the Laplacian of the updated pressure $\tilde{p}_{\text{updated}}^{k-1}$ is identical to the Laplacian of the pressure before the update on the interior of the domain (that is, $D^{k-1}(G_i^{k-1}(\tilde{p}_{\text{updated}}^{k-1})) = D^{k-1}(G_i^{k-1}(\tilde{p}^{k-1}))$), though the boundary conditions on the equations defining the pressure are different.

We begin by applying the discrete divergence operator (I.9) evaluated at $j = \frac{1}{2}$ and RK time level $k-1$,

$$D^{k-1}(q_i)_{\frac{1}{2}} = \frac{\delta_s q_{1,\frac{1}{2}}}{\delta \xi_1} + \frac{\left(\varphi_{1,1}^{k-1} \bar{q}_{1,1} + q_{2,1} + \varphi_{3,1}^{k-1} \bar{q}_{3,1} \right) - q_{2,0}}{\Delta \xi_{2,\frac{1}{2}}} + \frac{\delta_s q_{3,\frac{1}{2}}}{\delta \xi_3},$$

to Eq. (I.31). Then we may write

$$\begin{aligned} \frac{D^{k-1}(q_i^*)_{\frac{1}{2}} - D^{k-1}(q_i^{k-1})_{\frac{1}{2}}}{2\beta_k \Delta t} &= D^{k-1} \left[-T_i^{k-1} - N_i^{k-1} - J^{k-1} P_x^{k-1} \delta_{il} \right]_{\frac{1}{2}} - D^{k-1}(G_i^{k-1})_{\frac{1}{2}} \\ &+ \frac{1}{\Delta \xi_{2,\frac{1}{2}}} \left[-\frac{q_{2,0}^k - q_{2,0}^{k-1}}{2\beta_k \Delta t} - \tilde{T}_{2,0}^{k-1} - \tilde{N}_{2,0}^{k-1} + \nu \tilde{L}_{2,0}^{k-1} - \tilde{G}_{2,0}^{k-1} \right] \\ &- \frac{1}{\Delta \xi_{2,\frac{1}{2}}} J^{k-1} P_x^{k-1} \phi_{1,0}^{k-1}, \end{aligned} \quad (\text{I.32a})$$

where

$$\tilde{T}_{2,0}^{k-1} = \phi_{1,0}^{k-1} T_{1,0}^{k-1} + T_{2,0}^{k-1} + \phi_{3,0}^{k-1} T_{3,0}^{k-1}, \quad (\text{I.32b})$$

$$\tilde{N}_{2,0}^{k-1} = \phi_{1,0}^{k-1} N_{1,0}^{k-1} + N_{2,0}^{k-1} + \phi_{3,0}^{k-1} N_{3,0}^{k-1}, \quad (\text{I.32c})$$

$$\tilde{G}_{2,0}^{k-1} = \phi_{1,0}^{k-1} G_{1,0}^{k-1} + G_{2,0}^{k-1} + \phi_{3,0}^{k-1} G_{3,0}^{k-1}, \quad (\text{I.32d})$$

$$\tilde{L}_{2,0}^{k-1} = \phi_{1,1}^{k-1} \bar{L}_{1,1}^{k-1} + L_{2,1}^{k-1} + \phi_{3,1}^{k-1} \bar{L}_{3,1}^{k-1} + \Delta \xi_{2,\frac{1}{2}} \left(\frac{\delta_s L_{1,\frac{1}{2}}^{k-1}}{\delta \xi_1} + \frac{\delta_s L_{3,\frac{1}{2}}^{k-1}}{\delta \xi_3} \right), \quad (\text{I.32e})$$

where $\bar{L}_{2,0}^{k-1}$ represents the extrapolation of the viscous term at the boundary by using its divergence-free property. Note that by the metric invariants of the coordinate transformation,

$$D^{k-1}(J^{k-1} P_x^{k-1} \delta_{il})_{\frac{1}{2}} = 0,$$

and by the continuity constraint

$$D^{k-1}(q_i^{k-1})_{\frac{1}{2}} = 0.$$

We require

$$D^k(q_i^*)_{\frac{1}{2}} \approx D^{k-1}(q_i^*)_{\frac{1}{2}} + D^{k-1}(T_i^{k-1}) = 0.$$

We also notice that, from the previous timestep,

$$D^{k-1}(G_i^{k-1})_{\frac{1}{2}} + D^{k-1}(N_i^{k-1})_{\frac{1}{2}} = 0.$$

Now what is left in (I.32a) is a projection of the momentum equation onto the wall normal vector $\mathbf{n}_2 = (J\phi_1 \quad 1 \quad J\phi_3)^T$, i.e., a discretized form of (I.15), and may be rewritten as

$$\tilde{G}_{2,0}^{k-1}(\tilde{p}_{\text{updated}}^{k-1}) = -\frac{q_{2,0}^k - q_{2,0}^{k-1}}{2\beta_k \Delta t} - \tilde{T}_{2,0}^{k-1} - \tilde{N}_{2,0}^{k-1} + \nu \tilde{L}_{2,0}^{k-1} - P_{2,0}^{k-1}, \quad (\text{I.33})$$

where $P_{2,0}^{k-1} = J^{k-1} P_x^{k-1} \phi_{1,0}^{k-1}$.

Note that, in (I.33), $\tilde{L}_{2,0}$ is calculated from (I.32e), and $\tilde{T}_{2,0}$ and $\tilde{N}_{2,0}$ may be simplified and calculated using the velocity boundary conditions and the continuity constraint,

$$\tilde{T}_{2,0} \approx \left(\varphi_\tau \frac{\delta q_2}{\delta \xi_2} \right)_{j=0} = -\varphi_{\tau,0} \left(\frac{\delta \varphi_1 q_1}{\delta \xi_2} + \frac{\delta \varphi_1 q_3}{\delta \xi_2} \right)_{j=0},$$

$$\tilde{N}_{2,0} \approx 0.$$

The boundary condition of the pressure at the end of previous time step, $\tilde{G}_{2,0}^{k-1}(\tilde{p}^{k-1})$, may be computed from the available pressure $\tilde{p}^{k-1}$ using the definition (I.32d). Subtracting $\tilde{G}_{2,0}^{k-1}(\tilde{p}^{k-1})$ from both sides of the equation (I.33), we have an equation whose left hand side is

$$\tilde{G}_{2,0}^{k-1}(\tilde{p}_{\text{updated}}^{k-1}) - \tilde{G}_{2,0}^{k-1}(\tilde{p}^{k-1}) = \tilde{G}_{2,0}^k(\tilde{\phi}), \quad (\text{I.34})$$

which is the boundary condition at the lower wall of the Laplace equation for the pressure update

$$\tilde{\phi} = \tilde{p}_{\text{updated}}^{k-1} - \tilde{p}^{k-1},$$

$$\mathcal{L}^{k-1}(\tilde{\phi}) = 0. \quad (\text{I.35})$$

We may similarly obtain the boundary condition for $\tilde{\phi}$ at the upper wall, then solve this equation by splitting and solving in Fourier space using the same iterative procedure as shown in (II.42). Finally, we may determine the updated pressure, $\tilde{p}_{\text{updated}}^{k-1}$, from the pressure at the end of the previous timestep, $\tilde{p}^{k-1}$, and the projection variable $\tilde{\phi}$.

References

- AKSELVOLL, K. & MOIN, P. 1995 Large eddy simulation of turbulent confined coannular jets and turbulent flow over a backward facing step. *Tech. Rep.* TF-63. Thermosciences Division, Dept. of Mech. Eng., Stanford University.
- BENJAMIN, T. 1960 Effects of a flexible boundary on hydrodynamic stability. *J. Fluid Mech.* **9**, 513–532.
- BEWLEY, T. & PROTAS, B. 2004 Skin friction and pressure: the “footprints” of turbulence. *Physica D* **196**, 28–44.
- BEWLEY, T. R., MOIN, P. & TEMAM, R. 2001 DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *J. Fluid Mech.* **447**, 179–225.
- BLACKBURN, H. M., MANSOUR, N. N. & CANTWELL, B. J. 1996 Topology of fine-scale motions in turbulent channel flow. *J. Fluid Mech.* **310**, 269–292.
- BUSHNELL, D., HEFNER, J. & ASH, R. 1977 Effect of compliant wall motion on turbulent boundary layers. *Phys. Fluids A* **20** (10), S31–48.

- CARLSON, H. A., BERKOOZ, G. & LUMLEY, J. L. 1995 Direct numerical simulation of flow in a channel with complex, time-dependent wall geometries: A pseudospectral method. *J. Comput. Phys.* **121**, 155–175.
- CARLSON, H. A. & LUMLEY, J. L. 1996a Active control in the turbulent wall layer of a minimal flow unit. *J. Fluid Mech.* **329**, 341–371.
- CARLSON, H. A. & LUMLEY, J. L. 1996b Flow over an obstacle emerging from the wall of a channel. *AIAA J.* **34** (5), 924–931.
- CARPENTER, P., DAVIES, C. & LUCEY, A. 2000 Hydrodynamics and compliant walls: Does the dolphin have a secret? *Current Science* **79**, 758–765.
- CARPENTER, P. & GARRAD, A. 1985 The hydrodynamic stability of flow over Kramer-type compliant surfaces. I. Tollmien-Schlichting instabilities. *J. Fluid Mech.* **155**, 465–510.
- CARPENTER, P. & MORRIS, P. 1990 The effect of anisotropic wall compliance on boundary-layer stability and transition. *J. Fluid Mech.* **218**, 171–223.
- CHOI, H., MOIN, P. & KIM, J. 1993 Direct numerical simulation of turbulent flow over riblets. *J. Fluid Mech.* **255**, 503–539.
- CHOI, K.-S., X., Y., CLAYTON, B., GLOVER, E., ATLAR, M., SEMENOV, B. & KULIK, V. 1997 Turbulent drag reduction using compliant surfaces. *Proc. Royal Soc. Lond. A* **453**, 2229–2240.
- DANIEL, A., GASTER, M. & WILLIS, G. 1987 Boundary layer stability on compliant surfaces. *Tech. Rep.* Final Report No. 35020. British Maritime Technology Ltd., Teddington, Great Britain.
- DAVIES, C. & CARPENTER, P. 1997 Numerical simulations of the evolution of Tollmien-Schlichting waves over finite compliant panels. *J. Fluid Mech.* **335**, 361–392.
- DE ANGELIS, V., P., L. & BANERJEE, S. 1997 Direct numerical simulation of turbulent flow over a wavy wall. *Phys. Fluids* **9** (8), 2429–2442.
- ENDO, T. & HIMENO, R. 2002 Direct numerical simulation of turbulent flow over a compliant surface. *J. Turbulence* **3** (007), 1–10.
- FULGOSI, M., LAKEHAL, D., BANERJEE, S. & ANGELIS, V. D. 2003 Direct numerical simulation of turbulence in a sheared air-water flow with a deformable interface. *J. Fluid Mech.* **482**, 319–345.
- GASTER, M. 1988 Is the dolphin a red herring? In *Turbulence management and relaminarisation* (ed. H. Liepmann & R. Narasimha), pp. 285–304. Berlin: Springer.
- GRESHO, P. M. & SANI, R. L. 1987 On pressure boundary conditions for the incompressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids* **7**, 1111–1145.
- GAD-EL-HAK, M. 1986 The response of elastic and viscoelastic surfaces to a turbulent boundary layer. *J. Appl. Mech.* **53**, 206–212.
- GAD-EL-HAK, M. 1987 Compliant coatings research: a guide to the experimentalist. *J. Fluids Struct.* **1**, 55–70.

- GAD-EL-HAK, M. 1996 Compliant coatings: a decade of progress. *Appl. Mech. Rev. Part 2* **49** (10), S1–11.
- INGBER, D. E. 1997 Tensegrity: The architectural basis of cellular mechanotransduction. *Annual Review of Physiology* **59**, 575–599.
- INGBER, D. E. 1998 Architecture of life. *Scientific American* **January**, 48–57.
- JUNG, W. J., MANGIAVACCHI, N. & AKHAVAN, R. 1992 Suppression of turbulence in wall-bounded flows by high-frequency spanwise oscillations. *Phys. Fluids A* **4**, 1605–1607.
- LANDAHL, M. 1962 On the stability of a laminar incompressible boundary layer over a flexible surface. *J. Fluid Mech.* **13**, 609–632.
- LEE, T., FISHER, M. & SCHWARZ, W. 1993 Investigation of the stable interaction of a passive compliant surface with a turbulent boundary layer. *J. Fluid Mech.* **257**, 373–401.
- LUO, H. & BEWLEY, T. 2003 Design, modeling, and optimization of compliant tensegrity fabrics for the reduction of turbulent skin friction. *SPIE Paper* (5049-57).
- LUO, H. & BEWLEY, T. R. 2004 On the contravariant form of the Navier-Stokes equation in time-dependent curvilinear coordinate systems. *J. Comput. Phys.* **199** (1), 355–375.
- MASIC, M. & SKELTON, R. E. 2002 Open-loop shape control of stable unit tensegrity structures. In *Proceedings of 3rd World Congress of Structural Control, Como Italy*.
- MASON, P. J. & MORTON, B. R. 1987 Trailing vortices in the wakes of surface-mounted obstacles. *J. Fluid Mech.* **175**, 247–293.
- MITO, Y. & KASAGI, N. 1998 DNS study of turbulence modification with streamwise-uniform sinusoidal wall-oscillation. *Int. J. Heat & Fluid Flow* **19**, 470–481.
- OHTA, T., MIYAKE, Y. & KAJISHIMA, T. 1998 Direct numerical simulation of turbulent flow in a wavy channel. *JSME Int. J. B* **41** (2), 447–453.
- ORSZAG, S. A., ISRAELI, M. & DEVILLE, M. 1986 Boundary conditions for incompressible flows. *J. Scientific Computing* **1**, 75–111.
- PERRY, A. E. & CHONG, M. S. 1987 A description of eddying motions and flow patterns using critical-point concepts. *Ann. Rev. Fluid Mech.* **19**, 125–155.
- ROSENFELD, M. & KWAK, D. 1991 Time-dependent solutions of viscous incompressible flows in moving co-ordinates. *Int. J. Numerical Methods in Fluids* **13**, 1311–1328.
- SKELTON, R. E., HELTON, J. W., ADHIKARI, R., PINAUD, J. P. & CHAN, W. 2001 *An Introduction to the Mechanics of Tensegrity Structures. The Mechanical Systems Design Handbook: Modeling, Measurement, and Control*. CRC Press.
- TSANGARIS, S. & LEITER, E. 1984 On laminar steady flow in sinusoidal channels. *Journal of Engineering Mathematics* **18**, 89–103.
- XU, S., REMPFER, D. & LUMLEY, J. 2003 Turbulence over a compliant surface: numerical simulation and analysis. *J. Fluid Mech.* **478**, 11–34.

- ZILKER, D. P., COOK, G. W. & HANRATTY, T. J. 1977 Influence of the amplitude of a solid wavy wall on a turbulent flow. Part I. non-separated flows. *J. Fluid Mech.* **82**, 29–51.
- ZILKER, D. P. & HANRATTY, T. J. 1979 Influence of the amplitude of a solid wavy wall on a turbulent flow. Part II. separated flows. *J. Fluid Mech.* **90**, 257–271.

Chapter II

On the contravariant form of the Navier-Stokes equations in time-dependent curvilinear coordinate systems

This chapter has been published as

Luo, H. & Bewley, T. R. 2004 On the contravariant form of the Navier-Stokes equations in time-dependent curvilinear coordinate systems. *J. Comput. Phys.* **199** (1), 355–375.

Abstract

The contravariant form of the Navier-Stokes equations in a fixed curvilinear coordinate system is well known. However, when the curvilinear coordinate system is time-varying, such as when a body-fitted grid is used to compute the flow over a compliant surface, considerable care is needed to handle the momentum term correctly. The present paper derives the complete contravariant form of the Navier-Stokes equations in a time-dependent curvilinear coordinate system from the intrinsic derivative of contravariant vectors in a moving frame. The result is verified via direct transformation. These complete equations are then applied to compute incompressible flow in a 2D channel with prescribed boundary motion, and the significant effect of some terms which are sometimes either overlooked or assumed to be negligible in such a derivation is quantified.

II.1 Introduction

The Navier-Stokes equations in a fixed curvilinear coordinate system were established long ago using coordinate transformation; one may find the standard form of these equations and their derivation in tensor calculus textbooks (e.g. Aris, 1962). However, such a general form of the Navier-Stokes equations have not been used widely in numerical simulations, since the calculation of the covariant derivatives in curvilinear coordinate systems is generally quite expensive. Many researchers have opted for alternative forms of the Navier-Stokes equations when they deal with flows in complex geometries via mapping into a regular domain. Such an approach can also be applied to time-dependent curvilinear coordinate systems. For example, a formulation is widely used in which Cartesian based velocity components multiplied by the Jacobian of the transformation (i.e., the volume flux components) are used as the flow variables (e.g. Thomas & Lombard, 1979; Hixon, 2000). Another commonly used formulation incorporates the velocity vectors in both the Cartesian coordinate system and the curvilinear coordinate system (e.g. Pulliam & Steger, 1980; Smith & Shyy, 1995; Sheng *et al.*, 1995; Lei *et al.*, 2000; Hodges & Street, 1999). In this formulation, though the contravariant velocity vector is introduced to make the equations simpler, the acceleration of the momentum is ultimately determined in the Cartesian coordinate directions. Voke & Collins (1984) proposed a contravariant velocity-vorticity formulation of the Navier-Stokes equations for both compressible and incompressible flows in a fixed general coordinate system. Their formulation avoids explicit use of the connection coefficients and the transformation matrix elements in the governing equations at the expense of the computation of the contravariant form of the vorticity.

Rosenfeld & Kwak (1991) presented a discrete contravariant formulation of the incompressible Navier-Stokes equations in generalized moving coordinates using a finite volume method that satisfies the geometric conservation laws for time-varying computational cells. However, the corresponding PDE in the continuous setting is not readily apparent from this inherently discrete formulation.

Under some circumstances, for example, when the transformation is relatively simple, the use of the continuous tensorial formulation of Navier-Stokes equations is manageable. Carlson, Berkooz & Lumley (1995) extended the tensorial formulation to the moving coordinate system case and used direct numerical simulation to calculate turbulence in a channel with time-dependent wall geometries. Their formulation was used later by Xu, Rempfer & Lumley (2003) to simulate turbulent flow over a compliant surface. Because of the specific transformation used in their work, a change in orientation of a vector into the new coordinate system is ascribed only to its wall-normal component and many connection coefficients vanish. However, when deriving the temporal derivative of a vector tensor in a moving frame, one has to be careful, since additional terms may appear due to the moving coordinates. The derivation of Carlson *et al.* omits some of these potentially important terms. Simply treating the temporal derivative of the contravariant form of the velocity vector in the same way as for a scalar variable, such as density, fails to capture all of the terms in this formulation, thereby possibly compromising the accuracy of the subsequent computations.

Ogawa & Ishiguro (1987) also derived the temporal derivative of tensor vectors, as considered in the present work, via a different approach than that used here, specifically, by considering the infinitesimal geometric motion of the curvilinear coordinates. The form of the Navier-Stokes equations they obtained, which is consistent with the present derivation, involves the covariant derivatives of the velocity of the coordinates that are missing in the analysis of Carlson *et al.*. In the present work, we derive the intrinsic temporal derivative of tensor vectors using an alternative approach, the quotient rule of tensor analysis, and then obtain the complete contravariant form of the Navier-Stokes equations in time-dependent curvilinear coordinate systems. Unlike the derivation of Ogawa & Ishiguro, which is based on geometrical arguments, the tensor derivation given in the present paper may be easily generalized to tensorial equations of order higher than one (vectors) if necessary. We also demonstrate use of the equations by applying them to solve flows in 2D channels with moving boundaries. Note that, in addition to approaches based on coordinate transformation, there are many other available techniques for computational fluid dynamics in systems with moving boundaries, e.g., volume tracking methods, level-set methods, and immersed boundary methods,

etc. Readers are referred to Shyy *et al.* (1996), Sheth & Pozrikidis (1995), and Fadlun *et al.* (2000) for more information.

II.2 Derivation of the Navier-Stokes equations in moving coordinate systems

II.2.1 Equations of motion in a fixed coordinate system

In order to introduce the notation to be used, we first consider a time-invariant transformation $x^i = x^i(\xi^1, \xi^2, \xi^3)$ from the Cartesian coordinates $\mathbf{x}$ to the curvilinear coordinates $\boldsymbol{\xi}$. (Note that superscripts indicate contravariant components, not powers, in the present notation.) We define the transformation matrix

$$C = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}, \quad c_j^i = \frac{\partial x^i}{\partial \xi^j}, \quad (\text{II.1})$$

and its inverse

$$\bar{C} = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}, \quad \bar{c}_j^i = \frac{\partial \xi^i}{\partial x^j}. \quad (\text{II.2})$$

The metric tensor and its inverse are defined by

$$g_{ij} = c_i^k c_j^k, \quad g^{ij} = \bar{c}_k^i \bar{c}_k^j, \quad (\text{II.3})$$

respectively. The Jacobian of the transformation is defined by

$$J = |C|. \quad (\text{II.4})$$

The transformation relationship between the contravariant velocity vector $\mathbf{v}$ in the Cartesian coordinate system and its counterpart $\mathbf{u}$ in the curvilinear coordinate system is

$$v^j = c_j^i u^i, \quad \text{or} \quad u^i = \bar{c}_j^i v^j. \quad (\text{II.5})$$

The same relationship also applies to other contravariant vectors.

The mass and momentum conservation equations in contravariant form in a fixed coordinate system may be written as follows (Aris, 1962, pp. 178-179)

$$\begin{aligned} \frac{\partial \rho}{\partial t} + (\rho u^i)_{,i} &= 0 \\ \rho \left(\frac{\partial u^i}{\partial t} + u^j u_{,j}^i \right) &= \rho f^i + T_{,j}^{ij}, \end{aligned} \quad (\text{II.6})$$

where the contravariant vector f^i is the external body force per unit mass and T^{ij} is the contravariant stress tensor. In the above equations, a comma with an index in a subscript denotes covariant differentiation:

$$u^i_{,j} = \frac{\partial u^i}{\partial \xi^j} + \Gamma^i_{jk} u^k, \quad (\text{II.7})$$

where Γ^i_{jk} are the components of connection coefficients, also known as the Christoffel symbol of the second kind. Now consider a time-dependent transformation from a Cartesian coordinate system $\mathbf{x}$ to a curvilinear coordinate system ξ

$$\begin{cases} x^i = x^i(\xi^1, \xi^2, \xi^3, \tau) \\ t = \tau. \end{cases} \quad (\text{II.8})$$

It is tempting (see, e.g., the derivations of Carlson *et al.*, 1995) to simply apply the chain rule

$$\frac{\partial}{\partial t} \longrightarrow \frac{\partial}{\partial \tau} + \frac{\partial \xi^j}{\partial t} \frac{\partial}{\partial \xi^j} \quad (\text{II.9})$$

and the relations (II.1) - (II.5) in order to re-express the temporal derivative terms in (II.6) and to cast it in the moving coordinate system. But is this correct? The answer is not obvious for an equation as complicated as (II.6). However, we may use a simple counterexample to show that such a simple substitution in fact misses some terms which are sometimes important. Consider a uniform flow free from the external force. Its velocity components in the Cartesian coordinate system are $v^1 = 1$, $v^2 = v^3 = 0$. Suppose we use the following coordinate transformation:

$$\begin{cases} x^1 = \xi^1 \cos \tau - \xi^2 \sin \tau \\ x^2 = \xi^1 \sin \tau + \xi^2 \cos \tau \end{cases}, \quad C = \begin{pmatrix} \cos \tau & -\sin \tau \\ \sin \tau & \cos \tau \end{pmatrix}, \quad \bar{C} = \begin{pmatrix} \cos \tau & \sin \tau \\ -\sin \tau & \cos \tau \end{pmatrix}, \quad (\text{II.10})$$

which simply means that the new coordinate system is rotating counterclockwise at a constant speed. We can see immediately [by (II.5)] that the velocity components in the new coordinate system are $u^1 = \cos \tau$, $u^2 = -\sin \tau$ (the third component is neglected since this is a two-dimensional problem). Since $\mathbf{u}$ is independent of ξ , and the connection coefficients Γ^i_{jk} are all zero, by (II.7) we have $u^i_{,j} = 0$. However, applying (II.9) would produce the momentum equation

$$\frac{\partial u^i}{\partial t} = \frac{\partial u^i}{\partial \tau} + \frac{\partial \xi^j}{\partial t} \frac{\partial u^i}{\partial \xi^j} = \frac{\partial u^i}{\partial \tau} = 0, \quad (\text{II.11})$$

which is clearly incorrect. Apparently, the Coriolis force is not correctly accounted for in the rotating coordinates by following this approach.

To solve the apparent dilemma, we examine the intrinsic temporal derivative of a tensor vector in a moving frame and apply the differentiation to the contravariant velocity vector while

using Reynolds' transport theorem to derive correctly the desired form of the Navier-Stokes equations.

We shall rewrite the definition of Γ_{jk}^i , the Christoffel symbol of the second kind, to facilitate this derivation. The common definition of the Christoffel symbol is given by

$$\Gamma_{jk}^i = g^{ip} [jk, p] = \frac{1}{2} g^{ip} \left(\frac{\partial g_{pj}}{\partial \xi^k} + \frac{\partial g_{pk}}{\partial \xi^j} - \frac{\partial g_{jk}}{\partial \xi^p} \right),$$

where $[jk, p]$ is the Christoffel symbol of the first kind, as defined in (Aris, 1962, p. 162) by

$$[jk, p] = \frac{1}{2} \left(\frac{\partial g_{pj}}{\partial \xi^k} + \frac{\partial g_{pk}}{\partial \xi^j} - \frac{\partial g_{jk}}{\partial \xi^p} \right) = \sum_m \left[\frac{\partial x^m}{\partial \xi^p} \frac{\partial}{\partial \xi^j} \left(\frac{\partial x^m}{\partial \xi^k} \right) \right] = c_p^m \frac{\partial c_k^m}{\partial \xi^j}.$$

The Christoffel symbol of the second kind may thus be written

$$\Gamma_{jk}^i = g^{ip} [jk, p] = \bar{c}_l^i \bar{c}_l^p c_p^m \frac{\partial c_k^m}{\partial \xi^j} = \bar{c}_l^i \delta_l^m \frac{\partial c_k^m}{\partial \xi^j} = \bar{c}_l^i \frac{\partial c_k^l}{\partial \xi^j}. \quad (\text{II.12})$$

II.2.2 The intrinsic derivative

We now follow the procedure in (Aris, 1962, p. 166) to derive the intrinsic derivative. Consider a time-dependent coordinate transformation (II.8). The velocity vectors of the moving coordinates are

$$\bar{U}^i = \frac{\partial x^i}{\partial \tau}, \quad \text{and} \quad U^j = -\frac{\partial \xi^j}{\partial t} \quad (\text{II.13})$$

in the Cartesian space and curvilinear space, respectively, and they satisfy the contravariant transformation rule, i.e., $\bar{U}^i = c_j^i U^j$.

Let $\bar{B}_i$ be an arbitrary parallel covariant vector field with constant components in Cartesian space $(\mathbf{x}, t)$, and B_i be its covariant counterpart in the curvilinear space (ξ, τ) . Consider a curve describing the path of a fluid particle, parameterized by $x^i(t)$ and $\xi^i(t)$ in the two coordinate systems, respectively. Note that $\xi^i(t)$ can be determined from $x^i(t)$ and the implicit function $\xi^i = \xi^i(x^j, t)$ implied by the transformation (II.8). The two parametric equations $\mathbf{x}(t)$ and $\xi(t)$ satisfy

$$\frac{d\xi^k}{dt} = \frac{\partial \xi^k}{\partial t} + \frac{\partial \xi^k}{\partial x^p} \frac{dx^p}{dt} = -U^k + \bar{c}_p^k v^p = -U^k + u^k, \quad (\text{II.14})$$

where $v^p = \frac{dx^p}{dt}$ and $u^k = \bar{c}_p^k v^p$ are the velocities of the particle in the Cartesian space and curvilinear space, respectively. We are now looking for a derivative with respect to t , namely the intrinsic derivative, denoted as $\frac{D}{Dt}$, which meets two requirements: (1) it should reflect the total variation of a tensor vector along the curve due to infinitesimal change of t (correspondingly, in Cartesian coordinates, this derivative will reduce to the material derivative), and (2) it should preserve tensor

character so that it can be applied to any coordinate system. By the first requirement, $\frac{DB_i}{Dt}$ should vanish along the curve, as B_i represents physically the same constant parallel vector field as $\bar{B}_i$. Since we have $\frac{D\bar{B}_i}{Dt} = \frac{d\bar{B}_i}{dt} = 0$, which holds for all points along the curve, this condition is

$$\frac{d\bar{B}_i}{dt} = \frac{d}{dt}(\bar{c}_i^j B_j) = \bar{c}_i^j \frac{dB_j}{dt} + B_j \frac{d\bar{c}_i^j}{dt} = 0. \quad (\text{II.15})$$

Multiplying this equation by c_r^i and summing over i , noting that $c_r^i \bar{c}_i^j = \delta_r^j$, we obtain the condition for the covariant B_r to be a parallel field,

$$\frac{dB_r}{dt} + B_j c_r^i \frac{d\bar{c}_i^j}{dt} = 0. \quad (\text{II.16})$$

This suggests that, for a covariant vector B_r , the intrinsic derivative we are seeking is

$$\frac{DB_r}{Dt} = \frac{dB_r}{dt} + B_j c_r^i \frac{d\bar{c}_i^j}{dt},$$

which indeed satisfies both requirements [fulfillment of the second requirement is seen from the derivation of (II.16), noting that $\frac{DB_r}{Dt} = c_r^i \frac{D\bar{B}_i}{Dt}$]. In the same way, we may obtain a similar derivative for a parallel contravariant vector field.

More generally, we now use the quotient rule to derive the tensor character for such a derivative of an arbitrary contravariant vector field. Switching dummy indices in (II.16) by $i \rightarrow l$, and then $r \rightarrow i$,

$$\frac{dB_i}{dt} = -B_j c_i^l \frac{d\bar{c}_l^j}{dt} = B_j \left(\bar{c}_l^j \frac{dc_i^l}{dt} - \frac{dc_i^l \bar{c}_l^j}{dt} \right) = B_j \left(\bar{c}_l^j \frac{dc_i^l}{dt} - \frac{d\delta_i^j}{dt} \right) = B_j \bar{c}_l^j \frac{dc_i^l}{dt}. \quad (\text{II.17})$$

Now let A^i be any time-varying contravariant vector field in (ξ, τ) -space. Then $A^i B_i$ and its derivative along the curve $\xi(t)$, $\frac{d(A^i B_i)}{dt}$, are both scalars which are independent of the coordinate systems. However, by (II.17)

$$\frac{d(A^i B_i)}{dt} = \frac{dA^i}{dt} B_i + A^i \frac{dB_i}{dt} = \frac{dA^i}{dt} B_i + A^i B_j \bar{c}_l^j \frac{dc_i^l}{dt} = \left(\frac{dA^j}{dt} + A^i \bar{c}_l^j \frac{dc_i^l}{dt} \right) B_j. \quad (\text{II.18})$$

Note that the dummy index i in the first term has been switched to j . Since $\frac{d(A^i B_i)}{dt}$ is a scalar tensor and B_j is an arbitrary covariant vector, the quotient rule implies that the term in brackets is a contravariant vector. It is the intrinsic derivative we are seeking and may be written as

$$\frac{DA^j}{Dt} = \frac{dA^j}{dt} + A^i \bar{c}_l^j \frac{dc_i^l}{dt}. \quad (\text{II.19})$$

Note that the material derivative $\frac{d}{dt}$ along the curve $\xi(t)$ is

$$\frac{d}{dt} = \frac{\partial}{\partial \tau} \frac{d\tau}{dt} + \frac{\partial}{\partial \xi^k} \frac{d\xi^k}{dt} = \frac{\partial}{\partial \tau} + \frac{\partial}{\partial \xi^k} \frac{d\xi^k}{dt}. \quad (\text{II.20})$$

Substitute (II.14) and (II.20) into (II.19), and the intrinsic derivative becomes

$$\begin{aligned}
 \frac{DA^j}{Dt} &= \frac{\partial A^j}{\partial \tau} + \frac{\partial A^j}{\partial \xi^k} \frac{d\xi^k}{dt} + A^i \bar{c}_l^j \left(\frac{\partial c_l^i}{\partial \tau} + \frac{\partial c_l^i}{\partial \xi^k} \frac{d\xi^k}{dt} \right) \\
 &= \frac{\partial A^j}{\partial \tau} + \left(\frac{\partial A^j}{\partial \xi^k} + A^i \bar{c}_l^j \frac{\partial c_l^i}{\partial \xi^k} \right) \frac{d\xi^k}{dt} + A^i \bar{c}_l^j \frac{\partial c_l^i}{\partial \tau} \\
 &= \frac{\partial A^j}{\partial \tau} + \left(\frac{\partial A^j}{\partial \xi^k} + \Gamma_{ki}^j A^i \right) \frac{d\xi^k}{dt} + A^i \bar{c}_l^j \frac{\partial c_l^i}{\partial \tau} \\
 &= \frac{\partial A^j}{\partial \tau} + A_{,k}^j (u^k - U^k) + A^i \bar{c}_l^j \frac{\partial c_l^i}{\partial \tau},
 \end{aligned} \tag{II.21}$$

where the definition of Christoffel symbol of the second kind in (II.12) has been used. Furthermore, since

$$\frac{\partial c_l^i}{\partial \tau} = \frac{\partial}{\partial \tau} \left(\frac{\partial x^l}{\partial \xi^i} \right) = \frac{\partial}{\partial \xi^i} \left(\frac{\partial x^l}{\partial \tau} \right) = \frac{\partial \bar{U}^l}{\partial \xi^i} = \frac{\partial x^k}{\partial \xi^i} \frac{\partial \bar{U}^l}{\partial x^k} = \frac{\partial x^k}{\partial \xi^i} \bar{U}_{,k}^l$$

where the covariant derivative and partial derivative are the same in Cartesian space, and $\bar{U}_{,k}^l$ is actually a mixed second order tensor, then

$$\bar{c}_l^j \frac{\partial c_l^i}{\partial \tau} = \bar{c}_l^j c_i^k \bar{U}_{,k}^l = U_{,i}^j. \tag{II.22}$$

Finally, the intrinsic derivative of the contravariant vector A^j becomes

$$\frac{DA^j}{Dt} = \frac{\partial A^j}{\partial \tau} + A_{,i}^j (u^i - U^i) + A^i U_{,i}^j. \tag{II.23}$$

The additional terms that arise in the intrinsic temporal differentiation are due to the motion of the base vectors which are also spatially varying. To see this let us consider a Cartesian vector

$$\bar{\mathbf{a}} = a^j \mathbf{e}_{(j)},$$

where the coefficient a^j is the component of the contravariant counterpart of $\bar{\mathbf{a}}$ in the curvilinear coordinate system (ξ, τ) , and $\mathbf{e}_{(j)}$ is a set of Cartesian base vectors for (ξ, τ) -coordinates, defined by $\mathbf{e}_{(j)}^i = \frac{\partial x^i}{\partial \xi^j} = c_j^i$. The derivative of $\bar{\mathbf{a}}$ with respect to t along the curve $\xi(t)$ is

$$\begin{aligned}
 \frac{d\bar{\mathbf{a}}}{dt} &= \frac{\partial \bar{\mathbf{a}}}{\partial \tau} + \frac{\partial \bar{\mathbf{a}}}{\partial \xi^i} \frac{d\xi^i}{dt} \\
 &= \frac{\partial a^j}{\partial \tau} \mathbf{e}_{(j)} + a^j \frac{\partial \mathbf{e}_{(j)}}{\partial \tau} + \left(\frac{\partial a^j}{\partial \xi^i} \mathbf{e}_{(j)} + a^j \frac{\partial \mathbf{e}_{(j)}}{\partial \xi^i} \right) \frac{d\xi^i}{dt} \\
 &= \left[\frac{\partial a^j}{\partial \tau} + a^i U_{,i}^j + a_{,i}^j \frac{d\xi^i}{dt} \right] \mathbf{e}_{(j)},
 \end{aligned} \tag{II.24}$$

where the definition of $\mathbf{e}_{(j)}$, equations (II.12) and (II.22) have been used to show $\frac{\partial \mathbf{e}_{(j)}}{\partial \tau} = U_{,j}^i \mathbf{e}_{(i)}$ and $\frac{\partial \mathbf{e}_{(j)}}{\partial \xi^i} = \Gamma_{ji}^k \mathbf{e}_{(k)}$, and the dummy indices are swapped to complete the derivation. The expression in

brackets is just the contravariant form of the intrinsic derivative and shows that the additional terms come in with both the temporal and spatial variability of the base vectors.

It should be pointed out this differentiation is not the same as the convective derivative, as described in Oldroyd (1950) and (Aris, 1962, p. 185), which is used to study inherent material properties, especially for non-Newtonian flows, even if our moving coordinates are chosen to be the material coordinates in which case the intrinsic derivative reduces to $\frac{DA^j}{Dt} = \frac{\partial A^j}{\partial \tau} + A^i U_{,i}^j$. The convective derivative is simply $\frac{\partial A^j}{\partial \tau}$, which reflects the rate of change of A^j in material coordinates. Note that the convective derivative $\frac{\partial A^j}{\partial \tau}$, when expressed in the fixed curvilinear coordinates, has an involved form reminiscent of (II.23), but represents something different. The intrinsic derivative $\frac{DA^j}{Dt}$ also takes into account the change caused by the motion of material coordinates themselves.

II.2.3 Equations of motion in a moving coordinate system

We now apply Reynolds' transport theorem. Let $F(\xi, \tau)$ be any function and $V(t)$ be a closed volume moving with the fluid. Then

$$\frac{D}{Dt} \int \int \int_{V(t)} F dV = \int \int \int_{V(t)} \left[\frac{DF}{Dt} + F u_{,i}^i \right] dV \quad (\text{II.25})$$

holds for any tensor F . If F is a scalar, e.g., $F = \rho$, then the term $A^i U_{,i}^j$ in (II.23) does not exist and the covariant derivative reduces to the partial derivative, so we have $\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial \tau} + \frac{\partial \rho}{\partial \xi^i} (u^i - U^i)$. By the mass conservation law we require the integration in (II.25) to vanish, and thus obtain the continuity equation

$$\frac{\partial \rho}{\partial \tau} - \frac{\partial \rho}{\partial \xi^i} U^i + (\rho u^i)_{,i} = 0, \quad (\text{II.26})$$

which, incidentally, is exactly the same as what we get by replacing (II.9) into the mass conservation equation in (II.6) in a fixed curvilinear coordinate system. The temporal terms in the energy conservation equation transform in a similar manner.

If $F = \rho u^i$ is the momentum vector, by the definition of the intrinsic derivative in (II.23) and the momentum conservation law, we have

$$\begin{aligned} \frac{D}{Dt} \int \int \int_{V(t)} \rho u^i dV &= \int \int \int_{V(t)} \left[\frac{\partial \rho u^i}{\partial \tau} + (u^j - U^j) (\rho u^i)_{,j} + \rho u^j U_{,j}^i + \rho u^i u_{,j}^j \right] dV \\ &= \int \int \int_{V(t)} [\rho f^i + T_{,j}^{ij}] dV. \end{aligned} \quad (\text{II.27})$$

Therefore the momentum conservation equation may be written

$$\frac{\partial u^i}{\partial \tau} + (u^j - U^j) u_{,j}^i + u^j U_{,j}^i = f^i + \frac{1}{\rho} T_{,j}^{ij}. \quad (\text{II.28})$$

Note that the continuity equation (II.26) has been substituted to make the above equation simpler.

When the ξ coordinate system is fixed, i.e., $U^j = 0$, then the equations (II.26) and (II.28) reduce to (II.6) as expected. Conversely, however, the extra terms which arise from the temporal intrinsic differentiation of a tensor vector, $-U^j u_{,j}^i + u^j U_{,j}^i$, are not obviously seen in (II.6). Simply applying the chain rule (II.9) for temporal differentiation in the momentum conservation equation in (II.6) instead produces $-U^j \frac{\partial u^i}{\partial \xi^j}$, which is not equivalent to $-U^j u_{,j}^i + u^j U_{,j}^i$.

II.2.4 Example: Uniform flow in a rotating coordinate system

We now re-examine the example of the uniform flow in a spinning coordinate system, as introduced in (II.10). The velocity of the spinning coordinate system is

$$\mathbf{U} = -\frac{\partial \xi}{\partial t} = \begin{pmatrix} \sin t & -\cos t \\ \cos t & \sin t \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} -\xi^2 \\ \xi^1 \end{pmatrix}.$$

Since the covariant differentiation reduces to partial differentiation in this case and $u_{,j}^i = 0$, in this example equation (II.28) reduces to

$$\frac{\partial u^i}{\partial \tau} + u^j U_{,j}^i = 0,$$

which may be written

$$\begin{pmatrix} -\sin \tau \\ -\cos \tau \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos \tau \\ -\sin \tau \end{pmatrix} = \mathbf{0}.$$

Thus, (II.28) is seen to hold in the present example.

II.2.5 Example: Stagnation point flow in a material coordinate system

When $U^j = u^j$, that is, the velocity of the coordinate coincides with the velocity of the fluid, as Ogawa & Ishiguro (1987) pointed out, equations (II.26) and (II.28) become the expressions in Lagrange coordinates, or material coordinates, which are often used to describe the mechanics of solids. We now consider the problem of stagnation point flow where the Cartesian velocity components and pressure (divided by the constant density) are given by, $v^1 = \alpha x^1$, $v^2 = -\alpha x^2$, $p = -\frac{\alpha^2}{2} [(x^1)^2 + (x^2)^2]$, and α is a constant scalar. The velocity vector satisfies the Cartesian momentum equation which, in terms of the material derivative, reduces to

$$\frac{dv^i}{dt} = \bar{d}^i = \begin{pmatrix} \alpha^2 x^1 \\ \alpha^2 x^2 \end{pmatrix}, \quad (\text{II.29})$$

where $\bar{d}^i$ is the contravariant pressure gradient vector divided by the density in Cartesian space. Motivated by (II.29), we define the material coordinate system

$$\begin{cases} \xi^1 = x^1 e^{-\alpha t} \\ \xi^2 = x^2 e^{\alpha t} \end{cases}, \quad C = \begin{pmatrix} e^{\alpha t} & 0 \\ 0 & e^{-\alpha t} \end{pmatrix}, \quad \bar{C} = \begin{pmatrix} e^{-\alpha t} & 0 \\ 0 & e^{\alpha t} \end{pmatrix}.$$

Note that in this special flow, the lines $\xi^1 = \text{constant}$ remain parallel to the x^2 axis and the lines $\xi^2 = \text{constant}$ remain parallel to the x^1 axis though these lines move along with the fluid particles. Since $U^j = u^j$, the contravariant form of the momentum conservation equation (II.28) in this example reduces to

$$\frac{\partial u^i}{\partial \tau} + u^j U_{,j}^i = d^i, \quad (\text{II.30})$$

where d^i denotes the counterpart of $\bar{d}^i$ in the new coordinate system. Note that, by the definition of $\mathbf{U}$, we have

$$\mathbf{u} = \mathbf{U} = -\frac{\partial \xi}{\partial t} = \begin{pmatrix} \alpha \xi^1 \\ -\alpha \xi^2 \end{pmatrix},$$

and by the relationship between contravariant vectors we have

$$\mathbf{d} = \bar{C} \bar{\mathbf{d}} = \begin{pmatrix} \alpha^2 \xi^1 \\ \alpha^2 \xi^2 \end{pmatrix},$$

so that, again, (II.28) is seen to hold in the present example.

As it can be seen, in both examples, although the coordinate systems are orthogonal, the extra terms arising from temporal differentiation of tensor vectors in a moving frame do not vanish.

II.3 Derivation by direct transformation

The Navier-Stokes equations in a time-dependent curvilinear coordinate system may also be obtained by directly transforming the equations in Cartesian system using the chain rule, though this approach is somewhat cumbersome and does not shed any additional light on the derivation or its physical significance. That is, taking the Cartesian-based momentum equation

$$\frac{\partial v^i}{\partial t} + v^k \frac{\partial v^i}{\partial x^k} = \bar{f}^i + \frac{1}{\rho} \frac{\partial \bar{T}^{ik}}{\partial x^k}, \quad (\text{II.31})$$

where $\bar{f}^i$ and $\bar{T}^{ik}$ are the counterparts of f^i and T^{ik} respectively in Cartesian space, we may transform the equation by applying chain rule to all the derivatives, substituting $\mathbf{v}$ with $\mathbf{u}$, multiplying it by $\bar{c}_i^j$ and invoking index summation. From a physical perspective, this means projecting the momentum

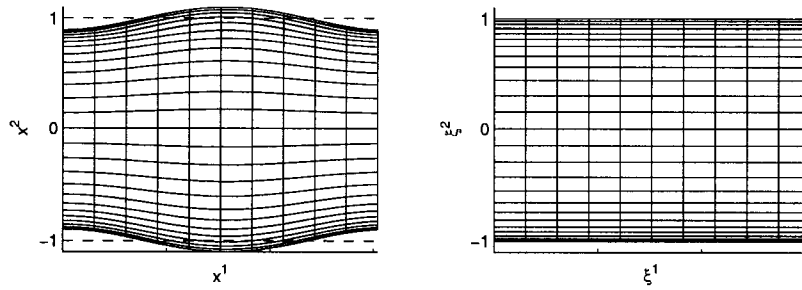


Figure II.1: Physical domain and the computational domain.

equation into the new coordinate system. We show here how the transformation of the temporal term is handled, which is the term of interest in this paper.

$$\begin{aligned}
 \bar{c}_i^j \frac{\partial v^i}{\partial t} &= \bar{c}_i^j \left[\frac{\partial c_k^i u^k}{\partial \tau} + \frac{\partial \xi^l}{\partial t} \frac{\partial c_k^i u^k}{\partial \xi^l} \right] \\
 &= \frac{\partial u^j}{\partial \tau} + \bar{c}_i^j \frac{\partial c_k^i}{\partial \tau} u^k + \frac{\partial \xi^l}{\partial t} \left(\bar{c}_i^j c_k^i \frac{\partial u^k}{\partial \xi^l} + \bar{c}_i^j \frac{\partial c_k^i}{\partial \xi^l} u^k \right) \\
 &= \frac{\partial u^j}{\partial \tau} + u^k U_{,k}^j - U^l \left(\frac{\partial u^j}{\partial \xi^l} + \Gamma_{lk}^j u^k \right) \\
 &= \frac{\partial u^j}{\partial \tau} + u^i U_{,i}^j - U^i u_{,i}^j.
 \end{aligned} \tag{II.32}$$

Similarly, the inertial term becomes

$$\bar{c}_i^j v^k \frac{\partial v^i}{\partial x^k} = u^i u_{,i}^j. \tag{II.33}$$

Clearly, (II.32) and (II.33) add up to the left-hand side of (II.28).

II.4 Application: flow in a channel with moving walls

A practical example of the use of such coordinate transformation is to compute the flow in a channel with moving walls. Consider first an incompressible flow in a periodic channel of length L_x whose two walls slightly deform about their nominal locations ($x^2 = 1$ and $x^2 = -1$) continuously with respect to time, as depicted in Fig. II.1. In the present study it is sufficient to demonstrate the relevant points in two dimensions; extension to three dimensions, though involved, is straightforward.

If we define $\eta_u(\xi^1, \tau)$ and $\eta_l(\xi^1, \tau)$ to be upper and lower wall displacement from their nominal positions, respectively, in the wall normal direction, and $\eta_1 = (\eta_u - \eta_l)/2$, $\eta_0 = (\eta_u +$

$\eta_l)/2$, then the time-dependent transformation

$$\begin{cases} x^1 = \xi^1 \\ x^2 = \xi^2(1 + \eta_1) + \eta_0 \\ t = \tau \end{cases} \quad (\text{II.34})$$

may be applied to transform the deformed domain into a rectangular domain. The time-independent version of this transformation has been used by Gal-Chen & Somerville (1975*b,a*) to simulate the meteorological phenomena of up-slope winds above a mountainous terrain. Carlson *et al.* (1995) and Kang & Choi (2000) used the three-dimensional version of the transformation to calculate turbulence in compliant channels. Note that conformal mapping or orthogonal transformation could be used to solve the present two-dimensional test problem with significantly less effort. However, our main purpose here is to illustrate a particular issue in general coordinate transformation on a 2D test problem.

By definitions (II.1) to (II.4), we have (2×2 instead of 3×3 matrices are considered for this 2D problem)

$$C = \begin{pmatrix} 1 & 0 \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} \end{pmatrix}, \quad \bar{C} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{J} \frac{\partial x^2}{\partial \xi^1} & \frac{1}{J} \end{pmatrix},$$

$$J = |C| = \frac{\partial x^2}{\partial \xi^2} = 1 + \eta_1, \quad \frac{\partial \xi^i}{\partial t} = \begin{pmatrix} 0 \\ -\frac{1}{J} \frac{\partial x^2}{\partial \tau} \end{pmatrix},$$

and the connection coefficients are

$$\Gamma_{jk}^1 = \Gamma_{22}^2 = 0,$$

$$\Gamma_{11}^2 = \frac{1}{J} \frac{\partial^2 x^2}{\partial (\xi^1)^2} = \frac{1}{1 + \eta_1} \left(\xi^2 \frac{\partial^2 \eta_1}{\partial (\xi^1)^2} + \frac{\partial^2 \eta_0}{\partial (\xi^1)^2} \right),$$

$$\Gamma_{12}^2 = \Gamma_{21}^2 = \frac{1}{J} \frac{\partial^2 x^2}{\partial \xi^1 \partial \xi^2} = \frac{1}{1 + \eta_1} \frac{\partial \eta_1}{\partial \xi^1}.$$

To simplify the notation, we define the following non-constant transformation coefficients

$$\varphi_1 = \frac{\partial \xi^2}{\partial x^1} = -\frac{1}{J} \frac{\partial x^2}{\partial \xi^1} = -\frac{1}{1 + \eta_1} \left(\xi^2 \frac{\partial \eta_1}{\partial \xi^1} + \frac{\partial \eta_0}{\partial \xi^1} \right),$$

$$\varphi_2 = \frac{\partial \xi^2}{\partial x^2} = \frac{1}{J} = \frac{1}{1 + \eta_1},$$

$$\varphi_\tau = \frac{\partial \xi^2}{\partial t} = -\frac{1}{J} \frac{\partial x^2}{\partial \tau} = -\frac{1}{1 + \eta_1} \left(\xi^2 \frac{\partial \eta_1}{\partial \tau} + \frac{\partial \eta_0}{\partial \tau} \right).$$

The two components of the momentum equation (II.28) may be written, after some manipulations,

as

$$\begin{aligned}
& \frac{\partial u^1}{\partial \tau} + \varphi_\tau \frac{\partial u^1}{\partial \xi^2} + u^1 \frac{\partial u^1}{\partial \xi^1} + u^2 \frac{\partial u^1}{\partial \xi^2} \\
&= -\frac{1}{\rho} \left(\frac{\partial p}{\partial \xi^1} + \varphi_1 \frac{\partial p}{\partial \xi^2} \right) + \nu \left[\left(\frac{\partial}{\partial \xi^1} + \varphi_1 \frac{\partial}{\partial \xi^2} \right) \left(\frac{\partial}{\partial \xi^1} + \varphi_1 \frac{\partial}{\partial \xi^2} \right) u^1 + \varphi_2^2 \frac{\partial^2 u^1}{\partial (\xi^2)^2} \right] - P_x, \\
& \frac{\partial u^2}{\partial \tau} + \varphi_\tau \frac{\partial u^2}{\partial \xi^2} - \underbrace{u^1 \frac{\partial \varphi_\tau}{\partial \xi^1} - u^2 \frac{\partial \varphi_\tau}{\partial \xi^2}} + u^1 \frac{\partial u^2}{\partial \xi^1} + u^2 \frac{\partial u^2}{\partial \xi^2} + u^1 u^2 \frac{2}{J} \frac{\partial J}{\partial \xi^1} + u^1 u^1 \frac{1}{J} \frac{\partial^2 x^2}{\partial (\xi^1)^2} \quad (\text{II.35}) \\
&= -\frac{1}{\rho} \varphi_1 \left(\frac{\partial p}{\partial \xi^1} + \varphi_1 \frac{\partial p}{\partial \xi^2} \right) - \frac{1}{\rho} \varphi_2^2 \frac{\partial p}{\partial \xi^2} - \varphi_1 P_x \\
& \quad + \nu \varphi_2 \left[\left(\frac{\partial}{\partial \xi^1} + \varphi_1 \frac{\partial}{\partial \xi^2} \right)^2 (Ju^2) + \varphi_2^2 \frac{\partial^2 Ju^2}{\partial (\xi^2)^2} + u^1 s_1 + 2 \frac{\partial u^1}{\partial \xi^1} s_2 + 2 \frac{\partial u^1}{\partial \xi^2} s_3 \right],
\end{aligned}$$

where ν is the kinematic viscosity, P_x is the uniform streamwise (x^1 -direction) pressure gradient that maintains the bulk flow, and

$$\begin{aligned}
s_1 &= \left(\frac{\partial}{\partial \xi^1} + \varphi_1 \frac{\partial}{\partial \xi^2} \right) s_2, \\
s_2 &= \frac{\partial^2 x^2}{\partial (\xi^1)^2} + \varphi_1 \frac{\partial J}{\partial \xi^1}, \\
s_3 &= \varphi_1 s_2 + \varphi_2^2 \frac{\partial J}{\partial \xi^1}.
\end{aligned}$$

By applying the metric invariants of the coordinate transformation, we may also express the conservative form of the momentum equation (see Appendix, equation (II.37)), which is preferable for implementation in a numerical algorithm. In the particular transformation shown above, applying (II.9) would cause the two under-braced terms $-u^1 \frac{\partial \varphi_\tau}{\partial \xi^1} - u^2 \frac{\partial \varphi_\tau}{\partial \xi^2}$ in the u^2 momentum equation in (II.35) to be absent. The continuity equation and boundary conditions are given in the Appendix.

II.4.1 Numerical algorithm

To solve (II.35), the volume flux components $q^1 = Ju^1$, $q^2 = Ju^2$, and the modified pressure $\bar{p} = Jp/\rho$ are chosen as the primitive variables. The numerical algorithm is based on that in Akselvoll & Moin (1995) and Bewley, Moin & Temam (2001). The grid is chosen to be evenly spaced in the streamwise direction (ξ^1) and non-staggered so that Fourier transformation techniques may be used to compute spatial derivatives in this direction. In the wall-normal direction (ξ^2) the grid is staggered and stretched using a hyperbolic tangent function allowing the near-wall region to be better resolved. Spatial derivatives of this direction is discretized using the second order centered finite difference scheme.

The flow is advanced in time using a low-storage third-order Runge-Kutta method. At each Runge-Kutta substep, all terms involving ξ^1 derivatives and cross derivatives of the primitive

variables are treated explicitly, and all terms involving only ξ^2 derivatives of the primitive variables are treated with an implicit Crank-Nicholson method.

At the beginning of a time step, a full pressure equation is solved with a Neumann boundary condition that is derived by imposing the discrete continuity constraint upon the discrete momentum equation. At the end of each Runge-Kutta substep, a projection function is solved to bring the velocity field to be solenoidal and to update the pressure. Readers are referred to the Appendix for the details of this numerical method.

To maintain the constant bulk velocity U_{bulk} (which is normalized by the nominal domain size), the uniform streamwise pressure gradient P_x is computed by integrating the u^1 momentum equation over the entire domain,

$$\begin{aligned} U_{bulk} &= \frac{1}{2L_x} \int_{-1+\eta_l}^{1+\eta_u} \int_0^{L_x} v^1 dx^1 dx^2 = \text{constant} \\ \Rightarrow P_x &= \frac{1}{2 \int J d\xi^1} \int \left[-\varphi_1 \bar{p} + v\varphi_1 \frac{\partial \varphi_1 q^1}{\partial \xi^2} + v\varphi_2^2 \frac{\partial q^1}{\partial \xi^2} \right]_{\xi^2=-1}^{\xi^2=1} d\xi^1. \end{aligned} \quad (\text{II.36})$$

The discrete numerical integration scheme in the code exactly conserves both mass and momentum. By computing P_x with (II.36), constant mass flux is numerically guaranteed. The Reynolds number is thus based on the bulk volume flux

$$Re = \frac{U_c h}{\nu} = \frac{3}{2} \frac{U_{bulk} h}{\nu},$$

where $U_c = \frac{3}{2} U_{bulk}$ is the centerline velocity of the corresponding Poiseuille flow with the same mass flux, and $h = 1$ is the half channel width. Time is normalized by $\frac{h}{U_c}$.

II.4.2 Laminar steady flow in channels with sinusoidal walls

The first case considered is the laminar flow in a symmetric channel with sinusoidal walls, as depicted in Fig. II.1, with $\eta_l = -\eta_u = \varepsilon \cos(\xi^1)$. This case, though not addressing the problem of a moving coordinate system, validates the correctness of the present code against an analytic result in the case of a stationary coordinate system. The parabolic laminar profile in an unperturbed channel is used as the initial condition. The wall deformation starts to grow gradually until the final geometry is reached and then remains unchanged, and the simulation continues until the steady state is reached. The results are compared with that from Tsangaris & Leiter (1984), who solved the laminar steady flow in sinusoidal channels for Reynolds numbers far above that for creeping flow. In their work, a perturbation method is used with the wall amplitude ε as the perturbation parameter.

The stream function is expanded in a series, and the first-order variation is derived, which boils down to solving numerically a linear system of 4th-order differential equations with two unknown functions and with variable coefficients.

Fig. 2 and Fig. 3 show the comparison of the Cartesian velocity components (v^1 and v^2) profiles for $\epsilon = 0.1$ and $\epsilon = 0.2$ at Reynolds numbers $Re = 1.0, 10, 75, 200, 400$. In our simulations, the number of Fourier modes is 32×64 in the ξ^1 and ξ^2 directions respectively (i.e., 48×64 dealiased collocation points), and the length of the computational domain is $L_x = 2\pi$. The resolution was doubled in both directions and the calculations repeated with no significant change of the results.

The comparisons show that our simulations agree very well with the results obtained by Tsangaris & Leiter when both the wall deformation parameter ϵ and the Reynolds number Re are small. However, the discrepancies become more evident as ϵ or Re is increased. The influence of ϵ is expected because the perturbation analysis of Tsangaris & Leiter is less accurate when the perturbation parameter ϵ is increased. The influence of Re is also expected since the leading-order truncation error from the perturbation series is related to Reynolds number as well.

Our simulations also show that the critical Reynolds number for flow separation to happen at $\epsilon = 0.2$ is about $Re_{crit} = 171$, and the separation point is about $\xi^1 = 2.6$ which is slightly upstream the maximum width of the wavy wall of the channel. These are slightly different from what Tsangaris & Leiter predicted, where $Re_{crit} = 185$ and the separation point is about $\xi^1 = 2.4$. As Reynolds number is increased above Re_{crit} , separation regions in the diverging portion of the channel are formed, as illustrated in Fig. II.4.

II.4.3 Moving boundary simulation

To illustrate the important effects of the sometimes-neglected terms in the Navier-Stokes equations in time-dependent curvilinear coordinates, we simulate the laminar flow in a channel with an oscillating Gaussian protuberance at center of the lower wall and corresponding blowing/suction applied at the opposite wall. The wall deformation is prescribed by

$$\eta_l(\xi^1, t) = \epsilon \sin(\omega t) \exp \left[-\frac{(\xi^1 - \frac{L_x}{2})^2}{\sigma^2} \right],$$

where ϵ is the amplitude of the wall deformation, ω is the oscillation frequency, and σ is a constant determining the width of the protuberance.

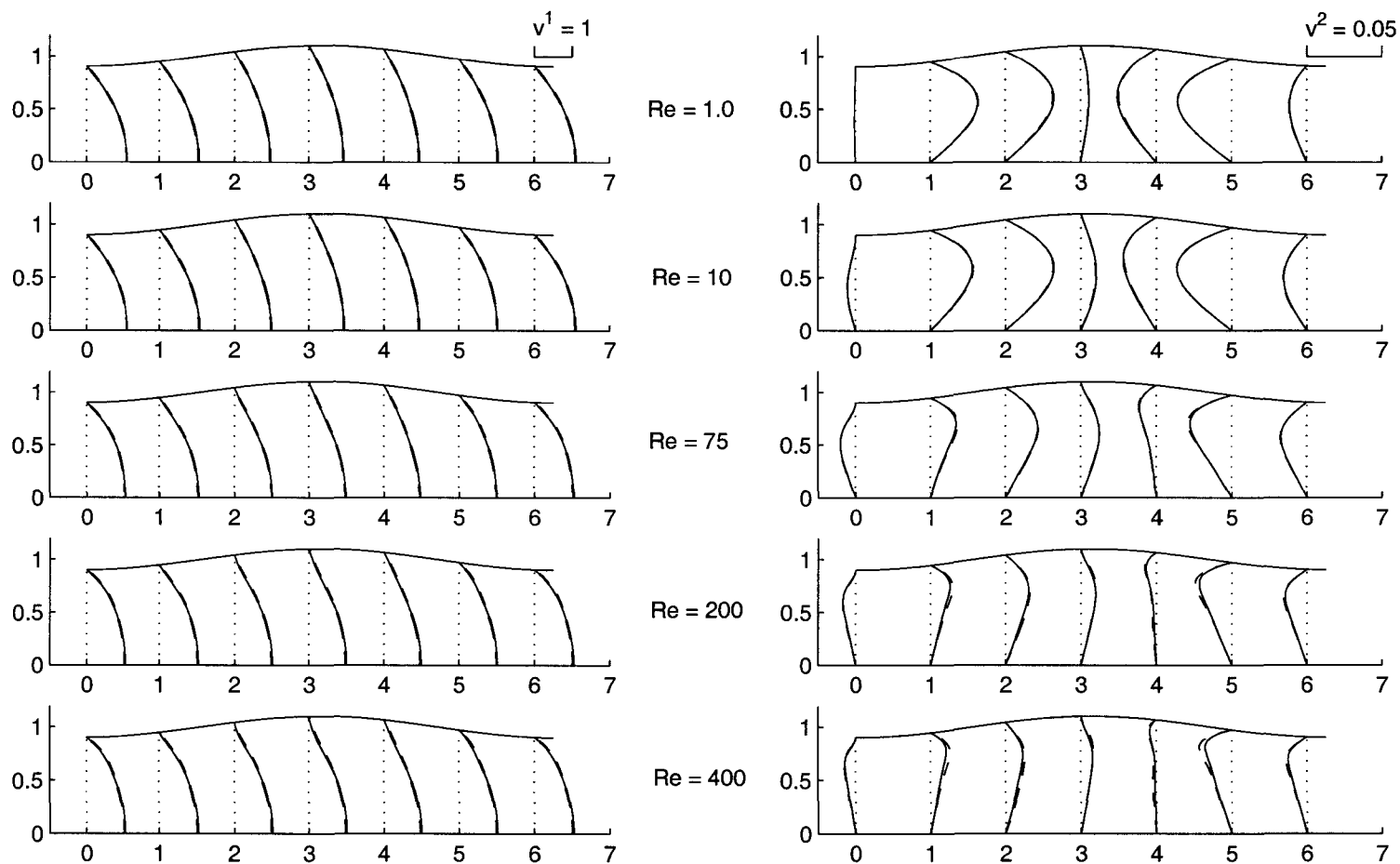


Figure II.2: Cartesian velocity profiles at various cross sections ($x^1 = 0, 1, 2, 3, 4, 5, 6$) of the channel for steady flow with $\epsilon = 0.1$ and various Reynolds numbers. Left: v^1 component; right: v^2 component. Solid: numerical results; dashed: perturbation analysis.

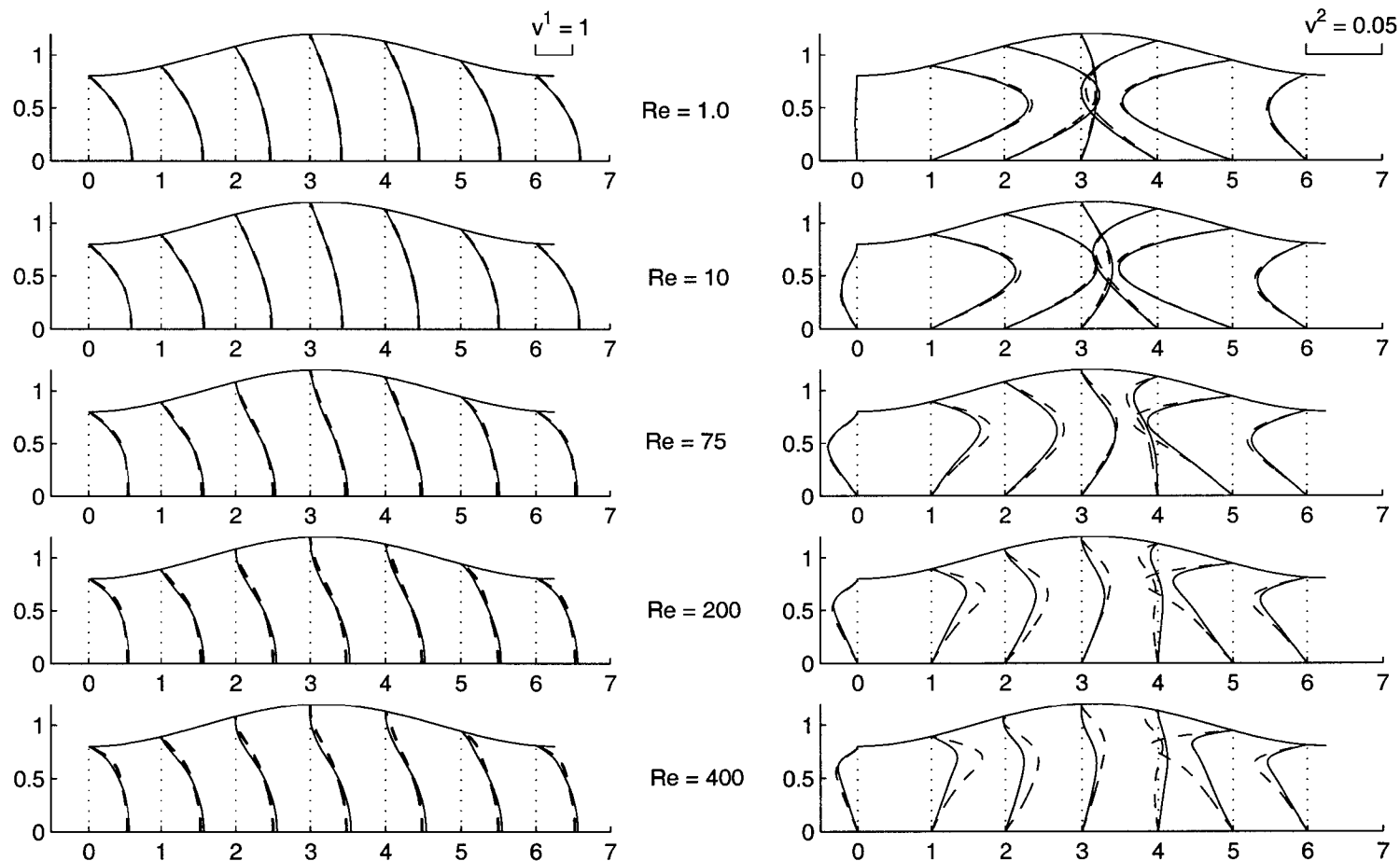


Figure II.3: Cartesian velocity profiles at various cross sections ($x^1 = 0, 1, 2, 3, 4, 5, 6$) of the channel for steady flow with $\varepsilon = 0.2$ and various Reynolds numbers. Left: v^1 component; right: v^2 component. Solid: numerical results; dashed: perturbation analysis.

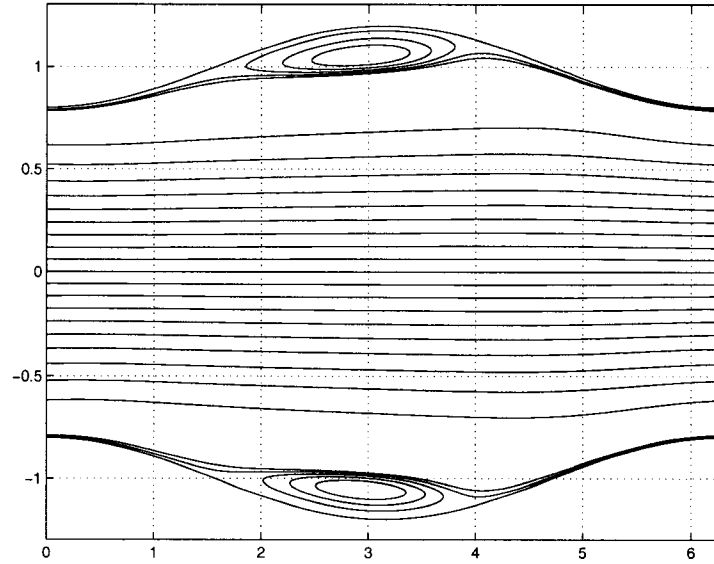


Figure II.4: The streamlines for the case with $\varepsilon = 0.2$ and $Re = 400$ showing the separation regions of the flow field.

To maintain the incompressibility of the flow, the upper wall is made porous and the velocity of the fluid through the upper wall is identical to the velocity of the lower wall. In all tests of this section, the channel length is chosen to be $L_x = \pi$, Reynolds number is $Re = 200$, and the wall deformation parameters are $\varepsilon = 0.1$ and $\sigma = 0.2$. In the simulation, the number of Fourier modes is 42×84 in the ξ^1 and ξ^2 directions respectively (i.e., 64×84 dealiased collocation points) and the time step is 0.01. The time step was reduced by a factor of 10 and there was no significant change of the results.

Due to the oscillation of the boundary, this flow exhibits time-periodic behavior. Fig. II.5 shows both the instantaneous streamlines' and pressure's oscillating patterns at different time phases within one period cycle $\frac{2\pi}{\omega} = \pi$ (i.e., $\omega = 2$). The streamline patterns show that center of the two walls serves as the fluid source or sink in the transversal direction of the flow. Whenever the lower wall is moving upward, serving as a source (i.e., $t = 0.1, 0.5, 2.5$), the instantaneous streamlines near the wall are combed downstream, which means the fluid is pushed up by the wall and drifts downstream with the main flow, and at the upper wall region, the instantaneous streamlines are contracted toward the wall by the corresponding fluid extraction. By similar reasoning, when the lower wall is moving downward, serving as a sink (i.e., $t = 1.0, 1.5, 2.0$), the instantaneous streamline patterns near the lower wall and the upper wall reverse. The pressure patterns oscillate in time as

well. At $t = 0.1$, a high-pressure region is formed at the front side of the bump due to the interaction between the viscous fluid and the rising obstacle, and a low pressure region is formed at the back. Form drag is therefore introduced in addition to the skin friction. The high pressure region gradually fades and the low pressure region shifts upstream as the bump descends ($t = 0.5$ and $t = 1.0$). When the wall moves down farther ($t = 1.5$), the low pressure region shifts to the upstream side of the bump and another high pressure region is formed downstream. Then the low pressure region gradually fades and the high pressure region shifts upstream ($t = 2.0$ and $t = 2.5$).

We also simulated the flow with under-braced terms in equation (II.35) intentionally omitted in the code. Since these terms arise from the temporal derivative, we may expect that the errors associated with their omission would be small if the wall motion is slow, but large if the wall motion is fast. Two comparisons are carried out, one with oscillation frequency $\omega = 0.5$, the other with $\omega = 4$. The resulting instantaneous streamlines and pressure contours at a phase of the oscillation are shown in Fig. II.6 and Fig. II.7. In the slow wall motion case, $\omega = 0.5$, the calculations with the terms omitted approximate our correct results fairly well. However, in the faster wall motion case, $\omega = 4$, the effects of omitted terms in the calculation become more evident. In Fig. II.7, where the wall is moving upward, the streamlines appear to overshoot above the bump, and undershoot downstream due to this omission. The pressure contours also become more irregular when the two terms are absent from the calculation.

II.5 Conclusions

In time-dependent curvilinear coordinates, the temporal derivative of a tensor vector is more complicated than the temporal derivative of a scalar. From equation (II.23), we can see that, for a contravariant vector A^j , its temporal intrinsic derivative involves its own covariant differentiation ($A^j_{;i}U^i$) and the covariant differentiation of the velocity of the coordinates ($A^iU^j_{;i}$). Treating A^j as a scalar during time differentiation is incorrect, as it drops some important terms. Since $U^j_{;i} = \tilde{c}^j_l \frac{\partial c^l_i}{\partial \tau}$ as we have shown, and $c^l_i = \frac{\partial x^l}{\partial \xi^i}$ is actually the component of the base vectors of the new coordinate system, the term $A^iU^j_{;i}$ arises because base vectors of the new coordinates are moving. Generally, in a time-dependent coordinate system $A^iU^j_{;i}$ won't vanish even if the coordinate lines are straight and/or orthogonal, as shown by the two examples in Sections II.2.4 and Sections II.2.5.

Assuming that the extra terms in question are small is only valid when the coordinate system is moving sufficiently slowly. We have demonstrated that these terms are not always negligible

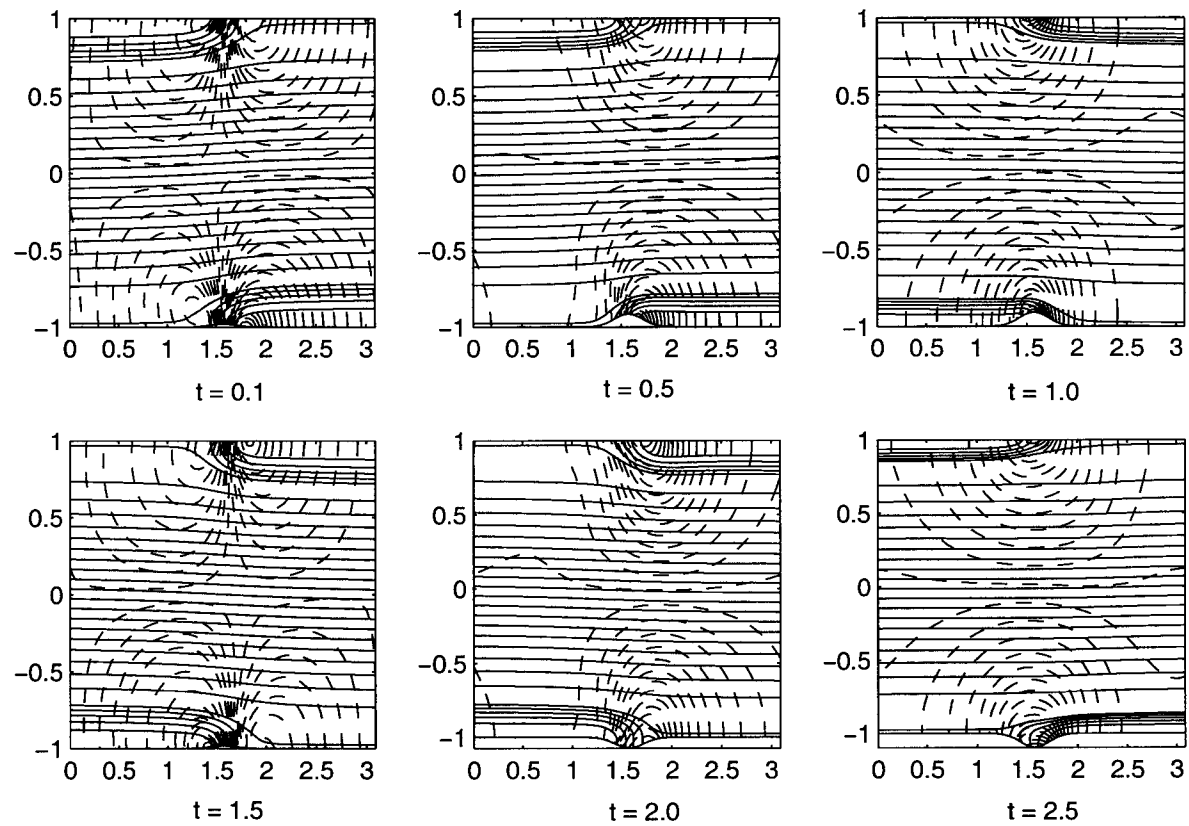


Figure II.5: Instantaneous streamlines (solid) and pressure (dashed) contours within one cycle of flow of $Re = 200$ in a channel with an oscillating Gaussian protuberance on the lower wall and corresponding blowing/suction on the upper wall. In all figures, the flow goes from left to right.

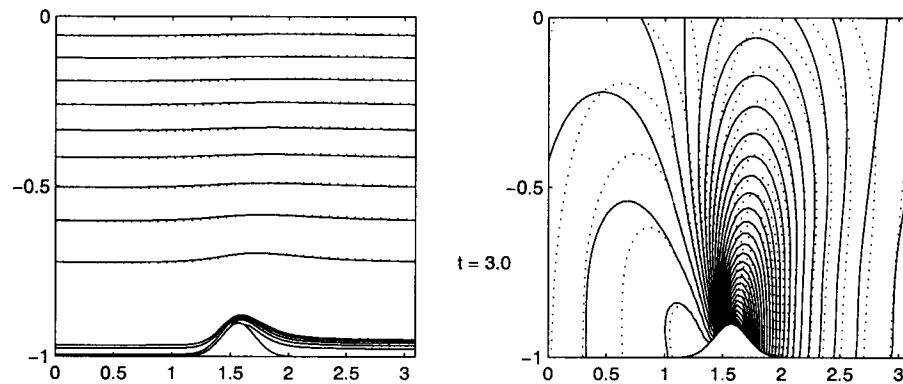


Figure II.6: Effects of the sometimes-neglected terms in the Navier-Stokes equations on flow of $Re = 200$ when wall oscillation is slow ($\omega = 0.5$). Time instance: $t = 3.0$. Left: instantaneous streamlines ψ ; right: pressure contours. Solid: the correct results; dotted: results with neglected terms. Quantification of error: $\|\psi_{error}\|_2 / \|\psi\|_2 = 0.2\%$; $\max_{\Omega}(\psi_{error}) / \|\psi\|_2 = 0.5\%$; $\|p_{error}\|_2 / \|p\|_2 = 17\%$; $\max_{\Omega}(p_{error}) / \|p\|_2 = 35\%$.

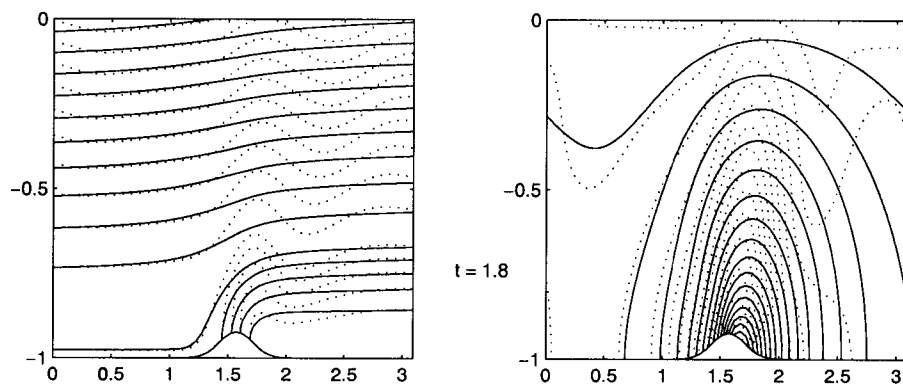


Figure II.7: Effects of the sometimes-neglected terms in the Navier-Stokes equations on flow of $Re = 200$ when wall oscillation is fast ($\omega = 4$). Time instance: $t = 1.8$. Left: instantaneous streamlines ψ ; right: pressure contours. Solid: the correct results; dotted: results with neglected terms. Quantification of error: $\|\psi_{error}\|_2 / \|\psi\|_2 = 1.6\%$; $\max_{\Omega}(\psi_{error}) / \|\psi\|_2 = 6.8\%$; $\|p_{error}\|_2 / \|p\|_2 = 15\%$; $\max_{\Omega}(p_{error}) / \|p\|_2 = 55\%$.

by simulating incompressible flows in a two-dimensional channel with prescribed boundary motion.

II.A Numerical implementation

In primitive variables. i.e., the volume flux components $q^1 = Ju^1$, $q^2 = Ju^2$, and the modified pressure $\bar{p} = Jp/\rho$, the conservative form of governing equation (II.35) for the compliant channel can be written as

$$\frac{\partial q^i}{\partial \tau} + T^i(q^j) + N^i(q^j) = -G^i(\bar{p}) + \nu L^i(q^i) - P^i, \quad (\text{II.37})$$

where $T^i(q^i)$ is the term involving ϕ_τ , $N^i(q^j)$ is the convection term, $G^i(\bar{p})$ is the pressure gradient term, $\nu L^i(q^i)$ is the diffusion term, and P^i is the uniform pressure gradient term. They are given by

$$\begin{aligned} T^1 &= \frac{\partial q^1 \phi_\tau}{\partial \xi^2} \\ T^2 &= \frac{\partial q^2 \phi_\tau}{\partial \xi^2} - q^1 \frac{\partial \phi_\tau}{\partial \xi^1} - q^2 \frac{\partial \phi_\tau}{\partial \xi^2} \\ N^1 &= \frac{\partial q^1 q^1 \phi_2}{\partial \xi^1} + \frac{\partial q^1 q^2 \phi_2}{\partial \xi^2} \\ N^2 &= \frac{\partial q^1 q^2 \phi_2}{\partial \xi^1} + \frac{\partial q^2 q^2 \phi_2}{\partial \xi^2} + 2\phi_2^2 q^1 q^2 \frac{\partial J}{\partial \xi^1} + \phi_2^2 q^1 q^1 \frac{\partial^2 x^2}{\partial (\xi^1)^2} \\ G^1 &= \frac{\partial \bar{p}}{\partial \xi^1} + \frac{\partial \bar{p} \phi_1}{\partial \xi^2} \\ G^2 &= \phi_1 \left(\frac{\partial \bar{p}}{\partial \xi^1} + \frac{\partial \bar{p} \phi_1}{\partial \xi^2} \right) + \phi_2^2 \frac{\partial \bar{p}}{\partial \xi^2} \\ L^1 &= \left(\frac{\partial}{\partial \xi^1} + \frac{\partial \phi_1}{\partial \xi^2} \right)^2 (q^1) + \phi_2^2 \frac{\partial^2 q^1}{\partial (\xi^2)^2} \\ L^2 &= \left(\frac{\partial}{\partial \xi^1} + \phi_1 \frac{\partial}{\partial \xi^2} \right)^2 (q^2) + \phi_2^2 \frac{\partial^2 q^2}{\partial (\xi^2)^2} + \phi_2 q^1 s_1 + 2 \frac{\partial \phi_2 q^1}{\partial \xi^1} s_2 + 2 \frac{\partial \phi_2 q^1}{\partial \xi^2} s_3 \\ P^1 &= JP_x \\ P^2 &= J\phi_1 P_x. \end{aligned}$$

Weighted by J , the continuity equation is

$$D_i q^i = \frac{\partial q^1}{\partial \xi^1} + \frac{\partial q^2}{\partial \xi^2} = 0, \quad (\text{II.38})$$

where D_i is the divergence operator. The boundary conditions are the no-slip and no-penetration boundary conditions which can be expressed as

$$\begin{aligned} q^1|_{\xi^2=\pm 1} &= 0 \\ q^2|_{\xi^2=+1} &= \frac{\partial \eta_u}{\partial \tau} \\ q^2|_{\xi^2=-1} &= \frac{\partial \eta_l}{\partial \tau}. \end{aligned} \quad (\text{II.39})$$

One exception in present work is the example of moving boundary with Gaussian protuberance where the upper wall is made porous and the boundary condition for the upper wall is thus $q^2|_{\xi^2=+1} = q^2|_{\xi^2=-1} = \frac{\partial \eta_l}{\partial \tau}$.

II.A.1 Temporal discretization

The flow is marched in time with a low-storage third-order Runge-Kutta method based on the scheme used by Akselvoll & Moin (1995) and Bewley *et al.* (2001). In each of the three Runge-Kutta substeps $k = 1, 2, 3$, two fractional steps are used: (1) an intermediate flow field q^{i*} is obtained by solving the momentum equation with some terms treated explicitly and some implicitly (Crank-Nicholson); (2) the velocities q^{i*} are projected to the divergence free space and the pressure is updated by the the projection function.

Let the operator A_i represent the terms treated explicitly and B_i represent the terms treated implicitly where the subscripts simply indicate the operator components, not covariant components. The discretized momentum equation may be written as

$$\begin{aligned} \frac{q^{i*} - q^{i^{k-1}}}{\delta \tau} &= \beta_k \left(B_i(q^{j*}) + B_i(q^{j^{k-1}}) \right) + \gamma_k A_i(q^{j^{k-1}}) + \zeta_k A_i(q^{j^{k-2}}) \\ &\quad + 2\beta_k \left(G^i(\bar{p}^{k-1}) - P^i \right) \end{aligned} \quad (\text{II.40})$$

where the explicit and implicit operators are given by:

$$\begin{aligned}
A_1(q^j) &= v \frac{\delta}{\delta \xi^1} \left(\frac{\delta q^1}{\delta \xi^1} + \frac{\delta \varphi_1 q^1}{\delta \xi^2} \right) + v \frac{\delta}{\delta \xi^2} \left(\varphi_1 \frac{\delta q^1}{\delta \xi^1} \right) - \frac{\delta q^1 q^1 \varphi_2}{\delta \xi^1} \\
A_2(q^j) &= v \frac{\delta}{\delta \xi^1} \left(\frac{\delta q^2}{\delta \xi^1} + \varphi_1 \frac{\delta q^2}{\delta \xi^2} \right) + v \varphi_1 \frac{\delta}{\delta \xi^2} \left(\frac{\delta q^2}{\delta \xi^1} \right) \\
&\quad + v \left(\varphi_2 q^1 s_1 + 2 \frac{\partial \varphi_2 q^1}{\partial \xi^1} s_2 + 2 \frac{\partial \varphi_2 q^1}{\partial \xi^2} s_3 \right) \\
&\quad - \frac{\delta q^1 q^2 \varphi_2}{\delta \xi^1} - \varphi_2^2 q^1 q^1 \frac{\delta}{\delta \xi^1} \left(\frac{\delta x^2}{\delta \xi^1} \right) + q^1 \frac{\delta \varphi_\tau}{\delta \xi^1} \\
B_1(q^j) &= v \frac{\delta}{\delta \xi^2} \left(\varphi_1 \frac{\delta \varphi_1 q^1}{\delta \xi^2} \right) + v \varphi_2^2 \frac{\delta}{\delta \xi^2} \left(\frac{\delta q^1}{\delta \xi^2} \right) - \frac{\delta q^1 q^2 \varphi_2}{\delta \xi^2} - \frac{\delta q^1 \varphi_\tau}{\delta \xi^2} \\
B_2(q^j) &= v \varphi_1 \frac{\delta}{\delta \xi^2} \left(\varphi_1 \frac{\delta q^2}{\delta \xi^2} \right) + v \varphi_2^2 \frac{\delta}{\delta \xi^2} \left(\frac{\delta q^2}{\delta \xi^2} \right) - \frac{\delta q^2 q^2 \varphi_2}{\delta \xi^2} - 2 \varphi_2^2 q^1 q^2 \frac{\delta J}{\delta \xi^1} \\
&\quad - \frac{\delta q^2 \varphi_\tau}{\delta \xi^2} + q^2 \frac{\delta \varphi_\tau}{\delta \xi^2},
\end{aligned}$$

and $\frac{\delta}{\delta \xi^i}$ means the numerical differentiation. The Runge-Kutta coefficients used in the present computations are:

$$\begin{aligned}
\beta_1 &= \frac{4}{15}, & \beta_2 &= \frac{1}{15}, & \beta_3 &= \frac{1}{6}, \\
\gamma_1 &= \frac{8}{15}, & \gamma_2 &= \frac{5}{12}, & \gamma_3 &= \frac{3}{4}, \\
\zeta_1 &= 0, & \zeta_2 &= -\frac{17}{60}, & \zeta_3 &= -\frac{5}{12}.
\end{aligned}$$

Note that, in the same way as in Akselvoll & Moin (1995), the non-linear term $q^2 q^2$ needs to be linearized for the present system to be solvable.

To make the intermediate flow field divergence free, we solve a projection function for $\phi = \bar{p}^k - \bar{p}^{k-1}$:

$$\Delta \phi = \frac{1}{2\beta_k \delta \tau} \frac{\delta q^{i*}}{\delta \xi^i},$$

where Δ is the Laplacian operator given by

$$\Delta \phi = D_i G^i(\phi) = \frac{\delta}{\delta \xi^1} \left(\frac{\delta \phi}{\delta \xi^1} + \frac{\delta \phi \varphi_1}{\delta \xi^2} \right) + \frac{\delta}{\delta \xi^2} \left[\varphi_1 \left(\frac{\delta \phi}{\delta \xi^1} + \frac{\delta \phi \varphi_1}{\delta \xi^2} \right) \right] + \varphi_2^2 \frac{\delta}{\delta \xi^2} \left(\frac{\delta \phi}{\delta \xi^2} \right). \quad (\text{II.41})$$

This Poisson equation is solved in Fourier space. Since the non-constant coefficients make the Laplacian non-invertible, we split the operator into two parts and solve the equation iteratively,

$$\tilde{\Delta} \phi^s = \frac{\delta}{\delta \xi^i} \left(\frac{\delta \phi^s}{\delta \xi^i} \right) = RHS^{s-1} = \frac{1}{2\beta_k \delta \tau} \frac{\delta q^{i*}}{\delta \xi^i} - (\Delta - \tilde{\Delta}) (\phi^{s-1}), \quad (\text{II.42})$$

where s is the iteration index. After ϕ converges, the volume flux components and pressure are updated by

$$q^k = q^{i*} - 2\beta_k \delta\tau G^i(\phi) \quad (\text{II.43})$$

and

$$\bar{p}^k = \bar{p}^{k-1} + \phi. \quad (\text{II.44})$$

II.A.2 Pressure equation

At the beginning of each time step, we solve a full pressure equation which is obtained by taking divergence of the equation (II.37):

$$\Delta \bar{p} = D_i (-T^i - N^i + \nu L^i). \quad (\text{II.45})$$

Note that the divergence of the uniform pressure gradient vanishes, which is true (within the machine accuracy) in the discrete case as well. The Laplacian Δ is the same as in (II.41), so the pressure equation is solved with the same iteration strategy as that used in (II.42).

The grid is discretized with a hyperbolic tangent stretching function and staggered in the wall-normal direction. q^2 is assigned on the family of gridpoints $j = 0, 1, 2, \dots$, where $j = 0$ corresponds to the lower wall, and $q^1, \bar{p}$ are assigned on the family of gridpoints $j = \frac{1}{2}, 1 + \frac{1}{2}, 2 + \frac{1}{2}, \dots$, where $j = \frac{1}{2}$ is midway between $j = 0$ and $j = 1$, and so on. Neumann boundary conditions for pressure are derived by enforcing continuity of the flow at the first interior gridpoint. We illustrate the procedure for the lower wall, $\xi^2 = -1$.

The discrete q^1 momentum at $j = \frac{1}{2}$ and the discrete q^2 momentum at $j = 1$, in an explicit Euler scheme, may be written as

$$\begin{aligned} \left. \frac{q^{1*} - q^{1^{k-1}}}{2\beta_k \delta\tau} \right|_{j=\frac{1}{2}} &= (-T^1 - N^1 - G^1 + \nu L^1 - P^1)|_{j=\frac{1}{2}} \\ \left. \frac{q^{2*} - q^{2^{k-1}}}{2\beta_k \delta\tau} \right|_{j=1} &= (-T^2 - N^2 - G^2 + \nu L^2 - P^2)|_{j=1}, \end{aligned} \quad (\text{II.46})$$

and the boundary condition for q^{2*} is

$$q^{2*}|_{j=0} = q^{2^k}|_{j=0}. \quad (\text{II.47})$$

Applying the discrete divergence operator to $\mathbf{q}^*$ at $j = \frac{1}{2}$, that is,

$$(D_i q^{i*})|_{j=\frac{1}{2}} = \frac{\delta q^{1*}|_{j=\frac{1}{2}}}{\delta \xi^1} + \frac{1}{h_{\frac{1}{2}}} (q^{2*}|_{j=1} - q^{2*}|_{j=0}),$$

where $h_{\frac{1}{2}}$ is the distance between gridpoints $j = 0$ and $j = 1$, and requiring it to be zero, we have

$$\begin{aligned}
 & -\frac{1}{2\beta_k\delta\tau} \left(\frac{\delta q^{1^{k-1}}|_{j=\frac{1}{2}}}{\delta\xi^1} + \frac{q^{2^{k-1}}|_{j=1} - q^{2^{k-1}}|_{j=0}}{h_{\frac{1}{2}}} \right) + \frac{1}{2\beta_k\delta\tau} \left(\frac{q^{2^k}|_{j=0} - q^{2^{k-1}}|_{j=0}}{h_{\frac{1}{2}}} \right) \\
 & = \frac{\delta}{\delta\xi^1} (-T^1 - N^1 - G^1 + \nu L^1 - P^1)|_{j=\frac{1}{2}} \\
 & \quad + \frac{1}{h_{\frac{1}{2}}} \left[(-T^2 - N^2 - G^2 + \nu L^2 - P^2)|_{j=1} + (N^2 + G^2)|_{j=0} - (N^2 + G^2)|_{j=0} \right],
 \end{aligned} \tag{II.48}$$

where the boundary nodes $q^{2^{k-1}}|_{j=0}$ and $(N^2 + G^2)|_{j=0}$ have been introduced. Note that the expression in the first parentheses on the left hand side vanishes since $\mathbf{q}^{k-1}$ is divergence free. We split (II.48) into two equations:

$$0 = \frac{\delta}{\delta\xi^1} (-N^1 - G^1)|_{j=\frac{1}{2}} + \frac{1}{h_{\frac{1}{2}}} [(-N^2 - G^2)|_{j=1} + (N^2 + G^2)|_{j=0}], \tag{II.49}$$

$$\begin{aligned}
 & \left. \frac{q^{2^k} - q^{2^{k-1}}}{2\beta_k\delta\tau} \right|_{j=0} + \left(h_{\frac{1}{2}} \frac{\delta T^1|_{j=\frac{1}{2}}}{\delta\xi^1} + T^2|_{j=1} \right) + N^2|_{j=0} \\
 & = -(G^2 + P^2)|_{j=0} + \nu \left(h_{\frac{1}{2}} \frac{\delta L^1|_{j=\frac{1}{2}}}{\delta\xi^1} + L^2|_{j=1} \right).
 \end{aligned} \tag{II.50}$$

where $h_{\frac{1}{2}} \frac{\delta P^1|_{j=\frac{1}{2}}}{\delta\xi^1} + P^2|_{j=1} = P^2|_{j=0}$ has been applied since P^i is divergence free. Note that (II.49) has the form of the simplified pressure equation, so it may be treated as the Poisson equation (II.45) evaluated at $j = \frac{1}{2}$. Equation (II.50) may be used to compute $G^2|_{j=0}$ which is the Neumann boundary condition for (II.45). Actually, by realizing that in theory T^i and L^i are divergence free, we have

$$h_{\frac{1}{2}} \frac{\delta T^1|_{j=\frac{1}{2}}}{\delta\xi^1} + T^2|_{j=1} \approx T^2|_{j=0}, \quad h_{\frac{1}{2}} \frac{\delta L^1|_{j=\frac{1}{2}}}{\delta\xi^1} + L^2|_{j=1} \approx L^2|_{j=0}.$$

Therefore, (II.50) is essentially the q^2 momentum equation evaluated at the lower boundary.

References

- AKSELVOLL, K. & MOIN, P. 1995 Large eddy simulation of turbulent confined coannular jets and turbulent flow over a backward facing step. *Tech. Rep.* TF-63. Thermosciences Division, Dept. of Mech. Eng., Stanford University.
- ARIS, R. 1962 *Vectors, tensors, and the basic equations of fluid mechanics*. Prentice-Hall, Englewood Cliffs, NJ.
- BEWLEY, T. R., MOIN, P. & TEMAM, R. 2001 DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *J. Fluid Mech.* **447**, 179–225.

- CARLSON, H. A., BERKOOZ, G. & LUMLEY, J. L. 1995 Direct numerical simulation of flow in a channel with complex, time-dependent wall geometries: A pseudospectral method. *J. Comput. Phys.* **121**, 155–175.
- FADLUN, E. A., VERZICCO, R., ORLANDI, P. & MOHD-YUSOF, J. 2000 Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J. Comput. Phys.* **161**, 35–60.
- GAL-CHEN, T. & SOMERVILLE, R. 1975*a* Numerical solution of the Navier-Stokes equations with topography. *J. Comput. Phys.* **17**, 276–310.
- GAL-CHEN, T. & SOMERVILLE, R. 1975*b* On the use of a coordinate transformation for the solution of the Navier-Stokes equations. *J. Comput. Phys.* **17**, 209–228.
- HIXON, R. 2000 Numerically consistent strong conservation grid motion for finite difference schemes. *AIAA Journal* **38** (9), 1586–1593.
- HODGES, B. R. & STREET, R. L. 1999 On simulation of turbulent nonlinear free-surface flows. *J. Comput. Phys.* **151**, 425–457.
- KANG, S. & CHOI, H. 2000 Active wall motions for skin-friction drag reduction. *Phys. Fluids* **12** (12), 3301–3304.
- LEI, Z., LIAO, G. & DELA PENA, G. C. 2000 A moving grid algorithm based on the deformation method. In *Computational Fluid Dynamics 2000: proceedings of the First International Conference on Computational Fluid Dynamics, ICCFD*. Kyoto, Japan.
- OGAWA, S. & ISHIGURO, T. 1987 A method for computing flow fields around moving bodies. *J. Comput. Phys.* **69**, 49–68.
- OLDROYD, J. G. 1950 On the formulation of rheological equations of state. *Proc. Roy. Soc. A* **200** (1063), 523–541.
- PULLIAM, T. H. & STEGER, J. L. 1980 Implicit finite-difference simulations of three dimensional compressible flow. *AIAA Journal* **18** (2), 159–167.
- ROSENFELD, M. & KWAK, D. 1991 Time-dependent solutions of viscous incompressible flows in moving co-ordinates. *Int. J. Numerical Methods in Fluids* **13**, 1311–1328.
- SHENG, C., TAYLOR, L. K. & WHITFIELD, D. 1995 Multigrid algorithm for three dimensional incompressible high-reynolds number turbulent flows. *AIAA Journal* **33** (11), 2073–2079.
- SHETH, K. S. & POZRIKIDIS, C. 1995 Effects of inertia on the deformation of liquid drops in simple shear flow. *Computers & Fluids* **24** (2), 101–119.
- SHYY, W., UDAYKUMAR, H. S., RAO, M. M. & SMITH, R. W. 1996 *Computational fluid dynamics with moving boundaries*. Taylor & Francis, Washington, DC.
- SMITH, R. & SHYY, W. 1995 Computation of unsteady laminar flow over a flexible two-dimensional membrane wing. *Phys. Fluids* **7** (9), 2175–2184.
- THOMAS, P. D. & LOMBARD, C. K. 1979 Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal* **17** (10), 1030–1037.
- TSANGARIS, S. & LEITER, E. 1984 On laminar steady flow in sinusoidal channels. *Journal of Engineering Mathematics* **18**, 89–103.
- VOKE, P. R. & COLLINS, M. W. 1984 Forms of the generalized Navier-Stokes equations. *Journal of Engineering Mathematics* **18**, 219–233.
- XU, S., REMPFER, D. & LUMLEY, J. 2003 Turbulence over a compliant surface: numerical simulation and analysis. *J. Fluid Mech.* **478**, 11–34.

Chapter III

Lattices for derivative-free optimization

This chapter has been submitted as

Bewley, T. R. & Luo, H. 2004 Lattices for derivative-free optimization. Part 1: Cubic ($\mathbf{Z}^n$), Hexagonal (A_2), FCC (D_3), BCC (D_3^*), Checkerboard (D_n), and Staggered (D_n^*). *SIAM J. Optim.*

Abstract

In the efficient derivative-free optimization of problems in several undetermined parameters, it is often beneficial to coordinate the search by performing it over a discrete set of points that is gradually refined. A general class of such discrete optimization algorithms for which convergence proofs are now available is called generalized pattern search (GPS) algorithms. In such algorithms, a positive basis is used to define the local pattern of test points at each poll step of the iterative search. This positive basis is selected from many possible choices based on the vectors from the current candidate minimum point (CMP) to the several nearby nodal points on the n -dimensional set of points being used to coordinate the search. However, if the usual n -dimensional cubic (a.k.a. Cartesian) mesh is used, the positive basis so constructed is either distributed nonuniformly in parameter space and based on nodal points other than the nearest neighbors to the CMP, or requires more new function evaluations than necessary to complete any given poll step. Such shortcomings can significantly reduce the effectiveness or efficiency of the poll step, thereby slowing the convergence of the optimization algorithm. Additionally, especially for large n , the cubic mesh does not “pack” or “cover” or “quantize” the parameter space very uniformly, thus requiring an excessive number of mesh refinements in order to approximate the actual minimum point sought with a desired degree of accuracy. The present paper considers alternative point sets, arranged as *lattices*, for GPS algorithms that may help to mitigate these shortcomings. These lattices may be thought of as n -dimensional extrapolations of the face-centered cubic and body-centered cubic crystalline structures, and are shown to provide significantly improved convergence when used as the underlying framework for GPS algorithms.

III.1 Introduction

Multidimensional optimization problems are most easily solved when derivative (that is, gradient, and sometimes Hessian) information can be computed or approximated. However, when the function to be minimized is inherently noisy, such as when it is obtained from lab experiments or as a statistical measure of a computer simulation of a chaotic system such as turbulence, it is often not feasible to perform derivative-based optimizations. In such cases, derivative-free (that is, function-based) optimization strategies are the methods of choice.

In the function-based optimization of a single variable (otherwise known as a line search), a key element of a robust, efficient optimization scheme is that it not “stall” by performing function

evaluations that are “too close” to one another too early on in the optimization. Implementing this concept involves a certain degree of *ad hoc* tuning of the numerical algorithm, but effective and general methods are by now quite well established (see, for example, Brent’s method in Press *et al.*, 1986).

In the function-based optimization of several variables, the fundamental idea of not performing function evaluations “too close” to one another in order to “get the most” out of each function evaluation still applies. An effective class of methods that has emerged to implement this fundamental idea, called generalized pattern search (GPS) algorithms, is to perform the optimization on a lattice covering the (appropriately nondimensionalized) constrained or unconstrained parameter domain under consideration, optimizing (at least, locally) the function evaluated on selected nodal points on this lattice, then refining the lattice and repeating the optimization process until a desired degree of localization of the optimum is obtained.

Local convergence proofs for GPS algorithms are by now quite well established (Torczon, 1997; Booker *et al.*, 1999; Coope & Price, 2001). In order to insure convergence, the essential step of any GPS algorithm once a candidate minimum point (CMP) is identified is the poll step, which in effect tests several nearby nodal points around the CMP. These test points are selected such that the set of vectors to each of these test points from the CMP create a “positive basis” (that is, a set of vectors such that any point in the vicinity of the CMP can be reached with a linear combination of these vectors with non-negative coefficients). If, at any time during the poll step, any of these test points turns out to have a smaller function value than the current CMP, the current poll step is terminated prematurely, a new CMP is identified, and a new poll step is initiated around the new CMP. If, however, all neighboring test points forming the positive basis around the CMP turn out to have larger function values than the function value at the CMP, then the CMP is taken to be a (local) minimizer on the current lattice. The lattice is then refined and the process repeated. Since the polling process is relatively “expensive” in terms of the number of function evaluations it requires, it is generally desirable to design the poll process to require the minimum number of new function evaluations possible.

To incorporate constraints, the lattice may be restricted to a range of acceptable values, and the routine forced to evaluate the function only on those “feasible” nodal points which satisfy the constraints. When a CMP occurs on a constraint boundary, in order to insure that the optimization doesn’t stop prematurely, it is important that the boundary of the lattice be “fit” to the constraint surface, and at each poll step a positive basis be tested within the constraint surface in addition to a

poll vector directed inward to the feasible domain from the constraint surface.

Note that GPS algorithms offer a lot of flexibility as to the choice of both the lattice overlying the parameter space and the pattern of test points selected from this lattice to be explored at each poll step, as long as the vectors from the CMP to these test points form a positive basis, which is required to insure convergence of the algorithm. Note that the directions of the positive basis vectors at each poll step are restricted to the directions defined by the vectors from the CMP to the nodal points in the vicinity of CMP, as the function evaluations are constrained to lie on the pre-defined lattice in such algorithms. Note also that, in current practice, the standard n -dimensional cubic lattice $\mathbf{Z}^n$, also referred to as a point lattice or the Cartesian mesh or grid, is used as the underlying lattice in almost all lattice-based optimization packages, such as the Genetic Algorithm and Direct Search Toolboxes in Matlab. This choice is so widespread that this lattice often underlies lattice-based optimization algorithms implicitly, without even being explicitly identified in the formulation of the optimization algorithm. The present work discusses some attractive alternatives to this default choice.

In a particularly significant development, Booker *et al.* (1999) proposed a powerful extension of the GPS algorithm that alternates between a *search* step and the standard GPS *poll* step. By utilizing the poll step, this method retains the local convergence proofs of GPS algorithms. However, by additionally utilizing the search step, the rate of convergence of the algorithm, in addition to the function value at the local optimum found by the algorithm, can be substantially improved. Any of a number of various strategies can be used during the search step to look for a nodal point on the lattice with an improved function value. In particular, Booker *et al.* (1999) proposed and characterized a new strategy, referred to as the Surrogate Management Framework (SMF), in which a “surrogate” approximating function (such as a Kriging function) is fit to the available function evaluations on the lattice. During the search step, the minima of this surrogate function are used to provide estimates of new areas of interest in the domain to explore with new function evaluations on the lattice. (This is akin to the inverse parabolic interpolations performed by Bent’s method in one-dimensional line searches.) The minima of the surrogate function may themselves be found fairly quickly with any of a variety of techniques, such as so-called evolution-based strategies, as the surrogate function is designed to be quite easy to compute. Other nodal points in the domain may also be probed during the search step in order to explore regions in the parameter space which are still relatively poorly characterized, though initializing the search using a relatively coarse lattice typically reduces this “extra probing” of the domain that is necessary to achieve an adequate result.

The SMF has proven to be especially suitable for problems in which the objective function is very expensive to evaluate, and has quickly grown in popularity for such problems. For example, the SMF has been successfully applied by Marsden *et al.* (2004) to optimize the shape of an airfoil in order to reduce the noise generated by the time-dependent flow near its trailing edge. In such a problem, which is typical of a wide range of related derivative-free optimization problems one might consider in fluid mechanics, each function evaluation entails an unsteady flow simulation requiring many hours of supercomputer time, and maximally-efficient optimization strategies are essential.

In the SMF, in addition to optimizing the efficiency of the poll steps, a well designed positive basis might also help to improve the accuracy of the surrogate function itself, thereby also improving the efficiency of the search steps. It appears as if such an improvement in the search steps would be a fairly problem-specific improvement, so this possibility is not explored further in the present paper. Rather, the focus of the present paper is on the several beneficial effects a well designed lattice and a carefully chosen positive basis has on the poll steps of GPS algorithms.

III.2 Favorable properties for packings used by GPS algorithms

The discretization of the parameter space onto an ordered set of nodal points, or a packing, is necessary for GPS algorithms, as described above. However, there is tremendous flexibility in the selection of this packing. We now identify four general characteristics that render such a packing well suited for GPS algorithms:

III.2.1 Enumeration

The packing, and the nearest neighbors of each nodal point in the packing, should be straightforward to enumerate. As the intent is that the function evaluations performed during the course of the optimization be sparsely distributed, it should be straightforward to calculate various nodal points of the packing when needed, without storing (unnecessarily) the coordinates of a high-dimensional array of node locations in the computer memory.

III.2.2 Refinement

When the packing is refined by a factor of two, the nodal points of the original packing should also be nodal points of the refined packing. This property insures that function evaluations on the original set of nodal points may be reused with maximal effectiveness in the subsequent opti-

mization on the refined set of nodal points, and thus a minimal number of new function evaluations will be needed in the optimization on the refined set of nodal points.

III.2.3 Uniformity

The packing should uniformly “pack” or “cover” or “quantize” the domain. This implies that, at every level of refinement, one of the nodal points in the packing is as close as possible to actual minimizer, wherever this minimizer happens to be in the domain. This facilitates the determination of a nodal point in the packing that approximates the actual minimum point with a minimal amount of error after the fewest possible number of refinements of the packing in the optimization algorithm.

III.2.4 Neighborhood

The nodal points in the neighborhood of any potential CMP in the packing should be arranged to facilitate the selection of a positive basis that a) requires as few new function evaluations as possible, b) is evaluated on the nearest neighbors to the CMP in the packing (if possible), and c) is distributed as uniformly as possible in terms of their directions from the candidate minimum point. These properties insure that the polling steps are completed as quickly as possible and effectively isolate the optimum of the function to the smallest region possible before moving to the refined packing, thereby (potentially) leading to a reduced number of points over which the algorithm will search on the refined packing. Also, in the case of the SMF, increased uniformity of test points at each poll step may lead to improved accuracy of the surrogate approximating function, as mentioned previously.

III.2.5 Objective of paper

This paper considers alternatives to the standard cubic lattice for GPS algorithms in n dimensions (focusing on cases with $2 \leq n \leq 15$) which, at the cost of being just slightly more difficult to enumerate, are as well suited or superior to the standard cubic lattice in terms of the other properties listed above.

III.3 Notation

This paper utilizes an extrapolation of the body-centered cubic and face-centered cubic crystalline structures to n dimensions, and discusses the suitability of these structures for forming the lattice underlying a pattern search algorithm. In fact, a substantial body of literature has grown around the topic of n -dimensional crystallography (see, e.g., Janssen *et al.*, 1999; Janner, 2001; Janssen *et al.*, 2002). The field has its origins in chemistry and physics for the case $n = 3$, and can be traced back to the second half of the nineteenth century. In fact, Hilbert's eighteenth problem (of the 23 problems stated by Hilbert (1900) at his historic address at the 1900 International Congress of Mathematicians) is closely related to this field of study. Schwarzenberger (1980, pages 132-135) provides a nice review of some of this older literature.

Many of the involved concepts of n -dimensional crystallography, such as detailed characterizations of symmetry groups, are not needed in this work. The more relevant body of literature for the present study is that of n -dimensional sphere packing, which has important applications in coding theory. The definitive reference on this and related subjects is the lucid and comprehensive textbook by Conway & Sloane (1999). We now summarize the central concepts and notation used in the present work, following the notation of Conway and Sloane to the maximum extent possible¹.

- The **inner product** of two vectors: $(u, v) = u \cdot v = \sum_{i=1}^n u_i v_i = u_1 v_1 + u_2 v_2 + \dots + u_n v_n$.
- The **vector length**: $|u| = \sqrt{(u, u)}$.
- The **angle between two vectors**: $\cos \angle(u, v) = \frac{(u, v)}{|u| |v|}$.
- The **volume of a sphere** of some radius r , defined as the set of all points in the domain of distance less than or equal to r from its center point, may be computed by straightforward integration. It is given by $V_n r^n$, where

$$V_2 = \pi, \quad V_3 = \frac{4\pi}{3}, \quad V_n = \frac{2\pi}{n} V_{(n-2)}.$$

- The **projection** p of a vector f onto the plane normal to the vector r is given by

$$p = f - (f, r) \frac{r}{|r|^2}. \quad (\text{III.1})$$

Note that $(r, p) = 0$.

- A **positive basis** is defined as a set of m vectors, where $n + 1 \leq m \leq 2n$, which span, with non-negative coefficients, all directions in the domain in the neighborhood of a CMP, with no subset

¹Note that, for brevity as well as generality, we suppress the use of the prefix "hyper-" oft used in $n > 3$ dimensions, preferring the simpler forms *sphere*, *cube*, *plane*, etc., instead of the more exotic constructs *hypersphere*, *hypercube*, *hyperplane*, etc.

of these vectors similarly spanning the same space. A **maximal positive basis** is a positive basis consisting of $2n$ vectors, and a **minimal positive basis** is a positive basis consisting of $n + 1$ vectors.

• **Necessary and sufficient condition for the set $\{v_1, v_2, \dots, v_m\}$ to be a positive basis** (paraphrased from Theory 3.6 of Davis (1954a)): *the matrix containing the vectors $\{v_1, v_2, \dots, v_m\}$ as columns has full row rank, and it is possible to write, for all i ,*

$$-v_i = \sum_{\substack{j=1 \\ i \neq j}}^m \alpha_j v_j \quad \text{with all } \alpha_j \geq 0,$$

whereas it is not possible to write this condition if any of the vectors v_i is eliminated from the set and m is reduced by one.

• The **skewness** of a minimal positive basis, S , is defined here as $S = (\alpha_{\max}/\alpha_{\min}) - 1$, where $\alpha_{\max}$ and $\alpha_{\min}$ are the maximum and minimum angles, respectively, between the vectors of the minimal positive basis. Note that $S = 0$ corresponds to a uniformly distributed minimal positive basis.

• A **packing** is an ordered set of nodal points in n -dimensional space.

• A **lattice packing**, or **lattice**, is a packing such that, if shifted such that the origin is a nodal point in the packing, then, if u and v are any two nodal points in the packing, $u + v$ and $u - v$ are also nodal points in the packing. Any n -dimensional lattice L_n has a **dual lattice**, L_n^* , given by

$$L_n^* = \{x \in \mathbf{R}^n : (x, u) \in \mathbf{Z} \quad \forall u \in L_n\}.$$

Two lattices M and M' are said to be **congruent** if one may be obtained by appropriately shifting, scaling, and rotating the other; this condition is denoted $M \cong M'$.

Note that lattice packings (such as the cubic, hexagonal, FCC, BCC, checkerboard, and staggered lattices used in this work) automatically satisfy the first and second properties identified in §2, whereas nonlattice packings (such as hexagonal close packing and diamond packing) often fail to satisfy one or both of these properties. Additionally, lattice packings have the convenient features that the distribution of nodal points in the neighborhood of any potential CMP in the domain is identical, thereby simplifying the implementations of the poll step in the optimization code. Thus, the rest of this paper focuses exclusively on lattice packings.

• The **packing radius** of a lattice, ρ , is the maximal radius of the spheres in a set of identical nonoverlapping spheres centered at each nodal point.

- The **packing density** of a lattice, Δ , is the fraction of the volume of the feasible domain included within a set of identical non-overlapping spheres of radius ρ centered at each nodal point on the lattice. Lattices that maximize this metric are referred to as **close-packed**.
- The **covering radius** of a lattice, R , is the maximum distance between any point in the feasible domain and the nearest nodal point on the lattice.
- The **covering thickness** of a lattice, Θ , is the number of spheres of radius R containing an arbitrary point in the domain, averaged over the domain.
- The **Voronoi cell** of a nodal point on a lattice, $V(P_i)$, consists of all points in the domain that are at least as close to the nodal point P_i as they are to any other nodal point P_j .
- The **mean squared quantization error per dimension** of a lattice, G , is the average mean square distance of any point in the domain and the nearest nodal point, divided by n and normalized by the appropriate power of the volume of the Voronoi cell. Shifting the origin to be at the centroid of a Voronoi cell $V(P_i)$, it is given by

$$G = \frac{\frac{1}{n} \int_{V(P_i)} |x|^2 dx}{\left[\int_{V(P_i)} dx \right]^{1+\frac{2}{n}}}.$$

Zador (1982) noticed that this metric may be reduced by quantizing space with lattices other than $\mathbf{Z}^n$, which is one of the primary motivations of the present paper.

- The **kissing number** of a lattice, τ , is the number of nearest neighbors to any given nodal point in the lattice. In other words, it is the number of spheres of radius ρ centered at the nodal points that touch, or “kiss”, a sphere centered at any given nodal point.

Note that the packing density Δ , the covering thickness Θ , and the normalized mean-squared quantization error G are related but distinct quantifications of the uniformity of the lattice, whereas the kissing number τ is an indicator of the degree of flexibility available when selecting a positive basis from nearest neighbors on the lattice.

III.4 Comparison of the cubic and hexagonal lattices for 2D optimization

We first compare the suitability of the two lattices depicted in Figure III.1 in terms of the four properties listed in §2 for GPS optimization in the case with $n = 2$.

III.4.1 Enumeration

Assuming hereafter that the coordinates are normalized by the unit cell size in each direction, the (x,y) coordinates of the nodal points on the **2D cubic lattice**, $\mathbf{Z}^2$, are enumerated by (i_1, i_2) , where i_1 and i_2 are any two integers. Similarly, the (x,y) coordinates of the nodal points on the **2D hexagonal lattice**, A_2 , are enumerated by $(i_1 + i_2 \frac{1}{2}, i_2 \frac{\sqrt{3}}{2})$. Constraints may be applied simply by restricting the acceptable range of i_1 and/or i_2 .

III.4.2 Refinement

When refined by a factor of two, the nodal points on the original lattice are also part of the refined lattice for both the 2D cubic and hexagonal lattices; the additional grid lines after the refinement are indicated with dashed lines in Figure III.1.

III.4.3 Uniformity

As discussed in §3, there are three natural techniques to quantify the uniformity of a lattice. The first technique is to consider a set of identical non-overlapping spheres (in 2D, disks) centered at each nodal point, as depicted in Figure III.2, and evaluate the maximal fraction of the domain contained in such spheres. As described in §3, this is referred to as the packing density of the lattice, Δ . The second technique is to consider a set of overlapping spheres centered at each nodal point that are just large enough that every point in the domain is covered by at least one sphere, as depicted in Figure III.3, and to quantify the degree of overlap that results by determining the average number of spheres containing any point in the domain. This is referred to as the covering thickness of the lattice, Θ . The third technique is to evaluate the mean squared quantization error per dimension of the lattice, G , as defined in §3.

The measure G is a quantification of the average quantization error. The measure Θ , on the other hand, might be thought of as a quantification of the “worst case” quantization error, as it is based on the most remote points in the domain (referred to as “deep holes”) that are farthest from the nodal points of the lattice. The measure Δ , presumably, is a quantification of lattice uniformity that might be thought of as somewhere between these two extremes. All three measures, in addition to the kissing number τ and skewness parameter S , are straightforward to calculate, and are listed in the 2D case in Table III.1. Note that the hexagonal lattice is superior to the 2D cubic lattice by all five of these measures.

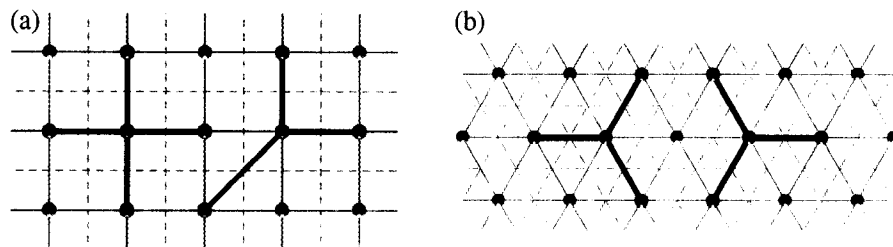


Figure III.1: Layout of the 2D cubic (left) and hexagonal (right) lattices.

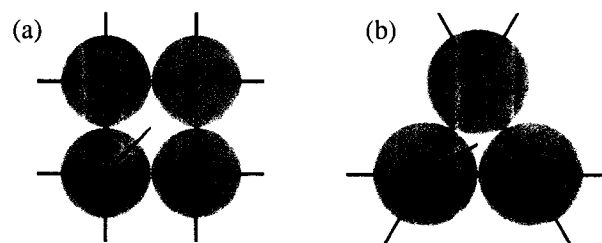


Figure III.2: Packing of the 2D cubic (left) and hexagonal (right) lattices.

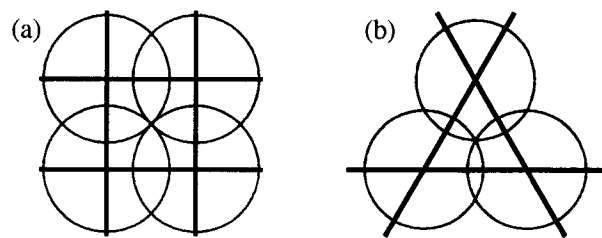


Figure III.3: Covering of the 2D cubic (left) and hexagonal (right) lattices.

n	lattice	Δ	Θ	G	τ	S
2	cubic ($\mathbf{Z}^2$)	0.785	1.571	0.08333	4	$0.5^{(1)}$
	hexagonal ($A_2 \cong A_2^*$)	$0.907^\dagger$	$1.209^\dagger$	$0.08019^\dagger$	$6^\dagger$	0

Table III.1: Characteristics of the 2D cubic and hexagonal lattices: the packing density Δ , the covering thickness Θ , and the mean squared quantization error per dimension, G , quantifying the lattice uniformity, the kissing number τ quantifying the flexibility available in selecting the positive basis from nearest neighbors, and the skewness parameter S quantifying the uniformity of the best (in terms of angular uniformity) minimal positive basis (with $S = 0$ indicating that a uniform distribution is possible). A number in parentheses in the superscript indicates that that number of polling points in the minimal positive basis referred to are *not* nearest neighbors to the CMP, but are taken from the next shell of neighbors, as no minimal positive bases can be constructed from nearest neighbors on the corresponding lattice. The dagger (†) denotes a value of a metric that is *known* to be optimal among all lattices of the same dimension, whereas the double dagger (‡) denotes a value of a metric that is *thought* to be optimal among all lattices of the same dimension. The reader is referred to Conway & Sloane (1999) for further discussion of the alternative lattice names, such as A_2^* , etc. See Table III.2 for the $n = 3$ case and Table III.3 for $4 \leq n \leq 15$.

III.4.4 Neighborhood

For the 2D cubic lattice, there are two ways to form a positive basis from nearby nodal points, neither of which is ideal. One way is simply to select the $2n = 4$ nearest neighbors to form a maximal positive basis, as indicated by the positive basis marked in the neighborhood of the CMP in the left half of Figure III.1a. Though the angle between the basis vectors are uniform, this positive basis consists of $(2n) - (n + 1) = 1$ more test point than it needs to, thus typically requiring an extra function evaluation at each poll step. If a minimal positive basis (using $n + 1 = 3$ test points) is desired in the 2D cubic case, it is necessary to select at least one test point which is *not* a nearest neighbor of the CMP, as indicated by the positive basis marked in the neighborhood of CMP in the right half of Figure III.1a. As is easily seen, the angles between the basis vectors in this case are highly nonuniform, with 90° between two of the vectors and 135° between either of these and the third. The relative lengths of these basis vectors are also nonuniform, with one $\sqrt{n} \approx 1.414$ times longer than the other three. On the other hand, for the 2D hexagonal lattice, there are two natural ways to form a positive basis (as indicated in Figure III.1b), both of which are ideal in that they are (a) minimal (requiring $n + 1 = 3$ function evaluations to complete the poll step), (b) evaluated on nearest neighbor nodal points, and (c) distributed uniformly in parameter space.

III.4.5 Summary of the 2D case

As discussed above and quantified in Table III.1, the hexagonal lattice is either equivalent or superior to the cubic lattice in terms of *all* four properties identified in §2 for lattice-based optimization in two dimensions using GPS algorithms. Therefore, the hexagonal lattice is (or, at least, should be) the preferred choice for unconstrained lattice-based optimization in two dimensions. (When constraints are applied, the problem of selecting the most suitable lattice becomes case specific, and a lattice which happens to fit the given constraint boundaries might be preferred.)

III.5 Comparison of the cubic, FCC, and BCC lattices for 3D optimization

We now describe briefly the three lattices depicted in Figures III.4-III.6 and compare their suitability in terms of the four properties listed in §2 for GPS optimization in the case $n = 3$. Note that all three of these lattices appear prominently in standard chemistry textbooks (see, e.g.,

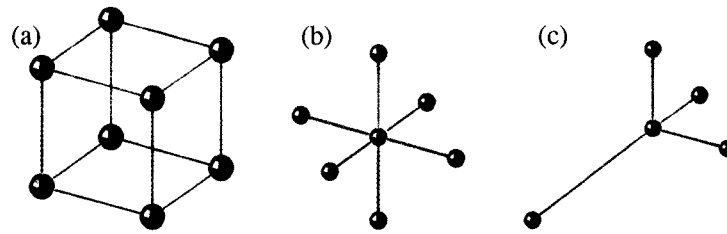


Figure III.4: The 3D cubic lattice. (a) The unit cell. (b) The arrangement of the $2n = 6$ nearest neighbors of a CMP (blue), which form a maximal positive basis. (c) A minimal positive basis using $n + 1 = 4$ nodal points that are *not* nearest neighbors of the CMP.

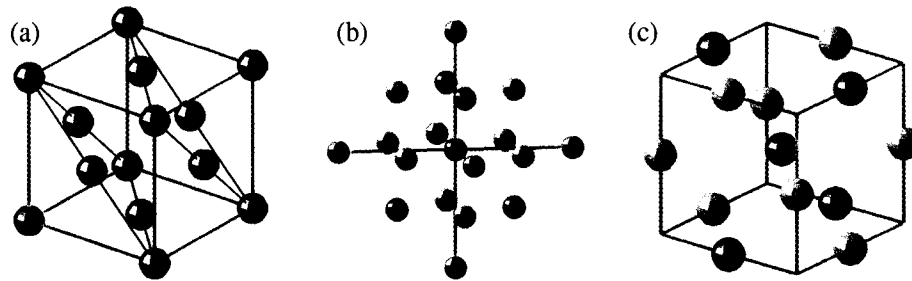


Figure III.5: The 3D FCC lattice. (a) The unit cell. (b) and (c) The arrangement of the $n2^{n-1} = 12$ nearest neighbors (yellow and red) of a CMP (dark blue). From a viewpoint normal to the diagonal planes indicated in (a), one sees several layers of nodal points on top of each other, with the nodal points in each layer configured in a hexagonal pattern, and the nodal points in one layer lying directly over the voids in the neighboring layers; this provides one (of two) “close-packed” configurations in 3D, and is sometimes referred to as cubic close packing.

Navrotsky, 1994).

III.5.1 Enumeration

Assuming as before that each coordinate direction is normalized by the unit cell size in that direction, the **3D cubic lattice**, $\mathbf{Z}^3$, is enumerated by (i_1, i_2, i_3) , where each i_j is an integer.

The **face-centered cubic (FCC) lattice**, D_3 , is a straightforward generalization of the 3D cubic lattice. It is formed by taking the union of four 3D cubic lattices, one of which might be described as the “base mesh” defining the unit cells, with the three others offset from the base

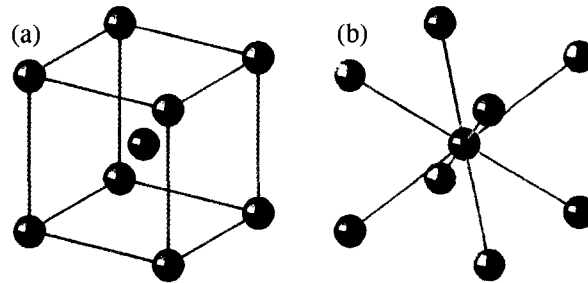


Figure III.6: The 3D BCC lattice. (a) The unit cell. (b) The arrangement of the $2^n = 8$ nearest neighbors of a CMP (blue). Note that, in this case, there are two families (red and green) of $n + 1 = 4$ nodal points forming regular tetrahedra around the CMP, thereby forming uniformly distributed minimal positive bases.

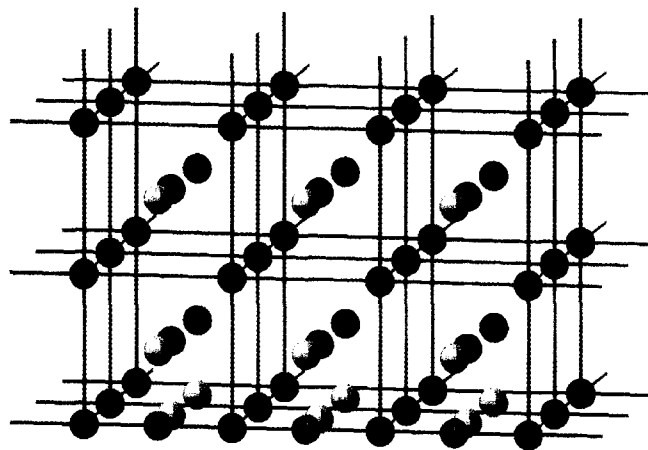


Figure III.7: The augmented BCC lattice.

mesh by $1/2$ of the unit cell size in exactly 2 of the coordinate directions, which we may denote $\mathbf{Z}^3 \cup ([2] + \mathbf{Z}^3)$ where $[2]$ denotes all possible permutations of the vector $(0, \frac{1}{2}, \frac{1}{2})$.

The **body-centered cubic (BCC) lattice**, D_3^* , is another straightforward generalization of the 3D cubic lattice. It is formed by taking the union of two 3D cubic lattices, one of which might again be described as the base mesh defining the unit cells, with the other offset from the base mesh by $1/2$ of the unit cell size in all of the coordinate directions, which we may denote $\mathbf{Z}^3 \cup ([1] + \mathbf{Z}^3)$ where $[1] = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$.

Constraints are applied by restricting the range of the component lattices in one or more directions. Note that the 3D cubic and BCC lattices restricted in one coordinate direction form 2D

cubic lattices on the boundary, whereas the FCC lattice restricted in one coordinate direction forms a rotated 2D cubic lattice on the boundary.

It is shown below (§5.2-5.5) that the BCC lattice appears to be superior in many relevant respects to the cubic lattice. It is thus natural to propose an “augmented” BCC packing which preserves these superior BCC characteristics for the optimization problem even when constraints are binding. For constraints which restrict the packing in one coordinate direction at a time, one might propose the strategy depicted in Figure III.7. The augmented packing depicted in this figure is a BCC lattice on the interior (red and blue points), with additional points (yellow) on the 2-dimensional boundaries where one constraint is binding and more additional points (cyan) on the 1-dimensional boundaries where two constraints are binding. This strategy is not well justified in the 3D case, however, as the 2D and 1D versions of the BCC lattice do not exhibit favorable properties (in fact, they simply reduce to $\mathbf{Z}^2$ and $\mathbf{Z}^1$). On the other hand, this strategy is easily extended to the n -dimensional extrapolation of the BCC lattice, D_n^* , as described in §6 and referred to as the “staggered” lattice. Following the same procedure as described here for the augmented BCC packing, the n -dimensional augmented staggered packing reduces to $(n - 1)$ -dimensional and $(n - 2)$ -dimensional augmented staggered packings when one or two constraints are binding, thereby retaining the favorable properties of the staggered lattice on the boundaries. The approach may be further extended to facilitate cases when even more constraints are binding simultaneously — in such cases, however, the optimization problem would most likely be considered as poorly parameterized, and some of the binding inequality constraints may be replaced by equality constraints during the optimization, thereby reducing the dimension of the optimization problem and substantially accelerating convergence.

III.5.2 Refinement

As may be verified by inspection, when refined by a factor of two, the nodal points on the original lattice are also part of the refined lattice for all three lattices depicted in Figures III.4-III.6.

III.5.3 Uniformity

All three metrics discussed in §4.3 to quantify the uniformity of the lattice (namely, Δ , Θ , and G), are also straightforward to calculate in the 3D case, as reported in Table III.2. Note that the BCC and FCC lattices are both superior to the 3D cubic lattice by all three measures.

n	lattice	Δ	Θ	G	τ	S
3	cubic ($\mathbf{Z}^3$)	0.524	2.721	0.08333	6	0.392 ⁽¹⁾
	FCC (D_3)	0.740 [†]	2.094	0.07875	12 [†]	0.333
	BCC (D_3^*)	0.680	1.464 [†]	0.07854 [†]	8	0

Table III.2: Characteristics of the 3D cubic, BCC, and FCC lattices. See Table III.1 legend for details.

III.5.4 Neighborhood

Figure III.4a depicts the unit cell of the cubic lattice in $n = 3$ dimensions. The situation is analogous to the 2D cubic case, as discussed in §4.4. As depicted in Figure III.4b, the arrangement of the $2n = 6$ nearest neighbors of a CMP (blue) in the cubic lattice form a maximal positive basis, meaning that it is impossible to select a positive basis with any subset of these nearest neighbors. As depicted in Figure III.4c, using $n + 1 = 4$ nodal points, one of which is *not* a nearest neighbor of the CMP, it is possible to construct a minimal positive basis. Note that the angles between the minimal positive basis vectors in this configuration are nonuniform, with 90° between any pair of three of the vectors and $\arccos(-1/\sqrt{n}) \approx 125.3^\circ$ between any of these three and the fourth. The relative lengths of these basis vectors are also nonuniform, with one $\sqrt{n} \approx 1.732$ times longer than the other three. These qualities make the standard cubic lattice less than ideal for pattern search algorithms.

Figure III.6a depicts the unit cell of the BCC lattice. As depicted in Figure III.6b, the arrangement of the $2^n = 8$ nearest neighbors of a CMP (blue) in the BCC lattice may be divided into two families (red and green) of $n + 1 = 4$ nodal points, each forming regular tetrahedra around the CMP. Either family may be selected to form uniformly distributed minimal positive bases.

Figure III.5a depicts the unit cell of the FCC lattice. As depicted in Figures III.5b and III.5c, the arrangement of the 12 nearest neighbors (yellow and red) of a CMP (dark blue) do not facilitate the construction of a uniformly distributed minimal positive bases from nearest-neighbor lattice points. However, it is straightforward to select a non-uniformly distributed minimal positive basis using $n + 1 = 4$ of these $n2^{n-1} = 12$ nearest neighbors, such as those indicated in red. These $n + 1$ points are all equidistant from the CMP and more uniformly distributed than the minimal positive basis in the 3D cubic case (Figure III.4c), with 90° between some of the vectors and 120° between others.

III.5.5 Summary of the 3D case

As discussed above and quantified in Table III.2, the BCC and FCC lattices are both either equivalent or superior to the cubic lattice in terms of *all* four properties identified in §2 for lattice-based optimization in three dimensions using GPS algorithms. Therefore, one of these two lattices is (or, at least, should be) the preferred choice for unconstrained lattice-based optimization in three dimensions.

Note that the BCC lattice was originally proposed for lattice-based optimization strategies in very early work (in 1954 and 1969 respectively) by Davis (1954*b*) and Box *et al.* (1969). This approach, referred to as fractional factorial design, was far ahead of its time, predating both the use of positive bases to insure convergence in lattice-based optimization and, in fact, the digital computer itself. As Davis and Box were not aware of the role of positive bases in the convergence of lattice-based optimization strategies, the advantages of the BCC lattice for lattice-based optimization were not fully recognized (until now).

III.6 Extending the approach to higher dimensions

As reviewed in Conway & Sloane (1999), there are a variety of families of n -dimensional lattices that reduce to the familiar BCC and FCC lattices in the special case $n = 3$. For brevity, in the present work, we restrict our attention to the following three lattices:

- The **n -dimensional cubic lattice**, $\mathbf{Z}^n$, though perhaps difficult (if not impossible) for humans to visualize, is quite straightforward to work with. It is enumerated by $(i_1, i_2, \dots, i_n)$, where each i_j is an integer.
- The **n -dimensional checkerboard lattice**, D_n , is enumerated by $(i_1, i_2, \dots, i_n)$, where each i_j is an integer *and* the sum $i_1 + i_2 + \dots + i_n$ is even. In three dimensions (scaled appropriately), this lattice is the FCC lattice studied previously.
- The **dual of the n -dimensional checkerboard lattice**, D_n^* , which for brevity we will call the **staggered lattice**, is defined by a simple combination of two n -dimensional cubic lattices offset by the vector $[1] = (\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$. In three dimensions (scaled appropriately), this lattice is the BCC lattice studied previously.

Except for some notable exceptions (identified in Table III.3), it is often not possible to select uniformly distributed minimal positive bases from neighboring nodal points on these lattices

in n -dimensions when $n > 3$. Further, it is difficult (if not impossible) to determine the most uniformly distributed minimal positive basis possible in any given case by inspection. Thus, a very practical problem from the standpoint of actually performing GPS optimization using these lattices is the identification of the most uniformly distributed minimal positive bases possible in such cases. This may be thought of as a discretized version of Tammes' problem, which is summarized nicely in Conway & Sloane (1999) as the question: "Where should m inimical dictators build their palaces on a planet so as to be as far away from each other as possible?". In the present work, we have an n -dimensional planet and would like to distribute optimally $m = n + 1$ such dictators on a large but discrete set of locations where they may build their palaces.

We have performed an intensive investigation to determine "stencils" for the most uniformly distributed minimal positive bases possible for both the staggered and checkerboard lattices for dimensions $n = 4$ to $n = 15$. The results of this search are listed in the appendix. To simplify the notation used, the staggered and checkerboard stencils listed in the appendix are shifted and scaled to put the CMP at the origin and the nodal points at integer coordinates. Note that, for economy of notation, the values "+1" and "-1" are denoted simply by "+" or "-" in the appendix.

In the checkerboard case, the nearest neighbors are located at all possible permutations of $(0, \dots, 0, \pm 1, \pm 1)$ for any choice of signs.

In the staggered case, the nearest neighbors are located, for $n \leq 3$, at $(\pm 1, \pm 1, \dots, \pm 1)$ for any combination of signs. The nearest neighbors for $n \geq 5$ are located at all possible permutations of $(0, \dots, 0, \pm 2)$ for any choice of sign. For $n = 4$, the nearest neighbors are located at the union of both of these sets of nodal points.

III.6.1 Exhaustive search strategy

The search for the optimized stencils for the most uniformly distributed minimal positive bases was in fact quite involved. When the dimension n is fairly low, an exhaustive search may be completed. Exploiting symmetries to the maximum extent possible and immediately disqualifying all cases that were not suitable candidates², we were able to explore exhaustively the staggered problem through $n = 8$ and the checkerboard problem through $n = 6$. Thus, it is known with certainty that the bases listed in the appendix for these cases are the most uniformly distributed bases possible.

²For example, those for which the angle between any two basis vectors is less than some prespecified threshold, which would fail to be optimal, or those for which two basis vectors are exactly 180° apart, which would fail to provide a minimal positive basis.

To search exhaustively for the most uniformly distributed minimal positive basis possible using nearest neighbor points in the staggered case, the coordinates were first shifted to put the CMP at $(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$, and the nearest neighbor points (the vertices of an n -dimensional cube) were enumerated from 0 to $2^n - 1$. The enumeration of each vertex was then represented in binary notation, which also (coincidentally) gave its physical coordinates. (For example, the third vertex in 3D has coordinates $[0, 1, 1]$, which is also the number three in binary notation.) The angular distance between any two vertices could then be characterized by the *Hamming distance* (Conway & Sloane, 1999, page 75), as provided by bitwise comparison of its enumeration in binary form. This led to a very fast code. The candidate combinations were then tested to see if they were indeed positive bases using the test described in §3. As the exhaustive search continued (using a fairly streamlined recursive algorithm), the most uniformly distributed minimal positive basis found so far was saved, and all others discarded.

The exhaustive search for the most uniformly distributed minimal positive basis in the checkerboard case was quite similar, with a different enumeration of the neighboring points.

III.6.2 “Force”-based search strategy

The difficulty of exploring all possible permutations to find the most uniformly distributed minimal positive basis grows extremely rapidly with the dimension of the problem. Thus, in high dimensional problems, even though the exhaustive search code was quite efficient, an alternative (non-exhaustive) search strategy proved to be quite useful. With this strategy, in both the staggered and checkerboard cases, $n + 1$ nearest neighbors to the CMP were first selected at random. An equal “charge” was then assigned to each of these points, as if each point was a charged particle constrained to lie on an n -dimensional sphere, and the “total force” f_j was computed at each “particle”. After computing the magnitude m_j of the force f_j projected onto the surface of the sphere [using (III.1)], the particle with the greatest value of m_j (that is, the particle that was “furthest from equilibrium”), was moved in order to minimize the resulting maximum value of m_j everywhere in the entire system, and the process repeated. Such an approach is akin to putting (randomly) $n + 1$ charged particles on an n -dimensional sphere and letting these particles come to their own equilibrium, except that the particles were constrained to lie on a set of several discrete points on this sphere, and (for computational expediency) only one point (that is, the point furthest from equilibrium) was allowed to move at each step. The force was always taken to be a repulsive force (opposite of the direction towards the particle creating the force). The model for the force magnitude was taken as $1/|\Delta x|^p$,

where Δx is the separation between the two points and p is an even positive integer. A fairly large value of p (e.g., 10) was found to work best, as using such a force model focused the algorithm on minimizing the “worst case” separation between particles, thereby resulting in a more uniform distribution.

The optimized minimal positive bases so determined (see the tables in the appendix) in fact may be used to generate entire *families* of bases by performing any of a large number of possible reflections (swapping + for – and – for + in one or more columns) and/or permutations (swapping one or more pairs of columns); such operations preserve the angles between the various vectors of the minimal positive bases so generated. Thus, we refer to the optimized minimal positive bases in the tables in the appendix as *stencils*, as they may be used to generate large families of optimized minimal positive bases. When performing a GPS, one may switch randomly (or according to some heuristic strategy) between different bases in this family at each poll step to prevent bias in any particular coordinate direction during the search.

A primary shortcoming of the cubic lattice in the optimization setting is its extremely low kissing number — it only has $2n$ nearest neighbors. Note that Charles & Dennis Jr. (2004) considered a strategy in which the positive basis was constructed not only from nearest-neighbor points on a cubic lattice, but also from several points that are farther away from the CMP, facilitating a larger range of choices for the vectors from which the positive basis could be constructed. The staggered (for $n = 4$) and checkerboard (for $n \geq 4$) lattices present a natural way to circumvent this difficulty, as these lattices have much higher kissing numbers, providing many more nearest neighbors from which a very large family of minimal positive bases may be constructed from the generating stencil. Thus, these lattices provide many more opportunities for distributing the vectors of a minimal positive basis in a desired fashion (according to any of a number of possible heuristic strategies) than the constraining set of nearest neighbors that the cubic lattice allows, without requiring the use of non-nearest neighbor test points during the poll step. This facilitates the efficient localization of the minimum point by the optimization algorithm to the smallest region possible on a given lattice before it is refined, thereby maximizing the effectiveness of each poll step.

III.6.3 Incorporating previous function evaluations

Note also that, if a function evaluation at one of the nearest neighbors to the CMP is already available before the poll step begins, the (randomized) basis (selected from the family described in the previous section) may be reflected such that one of the points in this basis is this

previously computed point, thereby reducing the number of new function evaluations required to complete the poll step to n .

In the case that generating a new positive basis is computationally less expensive than performing a function evaluation, if two or more function evaluations on nearest neighbor points are already available before the poll step begins, it is most expedient to generate a new positive basis from scratch, using the force-based method described above while fixing the particles corresponding to the points with the previous function evaluations and adjusting the remaining particles in order to minimize the maximum projected force over the entire set of particles, while selecting enough new points to insure that a positive basis is in fact generated using this technique, which may be verified *a posteriori*.

III.7 Comparison of the cubic, checkerboard, and staggered lattices for optimization in n dimensions

As we did in §4 for the case $n = 2$, and in §5 for the case $n = 3$, we now compare our three candidate lattices (cubic, staggered, and checkerboard) for n -dimensional optimization for cases with $4 \leq n \leq 15$ in terms of the four properties listed in §2 for GPS optimization. Note that cases with $n > 15$ may be studied following a similar procedure.

III.7.1 Enumeration

As already mentioned in the first paragraph of §6, the n -dimensional cubic, staggered, and checkerboard lattices are all straightforward to enumerate.

Constraints are applied by restricting the range of the component lattices in one or more directions. Note that an n -dimensional cubic lattice restricted in one coordinate direction forms an $(n - 1)$ -dimensional cubic lattice on the boundary, restricted in two coordinate directions forms an $(n - 2)$ -dimensional cubic lattice, etc. Similarly, an n -dimensional checkerboard lattice restricted in one coordinate direction forms an $(n - 1)$ -dimensional checkerboard lattice, restricted in two coordinate directions forms an $(n - 2)$ -dimensional checkerboard lattice, etc.

On the other hand, an n -dimensional staggered lattice restricted in one coordinate direction forms an $(n - 1)$ -dimensional cubic lattice on the boundary, and further restricted in two coordinate directions forms an $(n - 2)$ -dimensional cubic lattice, etc. Following the procedure described at the end of in §5.1, it is straightforward to augment the staggered lattice on the boundary in

order to retain the staggered nature of the packing (and its favorable properties) when one or more constraints are binding.

III.7.2 Refinement

It is readily seen from their enumerations in the first paragraph of §6 that, when refined by a factor of two, the nodal points on the original lattices are also part of the refined lattices.

n	lattice	Δ	Θ	G	τ	S
4	cubic ($\mathbf{Z}^4$)	0.308	4.934	0.08333	8	0.333 ⁽¹⁾
	checkerboard (D_4)	0.617 [†]	2.467	0.07660 [‡]	24 [†]	0.333
	$\cong$ staggered (D_4^*)					
5	cubic ($\mathbf{Z}^5$)	0.164	9.195	0.08333	10	0.295 ⁽¹⁾
	checkerboard (D_5)	0.465 [†]	4.598	0.07579	40 [†]	0.333
	staggered (D_5^*)	0.329	2.498	0.07563	10	0.617 ⁽⁶⁾
6	cubic ($\mathbf{Z}^6$)	0.0807	17.44	0.08333	12	0.268 ⁽¹⁾
	checkerboard (D_6)	0.323	8.721	0.07559	60	0.333
	staggered (D_6^*)	0.161	4.360	0.07512	12	0.552 ⁽⁷⁾
7	cubic ($\mathbf{Z}^7$)	0.0369	33.49	0.08333	14	0.247 ⁽¹⁾
	checkerboard (D_7)	0.209	16.75	0.07569	84	0.333
	staggered (D_7^*)	0.0738	4.569	0.07486	14	0 ⁽⁸⁾
8	cubic ($\mathbf{Z}^8$)	0.0159	64.94	0.08333	16	0.230 ⁽¹⁾
	checkerboard (D_8)	0.127	32.47	0.07591	112	0.333
	staggered (D_8^*)	0.0317	8.117	0.07474	16	0.333 ⁽⁹⁾
9	cubic ($\mathbf{Z}^9$)	6.44e-3	126.8	0.08333	18	0.216 ⁽¹⁾
	checkerboard (D_9)	0.0729	63.40	0.07620	144	0.333
	staggered (D_9^*)	0.0129	8.666	0.07469	18	0.309 ⁽¹⁰⁾
10	cubic ($\mathbf{Z}^{10}$)	2.49e-3	249.0	0.08333	20	0.205 ⁽¹⁾
	checkerboard (D_{10})	0.0398	124.5	0.07650	180	0.333
	staggered (D_{10}^*)	4.98e-3	15.56	0.07470	20	0.448 ⁽¹¹⁾
11	cubic ($\mathbf{Z}^{11}$)	9.20e-4	491.4	0.08333	22	0.195 ⁽¹⁾
	checkerboard (D_{11})	0.0208	245.7	0.07681	220	0.333
	staggered (D_{11}^*)	1.84e-3	16.81	0.07474	22	0 ⁽¹²⁾
12	cubic ($\mathbf{Z}^{12}$)	3.26e-4	973.4	0.08333	24	0.186 ⁽¹⁾
	checkerboard (D_{12})	0.0104	486.7	0.07710	264	0.333
	staggered (D_{12}^*)	6.52e-4	30.42	0.07480	24	0.216 ⁽¹³⁾

continued on next page

continued from previous page

n	lattice	Δ	Θ	G	τ	S
13	cubic ($\mathbf{Z}^{13}$)	1.11e-4	1935	0.08333	26	0.179 ⁽¹⁾
	checkerboard (D_{13})	5.03e-3	967.3	0.07737	312	0.333
	staggered (D_{13}^*)	2.22e-4	33.13	0.07487	26	0.207 ⁽¹⁴⁾
14	cubic ($\mathbf{Z}^{14}$)	3.66e-5	3856	0.08333	28	0.172 ⁽¹⁾
	checkerboard (D_{14})	2.34e-3	1928	0.07763	364	0.333
	staggered (D_{14}^*)	7.32e-5	60.24	0.07495	28	0.303 ⁽¹⁵⁾
15	cubic ($\mathbf{Z}^{15}$)	1.16e-5	7703	0.08333	30	0.166 ⁽¹⁾
	checkerboard (D_{15})	1.05e-3	3852	0.07788	420	0.333
	staggered (D_{15}^*)	2.33e-5	66.00	0.07504	30	0 ⁽¹⁶⁾

Table III.3: Characteristics of the n dimensional cubic, checkerboard, and staggered lattices for $4 \leq n \leq 15$. See Table III.1 legend for details.

III.7.3 Uniformity

All three metrics discussed in §4.3 and 5.3 to quantify the uniformity of the lattice (namely, Δ , Θ , and G) in the 2D and 3D cases may also be calculated in the n -dimensional case, as reported in Table III.3, though the formula for the calculation of G is fairly involved for the D_n^* lattice (Conway & Sloane, 1999). The staggered and checkerboard lattices are found to be superior to the cubic lattice by all three measures for all values of n in the table.

In particular, the checkerboard lattice has a covering thickness Θ which is a factor of two better than the cubic lattice for all n , whereas it has a packing density Δ which is a factor of two better for $n = 4$, growing to a full two orders of magnitude better for $n = 15$.

On the other hand, the staggered lattice has a packing density Δ which is a factor of two better than the cubic lattice for all n , whereas it has a covering thickness Θ which is a factor of two better for $n = 4$, growing to a full two orders of magnitude better for $n = 15$.

For $n > 4$, the staggered lattice has a slightly better mean squared quantization error per dimension, G , than the checkerboard lattice, both of which show a distinct improvement over the cubic lattice.

III.7.4 Neighborhood

As in the 2D and 3D cases, there are two ways to form a positive basis from nearby points on the n -dimensional cubic lattice, neither of which is ideal. One way is to select the $2n$ nearest neighbors to form a maximal positive basis. Though the angle between the basis vectors are uniform, this positive basis consists of $(2n) - (n + 1) = n - 1$ more points than it needs to, thus typically requiring an extra $n - 1$ function evaluations at each poll step. If a minimal positive basis (using $n + 1$ points) is desired, it is necessary to select at least one point which is not a nearest neighbor point. The angles between the basis vectors in this case are nonuniform, with 90° between any pair of n of the vectors and $\arccos(-1/\sqrt{n})$ between any of these vectors and the last, which is $\sqrt{n}$ times as far from the CMP as the others.

For $n > 4$, the staggered lattice has the same unfortunate properties, as it has the same nearest neighbors as the cubic lattice. However, the staggered lattice also has several more neighbors in a shell slightly outside these nearest neighbors. This shell of neighbors was used in the minimal positive bases reported in the second column of the appendix and in the skewness calculations reported in Table III.3.

On the other hand, the checkerboard lattice has a much higher kissing number. For this lattice, it was always found to be possible to construct a minimal positive basis (that is, using just $n + 1$ polling points) from nearest neighbor points alone.

As discussed in detail in §6 and tabulated in the appendix, a substantial investigation has been conducted to identify minimal positive bases for $4 \leq n \leq 15$ for both the staggered and checkerboard cases that are optimized based on their uniformity. The skewness of these optimized bases are reported in Table III.3. Remarkable, the positive bases in the 3D, 7D, 11D, and 15D staggered cases are uniformly distributed. The uniformity of the positive bases in the other cases are generally comparable to the cubic case.

As mentioned in the introduction, it might be desirable to select the positive basis from nearest neighbor points in order to effectively isolate the optimum of the function to the smallest region possible before moving to the refined lattice, thereby (potentially) leading to a reduced number of points over which the algorithm must search on the refined lattice. In order to develop a metric to quantify this concept, we may define r_{basis} as the radius of the sphere intersecting the minimal positive basis points, and $r_{\text{neighbors}}$ as the radius of the sphere intersecting the nearest-neighbor points. The ratio $r = r_{\text{basis}}/r_{\text{neighbors}}$ is depicted in Figure III.8 for several values of n for both the

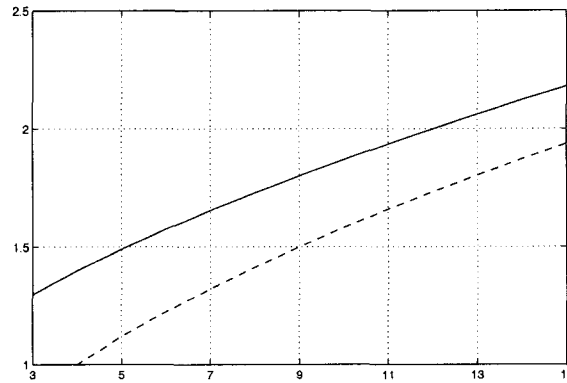


Figure III.8: Ratio $r = r_{\text{basis}}/r_{\text{neighbors}}$ of the radius r_{basis} of the sphere intersecting the cubic (solid) and staggered (dashed) minimal positive bases reported in Table III.3 and the radius $r_{\text{neighbors}}$ of the sphere intersecting the nearest-neighbor points (smaller is better); note that the sphere intersecting the cubic minimal positive basis is not centered at the CMP. For the checkerboard lattice, $r = 1$.

cubic case and the staggered case. (Note that ratio of the volumes of these n -dimensional spheres is given by r^n ; the ratio of the volumes is thus found to be quite large for large n .) A successful polling step might be thought of as isolating (approximately) the actual minimum point to lie within the sphere containing the polling points tested. The tighter isolation of the actual minimum point by a successful polling step on nearest neighbor points (as possible, e.g., using a minimal positive basis on the checkerboard lattice) thus might be preferred in the optimization setting to the looser isolation of the optimum point by a successful polling step on a minimal positive basis selected from non-nearest-neighbor points.

III.7.5 Summary of the n -dimensional case

In balance, in terms of the four properties identified in §2 for lattice-based optimization in n dimensions using GPS algorithms, the staggered and checkerboard lattices appear to be preferable to the cubic lattice. Note that we do not yet know which of the metrics reported is most strongly correlated with the convergence rate of the subsequent optimization calculations. However, given that the alternative lattices tested appear to be significantly better suited to the cubic lattice in terms of *any* particular metric of interest that quantifies the relevant properties (see §2), such alternative lattices appear to be quite promising for use in a variety of GPS algorithms.

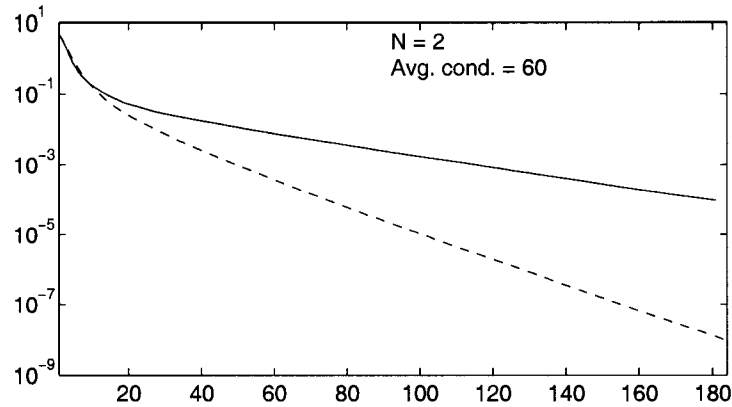


Figure III.9: Convergence of the GPS optimization strategy for $n = 2$, using the (solid) cubic and (dashed) hexagonal lattices, for a simple quadratic optimization problem. The cost function J is plotted as a function the cumulative number of function evaluations performed.

III.8 Convergence tests of GPS optimizations using the alternative lattices

We now perform several GPS optimizations in n dimensions (for $2 \leq n \leq 15$) using the cubic, hexagonal, BCC, FCC, checkerboard, and staggered lattices in order to compare their performance.

The problem that is optimized in these tests is both artificial and easy to solve by other means, but serves as a useful benchmark to quantify the relative performance of the GPS algorithm in various dimensions using various lattices. The problem we set out to solve is simply to minimize the quadratic function $f = (Ax - b)^2$ for several randomly-generated A and b with $|A| \neq 0$. The polling patterns used were the minimal positive bases listed in the Appendix, and were held constant during optimizations. The average performance from 1,000 tests in each case is presented, thereby providing well averaged statistics of the convergence rates. No “search” step (as used by the SMF) was used, so the tests reflect only the efficiency of the poll step. A simple “incomplete opportunistic poll procedure” was used: every time a reduced function value was found, the CMP was moved to this new point, a new poll step was initiated, and the first trial point in this new poll step was evaluated in the last direction moved. Once a poll step on an entire minimal positive basis completed successfully (without finding any reduced function values), the lattice was refined by a factor of 2, and the process repeated.

For each case, two different average condition numbers of $A^T A$ were tested in order to quantify the effect of condition number on convergence. This was done by taking A as an appropriate power of a randomly-generated nonsingular matrix. The results are shown in Figures III.9 - III.12, and speak for themselves. In the (more difficult) poorly conditions problems (that is, the right column of subfigures), the optimization schemes based on the checkerboard and staggered lattices are typically a factor of two or more faster to converge than the optimization scheme based on the standard cubic lattice.

Note that the GPS algorithm used in these tests is quite inefficient; there are many techniques that may be applied to significantly improve convergence, such as applying the SMF search step, reflecting/permutating the minimal positive basis used at each new poll step, performing lattice coarsening (when appropriate) in addition to lattice refinements, etc. However, the purpose of these tests was *not* to develop the most efficient GPS algorithm possible, but rather, simply, to indicate the gains possible by incorporation of the present lattices in a series of simple GPS optimizations. The tests performed are adequate for this purpose. Note also that, when the SMF search step is added to the GPS algorithm, different properties of the lattice (as listed in §2) might play a more dominant role, so strict decisions about the superiority of one mesh over another can not be made from these tests.

III.9 Concluding remarks

Various lattices have been investigated for derivative-free optimization schemes such as GPS algorithms. These lattices have been compared in terms of four generally desired properties for lattices underlying GPS optimizations (identified in §2), and have been characterized by five apparently relevant metrics. In all cases, the alternative lattices proposed exhibit distinct advantages over the standard cubic lattice, which is typically used as the default choice in GPS optimizations today. Further, numerical tests on a simple quadratic optimization problem provide solid numerical evidence for the potential advantages of the use of such alternative lattices in GPS optimization codes.

Various arguments may be made concerning which of the four properties identified, or the various metrics quantifying these properties, is “most significant” in the optimization setting. It is currently not known which argument is most valid. The present work thus raises as many questions as it answers. The problem of finding the best lattices to optimize Δ , Θ , G , and τ in

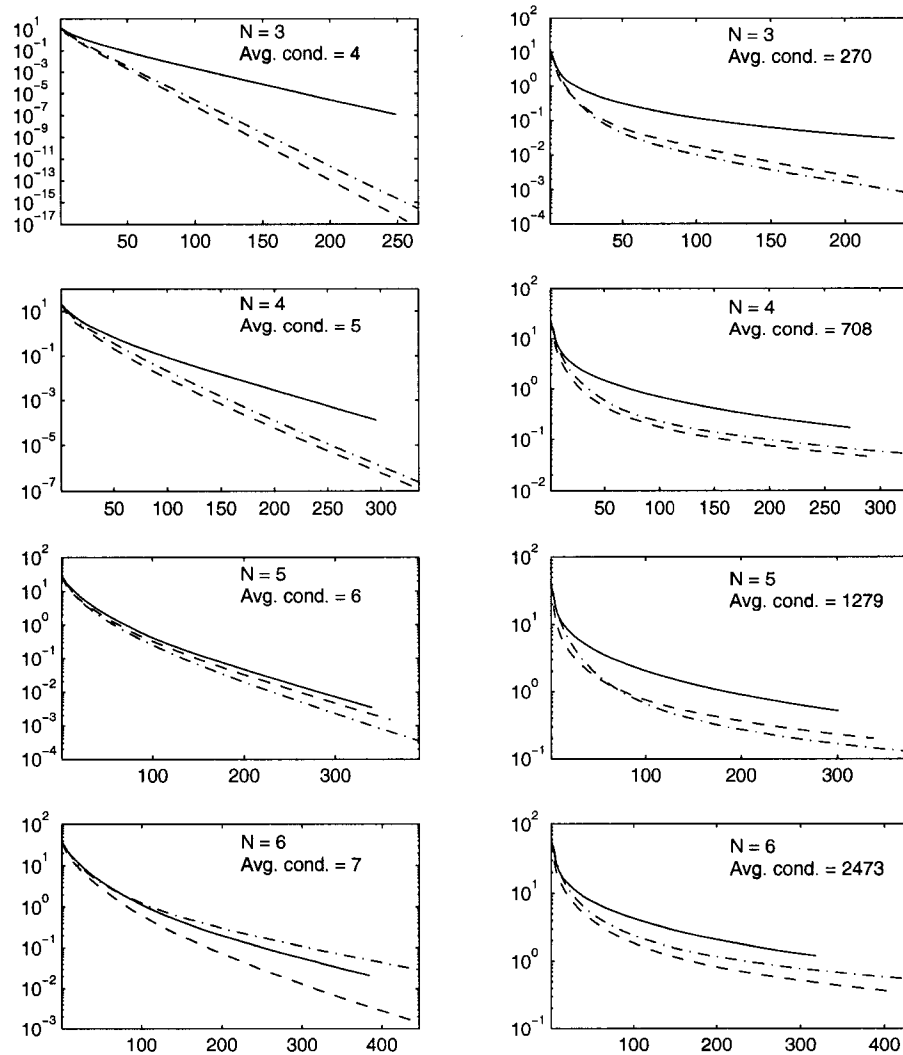


Figure III.10: Convergence of the GPS optimization strategy for $n = 3$ to $n = 6$, using the (solid) cubic, (dashed) staggered, and (dot-dashed) checkerboard lattices, for a simple quadratic optimization problem. The cost function J is plotted as a function the cumulative number of function evaluations performed.

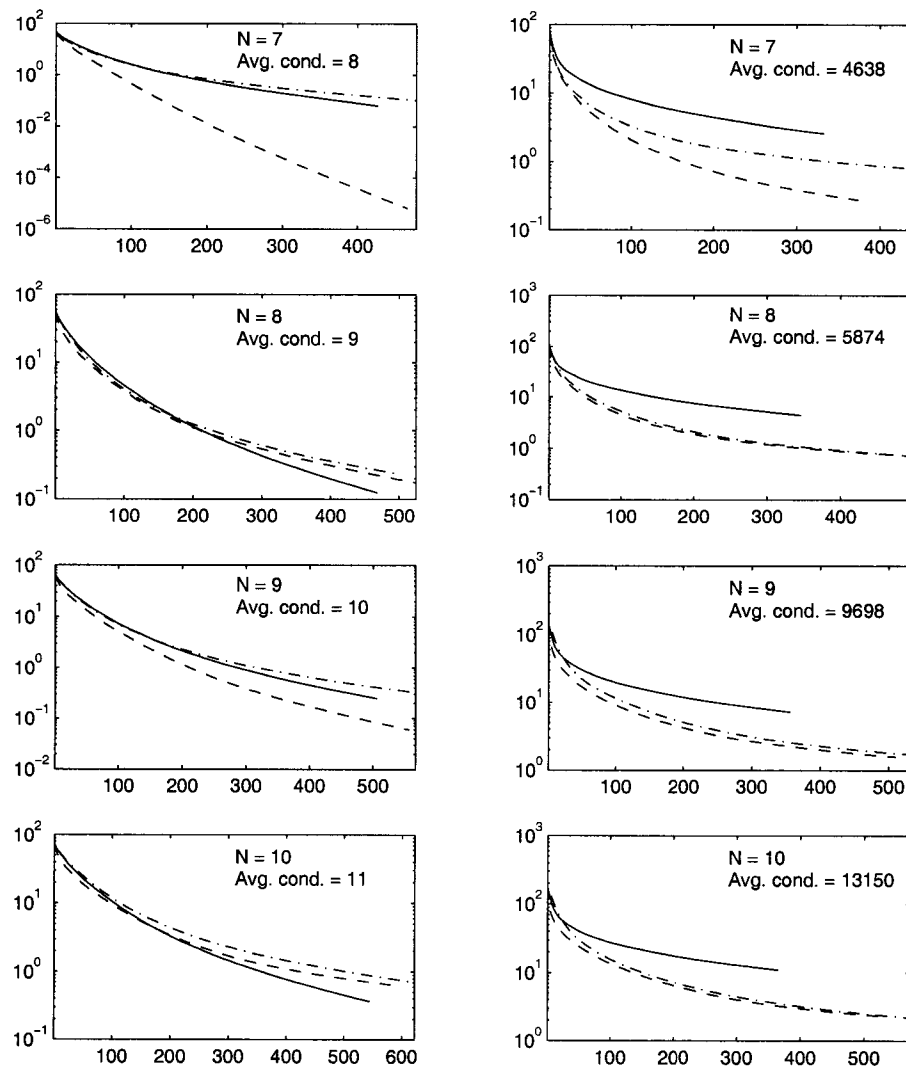


Figure III.11: Convergence of the GPS optimization strategy for $n = 7$ to $n = 10$. (See Figure III.10 for explanation.)

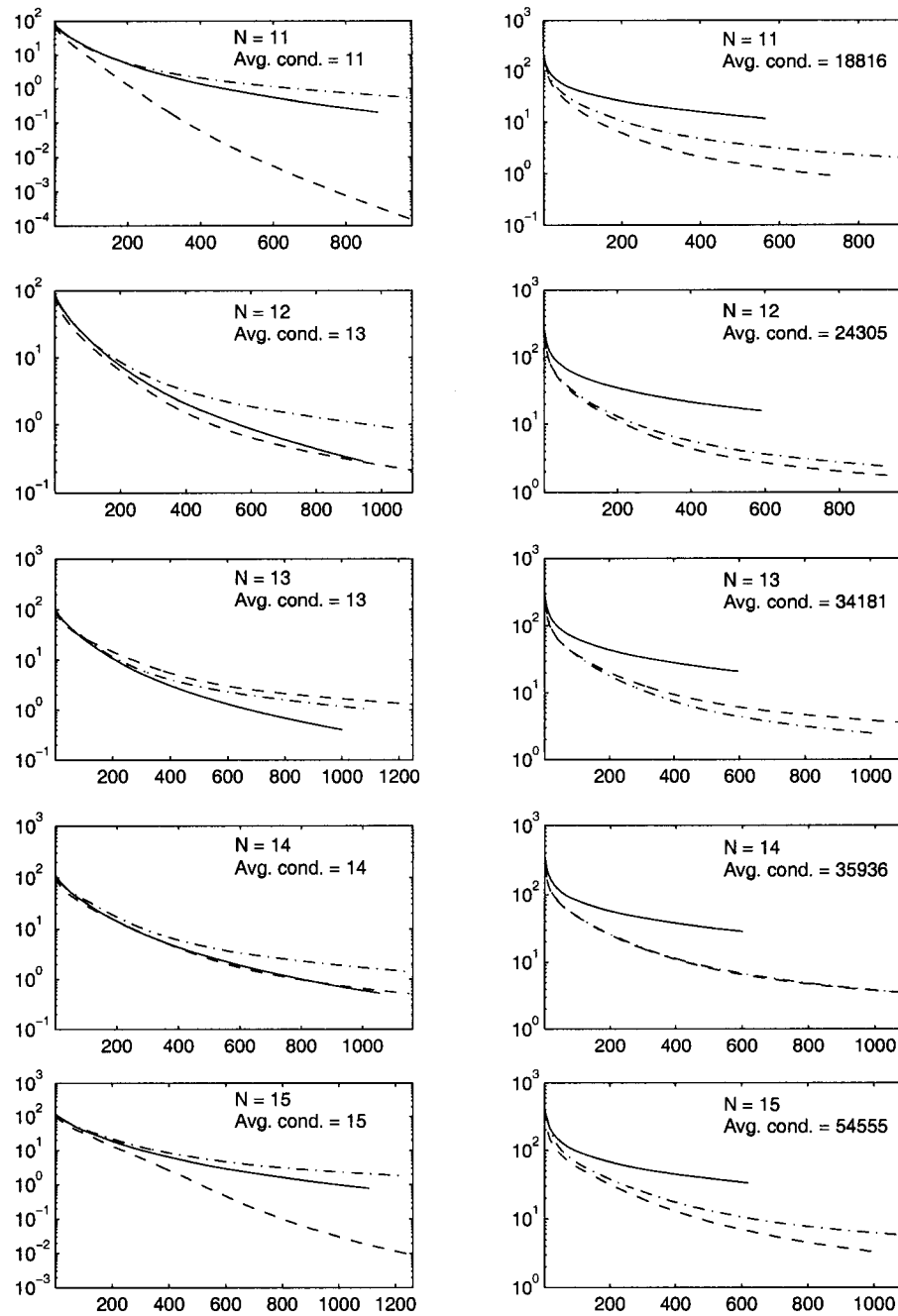


Figure III.12: Convergence of the GPS optimization strategy for $n = 11$ to $n = 15$. (See Figure III.10 for explanation.)

various dimensions has been the topic of intense scrutiny, as well summarized in Conway & Sloane (1999). Future work should explore these additional lattices in order to provide potentially greater gains in these four metrics, beyond that achieved by D_n and D_n^* . A fifth metric has been identified in this work, the skewness S of the optimized minimal positive basis on the lattice, which might also play a significant role in the selection of the best lattices for GPS optimization.

III.A Tables of optimized stencils of minimal positive bases

Table III.4 to III.6 on three following pages list the optimized stencils of minimal positive bases for $n = 3$ to 15. For explanation of both what these stencils mean and how they were obtained, see §6.

N	checkerboard (D_n) stencil										staggered (D_n^*) stencil																		
3	0	-	+	0																	-	+	-						
	0	-	+	0																	-	+	-						
	0	-	+	0																	-	+	-						
4	0	0	-	+	0																-	+	+	+					
	0	0	-	+	0																-	+	+	+					
	0	0	-	+	0																-	+	+	+					
5	0	0	0	-	+	0															-	+	+	+	-				
	0	0	0	-	+	0															-	+	+	+	-				
	0	0	0	-	+	0															-	+	+	+	-				
6	0	0	0	0	-	+	0														-	+	+	+	-	+			
	0	0	0	0	-	+	0														-	+	+	+	-	+			
	0	0	0	0	-	+	0														-	+	+	+	-	+			
7	0	0	0	0	0	-	+	0													-	+	+	+	-	+	+		
	0	0	0	0	0	-	+	0													-	+	+	+	-	+	+		
	0	0	0	0	0	-	+	0													-	+	+	+	-	+	+		
8	0	0	0	0	0	0	-	+	0												-	+	+	+	-	+	+	+	
	0	0	0	0	0	0	-	+	0												-	+	+	+	-	+	+	+	
	0	0	0	0	0	0	-	+	0												-	+	+	+	-	+	+	+	
9	0	0	0	0	0	0	0	-	+	0											-	+	+	+	-	+	+	+	+
	0	0	0	0	0	0	0	-	+	0											-	+	+	+	-	+	+	+	+
	0	0	0	0	0	0	0	-	+	0											-	+	+	+	-	+	+	+	+

Table III.4: Stencils of optimized minimal positive bases for $n = 3$ to 9.

[illegible]

Table III.5: Stencils of optimized minimal positive bases for $n = 10$ to 13.

References

- BOOKER, A., DENNIS, JR., J., FRANK, P., SERAFINI, D., TORCZON, V. & TROSSET, M. 1999 A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization* **17**, 1–13.
- BOX, M. J., DAVIES, D. & SWANN, W. H. 1969 *Non-Linear Optimization Techniques*. ICI Monograph No. 5, Oliver & Boyd, Edinburgh.
- CHARLES, A. & DENNIS JR., J. 2004 Mesh adaptive direct search algorithms for constrained optimization Submitted to SIAM J. Optim.
- CONWAY, J. & SLOANE, N. 1999 *Sphere Packings, Lattices, and Groups*. Springer-Verlag, third Edition.
- COOPE, I. D. & PRICE, C. J. 2001 On the convergence of grid-based methods for unconstrained optimization. *SIAM J. Optim.* **11** (4), 859–869.
- DAVIS, C. 1954a Theory of positive linear dependence. *American Journal of mathematics* **76** (4), 733–746.
- DAVIS, O. E. 1954b *The design and Analysis of Industrial Experiments*. Oliver and Boyd, Edinburgh.
- HILBERT, D. 1900 Mathematische probleme. *Göttinger Nachrichten* pp. 253–297.
- JANNER, A. 2001 Introduction to a general crystallography. *Acta Cryst* **A57**, 378–388.
- JANSSEN, T., BIRMAN, J., DÉNOYER, F., KOPTSIK, V., VERGER-GAUGRY, J., D., W., A., Y., ABRAHAMS, S. & KOPSKY, V. 2002 Report of a subcommittee on the nomenclature of n -dimensional crystallography. ii. symbols for arithmetic crystal classes, bravais classes and space groups. *Acta Cryst* **A58**, 605–621.
- JANSSEN, T., BIRMAN, J., KOPTSIK, V., SENECHAL, M., WEIGEL, D., YAMAMOTO, A., ABRAHAMS, S. & HAHN, T. 1999 Report of a subcommittee on the nomenclature of n -dimensional crystallography. i. symbols for point-group transformations, families, systems and geometric crystal classes. *Acta Cryst* **A55**, 761–782.
- MARSDEN, A. L., WANG, M., DENNIS, JR., J. E. & MOIN, P. 2004 Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering, special issue on "Surrogate Optimization"* In press.
- NAVROTSKY, A. 1994 *Physics and Chemistry of Earth Materials*. Cambridge.
- PRESS, W., FLANNERY, B., TEUKOLSKY, S. & VETTERLING, W. 1986 *Numerical Recipes*. Cambridge.
- SCHWARZENBERGER, R. 1980 *N-dimensional crystallography*. Pitman Publishing.
- TORCZON, V. 1997 On the convergence of pattern search algorithms. *SIAM J. Optim.* **7** (1), 1–25.
- ZADOR, P. 1982 Asymptotic quantization error of continuous signals and their quantization dimension. *IEEE Transactions on Information Theory* **28**, 139–149.

Appendix

Simulation of near-wall turbulence over a moving wall using the immersed boundary method

This chapter is taken from

Luo, H. & Bewley, T. R. 2003 Design, modeling, and optimization of tensegrity compliant surface for reduction of drag induced by the turbulent flow. *Smart structures and materials 2003: modeling, signal processing, and control (SPIE proceedings series)*, 3-6 March, San Diego, 5049-57.

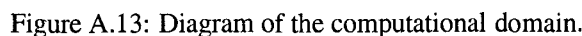
To accommodate the interface motion of the flow, we also investigated the potential use of the immersed boundary method in addition to the coordinate transformation method described in chapter I. Though this method is not accurate enough for relatively large wall deformations when applied in a pseudospectral flow simulation code, as used in the present work, it may still be useful for the purpose of linear stability analysis. Therefore, we discuss the method in this chapter.

A.1 Immersed boundary method

We use a Direct Numerical Simulation (DNS) code to model the incompressible flow in a channel. To accommodate the time-varying boundary, an immersed boundary method is used to avoid an expensive boundary-conforming grid reconstruction at each time step. Peskin (1977) first developed the immersed boundary method and applied it to biological systems such as the flow of blood in a heart. Several variations of this method have since been developed and applied to a variety of complex problems with time-varying geometries. The method is sometimes used in situations where there is a real “immersed” interface between two different fluids. For example, Sheth & Pozrikidis (1995) implemented a method with a pointwise body force distribution over the interface for solving the problem of the deformation of liquid drops in a shear flow. The method is also commonly used in situations in which there is no actual fluid on the other side of the boundary, but (for computational reasons) an artificial flow domain is defined so that the time-varying physical boundary of the fluid system essentially becomes “immersed”. For example, Goldstein, Handler & Sirovich (1993) presented a feedback scheme for the body force, and simulated the turbulent flow through a ribbed channel. Fadlun *et al.* (2000) applied a direct forcing scheme proposed by Mohd-Yusof (1997) to solve the flow problem inside an IC piston/cylinder assembly at high Reynolds number.

The basic idea of the immersed boundary method is that a time-invariant regular grid is used despite the boundary’s complexity. Flow fields on both sides of the “immersed” boundary are solved, even if one of these fields should be considered as artificial. Body forces are added within this artificial region to enforce the desired boundary conditions and dynamic motions of at the immersed interface.

In the present system, the flow is confined by the deformed and time-varying walls in a channel. We thus augment the physical flow domain, assuming there exists an artificial flow outside the channel walls with the same physical properties (mass, density) as the actual flow between the



A.2 DNS flow model

The flow, for the physical domain and the artificial domain alike, is governed by the incompressible Navier-Stokes equation

where p is the pressure divided by the density ρ and ν is the kinetic viscosity. The variables x_i are normalized by the half-width of the channel h , $\mathbf{u}$ is normalized by the mean friction velocity

$u_\tau \triangleq (\overline{\tau_w}/\rho)^{1/2}$, and t is normalized by h/u_τ . The Reynolds number based on the mean friction velocity and the half channel width is defined by $Re_\tau \triangleq u_\tau h/\nu$. $\mathbf{i}P_x$ is the time-varying but spatially-uniform pressure gradient in the x_1 direction, which is adjusted in such a way as to maintain constant mass flux in the physical domain at every time step.

Direct forcing $\mathbf{f}$ is applied in the virtual interface region such that the no-slip and no-penetration boundary conditions

$$\mathbf{u} = \dot{\boldsymbol{\eta}}, \quad (\text{A.3})$$

is satisfied at each time step, where $\boldsymbol{\eta}$ is the vertical motion of the wall. The horizontal motions are ignored since the wall deformation is very small.

Periodic boundary conditions are assumed in the streamwise (x_1) and spanwise (x_3) directions. The external boundaries are modeled with

$$u_2 = 0, \quad \frac{\partial u_3}{\partial x_2} = 0, \quad \text{on } \hat{\Gamma}^\pm; \quad \frac{\partial u_1}{\partial x_2} \Big|_{\hat{\Gamma}^\pm} = \text{constant} = \overline{\tau_w} \Big|_{\Gamma^\pm}. \quad (\text{A.4})$$

The “slip” condition is used to simplify the dynamics of the flow in the artificial region outside the immersed channel walls. These choices provide an approximately linear mean profile across the immersed interface, which improves accuracy in the numerical implementation.

A.3 Numerical scheme

The computational scheme is based on the numerical method adopted for the turbulent flow prediction in Bewley, Moin & Temam (2001). Details about the temporal discretization can be found in Akselvoll & Moin (1995). The scheme may be summarized as follows:

(1) A pseudospectral method is used for terms containing x_1 and x_3 derivatives, and a finite difference method is used for terms containing x_2 derivatives. A uniform, collocated grid is used in the x_1 and x_3 directions, and a stretched, staggered grid is used in the x_2 direction.

(2) A low-storage 3rd order Runge-Kutta scheme is used for the temporal evolution. The derivatives with respect to the homogeneous directions (x_1 and x_3) are treated explicitly in time, and the derivatives with respect to the inhomogeneous direction (x_2) are computed with the implicit Crank-Nicolson method in time.

Using to our “direct forcing” scheme, we do not calculate the external force $\mathbf{f}$ when solving the NS equation. Instead, we solve the equation with the no-slip/no-penetration constraint $g(\mathbf{u}) = 0$. In the discrete implementation, the positions of the interface are generally not coincident with the

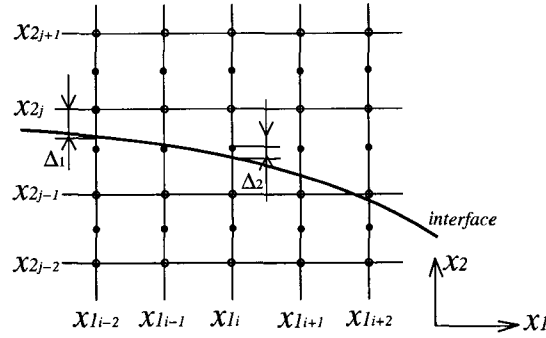


Figure A.14: Diagram of the deformed wall and the staggered grid.

grid points (see, e.g., Figure A.14), so the constraint $g(\mathbf{u}) = 0$ represents a numerical interpolation procedure to approximate the velocities at the immersed interface.

In the present work, we use an interpolation procedure based on Taylor series expansions. We use linear interpolation for the streamwise and spanwise velocity components and quadratic interpolation for the wall-normal velocity component. The velocities on the grid points that are close to the interface are constrained to satisfy the formulae

$$\begin{aligned}
 u_{1j-\frac{1}{2}} + \frac{u_{1j+\frac{1}{2}} - u_{1j-\frac{1}{2}}}{x_{2j+\frac{1}{2}} - x_{2j-\frac{1}{2}}} \Delta_2 &= 0, \\
 u_{2j} + \frac{u_{2j+1} - u_{2j-1}}{2(x_{2j+1} - x_{2j-1})} \Delta_1 + \frac{1}{2} \frac{u_{2j+1} - 2u_{2j} + u_{2j-1}}{(x_{2j+1} - x_{2j-1})^2} \Delta_1^2 &= \dot{\eta}, \\
 u_{3j-\frac{1}{2}} + \frac{u_{3j+\frac{1}{2}} - u_{3j-\frac{1}{2}}}{x_{2j+\frac{1}{2}} - x_{2j-\frac{1}{2}}} \Delta_2 &= 0,
 \end{aligned} \tag{A.5}$$

where Δ_1 and Δ_2 are illustrated in Figure A.14.

The external force $\mathbf{f}$ is directly evaluated by the NS equation.

$$\mathbf{f} = \frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{u}\mathbf{u}) + \nabla p - \nu \Delta \mathbf{u} + \mathbf{i}P_x. \tag{A.6}$$

Since the external force $\mathbf{f}$ is not divergence free, the term $\nabla \cdot \mathbf{f}$ is included when solving Poisson equation for the pressure.

A.4 Code validation

We first test our DNS code with the immersed boundary method on the canonical channel flow problem in which both immersed walls are stationary and not deformed. The Reynolds number is $Re_\tau = 100$ and $42 \times 64 \times 42$ Fourier modes are used (i.e., $64 \times 64 \times 64$ dealiased collocation grid

points). We compared the statistics with a simulation that does not have an immersed boundary and has been extensively validated by Bewley *et al.* (2001). Selected statistics are shown in Figure A.15. The correspondence of two simulations in the physical domain region is fairly good. In Figure A.15(b), we can see that the mean velocity profile is extended linearly into the two artificial regions, so the linear interpolation approximation for u_1 and u_3 at the interface regions is justified. The profile of the pressure fluctuations (not shown here) in the immersed boundary simulation shows there is a jump across the immersed interface, which implies that the interface provides something of a “barrier” between the real flow and the artificial flows.

The second test is the active wall motion control. The control scheme is that, based on measurements of the vertical velocity somewhere close to the wall, same amount of opposite control velocity as the measurements is applied at the wall. Choi *et al.* (1994) first investigated the scheme and obtained more than 20% drag reduction. The control actuation they used was unsteady blowing/suction. Inspired by this research, Endo, Kasagi & Suzuki (2000) employed vertical wall motion actuation based on the same kind of measurements. The simulations they did with $Re_\tau = 150$ and $\eta_{rms}^+ \approx 1$ showed that drag was reduced about 10%. Then Kang & Choi (2000) did the similar work with $Re_\tau = 140$ and $\eta_{max}^+ \leq 5$, and the drag was reduced up to 13% ~ 17%.

In our test, we prescribe the wall motion as follows

$$\frac{\partial \eta}{\partial t} = -\alpha u_2|_{x_2^+=15} - \beta \eta, \quad \alpha = \frac{\max\{u_2|_{x_2^+=15}\}}{V_{max}}, \quad (\text{A.7})$$

where V_{max} is a pre-defined constant. The formula means that the velocity of the wall is opposite to the vertical velocity component (u_2) 15 viscous unit away from the wall with the amount scaled by a factor α to reduce the control intensity. The second term is a damping term used by Endo *et al.*. The purpose is to slow down wall movement and reduce the deformation magnitude when the displacement is large.

Figure A.16 shows some statistics from the simulation for $Re_\tau = 100$ and $\eta_{rms}^+ \approx 0.03$. Time-averaged drag on the two immersed walls shows that drag on the lower wall is about 4.5% less than that on the upper wall, which is quite slight compared to what Endo and Kang have obtained. Note that the control we applied is very weak. Currently we are still tuning the code to accommodate stronger control actuation.

From the profile of the *RMS* of velocity fluctuations (Figure A.16(b)), we can see clearly that the streamwise velocity fluctuation intensity is lower at the bottom wall side compared the upper wall side. If we zoom in the region close to the lower wall (Figure A.16(c)), we see that the

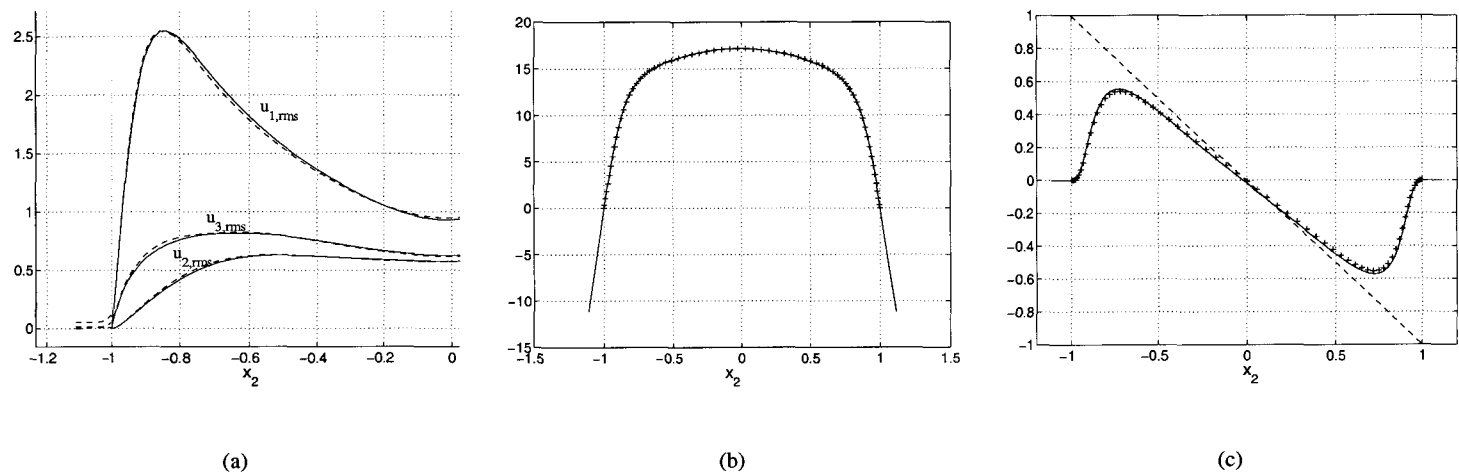


Figure A.15: Comparison of selected time-averaged statistics from simulations of the immersed boundary method and the regular domain. (a) Profiles of RMS velocity fluctuations; (b) mean velocity profiles; (c) profiles of Reynolds stress and total stress. Solid —, simulation from the immersed boundary method; dashed (---) in (a) and plus sign (+) in (b) and (c), simulation from the regular domain.

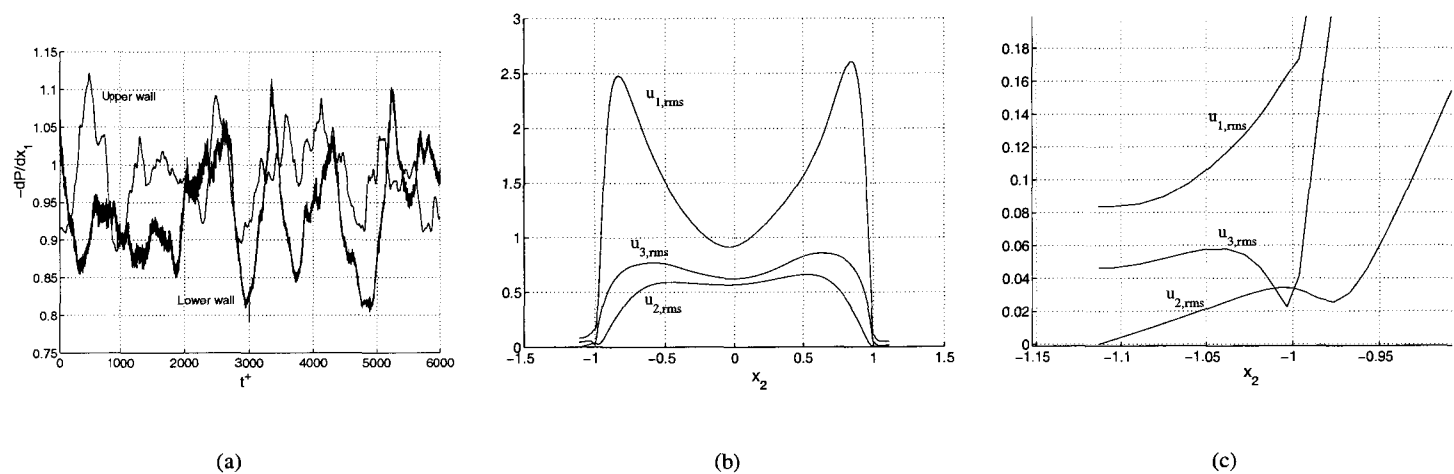


Figure A.16: Statistics for deforming wall opposition control using the immersed boundary method. (a) History of drag; (b) profiles of RMS velocity fluctuations; (c) zoom-in view of the lower wall region in (b).

minima of vertical velocity fluctuation is shifted away a little away from the wall. This is a typical feature of the opposition control which represents a virtual barrier above the wall preventing the high momentum flow from being transported to the wall, and thereby drag reduction is obtained.

References

- AKSELVOLL, K. & MOIN, P. 1995 Large eddy simulation of turbulent confined coannular jets and turbulent flow over a backward facing step. *Tech. Rep. TF-63*. Thermosciences Division, Dept. of Mech. Eng., Stanford University.
- BEWLEY, T. R., MOIN, P. & TEMAM, R. 2001 DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *J. Fluid Mech.* **447**, 179–225.
- CHOI, H., MOIN, P. & J., K. 1994 Active turbulence control for drag reduction in wall-bounded flows. *J. Fluid Mech.* **262**, 75–110.
- ENDO, T., KASAGI, N. & SUZUKI, Y. 2000 Feedback control of wall turbulence with wall deformation. *Int. J. Heat and Fluid Flow* **21**, 568–575.
- FADLUN, E. A., VERZICCO, R., ORLANDI, P. & MOHD-YUSOF, J. 2000 Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J. Comput. Phys.* **161**, 35–60.
- GOLDSTEIN, D., HANDLER, R. & SIROVICH, L. 1993 Modeling a no-slip flow boundary with an external force field. *J. Comput. Phys.* **105**, 354–366.
- KANG, S. & CHOI, H. 2000 Active wall motions for skin-friction drag reduction. *Phys. Fluids* **12** (12), 3301–3304.
- MOHD-YUSOF, J. 1997 Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries. *CTR Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford Univ.* pp. 317–327.
- PESKIN, C. S. 1977 Numerical analysis of blood flow in the heart. *J. Comput. Phys.* **25**, 220–252.
- SHETH, K. S. & POZRIKIDIS, C. 1995 Effects of inertia on the deformation of liquid drops in simple shear flow. *Computers & Fluids* **24** (2), 101–119.