2009

IMMERSED BOUNDARY CONDITIONS METHOD FOR COMPUTATIONAL FLUID DYNAMICS PROBLEMS

Syed Zahid Husain

IMMERSED BOUNDARY CONDITIONS METHOD FOR COMPUTATIONAL FLUID DYNAMICS PROBLEMS

(Spine title: IBC METHOD FOR FLOW PROBLEMS)

(Thesis format: Integrated-Article)

by

SYED ZAHID HUSAIN

Graduate Program
in
Engineering Science
Department of Mechanical and Materials Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

ABSTRACT

This dissertation presents implicit spectrally-accurate algorithms based on the concept of immersed boundary conditions (IBC) for solving a range of computational fluid dynamics (CFD) problems where the physical domains involve boundary irregularities. Both fixed and moving irregularities are considered with particular emphasis placed on the two-dimensional moving boundary problems. The physical model problems considered are comprised of the Laplace operator, the biharmonic operator and the Navier-Stokes equations, and thus cover the most commonly encountered types of operators in CFD analyses. The IBC algorithm uses a fixed and regular computational domain with flow domain immersed inside the computational domain. Boundary conditions along the edges of the time-dependent flow domain enter the algorithm in the form of internal constraints. Spectral spatial discretization for two-dimensional problems is based on Fourier expansions in the stream-wise direction and Chebyshev expansions in the normal-to-the-wall direction. Up to fourth-order implicit temporal discretization methods have been implemented. The IBC algorithm is shown to deliver the theoretically predicted accuracy in both time and space.

Construction of the boundary constraints in the IBC algorithm provides degrees of freedom in excess of that required to formulate a closed system of algebraic equations. The 'classical IBC formulation' works by retaining number boundary constraints that are just sufficient to form a closed system of equations. The use of additional boundary constraints leads to the 'over-determined formulation' of the IBC algorithm. Over-determined systems are explored in order to improve the accuracy of the IBC method and to expand its applicability to more extreme geometries. Standard direct over-determined solvers based on evaluation of pseudo-inverses of the complete coefficient matrices have been tested on three model problems, namely, the Laplace equation, the biharmonic equation and the Navier-Stokes equations. In all cases tested the over-determined

formulations based on standard solvers were found to improve the accuracy and the range of applicability of the IBC method.

Efficient linear solvers suitable for the spectral implementation of the IBC method have been developed and tested in the context of two-dimensional steady and unsteady Stokes flow in the presence of fixed boundary irregularities. These solvers can work with the classical as well as the over-determined formulations of the method. Significant acceleration of the computations as well as significant reduction of the memory requirements have been accomplished by taking advantage of the structure of the coefficient matrix resulting from the implementation of the IBC algorithm. Performances of the new solvers have been compared with the standard direct solvers and are shown to be of up to two orders of magnitude better. It has been determined that the new methods are by at least an order of magnitude faster than the iterative methods while removing restrictions based on the convergence criteria and thus expanding the severity of the geometries that can be dealt with using the IBC algorithm. The performance of the IBC method combined with the new solvers has been compared with the performance of a method based on the generation of the boundary conforming grids, and is found to be better by at least two orders of magnitude. Application of the new solvers to the unsteady problems also results in performance improvement of up to two orders of magnitude. The specialized solvers applied to the over-determined formulation is shown to be at least two orders of magnitude faster than their standard counterparts while capable of extending the range of applicability of the IBC algorithm by 50%-70% for the Stokes flow problem. The concept of the specialized solvers has been extended to solve two-dimensional moving boundary problems described by the Navier-Stokes equations, where the new solver has been shown to result in a significant acceleration of computations as well as substantial reduction in memory requirements.

The conceptual aspects of extending the IBC algorithm for solving three-dimensional problems have been presented using the vorticity-velocity formulation of the three-dimensional Navier-Stokes equations. Test results on the implementation of the IBC algorithm for three-dimensional problems are discussed in the context of heat diffusion

problems in the presence of fixed as well as moving boundaries. The algorithm is shown to be spectrally-accurate in space and capable of delivering theoretically predicted accuracy in time for the different test problems. Given a potentially large size of the resultant linear algebraic system, various methods that take advantage of the special structure of the coefficient matrix have been explored in search for an efficient solver, including two versions of the specialized direct solver as well as serial and parallel iterative solvers. Both versions of the specialized direct solver have been shown to be more computationally efficient than the other solution methods.

Possible applications of the IBC algorithm for analyzing physical problems have also been presented. The advantage of using IBC algorithm is illustrated by considering its application to two physical problems, which are – i) analysis of the effects of distributed roughness on friction factor and ii) analysis of traveling wave instability in wavy channels. These examples clearly show the attractiveness of the IBC algorithm for studying effects of a large array of boundary geometries on the flow field.

Keywords: Computational fluid dynamics (CFD), spectral method, immersed boundary conditions (IBC) method, distributed roughness, moving boundary problem, implicit method, Fourier expansion, Chebyshev polynomial, domain transformation (DT) method, direct numerical simulation (DNS).

CO-AUTHORSHIP

The following dissertation is presented in the integrated-article format. Chapters 2 through 6 are based on manuscripts that have been previously published, or submitted, or finalized for submission. I, Syed Zahid Husain, am the first author of all these manuscripts with Prof. J. M. Floryan as a co-author. The manuscript that forms the basis of Chapter 4 has Dr. J. Szumbarski as an additional co-author.

The formulations of the algorithms, computer programming and computations of results presented in Chapters 2 to 6 were performed by me. Mr. D.C. Del Rey Fernandez is responsible for the formulations of the algorithm, computer programming and computations of results associated with the discussions presented in Section 7.2.2 I have assisted Prof. J. M. Floryan in supervising Mr. Fernandez at the different stages of his Masters research including formulation and implementation of the algorithm along with presentation and discussion on the various aspects of the obtained results. I have also been a major contributor in writing the report, which forms the basis of Section 7.2.2.

DEDICATION

To our departed son Zayan Syed Husain.

ACKNOWLEDGEMENT

I would like to start by expressing my sincere gratitude to my supervisor Prof. J. M. Floryan for the tremendous support and advice I have received from him throughout the tenure of my doctoral research. I will remain indebted to Prof. Floryan for his precious time, patience and encouragement. The completion of this dissertation would not have been possible without his incomparable assistance and invaluable guidance.

I would like to thank the members of my advisory committee, Prof. A. G. Straatman and Prof. C. Zhang, for their valuable suggestions and encouragement.

I would also like to thank all of my colleagues (past and present), Dr. Stefan Krol, Dr. Yan Jiang, Dr. J. Szumbarski, Mathew Fotia, Mohammad Zakir Hossain, Aidin Keikhaee, David César Del Rey Fernandez, German Kalugin, Alireza Mohammadi, Mohammad Fazel Bakhsheshi and Hadi Vafadar Moradi for their friendship, cooperation and helping attitude. I am particularly grateful to Dr. J. Szumbarski, Mohammad Zakir Hossain, Alireza Mohammadi, Aidin Keikhaee and David Fernandez for the insightful scientific discussions I had with them that have been tremendously helpful for me in achieving my research objectives.

Words are not enough to express my gratitude towards my family. Without the unconditional love, sacrifice and support of my wife, Zinat Farah Naz, I would not have been able to accomplish my goals. The encouragements from my parents have always kept me together especially during the difficult times.

The research works presented in this dissertation were financially supported by NSERC of Canada and SHARCNET.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

NOMENCLATURE

COMMON NOMENCLATURE FOR ALL CHAPTERS

α	:	Wave number of geometric irregularities in the x-direction
ω	:	Frequency of the elastic standing wave
ξ	:	The horizontal coordinate in the transformed computational domain for the domain transformation (DT) method
η	:	The vertical coordinate in the transformed computational domain for the DT method
Γ	:	Constant of coordinate transformation for the IBC method
θ	:	Temperature
Ψ	:	Stream function
Ψ_1	:	Modification of stream function in the presence of boundary irregularities
Δt	:	Size of time-step used in different temporal discretization schemes
A_L, A_U	:	Coefficients of Fourier expansion describing the lower and upper walls in the computational domain for the IBC method
c	:	Phase speed of the elastic traveling wave
$(d_L)_k$, $(d_U)_k$	:	Coefficients of Fourier expansion of the first derivative of Chebyshev polynomial of order k along the lower and upper walls
H_L, H_U	:	Coefficients of Fourier expansion describing the lower and upper walls in the physical domain
M_M	:	Number of Fourier modes retained in the discretized boundary constraints for the over-determined formulations of the IBC algorithm
N_M	:	Number of Fourier modes used to discretize dependent variable in the field equation for the different model problems in the x-direction
N_T	:	Order of Chebyshev polynomial used to discretize the modal functions in the y-direction

N_A	:	Number of Fourier modes used to define the boundary irregularities
Q	:	Volume flux
S	:	Amplitude of wall corrugations
T_k	:	Chebyshev polynomial of order k
u_1, v_1, p_1	:	Velocity and pressure modifications in the presence of boundary irregularities
u_0, v_0, p_0	:	Velocity and pressure components associated with the reference plane Poiseuille flow
$\mathbf{V}$	:	Total velocity vector
u, v, p	:	Total velocity and pressure components
$(w_L)_k$, $(w_U)_k$	:	Coefficients of Fourier expansion of Chebyshev polynomial of order k along the lower and upper walls
x	:	The horizontal coordinate in the physical domain in the fixed frame of reference
X	:	The horizontal coordinate in the moving frame of reference
y	:	The wall-normal coordinate in the physical domain
$\hat{y}$	:	The wall-normal coordinate in the transformed computational domain for the IBC method
y_L, y_U	:	Shape of the lower and upper walls in the physical domain
$\hat{y}_L$, $\hat{y}_U$	:	Shape of the lower and upper walls in the computational domain for the IBC method
Y_L	:	Upper extremity of the upper wall above y=+1 in the physical domain
Y_U	:	Lower extremity of the lower wall below y=-1 in the physical domain

NOMENCLATURE SPECIFIC TO CHAPTER 2

$\|\theta_U(X)\|_\infty$	:	L_∞ norm of error in the enforcement of boundary conditions for the steady problem solved in the moving frame of reference (X,y)
$\|\theta_U(x,t)\|_\infty$	:	L_∞ norm of error in the enforcement of boundary conditions for the unsteady problems solved in the fixed frame of reference (x,y)

NOMENCLATURE SPECIFIC TO CHAPTER 3

$\|u_{er}(X)\|_\infty$, $\|v_{er}(X)\|_\infty$	:	L_∞ norms of error in the enforcement of u- and v-velocity boundary conditions for the steady problem solved in the moving frame of reference (X,y)
$\|u_{er}(x,t)\|_\infty$, $\|v_{er}(x,t)\|_\infty$	:	L_∞ norms of error in the enforcement of u and v-velocity boundary conditions for the unsteady problems solved in the fixed frame of reference (x,y)

NOMENCLATURE SPECIFIC TO CHAPTER 4

$\|\theta_{er}\|_\infty$	:	L_∞ norm of error in θ over the complete solution domain as compared against a machine accurate reference solution evaluated using the domain transformation (DT) method
$\|\theta_{er,bc}\|_\infty$	:	L_∞ norm of error in the enforcement of boundary conditions for θ
$\|u_{er}\|_\infty$, $\|v_{er}\|_\infty$	:	L_∞ norms of error in u- and v-velocities over the complete solution domain as compared against machine accurate reference solution evaluated using the DT method
$\|u_{er,bc}\|_\infty$, $\|v_{er,BC}\|_\infty$	:	L_∞ norms of error in the boundary conditions for the u- and v-velocities
w_t	:	Weight factor
h	:	Rank of the over-determined coefficient matrix

NOMENCLATURE SPECIFIC TO CHAPTER 5

$\|u_{er}\|_\infty$, $\|v_{er}\|_\infty$	:	L_∞ norms of error in u- and v-velocities over the complete solution domain as compared against machine accurate reference solution evaluated using the DT method
$\|u_{er,BC}\|_\infty$, $\|v_{er,BC}\|_\infty$	:	L_∞ norms of error in the boundary conditions for the u- and v-velocities

NOMENCLATURE SPECIFIC TO CHAPTER 6

$\|u_{er}(X)\|_\infty$, $\|v_{er}(X)\|_\infty$	:	L_∞ norms of error in u- and v-velocities over the complete solution domain for the steady problems solved in the moving frame of reference as compared against machine accurate reference solution evaluated using the DT method
$\|u_{er,BC}(X)\|_\infty$, $\|v_{er,BC}(X)\|_\infty$	:	L_∞ norms of error in the boundary conditions for the u- and v-velocities for the steady problems solved in the moving frame of reference
$\|u_{er,BC}(x,t)\|_\infty$, $\|v_{er,BC}(x,t)\|_\infty$	:	L_∞ norms of error in the boundary conditions for the u- and v-velocities for the unsteady problems solved in the fixed frame of reference

NOMENCLATURE SPECIFIC TO CHAPTER 7

δ	:	Stream-wise wave number of the disturbance field
μ	:	Span-wise wave number of the disturbance field
$\boldsymbol{\Omega}$	:	The total vorticity vector
$\boldsymbol{\Omega}_2$	:	The vorticity vector corresponding to the basic state
$\boldsymbol{\Omega}_3$	:	The disturbance vorticity vector
σ	:	Complex eigen value in the linear stability equations (7.3.2.11)
σ_i	:	Amplification rate of disturbance
σ_r	:	Frequency of disturbance
f	:	Friction factor
N_D	:	Number of Fourier modes used to define the amplitudes of the disturbance velocity field
u_2, v_2, p_2	:	Velocity and pressure components associated with the basic state
u_3, v_3	:	Disturbance velocity components

ABBREVIATIONS

CFD	:	Computational fluid dynamics

IBC	:	Immersed boundary conditions
DT	:	Domain transformation
DNS	:	Direct numerical simulation
SVD	:	Singular value decomposition

CHAPTER 1

Introduction

1.1. Objective

The dissertation develops efficient, spectrally-accurate, temporally implicit gridless algorithms that can be applied to various heat and fluid flow problems involving either fixed or moving boundary irregularities.

1.2. Motivations

Boundary irregularities are encountered in many practical engineering problems pertinent to flow of heat and fluids and are well known to affect the flow field. In general, boundary irregularities can be classified into two different categories, namely, fixed and moving. A regular two-dimensional physical domain that is devoid of any boundary irregularity assumes the shape of a rectangle when investigated in a Cartesian frame of reference while in a polar frame of reference the regular physical domain takes the form of a circle. Therefore, for three-dimensional problems the shape of the regular physical domain in the Cartesian coordinates would resemble a rectangular box with flat surfaces on all sides and in the three-dimensional cylindrical coordinates would resemble a plain cylinder. Flow problems involving physical domains with stationary irregularities are termed as the 'fixed boundary problems'. On the other hand, flow problems where the form of the boundary irregularities vary as a function of time with respect to the laboratory frame of reference in a known and well prescribed manner, are referred to as the 'moving boundary problems'. Unlike the 'free boundary problems', where the boundaries of the flow domain change shape in response to the evolution of flow, the

moving boundary problems allow accurate identification of spatial and temporal locations of the flow boundaries *a priori*.

We are particularly interested in investigating boundary irregularities that have the form of roughness, where the mean heights of the irregularities are very small compared to the characteristic length of the flow domain. The individual roughness elements can have different shapes and may be apart from each other by large enough distance forming isolated roughness elements. Irregularities can also be spread all along the walls of physical domain of interest, which are termed as distributed roughness. Corrugated boundaries, therefore, also belong to the class of distributed roughness. From a physical perspective, our research is focused on the analysis of the effects of distributed roughness.

Among various effects, it is well known that surface roughness exerts crucial influence on heat transfer rate [1], skin friction drag [1, 2] and transition of shear layers from laminar to turbulent states [3, 4]. One of the principal motivations for the research work has been the interest to examine the influence that fixed surface roughness can exert on the hydrodynamic stability of shear layers in channel flows. Such studies allow devising passive flow control strategies in order to accelerate or decelerate the laminar-turbulent transition process through the design of properly structured roughness [5]. In this regard, one first needs to compute the modified mean flow in the presence of stationary roughness by solving the steady Navier-Stokes equations where high spatial accuracy is warranted. The next step is to forecast the behavior of the mean flow when it is subjected to small perturbations using linear stability analysis, which predicts the growth rate of the disturbances. In addition to high accuracy, the computational efficiency of the mean flow solver is a substantial issue as a comprehensive hydrodynamic stability analysis requires investigation of a vast array of geometric shapes. Algorithms that permit simple and convenient modeling of the boundary geometries are tremendously advantageous in such scenarios and, therefore, developing an efficient algorithm has been an important objective for the present research. Finally, in order to study the evolution of instabilities towards the final saturation state a tool needs to be developed to perform direct numerical

simulation (DNS) where once again the importance of the efficiency of the DNS solver becomes as crucial as the accuracy it can deliver [6].

Flow problems with moving surface roughness also possess significant importance from physical perspective especially in designing active flow control strategies [5]. Highly accurate and efficient algorithms that are capable of solving such flows can be used as a means to design active flow control systems where the shape of the roughness is adjusted dynamically in response to flow evolution. Besides this, researchers from various other fields have been attracted to efficient algorithms for moving boundary problems. One of the major motivations for the present research has been to develop an algorithm capable of analyzing peristaltic and pulsatile flows that are encountered in the field of biomedical engineering when one is interested in studying flows through the esophagus and vasculatures, respectively. Any algorithm intended to examine such flows require the capabilities of handling flow involving deforming boundaries with high accuracy and also in an efficient manner. This explains our motivation for developing an algorithm that can efficiently handle moving boundary problems and deliver high spatial and temporal accuracy.

1.3. Fixed boundary problems

We shall first focus on flow problems involving boundary irregularities that remain stationary with respect to the laboratory frame of reference. In general, the numerical solution process works by approximating the governing partial differential equations describing the physical flow with a set of algebraic equations through discretization and the presence of boundary irregularities imposes a major challenge for any algorithm employed for discretization. Over the years, researchers have developed numerous algorithms to deal with such challenges. When the system of discretized equations is represented in the matrix form, the corresponding matrix of the coefficients of the unknowns is referred to as the coefficient matrix. In most algorithms, the algebraic equations are derived by discretizing the field equations into a number of finite points or

elements or elemental volumes, thereby making the resultant discretization grid-dependent. Methods based on finite-elements, finite-difference and finite-volumes are examples of grid-dependent discretization procedures. On the other hand, there are gridless methods where the discretization is not performed on any particular grid points or elements and generally work by assuming global solutions based on expansion of polynomials. For example, Galerkin-projection-based spectral method discretizes field equations in a gridless manner.

1.3.1. Review of the available algorithms for fixed boundary problems

Most of the conventional algorithms for solving flow problems involving fixed irregularities are based on the generation of boundary conforming grids that uses standard finite-element, finite-volume or finite-difference schemes [1, 7]. Here one of the grid lines coincides with each of the boundaries of the physical domain. These standard methods suffer from a common limitation of lower spatial accuracy mainly attributable to low-order of discretization schemes that are typically used for these methods. Higher absolute accuracy can nevertheless be achieved using very fine grids but results in unreasonable computational overhead while the use of higher-order schemes generally necessitate substantial increase in efforts associated with formulations, grid construction and programming implementations [8]. Moreover, computational efficiencies of all these methods drop sharply when assigned to investigate a large array of boundary geometries particularly due to substantial cost of generating the coefficient matrices for the different boundary shapes.

Some other methods are based either on numerical or analytical mapping of the physical domain. Domain transformation (DT) method relies on analytical mapping of the irregular physical domain into a regular computational domain allowing exact enforcement of flow boundary conditions. DT method coupled with spectral discretization of the transformed spatial coordinates is capable of delivering high spatial accuracy [9, 10, 11]. However, analytical mapping contributes to substantial

complication in the transformed field equations resulting in significant cost for generating the coefficient matrix. Therefore, spectral implementation of the DT method is recommendable for solving flow problems with high degree of irregularities only [10, 11].

Domain perturbation (DP) method offers an alternative simple approach for handling boundary irregularities. This method uses a regular computational domain and instead of enforcing the boundary conditions along the edges of the physical domain the method works by transferring the boundary conditions to a certain mean location [10, 11, 12]. However, the applicability of DP method is limited to problems with very small amplitudes of roughness [11]. For flow over a rough leading edge it has been found that the first-order DP method provides reasonable accuracy only for situations where flow modifications induced by the boundary irregularities can be approximated by a linear theory [13].

A novel approach for solving flow problems involving boundary irregularities was proposed by Peskin in 1982 [14]. The method was based on the concept of immersed boundaries. The idea was implemented for studying flow behavior around cardiac valves and is generally referred to as the 'immersed boundary (IB) method'. Similar to the DT and DP methods, the IB method also uses regular computational domain, however, the edges of the computational domain extends beyond the edges of the physical domain. Therefore, the computational and physical boundaries may not necessarily coincide. Unlike the DP method, the physical boundary conditions in the IB method are enforced along the edges of the physical domain that are immersed inside the extended computational domain. Since the first work of Peskin, a number of variants of the immersed boundary methods have been developed by various researchers and are reviewed in [15, 16]. The original IB method as proposed by Peskin in the context of cardiac mechanics problems has later turned into a popular tool for handling problems involving fluid-structure interactions [16]. The physical boundary conditions on the immersed boundaries are imposed using additional forcing and the nature of this forcing is the fundamental distinguishing feature of the different variants of the IB method [15].

The forcing, in general, may be either continuous or discrete [15]. In the continuous forcing approach, the continuous governing equation is first modified by using a source term or forcing function in order to account for the no-penetration and no-slip conditions at the boundaries. The modified governing equation is subsequently discretized on a Cartesian grid which implies that the general formulation involved is independent of discretization technique [15]. In the case of the discrete forcing approach, the governing equation is first discretized and then only the cells adjacent to the immersed boundaries are adjusted to account for the physical boundary conditions. The discretization method, therefore, plays an important role for the discrete forcing approach [15]. The elimination of the cost of generating body conforming grids make the IB methods computationally very efficient as compared to the methods that rely on constructing body conforming grids discussed earlier. However, most of the IB methods are based on the low-order finite-difference, finite-volume or finite-element techniques [15] and as a result suffer from a common issue of low spatial accuracy. Moreover, the very physical concept of local forcing along the immersed boundaries used to enforce the no-slip and no-penetration conditions adds another level of uncertainty in these algorithms. The forcing are capable of affecting the flow physics locally around the boundaries, which although is difficult to quantify may lead to inaccurate representation of local wall shear as well as other derivatives of the flow quantities. Any procedure that even locally changes the flow physics may be unable to predict the second derivative of mean flow with sufficient accuracy. This can lead to significant inaccuracy in the study of stability of shear layers which are strongly affected by the second derivative of the mean flow. Aiming for higher spatial accuracy may lead to even further deterioration of this problem. This particular aspect of uncertainty associated with the IB methods have not been studied thoroughly and requires systematic investigation.

1.3.2. The immersed boundary conditions (IBC) method

One of the common limitations for most of the methods discussed so far is low spatial accuracy resulting from various low-order discretization techniques, which these

methods are mostly based upon. Spectral methods are well known to provide the highest accuracy for spatial discretization of the field equations and, therefore, have become the preferred approach particularly for direct numerical simulations [7]. However, most of the standard implementations of the spectral method are restricted to regular geometries, i.e., physical domains involving straight walls. Szumbarski and Floryan [17] developed a novel algorithm for solving two-dimensional steady-state fixed boundary problems that combined the inherent efficiency of the immersed boundary concept with the high accuracy of spectral discretization. This method is referred to as the immersed boundary conditions (IBC) method in the present work. The discretization of the field equation in the stream-wise direction where the flow is assumed to be periodic is implemented using Fourier expansions, which reduces the governing partial differential equation into a set of ordinary differential equations. Discretization in the normal-to-the-wall direction is performed using expansions based on Chebyshev polynomials. Finally, a set of algebraic equations in terms of the unknown coefficients of the Chebyshev polynomials are derived using the method of Galerkin projection [18]. Boundary geometries in the IBC algorithm are assumed to be periodic and are modeled using Fourier expansions. Although it limits the applicability of the algorithm only to boundary shapes that are expressible using Fourier expansions the resultant discretization process is gridless as all possible variations of boundary geometries can be achieved by changing the Fourier coefficients only. Such modeling of the geometries also substantially simplifies programming numerous geometries, which are essentially defined by a set of Fourier coefficients only. Unlike the IB methods discussed in the previous section, the IBC method does not depend on any fictitious forcing to impose the physical boundary conditions, rather transforms the original boundary value problem into an internal value problem. The discretized boundary conditions, therefore, enter the algorithm in the form of internal constraints. However, construction of the boundary constraints in the IBC method provides degrees of freedom in excess of that required to formulate a closed system of algebraic equations. The standard implementation of the IBC method as presented in [17], accommodates the boundary constraints in a Tau like manner [18] resulting in a closed system of equations and is referred to as the 'classical IBC formulation'. The first unsteady implementation of the IBC algorithm was done by

Husain and Floryan [9] in the context of a simpler two-dimensional conductive heat flow problem involving fixed boundaries. The superiority in computational efficiency of the IBC method over the spectral implementation of the DT method is evident from the results presented in [9].

1.4. Moving boundary problems

1.4.1. Review of the available algorithms for moving boundary problems

Researchers have developed a variety of algorithms over the years in order to deal with the moving boundary problems. The different algorithms as a whole can be classified into two categories, Lagrangian and Eulerian [19, 20]. Combinations of Lagrangian and Eulerian schemes, termed as the mixed methods [19], have also been investigated.

Lagrangian methods are well suited for moving boundary problems [19, 20]. The coordinate systems in these methods move along with the fluid, which implies that the individual computational cells always contain the same fluid elements. The so called Lagrangian velocities are functions of time and initial positions of all elements of the fluid body. Lagrangian methods permit sharp identification and simple delineation of the moving boundaries thereby allowing precise enforcement of the boundary conditions. However, Lagrangian methods are not suitable for problems where there is a possibility of significant mesh distortion [19, 20]. For such problems the Lagrangian algorithms suffer from problems of mesh tangling and numerical instabilities related to highly distorted meshes that consequently results in loss of numerical accuracy [19, 20]. Possible remedial measures to deal with mesh distortion and associated problems are reviewed in [19, 20].

An Eulerian algorithm, in general, works with coordinate system that is stationary in a laboratory frame of reference. However, in certain cases the coordinate system for an Eulerian scheme may move in a prescribed manner in order to account for the

continuously changing solution domain [19, 20]. Eulerian algorithms can be divided for convenience into three sub-categories, namely, fixed grid, adaptive grid and various mapping methods [19].

In the fixed grid methods, the grid is fixed in the solution domain and the location of the moving boundaries can be tracked using either surface [21] or volume tracking procedures [22]. The algorithms based on surface tracking track the moving boundaries using a series of interpolated curves through a discrete set of points whose motions during the solution process allow precise identification of the boundary locations [22, 23]. The high memory requirements associated with the necessity to store the locations of the points and other relevant information is a major drawback for the surface tracking methods particularly for problems with highly deforming boundaries [20]. The volume tracking algorithms on the other hand work by reconstructing the boundaries using marker quantities whenever necessary and, therefore, do not require storage of the boundary locations. The presence of a convenient marker within a computational cell and its quantity forms the basis of the various boundary reconstruction methodologies. Different versions of volume tracking algorithms exist, e.g., volume of fluid (VOF) [24], marker and cell (MAC) [25] and level set [26, 27] methods. These methods are based on standard spatial discretization schemes with low order of accuracy for the field equations, which are consistent with the diffused locations resulting from the boundary reconstruction processes.

The adaptive grid Eulerian methods for moving boundary problems rely on numerical mappings to adjust the grids at each time step so that two of the grid lines always overlap with the moving boundaries. Very high computational costs attributable to grid reconstruction at each time step act as a major predicament for these methods. For example, around 75% of the total computational cost for the problem discussed in [28] is attributable to the grid construction process alone. The computational costs for these methods are not affected significantly by the choice of spatial discretization techniques. The requirement of high accuracy in solution may lead to numerous challenges as the overall error is contributed by the error in grid generation as well as error from spatial

and temporal discretizations of the field equations. Analytical mapping of the irregular physical domain into a rectangular computational domain in the case of DT method can help in improving the accuracy at the cost of increased complexity of the field equations [9, 29]. However, such mappings are available only for a limited class of geometries [19] and reconstruction of the coefficient matrix during each time step can add to the overall computational cost by a substantial margin [9].

On the basis of the discussions on the various algorithms in the present section, it is evident that optimizing the requirements of high accuracy and computational efficiency constitutes the greatest challenge for any algorithm employed for solving moving boundary problems. Immersed boundary (IB) methods, as discussed in Section 1.3.1, are inherently better suited to deal with the variability of geometries of the solution domain. In fact, the concept as first proposed by Peskin [14] was applied to blood flow around the valves of a beating heart. Although the basic underlying concept of the IB methods make them efficient in dealing with moving boundaries, most of the IB methods for the moving boundary problems suffer from similar limitations as has been discussed for the fixed boundary problems, i.e., low spatial accuracy and uncertainties associated with the use of additional forcing to satisfy the boundary conditions.

1.4.2. IBC algorithm as applied to the moving boundary problems

The accuracy of any algorithm applied for solving moving boundary problems is influenced by two dominating factors – i) the precision in representation of the temporal and spatial locations of the boundaries and ii) the error associated with the scheme employed to discretize the field equation. The IBC method relies on spectral discretization of the spatial coordinates of the field equation and allows sharp identification of the locations of time-dependent physical boundaries. The algorithm is conceptually similar to the fixed grid Eulerian methods. As the algorithm works with an extended computational domain, therefore, the physical domain at any instance of time is immersed inside the computational domain. The field equations are required to be

discretized only once for the complete simulation and during every time step only the boundary conditions are discretized using the IBC concept to account for the change in boundary locations. This reduces the cost of generating the coefficient matrix significantly as only the part of the matrix corresponding to the boundary conditions are required to be reconstructed at every time step. Special solution methodologies can be developed that take advantage of this characteristic of the IBC algorithm in order to reduce the effective size of the system of equations required to be solved at each time step by a substantial margin without comprising the spectral accuracy in the spatial dimensions.

1.5. Preview of the dissertation

The materials presented in this dissertation are organized in seven chapters. This section is intended to provide a preview of the material presented in the different chapters. Chapters 2 to 4 are derived from manuscripts that have already been published or accepted for publication in various scientific journals. Chapters 5 and 6 are based on manuscripts that have been submitted to scientific journals for publication and are currently under review. The structure of the dissertation follows the integrated-article format and therefore, Chapters 2 to 6 retain the contents of the relevant manuscripts in their entirety without the associated abstracts.

Chapter 1 provides information regarding the objectives of the present research along with the motivating factors. It also briefly discusses the various algorithms available for handling fixed and moving boundary problems along with their respective limitations with the aid of the existing literature. The chapter also introduces the concept of immersed boundary conditions in conjunction with a concise discussion on the basic traits and prospective advantages and limitations of the algorithm in the context of both fixed and moving boundary problems.

Chapter 2 presents the application of the IBC algorithm for moving boundary problems in the context of two-dimensional conductive heat flow. The physical mechanism of diffusion is governed by a linear second order partial differential equation described by the Laplace operator. The model problem, therefore, permits convenient characterization of the fundamental disposition of the IBC algorithm when employed for moving boundary problems. Comparisons of different versions of iterative solvers that take advantage of the unusual structure of coefficient matrix resulting from the implementation of the IBC method are also presented. The computational cost advantage of the IBC method as compared with the spectral implementation of the DT method is demonstrated as well with the aid of [9].

Chapter 3 deals with the extension of the IBC algorithm to two-dimensional Stokes flow involving moving boundaries. Stokes flow when formulated in terms of the stream function is described by the biharmonic operator, which is a fourth-order linear partial differential equation. One of our principal objectives is to employ the IBC algorithm for solving moving boundary problems described by the three-dimensional Navier-Stokes equations. When such a problem is posed in terms of the velocity-vorticity formulation (with elimination of the pressure), the Laplace and the biharmonic operators become the main constituents. Therefore, the Stokes flow problem chosen in Chapter 3 is the natural step in the investigation of the extension of the IBC algorithm from the heat conduction problem towards the full Navier-Stokes problem. Moreover, from a physical perspective most flows in micro-channels fall under the purview of the Stokes limitation and for that matter spectrally-accurate algorithm for handling Stokes flow is going to be very useful tool as well. Effectiveness of various iterative solution strategies that take advantage of the special structure of the coefficient matrix resulting from the implementation of the IBC algorithm are also explored in this chapter.

In Chapter 4, an over-determined formulation of the IBC algorithm (as opposed to the 'classical formulation' referred to earlier) is proposed by using additional boundary constraints that result from the discretization of the physical boundary conditions with the use of the IBC algorithm. Use of the additional constraints that leads to over-

determined systems are explored in this chapter in order to improve the accuracy of the IBC method and to expand the applicability of the method to more extreme geometries. Possible improvement of accuracy through the use of the over-determined formulation could be problem dependent. Therefore, the formulation has been tested on three model problems, i.e., the Laplace equation, the biharmonic equation and the Navier-Stokes equations; and thus the chapter covers the most commonly encountered types of operators in computational fluid dynamics problems.

Specialized linear solvers suitable for the spectral implementation of the Immersed Boundary Conditions (IBC) method are developed and tested in Chapter 5. For convenience, Stokes flow is considered as the model problem. The efficient specialized solvers have been implemented for both the classical and the over-determined IBC formulations. Performances of the new solvers have been compared with the standard direct solvers and different iterative solvers. The performance of the specialized IBC solvers has also been compared against spectrally-accurate domain transformation (DT) method, which is based on the generation of the boundary conforming grids. Application of the new solvers to the unsteady fixed boundary problems has also been investigated. The purpose of developing the new solvers has been to make three-dimensional calculations using the IBC method practically feasible by reducing the computing costs for the irregular geometries (e.g., channels bounded by corrugated walls) close to the computing costs for the regular geometries (e.g. channels with straight walls).

Chapter 6 presents the extension of the IBC algorithm as applied to simulations of viscous flows with moving boundaries described by the two-dimensional Navier-Stokes equations. Up to fourth-order implicit temporal discretization methods have been implemented. Performances of various linear solvers employed in the solution process have been evaluated and a new class of solver that has its root in the specialized solvers presented in Chapter 5 is proposed and tested.

Finally, Chapter 7 summarizes the various important observations made with regard to the different formulations (classical and over-determined) of IBC algorithm as well as

different solvers (direct and iterative) as applied to the different two-dimensional model problems in a summarized form. Possible extension of the IBC algorithm to three-dimensional problems is later laid out in the context three-dimensional velocity-vorticity formulation of the Navier-Stokes equations. The different issues with regard to the implementation of the IBC algorithm for three-dimensional problems are discussed using the test results based on a simpler physical problem that deals with unsteady heat diffusion in the presence of fixed as well as moving boundaries. The results to be presented in this regard are extracted from the report [30]. My colleague D.C. Del Rey Fernandez is responsible for the formulation, programming implementation and computations of the results presented in [30] as a part of his Masters research. I have assisted Prof. Floryan in supervising him throughout the different stages of formulation and implementation of the algorithm along with presentation and discussion on the various aspects of the computed results presented in [30]. I have also been a principal contributor in the writing of the report [30]. Various solvers that take advantage of the special structure of the coefficient matrix including a version of the specialized direct solvers based on the concept presented in Chapter 5 as well as serial and parallel iterative solvers are explored for the three-dimensional problem in search for an efficient solver. Possible applications of the research work associated with the dissertation are also briefly presented in this chapter using two illustrative examples. Results obtained by the IBC method are verified against those available in literature to validate the reliability of the algorithm. Finally, the conclusions on the various aspects of the research associated with this dissertation are presented in a summarized form along with recommendations for future work.

1.6. References

[1] Croce, G. and D'Agaro, P., Numerical simulation of roughness effect on microchannel heat transfer and pressure drop in laminar flow, *J. Phys. D: Appl. Phys.*, vol. 38, pp. 1518-1530 (2005).

[2] McLean, J.W., Computations of turbulent flow over a moving wavy boundary, *Phys. Fluids*, vol. 26, pp. 2065-2073 (1983).

[3] Floryan, J.M., Three-dimensional instabilities of laminar flow in a rough channel and the concept of hydraulically smooth wall, *Eur. J. Mech. B/Fluids*, vol. 26, pp. 305-329 (2007).

[4] Floryan, J.M., Centrifugal instability of Couette flow over a wavy wall, *Phys. Fluids*, vol. 14, pp. 312-322 (2002).

[5] Gad-el-Hak, M., Pollard, A. and Bonnet, J-P., Flow control: Fundamentals and practices, *Springer-Verlag* (1998).

[6] Moin, P. and Mahesh, K., Direct numerical simulations: a tool in turbulence research, *Ann. Rev. Fluid Meh.*, vol. 30, pp. 539-578 (1998).

[7] Yoon, S., Na, S., Wang, Z.J., Bons, J.P. and Shih, T.I-P., Flow and heat transfer over rough surfaces: usefulness of 2-D roughness-resolved simulations, *44th Aero. Sci. Meet. Exhibit*, Reno, Nevada (2006).

[8] Jiang, Y. and Floryan, J.M., Finite-difference 4th-order compact scheme for the direct numerical simulation of instabilities of shear layers, *Int. J. Numer. Meth. Fluids*, vol. 48, pp. 1259-1281 (2005).

[9] Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for unsteady flow problems described by the Laplace operator, *Int. J. Num. Meth. Fluids*, vol. 56, pp. 1765-1786 (2007).

[10] Angelis, V.De, Lombardi, P. and Banerjee, S., Direct numerical simulation of turbulent flow over a wavy wall, Phys. Fluids, vol. 9, pp. 2429-2442 (1997).

[11] Cabal, A., Szumbarski, J. and Floryan, J.M., Numerical simulation of flows over corrugated walls, Comp. Fluids, vol. 30, pp. 753-776 (2000).

[12] Tsangaris, S. and Potamitis, D., On laminar small Reynolds-number flow over wavy walls, *Acta Mechanica*, vol. 61, pp. 109-115 (1986).

[13] Floryan, J.M. and Dallmann, U., Flow over a leading edge with distributed roughness, *J. Fluid. Mech.*, vol. 216, pp. 629-656 (1990).

[14] Peskin, C.S., The fluid dynamics of heart valves: experimental, theoretical and computational methods, *Annu. Rev. Fluid Mech.*, vol. 14 pp. 235-59 (1982).

[15] Mittal, R. and Iaccarino, G., Immersed boundary methods, *Annu. Rev. Fluid Mech.*, vol. 37 pp. 239-261 (2005).

[16] Peskin, C.S., The immersed boundary method, *Acta Numerica*, 479-517 (2002).

[17] Szumbarski, J. and Floryan, J.M., A direct spectral method for determination of flows over corrugated boundaries, *J. Comp. Phys.*, vol. 153, pp. 378-402 (1999).

[18] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., Spectral methods in fluid dynamics, *Springer* (1987).

[19] Floryan, J.M. and Rasmussen, H., Numerical analysis of viscous flows with free surfaces, *Appl. Mech. Rev.*, vol. 42, pp. 323-341 (1989).

[20] Navti, S.E., Lewis, R.W. and Taylor, C., Numerical simulation of viscous free surface flows, *Int. J. Num. Meth. Heat & Fluids Flow*, vol. 8, pp. 445-464 (1997).

[21] Scardovelli, R. and Zaleski, S., Direct numerical simulation of free surface and interfacial flow, *Annu Rev. Fluid Mech.*, vol. 31, pp. 567-603 (1999).

[22] Hyman, J.M., Numerical methods for tracking of interfaces, *Physica,* vol. 12D, pp. 396-407 (1984).

[23] Glimm, J., Grove, J.W., Li, X.L., Shyue, K.M., Zeng, Y. and Zhang, Q., Three-dimensional front tracking, *SIAM J. Sci. Computing,* vol. 19, pp. 703-727 (1998).

[24] Hirt, C.W. and Nichols, B.D., Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *J.Comp.Phys.*, vol. 39, pp. 201-225 (1981).

[25] Harlow, F.H. and Welch, J.E., Numerical study of large amplitude free surface motions, *Phys.Fluids*, vol. 9, pp. 842-851 (1966).

[26] Osher, S.J. and Sethian, J.A., Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comp .Phys.*, vol. 79, pp. 12-49 (1988).

[27] Sethian, J.A. and Smereka, P., Level set methods for fluid interfaces, Annu. *Rev. Fluid Mech.*, vol. 35, pp. 341-372 (2003).

[28] Inculet, I., Floryan, J.M. and Haywood, R., Dynamics of water droplets break-up in electric fields, *IEEE Transactions on Industry Applications*, vol. 28, pp. 1203-1209 (1992).

[29] Hamed, M. and Floryan, J.M., Numerical simulation of unsteady nonisothermal capillary interfaces, *J.Comp.Phys.* vol. 145, pp. 110-140 (1998).

[30] Fernandez, D.C.D.R., Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for three-dimensional heat diffusion problems, *Expert Systems in Fluid Dynamics Research Laboratory Report ESFD-1/2009*, Department of Mechanical and Materials Engineering, The University of Western Ontario, London, Ontario, N6A 5B9, Canada (2009).

CHAPTER 2

Moving boundary problems described by the Laplace operator[1]

2.1. Introduction

The term 'moving boundary problem' refers to a problem where a boundary of the solution domain changes location during the solution process. The related problem where boundary motion is *a priori* unknown and has to be predicted as a part of the solution process is usually referred to as a 'free boundary problem'. In the present chapter we are interested in finding a spectrally-accurate solution to the first problem that maintains sharp resolution of the location of the boundary.

Moving-boundary algorithms have been reviewed in [1] and can be classified as Eulerian, Lagrangian and mixed. Eulerian algorithms rely on a coordinate system that is stationary in a laboratory frame of reference or moves in a prescribed manner (Galilean transformation). Such algorithms can be divided for convenience into fixed grid methods, adaptive grid methods and various mapping methods.

In the fixed grid methods the grid is fixed in the domain and, since in general, the location of the moving boundary does not overlap with a grid line, a special procedure must be added to identify the location of the moving boundary. This location can be

---

[1]A version of this chapter has been published as –

Husain, S.Z., and Floryan, J.M., Implicit Spectrally-Accurate Method for Moving Boundary Problems using Immersed Boundary Conditions Concept, *Journal of Computational Physics*, vol. 227(9), pp. 4459-4477 (2008).

tracked using either surface or volume tracking procedures. In the former approach a set of points is introduced to mark location of the boundary which is represented as a set of interpolated curves. These points are moved in a prescribed manner during the solution process [2,6] and provide ability for precise identification of boundary location. In the latter approach the information about the boundary location is not stored but the boundary is reconstructed whenever necessary on the basis of the presence of certain quantity of a convenient marker within computational cell, e.g., MAC - Marker and Cell [3], VOF -Volume of Fluid [4,12] and Level Set [9,13] methods. The reconstruction process leads to a diffused location of the boundary but the overall accuracy can be made consistent with low accuracy discretization schemes for the field equations.

In the adaptive grid methods the computational grid is continuously adapted so that one of the grid lines always overlaps with the moving boundary. This leads to a large computational overhead associated with grid re-generation at each simulation time step. In this sense numerical grid generation combined with finite-difference discretization of the field equations and grid generation for the finite-element discretization are conceptually identical. In certain situations numerical grid generation can be replaced by analytical mappings, but this does not alter the basic concept of such methods. The overall accuracy of tracking of the location of the boundary is determined by the accuracy of numerical implementation of various mapping schemes.

Lagrangian methods are characterized by a coordinate system that moves with the fluid. Each computational cell always contains the same fluid and its tracking requires solution of an initial value problem. These methods are well suited for moving boundary problems as they permit simple delineation of moving boundaries. The two main problems involve mesh tangling and loss of numerical accuracy associated with highly distorted meshes. Possible remedies are discussed in [1]. Mixed Lagrangian-Eulerian methods rely on the combination of concepts described above [1].

The accuracy of the available methods for solving moving boundary problems is dictated by the accuracy of representation of boundary location and by the discretization error of

the field equations. Spectral methods provide the lowest error for the field equations but are limited to solution domains with regular geometries. The variability of geometry of the solution domain represents the main challenge for implementation of these methods. One possible implementation can be based on the concept of immersed boundary conditions [11]. This concept has been proposed in [8,10] in the context of simulation of cardiac dynamics but its first spectrally accurate implementation for fixed boundary problems has been given in [11]. Immersed boundary conditions method uses fixed, regular solution domain for the required spectral expansions and the actual physical domain is submerged in the interior of the computational domain. Boundary conditions at the edges of the physical domain are replaced by constraints imposed inside computational domain. The method is analogous to the fixed grid methods discussed above.

This chapter describes extension of the algorithm proposed in [11] to the case of moving boundary problems. The analysis is carried out in the context of diffusion problem. The model problem is described in Section 2.2. The Immersed Boundary Conditions (IBC) method is discussed in Section 2.3. The Domain Transformation (DT) method that is used to demonstrate the efficiency of the IBC method is discussed in Section 2.4. Results of numerical tests are discussed in Section 2.5. Section 2.6 provides a short summary of the main conclusions.

2.2. Model problem

We shall pose the problem of interest in the context of heat conduction. Consider unsteady conductive heat flow in a slot bounded by corrugated walls whose geometry changes as a function of time and is described by the following relations (see Fig. 2.1)

$$y_L(x,t) = -1 + \sum_{n=-\infty}^{\infty} H_L^{(n)}(t) e^{inax}, \qquad y_U(x,t) = 1 + \sum_{n=-\infty}^{\infty} H_U^{(n)}(t) e^{inax} \qquad (2.2.1a,b)$$

where $H_L^{(n)} = H_L^{(-n)^*}$, $H_U^{(n)} = H_U^{(-n)^*}$ and star denotes complex conjugate. Such problems are of interest in the case of nano-channels and nano-switches where the effects of surface roughness could be significant [7]. The slot is periodic with wavelength $\lambda=2\pi/\alpha$ and extends to $\pm\infty$ in the x-direction. The dimensionless field equation describing heat flow has the form

$$\partial^2\theta/\partial x^2 + \partial^2\theta/\partial y^2 = \partial\theta/\partial t. \qquad (2.2.2)$$

where θ denotes temperature. This equation needs to be supplemented by suitable initial and boundary conditions, which are taken to be in the form

$$\theta(x,y,t=0) = \theta, \quad y_L(x,0)=y_{Li}(x), \quad y_U(x,0)=y_{Ui}(x), \qquad (2.2.3a\text{-}c)$$

$$\theta(x, y_L(x,t), t) = \theta_L(x,t), \quad \theta(x, y_U(x,t), t) = \theta_U(x,t) \qquad (2.2.4a,b)$$

where θ_i, y_{Li}, y_{Ui}, θ_L, θ_U are considered to be known.

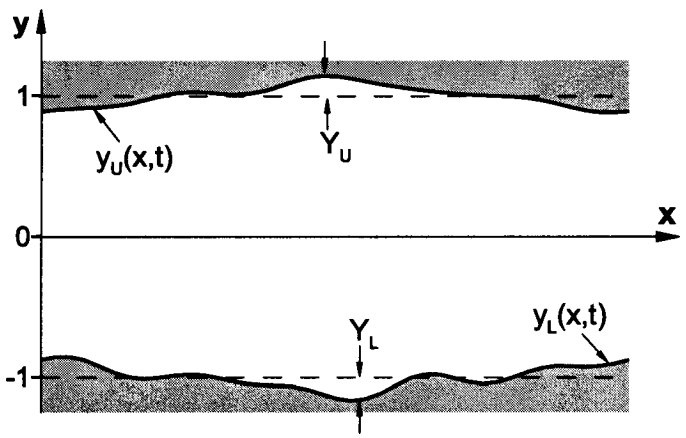


Figure 2.1. *Sketch of the instantaneous form of the domain of interest in the physical plane.*

The time variations of the temperature field result from the variations of the temperature along the edges of the physical domain as well as from the deformation of the domain. The main difficulty in finding a solution to (2.2.2)-(2.2.4) is associated with the

variations of the geometry of the physical domain, which leads to the moving boundary problem. We shall discuss two solution methods suitable for this problem, i.e., a method based on the concept of immersed boundary conditions (IBC) and a method based on the concept of domain transformation (DT). The latter method will be used to illustrate advantages of the IBC method. We shall begin our discussion with the former method.

2.3. The immersed boundary conditions (IBC) method

We are interested in the determination of the solution of (2.2.2)-(2.2.4) with spectral accuracy. We shall use Fourier expansions in the x-direction and expansions in terms of Chebyshev polynomials in the y-direction. Standard definition of Chebyshev polynomials uses the domain (-1,1), however the physical domain is confined between $(1+Y_U)$ and $(-1-Y_L)$ where Y_U and Y_L denote the locations of extremities of the domain of interest during time interval of interest (see Fig. 2.1). The first step in the solution process involves a mapping from the physical (x,y) coordinates to the computational $(x,\hat{y})$ coordinates in the form

$$\hat{y}=2\frac{y-(1+Y_U)}{1+Y_U-(-1-Y_L)}+1 \tag{2.3.1}$$

where $\hat{y}\in\langle -1,1\rangle$. The governing equation takes the form

$$\partial^2\theta/\partial x^2+\Gamma^2\,\partial^2\theta/\partial\hat{y}^2=\partial\theta/\partial t \tag{2.3.2}$$

where $\Gamma=2/(2+Y_U+Y_L)$ is a constant. Locations of the corrugated boundaries in the $(x,\hat{y})$ plane are given as

$$\hat{y}_L(x,t)=\sum_{n=-\infty}^{\infty}A_L^{(n)}(t)e^{in\alpha x}, \qquad \hat{y}_U(x,t)=\sum_{n=-\infty}^{\infty}A_U^{(n)}(t)e^{in\alpha x}, \tag{2.3.3a,b}$$

where $A_L^{(0)}(t)=1+\Gamma\left[-2-Y_U+H_L^{(0)}(t)\right]$, $A_L^{(n)}(t)=\Gamma H_L^{(n)}(t)$ for $n\neq 0$, $A_U^{(0)}(t)=1+\Gamma\left[-Y_U+H_U^{(0)}(t)\right]$, $A_U^{(n)}(t)=\Gamma H_U^{(n)}(t)$ for $n\neq 0$. The boundary conditions at the transformed boundaries become

$$\theta(x,\hat{y}_L(x,t),t)=\theta_L(x,t), \qquad \theta(x,\hat{y}_U(x,t),t)=\theta_U(x,t). \tag{2.3.4a,b}$$

The solution can be represented in the form of Fourier expansion

$$\theta(x,\hat{y},t)=\sum_{n=-\infty}^{\infty}\Phi^{(n)}(\hat{y},t)e^{in\alpha x}\approx\sum_{n=-N_M}^{N_M}\Phi^{(n)}(\hat{y},t)e^{in\alpha x} \tag{2.3.5}$$

where $\Phi^{(n)}(\hat{y},t)=\Phi^{(-n)^*}(\hat{y},t)$ and star denotes complex conjugate. Substitution of (2.3.5) into the field equation and separation of Fourier components lead to an uncoupled system of parabolic partial differential equations for $\Phi^{(n)}$, $n\in\langle 0,N_M\rangle$, of the type

$$\partial\Phi^{(n)}/\partial t=\left(\Gamma^2D^2-n^2\alpha^2\right)\Phi^{(n)} \tag{2.3.6}$$

where $D=d/d\hat{y}$. Two types of temporal discretizations are used. The two-step implicit method results in the following relations

$$\left[\Gamma^2D^2-\left(n^2\alpha^2+1.5\Delta t^{-1}\right)\right]\Phi_{\tau+1}^{(n)}=-2\Delta t^{-1}\Phi_{\tau}^{(n)}+0.5\Delta t^{-1}\Phi_{\tau-1}^{(n)}, \quad n\in\langle 0,N_M\rangle \tag{2.3.7a}$$

while similar relations resulting from the one-step, self-starting implicit method is given by

$$\left[\Gamma^2D^2-\left(n^2\alpha^2+\Delta t^{-1}\right)\right]\Phi_{\tau+1}^{(n)}=-\Delta t^{-1}\Phi_{\tau}^{(n)}, \quad n\in\langle 0,N_M\rangle. \tag{2.3.7b}$$

In the above the subscript τ denotes the time step and Δt stands for the (constant) length of the time step. Relation (2.3.7a) has the form of an inhomogeneous ordinary differential equation for $\Phi^{(n)}_{\tau+1}$. The following discussion will be carried out in the context of the two-step method, while the relevant relations for the one-step method can be readily deduced. The unknown function $\Phi^{(n)}_{\tau+1}$ can be represented in terms of expansions based on Chebyshev polynomials in the form

$$\Phi^{(n)}_{\tau+1}(\hat{y}) = \sum_{k=0}^{\infty} G^{(n)}_{k,\tau+1} T_k(\hat{y}) \approx \sum_{k=0}^{N_T} G^{(n)}_{k,\tau+1} T_k(\hat{y}) \tag{2.3.8}$$

where T_k denotes the Chebyshev polynomial of k^{th} order and $G^{(n)}_{k,\tau+1}$ are the unknown coefficients of the expansion. Substitution of (2.3.8) into (2.3.7a) gives

$$\left[\Gamma^2 D^2 - \left(n^2\alpha^2 + 1.5\Delta t^{-1}\right)\right]\sum_{k=0}^{\infty} G^{(n)}_{k,\tau+1} T_k = -2\Delta t^{-1}\sum_{k=0}^{\infty} G^{(n)}_{k,\tau} T_k + 0.5\Delta t^{-1}\sum_{k=0}^{\infty} G^{(n)}_{k,\tau-1} T_k \tag{2.3.9}$$

We use Galerkin procedure [14-16] to develop equations for the unknowns $G^{(n)}_{k,\tau+1}$, i.e., we multiply both sides of (2.3.9) by $T_j(\hat{y})$ and integrate with the weight function $\hat{\omega} = 1/\sqrt{1-\hat{y}^2}$ to get

$$\begin{aligned}\sum_{k=0}^{N_T}\left[\langle T_j, D^2 T_k\rangle - \left(n^2\alpha^2 + 1.5\Delta t^{-1}\right)\langle T_j, T_k\rangle\right] G^{(n)}_{k,\tau+1} &= -2\Delta t^{-1}\sum_{k=0}^{N_T}\langle T_j, T_k\rangle G^{(n)}_{k,\tau} \\ &+ 0.5\Delta t^{-1}\sum_{k=0}^{N_T}\langle T_j, T_k\rangle G^{(n)}_{k,\tau-1} \quad , \qquad j \in \langle 0, N_T\rangle\end{aligned} \tag{2.3.10}$$

where the inner product is defined as $\langle f_j(\hat{y}), g_k(\hat{y})\rangle = \int_{-1}^{1} f_j(\hat{y}) g_k(\hat{y})\hat{\omega}(\hat{y})d\hat{y}$. Evaluation of inner products can be significantly simplified by taking advantage of the well known orthogonality properties of the Chebyshev polynomials [14-16]. Equation (2.3.10) leads to N_T-1 decoupled algebraic equations for each Fourier mode; two additional equations

required in order to close the system need to be derived from the boundary conditions and these conditions provide coupling between different Fourier modes [5, 11].

The boundary conditions are to be enforced along the lines $\hat{y}_L(x,\tau+1)$ and $\hat{y}_U(x,\tau+1)$ while the solution domain remains fixed at $\hat{y} \in \langle -1,1 \rangle$. To explain the immersed boundary conditions method in general, at time τ+1 we evaluate the unknown $\theta_f(x,\tau+1) \equiv \theta(x,f(x,\tau+1),\tau+1)$ along an arbitrary time dependent line that at time $t=\tau+1$ occupies position $y=f(x,\tau+1)$, such that f is a periodic function with period $\lambda=2\pi/\alpha$ and $|f(x,\tau+1)|\leq 1$. The function $f(x,\tau+1)$ can be expressed without losing generality as

$$f(x,\tau+1) = \sum_{n=-\infty}^{\infty} P_{\tau+1}^{(n)} e^{in\alpha x} \approx \sum_{n=-N_A}^{N_A} P_{\tau+1}^{(n)} e^{in\alpha x} \tag{2.3.11}$$

where one in practice deals with a finite number of terms N_A. The unknown $\theta_f(x,\tau+1)$ is periodic in x with the same period λ and thus can be expressed in terms of Fourier series as

$$\theta_f(x,\tau+1) \equiv \theta(x,f(x,\tau+1),\tau+1) = \sum_{n=-N_\theta}^{N_\theta} \varphi_{\tau+1}^{(n)} e^{in\alpha x} \tag{2.3.12}$$

where $N_\theta = N_T N_A + N_M$. Since the flow representation is limited to N_M+1 modes, we will be able to enforce constraints only on the first (N_M+1) terms in (2.3.12). The same unknown can be expressed using the discretized form of the solution, i.e.,

$$\theta_f(x,\tau+1) = \sum_{n=-N_M}^{N_M} \Phi^{(n)}(f(x,\tau+1),\tau+1) e^{in\alpha x} = \sum_{n=-N_M}^{N_M} \sum_{k=0}^{N_T} G_{k,\tau+1}^{(n)} T_k(f(x,\tau+1)) e^{in\alpha x} \tag{2.3.13}$$

Since $T_k(f(x,\tau+1))$ is periodic in x, it can be expressed in terms of Fourier expansion as follows

$$T_k(f(x,\tau+1)) = \sum_{m=-N_s}^{N_s} w_{k,\tau+1}^{(m)} e^{im\alpha x}, \qquad (2.3.14)$$

where $\max(N_S)=N_T*N_A$. The expansion coefficients in (2.3.14) can be evaluated with the help of the recurrence relation for Chebyshev polynomials in the form $T_{k+1}(\hat{y}) = 2\hat{y}T_k(\hat{y}) - T_{k-1}(\hat{y})$ that leads to the following recurrence relation

$$w_{k+1,\tau+1}^{(m)} = 2\sum_{n=-\infty}^{\infty} P_{\tau+1}^{(n)} w_{k,\tau+1}^{(m-n)} - w_{k-1,\tau+1}^{(m)} \qquad (2.3.15)$$

whose evaluation begins at k=0 and results in

$$w_{0,\tau+1}^{(0)} = 1, \; w_{0,\tau+1}^{(m)} = 0 \;\; \text{for} \; |m| \geq 1; \qquad w_{1,\tau+1}^{(m)} = P_{\tau+1}^{(m)} \;\; \text{for} \; |m| \geq 0. \qquad (2.3.16)$$

Substitution of (2.3.14) into (2.3.13) gives

$$\begin{aligned} \theta_l(x,\tau+1) &= \sum_{n=-N_M}^{N_M} \sum_{k=0}^{N_T} \sum_{m=-N_s}^{N_s} G_{k,\tau+1}^{(n)} w_{k,\tau+1}^{(m)} e^{i(n+m)\alpha x} \\ &= \sum_{n=-N_\theta}^{N_\theta} \sum_{m=-N_M}^{N_M} \sum_{k=0}^{N_T} G_{k,\tau+1}^{(m)} w_{k,\tau+1}^{(n-m)} e^{in\alpha x} \end{aligned} \qquad (2.3.17)$$

and comparison of (2.3.12) with (2.3.17) gives

$$\varphi_{\tau+1}^{(n)} = \sum_{m=-N_M}^{N_M} \sum_{k=0}^{N_T} G_{k,\tau+1}^{(m)} w_{k,\tau+1}^{(n-m)}. \qquad (2.3.18)$$

Equation (2.3.18) can be used to express boundary conditions along the lines $\hat{y}_L(x,\tau+1)$ and $\hat{y}_U(x,\tau+1)$. In the case of our model problem these boundary conditions take the following form

$$\sum_{m=-N_M}^{N_M}\sum_{k=0}^{N_T} G_{k,\tau+1}^{(m)}\left(w_L\right)_{k,\tau+1}^{(n-m)} = \Phi_{L,\tau+1}^{(n)},$$

$$\sum_{m=-N_M}^{N_M}\sum_{k=0}^{N_T} G_{k,\tau+1}^{(m)}\left(w_U\right)_{k,\tau+1}^{(n-m)} = \Phi_{U,\tau+1}^{(n)} \qquad (2.3.19a,b)$$

where $\theta_L(x,\tau+1) = \sum_{n=-N_M}^{N_M}\Phi_{L,\tau+1}^{(n)}e^{in\alpha x}$, $\theta_U(x,\tau+1) = \sum_{n=-N_M}^{N_M}\Phi_{U,\tau+1}^{(n)}e^{in\alpha x}$ and (2.3.19a,b)

corresponds to the lower and upper walls, respectively. Equations (2.3.10) and (2.3.19) form a complete set of algebraic equations for the unknown coefficients $G_{k,\tau+1}^{(n)}$, k = 0,....,N_T, n=0,....,N_M. A direct solution of this system moves simulations by one time step forward and the resulting algorithm will be referred to as the direct algorithm.

In the case of time steps of constant length, the coefficients of the discretized Eqs (2.3.10) do not depend on time and thus this part of the coefficient matrix need to be constructed only once at the beginning of the solution process. The matrix entries corresponding to the boundary conditions (2.3.19) need to be reconstructed at every step of the simulations process; the reader should note that this is where the information about the motion of the boundaries enters into considerations.

The matrix of coefficients can be very large when a large number of Fourier modes are required and this motivates the search for an efficient solution method. Before solution is to be carried out the matrix is reduced to a real form by taking advantage of the complex conjugate properties $\Phi^{(n)}(\hat{y},t) = \Phi^{(-n)^*}(\hat{y},t)$ (see Eq. 2.3.5) resulting in a structure shown in Fig. 2.2 where the horizontal lines show the coupling effect of boundary conditions (2.3.19) and the blocks of coefficients in the upper triangular form resulting from the discretization of the differential equation (2.3.6) form a band along the diagonal. Such structure suggests the use of an iterative solution algorithm based on the decoupling of Fourier modes. The unknowns corresponding to a Fourier mode of interest in equation (2.3.19) at the current time step can be expressed in terms of the remaining Fourier

modes using their values from the previous time step (or from the previous iteration). The solution process begins with mode 0, proceeds to the next mode using the most recent information available and continues until the last mode N_M is reached, and then it is repeated until a convergence criterion is satisfied. In this way, the solution of the complete system of size $(N_T+1)*(2N_M+1)$ is replaced by a repetitive solution of systems of (N_T+1) equations for each Fourier mode. The rate of convergence is generally very good; it decreases with an increase of the current amplitude and the wave number characterizing wall geometry. We shall refer to the iterative algorithm as the "decoupled algorithm" and discuss performance of its various variants in Section 2.5. We wish to stress at this moment that the use of the decoupled algorithm significantly reduces memory requirements as one needs to work with many small matrices instead of one very large matrix as well as it opens the possibility for parallelization of the computations. This issue becomes significant in the case of three-dimensional problems and large number of Fourier modes and Chebyshev polynomials.

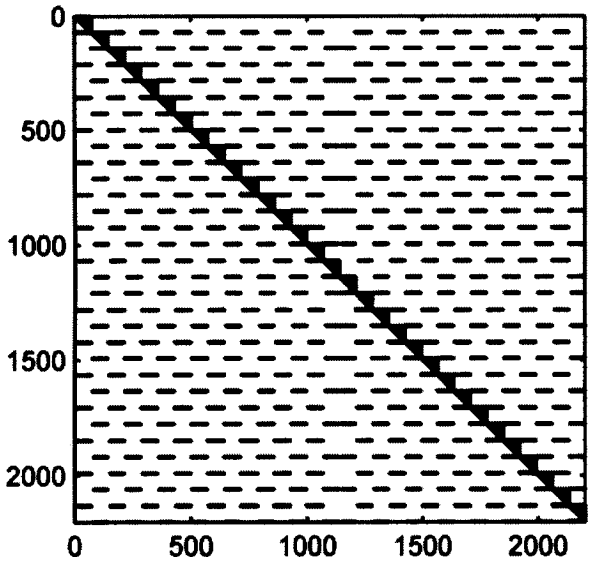


Figure 2.2. *Structure of the coefficient matrix resulting from the implementation of the Immersed Boundary Condition Method for N_M=15 and N_T=70 and the test problem (2.5.7)-(2.5.8). Only non-zero elements are marked.*

2.4. The domain transformation (DT) method

A very popular approach in dealing with moving boundary problems involves adjusting grid so that one of the grid lines always overlaps with the instantaneous location of the boundary [1]. The adjustments may have many forms, extending from analytical mappings to numerical coordinate generation. The adjustments in the shape of elements in the finite element method can also be viewed as a form of mapping. In order to judge the efficiency of the IBC method as compared to all other mapping-type methods, we selected a simple method that relies on analytical mapping. The current shape of the slot in the physical domain (x,y) is mapped into a straight slot in the computational domain (ξ,η) with $\eta \in <-1,1>$ using mapping in the form

$$\xi = x, \qquad \eta = 2\frac{(y - y_U(x,t))}{y_U(x,t) - y_L(x,t)} + 1 \tag{2.4.1}$$

After mapping Eq. (2.2.2) takes the form

$$\frac{\partial^2\theta}{\partial\eta^2} + W_1(\xi,\eta,t)\frac{\partial\theta}{\partial\eta} + W_2(\xi,\eta,t)\frac{\partial^2\theta}{\partial\xi\partial\eta} + W_3(\xi,\eta,t)\frac{\partial^2\theta}{\partial\xi^2} = W_3(\xi,\eta,t)\frac{\partial\theta}{\partial t} \tag{2.4.2}$$

where $W_1(\xi,\eta,t) = (\eta_{xx} - \eta_t)/(\eta_x^2 + \eta_y^2)$, $W_2(\xi,\eta,t) = 2\eta_x/(\eta_x^2 + \eta_y^2)$ and $W_3(\xi,\eta,t) = 1/(\eta_x^2 + \eta_y^2)$. Expressions for η_t, η_x, η_{xx} and η_y are given below

$$\eta_t = [(y_L)_t(\eta - 1) - (y_U)_t(\eta + 1)]/(y_U - y_L), \quad \eta_x = [(y_L)_\xi(\eta - 1) - (y_U)_\xi(\eta + 1)]/(y_U - y_L),$$

$$\eta_{xx} = \frac{\eta(y_U - y_L)\left[(y_L)_\xi - (y_U)_\xi\right]\left[(y_L)_{\xi\xi} - (y_U)_{\xi\xi}\right] + 2y_L(y_U)_{\xi\xi} - 2y_U(y_L)_{\xi\xi}}{(y_U - y_L)^2} +$$

$$\frac{-2\eta\left[(y_U)_\xi - (y_L)_\xi\right]^2 + 2\left[(y_U)_\xi^2 - (y_L)_\xi^2\right]}{(y_U - y_L)^2} \quad \text{and}$$

$$\eta_y = \frac{2}{y_U - y_L},$$

where subscripts ξ, t denote derivative $d/d\xi$, d/dt, respectively.

Solution of (2.4.2) is represented in terms of Fourier expansion as

$$\theta(\xi,\eta,t) = \sum_{n=-\infty}^{\infty} \Psi^{(n)}(\eta,t) e^{in\alpha\xi} \approx \sum_{n=-N_M}^{N_M} \Psi^{(n)}(\eta,t) e^{in\alpha\xi} \tag{2.4.3}$$

where $\Psi^{(n)} = \Psi^{(-n)^*}$ and the known coefficients W_1, W_2 and W_3 at each time $t=\tau$ are replaced by their Fourier expansions

$$W_1(\xi,\eta,\tau) = \sum_{m=-2N_M}^{2N_M} w_{1,\tau}^{(m)}(\eta) e^{im\alpha\xi}, \quad W_2(\xi,\eta,\tau) = \sum_{m=-2N_M}^{2N_M} w_{2,\tau}^{(m)}(\eta) e^{im\alpha\xi},$$

$$W_3(\xi,\eta,\tau) = \sum_{m=-2N_M}^{2N_M} w_{3,\tau}^{(m)}(\eta) e^{im\alpha\xi}. \tag{2.4.4}$$

Substitution of (2.4.3) and (2.4.4) into (2.4.2), separation of Fourier components and the use of a two step implicit method for the temporal discretization leads to

$$\begin{aligned} &D^2\Psi_{\tau+1}^{(n)} + \sum_{s=-N_M}^{N_M}\left[w_{1,\tau+1}^{(n-s)} + is\alpha w_{2,\tau+1}^{(n-s)}\right] D\Psi_{\tau+1}^{(s)} - \sum_{s=-N_M}^{N_M}\left[(s\alpha)^2 + 1.5\Delta t^{-1}\right] w_{3,\tau+1}^{(n-s)}\Psi_{\tau+1}^{(s)} \\ &= -2\Delta t^{-1}\sum_{s=-N_M}^{N_M} w_{3,\tau}^{(n-s)}\Psi_{\tau}^{(s)} + 0.5\Delta t^{-1}\sum_{s=-N_M}^{N_M} w_{3,\tau-1}^{(n-s)}\Psi_{\tau-1}^{(s)}, \qquad n \in \langle 0, N_M \rangle \end{aligned} \tag{2.4.5a}$$

where $D=d/d\eta$. A similar expression for the one step implicit method has the form

$$D^2\Psi_{\tau+1}^{(n)} + \sum_{s=-N_M}^{N_M}\left[w_{1,\tau+1}^{(n-s)} + is\alpha w_{2,\tau+1}^{(n-s)}\right]D\Psi_{\tau+1}^{(s)} - \sum_{s=-N_M}^{N_M}\left[(s\alpha)^2 + \Delta t^{-1}\right]w_{3,\tau+1}^{(n-s)}\Psi_{\tau+1}^{(s)}$$
$$= -\Delta t^{-1}\sum_{s=-N_M}^{N_M} w_{3,\tau}^{(n-s)}\Psi_{\tau}^{(s)} \qquad , \qquad n \in \langle 0, N_M \rangle \quad . \tag{2.4.5b}$$

The reader may note that all equations (2.4.5a,b) are coupled together through the known variable coefficients. The unknown $\Psi_{\tau+1}^{(n)}(\eta)$ can be expressed with spectral accuracy using the Chebyshev expansion in the form

$$\Psi_{\tau+1}^{(n)}(\eta) = \sum_{k=0}^{\infty} F_{k,\tau+1}^{(n)}\, T_k(\eta) \;\approx\; \sum_{k=0}^{N_T} F_{k,\tau+1}^{(n)}\, T_k(\eta) \;. \tag{2.4.6}$$

Use of the Galerkin procedure described in Section 2.3 leads to N_T-1 algebraic equations for $F_{k,\tau+1}^{(n)}$ for each Fourier mode. The remaining closing conditions come from the boundary conditions.

The treatment of boundary conditions follows standard procedures. The boundary conditions have the form

$$\theta(\xi, \eta = -1, t) = \theta_L(\xi, t) = \sum_{n=-N_M}^{N_M} \Psi_L^{(n)}(t) e^{in\alpha\xi},$$

$$\theta(\xi, \eta = 1, t) = \theta_U(\xi, t) = \sum_{n=-N_M}^{N_M} \Psi_U^{(n)}(t) e^{in\alpha\xi} \tag{2.4.7a,b}$$

Use of (2.4.3) leads to the boundary conditions for each Fourier mode in the form

$$\Psi^{(n)}(-1, t) = \Psi_L^{(n)}(t), \quad \Psi^{(n)}(1, t) = \Psi_U^{(n)}(t). \tag{2.4.8a,b}$$

Introduction of Chebyshev expansion (2.4.6) leads to the required closing conditions in the form

$$\sum_{k=0}^{N_T}(-1)^k F_{k,\tau+1}^{(n)} = \Psi_{L,\tau+1}^{(n)} \quad , \quad \sum_{k=0}^{N_T} F_{k,\tau+1}^{(n)} = \Psi_{U,\tau+1}^{(n)} \qquad (2.4.9a,b)$$

The above process leads to a system of equations with a full matrix of coefficients. Solution of this system moves simulations by one time step forward and this approach has been referred to in the previous section as the direct algorithm. As the matrix is full, no special iterative algorithm can be proposed. All entries corresponding to differential equations (2.4.5a) need to be reconstructed at the each time step while entries corresponding to boundary conditions remain essentially unchanged, which is opposite to what had to be done in the case of the IBC method. This observation suggests that the IBC method could be significantly faster when the cost of construction of the coefficient matrix is significant. This issue will be discussed in the next section.

2.5. Testing of the algorithms

We shall discuss performance of the algorithms in the context of two convenient test problems involving movements of boundaries, i.e., movements of the upper wall corresponding (i) to an elastic traveling wave and (ii) to a elastic standing wave.

2.5.1. Elastic traveling wave

Consider an elastic wave traveling along the upper wall with the lower wall being flat. The shape of the resulting slot can be described as

$$y_U(x,t) = 1+S\cos[\alpha(x-ct)], \qquad y_L=-1. \qquad (2.5.1)$$

where c denotes the phase speed, α denotes the wave number and S stands for the amplitude of the wave. Time variations of the shape of the slot are illustrated in Fig. 2.3.

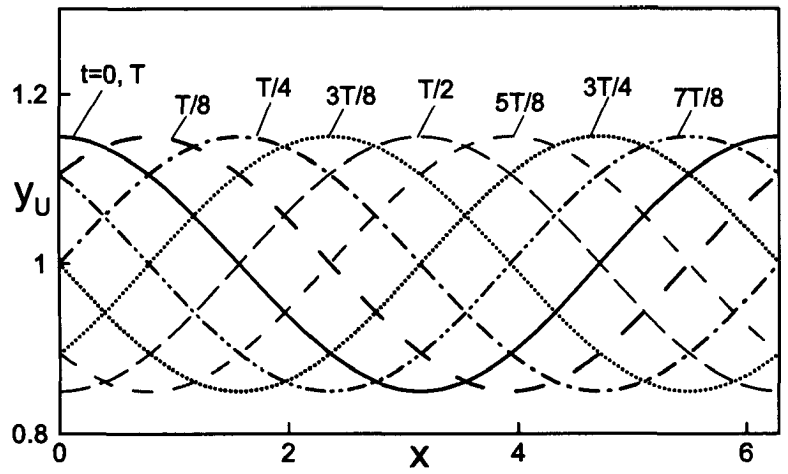


Figure 2.3. *Shape of the upper wall deformed by elastic traveling wave described by Eq.(2.5.1) with the amplitude S=0.15, the wave number α=1.0 and the phase speed c=π at t=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T denotes one time period.*

We wish to determine variations of the temperature in the interior of the slot while keeping the temperature of the upper wall at θ_U=0 and at the lower wall at θ_L=1. Use of the Galilean transformation

$$X=x-ct \tag{2.5.2}$$

transforms the unsteady moving boundary problem into a steady fixed boundary problem with a sinusoidal upper wall. The full problem in the moving frame of reference (X,y) has the form

$$\partial^2\theta/\partial X^2 + c\,\partial\theta/\partial X + \partial^2\theta/\partial y^2 = 0 \tag{2.5.3}$$

with boundary conditions in the form

$$\theta(X,-1)=1, \qquad \theta(y_U(X))=0 \text{ where } y_U=1+S\cos(\alpha X). \tag{2.5.4}$$

Problem (2.5.3)-(2.5.4) is solved using IBC method and its performance can be judged by determining the error of enforcement of boundary conditions at the upper wall as a function of the amplitude and the wave number of the wave. For convenience, we use the L_∞ norm for θ evaluated at the upper wall defined as

$$\left\|\theta_U(X)\right\|_\infty = \sup_{0\le X\le 2\pi/\alpha} \left|\theta(X, y_U(X))\right| \tag{2.5.5}$$

as a quantitative measure of the error. This norm is very strict and thus provides an unambiguous means to judge the accuracy of enforcement of boundary conditions. Dependence of boundary error as a function of geometric parameters for a fixed number of Fourier modes N_M is illustrated in Figs 2.4 and 2.5. The reader may note in judging these results that S=2 corresponds to a situation when the lowest extreme of the upper wall touches the lower wall. The available results suggest that the error is at machine accuracy level if α and S are below certain critical values. Once these values are reached, the error begins to increase rapidly in a fairly universal manner. This critical values of α and S can be increased by increasing the number of Fourier modes N_M used in the calculation, but the qualitative character of the error increase remains unchanged.

Distribution of temperature along the upper wall $\theta_U(X)$ over a single spatial period is displayed in Fig. 2.6. The expected value is zero, however the actual value is different and its magnitude illustrates the error in the enforcement of boundary conditions. This error is oscillatory in X with the maximum occurring around X=0 and λ, i.e., around the upper extreme of the deformed wall. The same problem has been solved directly in the fixed coordinate system as a moving boundary problem and the error variations as a function of time are illustrated in Fig. 2.7. It can be seen that the magnitude and pattern of distribution of the error remain the same as a function of time, however, the pattern shifts in the direction of the positive x-axis with the phase speed of the wave, i.e., its maximum follows the movement of the upper extreme of the wall.

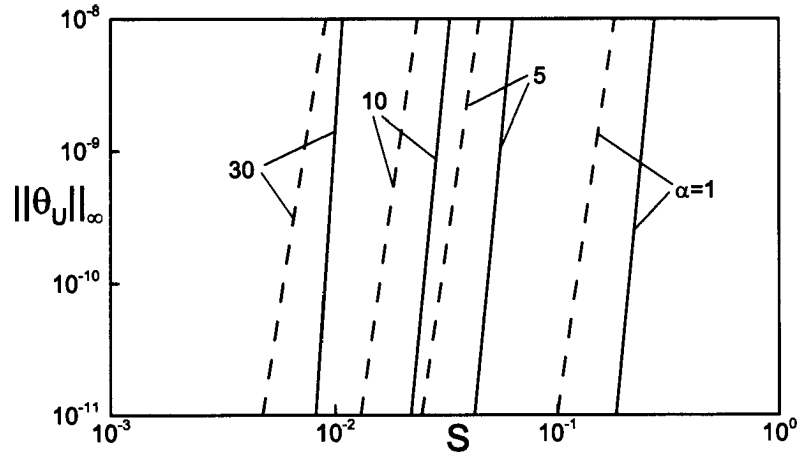


Figure 2.4. *The $\|\theta_U(X,t)\|_\infty$ norm (see Eq. 2.5.5) evaluated using the IBC method as a function of the wave amplitude S for selected values of the wave number α for the model problem (2.5.3)-(2.5.4) with $c=\pi$. The dashed and solid lines represent results obtained with the N_M = 10, 15 Fourier modes, respectively. The reader may note that S=2 corresponds to the trough of the wave reaching the bottom wall. N_T=70 Chebyshev polynomials were used in the calculations.*

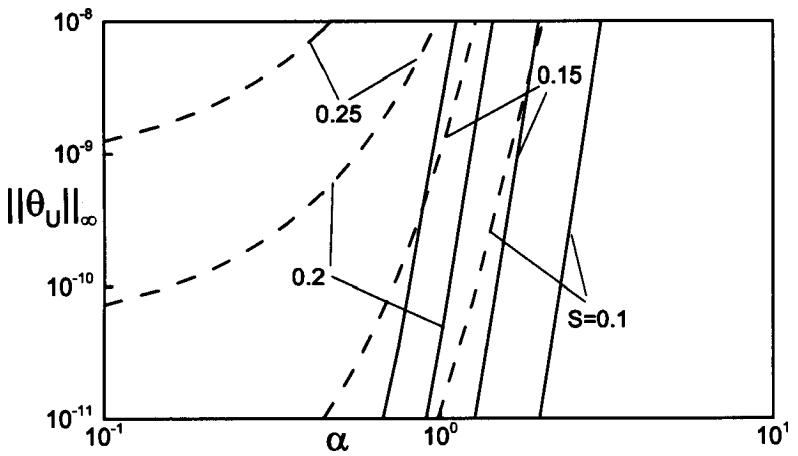


Figure 2.5. *The $\|\theta_U(X,t)\|_\infty$ norm (see Eq. 2.5.5) evaluated using the IBC method as a function of the wave number α for selected values of the wave amplitude S for the model problem (2.5.3)-(2.5.4) for $c=\pi$. The dashed and solid lines represent results obtained with the N_M = 10, 15 Fourier modes, respectively. The reader may note that S=2 corresponds to the trough of the wave reaching the bottom wall. N_T=70 Chebyshev polynomials were used in the calculations.*

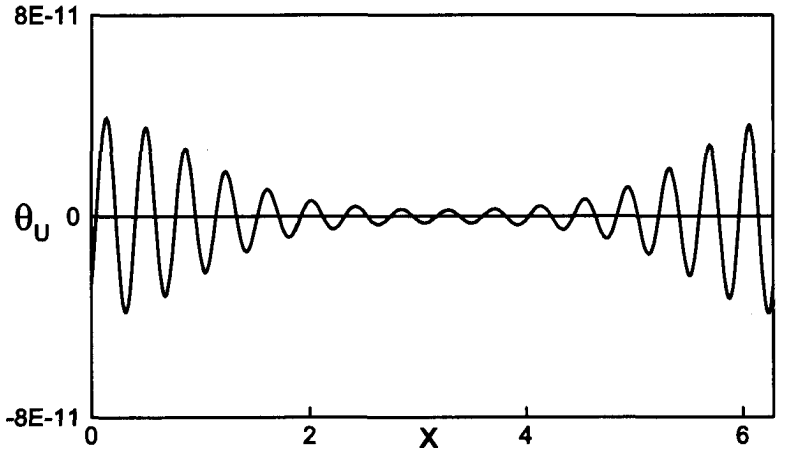


Figure 2.6. *Distribution of temperature at the upper wall θ_U for the model problem (2.5.3)-(2.5.4) with the phase speed $c=\pi$, the amplitude $S=0.2$ and the wave number $\alpha=1$. The presented results were obtained using the IBC method with $N_M=15$ Fourier modes and $N_T=70$ Chebyshev polynomials.*

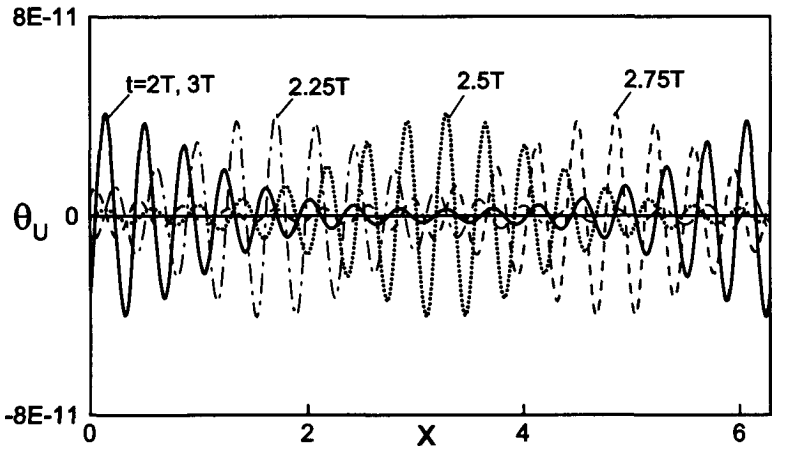


Figure 2.7. *Distribution of temperature at the upper wall θ_U at t = 2T, 2.25T, 2.5T, 2.75T and 3T, where T stand for one time period, for the model problem (2.5.3)-(2.5.4) with $S=0.2$, $\alpha=1.0$ and $c=\pi$. The presented results were obtained through a direct solution of the moving boundary problem in the fixed coordinates system (x,y) using the IBC method with $N_M=15$ Fourier modes and $N_T=70$ Chebyshev polynomials. Solution corresponding to the fixed boundary problem in the moving frame of reference was used as the initial condition. Presented results correspond to the 3rd period after initiations of the calculations.*

Figure 2.8 illustrates variations of the temperature at a few test points as a function of time over five time periods determined using direct solution of (2.5.3)-(2.5.4) in the fixed frame of reference, i.e., solving it as a moving boundary problem. It can be seen that the algorithm faithfully reproduces the oscillatory character of temperature variations and the phase shift between different spatial locations. Figure 2.9 illustrates spatial distribution of the heat flux at the upper wall for the same problem as a function of time over one time period. Changes in the variations of the spatial distributions of the heat flux as a function of time demonstrate a correct phase shift. Figure 2.10 illustrates variations of the maximum error in the evaluation of the temperature as a function of the temporal step size Δt. The character of these variations demonstrates that the algorithm delivers the first- and second-order accuracy for time simulations, as theoretically predicted. Solution of the fixed boundary problem in the moving frame of reference was used as the initial condition for the solution of the moving boundary problem.

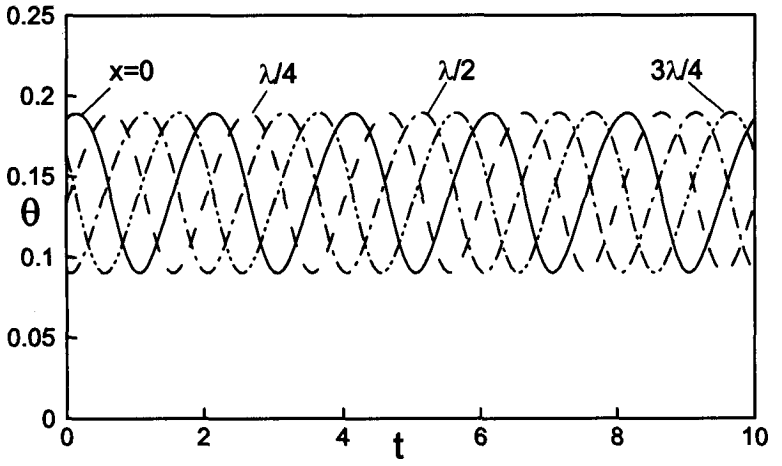


Figure 2.8. *Variations of the temperature θ at points (x,y) = (0, 0.7), (λ/4, 0.7), (λ/2, 0.7), (3λ/4, 0.7) for the model problem (2.5.3)-(2.5.4) with the amplitude S=0.15, the wave number α=1.0 and the phase speed c=π solved directly as a moving boundary problem in the fixed reference frame for five time periods. Solution of the corresponding fixed boundary problem in the moving frame of reference was used as the initial condition. N_M=15 Fourier modes and N_T=70 Chebyshev polynomials were used in the computations.*

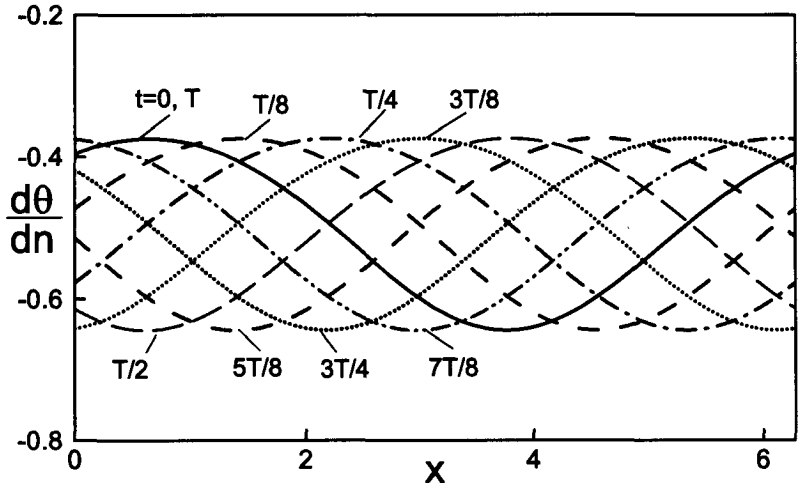


Figure 2.9. *Variations of heat flux distribution at the upper wall at times t=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T stand for one time period. Other conditions as in Fig. 2.8.*

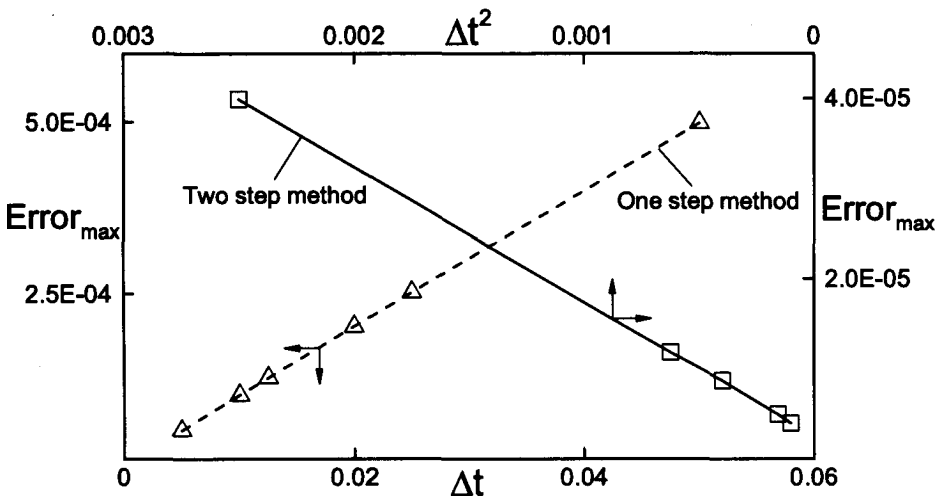


Figure 2.10. *Variations of the maximum error as a function of the step size Δt used in the temporal discretization in the case of the model problem (2.5.3)-(2.5.4) with the amplitude S=0.05, the wave number α=1.0 and the phase speed c=π. Error is defined as the difference between the results obtained through the direct solution of the moving boundary problem in the fixed frame of reference and the corresponding fixed boundary problem in the moving frame of reference at a time corresponding to t=1.0. N_M=10 Fourier modes and N_T=60 Chebyshev polynomials were used in the computations.*

2.5.2. Elastic standing wave

Consider elastic standing wave at the upper wall with the lower wall being flat. The shape of the slot can be described as

$$y_U(x,t) = 1+S\sin(\alpha x)\cos(\omega t), \qquad y_L=-1, \tag{2.5.6}$$

where α denotes the wave number, S stands for the amplitude and ω denotes the frequency of the wave. Time variations of the shape of the slot are illustrated in Fig. 2.11.

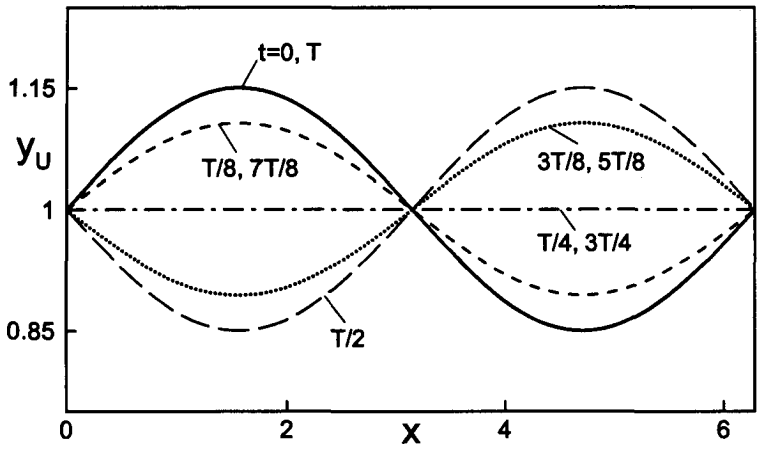


Figure 2.11. *Shape of the upper wall deformed by elastic standing wave described by Eq.(2.5.6) with the wave number α=1.0, the amplitude S=0.15 and the frequency $\omega=\pi$ at times t=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T denotes one time period.*

We wish to determine variations of the temperature in the interior of the slot while keeping the temperature of the upper wall at θ_U=0 and at the lower wall at θ_L=1. The full test problem has the form

$$-\partial\theta/\partial t+\partial^2\theta/\partial x^2+\partial^2\theta/\partial y^2=0 \tag{2.5.7}$$

with boundary conditions in the form

$$\theta(x,-1,t)=1, \qquad \theta(x, y_U(x,t), t)=0 \text{ where } y_U=1+S\sin(\alpha x)\cos(\omega t). \tag{2.5.8}$$

For convenience, we use the L_∞ norm for θ evaluated at the upper wall defined as

$$\left\|\theta_U(x,t)\right\|_\infty = \sup_{0\le X\le 2\pi/\alpha} \left|\theta(x, y_U(x,t), t)\right| \tag{2.5.9}$$

as a quantitative measure of the error.

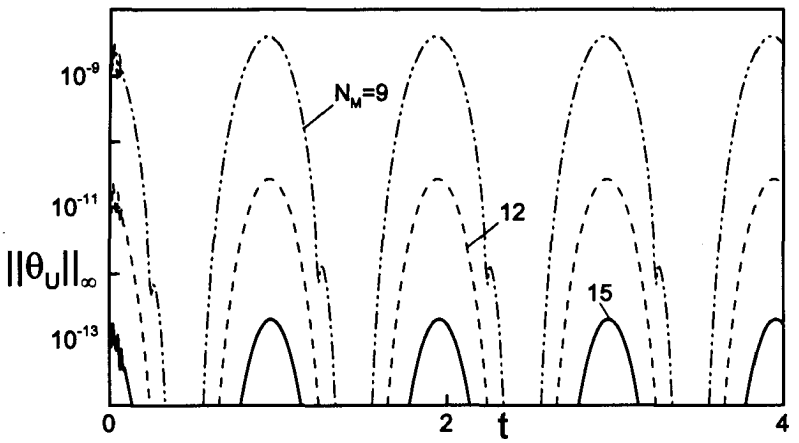


Figure 2.12. *Variation of the $\left\|\theta_U(x,t)\right\|_\infty$ norm as a function of time over two time periods for the model problem (2.5.7)-(2.5.8). N_T=70 Chebyshev polynomials were used and the solid, dash and dotted lines corresponds to N_M=15, 12, 9 Fourier modes. Solution of the fixed boundary problem for the geometry described by (2.5.8) with t=0 was used as an initial conditions. All other conditions are as in Fig. 2.11.*

Figure 2.12 illustrates variations of the above norm over two time periods with the initial conditions corresponding to the solution of a fixed boundary problem with the shape of the boundaries described by (2.5.8) at t=0. It can be seen that the error changes periodically in time with the frequency equal to double of the frequency of the standing wave. The maximum of the error occurs at times when the slot opening reaches maximum. The location of the maximum error follows location of the maximum slot

opening as illustrated in Fig. 2.13. The character of the error remains unchanged as a function of time which points to the robustness of the algorithm. The magnitude of the error can be reduced by increasing the number of Fourier modes used in the calculations but its qualitative features remain the same, as illustrated in Fig. 2.12.

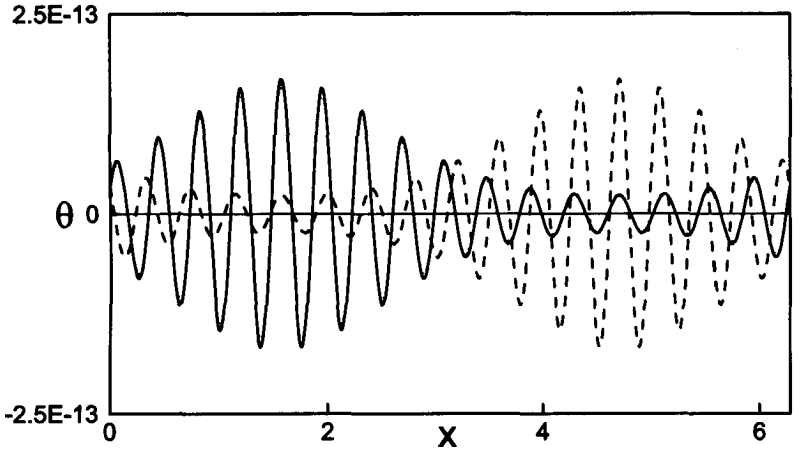


Figure 2.13. *Distribution of temperature of the upper wall θ_U after three (solid line) and three and half time periods (dashed line) for the model problem (2.5.7)-(2.5.8) evaluated using the IBC method with N_M=15 Fourier modes and N_T=70 Chebyshev polynomials. Solution of a fixed boundary problem corresponding to geometry described by Eq.(2.5.8) with t=0 was used as the initial condition. All other conditions are as in Fig. 2.11.*

The IBC algorithm enforces zero conditions for only the first N_M Fourier modes. Behavior of the remaining available Fourier modes (for $N_M+1 < n < N_\theta$) provides an illustration of the error behavior and this is illustrated in Fig. 2.14. The first N_M Fourier modes are indeed zero and the maximum error is associated with the first neglected mode. As the magnitude of the error is related to the size of the slot opening, this error is expected to be the same every half time period and this is indeed the case, as shown in Fig. 2.14. The same figure displays error associated with solving the same problem using different sizes of the "computational box" and thus using different forms of Fourier expansions. It can be seen that the error is associated with different (but easily predicted) terms of the Fourier expansions and its magnitude does not change. It has been observed

that the algorithm does not permit spurious error transfer between different Fourier modes and does not generate any spurious spatial oscillations.

The accuracy of determination of temperature θ in the interior of the slot is dominated by the error associated with the temporal discretization, as the spatial discretization has spectral character and thus is typically much more accurate. Results shown in Fig. 2.15 demonstrate that the algorithm does deliver the first- and second-order temporal accuracy, as theoretically predicted.

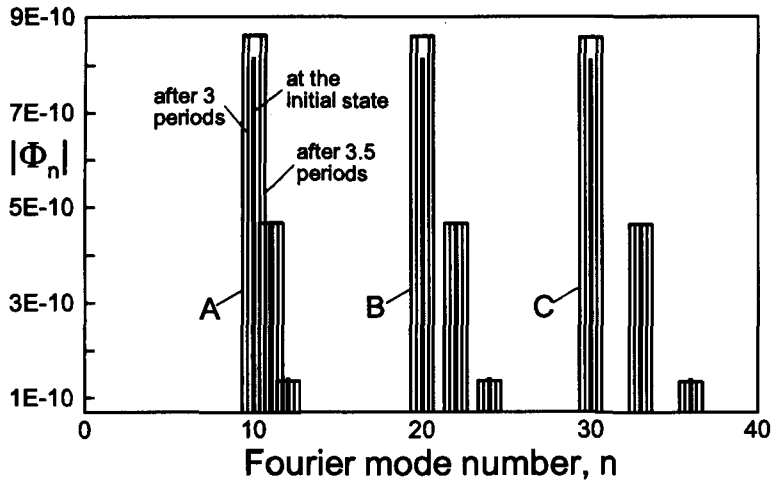


Figure 2.14. *Fourier spectra of temperature distribution at the upper wall θ_U for the model problem (2.5.7)-(2.5.8) with the amplitude S=0.15, the frequency $\omega=\pi$ and the wavelength of the standing wave $\lambda=2\pi$. The initial conditions correspond to the solution of a fixed boundary problem for the geometry given by (2.5.8) with t=0. Three different forms of Fourier expansions were considered, i.e., case A: $\alpha=1.0$, $N_M=9$; case B: $\alpha=0.5$, $N_M=18$ and case C: $\alpha=1/3$, $N_M=27$. $N_T=70$ Chebyshev polynomials used in all cases.*

Figure 2.16 illustrates time variations of the temperature in the interior of the slot. It can be seen that these variations have strictly periodic character, as expected. The instantaneous isotherms inside a slot deformed by an elastic wave with a more complex shape are illustrated in Fig. 2.17. The variations of the heat flux at the upper wall for the same slot are illustrated in Fig. 2.18. These results illustrate the ability of the algorithm

to deal with complex movements of the boundaries that require the use of several Fourier modes.

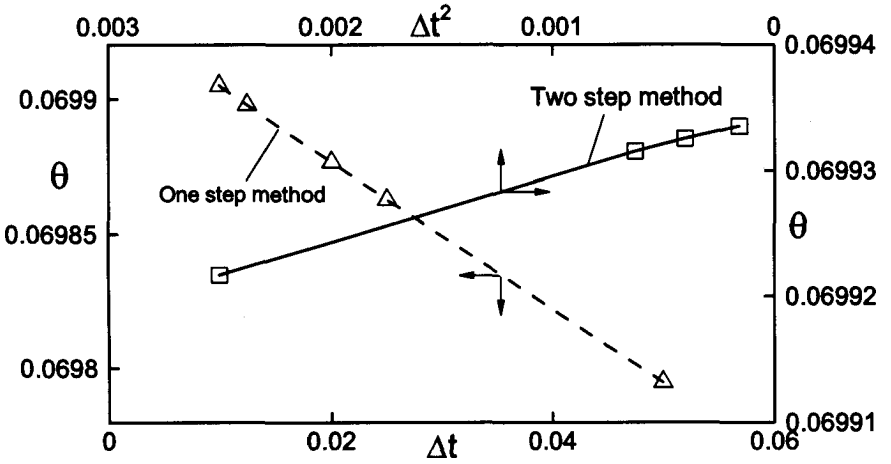


Figure 2.15. *Variations of temperature θ as a function of the step size Δt at a test point (2.x,y)=(3π/2, 0.7) for the model problem (2.5.7)-(2.5.8) with S=0.05, α=1 and ω=π at a time t=T+1.0 where T denotes time period computed with N_M=10 and N_T=60 . The initial conditions correspond to the solution of a fixed boundary problem for the geometry given by (2.5.8) with t=0.*

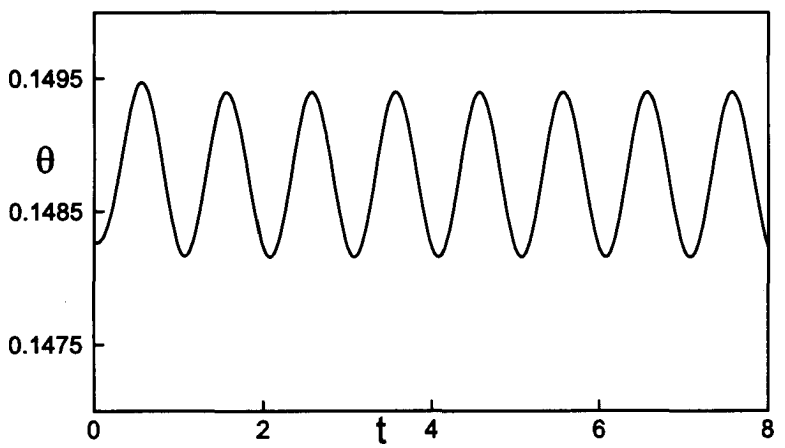


Figure 2.16. *The evolution of θ at a test point (x,y)=(0,0.7) during the first four time periods for the model problem (2.5.7)-(2.5.8) with the wave properties described in Fig. 2.11 computed with N_M=10 and N_T=60. The initial conditions correspond to the solution of a fixed boundary problem for the geometry given by (2.5.8) with t=0.*

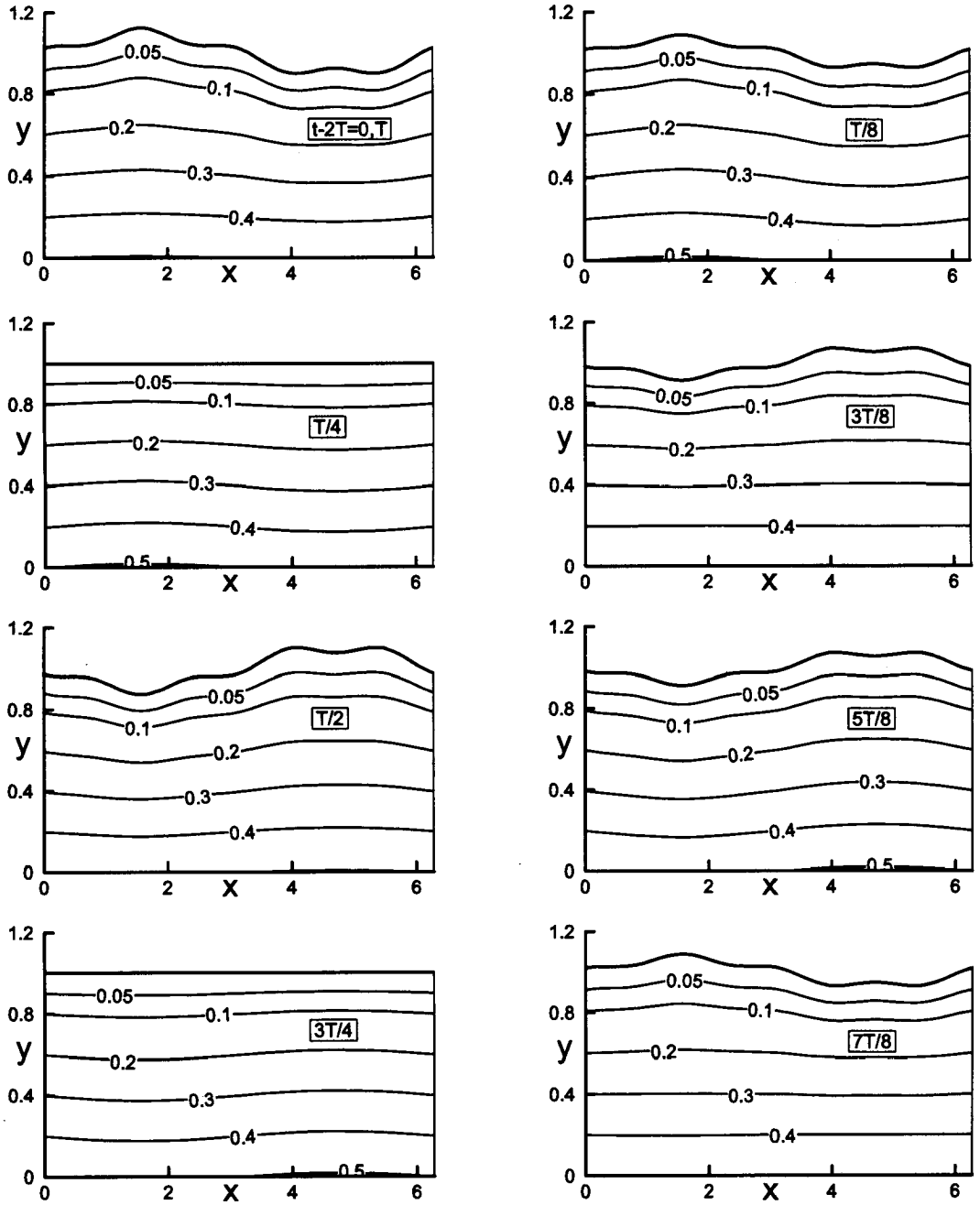


Figure 2.17. *Instantaneous isotherms in the upper part of a slot bounded by $y_L(x,t) = -1$, $y_U(x,t) = 1+(-0.05ie^{ix} + 0.0125e^{4ix} + CC)\cos(\omega t)$ for $\omega=\pi$ using $N_M=15$ and $N_T=70$ evaluated after two time periods at times t-2T=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T stands for one time period and CC is the abbreviation for complex conjugate. The initial conditions correspond to the solution of a fixed boundary problem in a slot with geometry given by the above equation with t=0.*

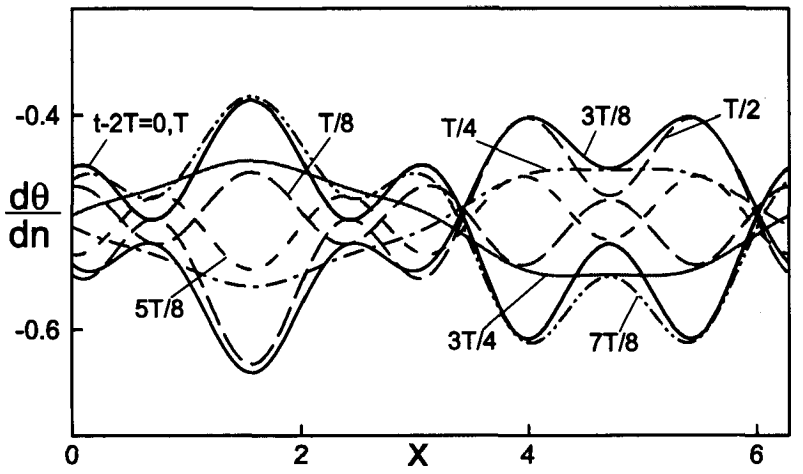


Figure 2.18. *Variations of heat flux distribution at the upper wall in a slot with the flat lower wall $y_L(x,t) = -1$ and a elastic standing wave in the form $y_U(x,t) = 1+(-0.5Sie^{i\alpha x} + 0.125Se^{4i\alpha x} + CC)\cos(\omega t)$ with $S=0.1$, $\alpha=1.0$ and $\omega=\pi$ at the upper wall at times t-2T=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T stands for one time period and CC is the abbreviation for complex conjugate. The initial conditions correspond to the solution of a fixed boundary problem for the geometry corresponding to the shape of the slot at time t=0. The results were determined using $N_M=15$ Fourier modes and $N_T=70$ Chebyshev polynomials.*

2.5.3. Efficiency of the algorithm

Efficiency of the algorithm is dictated by the cost of construction of the matrix of coefficients and solution of the resulting system of linear equations. Both actions have to be repeated at each step of the time-dependent solution process. All tests referred to in the discussion below had been carried out in the MATLAB computing environment using the same hardware.

Structure of the matrix in the case of IBC method has been discussed in Section 2.3. Entries corresponding to the field equations need to be computed only once at the very beginning of the computations as they remain unchanged. Entries corresponding to the boundary conditions need to be re-computed at each time step as they change due to the

movement of the boundaries. In contrary, all entries need to be recomputed at each time step in the DT method. In addition, the cost of a single construction of the coefficient matrix is significantly higher in the case of the DT method due to the presence of numerous and complex coefficients in the equations. The relative advantage of the IBC method is illustrated by noting that in the case of the model problem (2.5.7)-(2.5.8) with N_M=15 Fourier modes and N_T=70 Chebyshev polynomials it took 592 seconds to construct a complete matrix in the case of DT method but only about 8 seconds in the IBC method. The latter is significantly reduced at the next time step, while the former has to be entirely repeated, which adds to the relative advantage of the IBC method. For this reason the development of the DT method has not been pursued.

Table 2.1. *Time requirements for the direct and various iterative/decoupled schemes for the IBC algorithm for the model problem (5.7)-(5.8) with S=0.1, α=1.0, $\omega=\pi$ and N_T=70. Computations have been carried out using two-step implicit method with Δt=0.01.*

Number of Fourier modes, N_M	Time required for 100 time steps (in seconds)			
	Direct algorithm	Decoupled algorithm		
		Version A	Version B	Version C
10	299.9	34.8	32.9	29.2
15	880.7	86.6	84.3	83.2

Once constructed, the solution of the system of linear equations can be carried out using standard methods resulting in a direct algorithm, as discussed in Section 2.3. Since the coefficient matrix in the DT method is full, no savings associated with special features of the matrix can be expected. Matrix in the case of IBC method has a special structure and this provides opportunity to use iterative algorithm. The performance of different iterative/decoupled schemes is illustrated in Table 2.1 and compared with the direct algorithm. Since the overall accuracy of the time-dependent solution is determined by the error of the temporal discretization, the convergence criterion for the decoupled algorithms in these tests had been set at 10^{-5} in order to make it comparable with the error in time. The performance of the decoupled algorithm can be improved by taking advantage of the information available at the previous time step and re-arranging the iteration strategy. Three different strategies have been tested. Version A has been

described in Section 2.3, version B uses extrapolated values of the modal functions during the first iteration and continues iterations until convergence criteria are met and version C uses only extrapolated values of the modal functions and eliminates iterations. In all cases the method remains second-order accurate as the extrapolation provides second-order accuracy in time. The results (see Table 2.1) show significant savings resulting from the use of iterative/decoupled algorithms as compared with the direct algorithm. The distinction between different iterative algorithms and their limitations are illustrated in Table 2.2. The cost of the direct algorithm remains the same regardless of the magnitude of amplitude S of the standing wave. The cost of version A of the iterative algorithm increases as the amplitude S increase and, for sufficiently high value of S the algorithm diverges. Versions B and C behave in a similar manner but diverge for smaller values of S. The relative performances of different versions of the iterative algorithm are very similar, and they all require less computer time as compared with the direct algorithm by an order of magnitude. The direct algorithm however provides access to much larger amplitudes of boundary motions.

Table 2.2. *Time requirement for the direct and various iterative/decoupled schemes for the IBC algorithm for the model problem (5.7)-(5.8) with* $\alpha=1.0$, $\omega=\pi$, $N_M=15$ *and* $N_T=70$ *for various level of wave amplitude. All other conditions are same as Table 2.1.*

Wave amplitude, S	Time required for 100 time steps (in seconds)			
	Direct algorithm	Decoupled algorithm		
		Version A	Version B	Version C
0.05	880.0	84.6	83.2	83.1
0.10	880.7	86.6	84.3	83.2
0.15	879.7	94.4	88.4	*
0.20	880.2	104.4	110.3	*
0.25	880.2	156.9	*	*
0.30	880.3	*	*	*
0.35	880.5	*	*	*

* *indicates failure for the decoupled algorithm to satisfy the required convergence criteria*

2.6. Conclusions

We have developed a fully-implicit, spectrally-accurate algorithm for moving boundary problems using immersed boundary conditions concept. Computations are carried out using a fixed computational domain where the field equations are discretized using a combination of Fourier and Chebyshev expansions. Boundary of the physical domain moves through the computational domain and its location is tracked using the concept of immersed boundary conditions, i.e., the boundary conditions are replaced by constraints that enforce physical conditions along lines internal to the solution domain. The algorithm provides sharp resolution of the location of the moving boundary similar to the case of algorithms based on mapping methods where one of the coordinate lines overlap with the moving boundary. The computational cost of the algorithm is significantly smaller as compared with the mapping-based algorithms because the part of the coefficient matrix that corresponds to the field equations needs to be constructed only once and only entries corresponding to boundary conditions need to be recomputed at each time step. The coefficient matrix has special structure that provides opportunities for the use of much more efficient iterative solution methods. These methods can reduce the overall computational cost by an order of magnitude as compared to the direct method, however, they limit the ability of the algorithm to handle large-amplitude motions of the boundaries. Such algorithms remain of interest nevertheless as they can be easily parallelized and require less memory.

2.7. References

[1] Floryan, J.M. and Rasmussen, H., Numerical analysis of viscous flows with free surfaces, *Appl. Mech. Rev.*, vol. 42, pp. 323-341 (1989).

[2] Glimm, J., Grove, J.W., Li, X.L., Shyue, K.M., Zeng, Y. and Zhang, Q., Three-Dimensional Front Tracking, *SIAM J. Sci. Computing,* vol. 19, pp. 703-727 (1998).

[3] Harlow, F.H. and Welch, J.E., Numerical Study of Large Amplitude Free Surface Motions, *Phys.Fluids*, vol. 9, pp. 842-851 (1966).

[4] Hirt, C.W. and Nichols, B.D., Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *J.Comp.Phys.*, vol. 39, pp. 201-225 (1981).

[5] Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for unsteady flow problems described by the Laplace operator, *Int. J. Num. Meth. Fluids*, vol. 56, pp. 1765-1786 (2007).

[6] Hyman, J.M., Numerical Methods for Tracking of Interfaces, *Physica,* vol. 12D, pp. 396-407 (1984).

[7] Kogut, L., The Influence of Surface Topography on the Electromechanical Characteristics of Parallel-Plate Capacitors, *J. Micromech. Microeng.*, vol. 15, pp. 1068-1075 (2005).

[8] Mittal, R. and Iaccarino, G., Immersed boundary methods, *Annu. Rev. Fluid Mech.,* vol. 37 pp. 239-261 (2005).

[9] Osher, S.J. and Sethian, J.A., Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *J. Comp .Phys.*, vol. 79, pp. 12-49 (1988).

[10] Peskin, C.S., The fluid dynamics of heart valves: experimental, theoretical and computational methods, *Annu. Rev. Fluid Mech.,* vol. 14 pp. 235-59 (1982).

[11] Szumbarski, J. and Floryan, J.M., A direct spectral method for determination of flows over corrugated boundaries, *J. Comp. Phys.*, vol. 153, pp. 378-402 (1999).

[12] Scardovelli, R. and Zaleski, S., Direct Numerical Simulation of Free Surface and Interfacial Flow, *Annu Rev. Fluid Mech.*, vol. 31, pp. 567-603 (1999).

[13] Osher, S.J. and Sethian, J.A., Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *J. Comp .Phys.*, vol. 79, pp. 12-49 (1988).

[14] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., Spectral methods in fluid dynamics, *Springer* (1987).

[15] Peyret, R., Spectral methods for incompressible viscous flow, *Springer* (2002).

[16] Boyd, J.P., Chebyshev and Fourier spectral methods, *Dover Publishers* (2001).

CHAPTER 3

Moving boundary problems described by the biharmonic operator[1]

3.1. Introduction

The first step in the most common approach used for simulations of flows in complex geometries involves numerical modeling of geometry of the flow domain. A model of the domain is constructed using grid generator and flow boundary conditions are imposed on the edges of this domain, i.e., the flow domain overlaps with the computational domain. This process, which is frequently referred to as pre-processing, could consume majority of time required for preparation of the simulation. The flow boundary conditions are enforced exactly and one is concerned with various errors associated with spatial and temporal discretization of the field equations. Complications occur when geometry of the flow domain changes as a function of time, e.g., when one deals with the moving boundary problems. In the present work we are interested in finding a spectrally-accurate and computationally efficient solution to such problem while maintaining a sharp resolution of the location of the moving boundary.

Techniques developed for handling of the moving boundary problems have been reviewed in [1] with updates given in [2]. Such techniques can be classified as Eulerian, Lagrangian and mixed. Eulerian algorithms, which are of interest in the present work, rely on a coordinate system that is either stationary in a laboratory frame of reference or moves in a prescribed manner (i.e., describable by the Galilean transformation). These

---

[1]A version of this chapter has been published as –

Husain, S.Z., and Floryan, J.M., Gridless spectral algorithm for Stokes flow with moving boundaries, *Computer Methods in Applied Mechanics and Engineering*, vol. 198, pp. 245-259 (2008).

algorithms can be divided for convenience into fixed grid methods, adaptive grid methods and various mapping methods.

The adaptive grid methods adjust the grid at each time step so that one of the grid lines follows the moving boundary. The adjustment is done numerically through appropriate mappings. The process is conceptually identical regardless of whether one uses numerical grid generation combined with finite-difference discretization of the field equations or a grid generation scheme for the finite-element discretization of the same equations. These methods have high computational overhead as the grid needs to be reconstructed at each time step followed by solution of the field equations. In the case of the problem discussed in [3] and solved using finite volume discretization, the cost of grid construction reached the level of about 75% of the total cost of the computation. The boundary conditions are imposed exactly and thus the overall error includes contributions from the grid generation and the spatial and temporal discretizations of the field equations. Numerous problems occur when one is interested in high accuracy as this requires both high accuracy in grid generation and high accuracy discretization.

One can reduce the cost of computations and improve accuracy by using analytical transformations that map the irregular flow domain into a regular computational domain at the cost of increased complexity of the field equations (see [4] for an example). Such mappings are available for a limited class of geometries [1].

In the fixed grid methods the boundary of the flow domain is allowed to move through a fixed grid and this eliminates the high computational cost of grid re-adjustment. Various methods for modeling of movements of boundaries based either on the surface or volume tracking procedures are discussed in [1,5]. In the former approach a set of points is introduced to mark location of the boundary which is represented as a set of interpolated curves. These points are moved in a prescribed manner during the solution process [5,6] and provide ability for precise identification of boundary location. In the latter approach the information about the boundary location is not stored but the boundary is reconstructed whenever necessary on the basis of the presence of certain quantity of a

convenient marker within computational cell, e.g., MAC - Marker and Cell [7], VOF - Volume of Fluid [2, 8] and Level Set [9, 10] methods. The reconstruction process leads to a diffused location of the boundary but the overall accuracy can be made consistent with the low accuracy discretization schemes used for the field equations.

Lagrangian methods are characterized by a coordinate system that moves with the fluid. Each computational cell always contains the same fluid and its tracking requires solution of an initial value problem. These methods are well suited for moving boundary problems as they permit simple delineation of moving boundaries. The two main problems involve mesh tangling and loss of numerical accuracy associated with highly distorted meshes. Possible remedies are discussed in [1]. Mixed Lagrangian-Eulerian methods rely on the combination of concepts described above [1].

The available methods have low order accuracy and thus one is interested in the development of techniques that could overcome this limitation. The overall error is affected by the accuracy of representation of boundary location and by the accuracy associated with spatial discretization of the field equations. Spectral methods provide lowest error for the field equations but are limited to solution domains with regular geometries. The variability and complexity of geometry of the flow domain represents the main challenge for implementation of these methods, however, the use of the concept of immersed boundary conditions provides a way to combine both, i.e., the fixed, regular solution domain with a complex, time dependent flow domain [11]. The immersed boundary conditions concept has been proposed in [12] in the context of simulation of cardiac dynamics. Boundary conditions at the edges of the physical domain are replaced by constraints imposed inside the computational domain but these constraints have been constructed using physical arguments through introduction of forcing functions that reproduce the effect of the boundary [13]. In this sense the method is analogous to the fixed grid methods discussed above. The first spectrally accurate implementation for fixed boundary problems has been given in [11] with the constraints constructed in a formal manner by imposing conditions for elimination of certain terms in the spectral expansions representing flow boundary conditions. This implementation is limited to

geometries that can be represented by Fourier expansions but results in a gridless algorithm as all possible variations of boundary geometries are described in terms of the Fourier coefficients only. The programming effort associated with modeling of changes of geometry has been essentially removed as the only information required for specifying the new geometry is reduced to a set of Fourier coefficients provided as input to the code. This implementation has been extended to hydrodynamics stability problems [14] and applied successfully in the analysis of the effect of surface roughness [15].

This chapter is focused on the development of spectrally-accurate algorithms suitable for solution of time-dependent flow problems, including moving boundary problems. Such algorithms are of interest in the development of active flow control and management techniques using micro-electro-mechanical devices. As our interest is in flows through micro/nano channels, we focus our attention on the small Reynolds number limit, i.e., Stokes flow. Simpler problems, i.e., unsteady conduction problems in a corrugated channels have been considered earlier in [16, 17] without encountering numerical instabilities. The model problem for the present work is described in Section 3.2. Spatial and temporal discretizations are discussed in Section 3.3. Results of numerical tests are discussed in Section 3.4. Section 3.5 provides a short summary of the main conclusions.

3.2. Model problem

The model problem consists of an unsteady Stokes flow in a channel bounded by corrugated walls with geometry of the corrugation changing as a function of time (see Fig. 3.1). These geometries as well as motions of the walls are described by the following relations

$$y_L(x,t) = -1 + \sum_{n=-\infty}^{n=+\infty} H_L^{(n)}(t)\, e^{in\alpha x}, \qquad y_U(x,t) = 1 + \sum_{n=-\infty}^{n=+\infty} H_U^{(n)}(t) e^{in\alpha x} \qquad (3.2.1a,b)$$

where $H_L^{(n)} = H_L^{(-n)^*}$ and $H_U^{(n)} = H_U^{(-n)^*}$ are known and stars denote complex conjugates. The channel geometry is periodic in x with wavelength $\lambda=2\pi/\alpha$ and extends to $\pm\infty$ in the x-direction.

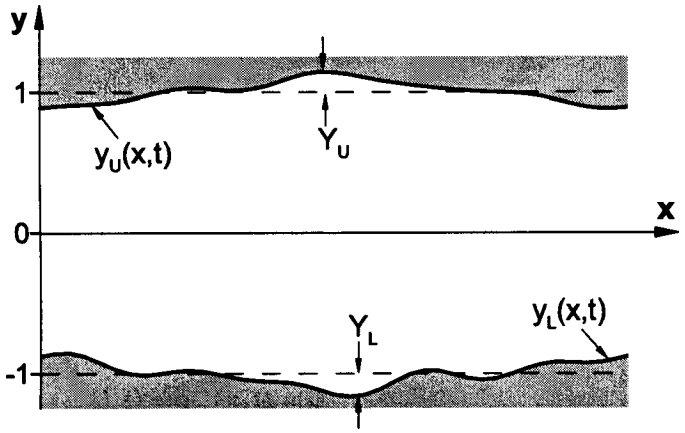


Figure 3.1. *Sketch of the instantaneous form of the flow domain.*

The dimensionless field equation has the form

$$\frac{\partial}{\partial t}\left(\nabla^2\Psi\right)=\nabla^2\left(\nabla^2\Psi\right). \tag{3.2.2a}$$

where Ψ denotes the stream function, which is defined as

$$u(x,y,t) = u_0(y) + u_1(x,y,t) = d\Psi_0/dy + \partial\Psi/\partial y = \partial\Psi_T/\partial y\,, \tag{3.2.2b}$$

$$v(x,y,t) = v_1(x,y,t) = -\partial\Psi/\partial x = -\partial\Psi_T/\partial x \tag{3.2.2c}$$

and $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$. Here u(x,y,t) and v(x,y,t) denote the total velocities in x- and y-directions, respectively, $u_1(x,y,t)$ and $v_1(x,y,t)$ denote velocity modifications due to the presence of boundary motion, the Poiseuille flow $u_0(y)=1-y^2$ is taken as the reference flow, $\Psi_0=-y^3/3+y+2/3$ denotes the stream function of this flow and Ψ_T stand for the

stream function of the complete flow (total stream function). The problem formulation needs to be supplemented with suitable initial conditions, which are taken to be in the form

$$u(x,y,0) = u_i(x,y),\ v(x,y,0) = v_i(x,y),\ y_L(x,0)=y_{Li}(x),\ y_U(x,0)=y_{Ui}(x), \qquad (3.2.3a\text{-}d)$$

where $u_i(x,y)$, $v_i(x,y)$, $y_{Li}(x)$, $y_{Ui}(x)$ are considered to be known, and boundary conditions at the solid walls, which are taken to be in the form

$$u_0(y_L(x,t)) + u_1(x, y_L(x,t), t) = u_L(x,t) = 0, \qquad (3.2.4a)$$

$$u_0(y_U(x,t)) + u_1(x, y_U(x,t), t) = u_U(x,t) = 0, \qquad (3.2.4b)$$

$$v_1(x, y_L(x,t), t) = v_L(x,t) = dy_L/dt = \sum_{n=-\infty}^{n=+\infty} \frac{dH_L^{(n)}}{dt} e^{in\alpha x}, \qquad (3.2.4c)$$

$$v_1(x, y_U(x,t), t) = v_U(x,t) = dy_U/dt = \sum_{n=-\infty}^{n=+\infty} \frac{dH_U^{(n)}}{dt} e^{in\alpha x}. \qquad (3.2.4d)$$

3.3. Discretization method

We are interested in the determination of solution of (3.2.2)-(3.2.4) with spectral accuracy. The main difficulty associated with the implementation of spectral discretization arises due to the irregularity and time-dependence of the solution domain. In order to overcome this problem, we select fixed rectangular computational domain extending over one period in the x-direction and extending sufficiently far in the y-direction so that the flow domain remains always immersed inside the computational domain during the time interval of interest. If we denote the locations of extremities in

the shape of the walls as Y_U and Y_L, the y-extent of the computational domain is set as (-1-Y_L,1+Y_U) without loss of generality.

The spatial discretization is based on the use of Fourier expansions in the x-direction due to periodicity of the geometry, and on expansions in terms of Chebyshev polynomials in the y-direction. We shall use standard definition of the Chebyshev polynomials and thus the y-extent of the computational domain needs to be mapped onto (-1,1) (see Fig. 3.1) before calculations can proceed. The required mapping has the form

$$\hat{y} = [y - (1 + Y_U)]\,\Gamma + 1 \tag{3.3.1}$$

where $\hat{y} \in \langle -1,1 \rangle$, $\Gamma = 2/(2+Y_U+Y_L)$ is a constant and the governing equation transforms into

$$\frac{\partial}{\partial t}\left[\frac{\partial^2}{\partial x^2} + \Gamma^2 \frac{\partial^2}{\partial \hat{y}^2}\right]\Psi = \left[\frac{\partial^4}{\partial x^4} + 2\Gamma^2 \frac{\partial^2}{\partial x^2 \partial \hat{y}^2} + \Gamma^4 \frac{\partial^4}{\partial \hat{y}^4}\right]\Psi . \tag{3.3.2}$$

Locations of the walls in the $(x, \hat{y})$ plane are given as

$$\hat{y}_L(x,t) = \sum_{n=-\infty}^{n=+\infty} A_L^{(n)}(t) e^{in\alpha x}, \qquad \hat{y}_U(x,t) = \sum_{n=-\infty}^{n=+\infty} A_U^{(n)}(t) e^{in\alpha x}, \tag{3.3.3a,b}$$

where $A_L^{(0)}(t) = 1 + \Gamma\left[-2 - Y_U + H_L^{(0)}(t)\right]$, $A_L^{(n)}(t) = \Gamma H_L^{(n)}(t)$ for $n \neq 0$, $A_U^{(0)}(t) = 1 + \Gamma\left[-Y_U + H_U^{(0)}(t)\right]$, $A_U^{(n)}(t) = \Gamma H_U^{(n)}(t)$ for $n \neq 0$. The boundary conditions at the transformed boundaries become

$$u_0(\hat{y}_L(x,t)) + u_1(x, \hat{y}_L(x,t), t) = 0, \tag{3.3.4a}$$

$$u_0(\hat{y}_U(x,t)) + u_1(x, \hat{y}_U(x,t), t) = 0, \tag{3.3.4b}$$

$$v_l(x,\hat{y}_L(x,t),t) = v_L(x,t) = \Gamma^{-1}\, d\hat{y}_L/dt = \Gamma^{-1}\sum_{n=-\infty}^{n=+\infty} \frac{dA_L^{(n)}}{dt} e^{in\alpha x}\ , \tag{3.3.4c}$$

$$v_l(x,\hat{y}_U(x,t),t) = v_U(x,t) = \Gamma^{-1}\, d\hat{y}_U/dt = \Gamma^{-1}\sum_{n=-\infty}^{n=+\infty} \frac{dA_U^{(n)}}{dt} e^{in\alpha x} \tag{3.3.4d}$$

The solution can be represented in the form of Fourier expansion

$$\Psi(x,\hat{y},t) = \sum_{n=-\infty}^{n=+\infty} \Phi^{(n)}(\hat{y},t) e^{in\alpha x} \approx \sum_{n=-N_M}^{n=+N_M} \Phi^{(n)}(\hat{y},t) e^{in\alpha x} \tag{3.3.5}$$

where $\Phi^{(n)}(\hat{y},t) = \Phi^{(-n)^*}(\hat{y},t)$ and star denotes complex conjugate. Substitution of (3.3.5) into the field equation (3.3.2) and separation of Fourier components lead to an uncoupled system of partial differential equations for $\Phi^{(n)}$, $n \in \langle 0, N_M \rangle$, of the type

$$\frac{\partial}{\partial t}\left[\Gamma^2 D^2 - (n\alpha)^2\right]\Phi^{(n)} = \left[\Gamma^4 D^4 - 2\Gamma^2 (n\alpha)^2 D^2 + (n\alpha)^4\right]\Phi^{(n)} \tag{3.3.6}$$

where D=d/d$\hat{y}$. Two types of temporal discretizations have been used. The two-step implicit method results in the following relations

$$\begin{aligned}&\left[\Gamma^4 D^4 + \left(-2n^2\alpha^2\Gamma^2 - 1.5\Gamma^2\Delta t^{-1}\right)D^2 + \left(n^4\alpha^4 + 1.5n^2\alpha^2\Delta t^{-1}\right)\right]\Phi_{\tau+1}^{(n)} = \\ &-2\Delta t^{-1}\left[\Gamma^2 D^2 - (n\alpha)^2\right]\Phi_{\tau}^{(n)} + 0.5\Delta t^{-1}\left[\Gamma^2 D^2 - (n\alpha)^2\right]\Phi_{\tau-1}^{(n)}, \quad n \in \langle 0, N_M \rangle\end{aligned} \tag{3.3.7a}$$

while similar relations resulting from the one-step, self-starting implicit method is

$$\begin{aligned}&\left[\Gamma^4 D^4 + \left(-2(n\alpha)^2\Gamma^2 - 0.5\Gamma^2\Delta t^{-1}\right)D^2 + \left((n\alpha)^4 + 0.5(n\alpha)^2\Delta t^{-1}\right)\right]\Phi_{\tau+1}^{(n)} = \\ &\qquad -0.5\Delta t^{-1}\left[\Gamma^2 D^2 - (n\alpha)^2\right]\Phi_{\tau}^{(n)}, \qquad n \in \langle 0, N_M \rangle\end{aligned} \tag{3.3.7b}$$

In the above, the subscript τ labels the time step and Δt stands for the (constant) length of the time step. Relation (3.3.7a) has the form of an inhomogeneous ordinary differential equation for $\Phi^{(n)}_{\tau+1}$. The following discussion will be carried out in the context of the two-step method, while the relevant relations for the one-step method can be readily deduced.

The unknown function $\Phi^{(n)}_{\tau+1}$ can be represented in terms of expansions based on Chebyshev polynomials in the form

$$\Phi^{(n)}_{\tau+1}(\hat{y}) = \sum_{k=0}^{k=\infty} G^{(n)}_{k,\tau+1} T_k(\hat{y}) \approx \sum_{k=0}^{k=N_T} G^{(n)}_{k,\tau+1} T_k(\hat{y}) \tag{3.3.8}$$

where T_k denotes the Chebyshev polynomial of k^{th} order and $G^{(n)}_{k,\tau+1}$ denotes the unknown coefficients of the expansion. Substitution of (3.3.8) into (3.3.7a) gives

$$\begin{aligned}&\left[\Gamma^4 D^4 + \left(-2n^2\alpha^2\Gamma^2 - 1.5\Gamma^2\Delta t^{-1}\right) D^2 + \left(n^4\alpha^4 + 1.5n^2\alpha^2\Delta t^{-1}\right)\right] \sum_{k=0}^{k=N_T} G^{(n)}_{k,\tau+1} T_k = \\ &-2\Delta t^{-1}\left[\Gamma^2 D^2 - (n\alpha)^2\right] \sum_{k=0}^{k=N_T} G^{(n)}_{k,\tau} T_k + 0.5\Delta t^{-1}\left[\Gamma^2 D^2 - (n\alpha)^2\right] \sum_{k=0}^{k=N_T} G^{(n)}_{k,\tau-1} T_k \,.\end{aligned} \tag{3.3.9}$$

Galerkin procedure is used to develop algebraic equations for the unknowns $G^{(n)}_{k,\tau+1}$, i.e., we multiply both sides of (3.3.9) by $T_j(\hat{y})$ and integrate with the weight function $\hat{\omega} = 1/\sqrt{1-\hat{y}^2}$ to get

$$\begin{aligned}&\sum_{k=0}^{k=N_T} \left[\Gamma^4 \left\langle T_j, D^4 T_k \right\rangle + \left(-2n^2\alpha^2\Gamma^2 - 1.5\Gamma^2\Delta t^{-1}\right)\left\langle T_j, D^2 T_k \right\rangle + \left(n^4\alpha^4 + 1.5n^2\alpha^2\Delta t^{-1}\right)\left\langle T_j, T_k \right\rangle\right] G^{(n)}_{k,\tau+1} \\ &= \sum_{k=0}^{k=N_T} \left[\Gamma^2 \left\langle T_j, D^2 T_k \right\rangle - (n\alpha n^2 \left\langle T_j, T_k \right\rangle\right] \left[-2\Delta t^{-1} G^{(n)}_{k,\tau} + 0.5\Delta t^{-1} G^{(n)}_{k,\tau-1}\right] \qquad j \in \langle 0, N_T - 4 \rangle\end{aligned} \tag{3.3.10}$$

where the inner product is defined as $\left\langle f_j(\hat{y}), g_k(\hat{y}) \right\rangle = \int_{-1}^{1} f_j(\hat{y}) g_k(\hat{y}) \hat{\omega}(\hat{y}) d\hat{y}$. Equation (3.3.10) leads to N_T-3 decoupled algebraic equations for each Fourier mode. Four additional equations required in order to close the system need to be derived from the boundary conditions and these conditions provide coupling between different Fourier modes.

The boundary conditions are to be enforced along the lines $\hat{y}_L(x, \tau+1)$ and $\hat{y}_U(x, \tau+1)$ that are inside the computational domain while the solution domain remains fixed at $\hat{y} \in \langle -1,1 \rangle$. This can be accomplished using the concept of immersed boundary conditions. We begin description of the implementation of boundary conditions by noting that at time τ+1 one needs to evaluate velocity components $u_f(x,\tau+1) \equiv u(x, f(x,\tau+1),\tau+1)$ and $v_f(x,\tau+1) \equiv v(x, f(x,\tau+1),\tau+1)$ along an arbitrary time dependent line l that at time $t = \tau+1$ occupies position $\hat{y} = f(x,\tau+1)$, such that f is a periodic function with period $\lambda = 2\pi/\alpha$ and $|f(x,\tau+1)| \leq 1$. The function $f(x,\tau+1)$ can be expressed without loss of generality as

$$f(x,\tau+1) = \sum_{n=-\infty}^{n=-\infty} P_{\tau+1}^{(n)} e^{in\alpha x} \approx \sum_{n=-N_A}^{n=+N_A} P_{\tau+1}^{(n)} e^{in\alpha x} \qquad (3.3.11)$$

where in the calculations the shape of the wall is approximated by the finite number of terms N_A. The velocity components $u_f(x,\tau+1)$ and $v_f(x,\tau+1)$ are periodic in x with the same period λ and thus can be expressed in terms of Fourier series as

$$u_f(x,\tau+1) \equiv u(x,f(x,\tau+1),\tau+1) = \sum_{n=-N_\theta}^{n=+N_\theta} U_{\tau+1}^{(n)} e^{in\alpha x},$$

$$v_f(x,\tau+1) \equiv v(x,f(x,\tau+1),\tau+1) = \sum_{n=-N_\theta}^{n=+N_\theta} V_{\tau+1}^{(n)} e^{in\alpha x}. \qquad (3.3.12a,b)$$

It will become obvious from the follow up discussion that the summation extends to $N_\theta=N_TN_A+N_M$. When $f(x,\tau+1)$ overlaps with the position of the wall, u_l and v_l are known and the coefficients in the above expansions can be determined. Since the flow representation used in the computations is limited to N_M+1 modes (see Eq. 3.3.5), only on the first (N_M+1) terms in (3.3.12) can be accounted for. We shall now discuss the method used to enforce these conditions.

Velocity components u_l and v_l can be evaluated along the wall at time $\tau+1$ using the discretized form of the solution, i.e.,

$$u_l(x,\tau+1) = u_0(f(x,\tau+1)) + \Gamma \sum_{n=-N_M}^{n=+N_M} D\Phi^{(n)}(f(x,\tau+1),\tau+1)e^{in\alpha x}$$

$$= u_0(f(x,\tau+1)) + \Gamma \sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} G^{(n)}_{k,\tau+1} DT_k(f(x,\tau+1))\, e^{in\alpha x} \quad , \qquad (3.3.13a)$$

$$v_l(x,\tau+1) = -\sum_{n=-N_M}^{n=+N_M} in\alpha\Phi^{(n)}(f(x,\tau+1),\tau+1)\, e^{in\alpha x}$$

$$= -\sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} in\alpha G^{(n)}_{k,\tau+1} T_k(f(x,\tau+1))\, e^{in\alpha x} \, . \qquad (3.3.13b)$$

Chebyshev polynomials and their derivatives evaluated at the wall, i.e., $T_k(f(x,\tau+1))$ and $DT_k(f(x,\tau+1))$, are periodic functions of x and thus can be expressed in terms of Fourier expansion as follows

$$T_k(f(x,\tau+1)) = \sum_{m=-N_s}^{m=+N_s} w^{(m)}_{k,\tau+1} e^{im\alpha x} \quad , \quad DT_k(f(x,\tau+1)) = \sum_{m=-N_s}^{m=+N_s} d^{(m)}_{k,\tau+1} e^{im\alpha x} \qquad (3.3.14a,b)$$

Where $\max(N_s)=N_TN_A$ and the method for evaluation of coefficients $w^{(m)}_{k,\tau+1}$ and $d^{(m)}_{k,\tau+1}$ is explained in Appendix A. Substitution of (3.3.14) into (3.3.13) gives

$$u_1(x,\tau+1)=\sum_{n=-N_F}^{n=+N_F}F^{(n)}(\tau+1)e^{in\alpha x}+\Gamma\sum_{n=-N_\theta}^{n=+N_\theta}\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_{k,\tau+1}^{(m)}d_{k,\tau+1}^{(n-m)}e^{in\alpha x}\quad,\tag{3.3.15a}$$

$$v_1(x,\tau+1)=-\sum_{n=-N_\theta}^{n=+N_\theta}\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}im\alpha G_{k,\tau+1}^{(m)}w_{k,\tau+1}^{(n-m)}e^{in\alpha x}$$

where

$$u_0(f(x,\tau+1))=\sum_{n=-N_F}^{n=+N_F}F^{(n)}(\tau+1)e^{in\alpha x}\ \text{where}\ N_F<N_M.\tag{3.3.16}$$

Comparison of (3.3.12) with (3.3.15) gives

$$U_{\tau+1}^{(n)}=F^{(n)}(\tau+1)+\Gamma\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_{k,\tau+1}^{(m)}d_{k,\tau+1}^{(n-m)}\ ,\ V_{\tau+1}^{(n)}=-\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}im\alpha G_{k,\tau+1}^{(m)}d_{k,\tau+1}^{(n-m)}\tag{3.3.17a,b}$$

The reader may note that $V_{\tau+1}^{(0)}$ is not independent but results from specification of $U_{\tau+1}^{(n)}$ (see Appendix B). Equations (3.3.17) can be used to express boundary conditions along the lower and upper walls, i.e., for at $\hat{y}_L(x,\tau+1)$ and $\hat{y}_U(x,\tau+1)$. In the case of our model problem the boundary conditions along the lower wall take following form

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_{k,\tau+1}^{(m)}\left(d_L\right)_{k,\tau+1}^{(n-m)}=-\Gamma^{-1}F_L^{(n)}(\tau+1),\qquad |n|\geq 0\tag{3.3.18a}$$

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}im\alpha G_{k,\tau+1}^{(m)}\left(w_L\right)_{k,\tau+1}^{(n-m)}=-\Gamma^{-1}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1},\qquad |n|\geq 1\tag{3.3.18b}$$

and are enforced for $|n| \le N_m$. Relations obtained for $|n| > N_m$ can be used as a measure of error in the enforcement of flow boundary conditions. Similar boundary conditions can be written for the upper wall by replacing subscript L with subscript U.

The above problem specification is incomplete due to the shortage of boundary conditions for the mode zero in Eq.(3.3.18b). While these conditions can be selected arbitrarily, we shall focus our attention on the fixed mass flow rate case, i.e., we shall require that the net mass flow rate in the x-direction is maintained constant during motion of the walls.

Volume flux Q along the channel can be evaluated by integrating the x-velocity component across the channel, i.e.,

$$Q(x,\tau+1) = \int_{\hat{y}_L}^{\hat{y}_U} \left(\frac{\partial \Psi_T}{\partial \hat{y}} \right)_{\tau+1} d\hat{y} = \Psi_T(x, \hat{y}_U(x,\tau+1), \tau+1) - \Psi_T(x, \hat{y}_L(x,\tau+1), \tau+1) \,. \quad (3.3.19)$$

The volume flux represents an x-periodic function that can be written in the form of a Fourier expansion

$$Q(x,\tau+1) = \sum_{n=-N_M}^{n=+N_M} Q^{(n)}(\tau+1) e^{in\alpha x} \quad (3.3.20)$$

where the zero term, i.e., $Q^{(0)}$, represents the net mass flux along the channel. Evaluation of this flux requires knowledge of Ψ_T at both walls. The evaluation process is very similar for both walls and thus we shall limit description to the lower wall only.

Values of Ψ_T at the lower wall can be evaluated by considering velocities of material points located on the line *l* overlapping with this wall. These velocities are known and can be expressed as

$$u_I(x,\tau+1)=0\,, \tag{3.3.21a}$$

$$v_I(x,\tau+1)=\frac{d\hat{y}_L(x,\tau+1)}{dt}\frac{dy}{d\hat{y}}=\Gamma^{-1}\sum_{n=-N_A,\,n\neq 0}^{k=+N_A}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x}\,. \tag{3.3.21b}$$

The Fourier expansion on the right hand side of (3.3.21b) does not contain mode zero as the mean position of the wall is assumed to be independent of time. The rate of change of Ψ_T along the wall can be written as

$$\frac{d\Psi_T}{dx}(x,\hat{y}_L(x,\tau+1))=\frac{\partial\Psi_T}{\partial x}(x,\hat{y}_L(x,\tau+1))+\Gamma\frac{\partial\Psi_T}{\partial\hat{y}}(x,\hat{y}_L(x,\tau+1))\frac{d\hat{y}_L}{dx}\,. \tag{3.3.22}$$

The first and second terms of the right hand side are replaced with (3.3.21b) and (3.3.21a), respectively, and the resulting expression is integrated in x resulting in

$$\Psi_T(x,\hat{y}_L(x,\tau+1))=-\sum_{n=-N_A,\,n\neq 0}^{n=+N_A}(in\alpha\Gamma)^{-1}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x}+C_L(\tau+1) \tag{3.3.23}$$

where C_L is an arbitrary function of time. An analogous expression written for the upper wall introduces another arbitrary function of time, i.e., C_U. Substitution of (3.3.23) and an equivalent relation for the upper wall into Eq.(3.3.19) and extraction of mode zero results in

$$C_U(\tau+1)=Q^{(0)}(\tau+1)+C_L(\tau+1)\,. \tag{3.3.24}$$

The value of $Q^{(0)}(\tau+1)$ is assumed in this analysis to be known and independent of time, and equal to the flow rate of the reference flow, i.e., $Q^{(0)}=\frac{4}{3}$.

One of the functions C_U and C_L can be selected arbitrarily and the other one follows from (3.3.24). In the description given below the latter one has been selected arbitrarily by

introducing condition $\Psi_T=0$ at a conveniently selected point x_0 at the lower wall resulting in

$$C_L(\tau+1)=\Gamma^{-1}\sum_{n=-N_A,n\neq 0}^{n=+N_A}(in\alpha)^{-1}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x_0}. \tag{3.3.25}$$

The computationally useful form of (3.3.24)-(3.3.25) requires introduction of Ψ as this is the unknown in our problem formulation (see Eq. 3.2.2; $\Psi=\Psi_T-\Psi_0$). Values of Ψ_0 evaluated along the lower wall represent a known function of t and x, periodic in x, that can be expressed as

$$\Psi_0(\hat{y}_L(x,\tau+1))=\sum_{n=-N_M}^{n=+N_M}\left(\Theta_L^{(n)}\right)_{\tau+1}e^{in\alpha x} \tag{3.3.26}$$

with a similar expression for the upper wall. Values of Ψ along the lower wall can be computed from (3.3.19), (3.3.23-3.3.26) as

$$\begin{aligned}\Psi(x,\hat{y}_L(x,\tau+1),\tau+1)=&-\sum_{n=-N_M,n\neq 0}^{n=+N_M}(in\alpha\Gamma)^{-1}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x}-\\&\sum_{n=-N_M}^{n=+N_M}\left(\Theta_L\right)_{\tau+1}^{(n)}e^{in\alpha x}+\sum_{n=-N_M,n\neq 0}^{n=+N_M}(in\alpha\Gamma)^{-1}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x_0}.\end{aligned} \tag{3.3.27a}$$

and for the upper wall as

$$\begin{aligned}\Psi(x,\hat{y}_U(x,\tau+1),\tau+1)=&-\sum_{n=-N_M,n\neq 0}^{n=+N_M}(in\alpha\Gamma)^{-1}\left(\frac{dA_U^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x}\\&-\sum_{n=-N_M}^{n=+N_M}\left(\Theta_U\right)_{\tau+1}^{(n)}e^{in\alpha x}+\sum_{n=-N_M,n\neq 0}^{n=+N_M}(in\alpha\Gamma)^{-1}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x_0}+Q^{(0)}\end{aligned} \tag{3.3.27b}$$

Stream function Ψ at the lower wall can be expressed in terms of the Fourier expansion (3.3.5) and the Chebyshev expansion (3.3.8) in the form

$$\Psi(x,\hat{y}_L(x,\tau+1),\tau+1)=\sum_{n=-N_M}^{n=+N_M}\Phi_L^{(n)}(\hat{y}_L(x,\tau+1))e^{in\alpha x} \tag{3.3.28}$$

where

$$\Phi_L^{(n)}(\hat{y}(x,\tau+1))=\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_{k,\tau+1}^{(m)}\left(w_{k,\tau+1}^{(n-m)}\right)_L . \tag{3.3.29}$$

Substitution of (3.3.28)-(3.3.29) and similar expressions for the upper wall into (3.3.27) results in the form of the closing conditions useful for numerical implementation, i.e.,

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_{k,\tau+1}^{(m)}\left(w_{k,\tau+1}^{(m)^*}\right)_L=-\Theta_L^{(0)}+\sum_{n=-N_M,n\neq0}^{n=+N_M}(in\alpha\Gamma)^{-1}\left(\frac{dA_L^{(n)}}{dt}\right)_{\tau+1}e^{in\alpha x_0}, \tag{3.3.30a}$$

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_{k,\tau+1}^{(m)}\left(w_{k,\tau+1}^{(m)^*}\right)_U=-\Theta_U^{(0)}+\sum_{n=-N_M,n\neq0}^{n=+N_M}(in\alpha\Gamma)^{-1}\left(\frac{dA_L^{(0)}}{dt}\right)_{\tau+1}e^{in\alpha x_0}+Q^{(0)}. \tag{3.3.30b}$$

Equations (3.3.10), (3.3.18) and (3.3.30) form a complete set of algebraic equations for the unknown coefficients $G_{k,\tau+1}^{(n)}$, k = 0,....,N_T, n=0,....,N_M. A solution of this system of equations moves calculations forward by one time step. Various methods of solutions will be discussed in the next section. Once the stream function has been determined, the velocity components can be computed from the definition of Ψ (see Eq. 3.2.2) while evaluation of pressure is discussed in Appendix C.

An alternative formulation frequently found in the literature [18] uses a constant pressure gradient constraint rather than the fixed mass flux constraint and the corresponding boundary conditions are discussed in Appendix D.

3.4. Testing of the algorithm

We shall discuss the performance of the algorithm in the context of two convenient test problems involving movements of boundaries, i.e., movements of the lower wall corresponding (i) to a traveling elastic wave and (ii) to a standing elastic wave.

3.4.1. Traveling elastic wave

3.4.1.1. Test problem

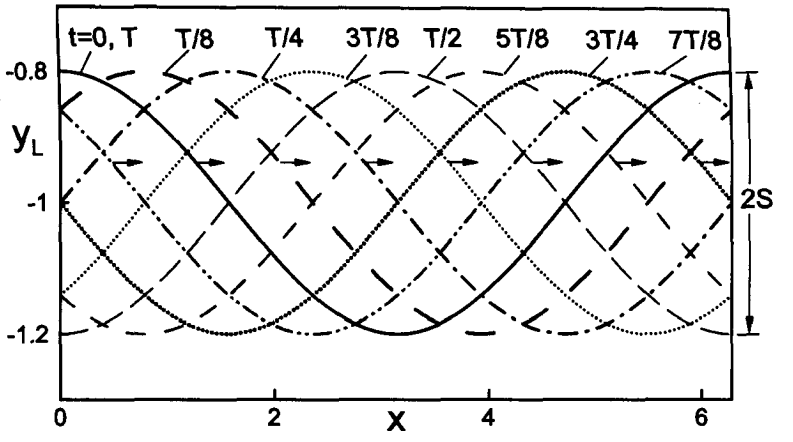


Figure 3.2. *Shape of the lower wall deformed by elastic traveling wave described by Eq.(3.4.1b) with the amplitude S=0.2, the wave number α=1.0 and the phase speed $c=\pi$ at t=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4 and 7T/8, where T denotes one time period.*

Consider an elastic wave traveling along the lower wall with the upper wall being flat. The shape of the resulting channel can be described as

$$y_L(x,t) = -1 + \sum_{n=-N_m,\, n\neq 0}^{n=+N_M} H_L^{(n)}\, e^{in\alpha(x-ct)}, \qquad y_U(x,t) = 1 \tag{3.4.1a}$$

and, in the simplest case of a sinusoidal wave, the shape of the lower wall becomes

$$y_L(x,t) = -1 + S\cos[\alpha(x-ct)] = -1 + (0.5Se^{i\alpha(x-ct)} + CC) \tag{3.4.1b}$$

where c denotes the phase speed, α denotes the wave number and S stands for the amplitude of the wave. Variations of the location of the lower wall as a function of time are illustrated for this particular case in Fig. 3.2.

Use of the Galilean transformation

$$X=x-ct \tag{3.4.2}$$

transforms the time-dependent moving boundary problem into a time-independent fixed boundary problem. The full problem in the moving frame of reference (X,y) takes the form

$$c\left[\frac{\partial^3}{\partial X^3}+\frac{\partial^3}{\partial X\partial y^2}\right]\Psi+\left[\frac{\partial^4}{\partial X^4}+2\frac{\partial^2}{\partial X^2\partial y^2}+\frac{\partial^4}{\partial y^4}\right]\Psi=0 \tag{3.4.3}$$

with boundary conditions in the form

$$u(X,+1)=0,\ v(X,+1)=0,\ u(X,y_L(X))=0,\quad v(y_L(X))=-c\sum_{n=-N_M,n\neq 0}^{n=+N_M} in\alpha H_L^{(n)}e^{in\alpha x}\ . \tag{3.4.4}$$

Elastic wave propagation problem represents a good test problem as it can be solved as a steady fixed-boundary problem in the moving reference frame and an unsteady moving-boundary problem in the fixed reference frame, and both solutions can be compared. The former one, i.e., (3.4.3)-(3.4.4), provides testing opportunity for the analysis of accuracy of spatial discretization and, especially, accuracy of enforcement of boundary conditions, and the latter one provides opportunity for testing of accuracy of temporal discretization. The former problem is solved using the steady version of the method discussed in the previous section.

3.4.1.2. Solution in the moving frame of reference

Problem (3.4.3)-(3.4.4) represents steady fixed boundary problem. As the first step, we wish to demonstrate the spectral accuracy of the spatial discretization used in the present work. While this discretization is fairly standard, one needs to pay special attention to the zone around the moving wall where the concept of immersed boundary conditions is used to impose flow boundary conditions due to possible problems associated with the formation of boundary layers in the distribution of modal functions. The Chebyshev expansions (3.3.8) with coefficients calculated using Galerkin procedure (3.3.10) are guaranteed to be spectrally accurate with the increasing number of terms N_T. In most cases sixty Chebyshev polynomials provide machine accuracy. When $\alpha\to\infty$ (corrugation with shorter wavelength) and higher Fourier modes begin to play a role, one needs to increase the number of Chebyshev polynomials in order to resolve wall boundary layers. These layers become extremely thin for larger values of α and for higher Fourier modes (see Fig. 3.3). Modal functions change very rapidly inside these layers while they are nearly zero in the rest of the domain. Typically one needs to use $N_T\approx 80$ for $\alpha=20$ and $N_T\approx 160$ for $\alpha=50$ in order to provide the required resolution.

The second aspect of spectral accuracy involves convergence of the truncated Fourier series (3.3.5) describing x-variations of the unknown. In all tests dealing with this issue the number of Chebyshev polynomials N_T was kept sufficiently large in order to reduce the associated error to machine accuracy. Chebyshev norm defined as

$$\left\|D\Phi^{(n)}\right\|_{\hat{\omega}} = \sqrt{\Gamma^2 \int_{-1}^{1} D\Phi^{(n)}(\hat{y},t) D\Phi^{(n)*}(\hat{y},t)\hat{\omega}(\hat{y})d\hat{y}} \quad , \quad \hat{\omega} = 1/\sqrt{1-\hat{y}^2} \tag{3.4.5}$$

had been adopted as a measure of the "magnitude" of the derivative of the modal function $\Phi^{(n)}$. Results displayed in Fig. 3.4 demonstrate that this norm decreases as a function of the mode number n with the rate of decrease very rapidly reaching the (asymptotic) exponential form.

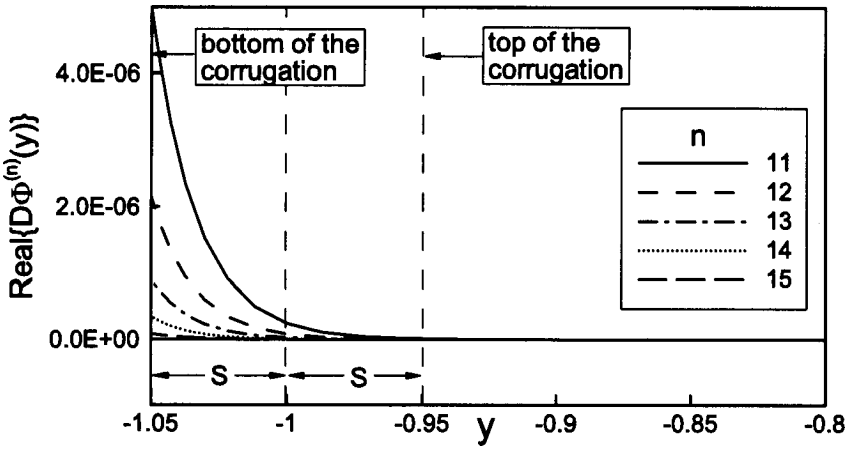


Figure 3.3. *Distribution of the real part of $D\Phi^{(n)}$ as a function of y for the higher modes (n>10) in the vicinity of the lower wall for the model problem (3.4.3)-(3.4.4) for α=5 and S=0.05. N_M=15 Fourier modes and N_T=80 Chebyshev polynomials used in the computations.*

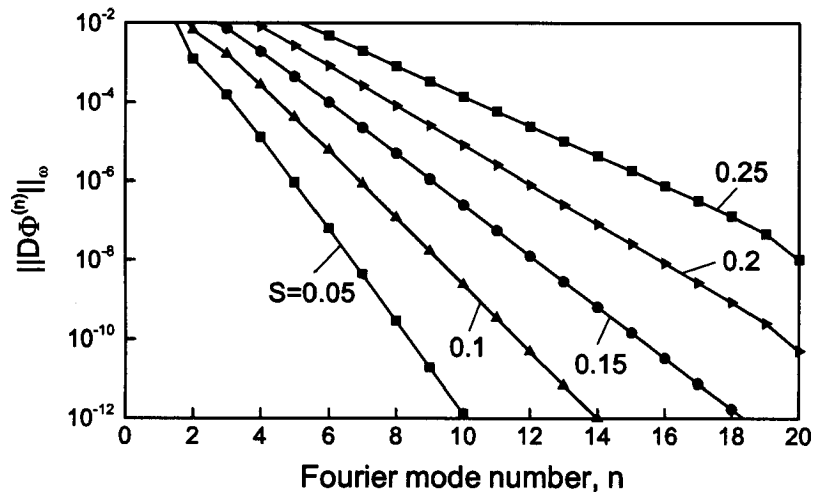


Figure 3.4. *Variations of the Chebyshev norm of $D\Phi^{(n)}$ (see Eq. 3.4.5) as a function of the Fourier mode number n for the model problem (3.4.3)-(3.4.4) for different wave amplitudes S for α=1.0. N_M=20 Fourier modes and N_T=60 Chebyshev polynomials used in the computations.*

Error in the enforcement of flow boundary conditions is of special interest. For convenience, we adopt the L_∞ norm for u- and v-velocity components evaluated at the lower wall as quantitative measures of this error. These norms are defined as

$$\|u_{er}(X)\|_\infty = \sup_{0\le X\le 2\pi/\alpha} |u_{er}(X)| \ , \quad \|v_{er}(X)\|_\infty = \sup_{0\le X\le 2\pi/\alpha} |v_{er}(X)| \tag{3.4.6a}$$

where

$$u_{er}(X) = u(X, y_L(X)) \ , \quad v_{er}(X) = v(X, y_L(X)) - v_L(X) \tag{3.4.6b}$$

and their distributions as functions of S and α are illustrated in Figs 3.5-3.6. The reader may note in judging these results that S=2 corresponds to a situation when the upper most point of the lower wall comes into contact with the upper wall.

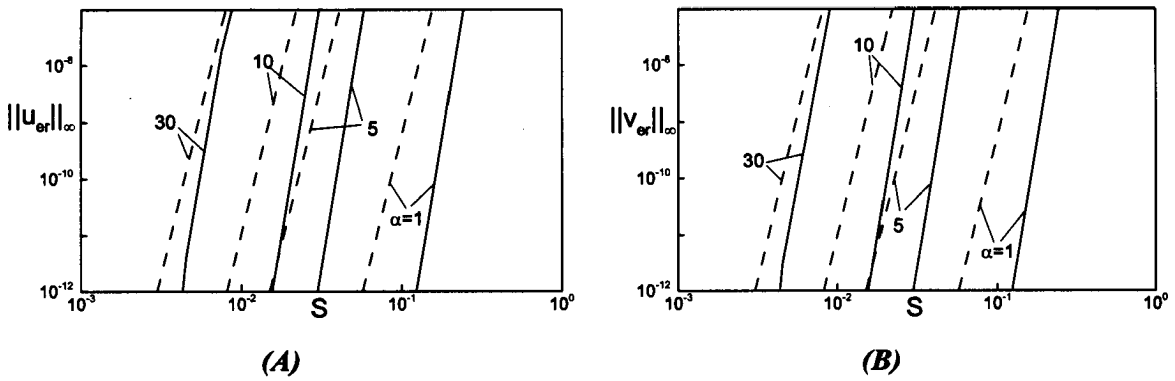


Figure 3.5. *Variations of the $\|u_{er}(X)\|_\infty$ (Fig. 3.5A) and $\|v_{er}(X)\|_\infty$ (Fig. 3.5B) norms (see Eq. 3.4.6a) as a function of the wave amplitude S for selected values of the wave number α for the model problem (3.4.3)-(3.4.4) with the phase speed $c=\pi$. The dashed and solid lines correspond to results obtained with the N_M = 10, 15 Fourier modes, respectively. The reader may note that S=2 corresponds to peak of the wave reaching the top wall. N_T=60 Chebyshev polynomials were used in the calculations.*

The available results suggest that the error is at machine accuracy level if α and S are below certain critical values. Once these values are reached, the error begins to increase rapidly in a fairly universal manner. The critical values of α and S can be increased by increasing the number of Fourier modes N_M used in the calculation, but the qualitative

character of the error increment remains unchanged. The location of the maximum error overlaps with the location of the lowest point in the lower wall, as illustrated by distribution of error along the lower boundary displayed in Fig. 3.7.

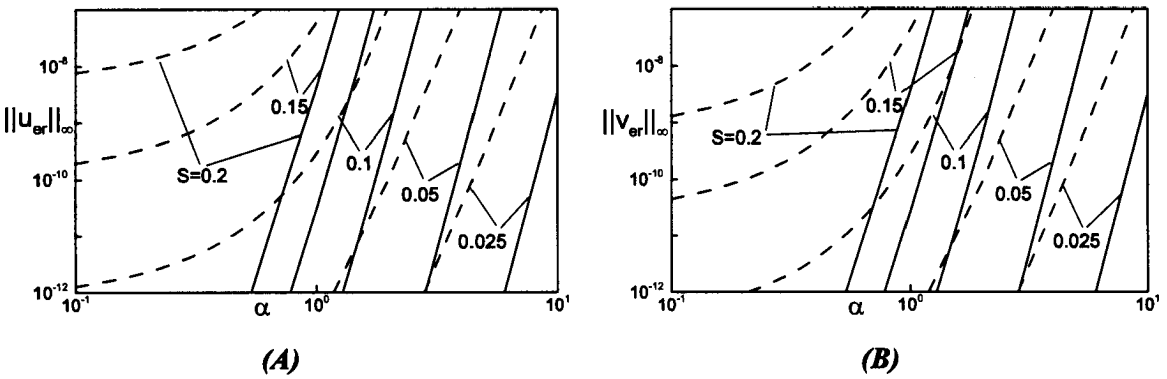


Figure 3.6. *Variations of the $\|u_{er}(X)\|_\infty$ (Fig. 3.6A) and $\|v_{er}(X)\|_\infty$ (Fig. 3.6B) norms (see Eq. 3.4.6a) as a function of the wave number α for selected values of the wave amplitude S for the model problem (3.4.3)-(3.4.4) for the phase speed $c=\pi$. Other parameters as in Fig. 3.5.*

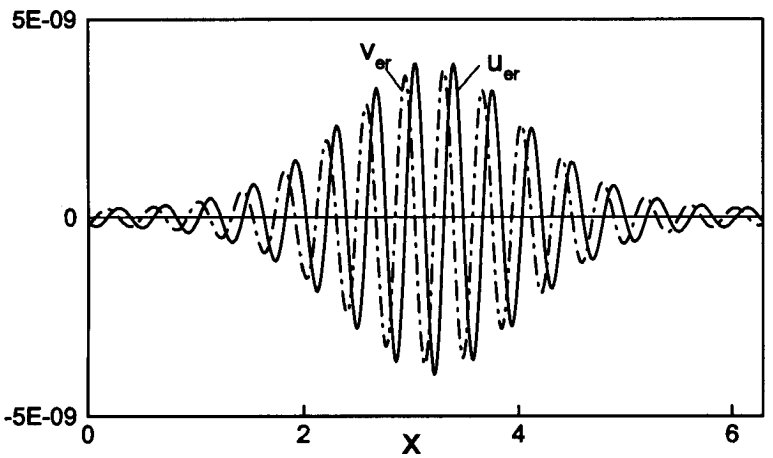


Figure 3.7. *Distribution of error u_{er} and v_{er} (see Eq. 3.4.6b) in the enforcement of flow boundary conditions at the lower wall for the model problem (3.4.3)-(3.4.4) with the $c=\pi$, S=0.2 and $\alpha=1$. The presented results were obtained with $N_M=15$ and $N_T=60$.*

3.4.1.3. Solution in the fixed frame of reference

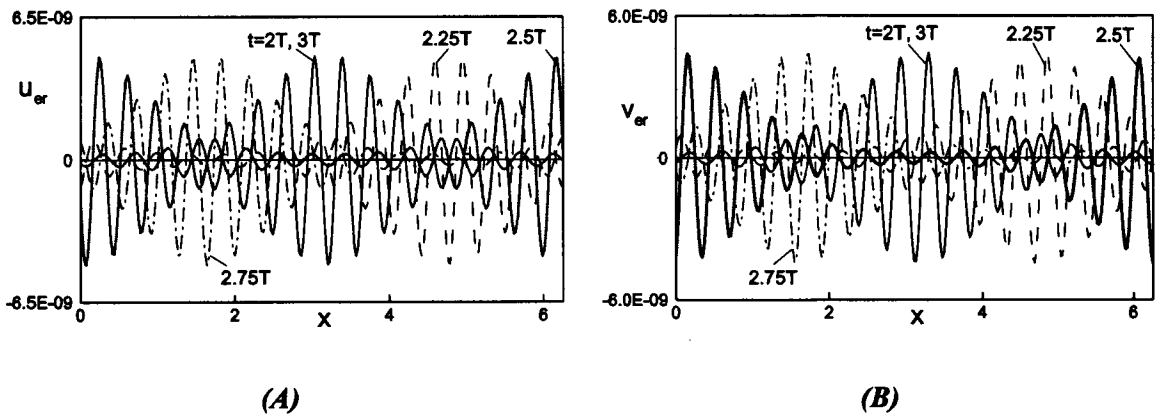


Figure 3.8. *Distribution of error in the enforcement of flow boundary condition u_{er} (Fig. 3.8A) and v_{er} (Fig. 3.8B) for the u- and v-velocity components (see Eq. 3.4.6b), respectively, at the lower wall at t = 2T, 2.25T, 2.5T, 2.75T and 3T, where T stand for one time period, for the model problem (3.4.3)-(3.4.4) with the amplitude S=0.2, the wave number α=1.0 and the phase speed $c=\pi$. The presented results were obtained through a direct solution of the moving boundary problem in the fixed coordinates system with N_M=15 Fourier modes and N_T=60 Chebyshev polynomials. Solution corresponding to the fixed boundary problem in the moving frame of reference (3.4.3)-(3.4.4) was used as the initial condition.*

The problem discussed in the previous section becomes a moving boundary problem when expressed in a fixed reference frame. We use solution obtained in the moving frame of reference as an initial condition and track evolution of the flow as a function of time. The time history of the error of enforcement of boundary conditions is illustrated in Fig. 3.8. It can be seen that the maximum error is similar to that found in the case of solution obtained in the moving frame of reference, location of this error follows location of the maximum opening of the slot as it moves in the x-direction, and the magnitude of this error remains approximately constant when several waves pass through the computational box.

Variations of the velocity components in selected test points are displayed in Fig. 3.9. The results shown in the figure demonstrate (i) the periodic variations in time with the expected period and (ii) the phase differences between different x-locations that match phase differences determined using solution in the moving frame of reference. Distribution of shear stress at the moving boundary displayed in Fig. 3.10 demonstrates that the expected periodicity and phase shifts have been reproduced. Results of the temporal grid convergence studies displayed in Fig. 3.11 demonstrate that the one-step and two-steps algorithms reproduce the expected first- and second-order accuracy in time, respectively.

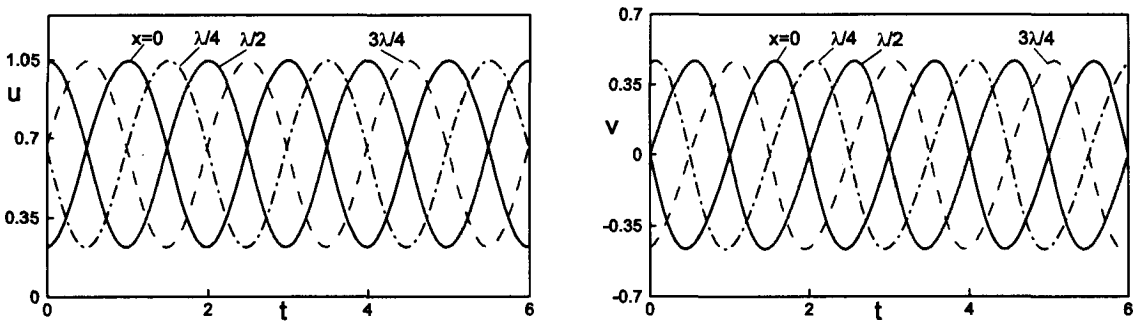


Figure 3.9. *Variations of the u-velocity (Fig. 3.9A) and v-velocity (Fig. 3.9B) components for three time periods at points (x,y) = (0, -0.6), (λ/4, -0.6), (λ/2, -0.6), (3λ/4, -0.6), $\lambda=2\pi/\alpha$, for the model problem (3.4.3)-(3.4.4) solved directly as a moving boundary problem in the fixed reference frame. Solution corresponding to the fixed boundary problem in the moving frame of reference (3.4.3)-(3.4.4) was used as the initial condition. N_M=15 Fourier modes and N_T=60 Chebyshev polynomials were used in the computations. Other conditions as in Fig. 3.2.*

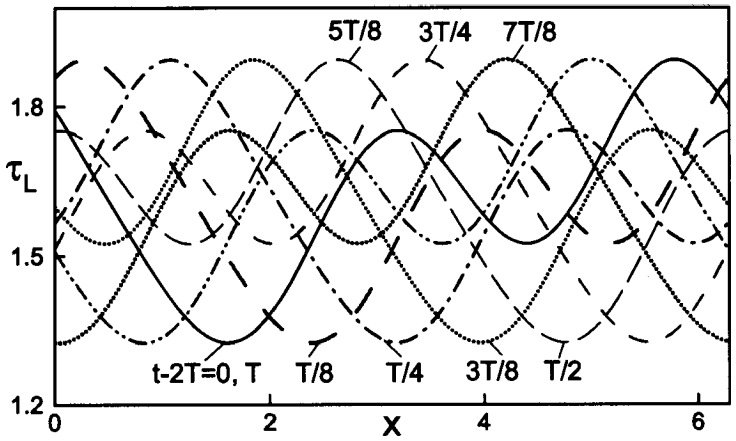


Figure 3.10. *Distribution of the wall shear stress at different time levels after two time periods for the model problem (3.4.3)-(3.4.4). Other parameters are as in Figs 3.2 and 3.9.*

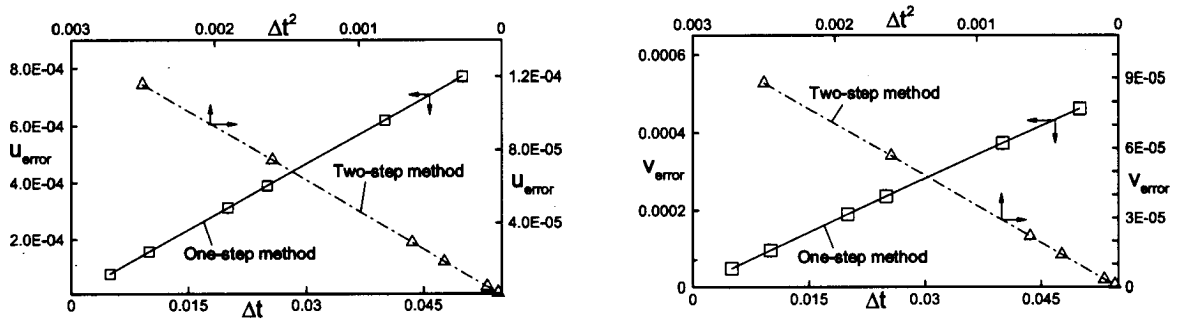


Figure 3.11. *Variations of error in the u- (Fig. 3.11A) and v- (Fig. 3.11B) velocity components as a function of the step size Δt used in the temporal discretization in the case of the model problem (3.4.3)-(3.4.4) with the amplitude $S=0.1$, the wave number $\alpha=1.0$ and the phase speed $c=\pi$. The error is defined as the maximum of the absolute value of the difference between the results obtained through the direct solution of the moving boundary problem in the fixed frame of reference and the corresponding fixed boundary problem in the moving frame of reference at a time corresponding to $t=1.0$. $N_M=10$ Fourier modes and $N_T=60$ Chebyshev polynomials were used in the computations. Solution of the model problem (3.4.3)-(3.4.4) in the moving frame of reference is taken as the initial condition for the direct solution method.*

3.4.2. Standing elastic wave

3.4.2.1. Test problem

Consider standing elastic wave at the lower wall with the upper wall being flat. The shape of the resulting channel can be described as

$$y_L(x,t) = -1 + \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_L^{(n)}(t)\, e^{in\alpha x}, \qquad y_U(x,t) = 1 \tag{3.4.7}$$

and, in the simplest case of a sinusoidal wave, the shape of the lower wall becomes

$$y_L(x,t) = -1 + S\cos(\omega t)\cos(\alpha x) = -1 + (0.5\, S\cos(\omega t)\, e^{i\alpha x} + CC) \tag{3.4.8}$$

where α denotes the wave number, S stands for the amplitude and ω denotes the frequency of the wave. Variations of the location of the lower wall as a function of time are illustrated for this particular case in Fig. 3.12.

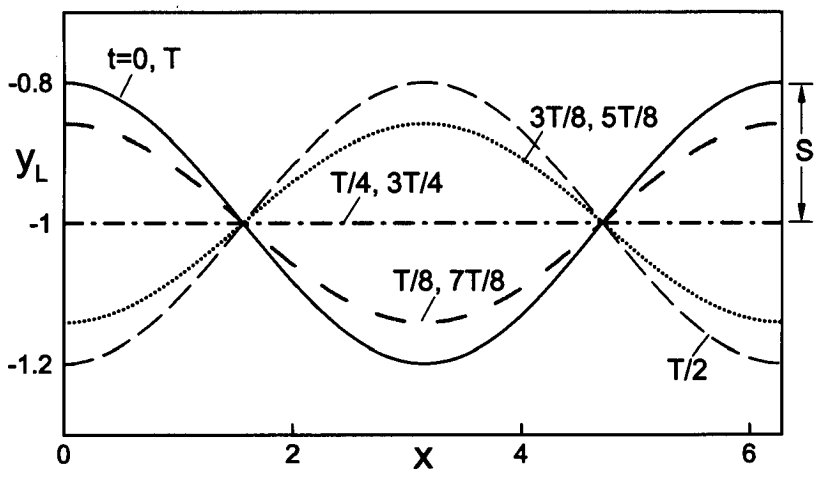


Figure 3.12. *Shape of the lower wall modified by elastic standing wave with the wave number α=1.0, the amplitude S=0.2 and the frequency $\omega=\pi$ at times t=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T denotes one time period.*

The full test problem has the form

$$\frac{\partial}{\partial t}\left[\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right]\Psi=\left[\frac{\partial^4}{\partial x^4}+\frac{\partial^2}{\partial x^2\partial y^2}+\frac{\partial^4}{\partial y^4}\right]\Psi \tag{3.4.9}$$

with boundary conditions in the form

$$u(x,+1,t)=0\,,\quad v(x,+1,t)=0\,,$$

$$u(x,y_L(x),t)=0,\quad v(x,y_L(x,t),t)=\sum_{n=-N_M,n\neq 0}^{n=+N_M}\frac{d}{dt}\left(H_L^{(n)}(t)\right)e^{in\alpha x} \tag{3.4.10}$$

For convenience, we use the L_∞ norm as a quantitative measure of error associated with the enforcement of velocity boundary conditions at the lower wall, i.e.,

$$\left\|u_{er}(x,t)\right\|_\infty=\sup_{0\le X\le 2\pi/\alpha}\left|u_{er}(x,t)\right|\ ,\ \left\|v_{er}(x,t)\right\|_\infty=\sup_{0\le X\le 2\pi/\alpha}\left|v_{er}(x,t)\right| \tag{3.4.11a}$$

where

$$u_{er}(x,t)=u(x,y_L(x,t),t)\ ,\ v_{er}(x,t)=v(x,y_L(x,t),t)-v_L(x,t))\,. \tag{3.4.11b}$$

3.4.2.2. Solution of the test problem

Figure 3.13 illustrates distribution of error in the enforcement of boundary conditions at the lower wall. It can be seen that the magnitude of the error changes periodically in time with frequency equal to double the frequency of the standing wave. Variations in the magnitude of the error can be correlated with variations of the shape of the wall, i.e., the maximum of error occurs at times when the channel opening is the largest. The

magnitude of error can be reduced by increasing the number of Fourier modes used in the calculations but the qualitative behaviour of error with time remains unchanged.

The spatial distribution of error is illustrated in Fig. 3.14 after 3 and 3.5 cycles of the wave motion. It can be seen that the maximum of error shifts to location corresponding to the maximum opening of the channel. The Fourier spectra of the error shown in Fig. 3.15 demonstrate that the N_M number of Fourier modes have been eliminated according to the construction of the immersed boundary conditions algorithm described in Section 3.3. The largest error is associated with the first Fourier mode omitted in the enforcement of boundary conditions. The error associated with the higher omitted Fourier modes rapidly decays as the mode number increases.

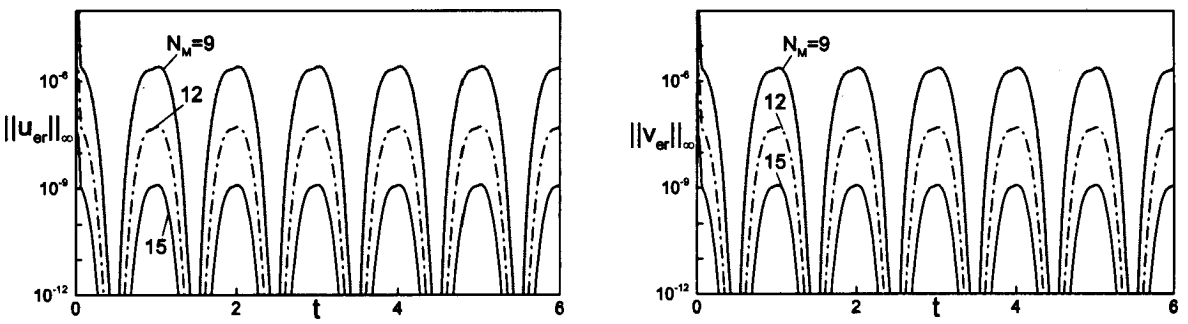
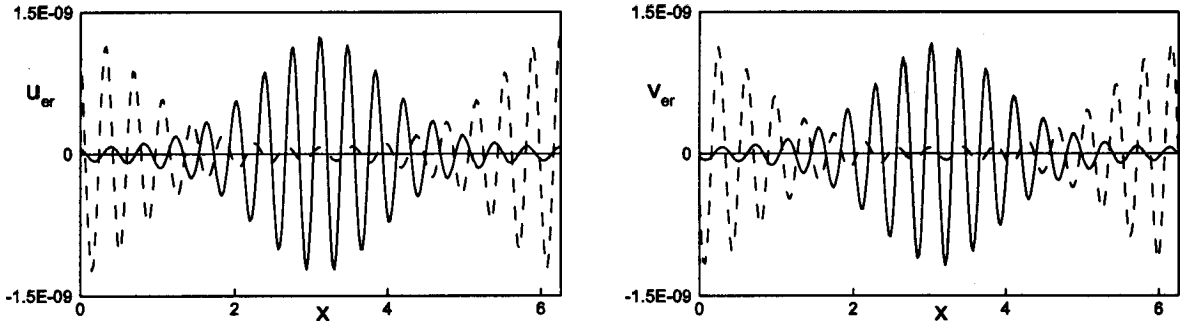


Figure 3.13. *Variations of the $\|u_{er}(x,t)\|_\infty$ (Fig. 3.13A) and $\|v_{er}(x,t)\|_\infty$ (Fig. 3.13B) norms (see Eq. 3.4.11a) as a function of time over three time periods for the moving boundary problem defined by (3.4.8)-(3.4.10) for α=1, S=0.2 and $\omega=\pi$ (shape of the wall is illustrated in Fig. 3.12) computed using N_T=60 and different values of N_M. The initial conditions used correspond to shape of the wall given by Eq. (3.4.8) at time t=0 and zero flow modifications. It can be seen that the initial transient dies out and the time-periodic flow response begins to dominate after 1-2 periods.*



(A) *(B)*

Figure 3.14. *Distribution of the error u_{er} (Fig. 3.14A) and v_{er} (Fig. 3.14B) in the u- and v-velocity components (see Eq. 3.4.11b) at the lower wall after three (solid line) and three and half time periods (dashed line) for the model problem (3.4.8)-(3.4.10) evaluated with N_M=15 Fourier modes. All other conditions are as in Figure 3.13.*

Figure 3.15 also displays results of tests carried out in order to check if the method produces any spurious spatial oscillations. Three cases were considered, i.e., in case (A) the wave was represented by the principal Fourier mode and the calculations had been carried out with N_M=9 Fourier modes, in case (B) the wave was represented by the second Fourier mode (the principal mode has the wave number α=0.5), and in case (C) the wave was represented by the third Fourier mode (the principal mode has the wave number α=1/3). In order to have fully equivalent representations, the number of Fourier modes used in cases (B) and (C) were N_M=18 and N_M=27, respectively. The selected representations admitted subharmonics of the 1/2 type in case (B) and 1/3 type in case (C). The Fourier spectra shown in Fig. 3.15 demonstrate the equivalency of the results in all three cases. No subharmonics had been produced during the solution process and the modes expected to produce zero contributions in cases (B) and (C) behaved as expected.

Results of temporal grid convergence studies are displayed in Fig. 3.16 and demonstrate that the one-step and two-steps algorithms reproduce the expected first- and second-order accuracy in time, respectively. Variations of velocity components at a test point displayed in Fig. 3.17 demonstrate the expected periodic variations in time.

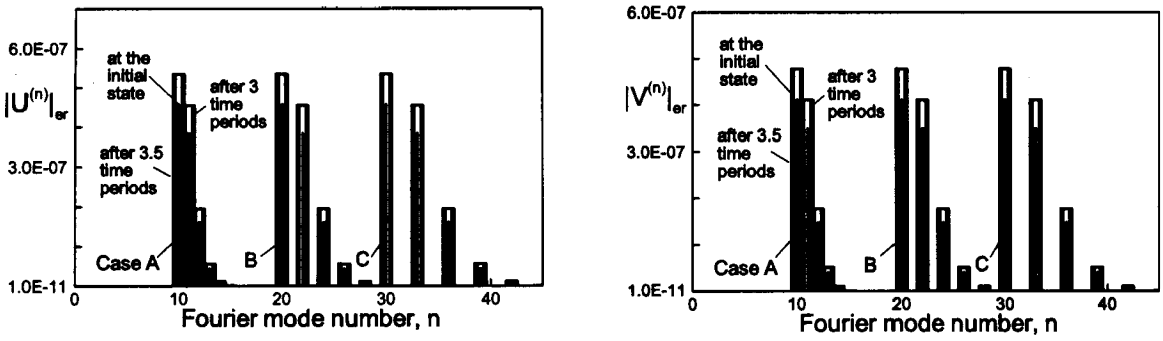


Figure 3.15. *Fourier spectra of error distributions u_{er} and v_{er} (see Eq. 3.4.11b) in the u- (Fig. 3.15A) and v- (Fig. 3.15B) velocity components at the lower wall for the model problem (4.8)-(4.10). Three different forms of Fourier expansions were considered, i.e., case A: α=1.0, N_M=9 ; case B: α=0.5, N_M=18 and case C: α=1/3, N_M=27. N_T=60 Chebyshev polynomials used in all cases. All other conditions as in Fig. 3.13.*

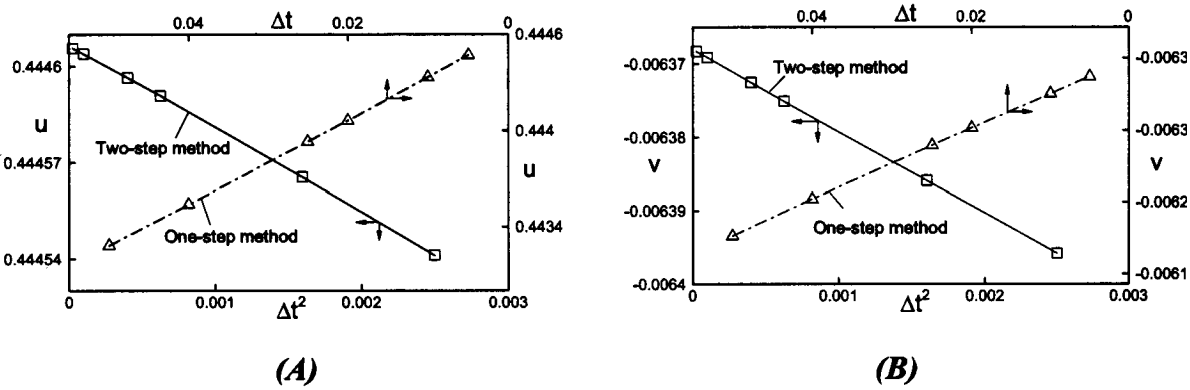


Figure 3.16. *Variations of the u- (Fig. 3.16A) and v- (Fig. 3.16B) velocity components as a function of the step size Δt used in the temporal discretization at a test point (x,y)=(π/2, -0.75) in the case of the model problem (3.4.8)-(3.4.10) with the amplitude S=0.1, the wave number α=1.0 and the frequency $\omega=\pi$ at time t=1.0. The initial conditions correspond to the solution of the fixed boundary problem for the wall shape given by Eq.(3.4.8) at t=0. N_M=10 Fourier modes and N_T=60 Chebyshev polynomials were used in the computations.*

Variations of the shear stress at the moving wall are displayed in Fig. 3.18 and also follow the expected periodic variations in time. The instantaneous streamlines associated

with a standing wave with a more complex form are displayed in Fig. 3.19 in order to illustrate the algorithm's ability to deal with more complex motions.

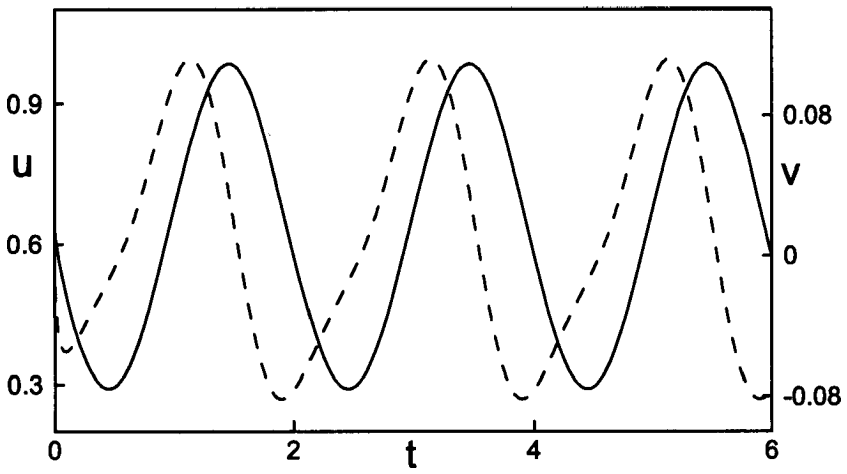


Figure 3.17. *The evolution of the u- (solid line) and v- (dashed line) velocity components at a test point (x,y)=(λ /4,-0.6) during the first four time periods for the model problem (3.4.8)-(3.4.10) evaluated with N_M=15 Fourier modes. All other conditions are as in Figure 3.13.*

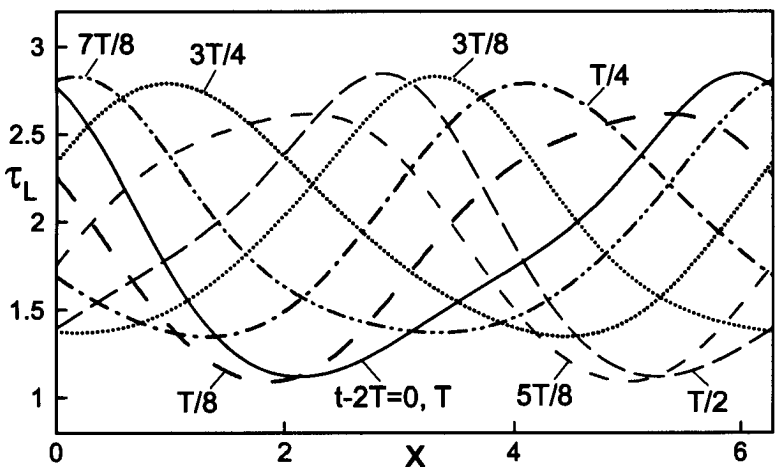


Figure 3.18. *Distribution of the wall shear stress at different time levels during the third time period. The results shown were determined using N_M=15 Fourier modes. All other conditions are as in Figure 3.13.*

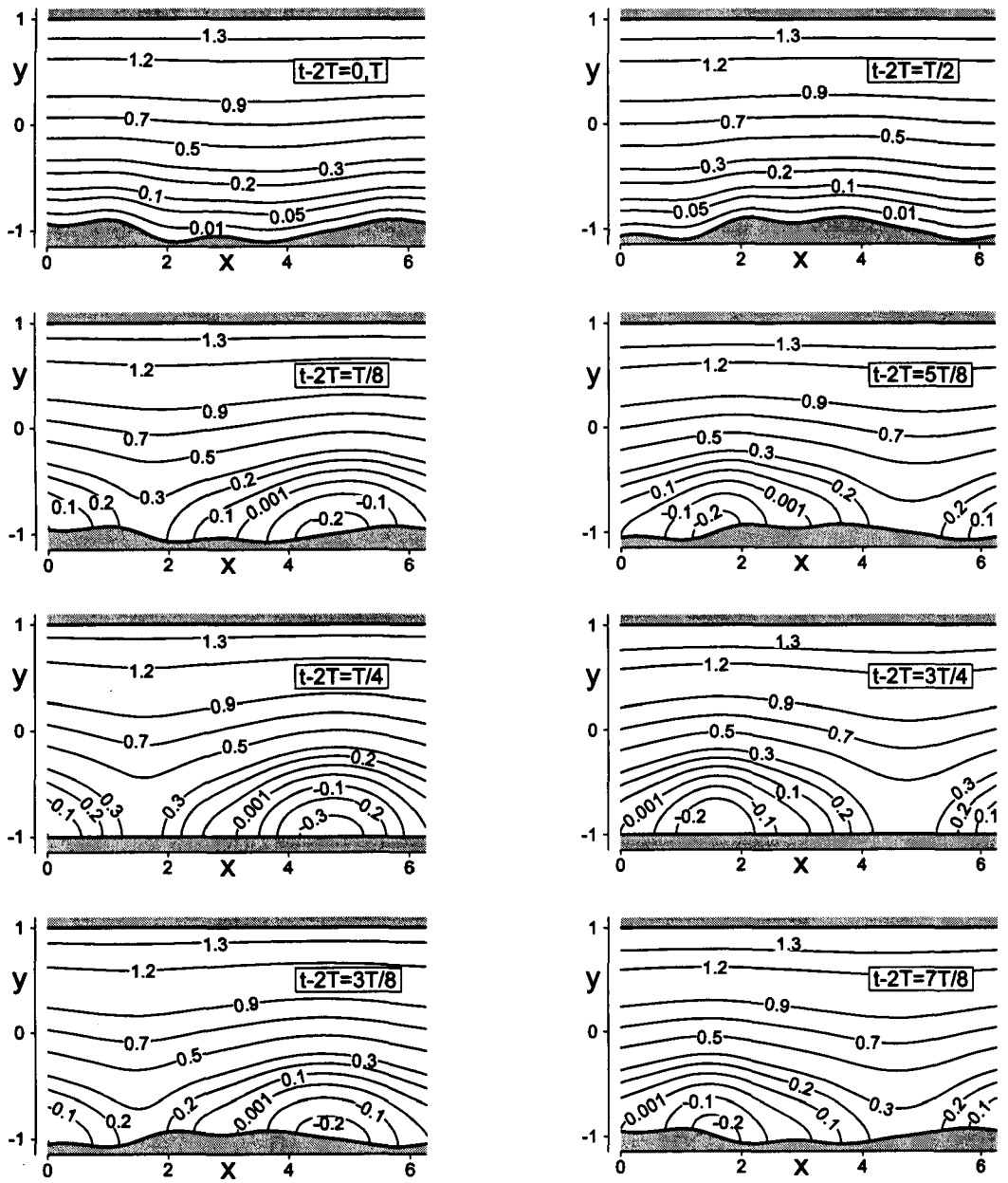


Figure 3.19. *Instantaneous streamlines in a channel bounded by* $y_L(x,t)=-1+(0.05ie^{ix}-0.0167ie^{3ix}+0.0125ie^{4ix}+CC)\cos(\omega t)$ *and* $y_U(x,t)=1$, *for* $\omega=\pi$ *using* $N_M=25$ *and* $N_T=80$ *evaluated after two time periods. Results are shown at times t-2T=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T stands for one time period. The initial conditions used correspond to shape of the walls at time t=0 i.e.* $y_L(x,0)=-1+(0.05ie^{ix}-0.0167ie^{3ix}+0.0125ie^{4ix}+CC)$ *and* $y_U(x,0)=1$, *and zero flow modifications.*

3.4.3. Efficiency of the algorithm

The proposed algorithm requires solution of a system of linear algebraic equations resulting from the discretization of the field equations and boundary conditions at each time step. The resulting matrix structure is displayed in Fig. 3.20 and underscores the fact that algebraic equations resulting from discretization of each modal equation of type (3.3.7a) are uncoupled, i.e., the corresponding coefficients form blocks in the upper triangular form. The only coupling between Fourier modes is provided through the boundary conditions, i.e., the corresponding entries form horizontal lines in the matrix (four lines per block).

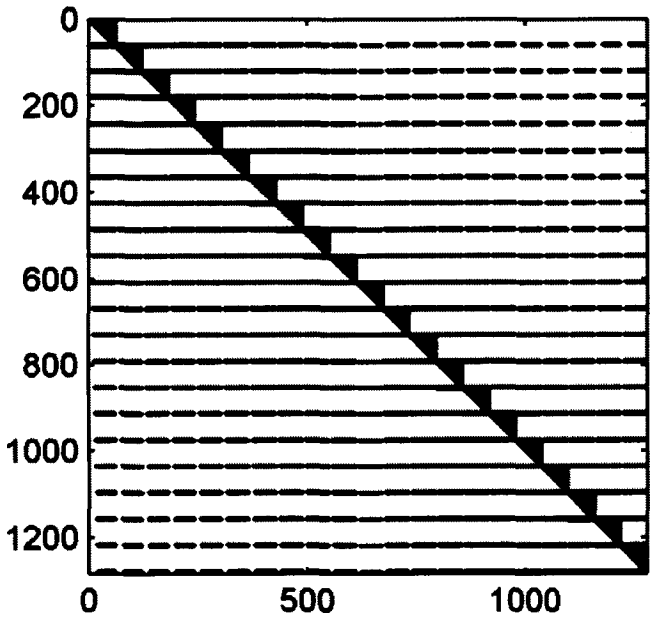


Figure 3.20. *Structure of the coefficient matrix for the test problem (3.4.8)-(3.4.10) obtained with* $N_M=10$ *and* $N_T=60$. *Non-zero entries are marked in black.*

Direct solution of the matrix equation leads to a solution method that we shall refer to as the 'direct algorithm'. The efficiency of this algorithm is very good as the only matrix entries that have to be recomputed at each time step correspond to the boundary conditions. In addition, the shape of the moving wall can be easily adjusted by changing

the magnitude of the relevant Fourier coefficients. In this sense, the proposed algorithm is superior with respects to algorithms based on dynamic grid adjustments and/or mapping methods as the cost of re-gridding and matrix construction have been eliminated, and comparable to algorithms based on various forms of interface tracking while delivering much higher accuracy.

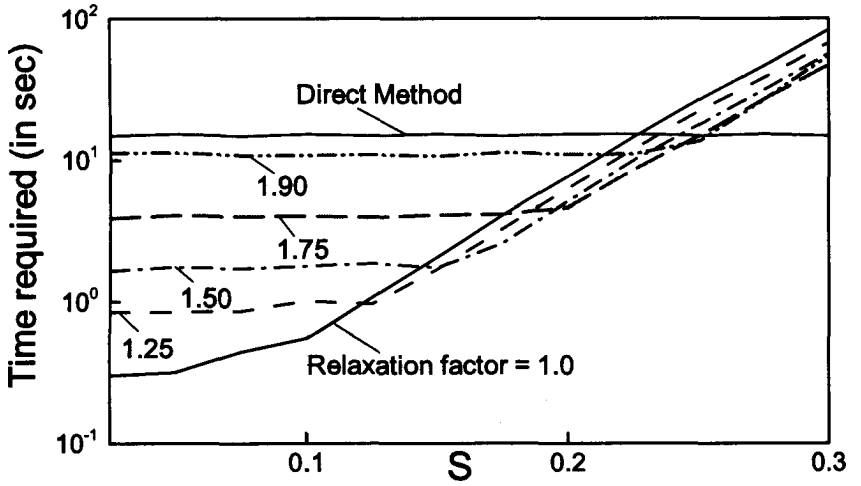


Figure 3.21. *Performance of the first version of the decoupled algorithm with different values of the over-relaxation factor for the model problem (3.4.3)-(3.4.4) as a function of the amplitude S of the traveling wave with the wave number α=1.0 and the phase speed $c=\pi$. Time required by the direct method is given for reference. All calculations have been carried out with N_M=15 Fourier modes and N_T=60 Chebyshev polynomials.*

The matrix of coefficients can be very large when a large number of Fourier modes are required and this motivates the search for a method of solution that avoids construction as well as inversion of the complete matrix. Structure of the matrix (see Fig. 3.20) suggests the use of an iterative solution algorithm based on the decoupling of Fourier modes. We have tested four versions of such algorithms which we shall refer to as the 'decoupled algorithms'.

In the first version, the unknowns corresponding to a Fourier mode of interest in equation (3.3.18) and (3.3.37) at the current time step are expressed in terms of the remaining

Fourier modes using their values from the previous time step (or from the previous iteration). The solution process begins with mode 0, proceeds to the next mode using the most recent information available and continues until the last mode N_M is reached, and then it is repeated until a convergence criterion is satisfied. In this way, the inversion of the complete matrix of size $(N_T+1)*(2N_M+1)$ is replaced by a repetitive solution of system of (N_T+1) equations for each Fourier mode.

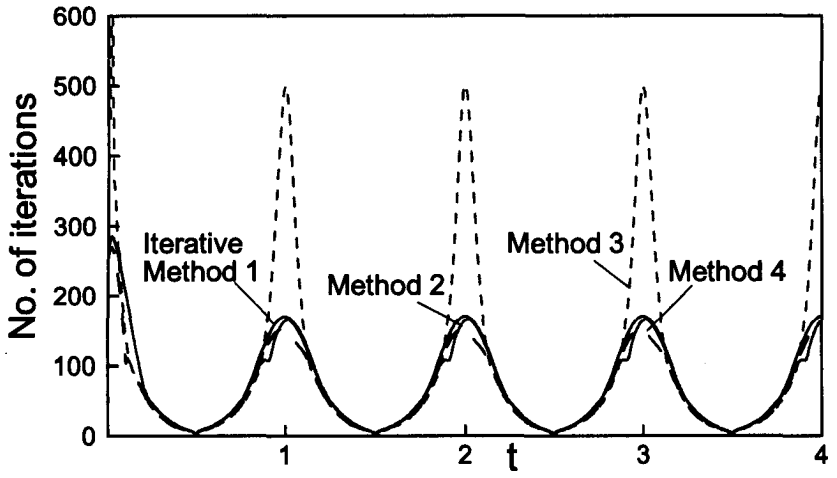


Figure 3.22. *Number of iterations required by different versions of the decoupled algorithm as a function of the simulation time (and thus position of the wall). Simulations have been carried out over two periods of motion of the standing wave described in Section 3.4.2 with the wave amplitude S=0.15, the wave number $\alpha=1.0$ and the frequency $\omega=\pi$ using zero initial condition, $N_M=15$ Fourier modes and $N_T=60$ Chebyshev polynomials.*

Versions 2-4 retain different couplings between the main modes as this might accelerate convergence. In version 2 we solve for each mode simultaneously with mode 0, e.g., we solve a system of equations for modes 0 and 1, then for modes 0 and 2, and then for modes 0 and 3, and so on. In version 3 we solve for each mode simultaneously with mode 1, i.e., mode 0 and 1, then 1 and 2, followed by 1 and 3, and so on. In the forth version, we solve for modes 0, 1 and 2 together and then separately for each of the remaining modes. It is necessary to point out that the decoupled algorithms reduce memory requirements as one needs to work with many small matrices rather then one

very large matrix as well as they open possibility for parallelization of the computations. This issue becomes significant in the case of three-dimensional problems and large number of Fourier modes and Chebyshev polynomials.

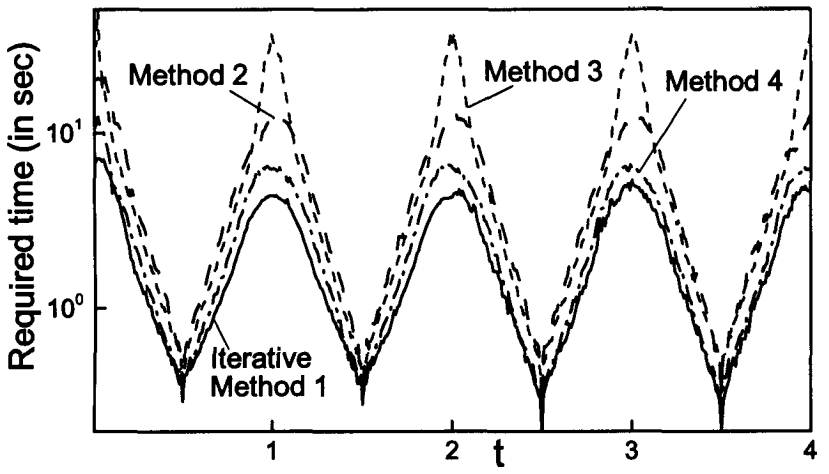


Figure 3.23. *Variations of time (in sec) required to achieve convergence using different versions of the decoupled algorithm as a function of the simulation time (and thus position of the wall). All the other conditions as in Fig. 3.22.*

The rates of convergence of all versions of the decoupled algorithm are generally very good; they decrease with an increase of the amplitude S and the wave number α characterizing the moving boundary. Their efficiency is illustrated in Fig. 3.21 in the context of the test problem discussed in Section 3.4.1 solved in the moving reference frame. It can be seen that the iterative method (version 1) requires less time than the direct method as long as the amplitude S is less than ~0.23. Over-relaxation decreases the efficiency of the method for S<~0.23 but improves efficiency for higher amplitudes S. The relative efficiency of different versions is illustrated in Figs 3.22 and 3.23 in the context of the test problem discussed in Section 3.4.2. Figure 3.21 displays variations in the number of iterations required for convergence as a function of simulation time. The maxima in the number of iterations correspond to instants of time when the channel opening passes through the maximum and minima correspond to times when the channel becomes flat. Figure 3.23 displays analogous information but expressed in terms of the computing time required for convergence. The local maxima correspond to instants of

time when the channel opening passes through the maximum and minima correspond to times when the channel becomes flat. It can be seen that version 1 of the decoupled algorithm is most efficient as inclusion of additional coupling between Fourier modes increase number of iterations and time required for convergence.

3.5. Conclusions

A gridless algorithm for unsteady flow problems described by the biharmonic operator (Stokes flow) has been presented and tested. The algorithm uses a fixed computational domain with the flow domain completely immersed inside the computational domain. The flow boundary conditions are imposed using the concept of immersed boundary conditions. Two versions of boundary condition had been presented, i.e., corresponding to the fixed mass flux constraint and to the fixed pressure gradient constraint. The algorithm is suitable for the flow domains that have form of a channel whose geometry is periodic in the stream-wise direction. The opening of the channel may change as a function of time in an arbitrary manner but resulting in an instantaneous geometry that is Fourier transformable. The algorithm uses Fourier expansions for the spatial discretization in the flow direction, Chebyshev expansions for the discretization in the transverse direction and the first- and second-order implicit temporal discretizations. Various tests confirm the spectral accuracy of the spatial discretization and the theoretically predicted accuracy of the temporal discretization. The algorithm is very efficient as the part of the coefficient matrix corresponding to the field equations needs to be computed only once and the only changes required at each time step are limited to entries corresponding to the boundary conditions. In addition, the algorithm is very flexible as far as adaptation to different geometries is concerned as the only changes required are limited to changes in the Fourier coefficients describing motions of the walls. The numerical error associated with treatment of flow boundary conditions is well controlled during the simulation process. No numerical instabilities have been observed. Direct and decoupled versions of the algorithm had been explored. It had been found that while the direct algorithm is more efficient in situations when large-amplitude motions of

the boundaries are of interest, a simple mode-decoupled algorithm is significantly faster when the interest is limited to small amplitude motions.

3.6. References

[1] Floryan, J.M. and Rasmussen, H., Numerical analysis of viscous flows with free surfaces, *Appl. Mech. Rev.*, vol. 42, pp. 323-341 (1989).

[2] Scardovelli, R. and Zaleski, S., Direct numerical simulation of free surface and interfacial flow, *Annu Rev. Fluid Mech.*, vol. 31, pp. 567-603 (1999).

[3] Inculet, I., Floryan, J.M. and Haywood, R., Dynamics of water droplets break-up in electric fields, *IEEE Transactions on Industry Applications*, vol. 28, pp. 1203-1209 (1992).

[4] Hamed, M. and Floryan, J.M., Numerical simulation of unsteady nonisothermal capillary interfaces, *J. Comp. Phys.* vol. 145, pp. 110-140 (1998).

[5] Hyman, J.M., Numerical methods for tracking of interfaces, *Physica,* vol. 12D, pp. 396-407 (1984).

[6] Glimm, J., Grove, J.W., Li, X.L., Shyue, K.M., Zeng, Y. and Zhang, Q., Three-dimensional front tracking, *SIAM J. Sci. Computing,* vol. 19, pp. 703-727 (1998).

[7] Harlow, F.H. and Welch, J.E., Numerical study of large amplitude free surface motions, *Phys. Fluids*, vol. 9, pp. 842-851 (1966).

[8] Hirt, C.W. and Nichols, B.D., Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comp. Phys.*, vol. 39, pp. 201-225 (1981).

[9] Osher, S.J. and Sethian, J.A., Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations, *J. Comp .Phys.*, vol. 79, pp. 12-49 (1988).

[10] Sethian, J.A. and Smereka, P., Level set methods for fluid interfaces, Annu. *Rev. Fluid Mech.*, vol. 35, pp. 341-372 (2003).

[11] Szumbarski, J. and Floryan, J.M., A direct spectral method for determination of flows over corrugated boundaries, *J. Comp. Phys.*, vol. 153, pp. 378-402 (1999).

[12] Peskin, C.S., The fluid dynamics of heart valves: experimental, theoretical and computational methods, *Annu. Rev. Fluid Mech.,* vol. 14 pp. 235-59 (1982).

[13] Mittal, R. and Iaccarino, G., Immersed boundary methods, *Annu. Rev. Fluid Mech.,* vol. 37 pp. 239-261 (2005).

[14] Floryan, J.M., Centrifugal instability of Couette flow over a wavy wall, *Phys. Fluids*, vol. 14, pp. 312-322 (2002).

[15] Floryan, J.M., Three-dimensional instabilities of laminar flow in a rough channel and the concept of hydraulically smooth wall, *Eur. J. Mech. B/Fluids*, vol. 26, pp. 305-329 (2007).

[16] Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for unsteady flow problems described by the Laplace operator, *Int. J. Num. Meth. Fluids*, vol. 56, pp. 1765-1786 (2007).

[17] Husain, S.Z. and Floryan, J.M., Implicit spectrally-accurate method for moving boundary problems using immersed boundary conditions concept, *J. Comp. Phys.*, vol. 227, pp. 4459-4477 (2008).

[18] Floryan, J.M., Stability of wall-bounded shear layers in the presence of simulated distributed roughness, *J. Fluid Mech.*, vol. 335, pp. 29-55 (1997).

CHAPTER 4

Over-determined formulation of the IBC method[1]

4.1. Introduction

'Immersed boundary' (IB) methods refer to a class of methods where the computational domain extends beyond the physical domain resulting in edges of the physical domain immersed inside the computational domain. The name has been coined by Peskin [1] in the context of cardiac mechanics problems. The concept is very attractive as one can work with a fixed, regular computational domain regardless of the shape of the physical domain, i.e., the cost of generating boundary conforming grid has been completely eliminated. Field equations can be discretized using a simple reference coordinate system and are never changed regardless of the geometry of the physical domain. The main challenge associated with this method is the development of procedures that result in the enforcement of physical boundary conditions along the physical boundaries located inside of the computational domain. There are no conditions to be imposed along the edges of the computational domain (unless the edges of the physical and computational domains coincide) and thus the problem formulation needs to be closed by a set of constraints rather then by the classical boundary values. The IB method has been developed primarily in the context of fluid flow problems. The prevailing procedure for imposition of constraints replacing physical boundary conditions involves introduction of additional forcing that makes the fluid to move along the physical boundary. This methodology has roots in the physics of the problem, requires good understanding of the

---

[1]A version of this chapter has been accepted for publication as –

Husain, S.Z., Floryan, J.M. and Szumbarski, J., Over-Determined Formulation of the Immersed Boundary Conditions Method, *Computer Methods in Applied Mechanics and Engineering*, doi: 10.1016/j.cma.2009.09.022 (2009).

problem, and algorithm calibration involves elements of trial and error. Details of procedures based on the so-called continuous and discrete forcing are reviewed in [2, 3].

Fictitious domain methods [4, 5] offer an alternative approach for handling boundary irregularities where problems formulated on a complicated domain are solved on a simpler domain that contains the complicated domain. Use of simple domain enables efficient computational grid generation. The fictitious domain method is very suitable for moving boundary problems as it does not require regeneration of grid to account for the changing boundary geometries [5].

A separate group of methods has its roots in the methodology developed for handling the moving boundary problems and has been reviewed in [6]. Here one should focus on the fixed grid methods where the motion of the interface is tracked through a reference fixed grid. The most popular methods are based on the fluid fluxes and are known as the volume of the fluid (VOF) method. More recent methods rely on the concept of level set [7,8]. All these methods are of low-order in terms of spatial accuracy as they are based on the low-order finite-difference and/or finite-volume discretizations, and the interface tracking procedures result in smearing of the interface.

An alternative direction for handling boundary irregularities has been proposed by Szumbarski and Floryan [9] and is referred to as the immersed boundary conditions (IBC) method in the present work. The IBC method is conceptually similar to the IB methods as the physical field of interest is immersed inside the computational domain. However, unlike the IB method, the IBC method does not use additional forcing to impose the physical boundary conditions rather transforms the original boundary value problem into an internal value problem. The concept behind the IBC method is also different from the fictitious domain method as it does not simplify the geometry of the problem to enforce the boundary conditions [4]. The IBC method leads to a formal construction of boundary constraints that provide the required closing relations. Spatial discretization relies on the spectral expansions and thus provides ability to reach machine level accuracy. The boundary constraints rely on the representation of physical boundary

in the spectral space and nullifying the relevant Fourier modes. Such implementation is limited to geometries that can be represented by Fourier expansions but results in a gridless algorithm as all possible variations of boundary geometries are described in terms of the Fourier coefficients only. The programming effort associated with modeling of changes of geometry has been essentially removed as the only information required for specifying the new geometry is reduced to a set of Fourier coefficients provided as input to the code. The additional attractiveness of this concept is associated with the precise mathematical formalism, high accuracy and sharp identification of the location of physical boundaries. This method has been successfully extended to unsteady problems [10] as well as moving boundary problems [11, 12] where the boundary geometries are time-dependent. The computational advantage of this approach over conventional mapping-based spectral algorithm is more evident for the moving boundary problems [11, 12], because only the entries in the coefficient matrix corresponding to the discretized boundary conditions are required to be computed at each time step while part of the coefficient matrix corresponding to the discretized field equations needs to be constructed only once. The special structure of the coefficient matrices resulting from the algorithm also provides opportunities for devising more efficient iterative solution methods [10-12]. While the IBC method has been successfully employed for various classes of problems, it has limitations in terms of severity of boundary geometry that can be handled accurately. Various tests have shown that if either the wave number of the physical boundary corrugation or the amplitude of this corrugation is too large, the method fails to provide an acceptable accuracy [9-12].

This chapter addresses the limitations of the IBC method discussed above. The IBC method relies on the Galerkin projection for the construction of discretized analog of the field equations. Some of the projection equations are eliminated to provide "space" for the boundary conditions, which are imposed in the Tau-like manner. The boundary conditions are "discretized' using the IBC concept resulting in a number of boundary constraints that is far in excess of that required to formulate a closed system of algebraic equations. In the "classical" formulation of the IBC method [9], only the number of boundary constraints required to form a closed system is retained and boundary

constraints corresponding to the lowest (dominant) Fourier modes are used for this purpose. Use of additional available constraints could lead to an increase in the accuracy of the IBC method and could extend its applicability to more extreme geometries, but leads to an over-determined formulation of the problem. Since we have chosen to work with the over-determined formulation, we can also explore whether the use of all available projection equations offers any computational advantage.

The possible gains associated with the over-determined formulation of the IBC method could be problem dependent. In order to provide a definite answer, we have tested this formulation on three model problems involving most commonly found operators, i.e., the Laplace operator, the biharmonic operator and the Navier-Stokes equations. In each case, we have used the same class of geometries for testing purposes so that the reader can identify issues associated with the progressively more complicated operators. Section 4.2 discusses model geometry. Section 4.3 provides description of the method for the Laplace equation. Section 4.4 is devoted to the solution of a model problem that leads to a biharmonic operator. Section 4.5 provides discussion of the solution of the Navier-Stokes equations. Section 4.6 provides a short summary of the main conclusions. In order to provide reliable testing of the accuracy of the over-determined formulation, we have determined reference solutions by solving all three model problems using the mapping method that leads to the classical treatment of boundary conditions. A brief outline of the relevant algorithms is given in Appendices E, F, G for the Laplace, biharmonic and Navier-Stokes problems, respectively.

4.2. Model geometry

We select model geometry in the form of a two-dimensional slot extending to $\pm\infty$ in the x-direction and periodic with the wavelength $\lambda=2\pi/\alpha$ (see Fig.1). The slot is bounded by walls whose geometry is expressed in terms of Fourier expansions in the form

$$y_L(x)=-1+\sum_{n=-N_A}^{n=+N_A} H_L^{(n)} e^{in\alpha x}\,, \qquad y_U(x)=1+\sum_{n=-N_A}^{n=+N_A} H_U^{(n)} e^{in\alpha x}\,, \tag{4.2.1a,b}$$

where $H_L^{(n)}=\left(H_L^{(-n)}\right)^*$, $H_U^{(n)}=\left(H_U^{(-n)}\right)^*$ and the asterisk denotes complex conjugate. Such geometries are of interest in simulations of various physical phenomena where surface roughness plays important role, e.g., electrical micro-capacitors, micro-heat exchangers, laminar-turbulent transition, electrostatic filters, etc. We have selected three types of field equations that the reader might encounter in such applications, i.e., the Laplace equation (discussed in Section 4.3), the biharmonic equation (discussed in Section 4.4) and the Navier-Stokes equations (discussed in Section 4.5). These equations are linear second-order, linear fourth-order and nonlinear fourth-order, respectively, and thus provide ample opportunity for illustration of the performance of the algorithm.

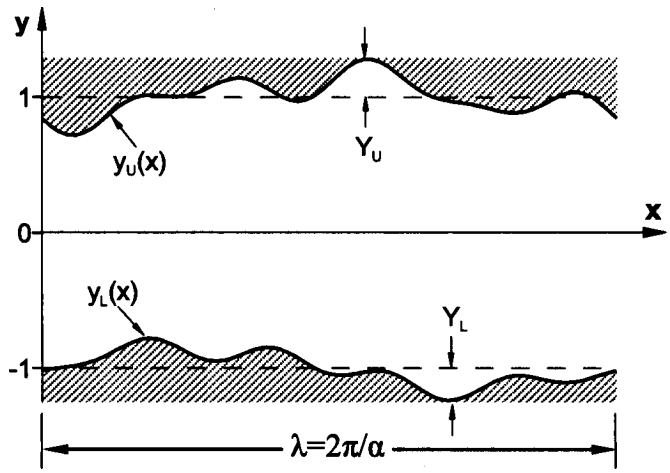


Figure 4.1. *Sketch of the domain of interest in the physical plane.*

4.3. Problems described by the Laplace equation

The Laplace equation governs different types of practical flow problems, e.g. conductive heat flow, ground-water hydrology, potential flow, etc. In our case we shall consider the

Laplace equation to be describing the conductive heat flow in a corrugated slot whose geometry has been defined in Section 4.2.

4.3.1. Problem formulation

The dimensionless field equation describing heat flow at steady state has the form

$$\nabla^2\theta = 0, \tag{4.3.1}$$

where $\nabla^2 = \partial_{xx} + \partial_{yy}$ is the Laplacian and the symbol ∂ denotes partial differentiation with subscripts x and y denoting the arguments of partial differentiations. Equation (4.3.1) needs to be supplemented by suitable boundary conditions. For convenience we select isothermal boundary conditions on both walls in the form

$$\theta(x, y_L(x)) = C_L = \text{const}, \qquad \theta(x, y_U(x)) = C_U = \text{const}, \tag{4.3.2a,b}$$

and the remaining boundary conditions follow from the assumption of periodicity of geometry. We shall now briefly describe the discretization process. The irregular wall geometry poses the main difficulty in solving problem (4.3.1)-(4.3.2). In order to determine solution with spectral accuracy, we use Fourier and Chebyshev expansions in the x- and y-directions, respectively. The physical domain is confined between $(1+Y_U)$ and $(-1-Y_L)$ which denote locations of extremities of the domain of interest (see Fig. 4.1). The first step in the solution process involves a mapping from the physical (x,y) coordinates to the computational $(x, \hat{y})$ coordinates in the form

$$\hat{y} = 2[y - (1 + Y_U)]/[1 + Y_U - (-1 - Y_L)] + 1, \tag{4.3.3}$$

which limits computational domain to $\hat{y} \in \langle -1,1 \rangle$ and permits the use of the standard definition of Chebyshev polynomials. The governing equation in the transformed

coordinates takes the form

$$\partial_{xx}\theta + \Gamma^2 \partial_{\hat{y}\hat{y}}\theta = 0, \tag{4.3.4}$$

where $\Gamma = 2/(2 + Y_U + Y_L)$. The boundaries in the $(x, \hat{y})$ plane become

$$\hat{y}_L(x) = \sum_{n=-N_A}^{n=+N_A} A_L^{(n)} e^{in\alpha x}, \qquad \hat{y}_U(x) = \sum_{n=-N_A}^{n=+N_A} A_U^{(n)} e^{in\alpha x}, \tag{4.3.5a,b}$$

where $A_L^{(0)} = 1 + \Gamma\left[-2 - Y_U + H_L^{(0)}\right]$, $A_L^{(n)} = \Gamma H_L^{(n)}$ for $n \neq 0$, $A_U^{(0)} = 1 + \Gamma\left[-Y_U + H_U^{(0)}\right]$, $A_U^{(n)} = \Gamma H_U^{(n)}$ for $n \neq 0$. The corresponding boundary conditions are

$$\theta(x, \hat{y}_L(x)) = C_L, \qquad \theta(x, \hat{y}_U(x)) = C_U. \tag{4.3.6}$$

The solution can be represented in the form of Fourier expansion

$$\theta(x, \hat{y}) = \sum_{n=-\infty}^{n=+\infty} \Theta^{(n)}(\hat{y}) e^{in\alpha x} \approx \sum_{n=-N_M}^{n=+N_M} \Theta^{(n)}(\hat{y}) e^{in\alpha x}, \tag{4.3.7a}$$

where $\Theta^{(n)}(\hat{y}) = \left(\Theta^{(-n)}(\hat{y})\right)^*$ and $\Theta^{(n)}$ satisfies the modal equations in the form

$$\Gamma^2 D^2 \Theta^{(n)} - n^2 \alpha^2 \Theta^{(n)} = 0 \tag{4.3.7b}$$

where $D^2 = d^2/d\hat{y}^2$, $n = 0, \ldots, N_M$. A minimum of two boundary conditions have to be extracted from (4.3.6) for each equation of type (4.3.7b) resulting in a total $2*(N_M+1)$ conditions required to close the system. These conditions take an explicit form for each modal equation in the case of smooth boundaries but all modes become coupled in the case of irregular geometry as will be shown later.

The unknown modal functions $\Theta^{(n)}(\hat{y})$ can be represented in terms of expansions based on Chebyshev polynomials in the form

$$\Theta^{(n)}(\hat{y}) = \sum_{k=0}^{k=\infty} Z_k^{(n)} T_k(\hat{y}) \approx \sum_{k=0}^{k=N_T} Z_k^{(n)} T_k(\hat{y}) \tag{4.3.8}$$

where T_k denotes the Chebyshev polynomial of k^{th} order and $Z_k^{(n)}$ are the unknown expansion coefficients. Substituting (4.3.8) into Eq.(4.3.7b) and applying the Galerkin projection technique, i.e., multiplying both sides of the resulting equations by $T_j(\hat{y})$ and integrating with the weight function $\hat{\omega} = 1/\sqrt{1-\hat{y}^2}$, we obtain

$$\sum_{k=0}^{k=N_T} \left[\left\langle T_j, D^2 T_k \right\rangle - \left(n^2\alpha^2 \right) \left\langle T_j, T_k \right\rangle \right] Z_k^{(n)} = 0 , \qquad j = 0,\ldots,N_T . \tag{4.3.9}$$

The inner products in Eq.(4.3.9) are defined as $\left\langle f_j(\hat{y}), g_k(\hat{y}) \right\rangle = \int_{-1}^{1} f_j(\hat{y}) g_k(\hat{y}) \hat{\omega}(\hat{y}) d\hat{y}$ and are evaluated using the well known orthogonality properties of Chebyshev polynomials. The reader may note that one can construct a maximum of $(N_T+1)*(N_M+1)$ equations of the type (4.3.9).

4.3.2. Enforcement of boundary conditions

The boundary conditions are to be enforced along the lines $\hat{y}_L(x)$ and $\hat{y}_U(x)$ while the solution domain remains fixed at $\hat{y} \in \langle -1,1 \rangle$. A computationally suitable form of boundary conditions will be constructed using the concept of Immersed Boundary Conditions (IBC). We start with explaining the process of evaluating the unknown temperature $\theta_f(x) \equiv \theta(x,f(x))$ along an arbitrary line, l:={(x,y): y=f(x)}, such that f is a periodic function with period $\lambda=2\pi/\alpha$ and $|f(x)|\leq 1$ which can be expressed as

$$f(x) = \sum_{n=-N_A}^{n=+N_A} P^{(n)} e^{in\alpha x} . \tag{4.3.10}$$

where all $P^{(n)}$'s are known. $\theta_f(x)$ is periodic with the same period λ and thus can be expressed in terms of Fourier series as

$$\theta_f(x) \equiv \theta(x, f(x)) = \sum_{n=-N_\theta}^{n=+N_\theta} \varphi^{(n)} e^{in\alpha x} \tag{4.3.11}$$

where the value of N_θ will be established later. Expressions for the expansion coefficients $\varphi^{(n)}$ can be determined by inserting the discretized form of solution into (4.3.11) resulting in

$$\theta_f(x) = \sum_{n=-N_M}^{n=+N_M} \Theta^{(n)}(f(x)) e^{in\alpha x} = \sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} Z_k^{(n)} T_k(f(x)) e^{in\alpha x} . \tag{4.3.12}$$

Since $T_k(f(x))$ is periodic in x, it can be expressed in terms of Fourier expansion as follows

$$T_k(f(x)) = \sum_{m=-N_S}^{m=N_S} w_k^{(m)} e^{im\alpha x} \tag{4.3.13}$$

where $\max(N_S)=N_T * N_A$. The expansion coefficients in (4.3.13) can be evaluated with the help of the recurrence relation for Chebyshev polynomials resulting in $w_0^{(0)} = 1,\ w_0^{(m)} = 0 \ \text{for} |m| \geq 1,\ w_1^{(m)} = P^{(m)} \ \text{for} |m| \geq 0$, and $w_{k+1}^{(m)} = 2 \sum_{n=-\infty}^{\infty} P^{(n)} w_k^{(m-n)} - w_{k-1}^{(m)}$.

Substitution of (4.3.13) into (4.3.12) gives

$$\theta_f(x) = \sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} \sum_{m=-N_S}^{m=+N_S} Z_k^{(n)} w_k^{(m)} e^{i(n+m)\alpha x} = \sum_{h=-N_\theta}^{h=+N_\theta} \sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} Z_k^{(n)} w_k^{(h-n)} e^{ih\alpha x} \tag{4.3.14}$$

where $h=n+m$ and $N_\theta=N_T N_A+N_M$. Comparison of (4.3.11) with (4.3.14) (with accounting for the change of labels in (4.3.14), i.e., $n\rightarrow m$, $h\rightarrow n$) gives

$$\varphi^{(n)} = \sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_k^{(m)} w_k^{(n-m)} \quad , \quad n \in 0,....,N_\theta \quad . \tag{4.3.15}$$

Equation (4.3.15) constitutes the basis for enforcing the boundary conditions along boundaries immersed inside the solution domain, i.e., along the boundary lines $\hat{y}_L(x)$ and $\hat{y}_U(x)$. In the case of our model problem, these boundary conditions lead to boundary constraints that take the following form

$$\sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_k^{(m)} \left(w_{L,U}\right)_k^{(n-m)} = \begin{cases} C_{L,U} & \text{for } n=0 \\ 0 & \text{for } n\neq 0 \end{cases} , \quad n=0,...,N_\theta \tag{4.3.16}$$

and the subscripts 'L' and 'U' denote lower and upper walls respectively. The reader may note that we have physical conditions at the boundaries that we refer to as boundary conditions, e.g., Eq.(4.3.6). The discretization of (4.3.6) leads to (4.3.16) which we shall refer to as the boundary constraints. The reader may also note that the assumed flow representation (4.3.7) is limited to N_M+1 Fourier modes that require $2*(N_M+1)$ of boundary constraints to form a closed system while Eq.(4.3.16) provides a larger pool, i.e., $2*(N_\theta+1)=2*(N_T N_A+N_M+1)$, of potential constraints that one can choose from. The discretization process results in $(N_M+1)*(N_T+1)$ projection equations and the same number of the unknown expansion coefficients $Z_k^{(m)}$. This provides certain flexibility for designing a suitable algorithm. Various options will be discussed below.

a. Classical IBC method

Two projection equations are eliminated for each Fourier mode (tau procedure) providing degrees of freedom required to enforce boundary conditions. The first $2*(N_M+1)$ conditions of type (4.3.16) are enforced at each wall resulting in a closed system of

equations. The values of the remaining $\varphi^{(n)}$'s for $n>(N_M+1)$ provide a measure of error in the enforcement of boundary conditions that may be used as a part of post-processing.

b. Pseudo-Classical IBC method

The number of the enforced boundary constraints of type (4.3.16) is increased to $2*(M_M+1)$, where $M_M>N_M$, and an equal number of projection equations is dropped resulting in a closed system of algebraic equations. In the special case of $M_M=N_M$, this method reduces to the classical IBC method.

c. Over-determined formulation of the IBC method - version 1

This method is based on the classical IBC method. The first N_T-1 projection equations for each Fourier mode is retained and the system of equations is made over-determined by enforcing $2*(M_M+1)$ boundary constraints of type (4.3.16) where $M_M>N_M$. In the special case of $M_M=N_M$ this formulation reduces to the classical IBC method.

d. Over-determined formulations of the IBC method - version 2

In this case, all projection equations for each Fourier mode are retained (i.e., N_T+1 equations) and $2*(M_M+1)$ of boundary constraints of type (4.3.16) are enforced, where $M_M>N_M$. A special variant of the over-determined formulation may be obtained when $M_M=N_M$.

4.3.3. Solution of the resulting system of equations

Let us assume that the final system of equations consisting of the projection equations and boundary constraints has the form

$$\mathbf{Ax=b} \tag{4.3.17}$$

where **A** is a p×q coefficient matrix, **x** is a q-dimensional column vector of unknowns and **b** is the known matrix-vector product which is a p-dimensional column vector. In

general, for the over-determined IBC problems, the matrix **A** has more rows than columns, i.e., p>q. But for the classical and pseudo-classical IBC methods, the matrix **A** is square, i.e., p=q.

The classical and pseudo-classical IBC problem can be solved using any standard technique, e.g., Gauss elimination, resulting in a unique solution. On the other hand, there are several methods that can be used to solve the over-determined system resulting in somewhat different solutions. We shall now discuss this issue in details. We wish to find a q-dimensional vector **x** such that it satisfies equation (4.3.17) where $\mathbf{A} \in \mathbf{C}^{p \times q}$, $\mathbf{b} \in \mathbf{C}^{p}$ and p>q. In general, the objective is to minimize the residual vector $\mathbf{r}=\mathbf{b}-\mathbf{A}\mathbf{x} \in \mathbf{C}^{p}$ although it may not be possible to make it equal to zero. The minimization can be done in the least squares sense, where the length of the residual is measured in terms of the 2-norm, i.e., $\mathbf{x} \in \mathbf{C}^{q}$ is obtained in a way that minimizes $\|\mathbf{r}\|_2 = \|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2$. If the rank h of matrix $\mathbf{A} \in \mathbf{C}^{p \times q}$ is full then the least squares solution to the over-determined problem is unique. So, one of the most important preconditions before solving an over-determined system is to make sure that the matrix $\mathbf{A} \in \mathbf{C}^{p \times q}$ has a full rank h=q, i.e., the matrix has q linearly independent columns and rows. The reader may note that a given full rank matrix may not be numerically full rank due to the limited length of the computer word.

The matrix $\mathbf{A} \in \mathbf{C}^{p \times q}$ and vector $\mathbf{b} \in \mathbf{C}^{p}$ have elements corresponding to the projection equations (4.3.9) and to the boundary constraints (4.3.16), respectively. We may define the residual in a new form as

$$\|\mathbf{r}_w\|_2 = \|\mathbf{r}_{PE}\|_2 + w_t \|\mathbf{r}_{BC}\|_2 \tag{4.3.18}$$

where $\|\mathbf{r}_{PE}\|_2$ is the residual corresponding to the projection equations, $\|\mathbf{r}_{BC}\|_2$ is the residual corresponding to the boundary constraints, w_t is the weight applied to the boundary constraints and $\|\mathbf{r}_w\|_2$ is the resultant weighted residual. When w_t>1, the boundary constraints become more important and, when solved in the least squares sense,

the norm $\|\mathbf{r}_{BC}\|_2$ is minimized. On the other hand, when $w_t<1$, the field equations become more dominant and the least squares solution minimizes the norm $\|\mathbf{r}_{PE}\|_2$. For the case $w_t=1$, the norm $\|\mathbf{r}_{PE}\|_2$ becomes equal to the regular norm $\|\mathbf{r}\|_2$. Unless otherwise stated, the default value of the weight factor $w_t=1$ is used in producing results presented in this chapter.

The general way of solving the least squares problem is to evaluate the pseudo-inverse or the generalized inverse $\mathbf{A}^+$ of the over-determined matrix $\mathbf{A} \in \mathbf{C}^{p\times q}$ such that

$$\mathbf{x}=\mathbf{A}^+\mathbf{b}. \tag{4.3.19}$$

Different methods may be employed to evaluate the pseudo-inverse. The three leading and most commonly used approaches are based on the

i. Normal equations (NE)

ii. QR factorization (QR), and

iii. Singular value decomposition (SVD).

Although other variants of these methods are possible, we shall limit our considerations only to those listed above.

i. Evaluation of pseudo-inverse using normal equations

If **A** has full rank then the easiest way to solve an over-determined system in the least squares sense is to solve the normal equations [Ref. 13] given by

$$\mathbf{A}^H\mathbf{A}\mathbf{x}=\mathbf{A}^H\mathbf{b} \tag{4.3.20}$$

where $\mathbf{A}^H$ is the conjugate transpose of **A**. The expression for the pseudo-inverse in this case becomes

$$\mathbf{A}^{+} = (\mathbf{A}^{H}\mathbf{A})^{-1}\mathbf{A}^{H} \in \mathbf{C}^{q\times p}. \tag{4.3.21}$$

The computational work involved in evaluating $\mathbf{A}^{+}$ is the smallest among the three methods considered. This method works better if A is a real matrix and it is a method of choice when the speed of calculations is the only consideration.

ii. Evaluation of pseudo-inverse using QR factorization

The most popular method for solving least squares problems is based upon the QR factorization [Ref. 14]. Here, the matrix $\mathbf{A} \in \mathbf{C}^{p\times q}$ is decomposed into a product of a unitary matrix $\mathbf{Q} \in \mathbf{C}^{p\times p}$ and an upper-triangular matrix $\mathbf{R} \in \mathbf{C}^{p\times q}$, i.e.,

$$\mathbf{A} = \mathbf{QR} = \mathbf{Q}\begin{pmatrix}\mathbf{R}_1\\ \mathbf{0}\end{pmatrix} \tag{4.3.22}$$

where $\mathbf{R}_1 \in \mathbf{C}^{q\times q}$ is an upper-triangular matrix. The pseudo-inverse $\mathbf{A}^{+}$ takes the form

$$\mathbf{A}^{+} = (\mathbf{R}_1)^{-1}\mathbf{Q}^{H} \in \mathbf{C}^{q\times p}. \tag{4.3.23}$$

iii. Evaluations of pseudo-inverse using SVD

The singular value decomposition (SVD) of a matrix $\mathbf{A} \in \mathbf{C}^{p\times q}$ is the most reliable method to evaluate the pseudo-inverse of the matrix A [Ref. 11,12], but at the same time, it is the most computationally expensive method. According to the SVD theorem, for any matrix $\mathbf{A} \in \mathbf{C}^{p\times q}$ of rank h there exists unitary matrices $\mathbf{U} \in \mathbf{C}^{p\times p}$ and $\mathbf{V} \in \mathbf{C}^{q\times q}$ such that

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{H}, \quad \Sigma = \begin{pmatrix}\Sigma_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}, \tag{4.3.24a}$$

where $\Sigma \in \mathbf{C}^{p\times q}$, $\Sigma_1 = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \ldots\ldots, \sigma_h)$ and

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots\ldots\ldots \geq \sigma_h > 0 \qquad (4.3.24b)$$

where σ_i are the singular values of **A**. The pseudo-inverse in this case becomes

$$\mathbf{A}^{+} = \mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{U}^{H}. \qquad (4.3.25)$$

As stated before, in order to obtain an accurate solution of an over-determined system by computing the pseudo-inverse using any method, it is imperative to ensure that the matrix $\mathbf{A} \in \mathbf{C}^{p\times q}$ has numerically full rank h=q where p>q. There are numerous techniques to compute the rank of an over-determined matrix. The most reliable method is based on SVD. In an ideal case, the number of nonzero singular values of a matrix is equal to the rank of the matrix. But due to accumulation of the round-off error in the process of computing the singular values, a tolerance greater than machine epsilon may need to be selected and then the number of singular values greater than the selected tolerance is taken to be the numerical rank of the matrix. In this analysis we have selected the frequently used definition of the tolerance in the form

$$\text{tol} = p\times\sigma_1\times\varepsilon \qquad (4.3.26)$$

where p is the maximum size of the matrix **A**, σ_1 stands for the maximum singular value of the matrix **A** and ε denotes the machine epsilon.

Any matrix $\mathbf{A} \in \mathbf{C}^{p\times q}$ with rank h<q (where p>q) is called a rank-deficient matrix. Although the SVD method is applicable for a matrix with any numerical rank, the accuracy of the solution will be smaller if the matrix is rank-deficient. Our results suggest that rank-deficiency may arise due to the fact that the elements in the coefficient matrix corresponding to the discretized boundary constraints are smaller in magnitudes than those corresponding to the projection equations. The difference in the magnitudes of the boundary-constraints-entries and the projection-equations-entries increases as the

channel geometry becomes more challenging, i.e., when the corrugation amplitude and the corrugation wave numbers increase and/or when the specification of boundary geometry involves increasing number of Fourier modes. Weighting the boundary constraints with a suitable weight-factor, as shown in Eq. (4.3.18), has potential to reduce the numerical rank-deficiency problem should there be any, and consequently, may improve the accuracy. Excessive weight factor, on the other hand, may reduce the overall accuracy of the solution. The importance of the weight factor will be explained in detail with the help of numerical results to be discussed in the next section.

4.3.4. Numerical tests

This section provides discussion of results of numerical tests carried out with various versions of the over-determined formulation of the IBC method. For convenience, we have assumed that the slot has flat upper wall and a wavy lower wall, i.e., its geometry is defined as

$$y_L(x) = -1 + S\cos(\alpha x), \qquad y_U(x) = 1, \tag{4.3.27a,b}$$

where S denotes the corrugation amplitude and α denotes the corrugation wave number. Both walls are isothermal, i.e., $\theta(y_L(x))=0$ and $\theta(1)=1$. In order to have a meaningful discussion of error, we have produced a machine-accurate reference solution using the Domain Transformation (DT) method where the irregular flow domain is analytically mapped into a regular computational domain resulting in the classical form of boundary conditions. We have used the same spatial discretization as in the case of the IBC method, i.e., Fourier expansions in the x-direction and expansions based on the Chebyshev polynomials in the y-direction. Details of the DT method can be found in Appendix E.

We define the error in the evaluation of θ at any (x,y) location as

$$\theta_{er}(x,y) = \theta_{IBC}(x,y) - \theta_{DT}(x,y), \tag{4.3.28}$$

where θ_{IBC} stands for the θ obtained using any version (classical or over-determined) of the IBC method and θ_{DT} denotes the solution obtained for the same physical problem using Domain Transformation (DT) method (with sufficient number of Fourier modes and Chebyshev polynomials to provide machine accuracy). Norm defined as

$$\left\|\theta_{er}\right\|_{\infty} = \sup_{\substack{0\le x\le\lambda \\ y_L(x)\le y\le y_U(x)}} \left|\theta_{er}(x,y)\right| \tag{4.3.29}$$

is used to compare accuracy of different versions of the IBC method. The error in the enforcement of physical boundary conditions along the lower wall $y_L(x)$ is defined as

$$\theta_{er,bc}(y_L(x)) = \theta_{IBC}(y_L(x)) \tag{4.3.30a}$$

and norm defined as

$$\left\|\theta_{er,bc}\right\|_{\infty} = \sup_{0\le x\le\lambda} \left|\theta_{er,bc}(x)\right| \tag{4.3.30b}$$

is used to compare accuracy in the enforcement of the boundary conditions using different versions of the method. The Fourier spectra of the boundary error are expressed as

$$\theta_{er,bc}(y_L(x)) = \sum_{n=-\infty}^{n=+\infty} \phi_{er,bc}^{(n)} e^{in\alpha x} \tag{4.3.31}$$

and provide useful information about the character and spatial distribution of this error.

We shall start our discussion with the pseudo-classical IBC method. Figure 4.2 shows variations of the accuracy of this method as a function of the corrugation amplitude for the corrugation wave number α=5. Accuracy of the classical IBC method is displayed for comparison purposes. It is evident that the pseudo-classical method is not capable of providing accuracy improvement with respect to the classical method and that its accuracy actually deteriorates as the value of M_M increases, i.e., when more boundary constraints are added at the expense of the equal numbers of projection equations. It can also be seen that the maximum error occurs inside the flow domain while for the classical IBC method it occurs at the corrugated boundary. The difference between the maximum error and the error at the boundary increases significantly for higher values of M_M.

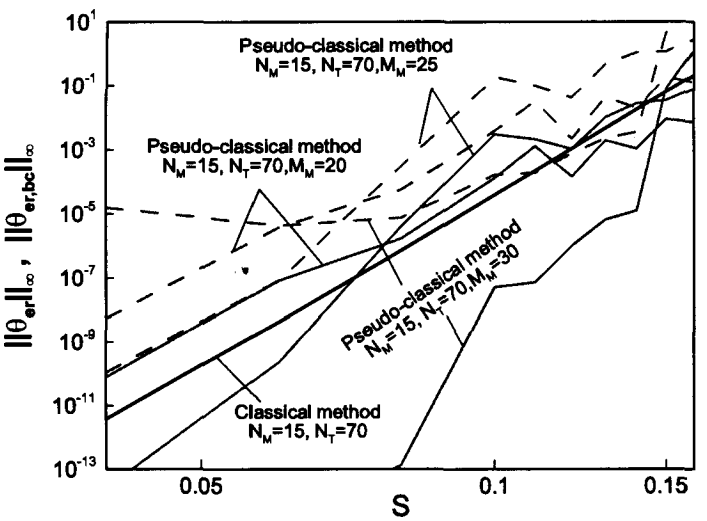


Figure 4.2. *Variations of the* $\|\theta_{er}(x,y)\|_\infty$ *(dash lines) and* $\|\theta_{er,bc}(x)\|_\infty$ *(solid lines) norms for the Laplace equation (Section 4.3) for the slot geometry defined by Eq. (4.3.27) as a function of the corrugation amplitude S for the corrugation wave number* α*=5.0 determined using the pseudo-classical IBC method with a different number of boundary constraints* $2*(M_M+1)$*. The reference solution has been determined using the domain transformation (DT) method with* N_M*=30 Fourier modes and* N_T*=100 Chebyshev polynomials. Results obtained using the classical IBC method are shown for comparison purposes (the maximum error for this method occurs at the boundary).*

Figures 4.3A and 4.3B show variations of the error resulting from the use of different solution methods for the system of equations resulting from the first version of the over-

determined formulation of the IBC method. The pseudo-inverse computed using the normal equations (Eq. 4.3.21), QR factorization (Eq. 4.3.23) and SVD (Eq. 4.3.25) are found to be at a similar level for higher values of wave amplitudes and wave numbers, and perform significantly better when compared to the classical IBC method for the equal number of Fourier modes N_M used to represent the solution. Since similar results were also observed for the more complex model problems to be discussed in Sections 4.4 and 4.5, all results presented from now on will be based on the SVD method unless explicitly stated otherwise. Additional argument in favor of the SVD method is the fact that this method is the most reliable method for computing the rank of a matrix and is the suggested method for evaluation of the pseudo-inverse if the coefficient matrix has potential of becoming numerically rank deficient [Ref. 13, 14].

Figures 4.3A and 4.3B also reveal that in the case of the classical IBC method there exists a certain critical combination (critical zone) of the corrugation wave number α and the corrugation amplitude S above which the method is unable to provide acceptable results. The reader may note that an increase in the number of Fourier modes and Chebyshev polynomials actually deteriorates accuracy of the solution for the conditions above this critical zone, but improves accuracy below this zone (as one would expect when "grid" resolution is increased). The over-determined formulation improves accuracy for high values of the corrugation amplitude S and thus extends the range of applicability of the IBC method – results displayed in Fig. 4.3 show more than two orders of magnitude accuracy improvement for S=0.125 and for α=5.0 (i.e., for parameter values corresponding approximately to the critical zone).

Analysis of accuracy of both versions of the over-determined formulation of the IBC method shows that these versions deliver the same accuracy, i.e., keeping additional projections equations in the computations does not provide any advantage under the conditions tested in this work. All further results discussed in this chapter have been obtained using the first version of the over-determined formulation. It should be understood that the results shown apply to both formulations.

Results displayed in Fig. 4.4A and 4.4B demonstrate that the accuracy systematically increases as the number of additional boundary constraints $2*(M_M-N_M)$ used in the computations increases, eventually reaching a saturation point after which the accuracy cannot be improved any more. The optimal value appears to be around M_M~25 for the conditions used in this particular test. Improvements obtained using M_M=30 over those obtained using M_M=25 are insignificant especially below the critical zone and thus use of $M_M \geq 30$ is not justified due to the increase of the computational costs associated with the increase of the size of system of equations. The user needs to choose the value of M_M carefully in order to make optimal use of computing resources. Numerous tests carried out as a part of this work suggest that the optimum range of M_M is approximately 30% to 40% higher than the number of Fourier modes N_M used to represent the solution.

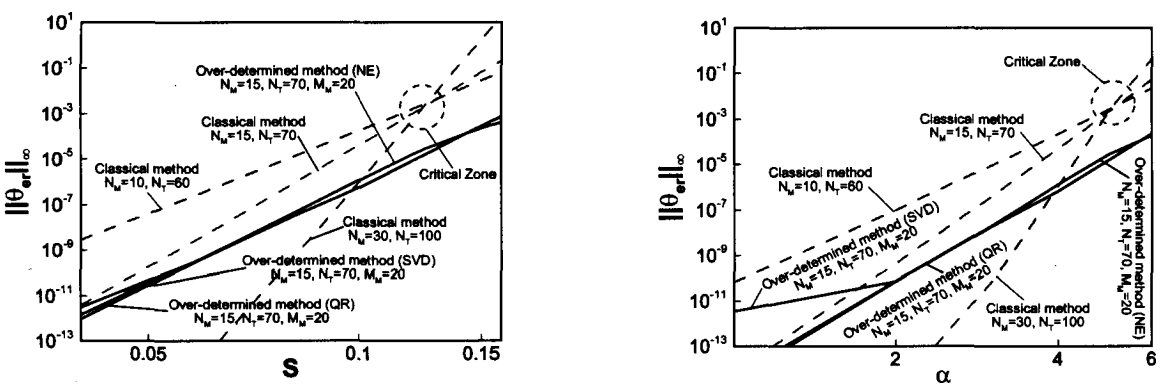


Figure 4.3. *Variations of the* $\|\theta_{er}(x,y)\|_\infty$ *norm as a function of the corrugation amplitude S (Fig. 4.3A with α=5.0) and as function of α (Fig. 4.3B with S=0.125) for the Laplace equation (Section 4.3) solved using the over-determined formulation (version 1) of the IBC method with* $2*(M_M+1)$ *number of boundary constraints for the slot geometry defined by Eq. (4.3.27). Solutions have been determined using different methods to evaluate the pseudo-inverse, i.e., NE-normal equations, QR – QR factorization, SVD – singular value decomposition. The reference solution has been determined using the domain transformation (DT) method with* N_M=30 *Fourier modes and* N_T=100 *Chebyshev polynomials. Results obtained using the classical IBC method are shown for comparison purposes (dash lines).*

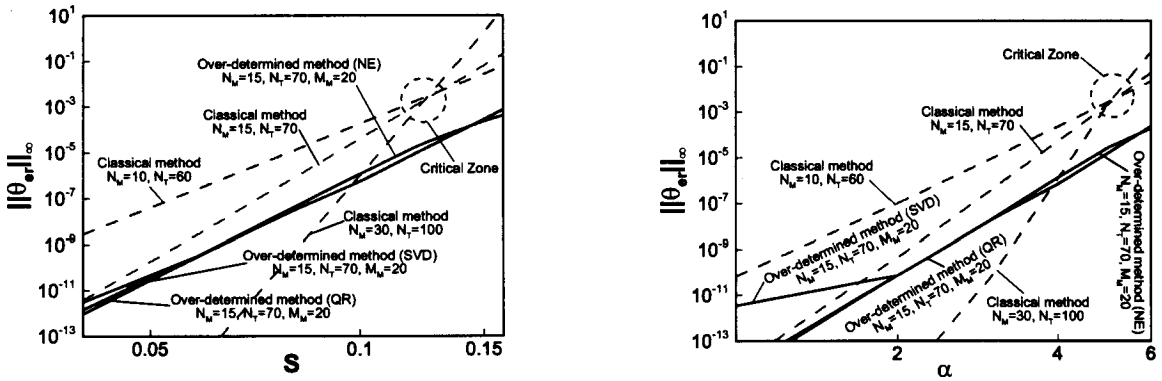


Figure 4.4. *Variations of the* $\|\theta_{er}(x,y)\|_\infty$ *norm as a function of the amplitude S with the corrugation wave number α=5.0 (Fig. 4.4A) and as a function of the wave number α with the amplitude S=0.125 (Fig. 4.4B) for the Laplace equation (Section 4.3) with the slot geometry defined by Eq. (4.3.27) for different number of boundary constraints 2*(M_M+1) used in the computations. The pseudo-inversed was computed using the QR-factorization. The reference solution has been determined using the domain transformation (DT) method with N_M=30 Fourier modes and N_T=100 Chebyshev polynomials. Results obtained using the classical IBC method are shown for comparison purposes (dash line).*

Results displayed in Fig. 4.4A and 4.4B also demonstrate that accuracy cannot be arbitrarily improved by increasing the number of additional boundary constraints 2*(M_M-N_M) used in the computations. The pseudo-inverses in these cases have been computed using the QR factorization technique. The optimal value appears to be around M_M~20 for the conditions used in this particular test. Improvements obtained using M_M=25 over those obtained using M_M=20 are insignificant especially below the critical zone and thus use of $M_M \geq 25$ is not justified due to the increase of the computational costs associated with the increase of the size of system of equations. The user needs to choose the value of M_M carefully in order to make optimal use of computing resources. Numerous tests carried out as a part of this work suggest that the optimum range of M_M is approximately 30% to 40% higher than the number of Fourier modes N_M used to represent the solution.

The distribution of the error along the lower boundary is displayed in Figure 4.5 for the wave number and the amplitude of the corrugation corresponding to the critical zone (see Fig. 4.3). It can be seen that the error is oscillatory in space with the amplitude of the oscillations significantly changing along the boundary for the classical IBC method, and with the location of the maximum amplitude overlapping with the location of the maximum slot opening. Error for the over-determined formulation remains oscillatory but the amplitude does not significantly change along the length of the slot. For the particular test case shown in Figure 4.5, the amplitude of the error changes by about five orders of magnitude along the length of the slot for the classical IBC method but only by about an order of magnitude for the over-determined method solved.

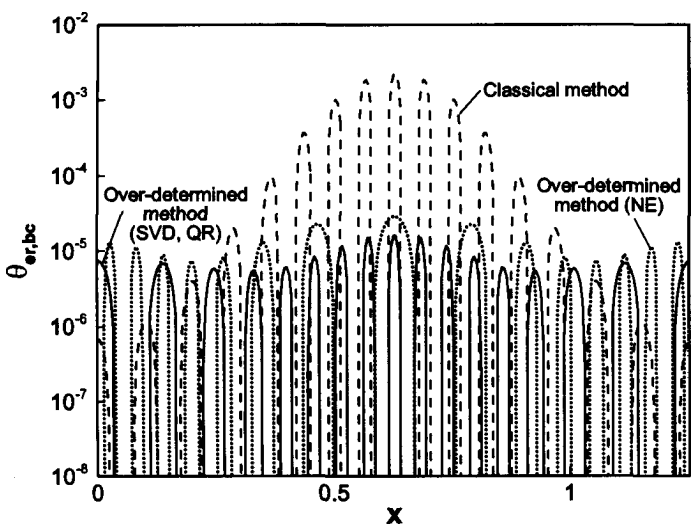


Figure 4.5. *Distribution of the boundary error $\theta_{er,bc}(y_L(x))$ along the corrugated wall for the Laplace equation (Section 4.3) for the slot geometry defined by Eq. (4.3.27) with the corrugation wave number α=5.0 and the corrugation amplitude S=0.125 evaluated using the over-determined formulation (version 1) of the IBC method with N_M=15 Fourier modes, N_T=70 Chebyshev polynomials and 2*(M_M+1)=42 boundary constraints used in the computations with the resulting system solved using the SVD, QR and NE methods. Results determined using the classical IBC method with the same number of Fourier modes and Chebyshev polynomials are shown for comparison purposes.*

The Fourier spectra of the error for the same test conditions (as presented in Fig. 4.5) are shown in Figure 4.6 and demonstrate that the first N_M+1 of Fourier modes have been eliminated in the case of the classical IBC method and that the largest error is associated with the first few Fourier modes omitted from the enforcement of boundary conditions. The spectrum of error in the case of the over-determined formulation is more uniform and is spread over all Fourier modes involved in the computations. Figure 4.6 also shows that the maximum error associated with any Fourier mode in the spectrum produced by the over-determined formulation is more than two orders of magnitude smaller than the maximum error associated with any Fourier mode produced by the classical formulation. These results suggest that the overall improvement in the accuracy of the over-determined formulation results from a more uniform spatial distribution of the error, e.g., Fig. 4.5, and from the distribution of the error among larger number of Fourier modes used to enforce boundary conditions, e.g., Fig. 4.6.

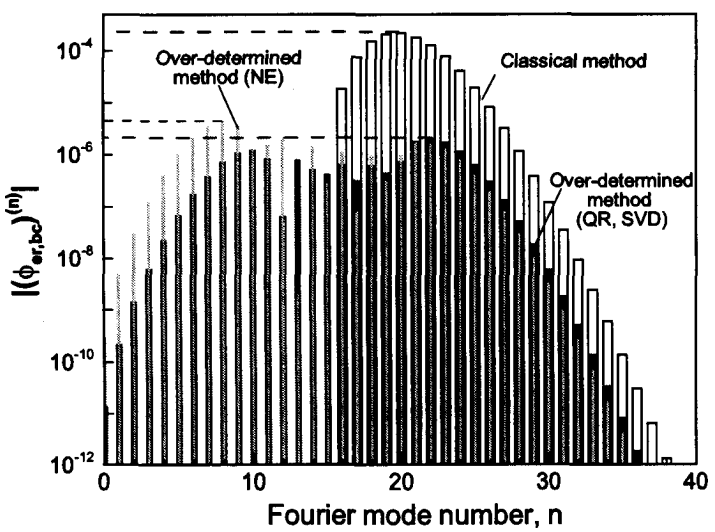


Figure 4.6. *Fourier spectra of the boundary error $\theta_{er,bc}(y_L(x))$ for the Laplace equation (Section 4.3) for the slot geometry defined by Eq. (4.3.27) with the corrugation wave number α=5.0 and the corrugation amplitude S=0.125 evaluated using the over-determined formulation (version 1) of the IBC method with N_M=15 Fourier modes, N_T=70 Chebyshev polynomials and 2*(M_M+1)=42 boundary constraints used in the computations with the resulting system solved using the SVD, QR and NE methods. Results determined using the classical IBC method with the same number of Fourier modes and Chebyshev polynomials are shown for comparison purposes.*

Variations of the error and the rank of the coefficient matrix as a function of the weight factor for the over-determined formulation are shown in Figure 4.7. It can be seen that the coefficient matrices for the model problem always have full rank (which is 2201 for this particular test). The error remains constant as the weight factors increases up to a certain limit, above which the error in the field begins to grow while the error in the enforcement of physical boundary conditions begins to decrease. Increase of the number of boundary constraints $2*(M_M+1)$ used in the computations increases the overall accuracy but does not change the qualitative character of variations of the error as a function of the weight factor.

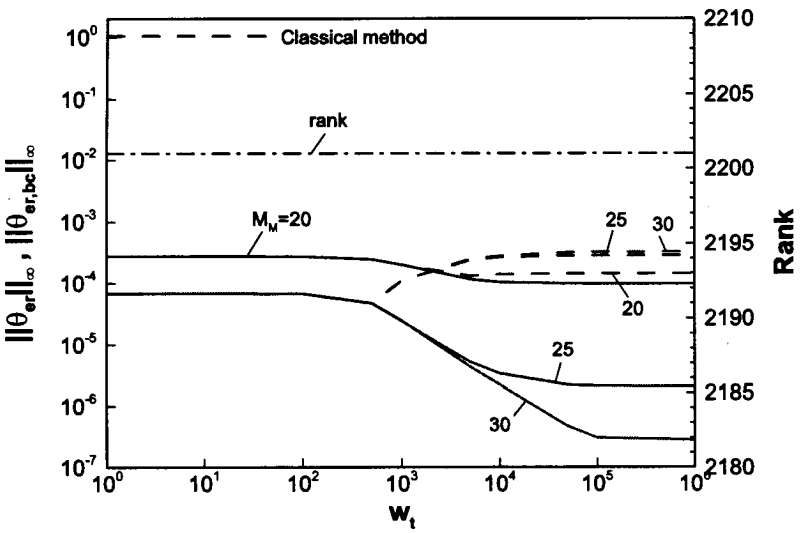


Figure 4.7. *Variations of the* $\|\theta_{er}(x,y)\|_\infty$ *(dash lines) and* $\|\theta_{er,bc}(x)\|_\infty$ *(solid lines) norms, and of the rank of the coefficient matrix (dash-dot line), as a function of the weight factor for the Laplace equation (Section 4.3) for the slot geometry defined by Eq. (4.3.27) with the corrugation amplitude S=0.15 and the corrugation wave number α=5.0 for the over-determined formulation (version 1) of the IBC method. The solutions were determined with* N_M*=15 Fourier modes,* N_T*=70 Chebyshev polynomials and with a different number* $2*(M_M+1)$ *of boundary constraints. The reference solution has been determined using the DT method with* N_M*=30 Fourier modes and* N_T*=100 Chebyshev polynomials. Accuracy of the classical IBC method used with the same number of Fourier modes and Chebyshev polynomials is shown for comparison purposes.*

The different aspects of the spectral accuracy of the classical IBC method have been investigated earlier in the context of both fixed [9, 10] and moving boundary problems [11, 12]. The discretization in the y-direction based on the Chebyshev expansions is guaranteed to be spectrally accurate with the increasing number of Chebyshev polynomials N_T used in the computations while the accuracy of discretization in the x-direction remains to be demonstrated. Figure 4.8 shows the variation of the error for the over-determined formulation of the IBC method as a function of the number of Fourier modes N_M used in computations. It can be seen that the error decreases exponentially with an increase of N_M and the magnitude of contributions of higher modes to the complete solution also decreases exponentially, similarly as in the case of the classical IBC formulation [9-12]. The over-determined IBC formulation retains spectral accuracy also in the case of more complex operators considered later in this work.

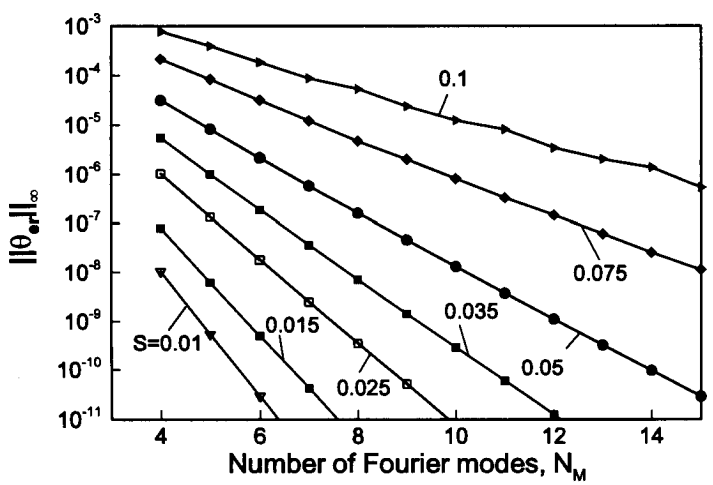


Figure 4.8. *Variations of the $\|\theta_{er}(x,y)\|_\infty$ norm as a function of the number of Fourier modes N_M used in the computations for the Laplace equation (Section 4.3) for the slot geometry defined by Eq. (4.3.27) with the corrugation wave number α=5.0 and different corrugation amplitudes S for the over-determined formulation (version 1) of the IBC method. The solutions were determined using N_T=70 Chebyshev polynomials and different number 2*(M_M+1) of boundary constraints, where M_M was selected to be approximately 30% higher than N_M. The reference solutions have been determined using the DT method with N_M=30 Fourier modes and N_T=100 Chebyshev polynomials.*

4.4. Problems described by the biharmonic equation

The biharmonic equation describes low Reynolds number flows as well as a number of problems in the theory of elasticity. This equation represents a convenient test problem that involves higher derivatives and requires more boundary conditions. In our case, we shall formulate the model problem in the context of Stokes flow in a corrugated slot whose geometry has been defined in Section 4.2.

4.4.1. Problem formulation

The dimensionless field equation has the form

$$\nabla^4 \Psi = 0 \tag{4.4.1}$$

where $\nabla^4 = [\partial_{xxxx} + 2\partial_{xxyy} + \partial_{yyyy}]$ is the biharmonic operator and Ψ denotes the stream function, which is defined as

$$u(x,y) = u_0(y) + u_1(x,y) = D\Psi_0 + \partial_y \Psi_1 = \partial_y \Psi, \tag{4.4.2a}$$

$$v(x,y) = v_1(x,y) = -\partial_x \Psi_1 = -\partial_x \Psi \tag{4.4.2b}$$

where D=d/dy. Here u(x,y) and v(x,y) denote the total velocities in x- and y-directions, respectively, $u_1(x,y)$ and $v_1(x,y)$ denote velocity modifications due to the presence of boundary corrugation, the reference flow in a smooth channel has the form of Poiseuille flow, i.e., $u_0(y)=1-y^2$, $\Psi(x,y)$ stands for the total stream function, $\Psi_1(x,y)$ denotes the modification of the stream function due to the boundary irregularities, and $\Psi_0=-y^3/3+y+2/3$ is the stream function of the reference flow. The problem formulation needs to be supplemented with suitable boundary conditions, which are taken to be in the following form

$$u_L(x) = u_0(y_L(x)) + u_1(x, y_L(x)) = D\Psi_0(y_L(x)) + \partial_y \Psi_1(x, y_L(x)) = 0\,, \tag{4.4.3a}$$

$$u_U(x) = u_0(y_U(x)) + u_1(x, y_U(x)) = D\Psi_0(y_U(x)) + \partial_y \Psi_1(x, y_U(x)) = 0\,, \tag{4.4.3b}$$

$$v_L(x) = v_1(x, y_L(x)) = -\partial_x \Psi_1(x, y_L(x)) = 0\,, \tag{4.4.3c}$$

$$v_U(x) = v_1(x, y_U(x)) = -\partial_x \Psi_1(x, y_U(x)) = 0\,, \tag{4.4.3d}$$

where the subscripts L and U refer to the lower and upper walls, respectively. Since boundary corrugations are periodic in x, the flow modifications will also be periodic and this leads to periodic boundary conditions for the flow quantities as well as an additional flow constraint which is typically taken to be either in the form of the mass flow constraint or the pressure gradient constraint. We shall return to this issue later in the text.

We want to formulate the problem for the stream function modification Ψ_1. The substitution of $\Psi(x,y)=\Psi_0(y)+\Psi_1(x,y)$ into (4.4.1) as well as application of the transformation (4.3.3) that makes the computational domain compliant with the standard definition of Chebyshev polynomials result in

$$[\partial_{xxxx} + 2\Gamma^2 \partial_{xx\hat{y}\hat{y}} + \Gamma^4 \partial_{\hat{y}\hat{y}\hat{y}\hat{y}}]\Psi_1 = 0\,. \tag{4.4.4}$$

Locations of the transformed boundaries are given by Eq. (4.3.5) and the boundary conditions corresponding to the transformed domain become

$$\partial_{\hat{y}} \Psi_1(x, \hat{y}_L(x)) = -D\Psi_0(\hat{y}_L(x))\,, \qquad \partial_{\hat{y}} \Psi_1(x, \hat{y}_U(x)) = -D\Psi_0(\hat{y}_U(x))\,, \tag{4.4.5a-b}$$

$$\Psi_1(x, \hat{y}_L(x)) = (\text{constant})_L\,, \qquad \Psi_1(x, \hat{y}_U(x)) = (\text{constant})_U\,, \tag{4.4.5c-d}$$

where $D = d/d\hat{y}$. In order to obtain solution to the problem (4.4.4)-(4.4.5) with spectral accuracy we shall apply the same spatial discretization technique as for the model problem discussed in Section 4.3, i.e., Fourier expansion in the x-direction (which satisfies periodicity conditions directly) and Chebyshev expansion in the $\hat{y}$-direction. The former one leads to

$$\Psi_1(x,\hat{y}) = \sum_{n=-\infty}^{n=+\infty} \Phi^{(n)}(\hat{y}) e^{in\alpha x} \approx \sum_{n=-N_M}^{n=+N_M} \Phi^{(n)}(\hat{y}) e^{in\alpha x} \tag{4.4.6a}$$

where $\Phi^{(n)}(\hat{y}) = \left(\Phi^{(-n)}(\hat{y})\right)^*$ and results in the modal equations in the form

$$(\Gamma^4 D^4 - 2n^2\alpha^2\Gamma^2 D^2 + n^4\alpha^4)\Phi^{(n)} = 0, \quad n=0,\ldots,N_M \tag{4.4.6b}$$

with each of them requiring four boundary conditions. The latter one leads to

$$\Phi^{(n)}(\hat{y}) = \sum_{k=0}^{k=\infty} G_k^{(n)} T_k(\hat{y}) \approx \sum_{k=0}^{k=N_T} G_k^{(n)} T_k(\hat{y}) \tag{4.4.7}$$

where T_k denotes the Chebyshev polynomial of k^{th} order and $G_k^{(n)}$ are the unknown coefficients of the expansions. Substituting (4.4.7) into (4.4.6b) and applying the Galerkin projection technique, we obtain a system of $(N_T+1)*(N_M+1)$ projection equations in the form

$$\sum_{k=0}^{k=N_T} \left[\Gamma^4 \langle T_j, D^4 T_k\rangle - \left(2n^2\alpha^2\Gamma^2\right)\langle T_j, D^2 T_k\rangle + \left(n^4\alpha^4\right)\langle T_j, T_k\rangle\right] G_k^{(n)} = 0,$$

$$j = 0,\ldots,N_T \;,\; n = 0,\ldots,N_M . \tag{4.4.8}$$

The discretized form of the boundary conditions are to be derived using the concept of immersed boundary conditions (IBC). The outline of this method has been provided in

Section 4.3, however, we shall repeat basic steps as there are sufficient differences that need to be explained.

One needs to evaluate $\partial_{\hat{y}}\Psi_1(x, f(x))$ and $\Psi_1(x, f(x))$ along an arbitrary line inside the computational domain $l:=\{(x, \hat{y}): \hat{y}=f(x)\}$, such that f is a periodic function with period $\lambda = 2\pi/\alpha$ and $|f(x)| \leq 1$. Both quantities, i.e., $\partial_{\hat{y}}\Psi_1(x, f(x))$ and $\Psi_1(x, f(x))$, are periodic in x with the same period λ and thus can be expressed in terms of Fourier series as

$$\partial_{\hat{y}}\Psi_1(x, f(x)) = \sum_{n=-N_\theta}^{n=+N_\theta} U^{(n)} e^{in\alpha x}, \qquad \Psi_1(x, f(x)) = \sum_{n=-N_\theta}^{n=+N_\theta} V^{(n)} e^{in\alpha x}, \qquad (4.4.9a,b)$$

and can be evaluated using the discretized form of the solution as

$$\partial_{\hat{y}}\Psi_1(x, f(x)) = \Gamma \sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} G_k^{(n)} DT_k(f(x)) e^{in\alpha x},$$

$$\Psi_1(x, f(x)) = \sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} G_k^{(n)} T_k(f(x)) e^{in\alpha x}. \qquad (4.4.10a,b)$$

Chebyshev polynomials and their derivatives evaluated at the line $f(x)$, i.e., $T_k(f(x))$ and $DT_k(f(x))$, are periodic functions of x and thus can be expressed in terms of Fourier expansions as

$$T_k(f(x)) = \sum_{m=-N_S}^{m=+N_S} w_k^{(m)} e^{im\alpha x}, \qquad DT_k(f(x)) = \sum_{m=-N_S}^{m=+N_S} d_k^{(m)} e^{im\alpha x}, \qquad (4.4.11a,b)$$

where $\max(N_S)=N_T*N_A$ and coefficients $w_k^{(m)}$ and $d_k^{(m)}$ are known from the properties of Chebyshev polynomials. Substituting (4.4.11) into (4.4.10), one obtains

$$\partial_{\hat{y}}\Psi_1(x,f(x))=\Gamma\sum_{n=-N_\theta}^{n=+N_\theta}\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_k^{(m)}d_k^{(n-m)}e^{in\alpha x},$$

$$\Psi_1(x,f(x))=\sum_{n=-N_\theta}^{n=+N_\theta}\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_k^{(m)}w_k^{(n-m)}e^{in\alpha x}. \qquad (4.4.12a,b)$$

where $N_\theta=N_TN_A+N_M$. Comparison of (4.4.12) and (4.4.9) gives

$$U^{(n)}=\Gamma\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_k^{(m)}d_k^{(n-m)}, \qquad V^{(n)}=\sum_{n=-N_M}^{n=+N_M}\sum_{k=0}^{k=N_T}G_k^{(m)}w_k^{(n-m)}. \qquad (4.4.13a,b)$$

$D\Psi_0(f(x))$ is known and can be expressed in terms of Fourier expansion as

$$D\Psi_0(f(x))=\sum_{n=-N_\theta}^{n=+N_\theta}F^{(n)}e^{in\alpha x}. \qquad (4.4.14)$$

The final form of boundary constraints can be written as

$$U_{L,U}^{(n)}=\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_k^{(m)}\left(d_{L,U}\right)_k^{(n-m)}=-\Gamma^{-1}F_{L,U}^{(n)}, \quad \text{for } |n|\geq 0 \qquad (4.4.15a)$$

$$V_{L,U}^{(n)}=\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_k^{(m)}\left(w_{L,U}\right)_k^{(n-m)}=0, \quad \text{for } n\neq 0 \qquad (4.4.15b)$$

$$V_{L,U}^{(0)}=\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}G_k^{(m)}\left(w_{L,U}\right)_k^{(-m)}=\left(\text{constant}\right)_{L,U}, \qquad (4.4.15c)$$

where $n=0,\ldots,N_\theta$ and the subscripts L and U denote lower and upper walls respectively. The form of constants that appear in Eq. (4.4.15c) can be determined using either the constant volume flux or the constant pressure gradient constraint. For convenience, we have selected the former one for this presentation, i.e., we assumed that the flow rates in

the smooth reference slot and in the corrugated slot remain the same. Volume flux Q can be evaluated by integrating the x-velocity component across the slot, i.e.,

$$Q(x)=\int_{\hat{y}_L(x)}^{\hat{y}_U(x)}\left(\frac{\partial\Psi}{\partial\hat{y}}\right)d\hat{y}=\Psi_0(\hat{y}_U(x))+\Psi_1(x,\hat{y}_U(x))-\Psi_0(\hat{y}_L(x))-\Psi_1(x,\hat{y}_L(x)) \qquad (4.4.16)$$

where the $\Psi_0(\hat{y}_U(x))$ and $\Psi_0(\hat{y}_L(x))$ are known periodic functions and can be expressed as Fourier series in the form

$$\Psi_0(\hat{y}_{L,U}(x))=\sum_{n=-M_M}^{n=+M_M}\Xi_{L,U}^{(n)}e^{in\alpha x}\,. \qquad (4.4.17)$$

The volume flux represents an x-periodic function that can be written in the form of a Fourier expansion

$$Q(x)=\sum_{n=-N_M}^{n=+N_M}Q^{(n)}e^{in\alpha x} \qquad (4.4.18)$$

where the zero term, i.e., $Q^{(0)}$, represents the net mass flux along the channel. Substituting Eq. (4.4.5c,d), (4.4.17) and (4.4.18) into Eq. (4.4.16) and extracting the zero mode results in

$$Q^{(0)}=(\text{constant})_U-(\text{constant})_L+\Xi_U^{(0)}-\Xi_L^{(0)}. \qquad (4.4.19)$$

As the definition of stream function is accurate up to a constant, one of the constants in Eq. (4.4.15c) can be assumed arbitrarily and other has to be evaluated using Eq. (4.4.19). For this analysis we have assumed the stream function to be zero at the lower wall, which leads to

$$(\text{constant})_L=0, \qquad (\text{constant})_U=Q^{(0)}-\Xi_U^{(0)}+\Xi_L^{(0)}. \qquad (4.4.20a,b)$$

The final form of the boundary conditions resulting from the constant volume flux constraint becomes

$$V_L^{(0)} = \sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} G_k^{(m)} \left(w_L\right)_k^{(-m)} = 0 \,,$$

$$V_U^{(0)} = \sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} G_k^{(m)} \left(w_U\right)_k^{(-m)} = Q^{(0)} - \Xi_U^{(0)} + \Xi_L^{(0)} \,. \qquad (4.4.21a,b)$$

In all calculations presented in this chapter, the volume flux through the corrugated slot has been taken as $Q^{(0)} = 4/3$, i.e., equal to the volume flux for Poiseuille flow in a smooth slot.

The projection equations (4.4.8) and the discretized boundary conditions (4.4.15) and (4.4.21) provide building blocks for the construction of different versions of the IBC method. Similarly as in Section 4.3, one can select a certain number of projection equations and a certain number of boundary constraints that need to be retained for each Fourier mode resulting either in a closed or an over-determined formulation. We have selected for numerical tests the same versions of the over-determined formulation of the IBC method that had already been tested in the context of the Laplace equation (see Section 4.3).

4.4.2. Numerical tests

This section provides discussion on the results of numerical tests carried out with various versions of the IBC method. For convenience, we selected the same test geometry as in Section 4.3 (see Eq. 4.3.27) and produced a machine-accurate reference solution using the Domain Transformation (DT) method based on the Fourier expansion in x-direction

and expansions based on the Chebyshev polynomials in y-direction. Details of the DT method applied to the biharmonic equation can be found in Appendix F.

We define the error in the evaluation of the u- and v-velocity components at any (x,y) location as

$$u_{er}(x,y) = u_{IBC}(x,y) - u_{DT}(x,y), \qquad v_{er}(x,y) = v_{IBC}(x,y) - v_{DT}(x,y) \quad (4.4.23a,b)$$

where the subscript 'IBC' stands for machine accurate the results obtained using any version of the IBC algorithm and the subscript 'DT' denotes the solution obtained using the DT method. We use the following norms to compare errors of different methods

$$\|u_{er}\|_{\infty} = \sup_{\substack{0\le x\le\lambda \\ y_L(x)\le y\le y_U(x)}} |u_{er}(x,y)|, \qquad \|v_{er}\|_{\infty} = \sup_{\substack{0\le x\le\lambda \\ y_L(x)\le y\le y_U(x)}} |v_{er}(x,y)|. \quad (4.4.24a,b)$$

The errors in the enforcement of the physical boundary conditions along the corrugated boundary $y_L(x)$ are defined as

$$u_{er,bc}(y_L(x)) = u_{IBC}(y_L(x)), \qquad v_{er,bc}(y_L(x)) = v_{IBC}(y_L(x)) \quad (4.4.25a)$$

and the norms defined as

$$\|u_{er,bc}\|_{\infty} = \sup_{0\le x\le\lambda} |u_{er,bc}(x)|, \qquad \|v_{er,bc}\|_{\infty} = \sup_{0\le x\le\lambda} |v_{er,bc}(x)| \quad (4.4.25b)$$

are used to compare accuracies in the enforcement of the boundary conditions using different versions of the method. The corresponding Fourier spectra of boundary errors are given by

$$u_{er,bc}(y_L(x)) = \sum_{n=-\infty}^{n=+\infty} U_{er,bc}^{(n)} e^{in\alpha x}, \quad v_{er,bc}(y_L(x)) = \sum_{n=-\infty}^{n=+\infty} V_{er,bc}^{(n)} e^{in\alpha x}. \quad (4.4.26a,b)$$

Since we have shown in Section 4.3 that (i) the pseudo-classical IBC method does not improve accuracy and (ii) the accuracy of the second version of the over-determined formulation is nearly the same as the accuracy of the first version, we shall omit both these methods/versions from any further discussion. We shall use SVD method for evaluation of the pseudo-inverse unless explicitly stated otherwise.

Figure 4.9 illustrates variations of the rank of the coefficient matrix for different values of the weight factor w_t as a function of the corrugation amplitude S and the corrugation wave number α. It can be seen that the "numerically determined" rank deteriorates as the severity of the geometry increases (larger S and α), but it can be corrected by increasing the weight factor.

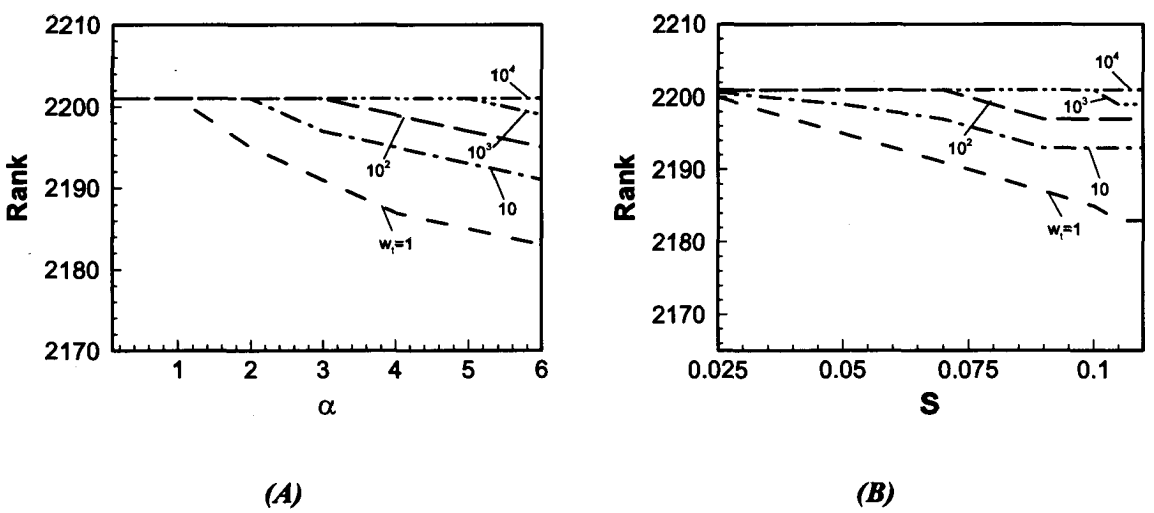


Figure 4.9. *Variations of the rank of the coefficient matrix as a function of the corrugation wave number α for the corrugation amplitude S=0.1 (Fig. 4.8A) and as a function of the corrugation amplitude S for the corrugation wave number α=5 (Fig. 4.8B) for the biharmonic equation (Section 4.4) with the slot geometry defined by Eq. (4.3.27) for the over-determined formulation (version 1) of the IBC method with N_M=15 Fourier modes, N_T=70 Chebyshev polynomials and 2*(M_M+1)=42 boundary constraints.*

Figure 4.10 illustrates variations of the error in the u-velocity component as well as variations of the rank of the coefficient matrix as a function of the weight factor for a

fairly severe geometry of the corrugation where one could expect numerical problems associated with the rank deficiency. It can be seen that the coefficient matrix is rank deficient when the weight factor w_t=1, i.e., when no weighting is used, and the corresponding error is quite large. An increase of the weight factor improves numerical conditioning of the matrix in the sense that the numerically-determined rank increases

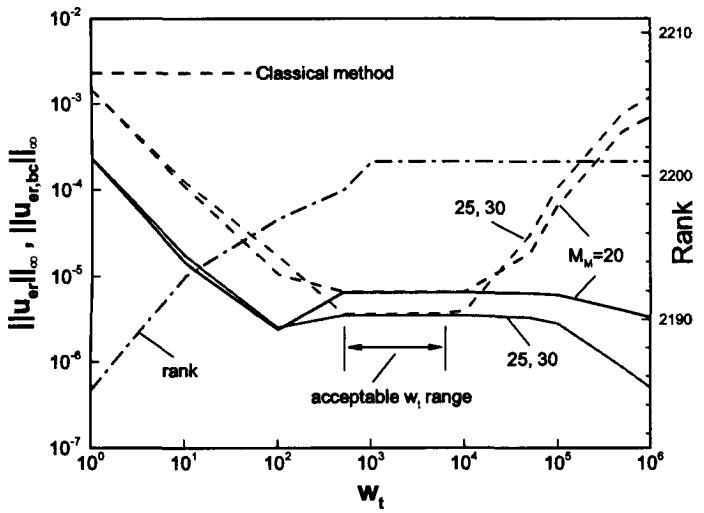


Figure 4.10. *Variations of the* $\|u_{er}(x,y)\|_\infty$ *(dash lines) and* $\|u_{er,bc}(x)\|_\infty$ *(solid lines) norms, and of the rank of the coefficient matrix (dash-dot line), as a function of the weight factor for the biharmonic equation (Section 4.4) for the slot geometry defined by Eq. (4.3.27) with the corrugation amplitude S=0.1 and the corrugation wave number α=5.0 for the over-determined formulation (version 1) of the IBC method. The solutions were determined with* N_M*=15 Fourier modes,* N_T*=70 Chebyshev polynomials and with a different number 2*(*M_M*+1) of boundary constraints. The reference solution has been determined using the DT method with* N_M*=30 Fourier modes and* N_T*=100 Chebyshev polynomials. Accuracy of the classical IBC method used with the same number of Fourier modes and Chebyshev polynomials is shown for comparison purposes.*

and, at the same time, the accuracy of the computed solution improves until the numerically determined rank becomes full. The rank and the accuracy remain constant with further increase of the weight factor until this factor becomes too large. Further weighting of boundary constraints increases the maximum error in the interior of the

solution domain although the error in the enforcement of boundary conditions may still decrease. It is obvious that the weighting that one should use in order to maximize accuracy cannot be too small, but weighting that is too big will also compromise accuracy. Since the correct weighting is problem dependent, results shown in Fig. 4.10 suggest that the most reliable way to choose the correct weighting, which we shall refer to as the optimal weight, is to select the minimum weighting that leads to a numerically full-rank coefficient matrix. Comparison of Figs 4.7 and 4.10 show that, when the correct weight factor is used (optimal weight in the case of the biharmonic equation), the maximum error at the boundary is of the same order of magnitude as the order over the whole domain.

Variations of the error in the u-velocity as a function of the corrugation amplitude S for the corrugation wave number α=5 are illustrated in Fig. 4.11. It can be seen that there exists a certain critical combination of the corrugation wave number and the corrugation amplitude (critical zone) beyond which the accuracy of the classical IBC method cannot be improved by increasing the number of Fourier modes N_M and Chebyshev polynomials N_T used in the computations. Both versions of the over-determined formulation provide similar and substantially improved accuracy (as compared with the classical method) around the critical zone with a proper selection of the weight factor, similarly as in the case of the Laplace equation. The same figure also demonstrates that an increase in the number of boundary constraints $2*(M_M+1)$ used in the computations cannot arbitrarily improve the accuracy, i.e., there exist a certain M_M that provides the best possible accuracy and this accuracy cannot be improved with further increase of M_M. Similar behavior has also been observed in the case of the Laplace equation discussed in Section 4.3.

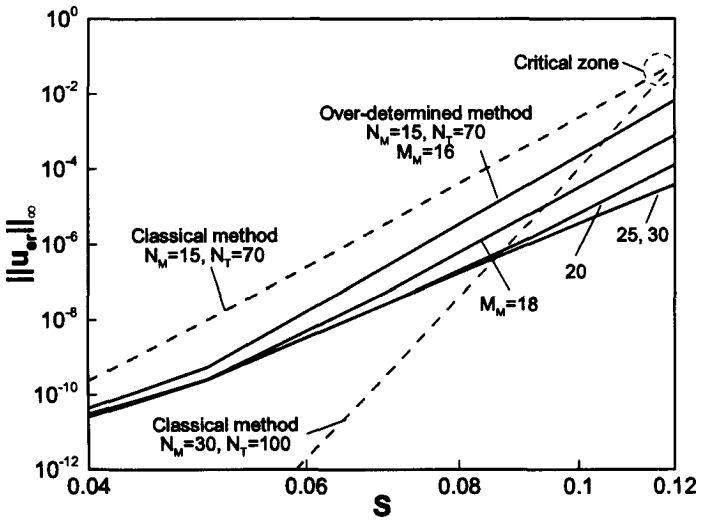


Figure 4.11. *Variations of the $\|u_{er}(x,y)\|_{\infty}$ norm as a function of the corrugation amplitude S for the biharmonic equation (Section 4.4) with the slot geometry defined by Eq. (4.3.27) for the corrugation wave number α=5.0 for the over-determined formulation (version 1) of the IBC method for different number of boundary constraints 2*(M_M+1) used in the computations. The boundary constraints for the over-determined methods were weighted with a factor $w_i=10^3$. The reference solution has been determined using the domain transformation (DT) method with N_M=30 Fourier modes and N_T=100 Chebyshev polynomials. Results obtained using the classical IBC method are shown for comparison purposes.*

Figure 4.12 illustrates the distribution of the boundary error along the corrugated wall. In the case of the classical IBC method the error oscillates along the length of the slot with the location of the maximum amplitude of the oscillations coinciding with the maximum slot opening. The difference between the maximum and minimum amplitudes is almost three orders of magnitude. The same amplitude is approximately constant along the length of the slot in the case of the over-determined formulation. The Fourier spectra of the boundary errors in the u-velocity displayed in Fig. 4.13 show that the first N_M+1 Fourier modes in the error distribution have zero amplitudes in the case of the classical IBC method, in agreement with the theoretical predictions. A similar spectrum for the error of the over-determined formulation demonstrates that the error is spread over all Fourier modes while, at the same time, the magnitude of the Fourier coefficients is reduced by almost two orders of magnitude as compared with the classical IBC method.

This observation is consistent with the accuracy improvement associated with the over-determined formulation (see Fig. 4.11) and is qualitatively similar with the observations made in the case of the Laplace equation (Section 4.3).

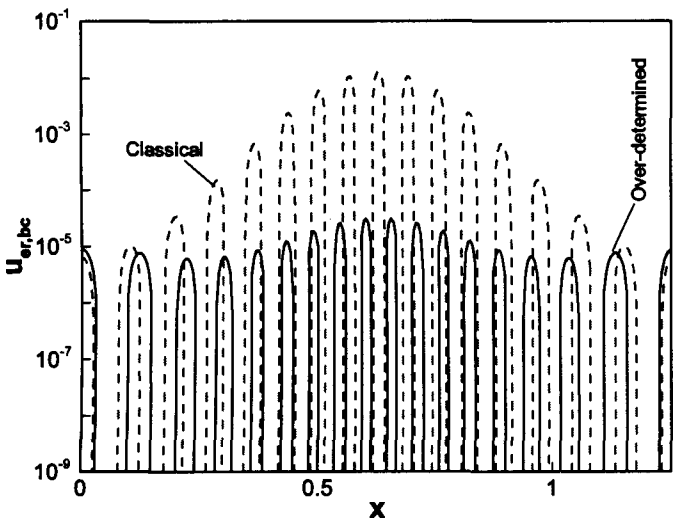


Figure 4.12. *Distribution of the boundary error $u_{er,bc}(y_L(x))$ for the biharmonic equation (Section 4.4) for the slot geometry defined by Eq. (4.3.27) for the corrugation wave number α=5.0 and the corrugation amplitude S=0.11 determined using the classical and over-determined (version 1) formulations of the IBC method and evaluated using N_M=15 Fourier modes, N_T=70 Chebyshev polynomials and, in the case of the over-determined method, $2*(M_M+1)$=42 boundary constraints. All other conditions as in Fig. 4.10.*

Analysis of results shown in Figs 4.11-4.13 demonstrates that the biggest gains in accuracy are achieved by adding just one additional boundary constraint (equivalent to M_M=16 for this test), as this leads to a more uniform error distribution and the associated reduction in its magnitude, e.g., Figs 4.12-4.14. Further increase of M_M is less efficient in error reduction, and increase of M_M beyond a certain critical values ($M_M \approx 30$ in the present case) brings no benefit at all.

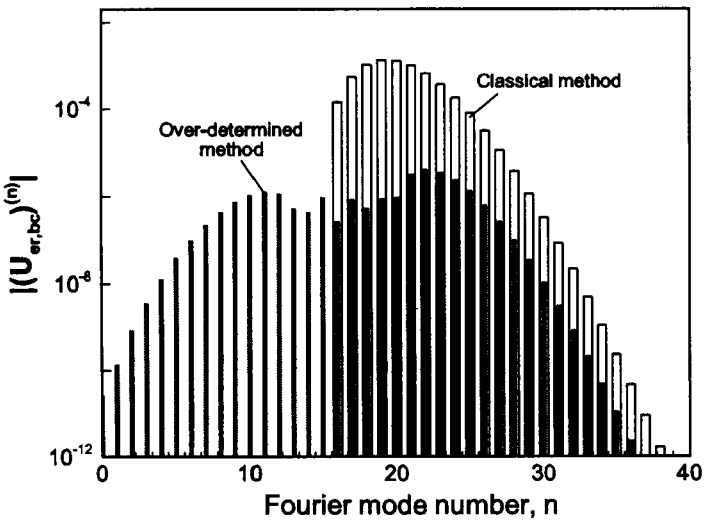


Figure 4.13. *Fourier spectra of the boundary error $u_{er,bc}(y_L(x))$ for the biharmonic equation (Section 4.4) with the slot geometry defined by Eq. (4.3.27) with the corrugation amplitude $S=0.11$ and the corrugation wave number $\alpha=5.0$ determined using the classical and over-determined formulations (version 1) of the IBC method and evaluated using $N_M=15$ Fourier modes, $N_T=70$ Chebyshev polynomials and, in the case of the over-determined formulation, $2*(M_M+1)=42$ boundary constraints. All other conditions as in Fig. 4.10.*

4.5. Flows described by the Navier-Stokes equations

The Navier-Stokes equations are of interest in many applications as they describe motions of many of the most common fluids. The general case of a three-dimensional flow can be reduced to a system of Laplace-like and biharmonic-like operators with nonlinear terms treated as known through various forms of linearization. Such operators have been discussed separately in Sections 4.3 and 4.4 and thus it remains to consider modification of the algorithm to account for the presence of nonlinear terms. Here we select the simplest case of a two-dimensional flow in a corrugated slot whose geometry is described by Eq.(4.2.1) as such problem contains all features of the more general three-dimensional problem yet remains sufficiently simple. The main difference between the

present problem and the problem discussed in Section 4.4 is the presence of nonlinear terms arising due to the convective effects.

4.5.1. Problem formulation

The reference flow, i.e., flow through a smooth slot, is the Poiseuille flow with the velocity and pressure fields in the form

$$u_0(x,y) \equiv u_0(y) = 1 - y^2, \quad v_0(x,y) = 0, \quad p_0(x,y) \equiv p_0(x) = -2x/Re \tag{4.5.1}$$

where the motion of the fluid is in the positive x-direction, the Reynolds number Re is based on the half channel height and the maximum velocity in the x-direction and the flow is driven by a constant negative pressure gradient. Introduction of the wall corrugation introduces flow modifications and thus the total flow quantities can be expressed as

$$u(x,y) = u_0(y) + u_1(x,y), \quad v(x,y) = v_1(x,y), \quad p(x,y) = p_0(x) + p_1(x,y) \tag{4.5.2}$$

where u, v and p denote the total velocities and pressure, and u_1, v_1 and p_1 denote velocity and pressure modifications due to the presence of boundary irregularities. Substitution of (4.5.2) into the Navier-Stokes and continuity equations result in the following form of governing equations

$$u_0\partial_x u_1 + u_1\partial_x u_1 + v_1 Du_0 + v_1\partial_y u_1 = -\partial_x p_1 + Re^{-1}\nabla^2 u_1, \tag{4.5.3a}$$

$$u_0\partial_x v_1 + u_1\partial_x v_1 + v_1\partial_y v_1 = -\partial_y p_1 + Re^{-1}\nabla^2 v_1, \tag{4.5.3b}$$

$$\partial_x u_1 + \partial_y v_1 = 0. \tag{4.5.3c}$$

The reader may note that the flow problem can be posed either for the complete flow quantities (u,v,p) or for the flow modifications (u_1,v_1,p_1). Both approaches have been considered and the latter one was selected due to the numerically smaller values of the resulting nonlinear terms and faster convergence of the iterative solution process.

Introduction of the stream function Ψ as defined by Eq. (4.4.2) and elimination of pressure lead to a single field equation in terms of the unknown modification of stream function Ψ_1 in the form

$$\begin{aligned} Re^{-1}\nabla^4\Psi_1 - [u_0\partial_x(\nabla^2) - D^2u_0\partial_x]\Psi_1 &= \partial_y[\partial_x(u_1u_1) + \partial_x(u_1v_1)] \\ &\quad -\partial_x[\partial_x(u_1v_1) + \partial_y(v_1v_1)]. \end{aligned} \tag{4.5.4}$$

The solution algorithm treats the nonlinear terms on the right hand side of (4.5.4) as known. These terms are improved iteratively until convergence is reached. The required boundary conditions are identical to those given by Eq. (4.4.5). Use of discretization technique described in Section 4.4, i.e., Fourier expansion in the x-direction and Chebyshev expansion in y-direction, and application of the Galerkin projection technique, result in a set of discretized algebraic equations for each Fourier mode in the form

$$\begin{aligned} &\sum_{k=0}^{n=N_T} [\Gamma^4 Re^{-1}\langle T_j, D^4T_k\rangle - 2n^2\alpha^2\Gamma^2 Re^{-1}\langle T_j, D^2T_k\rangle + n^4\alpha^4 Re^{-1}\langle T_j, T_k\rangle - \\ &\quad in\alpha\Gamma^2\langle T_j, u_0D^2T_k\rangle + in^3\alpha^3\langle T_j, u_0T_k\rangle]G_k^{(n)} \\ &= \sum_{k=0}^{k=N_T}\Big[\Big(in\alpha\Gamma\langle T_j, DT_k\rangle + n^4\alpha^4 Re^{-1}\langle T_j, T_k\rangle\Big)\hat{R}_{1,k}^{(n)} + \Big(\Gamma^2\langle T_j, D^2T_k\rangle\Big)\hat{R}_{2,k}^{(n)} - \Big(in\alpha\Gamma\langle T_j, DT_k\rangle\Big)\hat{R}_{3,k}^{(n)}\Big] \end{aligned} \tag{4.5.5}$$

where j=0,…,N_T, n=0,….,N_M. The products on the right hand side of (4.5.4) have been replaced by their Fourier expansions in the form

$$[(u_1u_1),(u_1v_1),(v_1v_1)](x,\hat{y}) = \sum_{n=-N_M}^{n=+N_M}[R_1^{(n)}, R_2^{(n)}, R_3^{(n)}](\hat{y})e^{in\alpha x} \tag{4.5.6a}$$

and the corresponding modal functions have been replaced by Chebyshev expansions in the form

$$\left[R_1^{(n)}, R_2^{(n)}, R_3^{(n)}\right](\hat{y}) = \sum_{k=0}^{k=N_T} \left[\hat{R}_{1,k}^{(n)}, \hat{R}_{2,k}^{(n)}, \hat{R}_{3,k}^{(n)}\right] T_k(\hat{y}). \tag{4.5.6b}$$

The discretized boundary conditions are the same as those for the biharmonic equation, i.e., they are given by Eqs. (4.4.15) and (4.4.21). The projection equations (4.5.5) and the discretized boundary constraints (4.4.15) and (4.4.21) provide building blocks for the construction of different versions of the IBC method. Similarly as in Section 4.4, one can select a certain number of projection equations that need to be retained for each Fourier mode and a certain number of boundary constraints resulting either in a closed or an over-determined formulation.

The solution process involves iterations due to nonlinearity of the problem. The left hand side coefficient matrix originating from the linear terms needs to be computed only once followed by the evaluation of the inverse for the classical method and the pseudo-inverse for the over-determined methods. At each iteration, the solution is obtained by multiplying the known nonlinear terms on the right hand side by the inverse or pseudo-inverse of the coefficient matrix followed by re-evaluation of the nonlinear terms. The computational cost of each iteration is dominated by the cost of evaluation of the non-linear terms. These terms are computed using Fast Fourier Transformation (FFT) and aliasing errors are controlled through the use of the padding method [Ref. 15]. The iterations continue until the change in the magnitude of the Chebyshev coefficients is less than the convergence criteria which, for all results presented in this chapter, were set to 10^{-14}.

4.5.2. Numerical tests

We shall now discuss results of numerical tests carried out with the various versions of the IBC method. For convenience, we have selected the same test geometry as already used in Sections 4.3 and 4.4 (see Eq. 4.3.27) and produced a machine-accurate reference solution using the Domain Transformation (DT) method based on the Fourier expansion in x-direction and expansions based on the Chebyshev polynomials in the y-direction. Details of the DT method applied to the Navier-Stokes equations can be found in Appendix G. For convenience, we shall use the same definition of errors as in the case of the biharmonic equation given by Eqs (4.4.23)-(4.4.26). All results presented in this section have been obtained using the first version of the over-determined IBC method. The second version has been tested and, since it produces nearly the same accuracy, it has been omitted from any discussion.

The reader may note that the dominant term on the left hand side of (4.5.4) is multiplied by Re^{-1} which could result in a boundary-layer like behavior of the solution for large Re. Such potential boundary layers, however, do not pose numerical problems as they can be easily resolved due to the use of Chebyshev expansions in the y-direction. One could multiply Eq.(4.5.4) by Re before starting the discretization process. This would result in a dominant biharmonic operator on the left hand side and nonlinear terms on the right hand side increased by Re. Solution based on this form of equations has been tested and it resulted in problems with the rank of the coefficient matrix similar to those described in Section 4.4 regardless of the value of Re, as well as in the increase of problems associated with the convergence of the iterative solution process. The form of the field equation given by Eq.(4.5.4) was judged to be the best for the numerical work and thus all tests described in the remainder of this section have been carried using this form.

Figure 4.14 illustrates variations of the numerically determined rank of the coefficient matrix as a function of the corrugation wave number α and the corrugation amplitude S for various levels of the weight factor w_t for two values of the Reynolds number. It can be seen that the numerically determined rank of the matrix deteriorates as the severity of

the geometry increases. This deterioration is significant for low values of Re (e.g., Re=0.1) but very small for higher values (e.g., Re=100) where the relative "weight" of the biharmonic operator in the projection equations is automatically "corrected" by the factor Re^{-1}. While in both cases, i.e., for small and large Re, the problem can be corrected by using appropriate weighting, a large weighting of $w_t=10^5$ is required when Re=0.1 but small weighting of $w_t=10$ is sufficient when Re=100. This auto-correction of the rank of the coefficient matrix for larger values of Re simplifies code calibration and selection of other numerical parameters.

Figure 4.15 illustrates variations of the accuracy of the method as well as the numerical rank of the coefficient matrix as a function of the weight factor w_t for different values of the Reynolds number for a fairly severe geometry of the corrugation where one could expect numerical problems associated with the rank deficiency. We shall begin our discussion with the low values of Re where the coefficient matrix is rank deficient when no weight-factor is used, i.e., when $w_t=1$. As the weight factor increases, the numerically-determined rank approaches the full rank and, at the same time, the accuracy of the results increases, similarly as in the case of the biharmonic equation discussed in Section 4.4. The reader should note that an excessive increase of w_t deteriorates the overall accuracy but may improve accuracy in the enforcement of the boundary conditions. One may conclude that there exists an optimal range of w_t where the results are most accurate, e.g. Fig. 4.15A. The best way to identify the correct value of w_t, which we shall refer to as an optimal weight factor, is to select the lowest value of w_t that results in the numerically full rank of the coefficient matrix. Results displayed in Fig. 4.15A show that the errors in boundary conditions and in the field are of the same order of magnitude when the optimal w_t is used, and they are smaller by about two orders of magnitude when compared with the error of the classical method.

The performance of the algorithm remains qualitatively the same as the Reynolds number increases, as illustrated in Figs 4.15B and 4.15C for Re=1 and Re=10, respectively. However, problems with the matrix rank are less pronounced and can be corrected with progressively smaller values of w_t. The error can be reduced by about two orders of

magnitude by using the over-determined formulation with the optimal weighting for the conditions used in these tests.

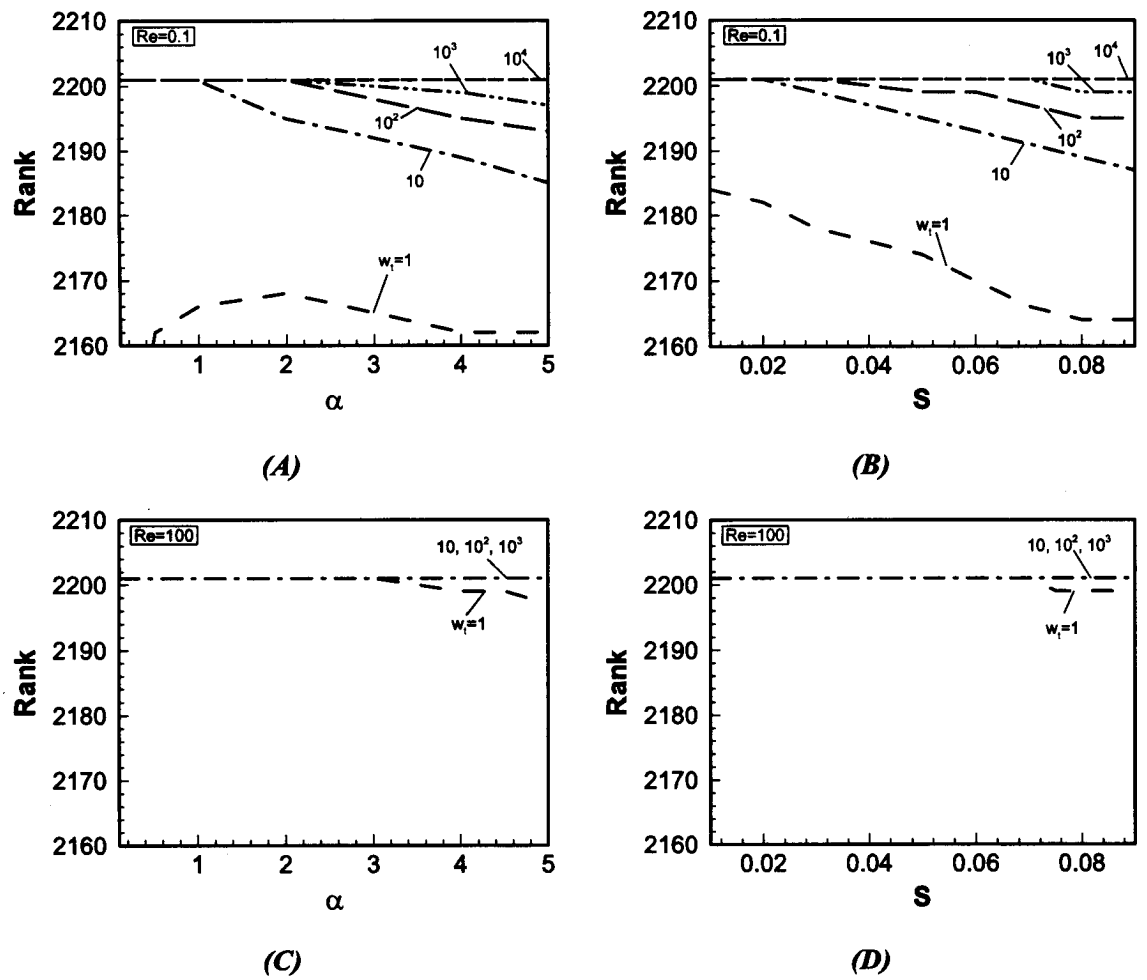


Figure 4.14. *Variations of the rank of the coefficient matrix as a function of the corrugation wave number α for the corrugation amplitude S=0.1 (Fig. 4.8A,C) and as a function of the corrugation amplitude S for the corrugation wave number α=5 (Fig. 4.8B,D) for the Navier-Stokes problem (Section 4.5) for different Reynolds numbers with the slot geometry defined by Eq. (4.3.27) for the over-determined formulation (version 1) of the IBC method. Computations have been carried out with N_M=15 Fourier modes, N_T=70 Chebyshev polynomials and 2*(M_M+1)=42 boundary constraints.*

Further increase of the Reynolds number changes characteristics of the algorithm. Problem with matrix rank largely disappears for Re=100, as illustrated in Fig. 4.15D. Error of the method is reduced by increasing the number of additional boundary constraints but an excessive increase of M_M leads to a large increase of the error, which is opposite to what has been found in the case of the Laplace and biharmonic operators as well as the Navier-Stokes equations for low Re. In the case of the test illustrated in Fig.15D, the smallest error is produced for M_M=20 and a further increase in the number of boundary constraints (i.e. M_M=25-30) increases the error by three orders of magnitude making the over-determined method worse than the classical method. As a result, selection of the "best" value of M_M becomes critically important; this value can always be identified on the basis of accuracy of enforcement of physical boundary conditions on the corrugated boundary as the boundary values are given exactly in the problem formulation. The same conclusions apply to the selection of the optimal weight factor.

Figure 4.16 illustrates variations of the accuracy of the method as a function of the corrugation amplitude S for a fixed corrugation wave number α=5. The improvements in the accuracy obtained using the over-determined formulation, when compared with the classical method, are clearly visible. The classical method fails to provide a reasonable accuracy beyond certain critical combination of the corrugation wave number α and the corrugation amplitude S (critical zone), and the accuracy cannot be improved by increasing the number of Fourier modes N_M and the number of Chebyshev polynomials N_T, similar to the cases of the model problems discussed in Sections 4.3 and 4.4. The over-determined formulation provides accuracy gains of about two orders of magnitude for geometries corresponding to the "critical zone" but only if Re is sufficiently small (e.g., Re≤10, see Fig. 4.16) and, in this sense it extends the range of applicability of the IBC method as far as severity of geometries is concerned. When Re is high enough (Re=100 in our test, see Fig. 4.16D) the accuracy gains are of about one order of magnitude for less severe geometries, but disappear for geometries where the classical method starts to experience accuracy problems (i.e., geometries corresponding to the critical zone).

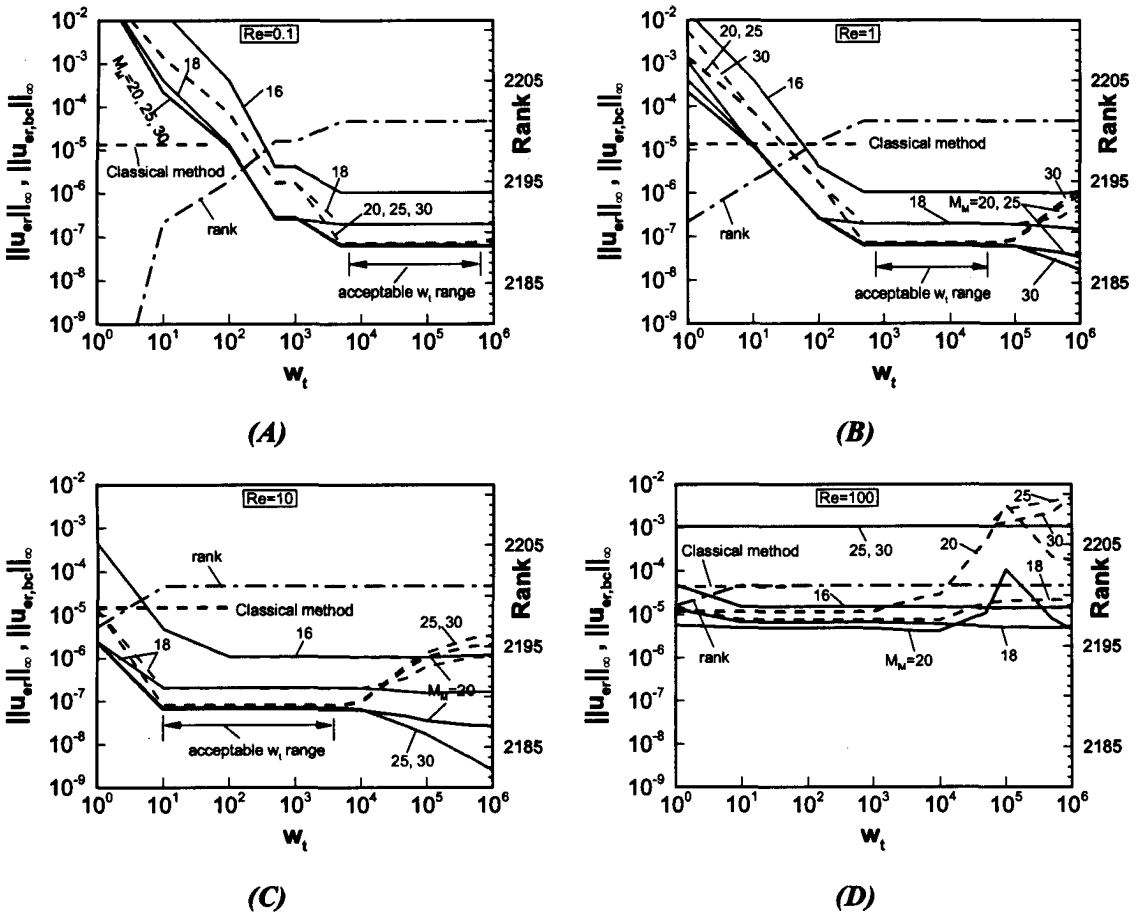


Figure 4.15. *Variations of the* $\|u_{er}(x,y)\|_\infty$ *(dash lines) and* $\|u_{er,bc}(x)\|_\infty$ *(solid lines) norms, and of the rank of the coefficient matrix (dash-dot line) as a function of the weight factor* w_t *for the Navier-Stokes equations (Section 4.5) with the slot geometry defined by Eq. (4.3.27) for the corrugation wave number* α*=5.0 and the corrugations amplitude S=0.075 for the over-determined formulation (version 1) of the IBC method and evaluated using* N_M*=15 Fourier modes and* N_T*=70 Chebyshev polynomials, and different number* $2*(M_M+1)$ *of boundary constraints. The reference solutions have been determined using the domain transformation (DT) method with* N_M*=35 Fourier modes and* N_T*=120 Chebyshev polynomials. Accuracy of the classical IBC method used with the same number of Fourier modes and Chebyshev polynomials is shown for comparison purposes.*

It can be concluded that an increase of the Reynolds number decreases accuracy gains associated with the over-determined formulation but these gains remain nevertheless of practical interest for less severe geometries (see Fig. 4.16). The use of the over-

determined formulation extend the range of applicability (in terms of the severity of geometry) of the IBC method, but this range decreases with an increase of Re and becomes virtually identical with the range of the classical method when Re≈100.

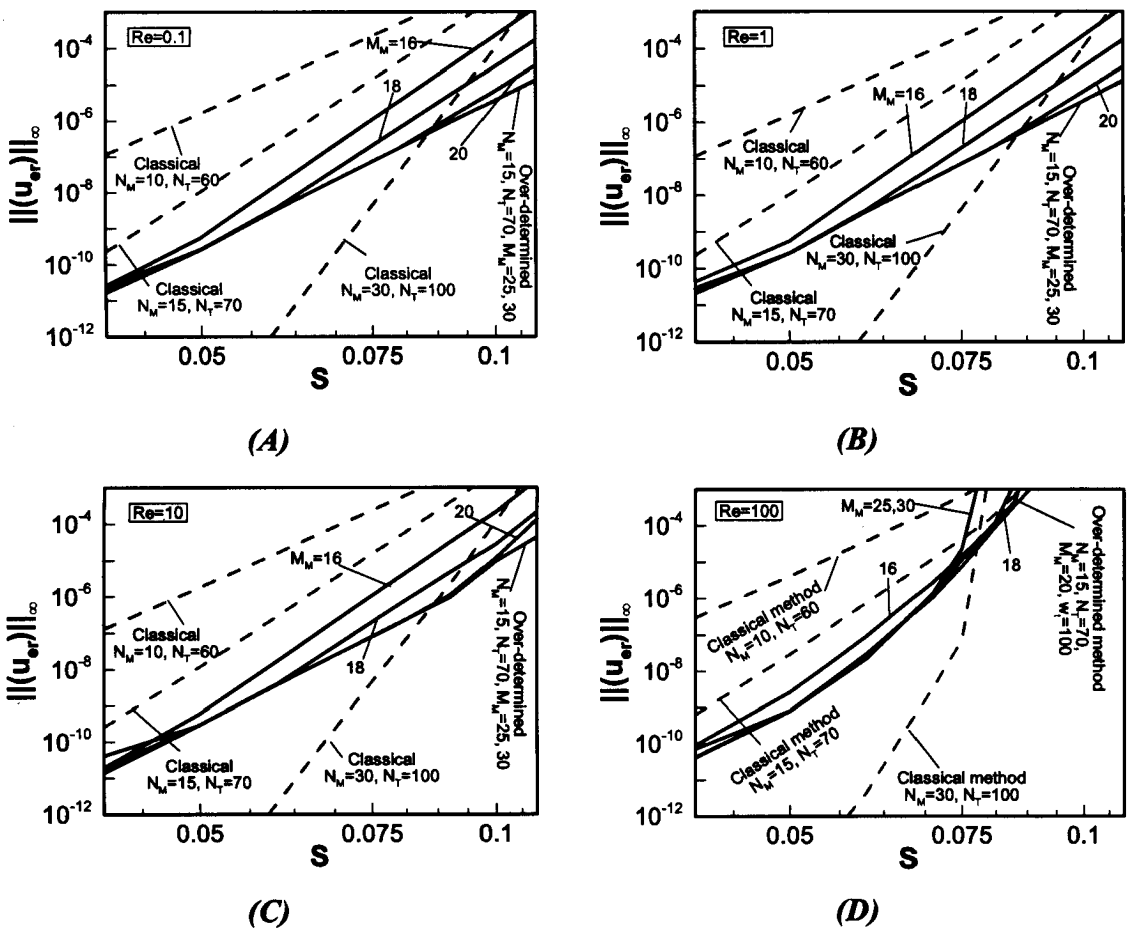


Figure 4.16. *Variations of the* $\|u_{er}(x,y)\|_\infty$ *norm as a function of the corrugation amplitude S for the Navier-Stokes equations (Section 4.5) with the slot geometry defined by Eq. (4.3.27) for the corrugation wave number α=5.0 for the over-determined formulation (version 1) of the IBC method. The boundary conditions for the over-determined method were weighted with a factor* $w_i=10^4$ *for Re=0.1 and Re=1,* $w_i=10^3$ *for Re=10 and* $w_i=10^2$ *for Re=100. Results obtained using the classical IBC method are shown for comparison purposes (dash lines). All other conditions are same as in Fig. 4.14.*

4.6. Summary

We have considered over-determined formulation of the immersed boundary conditions (IBC) method. The field equations are discretized on a regular computational domain using Fourier expansions in the periodic direction and Chebyshev expansions in the transverse direction, and are converted into algebraic equations using Galerkin projection method. The physical domain has an irregular shape and is submerged inside the regular computational domain. The physical boundary conditions enter the numerical algorithm in a Tau-like manner and their discretization leads to relations referred to as boundary constraints. The discretization process results in a number of boundary constraints that is larger than that required to form a closed system of algebraic equations. Use of additional boundary constraints as well as all available projection equations lead to an over-determined formulation and the resulting algebraic equations can be solved using the least squares method. Use of additional boundary constraints is explored in order (i) to increase the accuracy of the IBC method and (ii) to increase the range of applicability of the IBC method to more severe geometries.

Numerical tests have been carried out for the Laplace and biharmonic operators, and for the Navier-Stokes equations. It has been found that the accuracy of the results obtained using the NE, QR and SVD based evaluation of the pseudo-inverse of the coefficient matrix of the over-determined algebraic system were comparable, especially around the critical zone. However, SVD was chosen to evaluate the pseudo inverses for the model problems in Sections 4 and 5 as the coefficient matrices encountered for these problems are prone to numerical rank-deficiency. It has been determined that the retention of all projection equations does not offer any accuracy advantage. On the other hand, increase in the number of boundary constraints increases the accuracy of the method as the boundary error is spread over all Fourier modes involved in the computations resulting in a lower absolute value of the error. The accuracy increases by at least two orders of magnitude and the range of applicability of the method expands as compared with the classical formulation of the IBC method for the Laplace and biharmonic operators. There is a certain maximum number of boundary constraints to be included in the computations

that results in the best accuracy (30%-40% higher than the number of Fourier modes used to represent the solution) and further increase in the number of these constraints does not offer any advantage. The biharmonic operator suffers from the numerical rank deficiency but this problem can be corrected by re-weighting boundary constraints (i.e., by selecting optimal weighting factor). In the case of the Navier-Stokes equations the performance of the method depends on the Reynolds number Re. For low Re, the algorithm performs as in the biharmonic operator case. As Re increases, the problems with the numerical rank deficiency decreases, but the accuracy gains associated with the over-determined formulation and the increase in the range of applicability diminishes. The accuracy depends on the number of additional boundary constraints and its excessive increase leads to the deterioration of the accuracy as compared with the classical formulation of the method. The over-determined formulation provides nevertheless accuracy gain of at least one order of magnitude within its range of applicability when compared with the classical formulation for the highest value of Reynolds number tested, i.e., Re=100.

4.7. References

[1] Peskin, C.S., The fluid dynamics of heart valves: experimental, theoretical and computational methods, *Annu. Rev. Fluid Mech.*, vol. 14 pp. 235-59 (1982).

[2] Mittal, R. and Iaccarino, G., Immersed boundary methods, *Annu. Rev. Fluid Mech.*, vol. 37 pp. 239-261 (2005).

[3] Peskin, C.S., The immersed boundary method, *Acta Numerica*, 479-517 (2002).

[4] Girault, V., Glowinski, R. and Pan, T.W., A Fictitious-Domain Method with Distributed Multiplier for the Stokes Problem, *App. Nonlinear. Ana.*, Springer, pp. 159-174 (2000).

[5] Parussini, L., Fictitious Domain Approach Via Lagrange Multipliers with Least Squares Spectral Element Method, *J. Sci. Comput.*, vol. 37, pp. 316-315 (2008).

[6] Floryan, J.M. and Rasmussen, H., Numerical analysis of viscous flows with free surfaces, *Appl. Mech. Rev.*, vol. 42, pp. 323-341 (1989).

[7] Osher, S.J. and Fedkiw, R.P., Level Set Methods and Dynamic Implicit Surfaces, *Springer-Verlag* (2002).

[8] Sethian, J.A., Level set methods and fast marching methods : evolving interfaces in computational geometry, Fluid Mechanics, Computer Vision, and Materials Science, *Cambridge University Press* (1999).

[9] Szumbarski, J. and Floryan, J.M., A direct spectral method for determination of flows over corrugated boundaries, *J. Comp. Phys.*, vol. 153, pp. 378-402 (1999).

[10] Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for unsteady flow problems described by the Laplace operator, *Int. J. Num. Meth. Fluids*, vol. 56, pp. 1765-1786 (2007).

[11] Husain, S.Z. and Floryan, J.M., Implicit spectrally-accurate method for moving boundary problems using immersed boundary conditions concept, *J.Comp.Phys.*, vol. 227, pp. 4459-4477 (2008).

[12] Husain, S.Z. and Floryan, J.M., Gridless spectral algorithm for stokes flow with moving boundaries, *Comput. Methods Appl. Mech. Engrg*, doi:10.1016/j.cma.2008.07.007 (2008).

[13] Trefethen, L. N., and Bau, D., Numerical Linear Algebra, *SIAM* (1997).

[14] Björck, Å., Numerical Methods for Least Squares Problems, *SIAM* (1996).

[15] Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T.A., Spectral Methods in Fluid Dynamics, *Springer* (1987).

CHAPTER 5

Efficient linear solvers[1]

5.1. Introduction

Flow problems involving irregular physical domains pose considerable computational challenges associated with modeling of geometries. Typical approaches involve generation of boundary conforming grids and imposition of boundary conditions along the edges of flow domains. A conceptually different method, where one carries out computations in an extended fixed and regular computational domain that engulfs the physical domain regardless of its shape, is referred to as the "immersed boundary" (IB) method and has been proposed by Peskin [1] in the context of cardiac mechanics problems. The attractiveness of this concept arises from the elimination of the cost of generation of boundary conforming grids, while the main challenge is associated with the enforcement of the physical flow boundary conditions which are now located in the interior of the solution domain and enter formulation of the numerical problem in the form of constraints. In general, the immersed boundary methods introduce additional forcing in order to make the fluid to move along the physical boundaries. The forcing applied can be either continuous or discrete, and detailed procedures for its construction have been reviewed in [2, 3].

Most of the available implementations of the immersed boundary methods suffer from a limitation of low spatial accuracy attributable to their use of the low-order finite-

---

[1]A version of this chapter has been submitted for publication –

Husain, S.Z. and Floryan, J.M., Efficient linear solvers for problems arising from the spectral implementation of the immersed boundary conditions method (2009).

difference, finite-volume or finite-element technique for spatial discretization. Another uncertainty is associated with the use of the local forcing in order to enforce the no-slip and no-penetration conditions. Such forcing may affect flow physics around the boundaries of the flow domain leading to a potentially inaccurate representation of local wall shear as well as other derivatives of the flow quantities. Although estimating the magnitude of inaccuracies associated with such procedures is difficult, any procedure that even locally alters the flow physics may be unable to predict the second derivative of mean flow with sufficient accuracy, and this could lead to an inaccurate prediction of stability characteristics of shear layers. Implementation of higher-order spatial discretization schemes may amplify this problem.

An alternative version of the IB method, the "immersed boundary conditions" (IBC) method, which uses spectrally-accurate spatial discretization and formal construction of boundary constraints, has been proposed in [4]. The geometries of flow domains are described using Fourier expansions, which limits the type of geometries that can be considers but, on the other hand, provides ability to deliver a machine level accuracy. This method results in a gridless algorithm where all possible variations of the boundary geometries are described in terms of the Fourier coefficients and thus offers an opportunity to develop geometry optimization procedures. The IBC method has been successfully employed in the analyses of different shear layer instability problems [5-7] and has been extended to unsteady problems [8] as well as to moving boundary problems [9, 10].

The implementation of the IBC method results in a number of boundary constraints that is far in excess of that required to formulate a closed system of algebraic equations. In the "classical" formulation [8-10], only the number of boundary constraints required to form a closed system is retained and boundary constraints corresponding to the lowest (dominant) Fourier modes are used for this purpose. Various tests have shown that if either the wave number of the physical boundary corrugation or the amplitude of this corrugation is too large, the classical method fails to provide an acceptable accuracy [8-10]. Use of additional constraints leads to an over-determined formulation, and this

formulation has been found to improve the accuracy and the range of applicability of the IBC algorithm [14].

The IBC method for both the classical and over-determined formulations uses Fourier expansions in the periodic directions and Chebyshev expansions in the normal-to-the-wall direction. Although such discretizations yield spectral convergence properties, the computational cost of the method increases very rapidly with increasing severity of the boundary geometries [8-10] and/or due to three-dimensional character of the flow problem and may lead to computational requirements (processing power and memory) far in excess of what can be provided by the currently available hardware. The computational speed and memory issues become more prominent for the over-determined formulation which is solved in the least squares sense.

The work presented in this chapter explores the special structure of the coefficient matrix resulting from the implementation of the IBC method and explores different solution techniques in the search for an efficient solver of the linear system. Section 5.2 discusses the model problem. Section 5.3 describes the different categories of solvers for the classical implementation of the IBC algorithm. Section 5.4 considers efficient solutions of linear systems resulting from the over-determined formulations of the IBC method. Section 5.5 discusses efficient solutions of the unsteady problems involving both constant and time-dependent boundary conditions. Section 5.6 provides a short summary of the main conclusions. A machine-accurate reference solution determined using the domain transformation (DT) is described in Appendix F. This solution is used for testing of the accuracy of the IBC algorithm when required.

5.2. Model problem

Our interest is in the development of algorithms for solution of the Navier-Stokes equations. Since the nonlinear terms are irrelevant to the objectives of the present work, we shall omit them and thus we shall focus on the Stokes limit where the field equations

can be expressed in terms of the bi-harmonic operator. This operator describes a number of problems in the theory of elasticity which expands the range of applicability of the present work.

We shall state the model problem in the context of flow in a two-dimensional corrugated channel extending to $\pm\infty$ in the x-direction and periodic with the wavelength $\lambda=2\pi/\alpha$ (see Fig. 5.1). The following presentation is limited to a short outline which is required in order to understand the origins and structure of the linear systems to be solved. Details of the formulation can be found in [4-10].

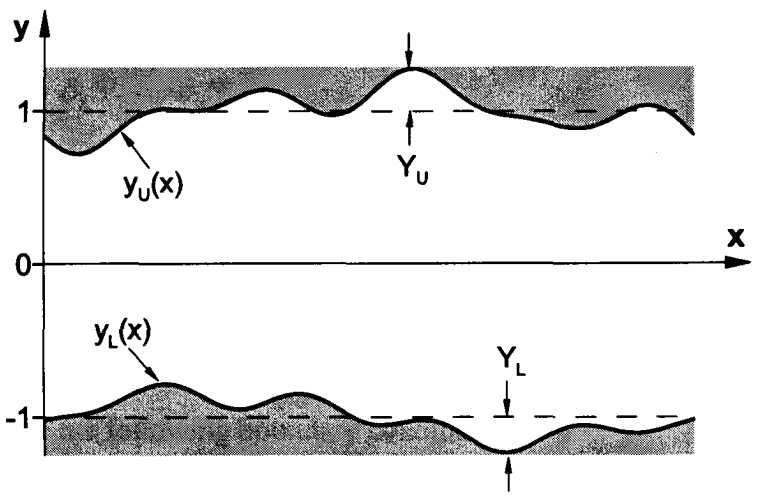


Figure 5.1. *Sketch of the solution domain.*

The channel of interest is bounded by walls whose geometry is expressed in terms of Fourier expansions in the form

$$y_L(x)=-1+\sum_{n=-N_A}^{n=+N_A} H_L^{(n)} e^{in\alpha x}\,, \qquad y_U(x)=1+\sum_{n=-N_A}^{n=+N_A} H_U^{(n)} e^{in\alpha x}\,, \qquad (5.2.1a,b)$$

where $H_L^{(n)} = \left(H_L^{(-n)}\right)^*$, $H_U^{(n)} = \left(H_U^{(-n)}\right)^*$ and stars denote complex conjugate. The dimensionless field equation describing flow in the interior of the channel has the form

$$\nabla^4 \Psi = 0 \tag{5.2.2}$$

where $\nabla^4 = [\partial_{xxxx} + 2\partial_{xxyy} + \partial_{yyyy}]$, ∂ denotes partial differentiation and subscripts x and y denote the arguments of the differentiations. Here, Ψ denotes the stream function, which is defined as

$$u(x,y) = u_0(y) + u_1(x,y) = D\Psi_0 + \partial_y \Psi_1 = \partial_y \Psi, \tag{5.2.3a}$$

$$v(x,y) = v_1(x,y) = -\partial_x \Psi_1 = -\partial_x \Psi \tag{5.2.3b}$$

where D=d/dy, $u_0(y)=1-y^2$, $\Psi_0=-y^3/3+y+2/3$ and u(x,y) and v(x,y) denote the total velocities in the x- and y-directions. The subscripts 0 and 1 denote quantities corresponding to the reference Poiseuille flow and modifications due to the presence of boundary corrugation, respectively. The assumptions of no penetration and no slip at the boundaries result in the following boundary conditions

$$\partial_y \Psi_1(x, y_L(x)) = -D\Psi_0(y_L(x)), \qquad \partial_y \Psi_1(x, y_U(x)) = -D\Psi_0(y_U(x)) \tag{5.2.4a,b}$$

$$\partial_x \Psi_1(x, y_L(x)) = 0, \qquad \partial_x \Psi_1(x, y_U(x)) = 0, \tag{5.2.4c,d}$$

where the subscripts L and U refer to the lower and upper walls, respectively.

Enforcement of the flow boundary conditions on the irregular wall geometry, which represents the main computational difficulty, is carried out using the Immersed Boundary Conditions method. We shall now briefly explain the whole discretization process.

The boundary corrugations are periodic in x and thus the velocity field modifications are also periodic in x. These modifications are represented in terms of Fourier expansions where the modal functions are represented in terms of Chebyshev expansions. The flow domain is confined between $(1+Y_U)$ and $(-1-Y_L)$ (see Fig. 5.1) and, since the algorithm uses standard definition of the Chebyshev polynomials, one must map the physical (x,y) coordinates to the computational $(x,\hat{y})$ coordinates where $\hat{y} \in \langle -1,1 \rangle$. The mapping has the form

$$\hat{y} = 2\frac{y-(1+Y_U)}{1+Y_U-(-1-Y_L)}+1, \tag{5.2.5}$$

and changes the field equation to the form

$$[\partial_{xxxx} + 2\Gamma^2\partial_{xx\hat{y}\hat{y}} + \Gamma^4\partial_{\hat{y}\hat{y}\hat{y}\hat{y}}]\Psi_1 = 0, \tag{5.2.6}$$

where $\Gamma = 2/(2+Y_U+Y_L)$. The locations of the boundaries in the $(x,\hat{y})$ plane are given as

$$\hat{y}_L(x) = \sum_{n=-N_A}^{n=+N_A} A_L^{(n)} e^{in\alpha x}, \qquad \hat{y}_U(x) = \sum_{n=-N_A}^{n=+N_A} A_U^{(n)} e^{in\alpha x}, \tag{5.2.7a,b}$$

where the quantities $A_L^{(n)}$ and $A_L^{(n)}$ can be easily deduced using (5.2.1) and (5.2.5). The corresponding boundary conditions are

$$\partial_{\hat{y}}\Psi_1(x,\hat{y}_L(x)) = -D\Psi_0(\hat{y}_L(x)), \qquad \partial_{\hat{y}}\Psi_1(x,\hat{y}_U(x)) = -D\Psi_0(\hat{y}_U(x)), \tag{5.2.8a-b}$$

$$\Psi_1(x,\hat{y}_L(x)) = (\text{constant})_L, \qquad \Psi_1(x,\hat{y}_U(x)) = (\text{constant})_U, \tag{5.2.8c-d}$$

where from now and on $D = d/d\hat{y}$. The unknown stream function Ψ_1 is represented as a Fourier expansion

$$\Psi_1(x,\hat{y}) = \sum_{n=-\infty}^{n=+\infty} \Phi^{(n)}(\hat{y}) e^{in\alpha x} \approx \sum_{n=-N_M}^{n=+N_M} \Phi^{(n)}(\hat{y}) e^{in\alpha x}, \tag{5.2.9a}$$

where $\Phi^{(n)}(\hat{y}) = \left(\Phi^{(-n)}(\hat{y})\right)^*$, the expansion is substituted into the field equation (5.2.6) and the Fourier components are separated resulting in an system of uncoupled equations for the modal functions $\Phi^{(n)}$, $n \in \langle 0, N_M \rangle$, of type

$$\left[\Gamma^4 D^4 - 2(\Gamma n\alpha)^2 D^2 + (n\alpha)^4\right]\Phi^{(n)} = 0. \tag{5.2.9b}$$

The modal functions are discretized using Chebyshev expansions in the form

$$\Phi^{(n)}(\hat{y}) = \sum_{k=0}^{k=\infty} G_k^{(n)} T_k(\hat{y}) \approx \sum_{k=0}^{k=N_T} G_k^{(n)} T_k(\hat{y}) \tag{5.2.10}$$

where T_k denotes the Chebyshev polynomial of k^{th} order and $G_k^{(n)}$ are the unknown coefficients of the expansions. Substitution of (5.2.9)-(5.2.10) into (5.2.6), separation of the coefficients of different Fourier modes and application of the Galerkin projection technique [8-11] lead to the $(N_T+1)*(N_M+1)$ projection equations in the form

$$\sum_{k=0}^{k=N_T} \left[\Gamma^4 \langle T_j, D^4 T_k \rangle - \left(2n^2\alpha^2\Gamma^2\right)\langle T_j, D^2 T_k \rangle + \left(n^4\alpha^4\right)\langle T_j, T_k \rangle\right] G_k^{(n)} = 0,$$

$$j = 0,\ldots,N_T \;, |n| = 0,\ldots,N_M. \tag{5.2.11}$$

The complete set of discretized algebraic equations requires discretization of the boundary conditions (5.2.8a-d) in terms of the unknown Chebyshev coefficients $G_k^{(n)}$. The following discussion will be presented in the context of the lower wall while relations for the upper wall can be easily deduced. One needs to represent the known

values of $\partial_{\hat{y}}\Psi_1(x,\hat{y}_L(x))$ and $\Psi_1(x,\hat{y}_L(x))$ in a convenient form; both quantities are periodic in x with the period $\lambda = 2\pi/\alpha$ and thus can be expressed in terms of Fourier expansions in the form

$$\partial_{\hat{y}}\Psi_1(x,\hat{y}_L(x))=\sum_{n=-N_U}^{n=+N_U} U^{(n)} e^{in\alpha x}\,, \quad \Psi_1(x,\hat{y}_L(x))=\sum_{n=-N_U}^{n=+N_U} V^{(n)} e^{in\alpha x}\,, \tag{5.2.12a,b}$$

where $U^{(n)}$ and $V^{(n)}$ are known and the summation limit $N_U > N_M$ while the actual value of N_U will be determined later. The terms $\partial_{\hat{y}}\Psi_1(x,\hat{y}_L(x))$ and $\Psi_1(x,\hat{y}_L(x))$ can be represented using the discretized form of the solution as

$$\partial_{\hat{y}}\Psi_1(x,\hat{y}_L(x))=\Gamma\sum_{n=-N_M}^{n=+N_M}\sum_{k=0}^{k=N_T} G_k^{(n)} DT_k(\hat{y}_L(x)) e^{in\alpha x}\,,$$

$$\Psi_1(x,\hat{y}_L(x))=\sum_{n=-N_M}^{n=+N_M}\sum_{k=0}^{k=N_T} G_k^{(n)} T_k(\hat{y}_L(x)) e^{in\alpha x}\,. \tag{5.2.13a,b}$$

Chebyshev polynomials of order k and their derivatives evaluated at $\hat{y}_L(x)$ are periodic functions of x and can be represented as

$$T_k(\hat{y}_L(x)) = \sum_{m=-N_S}^{m=+N_S} (w_L)_k^{(m)} e^{im\alpha x}\,, \qquad DT_k(\hat{y}_L(x)) = \sum_{m=-N_S}^{m=+N_S} (d_L)_k^{(m)} e^{im\alpha x}\,, \tag{5.2.14a,b}$$

where $\max(N_S)=N_T*N_A$. The coefficients $w_k^{(m)}$ and $d_k^{(m)}$ are known from the properties of Chebyshev polynomials. Substitution of (5.2.14) into (5.2.13) results in

$$\partial_{\hat{y}}\Psi_1(x,\hat{y}_L(x))=\Gamma\sum_{h=-N_U}^{h=+N_U}\sum_{n=-N_M}^{n=+N_M}\sum_{k=0}^{k=N_T} G_k^{(n)} (d_L)_k^{(h-n)} e^{ih\alpha x}\,,$$

$$\Psi_1(x,\hat{y}_L(x))=\sum_{h=-N_U}^{h=+N_U}\sum_{n=-N_M}^{n=+N_M}\sum_{k=0}^{k=N_T} G_k^{(n)}\left(w_L\right)_k^{(h-n)} e^{ih\alpha x}\,, \qquad (5.2.15a,b)$$

where $h=n+m$ and $N_U=N_TN_A+N_M$. $D\Psi_0(\hat{y}_{L,U}(x))$ are known and can be expressed in terms of Fourier expansion as

$$D\Psi_0(\hat{y}_{L,U}(x))=\sum_{n=-N_U}^{n=+N_U} F_{L,U}^{(n)}\, e^{in\alpha x}\,. \qquad (5.2.16)$$

Renaming of indices n by m and h by n in Eq. (5.2.15) and comparison of the resulting expressions with (5.2.12) provides the expressions for $U^{(n)}$ and $V^{(n)}$ that form the basis of boundary constraints. Substitution of these expressions along with (5.2.16) into the boundary conditions (5.2.8) results in the final form of boundary constraints, i.e.,

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T} G_k^{(m)}\left(d_{L,U}\right)_k^{(n-m)} = -\Gamma^{-1}F_{L,U}^{(n)}\,, \quad \text{for } |n|\geq 0 \qquad (5.2.17a)$$

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T} G_k^{(m)}\left(w_{L,U}\right)_k^{(n-m)} = 0\,, \quad \text{for } n\neq 0 \qquad (5.2.17b)$$

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T} G_k^{(m)}\left(w_{L,U}\right)_k^{(-m)} = \left(\text{constant}\right)_{L,U}\,, \qquad (5.2.17c)$$

where $|n|=0,...,N_U$ and the subscripts L and U denote lower and upper walls respectively. The form of constants that appear in Eq. (5.2.17c) can be determined using either the constant volume flux or the constant pressure gradient constraint. For convenience, we have selected the former one for this presentation, i.e., we assumed that the flow rates in the smooth reference channel and in the corrugated channel remain the same. As the definition of stream function is accurate up to a constant, one of the

constants can be assumed arbitrarily. For this analysis we have assumed the stream function to be zero at the lower wall, which leads to

$$\left(\text{constant}\right)_L = 0, \quad \left(\text{constant}\right)_U = Q^{(0)} - \Xi_U^{(0)} + \Xi_L^{(0)}. \tag{5.2.18a,b}$$

where $\Psi_0(\hat{y}_{L,U}(x)) = \sum_{n=-M_M}^{n=+M_M} \Xi_{L,U}^{(n)} e^{inax}$. In all calculations presented in this chapter, the volume flux through the corrugated channel has been taken as $Q^{(0)} = 4/3$, i.e., equal to the volume flux for Poiseuille flow in a smooth channel.

The discretized modal equations (5.2.9)-(5.2.11) and the discretized boundary constraints (5.2.17)-(5.2.18) provide building blocks for the construction of different versions of the IBC method.

There are two issues that are critical to the applicability of the IBC method outlined above. The modal functions are decoupled and could be considered separately in the solution process in the case of smooth walls. This leads to the numerical solution of several small linear systems which can be done efficiently and forms basis of the spectral DNS simulations of transition/turbulence in channel flows [12-13]. In the case of corrugated walls the modal functions become coupled due to boundary constraints and this necessitates numerical solution of a potentially very large linear system which, in addition, could be ill-conditioned. The reader should note that the structure of the algebraic problem remains the same in the case of the full Navier-Stokes equations as the nonlinear terms are evaluated on the basis of information taken from the previous iteration and thus end up on the right hand side of the linear system without providing any explicit coupling between the modal functions. The coupling between modal functions originates only from the geometry of the flow domain and cannot be removed through numerical operations. We need, therefore, to develop efficient solvers that can deal with such linear systems.

The second issue is related to the stream-wise discretization, which involves two Fourier expansions, one for the field equations (5.2.9)-(5.2.11) and one for the boundary constraints (5.2.15). These expansions may have different rates of convergence and thus selection of the correct truncation may affect the overall accuracy of the discretization as well as the size of the linear system to be solved. As the geometry of the corrugation becomes more demanding, one should expect reduction of the convergence rate of the expansions related to boundary constraints as the field equations do not directly carry information about the shape of the channel. Equation (5.2.11) provides N_T+1 projection equations for each modal function and the complete solution is defined using $(2N_M+1)$ modes. Equations (5.2.17)-(5.2.18) provide a pool of $4(2N_U+1)$ boundary constraints with $N_U = (N_T N_A+N_M)$ being larger than N_M. The classical formulation of the IBC method follows the tau procedure where four projection equations are eliminated for each modal function providing degrees of freedom required in order to enforce the flow boundary conditions. $4(2N_M+1)$ boundary constraints obtained from Eqs. (5.2.17)-(5.2.18) using $|n| = 0,...,N_M$ are employed to form a closed system of equations. The values of $U^{(n)}$ and $V^{(n)}$ for $|n|>(N_M+1)$ provide a measure of error in the enforcement of flow boundary conditions. If one of the Fourier expansions has slower rate of convergence, it could be advantageous to include more terms in the solution but this leads to the over-determined problem formulation.

Investigations reported in [11] showed that the use of additional projection equations did not improve the accuracy of the solution while the use of additional boundary relations did. Therefore, in the present work, we shall be retaining N_T-3 projection equations for each Fourier mode and the system of equations will be made over-determined by enforcing $4(2M_M+1)$ boundary constraints of type (5.2.17)-(5.2.18) using $|n| = 0,...,M_M$ where $N_U \geq M_M > N_M$. This process leads to the least-square solution of a potentially very large and ill posed linear system, and this necessitates search for an efficient solver.

5.3. Solvers suitable for the classical formulation of the IBC method

The linear system of discretized algebraic equations involved in the classical formulation has the general form

$$\mathbf{Lx=Z}, \tag{5.3.1}$$

where **L** is a p×p coefficient matrix with $p = (2N_M+1)(N_T+1)$, **x** is a p-dimensional column vector of unknowns and **Z** is a p-dimensional column vector whose values are known. The system is organized by grouping entries corresponding to the projection equations in matrix **H** of size q×p, $q = (2N_M+1)(N_T-3)$, and entries corresponding to boundary relations in matrix **K** of size (p-q)×p, resulting in the structure of coefficient matrix **L** illustrated in Fig. 5.2. Matrix **H** has the block diagonal structure with each block corresponding to a different modal equation and having the size $(N_T+1)\times(N_T-3)$. Matrix **K** is full and provides coupling among the different Fourier modes.

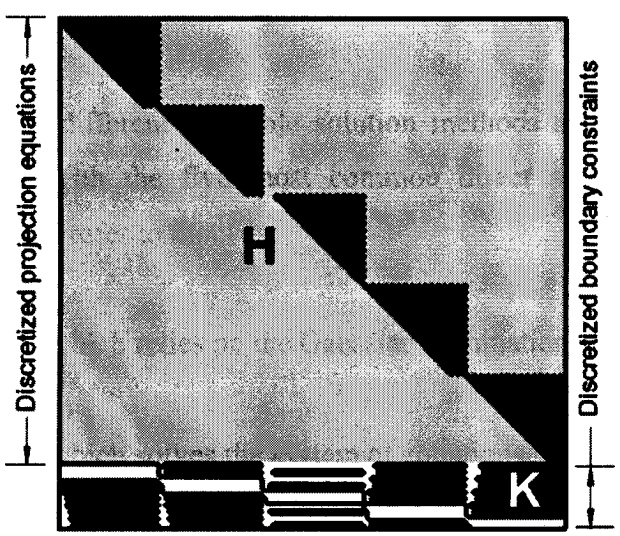


Figure 5.2. *Structure of the coefficient matrix arising in the context of the classical formulation of the IBC method constructed using N_M=2 Fourier modes and N_T=30 Chebyshev polynomials for the model problem (5.2.2)-(5.2.4). The black entries correspond to the non-zero elements.*

5.3.1. Standard direct solvers

The system (3.1) can be solved either directly or iteratively. The coefficient matrix **L**, whose structure is illustrated in Fig. 2, has a large proportion of zero elements. Sparsity of a matrix is generally defined as the ratio of the number of nonzero elements to the total number of elements. The sparsity of **L** can be approximately evaluated as $[(2N_M+1)(N_T-3)(N_T+1)+ 4(2N_M+1)^2 (N_T+1)]/[(2N_M+1)^2 (N_T+1)^2]$, i.e., it is inversely proportional to the number of Fourier modes and Chebyshev polynomials used in the computations. In other words, any increase in the number of Fourier modes and/or Chebyshev polynomials lead to the increase of the sparsity (as well as the size of the matrix). The reader may note that any change in the number of Fourier modes influences the sparsity more than a similar change in the number of Chebyshev polynomials. Treatment of **L** as a sparse matrix accompanied by the use of special solvers tuned for such matrices [14] can lead to potential efficiency gains for two reasons, i.e., i) data structures for sparse matrices require less computer memory, and ii) the algorithms developed for sparse matrices avoid operations on zero elements and thus are more efficient [14].

In order to characterize different available solution methods and to identify the most suitable one, we start with the five most common direct solvers using MATLAB implementation. We shall refer to them as:

i) *Classical Method A,* which relies on the Gaussian elimination,

ii) *Classical Method B*, which solves the system of equations using Gaussian elimination and taking advantage of the sparse structure of the coefficient matrix (**L** is first pre-processed as a sparse matrix [14] and the system of equations is then solved using a specialized solver that performs Gaussian elimination on sparse matrices),

iii) *Classical Method C,* which is based on the LU decomposition,

iv) *Classical Method D,* which is also based on the LU decomposition, however, it first converts the coefficient matrix **L** into a sparse matrix and then uses specialized solver for computing LU factors of sparse matrices,

v) *Classical Method E,* which solves the system of equations by computing the inverse of the coefficient matrix **L**.

In the following discussion method A is used as the reference method. The performance gains achieved by the other methods are quantified as the ratio of computational time involved in solving (5.3.1) by method A to that required by the method in question and is denoted by $\text{Gain}_{\text{Classical}}$. Larger values of $\text{Gain}_{\text{Classical}}$ correspond to more efficient methods.

The relative performances of different solvers are assessed using a convenient test problem. Here we have assumed that the test channel has flat upper wall and a wavy lower wall, i.e., its geometry is defined as

$$y_L(x) = -1 + S\cos(\alpha x), \qquad y_U(x) = 1, \qquad (5.3.2a,b)$$

where S denotes the corrugation amplitude and α stands for the wave number of corrugation.

The performance gains for methods B, C, D and E are illustrated in Figure 5.3. It can be seen that method E, i.e., computing the inverse of **L** is the least efficient, while method B that performs Gaussian elimination on **L** after rearranging it as a sparse matrix is the most efficient. Method C or LU decomposition of **L** delivers similar performance as the reference method (method A). Figure 5.3 also shows that performing LU decomposition on **L** converted into a sparse matrix provides significant performance improvement, as expected. Performance gains obtained using methods C and E are almost independent of the problem size for higher values of N_M and N_T, while the efficiencies of methods B and D become more prominent with increasing values of N_M, which results in an increase in

the number of blocks of the same size (Fig. 5.3A) as well as a higher degree of sparsity of **L**. Efficiencies of methods B and D remain almost unchanged when one works with a constant number of blocks of increasing size (Fig. 5.3B) due to relatively smaller change in the sparsity of **L**.

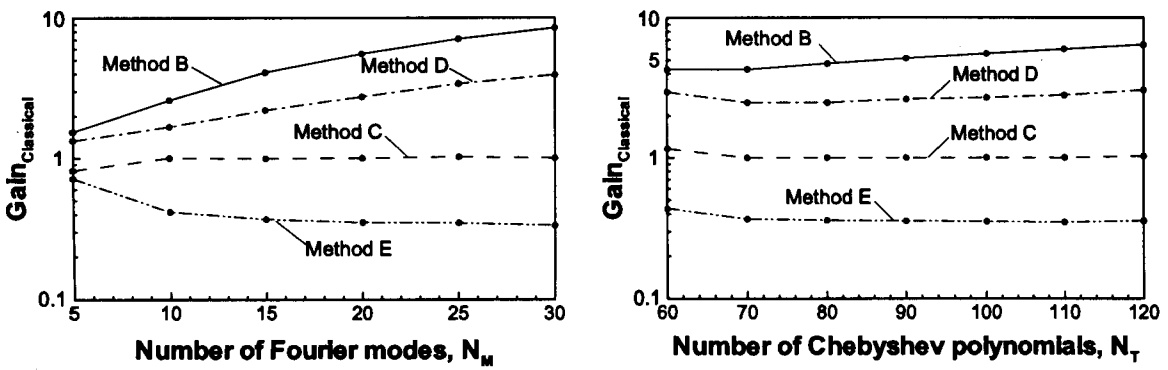


Figure 5.3. *Variations of the performance gains associated with the use of the classical methods B, C, D and E as a function of the number of Fourier modes N_M (Fig. 5.3A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.3B, N_M=20) used in the computations. The tests have been carried out using the model problem (5.3.2) with α=3.0 and S=0.05.*

5.3.2. The domain transformation method

The main advantage of the IBC method in solving flow problems involving irregular boundaries lies in the spectral convergence characteristics of the spatial discretization. The Domain Transformation (DT) method is another algorithm capable of delivering spectral convergence, which is based on analytical mapping of the irregular flow domain into a regular computational domain [8, 9, 11]. The mapping results in the classical treatment of boundary conditions at the expense of significant complication of the field equations. This method can be viewed as a representative of methods involving construction of boundary conforming grids; such grids in general need to be constructed

numerically, however, for the DT method this part of numerical cost is avoided through an analytical mapping. The DT method serves two purposes in the current work. First, it is used to illustrate the comparative performance of different versions of the classical IBC method. Secondly, it is used to calculate the machine-accurate reference solutions that are useful when characterizing error of the different over-determined versions of the IBC method to be discussed in the next section. The solution for the DT method is obtained using the same spatial discretization methodology as used for the IBC method, i.e., Fourier expansions in the x-direction and Chebyshev expansions in the y-direction. Further details on the DT method can be found in Appendix F.

The performance gains for different classical IBC methods over the DT method are quantified as the ratio of computational time required to determine solution of the model problem (5.2.2)-(5.2.4), (5.3.2) by the DT method to that required by the classical method in question and is denoted by $\text{Gain}_{\text{Domain Transformation}}$. Larger values of $\text{Gain}_{\text{Domain Transformation}}$ signify disadvantage of the DT method as compared against the classical IBC method in question. Figure 5.4 reveals that the reference IBC method A performs significantly better than the DT method for any given combination of Fourier modes and Chebyshev polynomials. Unlike the IBC method, the modal field equations in the case of the DT method are coupled (see Appendix F), which increases the cost of constructing the coefficient matrix substantially and, therefore, is responsible for the huge difference in the relative performance. The performance data shown in Fig. 5.4 provide however an incomplete picture as it is known that the DT method yields better accuracy compared to the IBC method for smaller number Fourier modes and/or Chebyshev polynomials [8].

Figure 5.5 provides information that helps in assessing this aspect of the performance, however, one needs to define a suitable measure of error before proceeding with the discussion. We shall use the L_∞ norms defined as

$$\|u_{er}\|_\infty = \sup_{\substack{0\le x\le 2\pi/\alpha \\ y_L(x)\le x\le y_U(x)}} |u_{er}(x,y)|, \qquad \|v_{er}\|_\infty = \sup_{\substack{0\le x\le 2\pi/\alpha \\ y_L(x)\le X\le y_U(x)}} |v_{er}(x,y)|, \qquad (5.3.3a,b)$$

in order to quantify the error for the DT and IBC methods over the whole flow domain. The quantities $u_{er}(x,y)$ and $v_{er}(x,y)$ for the DT method are computed using a sufficiently high number of Chebyshev polynomials ($\geq$100) and then looking at variations of properties of the solution as a function of the number of Fourier modes N_M used in the computations. This process is analogous to the grid-convergence testing techniques widely used for the methods involving standard finite-difference, finite-volume or finite-element discretizations. In the case of the IBC methods, the quantities $u_{er}(x,y)$ and $v_{er}(x,y)$ are computed in a different manner, i.e., using the following relations

$$u_{er}(x,y)=u_{IBC}(x,y)-u_{DT}(x,y), \qquad u_{er}(x,y)=u_{IBC}(x,y)-u_{DT}(x,y), \quad (5.3.4a,b)$$

where the subscript 'IBC' denotes solution obtained using the IBC method and the subscript 'DT' denotes the machine-accurate reference solution determined using the DT method. For the IBC methods, a second measure of error associated with the accuracy in the enforcement of the physical boundary conditions is used. This error is defined using the following L_∞ norms

$$\left\|\left(u_{er,BC}(x)\right)_{L,U}\right\|_\infty = \sup_{0\leq x\leq 2\pi/\alpha}\left|\left(u_{er,BC}(x)\right)_{L,U}\right|,$$

$$\left\|\left(v_{er,BC}(x)\right)_{L,U}\right\|_\infty = \sup_{0\leq x\leq 2\pi/\alpha}\left|\left(v_{er,BC}(x)\right)_{L,U}\right|, \quad (5.3.5a,b)$$

where,

$$\left(u_{er,BC}(x)\right)_{L,U}=u(x,y_{L,U}(x)), \qquad \left(v_{er,BC}(x)\right)_{L,U}=v(x,y_{L,U}(x)). \quad (5.3.6a,b)$$

The subscripts L and U in Eqs. (5.3.5)-(5.3.6) denote the lower and upper walls of the channel, respectively. Since the error in the u-velocity component is related to the error

of the y-derivative of the stream function, which needs to be evaluated numerically and thus is potentially less accurate than evaluation of v, we shall focus our attention on u_{er} in the further discussion. Due to the particular definition of the test geometry (5.3.2), we shall use the norm $\|u_{er,BC}\|_\infty \equiv \|u_{er,BC}(x)_L\|_\infty$ to measure the solution error.

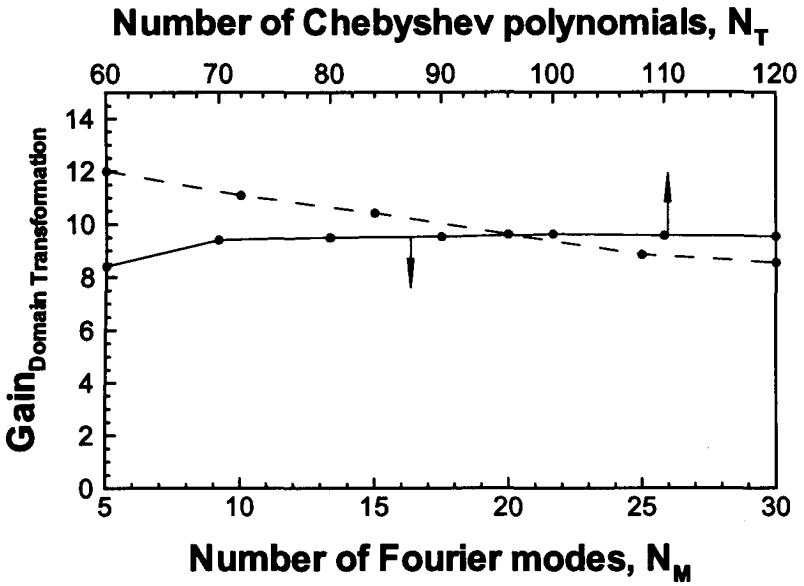


Figure 5.4. *Performance gains associated with the use of the classical method A as compared with the domain transformation method as a function of the number of Fourier modes N_M (solid line, N_T=100) and as a function of the number of Chebyshev polynomials N_T (dash line, N_M=20) used in the computations. Other conditions as in Fig. 5.3.*

It has been established that in the case of the classical IBC methods the error in the enforcement of the physical boundary conditions provides a very convenient and strict measure of the overall error, provided a sufficient number of Chebyshev polynomials has been used to represent the solution in (5.2.10) [8-10]. In other words, if the solutions are independent of any increment of N_T then the maximum error over the computational domain becomes equal to the maximum error in the enforcement of the flow boundary conditions, i.e.,

$$\|u_{er}\|_\infty = \|u_{er,BC}\|_\infty \text{ when } \frac{d}{dN_T}(\|u_{er}\|_\infty) \approx 0. \tag{5.3.7}$$

The above definitions provide basis for discussion of the results displayed in Fig. 5.5A, which illustrate variations of the performance gains of the IBC methods A and C with respect to the DT method as a function of the corrugation amplitude S. It can be seen that the IBC methods outperform the DT method, however, the advantage decreases as the number of Fourier modes that must be used in the computations in order to deliver the desired accuracy increases (see Fig. 5.5B). The relative advantage of the method A diminishes rapidly with increasing value of S while method C maintains considerable advantage. One can conclude from the above discussion that the IBC method with a standard "un-optimized" but properly selected linear solver is computationally significantly more efficient then the DT method and, by implication, is more efficient then any method relying on the construction of boundary-conforming grids.

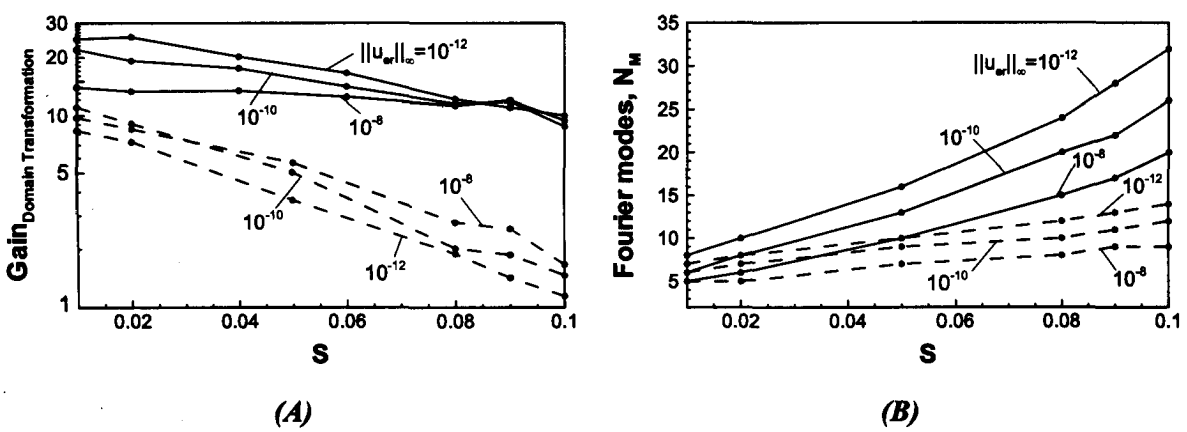


Figure 5.5. *Variations of the performance gains associated with the use of the classical methods A (dash line) and C (solid line) as compared with the DT method as a function the corrugation amplitude S (Fig. 5.5A). Tests have been carried out for different levels of solution accuracy. The number of Fourier modes N_M required to reach the desired accuracy is shown in Fig. 5.5B. Other conditions as in Fig. 5.3.*

We shall now focus our attention on the issue of development of specialized solvers that can take advantage of the special structure of the coefficient matrix.

5.3.3. Iterative solvers

Analyses of three-dimensional flows as well as flows in channels with complicated boundaries may require a large number of Fourier modes N_M and/or Chebyshev polynomials N_T in order to deliver an acceptable accuracy. Use of large values of N_M and N_T poses two inter-connected difficulties, i.e., the size of the problem increases resulting in a potentially unacceptable execution time and memory requirements increase making problem potentially un-accessible to the standard hardware. The use of the sparse matrix techniques, as discussed in Section 3.1, reduces the memory requirements and the operation count, with the former one being much more significant [11]. Therefore, the performance gains of sparse solvers as compared with the direct Gaussian elimination of the full matrix are found to decrease with increasing the size of the problems. In addition, storing matrices in a sparse form requires storing locations in addition to the values of the nonzero elements of the matrix and thus can play a limiting role when one needs to use a very large number of Fourier modes and Chebyshev polynomials. Therefore, the development of specialized linear solvers that address the issues of computer memory and computational efficiency is necessary in order to provide access to parameter ranges that are expected to be of interest in applications. We shall present two distinct types of solution approaches that address the above issues.

The first approach is based on the concept of mode decoupling [8-10]. Since the projection equations (5.2.11) are not coupled directly and the only coupling is provided by the discretized boundary constraints (5.2.17)-(5.2.18), one may consider the projection equations and the boundary constraints separately. In the iterative solvers, the unknowns corresponding to a Fourier mode of interest are expressed in terms of the remaining Fourier modes using their values from the previous iteration. For the first iteration a suitable initial approximation is used. Two versions of the iterative solvers were tested, namely the Gauss-Seidel and the Jacobi iterative methods. In the case of the Gauss-Seidel method, the solution process begins with mode 0 and proceeds to the next mode using the most recent information available and continues until the last mode N_M is reached, and then it is repeated until a convergence criterion ε is satisfied. In the Jacobi

iteration the solution computed for any mode during any given iteration is not affected by the solutions computed for the remaining modes during the same iteration, thereby providing the opportunity for potential parallelization of the iterative process. The convergence criterion ε used here for both methods is based on the absolute difference between the solutions from two consecutive iterations.

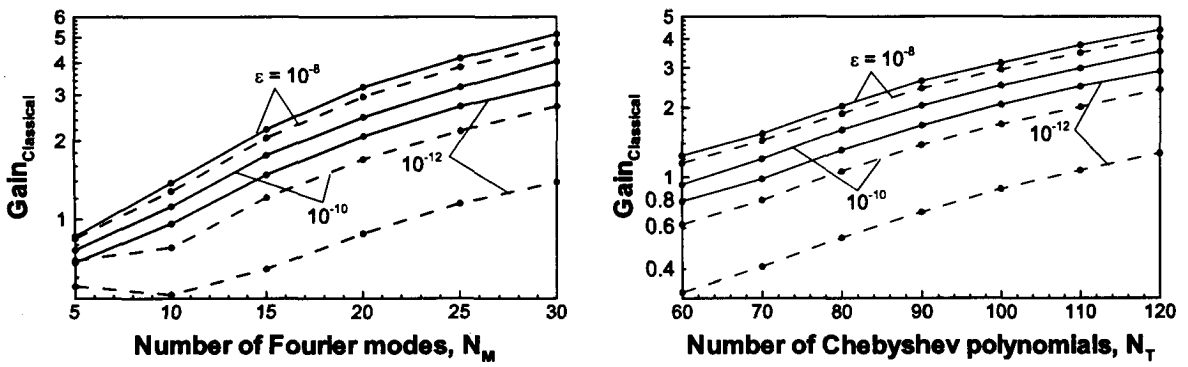


Figure 5.6. *Variations of the performance gains associated with the use of the Gauss-Seidel (solid line) and Jacobi (dash line) iterative methods as a function of the number of Fourier modes N_M (Fig. 5.6A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.6B, N_M=20) used in the computations for different values of convergence criterion ε. Other conditions as in Fig. 5.3*

Figure 5.6 demonstrates that the performance gains obtained by both iterative techniques increase with increasing either the number of Fourier modes or the number of Chebyshev polynomials used in the computations. While performance of the both methods is enhanced for larger values of ε, i.e., for lower accuracy, it should be also noticed that the Jacobi method is slightly slower compared to the Gauss-Seidel method as the Jacobi method requires more iterations to satisfy a given convergence criterion. Parallelization tests based on the Jacobi method and a small number of processors (max 4) did not show any efficiency gains and therefore, results in this regard are not presented.

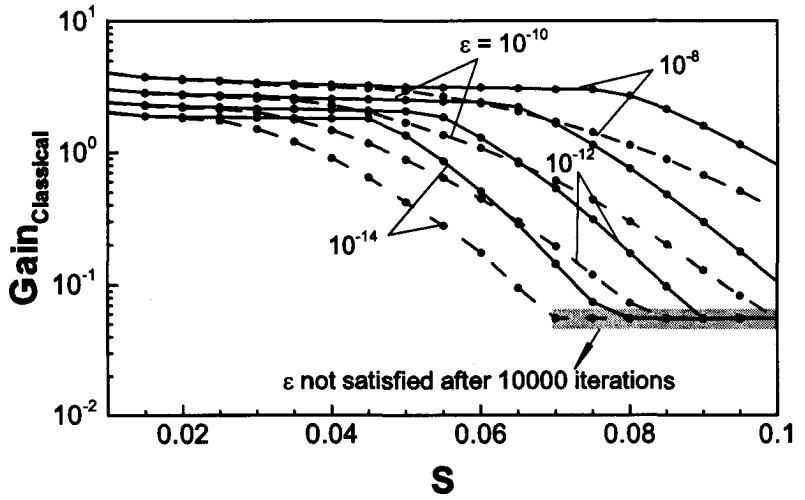


Figure 5.7. *Variations of the performance gains associated with the use of the Jacobi (dash line) and Gauss-Seidel (solid line) iterative methods as a function the corrugation amplitude S for different convergence criterion ε. Computations were carried out with N_M=20 and N_T=100. Other conditions as in Fig. 5.3.*

Figure 5.7 shows that, although the iterative solvers perform better for smaller corrugation amplitudes S for a given corrugation wave number α, they do not guarantee convergence. The no-convergence or extremely slow convergence problems limits applicability of these methods to less extreme geometries, i.e., to small enough S and small enough α. On the good side, these methods are memory efficient as one needs to store only diagonal blocks in matrix **H** and along with matrix **K** from the general matrix **L** (see Fig. 5.8).

5.3.4. Specialized direct solvers

The second group of specialized methods to be explored represents direct solvers that take advantage to the special structure of the coefficient matrix **L**. Part **H** of **L** is a block diagonal matrix where each block corresponds to a different Fourier mode, and part **K** is a full matrix corresponding to the discretized boundary constraints.

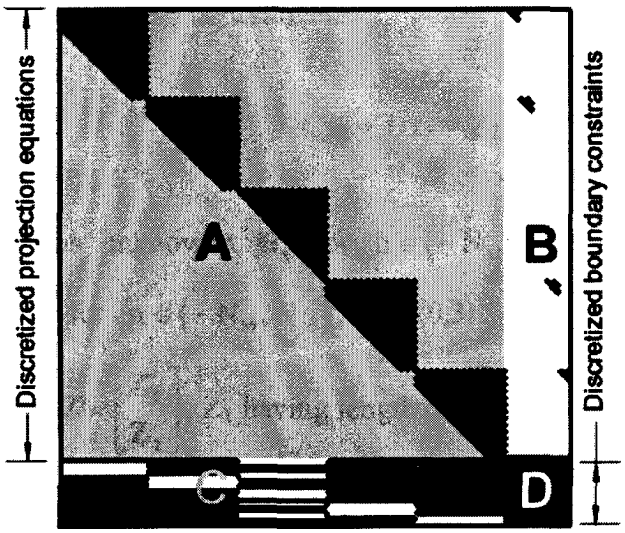


Figure 5.8. *Structure of the re-arranged coefficient matrix arising in the context of the classical formulation of the IBC method constructed using N_M=2 Fourier modes and N_T=30 Chebyshev polynomials for the model problem (5.2.2)-(5.2.4). The black entries correspond to the non-zero elements.*

We begin by extracting the largest square matrix **A** of the size q×q from **H**; this requires reorganization of the vector of unknowns by relocating four coefficients of the Chebyshev expansions for each Fourier mode to the end of the vector. Selection of the coefficients for the relocation is dictated by the structure of the block matrices in **H**, especially the block corresponding to mode zero which has an upper triangular structure. Removal of the last four coefficients from this block would lead to an under-determined non-invertible matrix. Therefore, one is free to remove any four unknowns from the diagonal blocks except the last four. However, it can be shown that the first four unknowns of mode zero have zero coefficients. As a result, we have opted for removing the coefficients of the first four unknowns for each mode resulting in the structure of coefficient matrix as illustrated in Fig. 8. The resultant square matrix **A** of size q×q has a block diagonal structure with each block of size $(N_T-3)\times(N_T-3)$. The rectangular matrix **B** of size q×(p-q) also has a block diagonal structure with each block of size $(N_T-3)\times4$ while

the full rectangular matrix **C** has size (p-q)×q and the full square matrix **D** has size (p-q)×(p-q). The complete system can be written as

$$\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2 = \mathbf{Z}_1, \qquad \mathbf{C}\mathbf{x}_1 + \mathbf{D}\mathbf{x}_2 = \mathbf{Z}_2, \tag{5.3.8a,b}$$

where vector $\mathbf{x}_1$ contains unknowns $G_k^{(n)}$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 4, N_T \rangle$, and $\mathbf{x}_2$ contains unknowns $G_k^{(n)}$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 0,3 \rangle$. The right hand side vector remains unchanged with $\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix}$, $\mathbf{Z}_1$ having length q and $\mathbf{Z}_2$ having length (p-q). **B** and **D** contain information associated with $G_0^{(n)}$, $G_1^{(n)}$, $G_2^{(n)}$ and $G_3^{(n)}$, while **A** and **C** contain the rest. Solution of (5.3.8) can be written as

$$\mathbf{x}_2 = \mathbf{E}^{-1}\mathbf{Z}_2 - \mathbf{E}^{-1}\mathbf{C}\mathbf{A}^{-1}\mathbf{Z}_1, \qquad \mathbf{x}_1 = \mathbf{A}^{-1}\mathbf{Z}_1 - \mathbf{A}^{-1}\mathbf{B}\mathbf{x}_2, \tag{5.3.9a,b}$$

where $\mathbf{E} = (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ has the size (p-q)×(p-q).

The approach described by (3.8)-(3.9) is based on the concept of Schur complement [15] which reduces the solution of system of size $[(2N_M+1)(N_T+1)]\times[(2N_M+1)(N_T+1)]$ into a solution of two systems of sizes $[(2N_M+1)(N_T-3)]\times[(2N_M+1)(N_T-3)]$ and $[4(2N_M+1)]\times[4(2N_M+1)]$. However, the potential advantage associated with the use of the Schur complement can be extended even further by considering the fact that the quantities $\mathbf{A}^{-1}\mathbf{Z}_1$ and $\mathbf{A}^{-1}\mathbf{B}$ can be computed block by block instead of operating on the complete **A** and **B**. In other words, the process of inversion of a big matrix **A** of the size $[(2N_M+1)(N_T-3)]\times[(2N_M+1)(N_T-3)]$ is reduced into a process of inverting $(2N_M+1)$ small sub-matrices of the size $(N_T-3)\times(N_T-3)$. The number of arithmetic operations in this approach would be significantly smaller than those required by any of the standard direct methods, including the sparse solvers, as one will be computing the inverses or equivalent factors of a number of small sub-matrices. This solution process brings the performance of the method to the level found in the case of smooth walls where the geometry-based

coupling between different modal function does not exist (see discussion at the end of Section 2). The method also reduces the memory requirements substantially as only the non-zero diagonal blocks in the **A** and **B** are needed to be stored instead of the complete **A** and **B**. In order to avoid the numerical issues associated with the process of computing inverse of ill-conditioned matrices the solution method relies on LU decomposition to compute the quantities $\mathbf{A}^{-1}\mathbf{Z}_1$ and $\mathbf{A}^{-1}\mathbf{B}$. This solution method is referred to as the *Classical Method F.*

The blocks in **A** and **B** corresponding to mode n are complex conjugates of those corresponding to mode –n. The *Classical Method G* takes advantage of this property and computes $\mathbf{A}^{-1}$ by inverting (N_M+1) small matrices of the size $(N_T-3)\times(N_T-3)$.

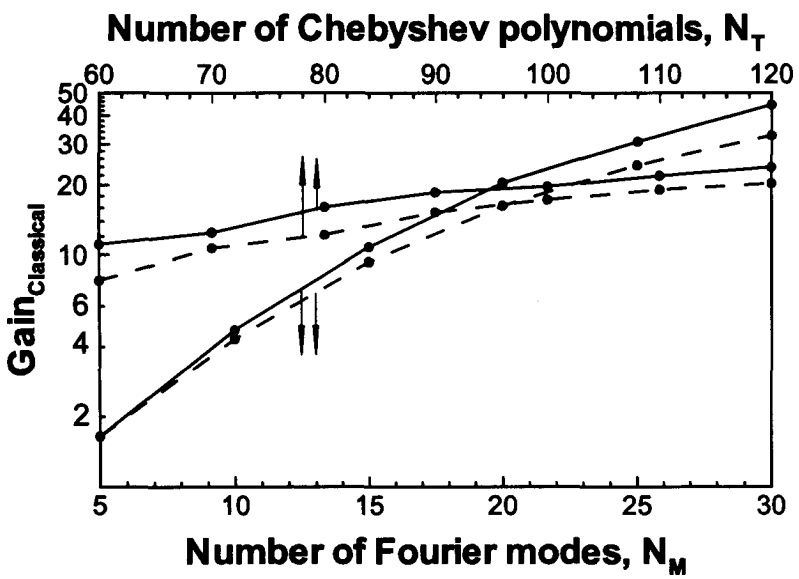


Figure 5.9. *Variations of the performance gains associated with the use of the classical methods F (dash lines) and G (solid lines) as a function of the number of Fourier modes N_M (Fig. 5.9A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.9B, N_M=20) used in the computations. Other conditions as in Fig. 5.3.*

Unlike the iterative solvers discussed in Section 5.3.3, the specialized methods F and G do not impose any restriction on the usability of the classical IBC algorithm as the

solutions are computed directly. Figure 5.9 illustrates performance of methods F and G (as compared against method A) as a function of the number of Fourier modes N_M and Chebyshev polynomials N_T used in the computations. The performance of both methods increases significantly as the number blocks $(2N_M+1)$ in A is increased and, particularly, for the method G the gain increases from approximately 1 to 50. Increasing the size of individual blocks while keeping the number of blocks unchanged also increases the performance gains; in the case of method G the gain increases from approximately 10 to 20 when N_T increases from 60 to 120 while N_M is kept fixed at 20. In general, the G method provides better performance compared to the F method and its advantage increases for larger values of N_M and N_T, which is expected.

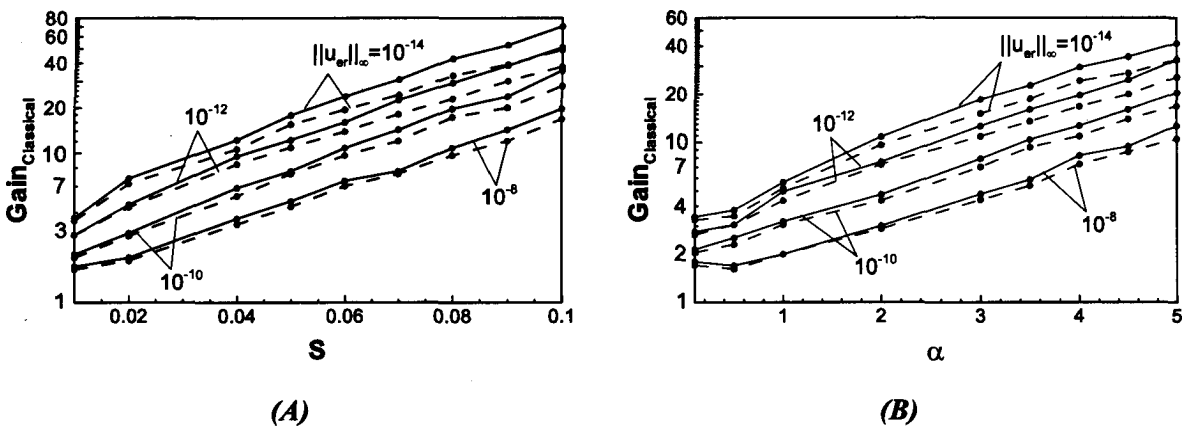


Figure 5.10. *Variations of the performance gains delivered by the classical methods F (dash line) and G (solid line) as a function of the corrugation amplitude S (Fig. 5.10A, α=3.0) and as a function of the corrugation wave number α (Fig. 5.10B, S=0.05). Computations were carried out using N_T=100 Chebyshev polynomials while the number of Fourier modes N_M has been selected in order to meet the desired accuracy stated in terms of $\|u_{er}(x,y)\|_\infty$ and thus depended on the values of S and α.*

Figure 5.10 shows performance enhancements provided by methods F and G as a function of the corrugation amplitude S for a fixed corrugation wave number (Fig. 5.10A), as well as a function of the corrugation wave number α for a fixed corrugation amplitude (Fig. 5.10B). The reader may recall that the number of Fourier modes used in

the computations has to increase in order to maintain the specified accuracy when the severity of the corrugation geometry increases. Results displayed in Fig. 5.10 demonstrate that the advantage of the F and G methods increase exponentially with an increase of the severity of geometry (as measured by higher values of S and α). It can also be seen that the advantage of methods F and G increases when higher accuracy is desired, i.e., when value of $\|u_{er}\|_\infty$ decreases, and this trend is opposite to that found in the case of iterative solvers. The reader should note that the specialized solvers are about an order of magnitude faster than the iterative solvers (compare Figs. 5.6 and 5.9-5.10) and obviously does not suffer from the convergence problems.

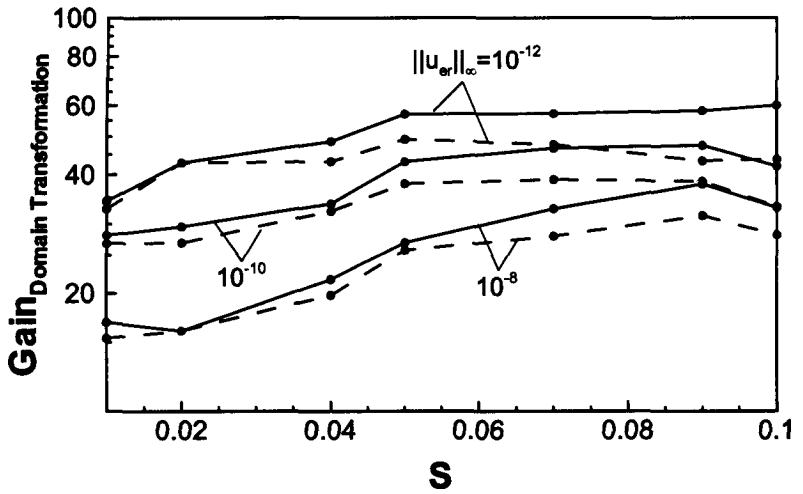


Figure 5.11. *Variations of the performance gains associated with the use of the classical methods F (dash line) and G (solid line) as compared with the DT method as a function of the corrugation amplitude S. Tests have been carried out for different levels of solution accuracy. The number of Fourier modes N_M required to reach the desired accuracy is shown in Fig.5B. Other conditions as in Fig. 5.3.*

A comparison of performance between the DT method and the classical IBC F and G methods is shown in Figure 5.11. It can be seen that the F and G methods provide substantial acceleration of computations, particularly for higher values of the corrugation amplitudes. Unlike the method A, the performance of the F and G methods increase as

the desired solution accuracy is increased. For the target accuracy of $\|u_{er}\|_\infty = 10^{-12}$ for the corrugation with S=0.1 and α=3, method G computes the solution approximately 50 times faster than the DT method, while the fastest of the direct methods that operate on the complete **L** matrix (method C) computes solution approximately 11 faster (see Figure 5.5A).

5.3.5. Memory requirements

Memory requirement imposes another potentially severe restriction. Typically, computations of the inverse or LU factors of any matrix require twice the memory needed to hold the information pertinent to the matrix itself. The most common implementations of the IBC algorithm [8-10] rely on the construction and use of complete **L** resulting in very high memory requirements when a large number of Fourier modes and Chebyshev polynomials need to be used. However, the specialized classical IBC methods F and G work by computing LU factors of the individual blocks of **A** thereby computing individual blocks for the quantities $\mathbf{A^{-1}Z_1}$, $\mathbf{A^{-1}B}$ and $\mathbf{CA^{-1}B}$. In that case only the nonzero diagonal blocks of $\mathbf{A^{-1}}$ and **B** need to be stored in the memory. The amount of memory required by the iterative solvers is equivalent to that required by the specialized direct solvers (methods F and G), however, these methods are limited in usability due to the restrictions imposed by their convergence characteristics.

Figure 5.12 shows the reduction in memory usage by the classical IBC methods F and G over the classical method A as a function of the number of Fourier modes as well as Chebyshev polynomials used to represent the solution. This figure reveals that memory savings obtained by the methods F and G grows rapidly with the increase in the values of N_M and N_T with method G being the most memory efficient, as expected. In the case of 30 Fourier modes and 100 Chebyshev polynomials, method G uses approximately 40 times less memory compared to method A, while the savings by method F is approximately 34.

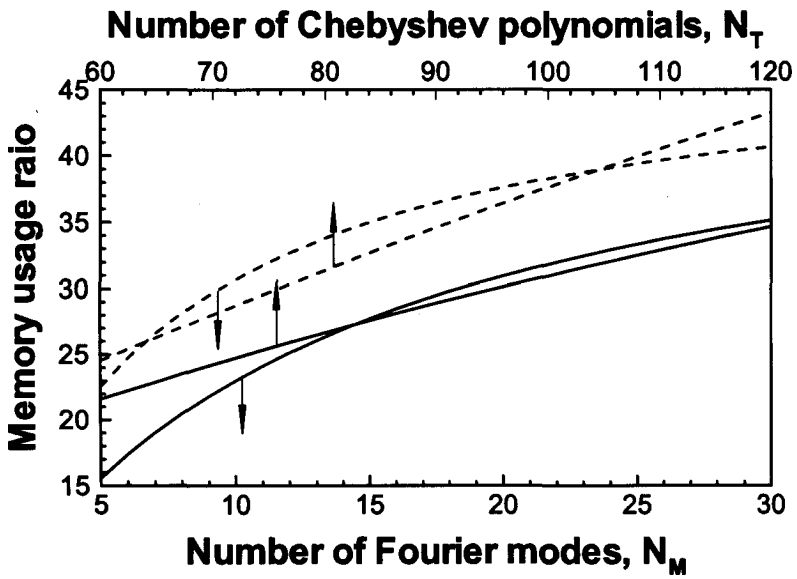


Figure 5.12. *Variations of the memory use by the classical IBC methods F (solid line) and G (dash line) compared to the classical method A as a function of the number of Fourier modes N_M (with N_T=100) and Chebyshev polynomials N_T (with N_M=20) used in the computations. Other conditions as in Fig. 5.3.*

5.4. Solvers suitable for the over-determined formulation of the IBC method

The classical formulation of the IBC method has been successfully employed for various classes of problems [8-10]. However, its applicability is limited by the severity of the boundary geometries that can be handled with a reasonable accuracy [8-10]. Figure 5.13 illustrates the limitation of the classical formulation in an unambiguous manner. The reader may recall that this formulation uses the same number of Fourier modes in the expansions for the field variable and for the boundary constraints. Results displayed in Fig. 5.13 demonstrate that there exists a critical corrugation amplitude $S_{critical}$, which can be found for each corrugation wave number α, beyond which the use of additional Fourier modes N_M in the solution does not improve the accuracy (the accuracy is improved below $S_{critical}$). As the geometry becomes more demanding (as S increases), the Fourier expansions describing boundary constraints have smaller rate of convergence

and, potentially, may fail to converge. It has been shown that the use of a larger number of terms in these expansions, i.e., a larger number of boundary relations, improves convergence [11] and thus provides access to a wider range of values of S and α. This leads to the over-determined formulation of the IBC method and the relevant linear system can be solved only in the least squares sense. There are several possible solution strategies based on the fact that the entries into the coefficient matrix can be divided into two classes, i.e., those corresponding to the field equations and those originating in the boundary constraints. In the solution process one could assign different weights to both classes of entries, potentially solving one of them exactly while the other one only in the least square sense. It has been shown in [11] that the selection of the weight factor affects the final accuracy and thus one needs to search for the optimal weight, and this process is computationally expensive.

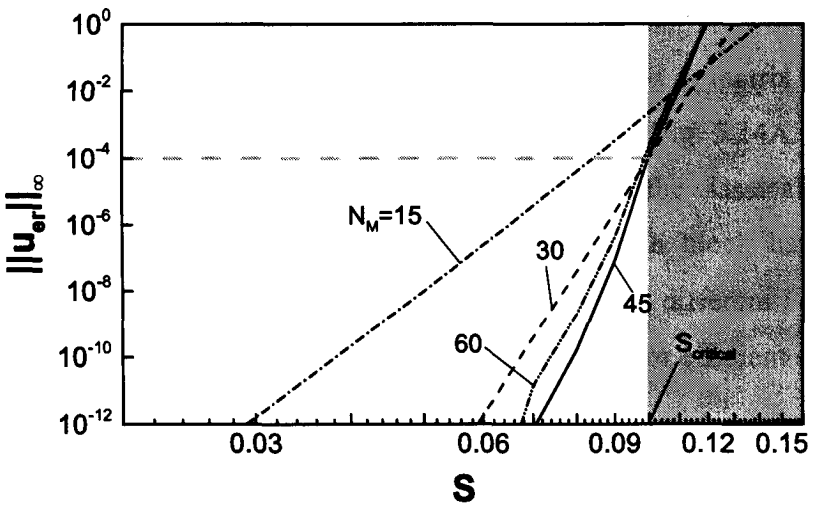


Figure 5.13. *Variations of the $\|u_{er}(x,y)\|_\infty$ norm as a function of the corrugation amplitude S for the classical formulation of the IBC method for different number of Fourier modes N_M used in the computations. Tests were carried out for the channel geometry defined by Eq.(5.3.2) with the corrugation wave number α=5.0 using N_T=100 Chebyshev polynomials. $S_{critical}$ defines value of S above which the use of additional Fourier modes cannot improve the accuracy of the solution. Grey zone identifies area where the IBC method is unable to deliver the required accuracy.*

The construction of the over-determined system of equations is based on the utilization of $4(2M_M+1)$ boundary constraints of type (5.2.17)-(5.2.18) using $|n| = 0,...,M_M$, where $N_U \geq M_M > N_M$ with N_M being the number of Fourier modes used in defining the solution in (5.2.9). In accordance with the classical formulations N_T-3 projection equations of type (5.2.11) for each Fourier mode are retained. Therefore, the system of equations will reduce to the classical formulation for the special case of $N_M=M_M$. The complete over-determined system of algebraic equations has the general form

$$\mathbf{L_o x = Z_o}, \tag{5.4.1}$$

where $\mathbf{L_o}$ is a r×p coefficient matrix with $p = (2N_M+1)(N_T+1)$ and $r = p+8(M_M - N_M)$, $\mathbf{x}$ is a p-dimensional column vector of unknowns and $\mathbf{Z_o}$ is a r-dimensional column vector whose values are known. Similar to the classical formulation, the system is organized by grouping entries corresponding to the projection equations in matrix $\mathbf{H}$ of size q×p, $q = (2N_M+1)(N_T-3)$, and entries corresponding to boundary relations in matrix $\mathbf{K_o}$ of size (r-q)×p. The structure of the coefficient matrix $\mathbf{L_o}$ is illustrated in Fig. 5.14A. For the same physical problem, the matrices $\mathbf{H}$ for the over-determined and the classical formulations are the same and have a block diagonal structure with each block having the size $(N_T+1)\times(N_T-3)$. Matrix $\mathbf{K_o}$ is full and provides coupling among different Fourier modes. System (5.4.1) could be very large and this necessitates search for efficient solvers.

5.4.1. Standard direct solvers

System (5.4.1) can be solved directly using various standard algorithms that evaluate the unknown vector $\mathbf{x} \in C^p$ in the least squares sense by minimizing the 2-norm of the residual vector for the complete system denoted by $\mathbf{r=Z_o-L_o x} \in C^r$. The general way of solving the least squares problem (5.4.1) is to evaluate the pseudo-inverse or the generalized inverse $\mathbf{L_o^+}$ of the over-determined matrix $\mathbf{L_o}$ such that

$$\mathbf{x} = \mathbf{L}_o^{+}\mathbf{Z}_o . \tag{5.4.2}$$

The pseudo inverse $\mathbf{L}_o^{+}$ was calculated in [11] using the method of normal equations, QR factorization and the SVD method. All these methods have been found to provide a similar level of accuracy, especially around and beyond the critical point. In the present work we shall explore two specialized methods that provide more efficient solution of (5.4.2) with details to be presented in Section 5.4.2. We begin with the discussion of the standard methods using MATLAB implementation, i.e.,

i) Over-determined Method A, which relies on QR factorization. This is the most popular method for solving least squares problems [17]. Here, the matrix $\mathbf{L}_o \in \mathbf{C}^{r\times p}$ is decomposed into a product of a unitary matrix $\mathbf{Q} \in \mathbf{C}^{r\times r}$ and an upper-triangular matrix $\mathbf{R} \in \mathbf{C}^{r\times p}$, i.e.,

$$\mathbf{L}_o = \mathbf{QR} = \mathbf{Q}\begin{pmatrix}\mathbf{R}_1\\ \mathbf{0}\end{pmatrix} \tag{5.4.3a}$$

where $\mathbf{R}_1 \in \mathbf{C}^{p\times p}$ is an upper-triangular matrix. The pseudo-inverse $\mathbf{L}_o^{+}$ takes the form

$$\mathbf{L}_o^{+} = (\mathbf{R}_1)^{-1}\mathbf{Q}^{H} \in \mathbf{C}^{p\times r} . \tag{5.4.3b}$$

ii) Over-determined Method B, which solves the least squares problem (5.4.1) using the singular value decomposition (SVD) algorithm. The SVD of a matrix $\mathbf{L}_o \in \mathbf{C}^{r\times p}$ is the most reliable method to evaluate the pseudo-inverse of the matrix $\mathbf{L}_o$ [16,17] although it is the most computationally expensive method. According to the SVD theorem, for any matrix $\mathbf{L}_o \in \mathbf{C}^{r\times p}$ of rank h there exists unitary matrices $\mathbf{U} \in \mathbf{C}^{r\times r}$ and $\mathbf{U} \in \mathbf{C}^{p\times p}$ such that

$$\mathbf{L}_o = \mathbf{U}\Sigma\mathbf{V}^{H}, \quad \Sigma = \begin{pmatrix}\Sigma_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}, \tag{5.4.4a}$$

where $\Sigma \in C^{r\times p}$, $\Sigma_1 = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, ..., \sigma_h)$ and

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq ..... \geq \sigma_h > 0 \qquad (5.4.4b)$$

where σ_i are the singular values of $\mathbf{L_o}$. The pseudo-inverse in this case becomes

$$\mathbf{L_o}^{+} = \mathbf{V}\begin{pmatrix} \Sigma_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{U}^{H}. \qquad (5.4.4c)$$

5.4.2. Specialized Direct Solvers

We are interested in the development of efficient solvers that could significantly reduce the computing time. In order to take advantage of the structure of the matrix, we start with the classical methods F and G described in Section 5.3.4 and generalize them to the over determined formulation.

The largest square matrix **A** of the size q×q is extracted from **H**, which leads to three other sub matrices **B**, $\mathbf{C_o}$ and $\mathbf{D_o}$ (see Fig. 5.14B). The resultant **A** and **B** matrices have block diagonal structures as discussed in Section 5.3.4, while the full rectangular matrices $\mathbf{C_o}$ and $\mathbf{D_o}$ have sizes (r-q)×q and (r-q)×(p-q), respectively. The vector of unknowns **x** is re-organized by re-locating the first four coefficients of Chebyshev expansions for each Fourier mode resulting in the structure of the coefficient matrix shown in Fig. 5.14B. The complete system can be written as

$$\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2 = (\mathbf{Z_o})_1, \qquad \mathbf{C_o}\mathbf{x}_1 + \mathbf{D_o}\mathbf{x}_2 = (\mathbf{Z_o})_2, \qquad (4.5a,b)$$

where vector $\mathbf{x}_1$ contains unknowns $G_k^{(n)}$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 4, N_T \rangle$, and $\mathbf{x}_2$ contains unknowns $G_k^{(n)}$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 0,3 \rangle$. The right hand side vector

remains unchanged with $\mathbf{Z}_o = \begin{bmatrix} (\mathbf{Z}_o)_1 \\ (\mathbf{Z}_o)_2 \end{bmatrix}$, $(\mathbf{Z}_o)_1$ having length q and $(\mathbf{Z}_o)_2$ having length (r-q). Matrices **B** and $\mathbf{D}_o$ contain information associated with $G_0^{(n)}$, $G_1^{(n)}$, $G_2^{(n)}$ and $G_3^{(n)}$, while matrices **A** and $\mathbf{C}_o$ contain the rest. Solution of (5.3.8) can be written as

$$\mathbf{x}_2 = \mathbf{E}_o^{+}(\mathbf{Z}_o)_2 - \mathbf{E}_o^{+}\mathbf{C}_o\mathbf{A}^{-1}(\mathbf{Z}_o)_1, \qquad \mathbf{x}_1 = \mathbf{A}^{-1}(\mathbf{Z}_o)_1 - \mathbf{A}^{-1}\mathbf{B}\mathbf{x}_2, \qquad (5.4.6a,b)$$

where $\mathbf{E}_0^{+} = (\mathbf{D}_o - \mathbf{C}_o\mathbf{A}^{-1}\mathbf{B})$ is the pseudo-inverse of the over-determined matrix $\mathbf{E}_o$ and has the size (p-q)×(r-q).

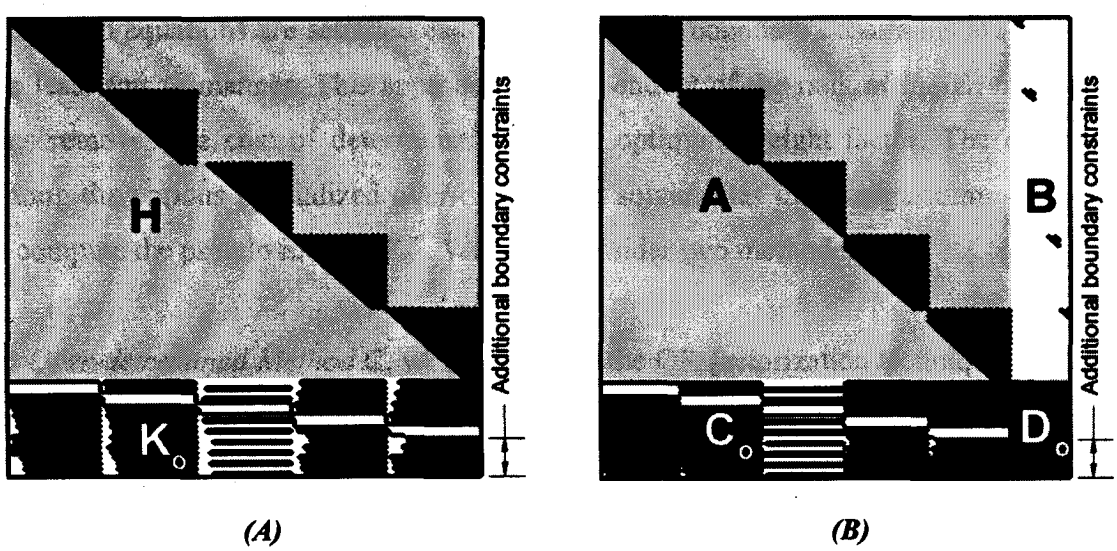


Figure 5.14. *Structure of the coefficient matrix for the A and B (Fig. 5.14A) and C and D (Fig. 5.14B) over-constrained methods.*

The quantities $\mathbf{A}^{-1}(\mathbf{Z}_o)_1$ and $\mathbf{A}^{-1}\mathbf{B}$ can be computed block by block thereby reducing computational time and memory requirements. In practice the quantities $\mathbf{A}^{-1}(\mathbf{Z}_o)_1$ and $\mathbf{A}^{-1}\mathbf{B}$ are computed using the LU decomposition technique in order to avoid any numerical issues associated with the evaluation of the inverse of ill-conditioned matrices.

The use of (5.4.5)-(5.4.6) in solving the over-determined problem (5.4.1) is fundamentally different from the process used in the over-determined methods A and B, which were explored in Ref. 14. Methods A and B minimize the 2-norm of the residual vector $\mathbf{r}=\mathbf{Z}_o-\mathbf{L}_o\mathbf{x} \in C^r$, which eventually minimizes residual attributable to both the projection equations and the boundary constraints. In Ref. 14 a weighted residual method was investigated where one can put more weight on either the projection equations or the boundary constraints in order to obtain the most accurate solution. Determining the optimum weight factor in this approach can be computationally expensive, particularly for problems that result in numerically rank deficient coefficient matrices. For the over-determined methods A and B, all test results to be discussed correspond to the optimum weight factors. On the other hand, in the new approach described by (5.4.5)-(5.4.6), the projection equations are satisfied exactly whereas the boundary constraints are solved in the least-square manner. This approach is independent of the rank of the $\mathbf{E}_o$ matrix and thus removes the cost of determination of the optimum weight factor. The difference among the various specialized over-determined solvers lies in the algorithms employed to compute the pseudo inverse $\mathbf{E}_o^+$. We shall consider two methods, i.e.,

iii) Over-determined Method C, which relies on the QR factorization technique described earlier in (5.4.3a,b) to compute $\mathbf{E}_o^+$ and

iv) Over-determined Method D, which evaluates $\mathbf{E}_o^+$ by applying the SVD technique described using (5.4.4a-c).

Both specialized over-determined solvers C and D take advantage of the complex conjugate properties of the diagonal blocks of **A** in order to achieve the highest efficiency of the computations as well the lowest usage of memory, similarly as in the classical method G. We shall use the same test problems and error norms as described in Section 5.3. The over-determined method A is used as the reference method. The performance gain achieved by other over-determined solvers will be quantified as the ratio of the computational time involved in solving (5.4.1) by the over-determined method A to that

required by the over-determined method in question and is denoted by $\text{Gain}_{\text{Over-determined}}$. Larger values of $\text{Gain}_{\text{Over-determined}}$ correspond to a more efficient method.

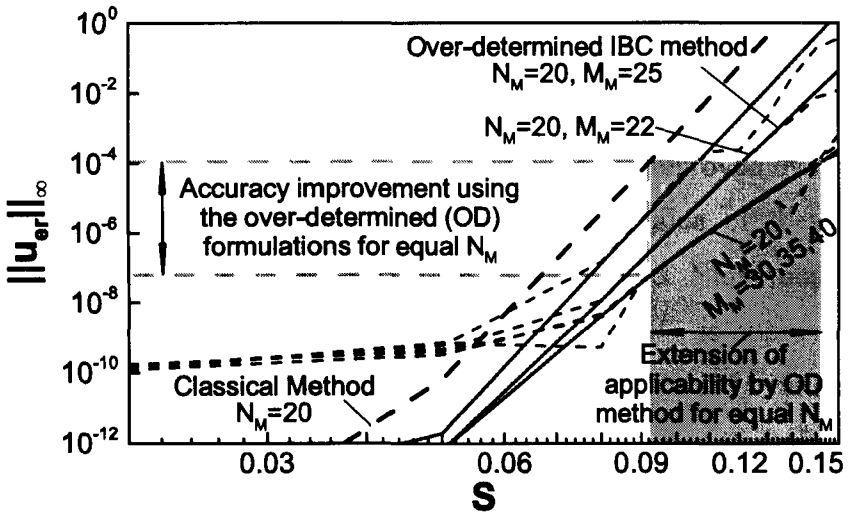


Figure 5.15. *Variations of the $\|u_{er}(x,y)\|_{\infty}$ norm as a function of the corrugation amplitude S for the channel geometry given by Eq. (5.3.2) with the corrugation wave number α=5.0 computed using the over-determined IBC methods A (dotted line), B (dash line), C (dash-dotted line) and D (solid line) employing $4(2M_M+1)$ number of boundary constraints with different values of M_M. Computations were carried out using the same number of Fourier modes to represent the field variable, i.e., N_M=20, and N_T=100 Chebyshev polynomials. The reference solution has been determined using the domain transformation (DT) method with N_M=45 Fourier modes and N_T=125 Chebyshev polynomials. Results obtained using the classical IBC method are shown for comparison purposes. Accuracies of methods A, C and D are similar and the relevant curves coincide. The grey zone illustrates expansion of the range of applicability of the IBC method due to the use of the over-determined formulation.*

Results displayed in Figure 5.15 illustrate variations of the solution accuracy obtained using four different over-determined solvers employed with the same number of Fourier mode N_M=20, but with different numbers of additional boundary constraints $8(M_M-N_M)$, as a function of the corrugation amplitude S. In order to measure the advantage of the over-determined methods, we assume that the maximum acceptable error is 10^{-4} and identify S_{ref} as the maximum corrugation amplitude for which the classical IBC method

is able to deliver this accuracy using the same number of Fourier modes, i.e., with N_M=20. It can be seen that the accuracies of different over-determined solvers are approximately similar around S_{ref}; these accuracies are substantially better than the reference value of 10^{-4}. The range of S for which the over-determined solvers can provide no less than the four digits accuracy is significantly expanded.

Figure 5.15 also shows that the accuracy cannot be arbitrarily improved by increasing the number of additional boundary constraints. The optimal value of these constraints appears to be around M_M~30 for this particular test conditions. Numerous tests carried out as part of this study suggest that the optimal value of M_M is approximately 50% higher than the number of Fourier modes N_M used to represent the field variable.

Figure 5.16 illustrates improvements of the accuracy delivered by the over-determined solvers due to the increase in the number of Fourier modes N_M used to represent the field variable while employing the "optimal" number of additional constraints, i.e., with M_M being approximately 50% higher than N_M. It can be seen that an increase of N_M substantially improves the accuracy as well as expands the range of applicability for the specialized over-determined methods. On the other hand, increasing the number of Fourier modes N_M beyond 30 actually degrades the accuracy of the solution for the standard over-determined methods A and B with method A being more adversely affected than method B.

The complexity of geometries that the IBC method can handle accurately is defined using the critical corrugation amplitude $S_{critical}$ which takes different value for each corrugation wave number α (see Fig. 5.13) and depends on the type of the formulation, i.e., classical or over-determined. For the particular test conditions shown in Fig. 5.16, $S_{critical}\approx 0.1$ for the classical formulation (see Fig. 5.13). The over-determined method D improves accuracy by approximately eight orders of magnitude for the same geometry. If one keeps accuracy fixed at 10^{-4}, the over-determined method D extends the applicability of the IBC method by increasing $S_{critical}$ by more than 70%. Comparisons of accuracies obtained with different over-determined solvers shows that the over-determined method

A produces lowest accuracy while the method D provides the highest accuracy, and for the specialized methods C and D, an increase of the number of Fourier modes used in the computations by 10 produces accuracy improvement of at least two-orders of magnitude.

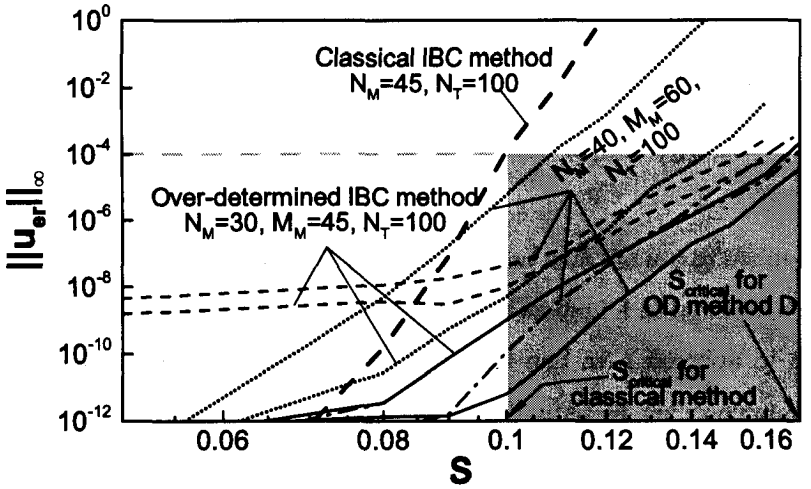


Figure 5.16. *Variations of the* $\|u_{er}(x,y)\|_{\infty}$ *norm as a function of the corrugation amplitude S computed using the over-determined IBC methods A (dotted line), B (dash line), C (dash-dotted line) and D (solid line) employing different number of Fourier modes* N_M *and* $4(M_M+1)$ *boundary constraints. Other conditions as in Fig. 5.15. The grey zone illustrates the largest possible expansion of the range of applicability of the IBC method due the use of the over-determined formulation (e.g., this range cannot be expanded any further through an increase in the number of Fourier modes* N_M *used in the computations).*

Variations of the computational efficiency of different over-determined solvers, as compared to the over-determined method A, as a function of the number of Fourier modes and Chebyshev polynomials used in the computations are illustrated in Fig. 5.17. The over-determined method B is the slowest; methods C and D outperform method A by a factor of 100 or more for higher values of N_M and N_T. The over-determined method D is slightly slower than method C while delivering substantial accuracy improvement for higher values of N_M (see Fig. 5.16) and this leaves the user with a clear choice as far as the best solver for the application at hand.

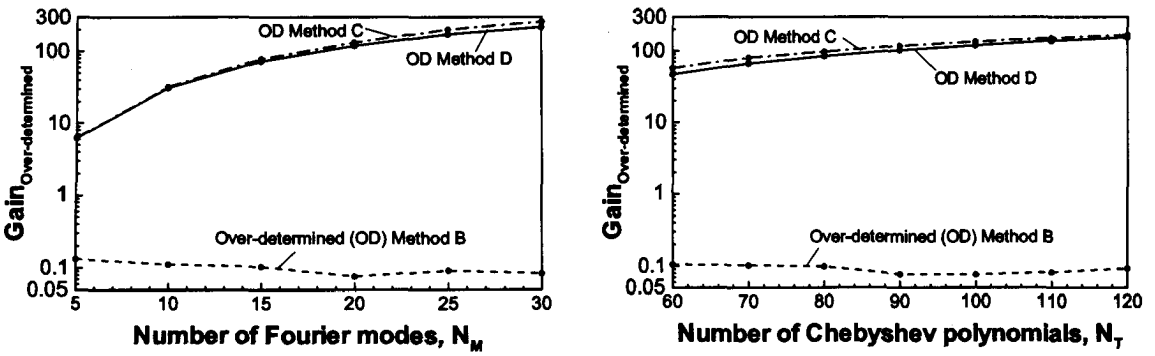


Figure 5.17. *Variations of the performance gains associated with the use of the over-determined methods B, C and D as compared with the over-determined method A as a function of the number of Fourier modes N_M (Fig. 5.17A, N_T=100, $M_M \approx 1.5 * N_M$) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.17B, N_M=20, M_M=30) used in the computations with the channel shapes defined by Eq.(5.3.2) with α=5.0 and S=0.13.*

Figure 5.18 illustrates variations of the performance of different over-determined methods as a function of the number of additional boundary constraints used in the solution while keeping the same number of Fourier modes and Chebyshev polynomials. It can be seen that the relative performance of methods C and D slightly decreases as expected. A useful comparison between the classical method A and different over-determined methods is presented in Figure 5.19. The performance gains by the over-determined methods C and D increase substantially as the number and size of the blocks in the **A** matrix increase, however, these gains are not as impressive as in the case of the classical methods F and G (see Fig. 5.9). This is mainly due to the fact that the cost of computing $\mathbf{E_0}^+$ is significantly higher than the cost of computing the equivalent of $\mathbf{E}^{-1}$ and there is an additional computing cost associated with the construction of the additional boundary constraints. Nevertheless, the over-determined method C is about 25 times faster than the classical method A for N_M=30 and N_T=100 while the over-determined method D is about 21 times faster. It should be stressed that for the higher values of the corrugation wave number α ($\geq$5), the classical methods would require a larger number of Fourier modes N_M while the over-determined methods can work with a smaller number of such modes and with just a few more boundary relations in order to

deliver the same accuracy, and this would further improve the relative performance of the over-determined methods C and D.

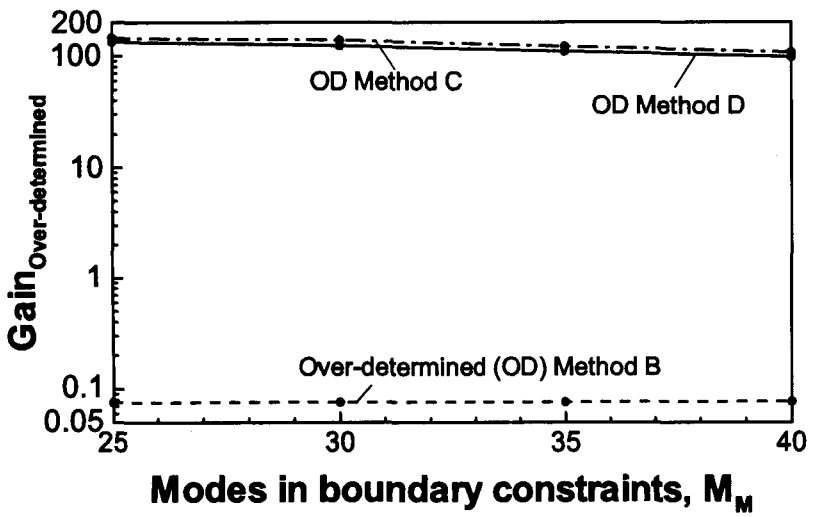


Figure 5.18. *Variations of the performance gains associated with the use of the over-determined methods B, C and D as compared with the over-determined method A as a function of the number of boundary constraints $4(M_M+1)$ used in the computation. $N_M=20$ and $N_T=100$ were used in the computations. Other conditions as in Fig. 5.17.*

Figure 5.20 shows that the error in the enforcement of the physical boundary conditions $\|u_{er,BC}\|_\infty$ for the over-determined method D is equal to the maximum error over the whole solution domain $\|u_{er}\|_\infty$ provided that a sufficient number of Chebyshev polynomials have been used in the discretization. This observation is very important as it allows the use of the same strict measure of error as introduced in the case of the classical IBC method, e.g., Eq. (5.3.7).

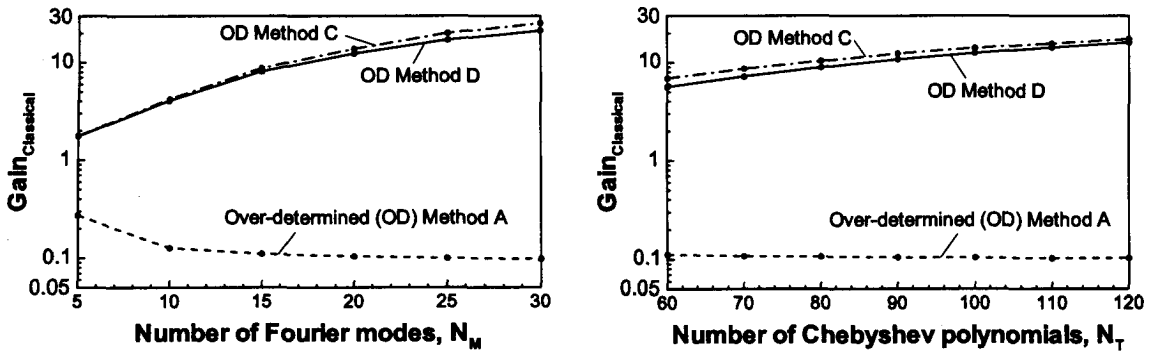


Figure 5.19. *Variations of the performance gains associated with the use of the over-determined methods B, C and D as compared with the classical method A as a function of the number of Fourier modes N_M (Fig. 5.19A, N_T=100, M_M≈1.5*N_M) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.19B, N_M=20, M_M=30) used in the computations. All tests have been carried out for the channel geometry defined by Eq.(5.3.2) with α=3.0 and S=0.1.*

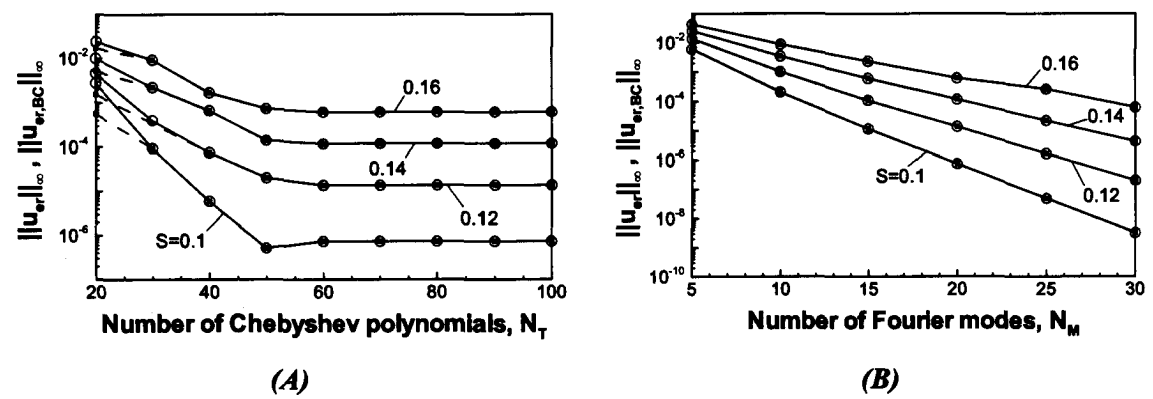


Figure 5.20. *Variations of the norms $\|u_{er}\|_\infty$ (dash line) and $\|u_{er,BC}\|_\infty$ (solid line) evaluated using the over-determined method D as a function of the number of Fourier modes N_M (Fig. 5.20A, N_T=100, M_M≈1.5*N_M) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.20B, N_M=20, M_M=30) used in the computations. The reference solution has been determined using the domain transformation (DT) method with N_M=45 Fourier modes and N_T=125 Chebyshev polynomials. All tests have been carried out for the channel geometry defined by Eq.(5.3.2) with α=3.0 for a few selected values of S. For certain parameter ranges the dash lines may not be visible as they overlap with the solid lines.*

Figure 5.21 illustrates variations of the critical amplitude $S_{critical}$ of the corrugation (determined using the classical formulation) as a function of the corrugation wave number α. $S_{critical}$ decreases from ~0.5 for α=1 to ~0.05 for α=10. The same figure illustrates the best possible accuracies that can be produced by the classical and the over-determined formulations for the corrugation amplitudes $S=S_{critical}$. The over-determined method D improves accuracy by ten orders of magnitude for α=1 and this gain drops to six orders of magnitude for α=10.

Figure 5.22 compares values of $S_{critical}$ for the classical and over-determined formulations. It can be seen that $S_{critical}$ increases by approximately 50%-70% depending on the value of the corrugations wave number α when replacing the classical formulation with the over-determined formulation. The same figure illustrates the best accuracy that can be obtained using the over-determined formulation under conditions corresponding to $S_{critical}$ for this method.

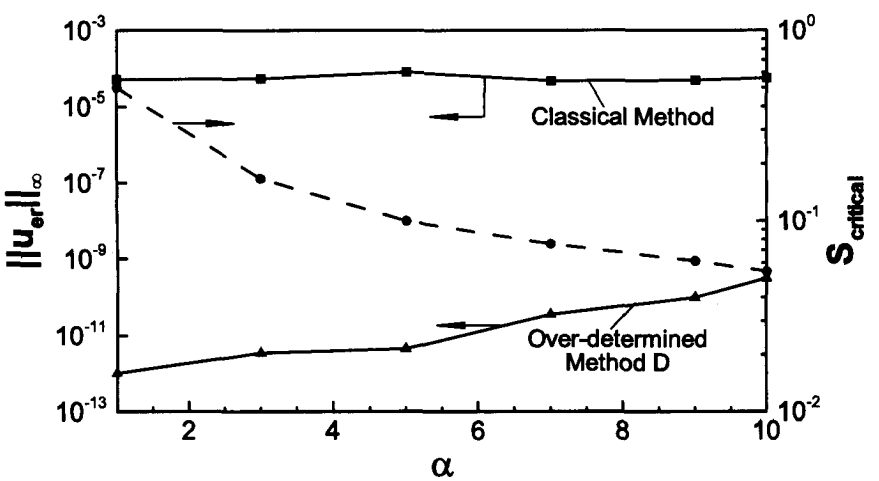


Figure 5.21. *Variations of the critical corrugation amplitude $S_{critical}$ determined using the classical formulation (dash line) as well as the best accuracy produced by the classical formulation for the corrugation amplitude $S=S_{critical}$ (upper solid line) as a function of the corrugation wave number α. Accuracy produced by the over-determined formulation (method D, lower solid line) for the same geometries is shown for reference. All tests have been carried out for the channel geometry defined by Eq.(5.3.2).*

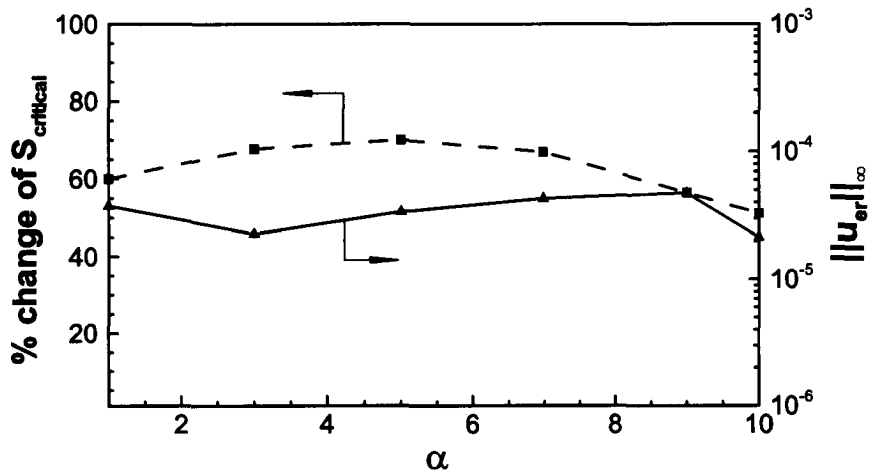


Figure 5.22. *Variations of the ratio of critical amplitudes $S_{critical}$ computed on the basis of the over-determined (method D) and classical formulations as a function of the corrugation wave number α for the channel geometries defined by Eq. (5.3.2) (dash line). The solid line illustrates the best accuracy obtained by the over-determined IBC method D for the critical values of the corrugation amplitude $S_{critical}$ corresponding to this method.*

Similarly as in the case of the classical methods F and G, the over-determined solvers C and D require significantly less memory. Both over-determined methods C and D require similar memory as the only difference between them is the use of different schemes to compute the pseudo inverse. Figure 5.23 illustrates rapid reduction in the memory use compared with the over-determined methods A or B, e.g., for N_M=30 Fourier modes and N_T=100 Chebyshev polynomials used in the computations the C and D methods require about 25 times less memory than the A and B methods.

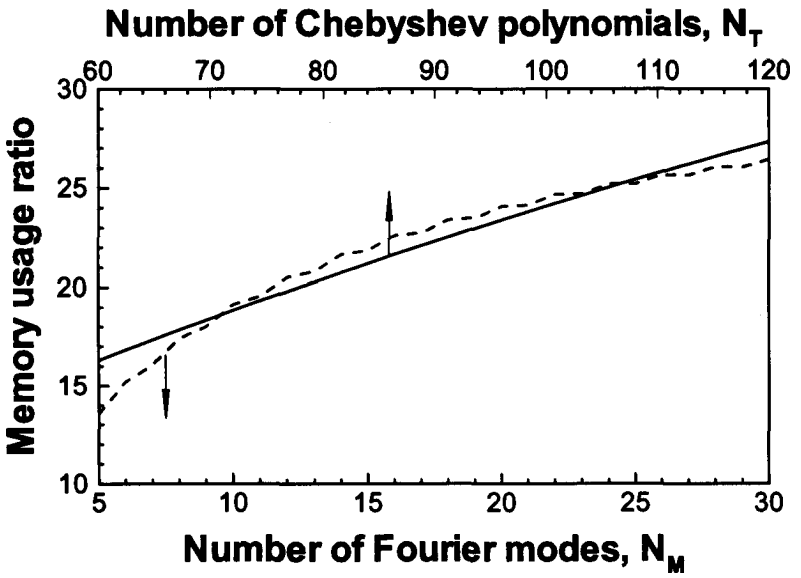


Figure 5.23. *Variations of the ratio of memory required by the over-determined methods A and B and the over-determined methods C and D as a function of the number of Fourier modes N_M and the number of Chebyshev polynomials N_T used in the computations for the channel geometries defined by Eq. (5.3.2). Methods A and B, and C and D require similar memory.*

5.5. Unsteady problems

The specialized direct solvers discussed in Sections 5.3.4 and 5.4.2 can be extended to solve the linear systems resulting form the discretization of the unsteady flow problems, including cases with time-dependent boundary conditions. The general form of the algebraic system that needs to be solved at each time step can be expressed as

$$\mathbf{L}\,\mathbf{x}(t) = \mathbf{Z}(t), \tag{5.5.1}$$

where $\mathbf{L}$ is a r×p coefficient matrix with $p = (2N_M+1)(N_T+1)$ and $r = p+8(M_M - N_M)$, $\mathbf{x}$ is a time-dependent p-dimensional column vector of unknowns and $\mathbf{Z}$ is a known time-dependent r-dimensional column vector. For the over-determined IBC formulation, M_M

is greater than N_M leading to r>p, while the assumption of M_M=N_M leads to the classical formulation for which r=p. The elements in **Z** for the current time-step are functions of both the solution vectors **x**(t) evaluated at previous time-step(s) as well as the physical boundary conditions [8-10]. The basic structure of the coefficient matrix **L** would be similar to that shown in Fig. 5.2 for the classical formulation and would take the form shown in Fig. 5.14A for the over-determined formulation. In both cases the complete system can be divided into two sub-systems of the form

$$\mathbf{Hx}(t) = \mathbf{Z}_1(t)\,, \qquad \mathbf{Kx}(t) = \mathbf{Z}_2(t), \qquad (5.5.2a,b)$$

where the block diagonal matrix **H** of size q×p, $q = (2N_M+1)(N_T-3)$, groups entries corresponding to the projection equations and the full matrix **K** of size (r-q)×p contains entries corresponding to the boundary constraints. The right-hand-side vector $\mathbf{Z}_1$ is always a function of time and depends on the type of temporal discretization while the right-hand-side vector $\mathbf{Z}_2$ remains unchanged unless the physical boundary conditions depend on time, e.g., time-dependent wall transpirations.

5.5.1. Standard direct solvers

Since the coefficient matrix **L** in Eq.(5.5.1) remains unchanged during the time stepping, one needs to compute either the inverse or the LU factors of **L** only once in the case of the classical formulation and $\mathbf{L}^+$ in the case of the over-determined formulation. The following discussion of the performance of various standard unsteady solvers will be based on the classical formulation while performance of solvers suitable for the over-determined formulation can be deduced by comparing data to be presented in this section with those discussed in Sections 5.3 and 5.4. We have explored three different standard direct solvers:

i. *Unsteady classical method A,* which computes the LU factors of **L** once with the solution process for the subsequent time steps reduced to a combination of backward substitutions and forward eliminations,

ii. *Unsteady classical method B*, which also relies on the LU decomposition but applied to **L** after it has been converted into a sparse matrix (similarly as in the case of the classical method D),

iii. *Unsteady classical method C*, which computes the inverse of **L** once with the solution process for the subsequent time steps reduced to matrix multiplications.

5.5.2. Specialized direct solvers

Specialized solvers are developed following the same ideas as described in Section 5.3.4. The first four coefficients of the Chebyshev expansions for each Fourier mode are relocated resulting in the extraction of the largest square block diagonal matrix $\mathbf{A} \in \mathbf{C}^{q \times q}$ from the complete matrix **L**. The re-arranged system for flow problems with time-dependent boundary conditions can be written as

$$\mathbf{A}\mathbf{x}_1(t)+\mathbf{B}\mathbf{x}_2(t)=\mathbf{Z}_1(t), \qquad \mathbf{C}\mathbf{x}_1(t)+\mathbf{D}\mathbf{x}_2(t)=\mathbf{Z}_2(t), \qquad (5.5.3a,b)$$

where vector $\mathbf{x}_1(t)$ contains unknowns $G_k^{(n)}(t)$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 4, N_T \rangle$, and vector $\mathbf{x}_2(t)$ contains unknowns $G_k^{(n)}(t)$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 0,3 \rangle$. Matrices $\mathbf{A} \in \mathbf{C}^{q \times q}$ and $\mathbf{C} \in \mathbf{C}^{(r-q) \times q}$ contain information associated with $\mathbf{x}_1(t)$, while matrices $\mathbf{B} \in \mathbf{C}^{q \times (p-q)}$ and $\mathbf{D} \in \mathbf{C}^{(r-q) \times (p-q)}$ contain information associated with $\mathbf{x}_2(t)$. Solution of (5.5.3) can be written as

$$\mathbf{x}_2(t) = \mathbf{E}^{+}\mathbf{Z}_2(t) - \mathbf{E}^{+}\mathbf{C}\mathbf{A}^{-1}\mathbf{Z}_1(t), \qquad \mathbf{x}_1(t) = \mathbf{A}^{-1}\mathbf{Z}_1(t) - \mathbf{A}^{-1}\mathbf{B}\mathbf{x}_2(t), \qquad (5.5.4a,b)$$

where $\mathbf{E}^{+} = (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ is the pseudo-inverse of the over-determined matrix **E** and has the size (p-q)×(r-q). In the case of the classical formulation, r is equal to p and, therefore,

$\mathbf{E}^{+}$ in Eq. (5.5.4a) should be replaced by $\mathbf{E}^{-1}$. We shall limit further discussion to the classical formulation only.

The solution process starts with the evaluation of $\mathbf{A}^{-1}$, $\mathbf{A}^{-1}\mathbf{B}$, $\mathbf{E}^{-1}$ and $\mathbf{E}^{-1}\mathbf{C}$, or their equivalents, all of which are computed block-by-block taking advantage of the block diagonal structure of **A** and **B**, and then stored in the memory for repetitive use at each of the subsequent time steps. The quantity $\mathbf{A}^{-1}\mathbf{Z}_1(t)$ needs to be computed at each time step and this is done in the block-by-block manner multiplying each block of $\mathbf{A}^{-1}$ with the corresponding elements of $\mathbf{Z}_1(t)$. Two versions of the specialized classical solvers have been tested:

iv. *Unsteady classical method D*, which uses matrix inversion technique to compute $\mathbf{A}^{-1}$ and $\mathbf{E}^{-1}$,

v. *Unsteady classical method E*, which computes the LU factors of **A** and **E** in order to compute $\mathbf{A}^{-1}\mathbf{B}$, $\mathbf{E}^{-1}\mathbf{C}$, $\mathbf{A}^{-1}\mathbf{Z}_1(t)$ and $\mathbf{E}^{-1}\mathbf{Z}_2(t)$.

5.5.3. Comparison of the performance of different solvers

The right-hand-side vector $\mathbf{Z}_2$ remains constant for problems with constant boundary conditions while it needs to be re-computed at each time step in the case of the time-dependent boundary conditions. This difference affects the way quantity $\mathbf{E}^{-1}\mathbf{Z}_2(t)$ needs to be evaluated; in the latter case $\mathbf{E}^{-1}\mathbf{Z}_2(t)$ needs to be computed at each time step while in the former case $\mathbf{E}^{-1}\mathbf{Z}_2$ can be computed only once and stored for subsequent repetitive use. Because of the small size of **E**, the difference in the computational time associated with the change between the constant and the time-dependent boundary conditions is insignificant and thus all tests reported in this section have been carried out with the time-independent boundary conditions using the same reference geometry as used in Sections 5.3 and 5.4, i.e., geometry described by Eq.(5.3.2).

The total computational cost of the solution depends on the number of time steps and on the size of the problem dictated by the spatial discretization. The unsteady classical method A is used as the reference method in the following discussion. The total performance gain achieved by any other unsteady solver is quantified as the ratio of the total computational time involved in solving (5.5.1) by the unsteady method A to that required by the method in question and is denoted by $Gain_{Unsteady}$. Larger values of $Gain_{Unsteady}$ indicate advantage for the method in question over method A. Performances of different unsteady solvers for two different numbers of time steps are illustrated in Figure 5.24. It can be seen that the efficiency of all methods, except method C, improves with increasing the number of Fourier modes N_M or the number of Chebyshev polynomials N_T. Method E was found to be the fastest while methods B and C were found to be rather slow for problems involving larger values of N_M and N_T.

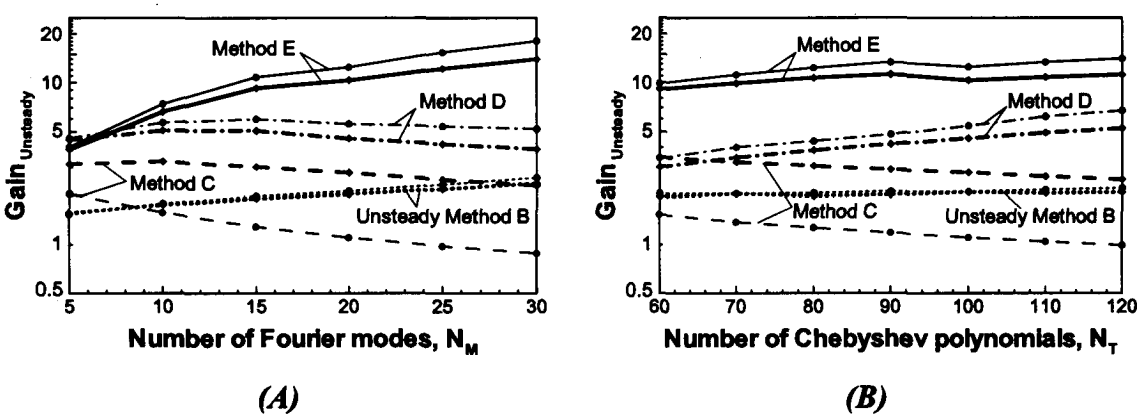


Figure 5.24. *Variations of the total performance gains $Gain_{Unsteady}$ resulting from the use of the classical unsteady methods B (short dash lines), C (long dash lines), D (Dash-dot lines) and E (Solid lines) as a function of the number of Fourier modes N_M (Fig. 5.24A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.24B, N_M=20) used in the computations. The tests have been carried out using the model problem (5.3.2) with α=3.0 and S=0.05. Two-step implicit method was used for the temporal discretization. The thinner and thicker lines correspond to simulation carried out with 100 and 500 time steps, respectively.*

Figure 5.24 shows a surprising change in the performance of the methods, depending on the number of time steps used in the simulations. The increase in the number of steps

decreases the performance of all methods, with exception of method C, which displays fairly large improvement in the performance. The reader may note that the complete solution process consists of two segments. The first one involves the construction of either the complete coefficient matrix or its relevant parts and evaluation of either the inverse or the LU factors, which are stored for the subsequent use. The computational cost of this segment shall be referred to as the *fixed cost* as it does not depend on the number of time steps used in the simulations. The second segment consists of the repetitive use at every time step of information generated during the first segment. The cost per single time step is referred to as the *variable cost*. The total computational cost consists of the sum of the fixed cost and the variable cost multiplied by the number of time steps. The relative performance of different methods in all these areas can be measured using additional performance measures defined as follows:

i) $\text{Ratio}_{\text{Performance}}$ is defined as the ratio of the fixed and the variable costs and provides an "absolute" measure of the efficiency of the method of interest. Higher value of $\text{Ratio}_{\text{Performance}}$ corresponds to higher cost of construction and pre-processing of the coefficient matrix and usually indicates disadvantage for this method,

ii) $\text{Gain}_{\text{Fixed}}$ is defined as the ratio of the fixed cost of the unsteady method A to the fixed cost of another method and provides a measure of gains in the fixed component of the computational cost of the method in question (larger values of $\text{Gain}_{\text{Fixed}}$ indicate advantage of the method in question),

iii) $\text{Gain}_{\text{Variable}}$ is defined as the ratio of the variable cost of the unsteady method A to the variable cost of another method and provides a measure of gains in the variable component of the computational cost of the method in question (larger values of $\text{Gain}_{\text{Variable}}$ indicate advantage of the method in question).

The total gain, as measured by $Gain_{Unsteady}$, is a function of $Gain_{Fixed}$ and $Gain_{Variable}$ as well as the number of time steps involved in the simulation. $Gain_{Unsteady}$ is dominated by $Gain_{Fixed}$ if either $Gain_{Fixed}$ is very large or a small number of time steps is used in the simulation; $Gain_{Unsteady}$ is dominated by $Gain_{Variable}$ when either $Gain_{Fixed}$ is small or when a large number of time steps is used in the simulations.

Figure 5.25 shows that the unsteady method C has values of the $Ratio_{Performance}$ substantially higher than the remaining methods, while values of $Ratio_{Performance}$ for the methods A and B drop by about two orders of magnitude and values for the methods D and E drop by additional two-orders of magnitude. Method D has the lowest value of $Ratio_{Performance}$, i.e., the smallest fixed cost when compared against the variable cost.

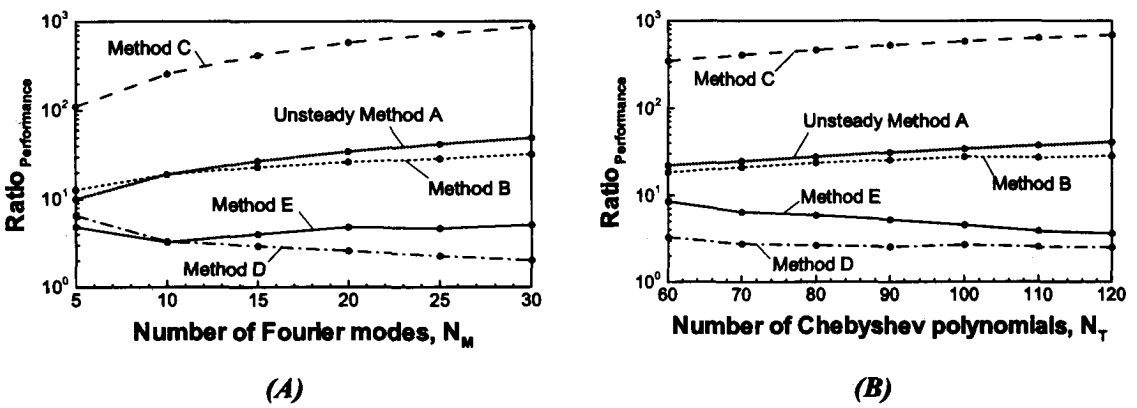


Figure 5.25. *Variations of the $Ratio_{Performance}$ associated with the use of the classical unsteady methods A (Dot line), B (short dash line), C (long dash line), D (Dash-dot line) and E (Solid line) as a function of the number of Fourier modes N_M (Fig. 5.25A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.25B, N_M=20) used in the computations. All other conditions as in Figure 5.24.*

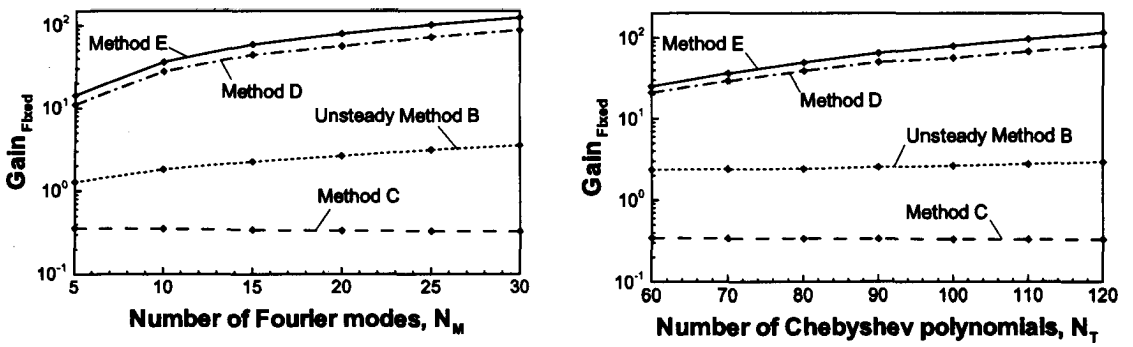


Figure 5.26. *Variations of the performance gains $Gain_{Fixed}$ associated with the construction and pre-processing of the coefficient matrix using the classical unsteady methods B (short dash line), C (long dash line), D (Dash-dot line) and E (Solid line) as a function of the number of Fourier modes N_M (Fig. 5.26A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.26B, N_M=20) used in the computations. All other conditions as in Figure 5.24.*

Comparison of the fixed cost among the various methods is illustrated in Fig. 5.26. It can be seen that $Gain_{Fixed}$ is the lowest for method C and the highest for method E, indicating that method C is the most expensive while E is the least expensive. $Gain_{Fixed}$ varies by almost three orders of magnitude between different methods. Comparison of the variable cost is illustrated in Fig. 5.27. Method B is the most expensive while method E is the least expensive. The relatively low variable cost for method C suggests that the disadvantage of this method due to the high fixed cost decreases as the number of time steps increases, thereby improving the over-all performance as quantified by $Gain_{Unsteady}$ (see Fig. 5.24).

Analysis of data displayed in Figures 5.24-5.27 shows that method E is the most efficient, as quantified by $Gain_{Unsteady}$ (see Fig. 5.24), in spite of having values of $Ratio_{Performance}$ a little bigger than those for the method D (see Fig. 5.25). The best efficiency of this method results from the lowest fixed cost and the lowest variable cost (see Figs. 5.26 and 5.27). Memory requirements provide another limitation for applicability of each of the methods. The unsteady methods E and D require storage for

only certain segments of the complete matrix **L** and this leads to the reduction in memory use similar to those found in the case of the classical steady method G (see Fig. 5.12).

The above discussion shows that method E represents the best unsteady solver from the point of view of execution time as well as memory requirements. Performance of this solver should be closed to the performance of any other solver used for DNS as the problems associated with the geometry-based coupling between the various Fourier modes have been efficiently dealt with through the development of the specialized solution techniques.

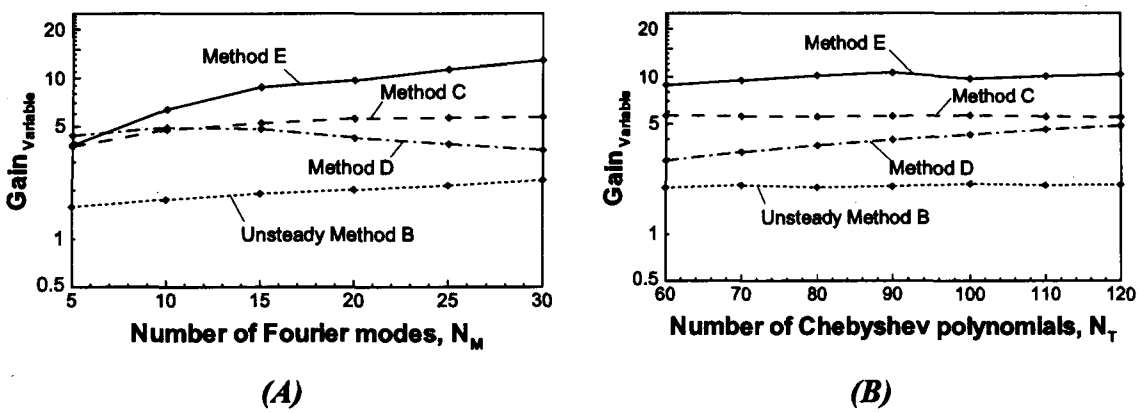


Figure 5.27. *Variations of the performance gains $Gain_{Variable}$ associated with advancing solution one time step forward using the classical unsteady methods B (short dash line), C (long dash line), D (Dash-dot line) and E (Solid line) as a function of the number of Fourier modes N_M (Fig. 5.27A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 5.27B, N_M=20) used in the computations. All other conditions as in Figure 5.24.*

5.6. Summary

Efficient linear solvers for problems arising from the spectral implementation of the immersed boundary conditions (IBC) method have been proposed and tested. The

classical as well as the over-determined formulations of the IBC method have been considered. The testing and demonstration of the efficiency of these solvers have been carried out in the context of a model problem consisting of Stokes flow in a corrugated channel. The specialized solvers take advantage of the structure of the coefficient matrix and provide acceleration of computations by up to two orders of magnitude and a similar reduction in the memory requirements when compared with the performance of the standard solvers. These solvers are also about an order of magnitude faster than the iterative solvers and their applicability is not constrained by the any convergence limitations. Comparison of the performance of the specialized solvers with the performance of the solvers based on the construction of the boundary conforming coordinates (which were represented by the Domain Transformation method in the present work) demonstrate approximately two-orders of magnitude of better performance of the IBC method.

The specialized solvers applied to the over-determined formulation of the IBC algorithm significantly improve the accuracy of the enforcement of the flow boundary conditions while extending the applicability of the IBC algorithm to more severe geometries (5.50%-70% increase of the corrugation amplitude, depending on the corrugation wave number) with a very reasonable increase of the computational costs.

Analysis of performance of the specialized solvers applied to unsteady problems has been presented in the context of the unsteady Stokes flow with constant as well as time-dependent boundary conditions. These solvers were found to improve the computational efficiency and reduce the memory use by up to two orders of magnitude when compared with the standard solvers and thus bring the performance of the IBC algorithm very close to the performance of the standard DNS solvers used for analysis of flows in smooth channels.

5.7. References

[1] Peskin, C.S., The fluid dynamics of heart valves: experimental, theoretical and computational methods, *Annu. Rev. Fluid Mech.,* vol. 14 pp. 235-59 (1982).

[2] Mittal, R. and Iaccarino, G., Immersed boundary methods, *Annu. Rev. Fluid Mech.,* vol. 37 pp. 239-261 (2005).

[3] Peskin, C.S., The immersed boundary method, *Acta Numerica*, 479-517 (2002).

[4] Szumbarski, J. and Floryan, J.M., A direct spectral method for determination of flows over corrugated boundaries, *J. Comp. Phys.*, vol. 153, pp. 378-402 (1999).

[5] Floryan, J.M., Centrifugal instability of couette flow over a wavy wall, *Phys. Fluids*, vol. 14, pp. 312-322 (2002).

[6] J.M. Floryan, Vortex instability in a converging-diverging channel, *J. Fluid Mech.*, vol. 482, pp. 17-50 (2003).

[7] Floryan, J.M., Three-dimensional instabilities of laminar flow in a rough channel and the concept of hydraulically smooth wall, *Eur. J. Mech. B/Fluids*, vol. 26, pp. 305-329 (2007).

[8] Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for unsteady flow problems described by the Laplace operator, *Int. J. Num. Meth. Fluids*, vol. 56, pp. 1765-1786 (2007).

[9] S.Z. Husain, J.M. Floryan, Implicit spectrally-accurate method for moving boundary problems using immersed boundary conditions concept, *J. Comp. Phys.*, vol. vol. 227, pp. 4459-4477 (2008).

[10] S.Z. Husain, J.M. Floryan, Gridless spectral algorithm for stokes flow with moving boundaries, *Comput. Methods Appl. Mech. Engrg.,* vol. 198, pp. 245-259 (2008).

[11] Husain, S.Z., Floryan, J.M. and Szumbarski, J., Over-determined formulation of the immersed boundary conditions method, *Comput. Methods Appl. Mech. Engrg.* doi: 10.1016/j.cma.2009.09.022 (2009).

[12] Spalart, P.R., Moser, R.D. and Rogers, M.M., Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions, *J. Comp. Phys.*,vol. 96, pp. 297-324 (1991).

[13] Jiang, Y. and Floryan, J.M., Finite-difference 4^{th}-order compact scheme for the direct numerical simulation of instabilities of shear layers, *Int. J. Num. Meth. Fluids*, vol. 48, pp. 1259-1281 (2005).

[14] Gilbert, J. R., Moler, C. and Schreiber, R., Sparse matrices in MATLAB: Design and implementation, *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 333-356 (1992).

[15] Zhang, F., *The Schur complement and its applications*, *Springer* (2005).

[16] Trefethen, L.N., Bau, D., Numerical linear algebra, *SIAM* (1997).

[17] Björck, Å, Numerical methods for least squares problems, *SIAM* (1996).

[18] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A. Spectral methods in fluid dynamics, *Springer* (1987).

CHAPTER 6

Moving boundary problems described by the Navier-Stokes equations[1]

6.1. Introduction

'Moving boundary problems' refer to situations where the boundaries of the flow domain change locations as a function of time in a known and well prescribed manner. Such problems attracted attention of researchers from various fields for many years and the relevant algorithms have been pursued with considerable interest particularly in the field of biomedical engineering. Practical examples include peristaltic and pulsatile flows that define the flows in the esophagus and flows through the vasculatures due to cardiac actions. Various available algorithms can be classified mainly as Lagrangian and Eulerian [1]. Mixed methods, that are combinations of the Lagrangian and Eulerian techniques, have also been pursued [1].

In algorithms based on the Lagrangian concepts, each fluid element is followed individually resulting in a need for a coordinate system that moves with the fluid. Mesh tangling and associated loss of numerical accuracy poses significant restrictions on the overall applicability of these methods [1].

The Eulerian algorithms rely on coordinate systems that are stationary in a laboratory frame of reference or may move in a prescribed manner. Such algorithms can be divided

---

[1]A version of this chapter has been submitted for publication –

Husain, S.Z. and Floryan, J.M., Spectrally-accurate algorithm for moving boundary problems for the Navier-Stokes equations (2009).

for convenience into fixed grid, adaptive grid and various mapping methods.

In the fixed grid methods, the grid is fixed in the solution domain and the locations of the moving boundaries are tracked using either surface [2] or volume tracking procedures [3]. The surface tracking relies on a set of points whose motion is tracked during the solution process allowing precise identification of the boundary locations; these boundaries are represented as a set of interpolated curves [3, 4]. The volume tracking algorithms on the other hand work by reconstructing the boundary whenever necessary instead of storing the boundary locations. The presence of a convenient marker within a computational cell and its quantity form the basis of the various reconstruction methodologies. Different versions of volume tracking algorithms exist, e.g., VOF (Volume of Fluid) [5], MAC (Marker and Cell) [6] and Level Set [7, 8] methods. These methods are based on the standard spatial discretization schemes with low order of accuracy for the field equations, which are consistent with the diffused boundary locations resulting from the boundary reconstruction processes.

The adaptive grid methods use numerical mappings to adjust the grids at each time step so that one of the grid lines always overlaps with the boundary location. The computational costs of these methods are very high due to the requirement of grid reconstruction at each time step. For example, the grid construction process contributed to about 75% of the total computational cost for the problem discussed in [9]. The choice of spatial discretization technique has a smaller effect on the overall computational costs. The requirement of high accuracy in solution may lead to numerous challenges as the total error has contributions from the error in the grid generation as well as from the error due to spatial and temporal discretizations of the field equations.

Analytical mapping of the irregular physical domain into a rectangular computational domain can help in improving the accuracy at the cost of increased complexity of the field equations [10, 18]. However, such mappings are available only for a limited class of geometries [1] and reconstruction of the coefficient matrix during each time step can add to the overall computational cost by a substantial margin [19].

Mixed Lagrangian-Eulerian methods rely on the combination of the concepts described above [1].

The increase of accuracy while maintaining computational efficiency poses a significant challenge for any algorithm to be developed for moving boundary problems. One of the new concepts involves the use of the immersed or fictitious boundaries. This concept was first proposed by Peskin [11] in the context of cardiac dynamics and its various variants have been reviewed in [12, 13]. The common limitation is the spatial accuracy, as most of these methods are based on the low-order finite-difference, finite-volume or finite-element technique [13-16]. The second, less known limitation is associated with the use of the local fictitious forces required to enforce the no-slip and no-penetration conditions. These forces locally affect the flow physics and this may lead to the incorrect estimates of derivatives of flow quantities, i.e., misrepresentation of the local wall shear. It is difficult to estimate the potential error associated with such procedures, but it is known that in the case of hydrodynamic instabilities the second derivative of mean flow plays a very strong role in determining the flow response, and this derivative may not be predicted with sufficient accuracy using a procedure that changes flow physics (even locally). This problem is likely to be more pronounced in the case of methods with higher spatial accuracy. Sharp interface method as presented in [17] also uses Cartesian grid to enforce boundary conditions along the immersed boundaries that may be either stationary or moving. The method relies on the level set description of the immersed interface along with finite-difference discretization and special treatment for points close to the interface resulting in a sharp resolution of the interface. This method requires modest programming effort and retains low spatial accuracy, similar to other immersed boundary methods discussed earlier.

Spectral methods provide the lowest error for spatial discretization of the field equations but are generally limited to solution domains with regular geometries. The first spectrally accurate implementation of the immersed boundary concept was developed by Szumbarski and Floryan in the context of a fixed boundary problem [18] and is referred

to as the Immersed Boundary Conditions (IBC) method in the rest of this discussion. This method does not use any fictitious boundaries or fictitious forces but relies on a purely formal construction of boundary constraints in order to generate the required closing relations. The method is analogous to the fixed grid Eulerian methods. Spatial discretization relies on the spectral expansions based on Fourier and Chebyshev expansions in the stream-wise and normal-to-the-wall direction respectively and thus provides ability to reach machine level accuracy. The construction of boundary constraints relies on the representation of the physical boundaries in the spectral space and nullifying the relevant Fourier modes. Such implementation is limited to geometries that can be represented by Fourier expansions but results in a gridless algorithm as all possible variations of boundary geometries are described in terms of the Fourier coefficients only. The programming effort associated with modeling the changes of geometry has been essentially removed as the only information required for specifying the new geometry is reduced to a set of Fourier coefficients provided as an input to the code. The additional attractiveness of this concept is associated with the precise mathematical formalism, high accuracy and sharp identification of the location of time-dependent physical boundaries. The method has been implemented to study problems involving hydrodynamic instabilities induced by surface roughness [19, 20] and has been successfully extended to unsteady problems [21] as well as moving boundary problems involving Laplace and biharmonic operators [22, 23]. Extension of the IBC algorithm to three-dimensional problems is fairly simple. The flow-field needs to be assumed to be periodic in two spatial dimensions, which are discretized using Fourier expansions, while the remaining aperiodic dimension is discretized using Chebyshev expansions. The boundary geometries/motions can be modeled using Fourier expansions in the periodic dimensions and the enforcement of the boundary conditions would follow the concept of the IBC algorithm as presented in the case of two-dimensional problems.

The work presented in this chapter has two goals. The first one deals with the extension of the IBC algorithm to moving boundary problems described by the two-dimensional Navier-Stokes equations. The second one deals with the question of efficient implementation of the algorithm. The chapter is organized as follows. Section 6.2

provides description of the model problem used for presentation of the algorithm. Section 6.3 describes numerical implementation of the flow boundary conditions. Section 6.4 discusses performance of the algorithm using carefully selected test problems. In particular, Section 6.4.1 describes peristaltic flow problem while Section 6.4.2 considers pulsatile flow problem. Section 6.4.3 provides a brief discussion of issues associated with different temporal discretization schemes. Section 6.4.4 discusses issues related to the efficient implementation of the algorithm. Section 6.5 provides a short summary of the main conclusions.

6.2. Problem formulation

The problem to be investigated consists of an incompressible unsteady viscous flow in a conduit bounded by moving walls (see Fig. 6.1) whose shapes and motions are described by the following relations

$$y_L(x,t) = -1 + \sum_{n=-\infty}^{n=+\infty} H_L^{(n)}(t)\, e^{in\alpha x}, \qquad y_U(x,t) = 1 + \sum_{n=-\infty}^{n=+\infty} H_U^{(n)}(t) e^{in\alpha x}. \qquad (6.2.1a,b)$$

where L and U stand for lower and upper walls respectively. Here $H_L^{(n)} = H_L^{(-n)^*}$ and $H_U^{(n)} = H_U^{(-n)^*}$ are known and stars denote complex conjugates. At any instant of time, the conduit extends to $\pm\infty$ in the x-direction and its geometry remain periodic with the wavelength $\lambda = 2\pi/\alpha$.

The reference flow, i.e., steady flow through a straight conduit, is the Poiseuille flow with the velocity and pressure fields in the form

$$u_0(x,y) \equiv u_0(y) = 1 - y^2, \quad v_0(x,y) = 0, \quad p_0(x,y) \equiv p_0(x) = -2x/Re \qquad (6.2.1)$$

where the motion of the fluid is in the positive x-direction, the Reynolds number Re is based on the half conduit height and the maximum velocity in the x-direction, and the flow is driven by a constant negative pressure gradient. The pressure and time scales have the form $\rho\upsilon U_{max}/L$ and L^2/υ, respectively, where υ stands for the kinematic viscosity and ρ denotes the density of the fluid. Introduction of the wall motions induces flow modifications and thus the total flow quantities can be expressed as

$$u(x,y,t) = u_0(y) + u_1(x,y,t), \quad v(x,y,t) = v_1(x,y,t), \quad p(x,y,t) = p_0(x) + p_1(x,y,t) \quad (6.2.3)$$

where u, v and p denote the total velocities and pressure, and u_1, v_1 and p_1 denote velocity and pressure modifications induced by the boundary motions.

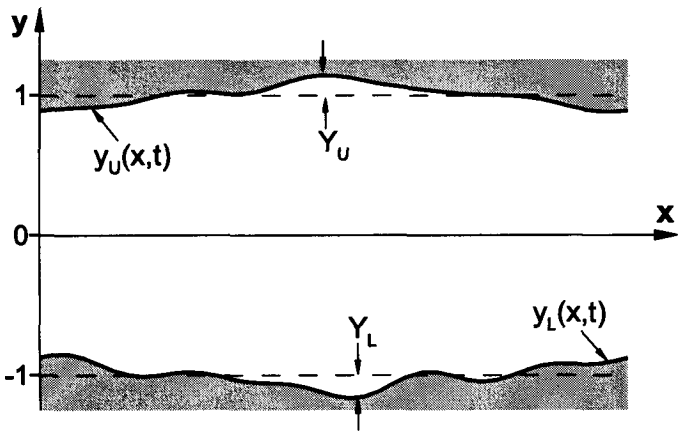


Figure 6.1. *Sketch of the instantaneous form of the flow domain.*

Substitution of (6.2.3) into the Navier-Stokes and continuity equations result in the following form of governing equations

$$Re^{-1}\partial_t u_1 + u_0\partial_x u_1 + u_1\partial_x u_1 + v_1 Du_0 + v_1\partial_y u_1 = -\partial_x p_1 + Re^{-1}\nabla^2 u_1, \quad (6.2.4a)$$

$$Re^{-1}\partial_t v_1 + u_0\partial_x v_1 + u_1\partial_x v_1 + v_1\partial_y v_1 = -\partial_y p_1 + Re^{-1}\nabla^2 v_1, \quad (6.2.4b)$$

$$\partial_x u_1 + \partial_y v_1 = 0, \tag{6.2.4c}$$

where D ≡ d/dy, $\nabla^2 = \partial_{xx} + \partial_{yy}$ is the Laplacian and the symbol ∂ denotes partial differentiation with subscripts x, y and t denoting the argument of the differentiation. The flow problem can be posed either for the complete flow quantities (u,v,p) or for the flow modifications (u_1,v_1,p_1). The latter approach was selected for the present work as it results in numerically smaller nonlinear terms thereby providing faster convergence when iterative solution processes are employed.

The problem formulation needs to be supplemented with suitable initial and boundary conditions. The initial conditions are taken to be in the form

$$u(x,y,0) = u_i(x,y),\ v(x,y,0) = v_i(x,y),\ y_L(x,0)=y_{Li}(x),\ y_U(x,0)=y_{Ui}(x), \tag{6.2.5a-d}$$

where $u_i(x,y)$, $v_i(x,y)$, $y_{Li}(x)$, $y_{Ui}(x)$ are considered to be known, and the boundary conditions at the solid walls are given by

$$u_0(y_L(x,t)) + u_1(x, y_L(x,t), t) = u_L(x,t) = 0, \tag{6.2.6a}$$

$$u_0(y_U(x,t)) + u_1(x, y_U(x,t), t) = u_U(x,t) = 0, \tag{6.2.6b}$$

$$v_1(x, y_L(x,t), t) = v_L(x,t) = (y_L)' = \sum_{n=-\infty}^{n=+\infty} (H_L^{(n)})' e^{in\alpha x}\ , \tag{6.2.6c}$$

$$v_1(x, y_U(x,t), t) = v_U(x,t) = (y_U)' = \sum_{n=-\infty}^{n=+\infty} (H_U^{(n)})' e^{in\alpha x}\ , \tag{6.2.6d}$$

where the ' ' sign denotes derivative with respect to time.

Introduction of the stream function Ψ and elimination of pressure lead to a single field equation in terms of the unknown modification of stream function Ψ_1 of the form

$$\begin{aligned} &Re^{-1}\partial_t[\nabla^2\Psi_1]-Re^{-1}\nabla^4\Psi_1+[u_0\partial_x(\nabla^2)-D^2u_0\partial_x]\Psi_1= \\ &\quad -\partial_y[\partial_x(u_1u_1)+\partial_y(u_1v_1)]+\partial_x[\partial_x(u_1v_1)+\partial_y(v_1v_1)]. \end{aligned} \tag{6.2.7}$$

where $\nabla^2=\partial_{xx}+\partial_{yy}$ is the Laplacian and

$$u(x,y,t)=u_0(y)+u_1(x,y,t)=D\Psi_0+\partial_y\Psi_1=\partial\Psi_T/\partial y, \tag{6.2.8a}$$

$$v(x,y,t)=v_1(x,y,t)=-\partial_x\Psi_1=-\partial_x\Psi_T, \tag{6.2.8b}$$

where $\Psi_0 = -y^3/3+y+2/3$ denotes the stream function of this flow and Ψ_T stand for the stream function of the complete flow (i.e., the total stream function). The boundary conditions in terms of the stream function thus take the following form

$$\partial_y\Psi_1(x,y_L(x,t),t)=-D\Psi_0(y_L(x,t)), \tag{6.2.9a}$$

$$\partial_y\Psi_1(x,y_U(x,t),t)=-D\Psi_0(y_U(x,t)), \tag{6.2.9b}$$

$$\partial_x\Psi_1(x,y_L(x,t),t)=-\sum_{n=-\infty}^{n=+\infty}(H_L^{(n)})'e^{in\alpha x}, \tag{6.2.9c}$$

$$\partial_x\Psi_1(x,y_U(x,t),t)=-\sum_{n=-\infty}^{n=+\infty}(H_U^{(n)})'e^{in\alpha x}. \tag{6.2.9d}$$

We are interested in the determination of solution of the flow problem described by Eqs (6.2.7)-(6.2.9) with the spectral accuracy in space and the desired accuracy in time. The main difficulty associated with the implementation of the spectral spatial discretization arises due to the irregularity and time-dependence of the solution domain.

In order to overcome problems associated the with spatial discretization, we select fixed rectangular computational domain extending over one period in the x-direction and extending sufficiently far in the y-direction so that the flow domain always remains immersed inside the computational domain during the time interval of interest. If we denote the locations of extremities of the walls as Y_U and Y_L within the time interval under investigation, then the y-extent of the computational domain is set as $(-1-Y_L, 1+Y_U)$ without loss of generality. The spatial discretization is based on the use of Fourier series in the x-direction due to periodicity of the geometry, and on expansions in terms of the Chebyshev polynomials in the y-direction. We shall use standard definition of the Chebyshev polynomials and thus the y-extent of the computational domain needs to be mapped onto (-1,1) (see Fig. 6.1) before calculations can proceed. The required mapping has the form

$$\hat{y} = [y - (1 + Y_U)]\,\Gamma + 1 \tag{6.2.10}$$

where $\hat{y} \in \langle -1,1 \rangle$ and $\Gamma = 2/(2+Y_U+Y_L)$ is a constant. Application of (6.2.10) transforms the governing equation into

$$\begin{aligned} &Re^{-1}\partial_t[\hat{\nabla}^2\Psi_1] - Re^{-1}\hat{\nabla}^4\Psi_1 + [u_0\partial_x(\hat{\nabla}^2) - \Gamma^2\hat{D}^2u_0\partial_x]\Psi_1 \\ &\quad = -\Gamma\partial_{\hat{y}}[\partial_x(u_1u_1) + \partial_{\hat{y}}(u_1v_1)] + \partial_x[\partial_x(u_1v_1) + \Gamma\partial_{\hat{y}}(v_1v_1)] \end{aligned}, \tag{6.2.11}$$

where $\hat{\nabla}^2 = \partial_{xx} + \Gamma^2\partial_{\hat{y}\hat{y}}$ and $\hat{D} = d/d\hat{y}$. Locations of the walls in the $(x,\hat{y})$ plane are given as

$$\hat{y}_L(x,t) = \sum_{n=-\infty}^{n=+\infty} A_L^{(n)}(t)e^{in\alpha x}, \qquad \hat{y}_U(x,t) = \sum_{n=-\infty}^{n=+\infty} A_U^{(n)}(t)e^{in\alpha x}, \tag{6.2.12a,b}$$

where $A_L^{(0)}(t) = 1 + \Gamma[-2 - Y_U + H_L^{(0)}(t)]$, $A_L^{(n)}(t) = \Gamma H_L^{(n)}(t)$ for $n \neq 0$, $A_U^{(0)}(t) = 1 + \Gamma[-Y_U + H_U^{(0)}(t)]$, $A_U^{(n)}(t) = \Gamma H_U^{(n)}(t)$ for $n \neq 0$.

The boundary conditions at the transformed boundaries become

$$\partial_{\hat{y}}\Psi_1(x,\hat{y}_L(x,t),t) = -\hat{D}\Psi_0(\hat{y}_L(x,t)), \tag{6.2.13a}$$

$$\partial_{\hat{y}}\Psi_1(x,\hat{y}_U(x,t),t) = -\hat{D}\Psi_0(\hat{y}_U(x,t)), \tag{6.2.13b}$$

$$\Psi_1(x,\hat{y}_L(x,t),t) = -\sum_{n=-\infty,n\neq 0}^{n=+\infty}(in\alpha\Gamma)^{-1}(A_L^{(n)})'e^{in\alpha x} + C_L(t), \tag{6.2.13c}$$

$$\Psi_1(x,\hat{y}_U(x,t),t) = -\sum_{n=-\infty,n\neq 0}^{n=+\infty}(in\alpha\Gamma)^{-1}(A_U^{(n)})'e^{in\alpha x} + C_U(t), \tag{6.2.13d}$$

where $C_U(t)$ and $C_L(t)$ are time-dependent constants resulting from integration of Eq. (6.2.9c) and (6.2.9d), respectively.

The solution can be represented in the form of Fourier expansion

$$\Psi_1(x,\hat{y},t) = \sum_{n=-\infty}^{n=+\infty}\Phi^{(n)}(\hat{y},t)e^{in\alpha x} \approx \sum_{n=-N_M}^{n=+N_M}\Phi^{(n)}(\hat{y},t)e^{in\alpha x}, \tag{6.2.14}$$

where $\Phi^{(n)}(\hat{y},t) = \Phi^{(-n)^*}(\hat{y},t)$. Substitution of (6.2.14) into Eq. (6.2.11) and separation of Fourier components lead to a system of partial differential equations for $\Phi^{(n)}$, $n \in \langle 0, N_M \rangle$, of the type

$$\begin{aligned} &Re^{-1}\partial_t\left[\Gamma^2\hat{D}^2 - (n\alpha)^2\right]\Phi^{(n)} - Re^{-1}\left[\Gamma^4\hat{D}^4 - 2\Gamma^2(n\alpha)^2\hat{D}^2 + (n\alpha)^4\right]\Phi^{(n)} + \\ &\left[in\alpha\Gamma^2u_0\hat{D}^2 - i(n\alpha)^3u_0 - in\alpha\Gamma^2\hat{D}^2u_0\right]\Phi^{(n)} = \\ &-in\alpha\Gamma\hat{D}(R_1)^{(n)} - \Gamma^2\hat{D}^2(R_2)^{(n)} - (n\alpha)^2(R_2)^{(n)} + in\alpha\Gamma\hat{D}(R_3)^{(n)} \end{aligned}, \tag{6.2.15}$$

where the nonlinear velocity products are periodic and thus are expressed in terms of the relevant Fourier expansion in the form

$$[(u_1u_1),(u_1v_1),(v_1v_1)](x,\hat{y},t)=\sum_{n=-N_M}^{n=+N_M}\left[(R_1)^{(n)},(R_2)^{(n)},(R_3)^{(n)}\right](\hat{y},t)e^{in\alpha x}. \tag{6.2.16}$$

Four types of temporal discretizations have been used. The third-order implicit method results in the following relations

$$\begin{aligned}
&\left[-Re^{-1}\Gamma^4\hat{D}^4+\left(11\Gamma^2Re^{-1}\Delta t^{-1}/6+2n^2\alpha^2\Gamma^2Re^{-1}+in\alpha\Gamma^2u_0\right)\hat{D}^2+\right.\\
&\qquad\left.\left(-n^4\alpha^4Re^{-1}-11n^2\alpha^2\Gamma^2\Delta t^{-1}Re^{-1}/6-in\alpha\Gamma^2\hat{D}^2u_0-in^3\alpha^3u_0\right)\right]\left(\Phi_{\tau+1}^{(n)}\right)^{J+1}=\\
&\qquad\left[-in\alpha\Gamma\hat{D}(R_1)_{\tau+1}^{(n)}-\Gamma^2\hat{D}^2(R_2)_{\tau+1}^{(n)}-(n\alpha)^2(R_2)_{\tau+1}^{(n)}+in\alpha\Gamma\hat{D}(R_3)_{\tau+1}^{(n)}\right]^J\\
&\qquad+3Re^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau}^{(n)}-(3/2)Re^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau-1}^{(n)}\\
&\qquad+(1/3)Re^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau-2}^{(n)},\qquad\qquad n\in\langle -N_M,N_M\rangle,
\end{aligned}\tag{6.2.17}$$

while similar relations resulting from the first-, second- and fourth-order implicit methods are shown in the Appendix I. In the above, the subscript τ denotes the time step and Δt stands for the (constant) length of the time step. The solution is obtained in an iterative manner during each time step with the superscript J denoting the iteration number.

Relation (6.2.17) has the form of an inhomogeneous ordinary differential equation for $\Phi_{\tau+1}^{(n)}$. The following discussion will be carried out in the context of the third-order implicit scheme, while the relevant relations for the other schemes can be readily deduced. A detail discussion of issues associated with the performance and numerical implementation of different temporal schemes is given in Section 6.4.3.

The unknown function $\Phi_{\tau+1}^{(n)}$ can be represented in terms of expansions based on the Chebyshev polynomials in the form

$$\Phi_{\tau+1}^{(n)}(\hat{y})=\sum_{k=0}^{k=\infty}Z_{k,\tau+1}^{(n)}T_k(\hat{y})\approx\sum_{k=0}^{k=N_T}Z_{k,\tau+1}^{(n)}T_k(\hat{y}) \tag{6.2.18}$$

where T_k denotes the Chebyshev polynomial of k^{th} order and $Z^{(n)}_{k,\tau+1}$ denotes the unknown coefficients of the expansion.

The modal functions associated with the Fourier expansions of the nonlinear terms can also be expressed in terms of expansions based on the Chebyshev polynomials in the form

$$\left[(R_1)^{(n)}_{\tau+1},(R_2)^{(n)}_{\tau+1},(R_3)^{(n)}_{\tau+1}\right](\hat{y}) = \sum_{k=0}^{k=N_T}\left[(\tilde{R}_1)^{(n)}_{k,\tau+1},(\tilde{R}_2)^{(n)}_{k,\tau+1},(\tilde{R}_3)^{(n)}_{k,\tau+1}\right]T_k(\hat{y}). \tag{6.2.19}$$

Relations (6.2.18)-(6.2.19) are substituted into (6.2.17) and Galerkin procedure [24] is used to develop a set of algebraic equations for the unknown coefficients $Z^{(n)}_{k,\tau+1}$ in the form

$$\begin{aligned}
&\sum_{k=0}^{k=N_T}\begin{bmatrix} -\mathrm{Re}^{-1}\Gamma^4\langle T_j,\hat{D}^4T_k\rangle+\left(11\Gamma^2\mathrm{Re}^{-1}\Delta t^{-1}/6+2n^2\alpha^2\Gamma^2\mathrm{Re}^{-1}\right)\langle T_j,\hat{D}^2T_k\rangle+ \\ in\alpha\Gamma^2\langle T_j,u_0\hat{D}^2T_k\rangle+\left(-n^4\alpha^4\mathrm{Re}^{-1}-11n^2\alpha^2\Gamma^2\Delta t^{-1}\mathrm{Re}^{-1}/6\right)\langle T_j,T_k\rangle \\ -in\alpha\Gamma^2\langle T_j,\hat{D}^2u_0T_k\rangle-in^3\alpha^3\langle T_j,u_0T_k\rangle \end{bmatrix}\left(Z^{(n)}_{k,\tau+1}\right)^{J+1} = \\
&\quad\sum_{k=0}^{k=N_T}\begin{bmatrix} -in\alpha\Gamma\langle T_j,\hat{D}T_k\rangle(\tilde{R}_1)^{(n)}_{k,\tau+1}-\Gamma^2\langle T_j,\hat{D}^2T_k\rangle(\tilde{R}_2)^{(n)}_{k,\tau+1}- \\ (n\alpha)^2\langle T_j,T_k\rangle(\tilde{R}_2)^{(n)}_{k,\tau+1}+in\alpha\Gamma\langle T_j,\hat{D}T_k\rangle(\tilde{R}_3)^{(n)}_{k,\tau+1} \end{bmatrix}^J \\
&\quad+\sum_{k=0}^{k=N_T}(\mathrm{Re}\Delta t)^{-1}\left[\Gamma^2\langle T_j,\hat{D}^2T_k\rangle-(n\alpha)^2\langle T_j,T_k\rangle\right] \\
&\qquad\left[3Z^{(n)}_{k,\tau}-(3/2)Z^{(n)}_{k,\tau-1}+(1/3)Z^{(n)}_{k,\tau-2}\right],\quad n\in\langle -N_M,N_M\rangle,\quad j\in\langle 0,N_T\rangle,
\end{aligned} \tag{6.2.20}$$

where the inner product is defined as $\langle f_j(\hat{y}),g_k(\hat{y})\rangle=\int_{-1}^{1}f_j(\hat{y})g_k(\hat{y})\hat{\omega}(\hat{y})d\hat{y}$ and $\hat{\omega}=1/\sqrt{1-\hat{y}^2}$ denotes the weight function [24]. The inner products are evaluated using the orthogonality properties of the polynomials. Equation (2.20) leads to N_T+1 algebraic equations for each Fourier mode; these equations are coupled through the nonlinear terms. Since the Chebyshev polynomials do not individually satisfy the boundary

conditions, special construction is required to ensure that the global solution (2.14) satisfies the boundary conditions. These conditions will be accommodated in the Tau-like manner [24] where the four highest projection equations for each Fourier mode are dropped in order to make space for the relations that ensure that the discretized boundary conditions are satisfied. The numerical treatment and discretization of boundary conditions are discussed in the next Section and show additional source of coupling among different Fourier modes.

The evaluation of the nonlinear modal functions $R_1^{(n)}$, $R_2^{(n)}$ and $R_3^{(n)}$ involves determination of velocity components u_l and v_l in the physical space, evaluation of velocity products in the physical space, evaluation of the Fourier transforms of the velocity products and, finally, evaluation of the coefficients of Chebyshev expansions representing modal functions of the velocity products. Aliasing errors are controlled through the use of the padding method [24]. The padding method requires evaluation of the velocity components u_l and v_l in $(2M_M+1)$ locations in the x-direction for each y-location where $M_M \geq 1.5N_M$. In order to take advantage of the properties of the FFT algorithm, it is recommended to select values of M_M equal to a power of 2. The FFT algorithm generates $(2M_M+1)$ modes but only modes $<-N_M,N_M>$ are retained. The cost of evaluation of the non-linear terms is one of the dominant factors that affect the total computational cost per iteration. The iterations continue until the change in the magnitude of the Chebyshev coefficients for two consecutive iterations is less than the convergence criteria which, for all results presented in this paper, were set to 10^{-14}.

6.3. Numerical Treatment of Boundary Conditions

The flow boundary conditions are imposed using the Immersed Boundary Conditions (IBC) concept. Transformation (6.2.10) ensures that the boundary extremities at any given time are contained within the computational domain $\hat{y} \in \langle -1,1 \rangle$. The flow boundary conditions are to be enforced along the lines $\hat{y}_L(x,\tau+1)$ and $\hat{y}_U(x,\tau+1)$ that define the

physical boundaries in the transformed coordinates $(x,\hat{y})$, where the locations of these lines are represented in the form

$$\hat{y}_L(x,\tau+1)\approx\sum_{n=-N_A}^{n=+N_A}A_{L,\tau+1}^{(n)}e^{in\alpha x},\qquad \hat{y}_U(x,\tau+1)\approx\sum_{n=-N_A}^{n=+N_A}A_{U,\tau+1}^{(n)}e^{in\alpha x}. \qquad (6.3.1a,b)$$

The following discussion will be carried out in the context of the upper wall with the developments for the lower wall being analogous. At time τ+1 one needs to evaluate $\left(\partial_{\hat{y}}\Psi_1\right)_U(x,\tau+1)\equiv\partial_{\hat{y}}\Psi_1(x,\hat{y}_U(x,\tau+1),\tau+1)$ and $\left(\Psi_1\right)_U(x,\tau+1)\equiv\Psi_1(x,\hat{y}_U(x,\tau+1),\tau+1)$ along the time dependent line $\hat{y}_U(x,\tau+1)$.

The terms $\left(\partial_{\hat{y}}\Psi_1\right)_U(x,\tau+1)$ and $\left(\Psi_1\right)_U(x,\tau+1)$ are periodic in x with the period $\lambda = 2\pi/\alpha$ and thus can be expressed in terms of Fourier series as

$$\left(\partial_{\hat{y}}\Psi_1\right)_U(x,\tau+1)\equiv\partial_{\hat{y}}\Psi_1(x,\hat{y}_U(x,\tau+1),\tau+1)=\sum_{n=-N_U}^{n=+N_U}U_{\tau+1}^{(n)}e^{in\alpha x},$$

$$\left(\Psi_1\right)_U(x,\tau+1)\equiv\Psi_1(x,\hat{y}_U(x,\tau+1),\tau+1)=\sum_{n=-N_U}^{n=+N_U}V_{\tau+1}^{(n)}e^{in\alpha x}. \qquad (6.3.2a,b)$$

The follow up discussion shows that the summation extends to $N_U=N_TN_A+N_M$. Since the flow representation used in the computations is limited to N_M+1 modes (see Eq. 2.14), only the first (N_M+1) terms in (6.3.2) can be accounted for. The components in (6.3.2) can also be evaluated along the wall at time τ+1 using the discretized form of the solution, i.e.,

$$\left(\partial_{\hat{y}}\Psi_1\right)_U(x,\tau+1)\equiv\partial_{\hat{y}}\Psi_1(x,\hat{y}_U(x,\tau+1),\tau+1)$$
$$=\sum_{n=-N_M}^{n=+N_M}\sum_{k=0}^{k=N_T}Z_{k,\tau+1}^{(n)}\hat{D}T_k(\hat{y}_U(x,\tau+1))e^{in\alpha x}, \qquad (6.3.3a)$$

$$\left(\Psi_1\right)_U(x,\tau+1) \equiv \Psi_1(x,\hat{y}_U(x,\tau+1),\tau+1)$$
$$= \sum_{n=-N_M}^{n=+N_M} \sum_{k=0}^{k=N_T} Z_{k,\tau+1}^{(n)} T_k(\hat{y}_U(x,\tau+1)) e^{in\alpha x} . \qquad (6.3.3b)$$

Chebyshev polynomials and their derivatives evaluated at the wall, i.e., $T_k(\hat{y}_U(x,\tau+1)$ and $\hat{D}T_k(\hat{y}_U(x,\tau+1)$, are periodic functions of x and thus can be expressed in terms of Fourier expansion as follows

$$T_k\left(\hat{y}_U(x,\tau+1)\right) = \sum_{m=-N_S}^{m=+N_S} \left(w_U\right)_{k,\tau+1}^{(m)} e^{im\alpha x}, \quad \hat{D}T_k\left(\hat{y}_U(x,\tau+1)\right) = \sum_{m=-N_S}^{m=+N_S} \left(d_U\right)_{k,\tau+1}^{(m)} e^{im\alpha x} \qquad (6.3.4a,b)$$

where $\max(N_S)=N_TN_A$. The method for evaluation of coefficients $\left(w_U\right)_{k,\tau+1}^{(m)}$ and $\left(d_U\right)_{k,\tau+1}^{(m)}$ is explained in the context of Stokes flow in Appendix A. Substitution of (6.3.4) into (6.3.3) gives

$$\left(\partial_{\hat{y}}\Psi_1\right)_U(x,\tau+1) = \sum_{n=-N_U}^{n=+N_U} \sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_{k,\tau+1}^{(m)} \left(d_U\right)_{k,\tau+1}^{(n-m)} e^{in\alpha x} , \qquad (6.3.5a)$$

$$\left(\Psi_1\right)_U(x,\tau+1) = \sum_{n=-N_U}^{n=+N_U} \sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_{k,\tau+1}^{(m)} \left(w_U\right)_{k,\tau+1}^{(n-m)} e^{in\alpha x} . \qquad (6.3.5b)$$

Comparison of (6.3.2) with (6.3.5) gives

$$U_{\tau+1}^{(n)} = \sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_{k,\tau+1}^{(m)} \left(d_U\right)_{k,\tau+1}^{(n-m)} , \qquad V_{\tau+1}^{(n)} = \sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_{k,\tau+1}^{(m)} \left(w_U\right)_{k,\tau+1}^{(n-m)} . \qquad (6.3.6a,b)$$

The function $\left(\hat{D}\Psi_0\right)_U(x,\tau+1)$ is known from the reference flow and can be expressed as

$$\left(\hat{D}\Psi_0\right)_U(x,\tau+1)=\sum_{n=-\infty}^{n=+\infty}F_U^{(n)}(\tau+1)e^{in\alpha x}. \tag{6.3.7}$$

Substitution of (6.3.2)-(6.3.7) into (6.2.13) leads to boundary conditions along the lower and upper walls in the form

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}Z_{k,\tau+1}^{(m)}\left(d_{L,U}\right)_{k,\tau+1}^{(n-m)}=-F_{L,U}^{(n)}(\tau+1),\qquad |n|\geq 0 \tag{6.3.8a}$$

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}Z_{k,\tau+1}^{(m)}\left(w_{L,U}\right)_{k,\tau+1}^{(n-m)}=-(in\alpha\Gamma)^{-1}(A_{L,U}^{(n)})'_{\tau+1},\qquad |n|\geq 1 \tag{6.3.8b}$$

$$\sum_{m=-N_M}^{m=+N_M}\sum_{k=0}^{k=N_T}Z_{k,\tau+1}^{(m)}\left(w_{L,U}\right)_{k,\tau+1}^{(-m)}=C_{L,U}(\tau+1), \tag{6.3.8c}$$

that are enforced for $|n|\leq N_m$. Relations obtained for $|n|>N_m$ can be used as a measure of error. The form of constants $C_{L,U}(\tau+1)$ that appear in Eq. (6.3.8c) can be determined using either the constant volume flux or the constant pressure gradient constraint. For convenience, we have selected the former one for this presentation, i.e., we assumed that the flow rates in the smooth reference conduit and in the deforming conduit remained the same. Volume flux Q along the conduit can be evaluated by integrating the x-velocity component across the conduit, i.e.,

$$Q(x,\tau+1)=\int_{\hat{y}_L}^{\hat{y}_U}\left(\frac{\partial\Psi_T}{\partial\hat{y}}\right)_{\tau+1}d\hat{y}=\Psi_T(x,\hat{y}_U(x,\tau+1),\tau+1)-\Psi_T(x,\hat{y}_L(x,\tau+1),\tau+1). \tag{6.3.9}$$

where according to the definition of the total stream function

$$\Psi_T(x,\hat{y}_{L,U}(x,\tau+1),\tau+1)=\Psi_0(x,\hat{y}_{L,U}(x,\tau+1))+\Psi_1(x,\hat{y}_{L,U}(x,\tau+1),\tau+1). \tag{6.3.10}$$

Values of Ψ_0 evaluated along the lower wall represent a known function of t and x, periodic in x, that can be expressed as

$$\Psi_0(\hat{y}_{L,U}(x,\tau+1)) = \sum_{n=-N_M}^{n=+N_M} \left(\Xi_{L,U}^{(n)}\right)_{\tau+1} e^{in\alpha x} . \tag{6.3.11}$$

The volume flux represents an x-periodic function that can be written in the form of a Fourier expansion

$$Q(x,\tau+1) = \sum_{n=-N_M}^{n=+N_M} q^{(n)}(\tau+1) e^{in\alpha x} , \tag{6.3.12}$$

where the zero term, i.e., $q^{(0)}$, represents the net mass flux along the conduit. The value of $q^{(0)}(\tau+1)$ is assumed in this analysis to be known and independent of time, and equal to the flow rate of the reference flow, i.e., $q^{(0)} = \frac{4}{3}$.

Substitution of (6.2.13), (6.3.10)-(6.3.12) into Eq. (6.3.9) and extraction of mode zero results in

$$C_U(\tau+1) = q^{(0)} + C_L(\tau+1) + \Xi_L^{(0)} - \Xi_U^{(0)} . \tag{6.3.13}$$

One of the constants, either C_U or C_L, can be selected arbitrarily and the other one follows from (6.3.13). In the description given below the latter one has been selected arbitrarily by introducing condition Ψ_T=0 at a conveniently selected point x=x_0 at the lower wall resulting in

$$C_L(\tau+1) = -\Xi_L^{(0)} + \sum_{n=-N_A, n\neq 0}^{n=+N_A} (in\alpha\Gamma)^{-1} (A_L^{(n)})'_{\tau+1} e^{in\alpha x_0} . \tag{6.3.14}$$

Substitution of (6.3.14) into (6.3.13) gives

$$C_U(\tau+1) = -\Xi_U^{(0)} + \sum_{n=-N_A, n\neq 0}^{n=+N_A} (in\alpha\Gamma)^{-1} (A_L^{(n)})'_{\tau+1}\, e^{in\alpha x_0} + q^{(0)}. \tag{6.3.15}$$

Substitution of (6.3.14)-(6.3.15) into (6.3.8c) results in the form of the closing conditions useful for numerical implementation, i.e.,

$$\sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_{k,\tau+1}^{(m)} \left(w_{k,\tau+1}^{(m)^*} \right)_L = -\Xi_L^{(0)} + \sum_{n=-N_A, n\neq 0}^{n=+N_A} (in\alpha\Gamma)^{-1} (A_L^{(n)})'_{\tau+1}\, e^{in\alpha x_0}, \tag{6.3.16a}$$

$$\sum_{m=-N_M}^{m=+N_M} \sum_{k=0}^{k=N_T} Z_{k,\tau+1}^{(m)} \left(w_{k,\tau+1}^{(m)^*} \right)_U = -\Xi_U^{(0)} + \sum_{n=-N_A, n\neq 0}^{n=+N_A} (in\alpha\Gamma)^{-1} (A_L^{(n)})'_{\tau+1}\, e^{in\alpha x_0} + q^{(0)}. \tag{6.3.16b}$$

Equations (6.2.20), (6.3.8a-b) and (6.3.16) form a complete set of algebraic equations for the unknown coefficients $Z_{k,\tau+1}^{(n)}$, k = 0,....,N_T, n=0,....,N_M. A solution of this set moves calculations forward by one time step. The performance of the algorithm and various methods of solution of the algebraic equations are discussed in the next section.

6.4. Performance of the algorithm

Two different problems have been selected to test and characterize the performance of the algorithm. The test problems involve (i) peristaltic flow and (ii) pulsatile flow. The selection of these test problems enables us to show the applicability of the proposed algorithm in solving flow problems of considerable practical interest, particularly in the field of biomedical engineering.

6.4.1. Peristaltic flow

6.4.1.1. Problem prototype

A general form of peristaltic flow can be modelled by considering peristaltic wave traveling along the conduit walls, whose locations can be described as

$$y_L(x,t) = -1 + \sum_{n=-N_m, n\neq 0}^{n=+N_M} H_L^{(n)} e^{in\alpha(x-ct)}, \qquad y_U(x,t) = 1 + \sum_{n=-N_m, n\neq 0}^{n=+N_M} H_U^{(n)} e^{in\alpha(x-ct)}, \quad (6.4.1a,b)$$

where c denotes the phase speed of the wave and α stands for its wave number. The simplest situation corresponds to the wave profiles described by only one Fourier mode, resulting in the wall motions described by

$$y_L(x,t) = -1 + S\cos[\alpha(x-ct)] = -1 + (0.5Se^{i\alpha(x-ct)} + CC), \quad (6.4.2a)$$

$$y_U(x,t) = 1 - S\cos[\alpha(x-ct)] = 1 - (0.5Se^{i\alpha(x-ct)} + CC), \quad (6.4.2b)$$

where S stands for the amplitude of the wave and CC denotes complex conjugate. Figure 6.2 illustrates changes in the location of the walls as a function of time.

Peristaltic flow represents a convenient test problem because it can be converted into a steady, fixed boundary problem in the moving frame of reference (X,y) using Galilean transformation in the form

$$X = x - ct. \quad (6.4.3)$$

This is an easier problem and its solution provides a convenient comparison problem for the direct numerical solution of the time dependent, moving boundary problem in the fixed frame of reference. In addition, this problem provides a convenient test for characterization of the accuracy of spatial discretization.

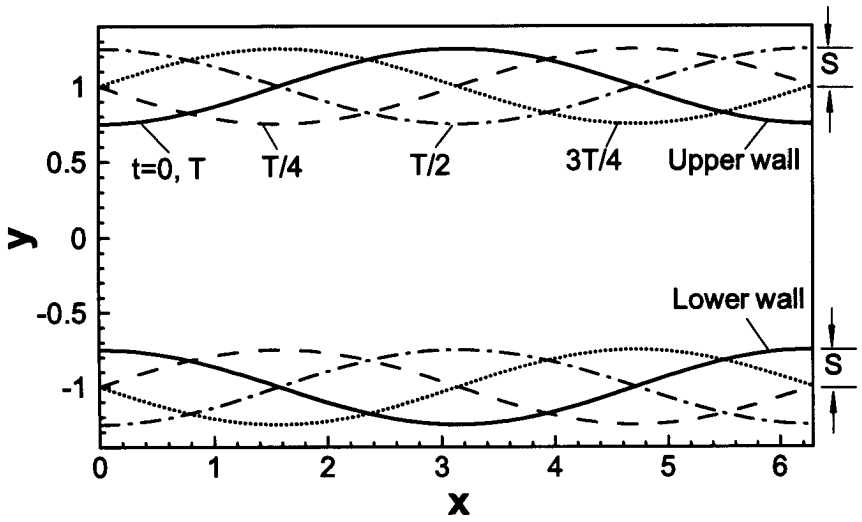


Figure 6.2. *Locations of the conduit's walls deformed by peristaltic wave described by Eq.(6.4.2) with the amplitude S=0.25, the wave number α=1.0 and the phase speed c=π at times t=0, T/4, T/2, 3T/4 and T, where T denotes one time period.*

The full problem in the moving frame of reference (X,y) takes the form

$$-\mathrm{Re}^{-1}\nabla^4\Psi_1+[(u_0-c)\partial_X(\nabla^2)-D^2u_0\partial_X]\Psi_1=-\partial_y[\partial_X(u_1u_1)+\partial_y(u_1v_1)]+\partial_X[\partial_X(u_1v_1)+\partial_y(v_1v_1)] \tag{6.4.4}$$

where $\nabla^2=\partial_{XX}+\partial_{yy}$ and the boundary shapes take the forms

$$y_L(X)=-1+(0.5Se^{i\alpha X}+CC), \qquad y_U(X)=1-(0.5Se^{i\alpha X}+CC). \tag{6.4.5a,b}$$

The boundary conditions in the moving frame of reference (X,y) have the form

$$u(X,y_{L,U}(X))=0, \quad v(y_{L,U}(X))=-c\sum_{n=-N_M,n\neq0}^{n=+N_M} in\alpha H_{L,U}^{(n)}e^{in\alpha x}. \tag{6.4.6a,b}$$

The stationary problem (6.4.4)-(6.4.6) shall be referred to as the model problem #1, and will be solved using the steady version of the technique discussed in the previous sections.

The same problem expressed in the fixed frame of reference (x,y) consists of the field equation (6.2.7), boundary conditions (6.2.9) with wall motions given by (6.4.2). This problem can be solved directly as a time dependent, moving boundary problem - we shall refer to this problem as the model problem #2. The solution of the model problem #1 is used as initial condition for the model problem #2 in various tests discussed later in the text, unless explicitly stated otherwise.

6.4.1.2. Solution of the model problem #1

Model problem #1 provides an opportunity to demonstrate the spectral accuracy of the spatial discretization. In order to have a meaningful discussion of error, we have produced a machine-accurate reference solution for this problem using the Domain Transformation (DT) method where the irregular flow domain is analytically mapped into a regular computational domain resulting in the classical treatment of boundary conditions. We have used the same spatial discretization as in the case of the IBC method, i.e., Fourier expansions in the X-direction and Chebyshev expansions in the y-direction. Details of the DT method can be found in Appendix J.

We define error in the evaluation of the u- and v-velocity components at any location (X,y) as

$$u_{er}(X,y) = u_{IBC}(X,y) - u_{DT}(X,y), \qquad v_{er}(X,y) = v_{IBC}(X,y) - v_{DT}(X,y), \quad (6.4.7a,b)$$

where the subscript 'IBC' denotes solution obtained using the IBC algorithm and the subscript 'DT' denotes the machine-accurate reference solution determined using the DT method. We shall use the L_∞ norms defined as

$$\|u_{er}\|_\infty = \sup_{\substack{0\le X\le 2\pi/\alpha \\ y_L(X)\le X\le y_U(X)}} |u_{er}(X,y)|, \qquad \|v_{er}\|_\infty = \sup_{\substack{0\le X\le 2\pi/\alpha \\ y_L(X)\le X\le y_U(X)}} |v_{er}(X,y)|, \qquad (6.4.8a,b)$$

as a measure of error over the whole flow domain. Error in the enforcement of flow boundary conditions is of special interest and is measured using the L_∞ norms defined as

$$\|u_{er,BC}(X)\|_\infty = \sup_{0\le X\le 2\pi/\alpha} \left\{ \left|\left(u_{er,BC}(X)\right)_L\right|, \left|\left(u_{er,BC}(X)\right)_U\right| \right\}, \qquad (6.4.9a)$$

$$\|v_{er,BC}(X)\|_\infty = \sup_{0\le X\le 2\pi/\alpha} \left\{ \left|\left(v_{er,BC}(X)\right)_L\right|, \left|\left(v_{er,BC}(X)\right)_U\right| \right\}, \qquad (6.4.9b)$$

where

$$\left(u_{er,BC}(X)\right)_{L,U} = u(X, y_{L,U}(X)) ,$$

$$\left(v_{er,BC}(X)\right)_{L,U} = v(X, y_{L,U}(X)) - \left(v_{wall}(X)\right)_{L,U} . \qquad (6.4.10a,b)$$

Here the terms $\left(v_{wall}(X)\right)_L$ and $\left(v_{wall}(X)\right)_U$ stand for the actual velocities of the lower and upper walls, respectively. We shall focus further discussion on the error in the u-velocity component, i.e., u_{er} . This error is related to the error of the y-derivative of the stream function which needs to be evaluated numerically and thus evaluation of u is potentially less accurate than evaluation of v.

The Chebyshev expansions (6.2.18) with coefficients calculated using the Galerkin procedure are expected to deliver spectral accuracy in the y-direction with the increasing number of terms N_T. The validation of this fact is illustrated in Fig. 6.3.

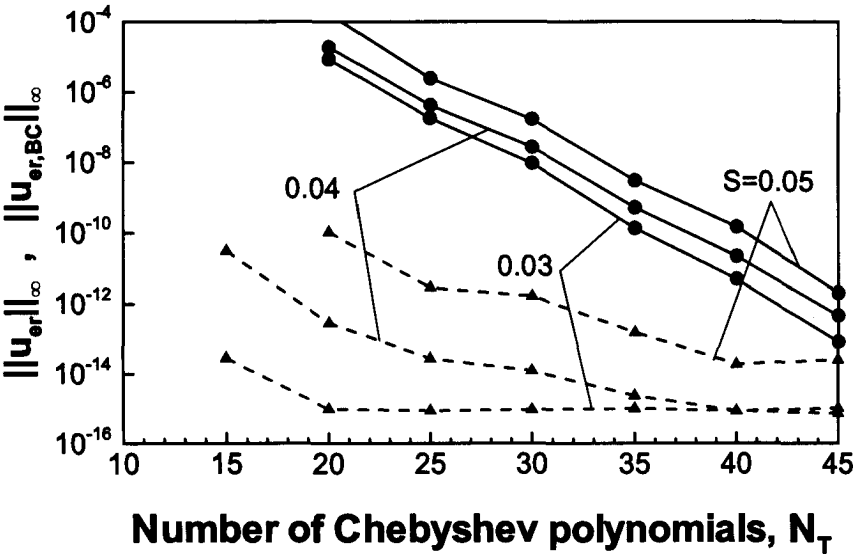


Figure 6.3. *Variations of the norms* $\|u_{er}\|_\infty$ *(solid lines) and* $\|u_{er,BC}\|_\infty$ *(dashed lines) as a function of the number of Chebyshev polynomials* N_T *used in the computations for the model problem #1 (see Eqs (6.4.4)-(6.4.6)) for the peristaltic wave with the wave number* $\alpha=1$*, the phase speed* $c=\pi$ *and different amplitudes S. Computations have been carried out for the flow Reynolds number* $Re=100$ *using* $N_M=20$ *Fourier modes.*

Figure 6.3 also shows that the use of a single norm $\|u_{er,BC}\|_\infty$ as an error measure could be misleading when the number of Chebyshev polynomials employed in the computations is not adequate. The reason behind this property can be explained with the aid results displayed in Fig. 6.4, which shows formation of boundary layers in the distribution of modal functions in the zones around the walls. Although the discretization in the y-direction using the Chebyshev expansions is fairly standard, the existence of boundary layers creates special problems. Inadequate number of polynomials can lead to spurious oscillations in the distribution of the modal functions and increase the error in the overall solution although the boundary conditions may still be satisfied with very high accuracy (see Fig. 6.3). In most cases, sixty Chebyshev polynomials provide machine accuracy. However, when the wavelengths of the peristaltic wave become shorter (α increases), higher Fourier modes begin to play important role and one needs to increase the number of Chebyshev polynomials in order to resolve the wall boundary layers with reasonable

accuracy. Figure 6.4 shows that these layers become extremely thin for larger values of α and for higher Fourier modes. Modal functions change very rapidly inside these layers while they are nearly zero in the rest of the domain.

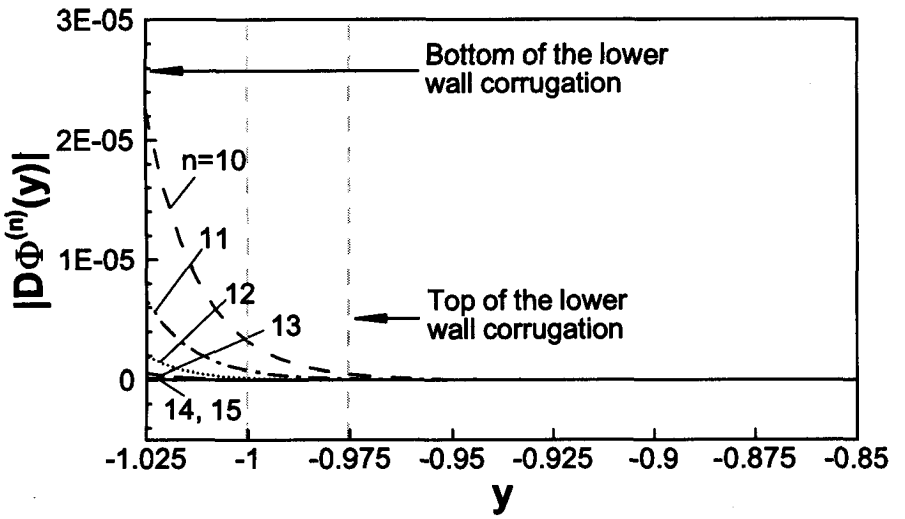


Figure 6.4. *Distribution of $|D\Phi^{(n)}|$ for higher modes ($n \geq 10$) in the vicinity of the lower wall for the model problem #1 (see Eqs (6.4.4)-(6.4.6)) for the peristaltic wave with the wave number $\alpha=5$, the amplitude $S=0.025$ and the phase speed $c=\pi$. Computations have bee carried out for the flow Reynolds number $Re=100$ using $N_M=15$ Fourier modes and $N_T=100$ Chebyshev polynomials.*

The convergence of the truncated Fourier series is pertinent to the second aspect of the spatial discretization, i.e., accuracy of discretization in the x-direction. Chebyshev norm defined as

$$\left\|D\Phi^{(n)}\right\|_{\hat{\omega}} = \sqrt{\Gamma^2 \int_{-1}^{1} D\Phi^{(n)}(\hat{y},t) D\Phi^{(n)*}(\hat{y},t)\hat{\omega}(\hat{y})d\hat{y}} \quad , \quad \hat{\omega} = 1/\sqrt{1-\hat{y}^2} \tag{6.4.11}$$

is used as a measure of the magnitude of the derivative of the modal function $\Phi^{(n)}$ (i.e., the u-velocity component). Results displayed in Fig. 6.5 demonstrate that this norm decreases exponentially as a function of the mode number n.

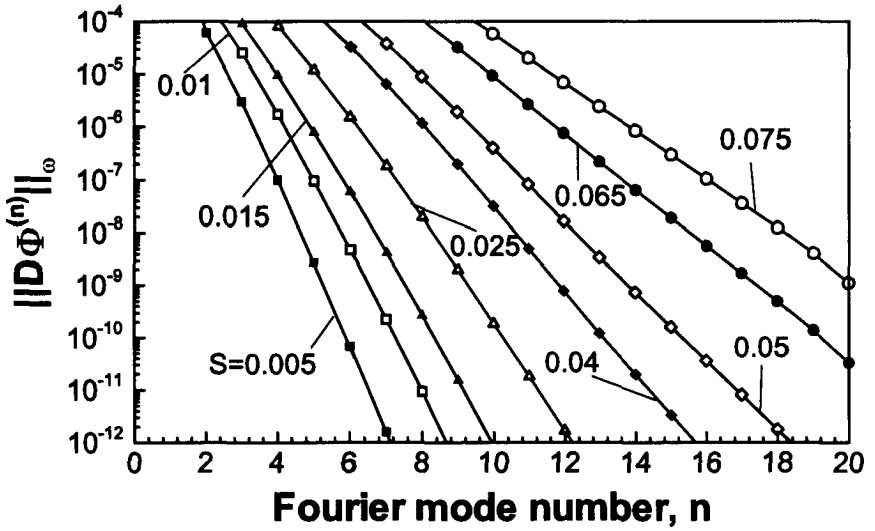


Figure 6.5. *Variations of the Chebyshev norm of $D\Phi^{(n)}$ (see Eq.4.5) as a function of the Fourier mode number n for the model problem #1 (see Eqs (6.4.4)-(6.4.6)) for different amplitudes S of the peristaltic wave with the wave number α=1.0 and the phase speed $c=\pi$. Computations have been carried out for the flow Reynolds number Re=100 using N_M=20 Fourier modes and N_T=80 Chebyshev polynomials.*

The next question of interest is the identification of the number of Fourier modes N_M required for a desired level of accuracy of the solution. Results displayed in Fig. 6.6 demonstrate that when a sufficient number of Chebyshev polynomials is used (see Fig. 6.3), the maximum error over the computational domain becomes equal to the maximum error in the enforcement of flow boundary conditions, i.e.,

$$\|u_{er}\|_\infty = \|u_{er,BC}\|_\infty \quad \text{and} \quad \|v_{er}\|_\infty = \|v_{er,BC}\|_\infty \quad \text{when} \quad \frac{d}{dN_T}(\|u_{er}\|_\infty, \|v_{er}\|_\infty) \approx 0 . \tag{6.4.12}$$

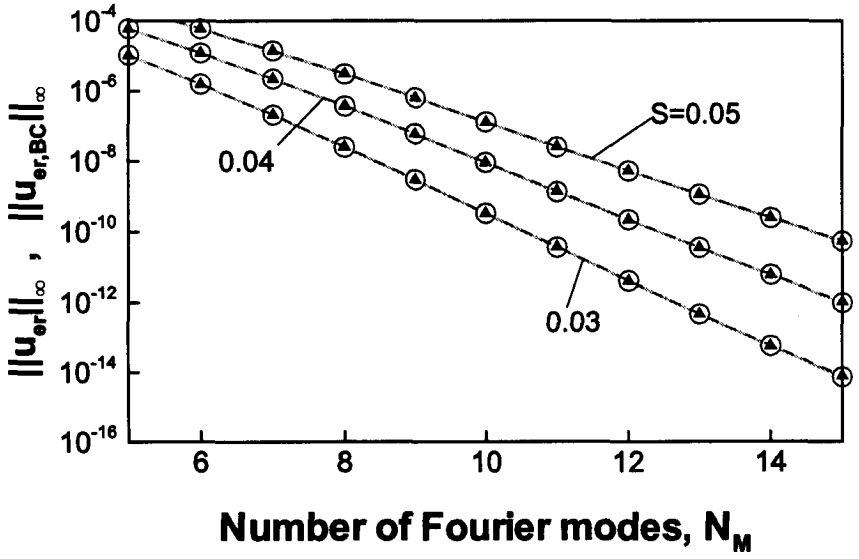


Figure 6.6. *Variations of the norms* $\|u_{er}\|_\infty$ *(circles) and* $\|u_{er,BC}\|_\infty$ *(triangles) as a function of the number of Fourier modes* N_M *used in the computations for the model problem #1 (see Eqs (6.4.4)-(6.4.6)) for the peristaltic wave with the wave number* $\alpha=1$*, the phase velocity* $c=\pi$ *and selected values of the amplitudes S. Computations have been carried out for the flow Reynolds number Re=100 using* $N_T=80$ *Chebyshev polynomials*

Therefore, the number of Fourier modes N_M used in the computations contributes only to the error in the enforcement of flow boundary conditions while the magnitude of the error inside the solution domain is determined by the number of Chebyshev polynomials. This fact is further substantiated through the results displayed in Fig. 6.7, which demonstrate that the error reaches maximum around the wall and decreases rapidly as one moves away from the wall. Thus, one can use the norm $\|u_{er,BC}\|_\infty$ to quantify the maximum error for any number of Fourier modes N_M assuming that a sufficient number of Chebyshev polynomials N_T have been used.

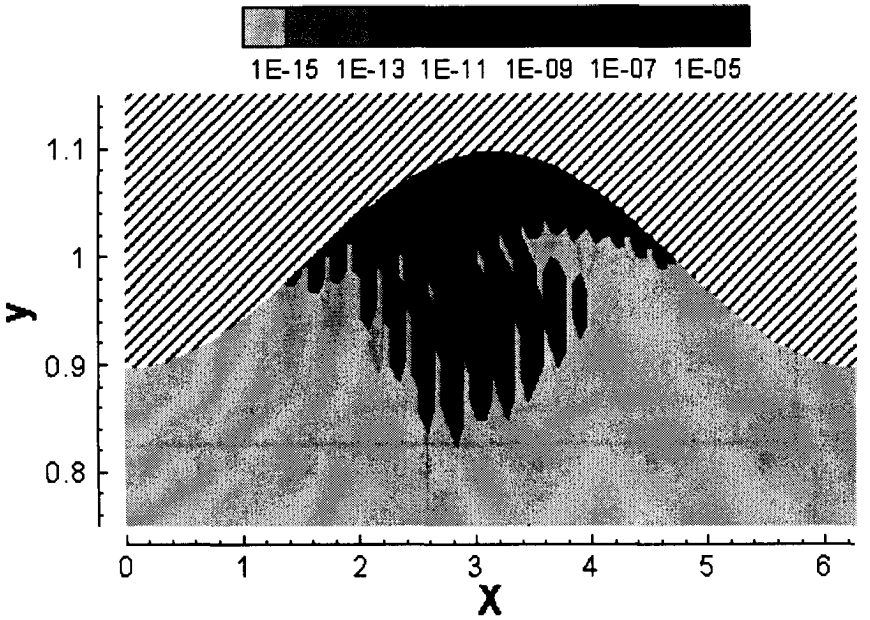


Figure 6.7. *Distribution of the absolute value of the error $u_{er}(X,y)$ (see Eq.(6.4.7)) around the upper wall for the model problem #1 (see Eqs (6.4.4)-(6.4.6)) for the peristaltic wave with the wave number $\alpha=1.0$, the amplitude $S=0.1$ and the phase speed $c=\pi$. Computations have been carried out for the flow Reynolds number $Re=100$ using $N_M=20$ Fourier modes and $N_T=100$ Chebyshev polynomials.*

Figure 6.8 shows distribution of error in the enforcement of flow boundary conditions for the u- and v-velocity components along the upper wall. It can be seen that the errors oscillate along the conduit and that the locations of the maxima of the amplitudes overlap with the location where the peristaltic wave maximizes conduit opening. This figure also illustrates that the magnitude of the error in the u-velocity component is slightly higher than that in the v-velocity component, as had already been pointed out in the preceding discussion.

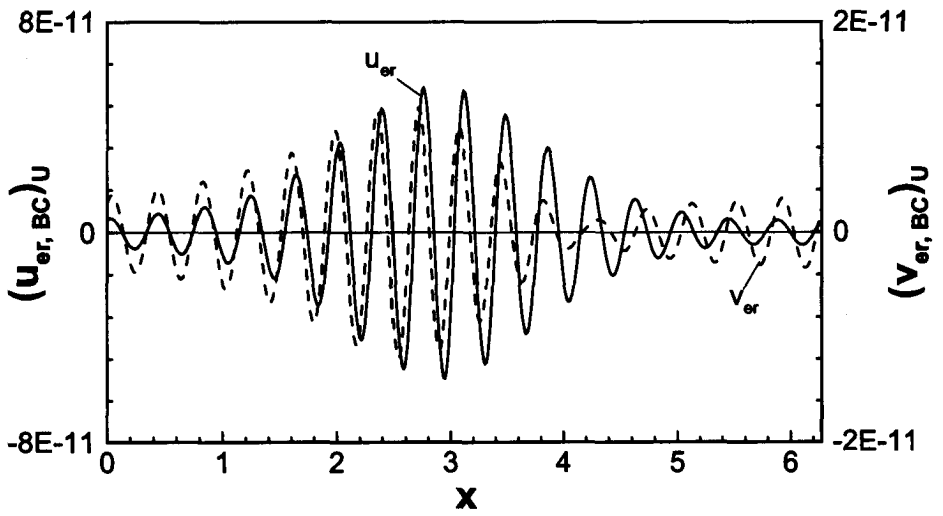


Figure 6.8. *Distribution of the boundary errors $(u_{er,BC})_U$ (see Eq.(6.4.10a), solid line) and $(v_{er,BC})_U$ (see Eq.(6.4.10b), dash line) for the model problem #1 (see Eqs (6.4.4)-(6.4.6)) for the peristaltic wave with the phase speed $c=\pi$, the amplitude $S=0.05$ and the wave number $\alpha=1$. Computations have been carried out for the flow Reynolds number $Re=100$ using $N_M=15$ Fourier modes and $N_T=80$ Chebyshev polynomials.*

Variations of the error $\|u_{er}\|_\infty$ as a function of the amplitude S of the peristaltic wave for fixed values of its wave number α, and as a function of its wave number α for fixed amplitudes S are illustrated in Figs 9 and 10, respectively. The results shown demonstrate that for certain combinations of α and S, the accuracy can be maintained at the machine level. However, the error increases almost exponentially when values of α and S reach certain critical threshold. The error can be controlled by increasing the number of Fourier modes used in the computations and/or by using the over-determined formulation [25], which provides ability to simulate dynamics of peristaltic waves with more complex profiles. Both techniques do not however affect the qualitative character of the error variations; they merely increase the threshold of α and S that leads to a rapid error increase.

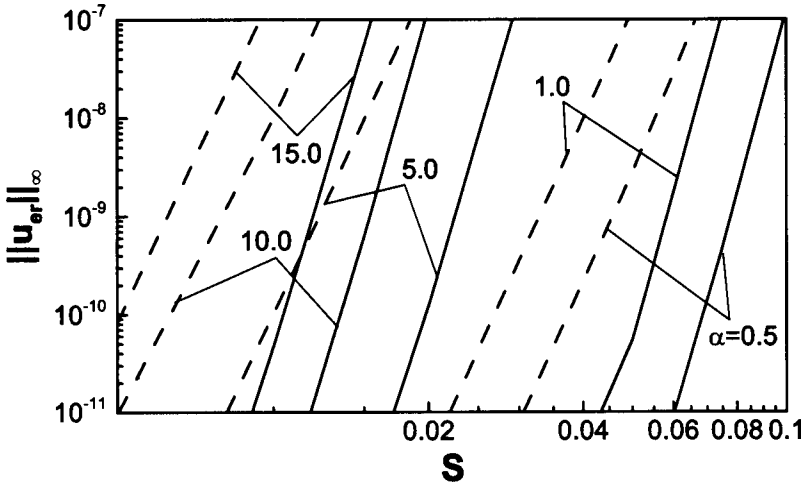


Figure 6.9. *Variations of the $\|u_{er}(X)\|_\infty$ norm (see Eq. (6.4.8)) as a function of the amplitude S of the peristaltic wave with the phase speed $c=\pi$ and with selected values of the wave number α in the case of flow with the Reynolds number Re=100 (model problem #1, see Eqs (6.4.4)-(6.4.6)). The dashed and solid lines correspond to results obtained with the N_M = 10 and 15 Fourier modes, respectively. N_T=80 Chebyshev polynomials were used in the computations.*

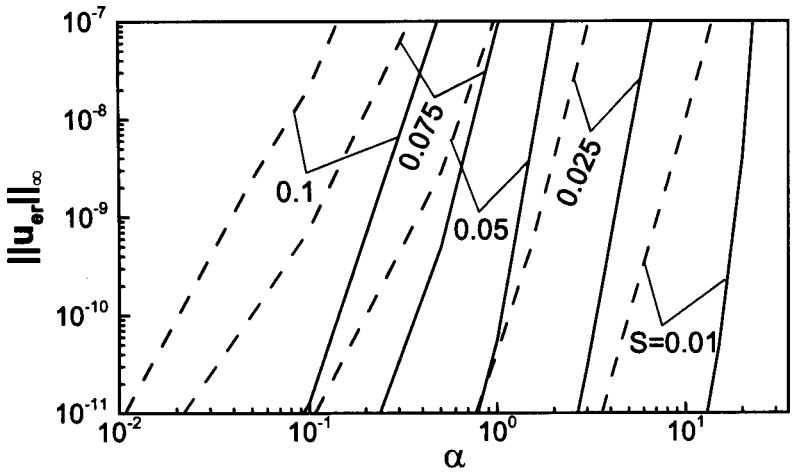


Figure 6.10. *Variations of the $\|u_{er}(X)\|_\infty$ norm (see Eq. (6.4.8)) as a function of the wave number α of the peristaltic wave with the wave speed $c=\pi$ and with selected values of the amplitude S. Other parameters are as in Fig. 6.9.*

6.4.1.3. Solution of the model problem #2

We shall now focus our discussion on the model problem #2, i.e., a moving boundary problem expressed in a fixed frame of reference (x,y). Solution of the model problem #1 provides initial conditions via the reversed Galileo transformation. We shall use the L_∞ norms as quantitative measures of error associated with the enforcement of flow boundary conditions, i.e.,

$$\left\|u_{er,BC}(x,t)\right\|_\infty = \sup_{0\le X\le 2\pi/\alpha} \left\{ \left(\left|u_{er,BC}(x,t)\right|\right)_L, \left(\left|u_{er,BC}(x,t)\right|\right)_U \right\} , \tag{6.4.13a}$$

$$\left\|v_{er,BC}(x,t)\right\|_\infty = \sup_{0\le X\le 2\pi/\alpha} \left\{ \left(\left|v_{er,BC}(x,t)\right|\right)_L, \left(\left|v_{er,BC}(x,t)\right|\right)_U \right\} , \tag{6.4.13b}$$

where

$$\left(u_{er,BC}(x,t)\right)_{L,U} = u(x, y_{L,U}(x,t), t) \ ,$$

$$\left(v_{er,BC}(x,t)\right)_{L,U} = v(x, y_{L,U}(x,t), t) - (v_{wall})_{L,U}(x,t) . \tag{6.4.14a,b}$$

The test problem is solved using different temporal discretizations and relevant discussion is provided in Section 6.2. Figure 6.11 illustrates the time history of $\left\|u_{er}(x,t)\right\|_\infty$ and demonstrates that the location of the maximum error follows the location of the maximum conduit opening as it moves in the positive x-direction, similarly as in the case of solution obtained in the moving frame of reference (see Fig. 6.8). The magnitude of this error remains approximately constant while several waves pass through the computational box.

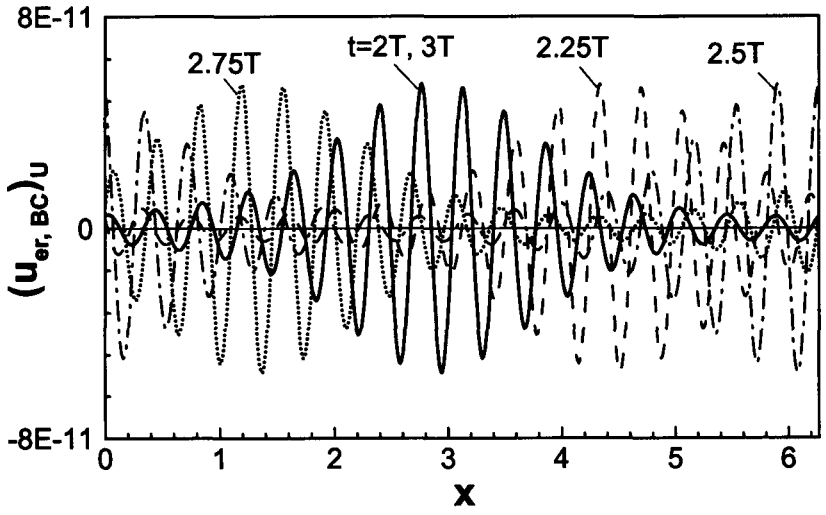


Figure 6.11. *Distribution of the boundary error $(u_{er,BC})_U$ (see Eq. (6.4.14)) at times t = 2T, 2.25T, 2.5T, 2.75T and 3T, where T stand for one time period, for the model problem #2 with the amplitude of the peristaltic wave S=0.05, the wave number α=1.0 and the phase speed c=π. Computations have been carried out for the flow Reynolds number Re=100 using N_M=15 Fourier modes, N_T=80 Chebyshev polynomials and the third-order implicit temporal discretization scheme with the time step Δt=0.001. Solution of the model problem #1 was used as the initial condition.*

Variations of the u-velocity component at a few test points are displayed in Fig. 6.12. The test points have the same y-coordinates but are spaced apart in the x-direction by a distance of λ/4, where λ is the wavelength of the peristaltic wave. The results show the expected phase differences associated with different locations of the test points. Both figures, i.e., Figs 11 and 12, demonstrate the expected periodic variations in time of the computed quantities.

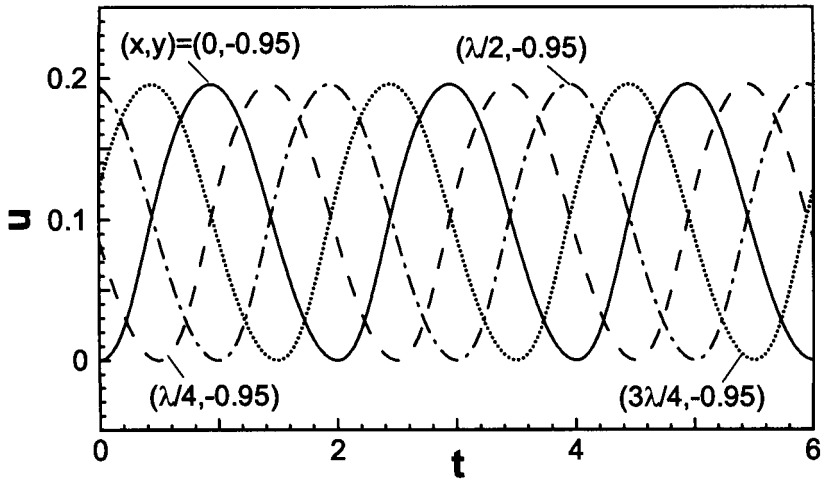


Figure 6.12. *Variations of the u-velocity component over three time periods at four test points for the model problem #2 solved directly as a moving boundary problem in the fixed reference frame. $\lambda=2\pi/\alpha$ stands for the wavelength of the peristaltic wave. Other conditions are as in Fig. 6.11.*

6.4.2. Pulsatile Flow

6.4.2.1. Problem Prototype

Pulsatile flows can be modelled by replacing the flat conduit walls with elastic standing waves, which can be expressed as

$$y_L(x,t) = -1 + \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_L^{(n)}(t)\, e^{in\alpha x}, \qquad y_U(x,t) = 1 + \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_L^{(n)}(t)\, e^{in\alpha x}, \quad (6.4.15a,b)$$

and, in the simplest case of a sinusoidal wave, the shape and motion of the walls can be described as

$$y_L(x,t) = -1 + S\cos(\omega t)\cos(\alpha x) = -1 + (0.5\, S\cos(\omega t)\, e^{i\alpha x} + CC), \quad (6.4.16a)$$

$$y_U(x,t) = 1 - S\cos(\omega t)\cos(\alpha x) = 1 - (0.5\, S\cos(\omega t)\, e^{i\alpha x} + CC), \qquad (6.4.16b)$$

where α denotes the wave number of the standing wave, S stands for its amplitude and ω denotes its frequency. In the above, CC implies complex conjugate. The character of motion of the walls in this case is illustrated in Fig. 6.13.

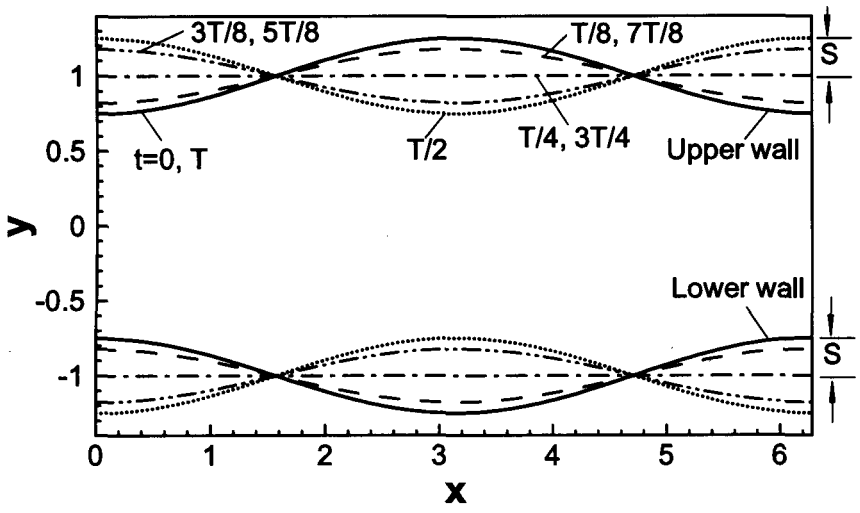


Figure 6.13. *Locations of the conduit's walls deformed by an elastic standing wave with the wave number α=1.0, the amplitude S=0.25 and the frequency $\omega=\pi$ at times t=0, T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8 and T, where T denotes one time period.*

The complete test problem consists of the discretized field equations (6.2.20), the boundary constraints (6.3.8a-b, 6.3.16), the boundary motions described by (6.4.16) and suitable initial conditions consistent with the boundary shapes. We shall refer to this problem as the model problem #3. For convenience, we shall use the L_∞ norms defined by Eq.(6.4.13) as quantitative measures of error associated with the enforcement of flow boundary conditions.

6.4.2.2. Solution of the model problem #3

Figure 6.14 illustrates variations of the maximum error in the enforcement of the flow boundary condition for the u-velocity as a function of time over two time periods. It can be seen that the magnitude of the error changes periodically in time with frequency equal to double the frequency of the wave. The occurrence of the maximum error has certain phase lag with respect to the occurrence of the maximum conduit opening, with the phase lag increasing with an increase of Re. Results displayed in Fig. 6.15 demonstrate that the time lag does not depend on the wave amplitude, which implies that this lag is a function of the Reynolds number and the wave frequency only.

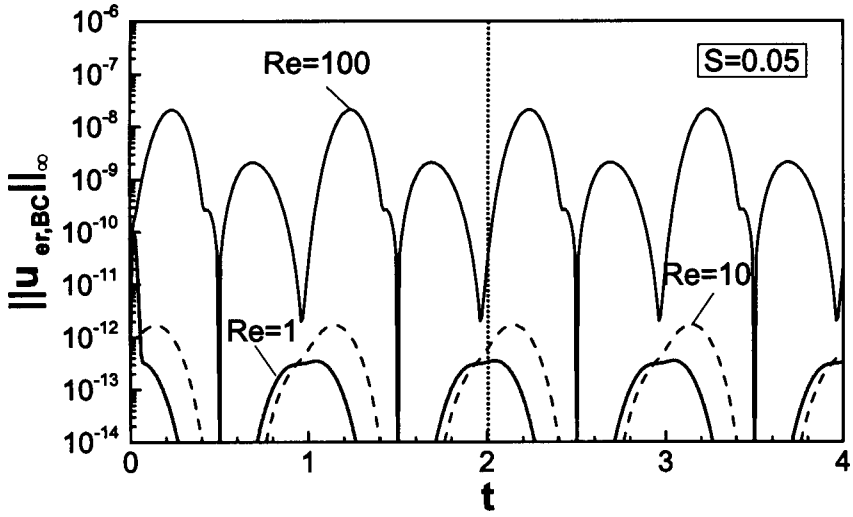


Figure 6.14. *Variations of the* $\|u_{er,BC}(x,t)\|_\infty$ *norm (see Eq. (6.4.13)) as a function of time over two time periods for the standing wave problem (model problem #3) with the wave number* $\alpha=1$*, the amplitude* $S=0.05$ *and the frequency* $\omega=\pi$ *for three Reynolds numbers, i.e.,* $Re=1$*, 10 and 100. Computations have been carried out using* $N_M=15$ *Fourier modes,* $N_T=80$ *Chebyshev polynomials and the third-order implicit temporal discretization scheme with the time step* $\Delta t=0.001$*. Solution of the steady, fixed boundary problem with the shape of the walls given by Eq.(6.4.16) was used as the initial condition.*

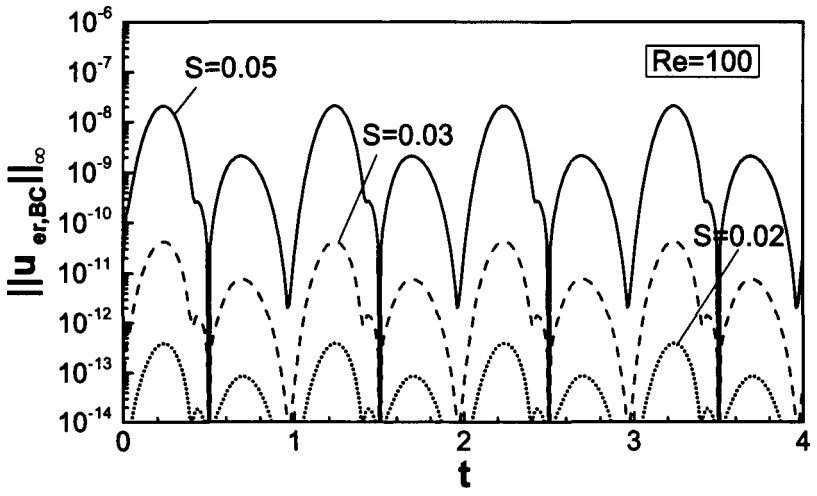


Figure 6.15. *Variations of the* $\|u_{er,BC}(x,t)\|_\infty$ *norm (see Eq. (6.4.13)) as a function of time over two time periods for the standing wave problem (model problem #3) with the wave number* $\alpha=1$ *and the frequency* $\omega=\pi$ *for three values of the wave amplitude, i.e.,* $S=0.02$, 0.03 *and* 0.05. *Other conditions are as in Fig. 6.14.*

Results shown in Fig. 6.16 demonstrate that the magnitude of the error can be reduced by increasing the number of Fourier modes used in the computations but its qualitative character remains unchanged. The spatial distribution of the error is illustrated in Fig. 6.17 after 2.125 and 2.625 cycles of the wave motion, i.e., when the error is largest (see Fig. 6.16). It can be seen that the maximum of the error occurs at a location corresponding to the maximum conduit opening at a given instant of time and the magnitudes of the maxima at the two time levels are essentially identical. The reader should note that the maximum of this error as a function of time occurs after the walls have already moved passed the position corresponding to the maximum possible conduit opening, as discussed in the previous paragraph.

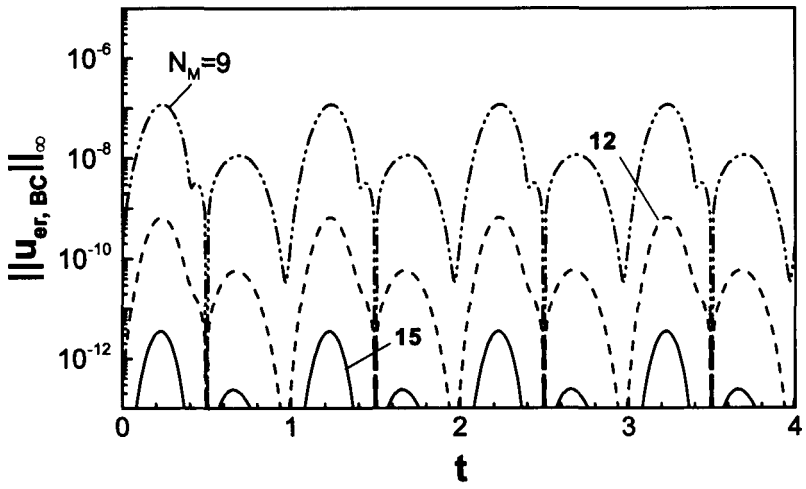


Figure 6.16. *Variations of the $\|u_{er,BC}(x,t)\|_\infty$ norm (see Eq. (6.4.13)) as a function of time over three time periods for the standing wave problem (model problem #3) with the wave number α=1, the amplitude S=0.025 and the frequency $\omega=\pi$. Computations have been carried out for the flow Reynolds number Re=100 using N_M = 9, 12 and 15 Fourier modes. Other conditions are as in Fig. 6.14.*

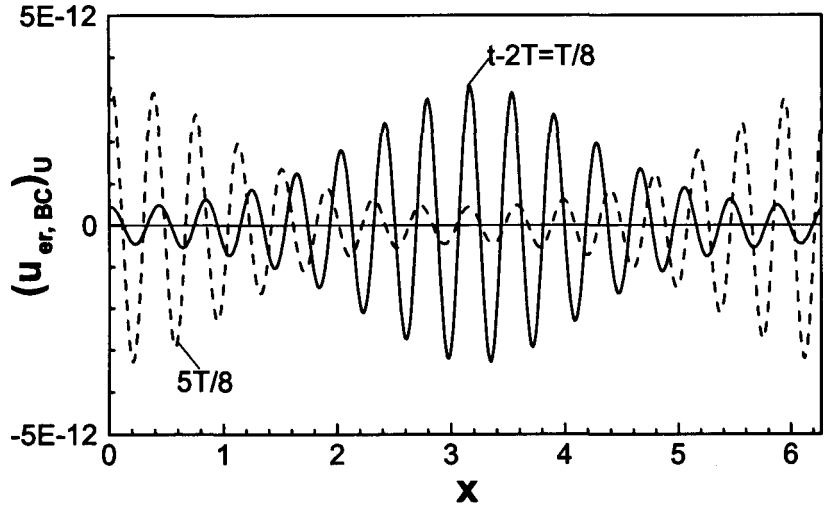


Figure 6.17. *Spatial distribution of the error $(u_{er,BC})_U$ (see Eq.(6.4.14)) at two different instances of time for the same standing wave problem as described in Fig. 6.16. Other conditions are as in Fig. 6.14.*

The Fourier spectra of the error in the enforcement of the u-velocity boundary conditions is given by the following relation

$$\left(u_{er,BC}(x,t)\right)_{L,U} = \sum_{n=-\infty}^{n=+\infty} \left(U^{(n)}_{er,BC}(t)\right)_{L,U} e^{inax} . \qquad (6.4.17)$$

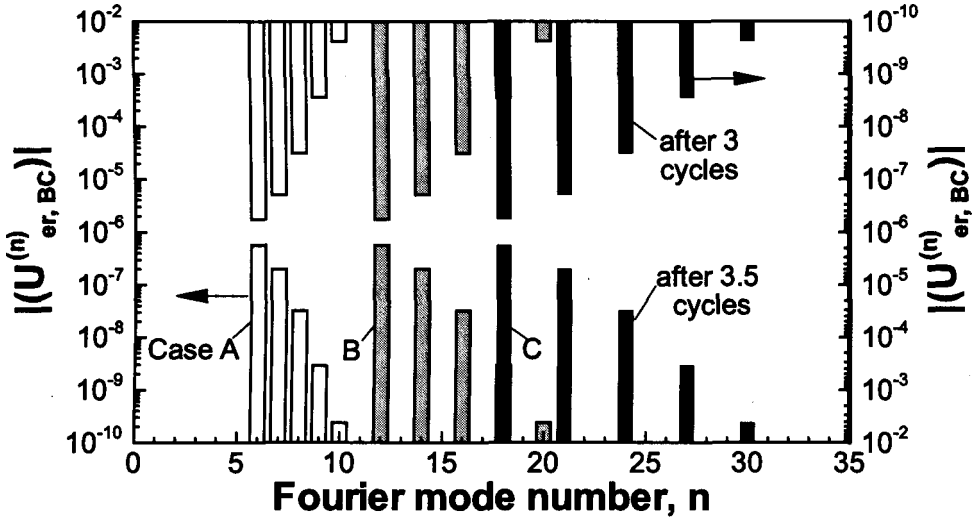


Figure 6.18. *Fourier spectra of distribution of the boundary error $(u_{er,BC})_U$ (see Eq. (6.4.17)) for the standing wave problem (model problem #3) with the wave amplitude S=0.025 and the frequency $\omega=\pi$. Three different forms of Fourier expansions were considered, i.e., case A: $\alpha=1.0$, $N_M=5$; case B: $\alpha=0.5$, $N_M=10$ and case C: $\alpha=1/3$, $N_M=15$. Computations have been carried out using $N_T=80$ Chebyshev polynomials and the third-order implicit temporal discretization scheme with the time step $\Delta t=0.001$. Solution of the steady, fixed boundary problem with the shape of the walls given by Eq.(6.4.16) was used as the initial condition.*

The computed spectra shown in Fig. 6.18 demonstrate that the first N_M Fourier modes have been eliminated, according to the construction of the boundary constraints described in Section 6.3. The largest error is associated with the first Fourier mode omitted in the enforcement of the flow boundary conditions; the error associated with the following modes rapidly decreases as the mode number increases. Figure 6.18 also displays results of tests carried out in order to check if the method produces any spurious spatial oscillations. Three cases were considered, i.e., in case (A) the wave was represented by

the principal Fourier mode and the calculations had been carried out with N_M=5 Fourier modes, in case (B) the wave was represented by the second Fourier mode (the principal mode had the wave number α=0.5), and in case (C) the wave was represented by the third Fourier mode (the principal mode had the wave number α=1/3). In order to have fully equivalent representations, the number of Fourier modes used in cases (B) and (C) were N_M=10 and N_M=15, respectively. The problem set up admitted sub-harmonics of the 1/2 type in case (B) and 1/3 type in case (C). The Fourier spectra shown in Fig. 6.18 demonstrate the equivalency of the results in all three cases. No sub-harmonic had been produced during the solution process and the modes expected to produce zero contributions in cases (B) and (C) behaved as expected. Temporal variations of the u- and v-velocity components at a test point displayed in Fig. 6.19 demonstrate the expected periodic variations in time, with the error of u being larger than error of v, as had already been discussed.

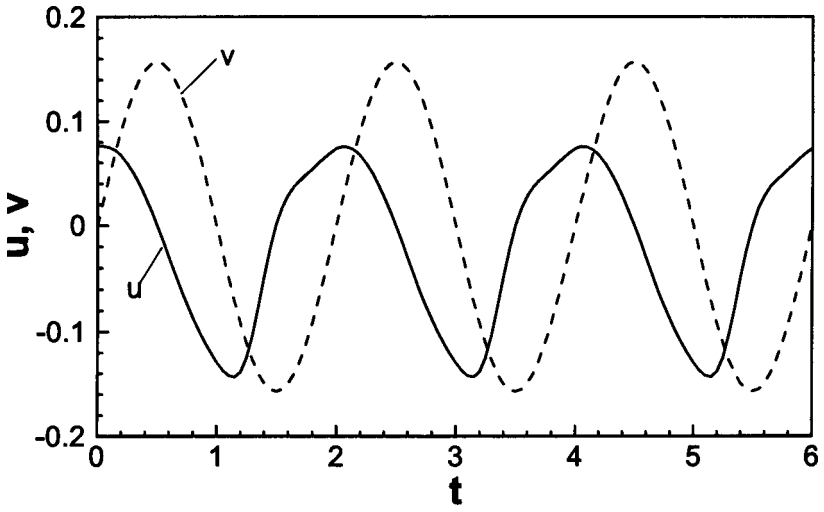


Figure 6.19. *The evolution of the u- (solid line) and v- (dashed line) velocity components at a test point (x,y)=(λ/4,-0.9) during the first four cycles of the standing wave (model problem #3). All other conditions are as in Figure 6.17.*

6.4.3. Temporal discretization

In the course of the present work we have tested four different fully-implicit temporal discretization schemes, which can deliver first-, second-, third- and fourth-order accuracy. The schemes treat all terms implicitly resulting in a nonlinear system of algebraic equations that has to be solved iteratively at each time step. The general form of the modal field equations (6.2.15) can be written as

$$\partial_t[\mathcal{L}_1\Phi^{(n)}] = \mathcal{L}_2\Phi^{(n)} + \mathcal{N}\Phi^{(n)}, \tag{6.4.18}$$

where $\mathcal{L}_1$, $\mathcal{L}_2$ denote linear and $\mathcal{N}$ stands for nonlinear differential operators acting on the unknown modal functions $\Phi^{(n)}$. The expression for advancing from $\Phi_\tau^{(n)}$ at time t to $\Phi_{\tau+1}^{(n)}$ at time t+Δt using the first-, the second-, the third- and the fourth-order schemes have the forms

$$[\Delta t^{-1}\mathcal{L}_1 - \mathcal{L}_2]\left\{\Phi_{\tau+1}^{(n)}\right\}^{J+1} = \mathcal{N}\left\{\Phi_{\tau+1}^{(n)}\right\}^{J} + \Delta t^{-1}\mathcal{L}_1\Phi_\tau^{(n)}, \tag{6.4.19a}$$

$$[(3/2)\Delta t^{-1}\mathcal{L}_1 - \mathcal{L}_2]\left\{\Phi_{\tau+1}^{(n)}\right\}^{J+1} = \mathcal{N}\left\{\Phi_{\tau+1}^{(n)}\right\}^{J} + \Delta t^{-1}\mathcal{L}_1[2\Phi_\tau^{(n)} - (1/2)\Phi_{\tau-1}^{(n)}], \tag{6.4.19b}$$

$$[(11/6)\Delta t^{-1}\mathcal{L}_1 - \mathcal{L}_2]\left\{\Phi_{\tau+1}^{(n)}\right\}^{J+1} = \mathcal{N}\left\{\Phi_{\tau+1}^{(n)}\right\}^{J} + \Delta t^{-1}\mathcal{L}_1[3\Phi_\tau^{(n)} - (3/2)\Phi_{\tau-1}^{(n)} + (1/3)\Phi_{\tau-2}^{(n)}], \tag{6.4.19c}$$

$$[(25/12)\Delta t^{-1}\mathcal{L}_1 - \mathcal{L}_2]\left\{\Phi_{\tau+1}^{(n)}\right\}^{J+1} = \mathcal{N}\left\{\Phi_{\tau+1}^{(n)}\right\}^{J} + \Delta t^{-1}\mathcal{L}_1[4\Phi_\tau^{(n)} - 3\Phi_{\tau-1}^{(n)} + (4/3)\Phi_{\tau-2}^{(n)} - (1/4)\Phi_{\tau-3}^{(n)}], \tag{6.4.19d}$$

respectively, where the superscript J+1 denotes the current iteration and J denotes the previous iteration The values of the modal functions $\Phi_{\tau+1}^{(n)}$ on the right hand side of Eqs

(6.4.19a)-(6.4.19d) associated with the nonlinear terms are initialized with the values of $\Phi_{\tau}^{(n)}$, i.e., with the solution from the previous time step. The first-order implicit scheme is self-starting, and is used to start the second-, the third- and the fourth-order methods. All methods were found to be numerically stable.

Computational cost of the implicit schemes cannot be predicted beforehand, as it is dominated by the number of iterations required in order to solve the nonlinear algebraic equations with the specified accuracy. We have used the convergence criteria based on the absolute difference between two subsequent solutions, and this difference had been set to 10^{-14} in all tests reported in this chapter. The required number of iterations is influenced by several factors including wave profiles, (e.g.., their wave number, amplitude, phase speed, frequency), the number of Fourier modes and Chebyshev polynomials used in the spatial discretization, the type of temporal discretization and the time step-size.

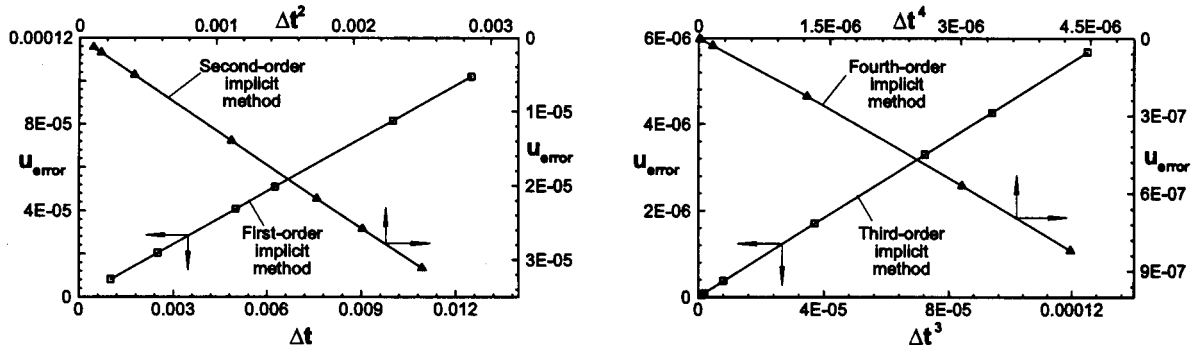
Results of the temporal grid convergence studies for the model problem #2 are reported using the error defined as

$$u_{er} = \sup_{\substack{0\le x\le 2\pi/\alpha \\ y_L(x,t)\le y\le y_U(x,t)}} \left| u_{difference}(x,y,t) \right|, \tag{6.4.20}$$

where

$$u_{difference}(x,y,t) = u_{MP\#2}(x,y,t) - u_{MP\#1}(x,y,t). \tag{6.4.21}$$

Here the term $u_{MP\#2}$ refers to the solution obtained through the direct solution of the moving boundary problem in the fixed frame of reference and the term $u_{MP\#1}$ denotes solution of the corresponding fixed boundary problem in the moving frame of reference and converted into the fixed-frame of reference using the reversed Galileo transformation. The results shown in Fig. 6.20 corresponds to time t=1.0 and demonstrate that all the temporal schemes deliver the theoretically predicted accuracy.



(A) *(B)*

Figure 6.20. *Variations of the error u_{er} (see Eq.(6.4.20)) as a function of the time step Δt. Results shown were obtained for the travelling wave problem (model problem #2) with the wave amplitude $S=0.05$, the wave number $\alpha=1.0$ and the phase speed $c=\pi$. Computations have been carried out for the Reynolds number $Re=100$ using $N_M=10$ number of Fourier modes and $N_T=100$ number of Chebyshev polynomials. Solution of the model problem #1 was taken as the initial condition. Fig. 6.20A – first- and second-order methods, Fig. 6.20B – third- and forth-order methods.*

6.4.4. Computational efficiency and effectiveness

All implicit algorithms require solution of a large system of nonlinear algebraic equations at each time step. The nonlinear system is solved in an iterative manner, i.e., the nonlinear terms are moved to the right hand side and the resulting linear system is solved repetitively with the right hand side being updated after each iteration. The solution process requires evaluation of the inverse (or equivalent) of the coefficient matrix only once for the first iteration, with subsequent iterations reduced to multiplication of the inverse (or equivalent) with the recomputed right hand side vector. Because of the size of the system, one needs to evaluate the performance of various possible solution strategies of the relevant linear system.

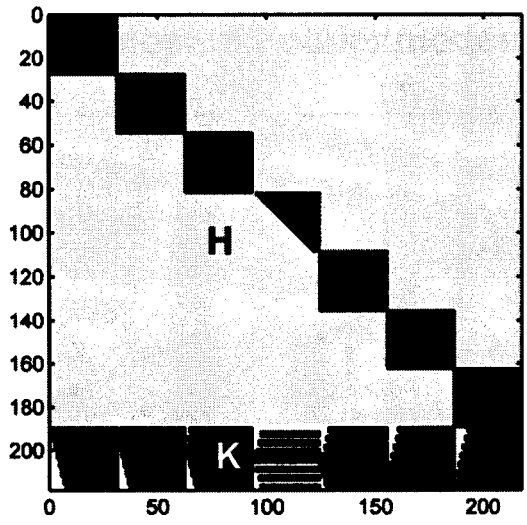


Figure 6.21. *Structure of the coefficient matrix for the model problem #2 constructed using N_M=3 Fourier modes and N_T=30 Chebyshev polynomials. Non-zero entries are marked in black.*

The linear system has the form

$$\mathbf{Lx=R} \tag{6.4.22}$$

where **L** is a p×p coefficient matrix with $p = (2N_M+1)(N_T+1)$, **x** is a p-dimensional column vector of unknowns and **R** is a p-dimensional column vector representing information contained in the nonlinear terms and taken either from the previous time step or from the previous iteration. The system is organized by grouping entries corresponding to the linear part of the field equations in matrix **H** of size q×p, $q = (2N_M+1)(N_T-3)$, and entries corresponding to boundary relations in matrix **K** of size (p-q)×p, resulting in the structure of coefficient matrix **L** illustrated in Fig. 6.21. Matrix **H** has the block diagonal structure with each block corresponding to a different modal equation and having the size $(N_T+1)\times(N_T-3)$. This matrix needs to be computed only ones as it does not depend on boundary motions. Matrix **K** is full as it provides coupling between different modes and it needs to be recomputed at each time step in order to capture boundary motions. Structure of matrix **L** illustrates advantage of the IBC algorithm.

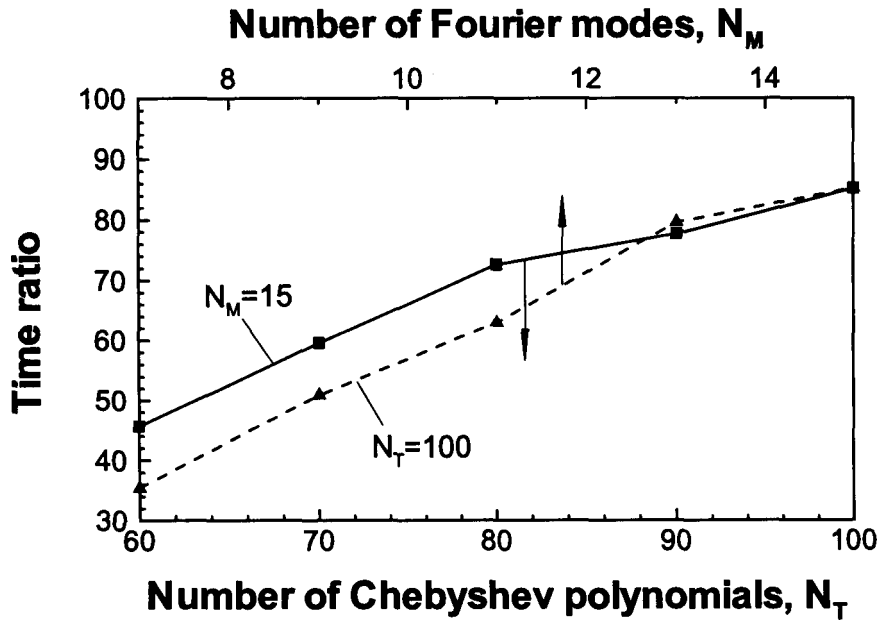


Figure 6.22. *Variations of the ratio of time required to construct the coefficient matrix using the DT method as compared with the IBC method for the model problem #1 as a function of the number of Fourier modes N_M (dash line, N_T=100) and as a function of the number of Chebyshev polynomials N_T (solid line, N_M=15) used in the computations.*

Algorithms based on the dynamic grid adjustments and/or mappings require an effort equivalent to evaluation of the complete coefficient matrix **L** at each time step while the IBC algorithm requires evaluation of only matrix **K** which represents a small portion of the complete matrix **L**. The additional cost associated with the evaluation of the complete matrix can be illustrated using the domain transformation (DT) method discussed in the context of the test problem #1 as a representative of the methods requiring grid adjustment. Figure 6.22 shows the ratio of time required to construct the coefficient matrix for the DT and the IBC methods using different numbers of Fourier modes N_M and Chebyshev polynomials N_T. The IBC method requires approximately 30 times less time for the smaller values of N_M and N_T and the savings increase almost linearly with the increase of either N_M or N_T. The cost of the matrix construction further significantly increases (relative to the IBC method) if the grid adjustment has to rely on the numerical grid generation.

Equation (6.4.22) can be solved directly using various algorithms, with potential efficiency gains associated with taking advantage of the sparse character of the coefficient matrix. We have tested four different methods in an effort to find an efficient solution strategy.

vi) *Method A:* This method relies on the LU decomposition. The lower and upper triangular matrices are computed at the beginning of each time step and the subsequent iterations are reduced to a simple backward eliminations and forward substitutions in a repetitive fashion.

vii) *Method B:* The solution process for this method is also based on LU decomposition, however, it takes advantage of the sparse structure of the coefficient matrix **L** and uses specialized solver for computing LU factors of sparse matrices.

viii) *Method C*: The inverse of the coefficient matrix **L** is computed once at the beginning of each time step and the subsequent iterations are reduced to the multiplication of the inverted matrix with the updated right hand side vector.

ix) *Method D:* Similar to method C this method computes the inverse of the coefficient matrix using specialized procedures for sparse matrices.

Method A is used as the reference method in the following discussion. The " performance gain" achieved by other methods can be quantified as the ratio of computational time involved in advancing solution by one time step using method in question to that required by method A. Figure 6.23 illustrates performance gains of methods B, C and D. It can be seen that method B is the most efficient while method D is the least efficient. The efficiency of methods B and C depends very little on the problem size, while the efficiency of method B increases almost linearly from ~1.15 to ~1.5 when increasing the number of blocks of the same size (Fig. 6.23a). The efficiency of this

method is almost unchanged when one works with a constant number of blocks of increasing size (Fig. 6.23b).

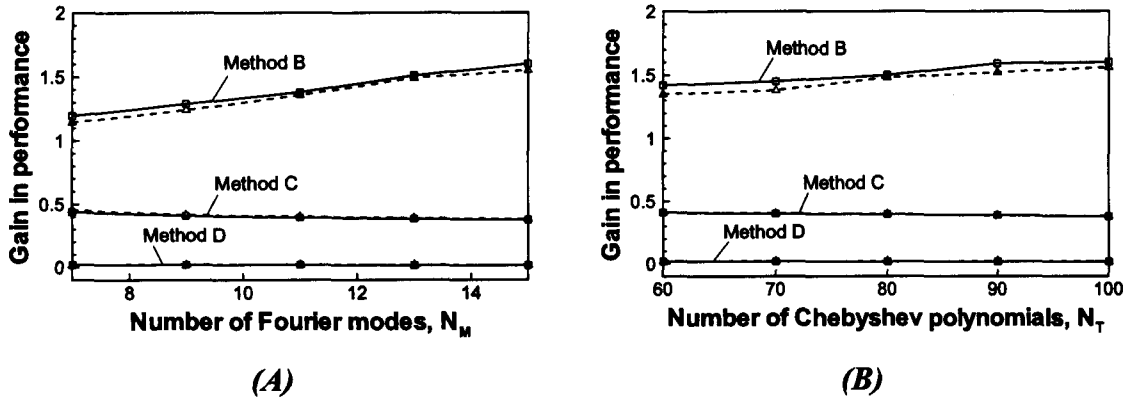


Figure 6.23. *Variations of the performance gains associated with the use of methods B, C and D for solution of the linear problem as a function of the number of Fourier modes N_M (Fig. 6.23A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 6.23B, N_M=15) used in the computations. The tests have been carried out for the model problem #2 using the third-order implicit discretization with α=1.0, S=0.05, Re=100 and c=π. The solid and dashed lines correspond to results obtained using time steps Δt=0.001 and Δt=0.01, respectively.*

When a large number of Fourier modes and/or Chebyshev polynomials are used, the resulting matrix could be very large leading to the computationally unacceptable solution cost, in spite of a minimal cost associated with the generation of matrix **L** at each time step. The solution cost can be reduced using iterative solver based on the mode decoupling concept [22, 23, 26]. It had been demonstrated that such solvers lead to a significant acceleration of computations in the case of flows governed by linear operators, with further efficiency gains possible through parallelization as the mode decoupling method is very suitable for applications on distributed processors. On the negative side, the iterative algorithm limits the range of geometries that can be accessed due to convergence problems, as documented in the context of linear operators [22, 23, 26]. The convergence problems are more severe in the case of nonlinear operators, which further limit the range of geometries that can be modeled.

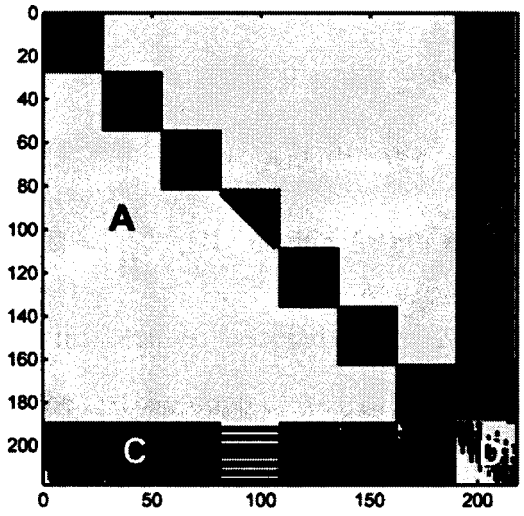


Figure 6.24. *Structure of the modified coefficient matrix for the model problem #2 constructed using N_M=3 Fourier modes and N_T=30 Chebyshev polynomials (see Eqs 4.23-4.24). Non-zero entries are marked in black.*

Because limitations associated with the iterative solver deemed to be too constraining, a new direct solver has been developed following concepts described in [26] where advantage was taken of the fact that part **H** of the matrix **L** does not change during time advancement. Part of this matrix can be inverted once and the overall solution can be reduced to a solution of a much smaller algebraic system supplemented by multiplication by the inverted matrix. The largest square matrix that can be extracted from **H** has the size q×q. For that purpose, the vector of unknowns has been re-organized by re-locating the first four coefficients of Chebyshev expansions for each Fourier mode to the end of the vector resulting in the structure of coefficient matrix illustrated in Fig. 6.24. Square matrix **A** of size q×q has block diagonal structure with each block of size $(N_T\text{-}3)\times(N_T\text{-}3)$, rectangular matrix **B** of size q×(p-q) has block diagonal structure with each block of size $(N_T\text{-}3)\times4$, full rectangular matrix **C** has size (p-q)×q and the square matrix **D** has size (p-q)×(p-q). Matrices **A** and **B** remain unchanged during time advancement while **C** and **D** need to be recomputed. The system can be written as

$$\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2 = \mathbf{R}_1, \qquad \mathbf{C}\mathbf{x}_1 + \mathbf{D}\mathbf{x}_2 = \mathbf{R}_2, \qquad (6.4.23a,b)$$

where vector $\mathbf{x}_1$ contains unknowns $Z_k^{(n)}$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 4, N_T \rangle$, and $\mathbf{x}_2$ contains unknowns $Z_k^{(n)}$ for $n \in \langle -N_M, N_M \rangle$, $k \in \langle 0,3 \rangle$. The right hand side vector remains unchanged with $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix}$, $\mathbf{R}_1$ having length q and $\mathbf{R}_2$ having length (p-q). Matrices **B** and **D** contain information associated with $Z_0^{(n)}$, $Z_1^{(n)}$, $Z_2^{(n)}$ and $Z_3^{(n)}$, while matrices **A** and **C** contain the rest. Solution of (6.4.23) can be written as

$$\mathbf{x}_2 = \mathbf{E}^{-1}\mathbf{R}_2 - \mathbf{E}^{-1}\mathbf{C}\mathbf{A}^{-1}\mathbf{R}_1, \qquad \mathbf{x}_1 = \mathbf{A}^{-1}\mathbf{R}_1 - \mathbf{A}^{-1}\mathbf{B}\mathbf{x}_2, \qquad (6.4.24a,b)$$

where $\mathbf{E} = (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ has the size (p-q)×(p-q). Matrices $\mathbf{A}^{-1}$ and $\mathbf{A}^{-1}\mathbf{B}$ need to be computed only once and remain unchanged during time advancement; the main computational effort at each time step is reduced to solving a system of equation of the size (p-q)×(p-q) resulting in method E.

v) *Method E*: Method based on Eqs (6.4.23)-(6.4.24) where all operations equivalent to evaluation of the inverse matrices are carried out using LU decomposition.

Figure 6.25 shows that performance gains vary approximately from 1.5 to 5 depending on the severity of geometry and problem size, and are approximately the same regardless if one uses more smaller blocks (more Fourier modes, Fig. 6.25a) or works with fewer but larger blocks (more Chebyshev polynomials, Fig. 6.25b). The reader should note that an increase of severity of boundary motions (larger amplitude S of boundary motions) increases the number of iterations required per time step. Since the largest performance gains are associated with the first iteration, the performance gains for the more extreme wave profiles are less pronounced. In a similar way, use of smaller time steps reduces advantage of this method as it decreases the number of iterations per time step.

Method E can be further improved by taking advantage of the block-diagonal structure of matrix **A**, which results in method F.

vi) *Method F*: The inverses $\mathbf{A}^{-1}$ and $\mathbf{A}^{-1}\mathbf{B}$ are computed by inverting individual blocks (rather than inverting the whole matrices) using the LU decomposition.

Figure 6.25 illustrates that performance gains associated with method F vary from 4 to 12 depending on the severity of geometry and problem size, similarly as in the case of method E. These gains increase rapidly with an increasing number of Fourier modes and Chebyshev polynomials, and are approximately the same regardless if one uses more smaller blocks (more Fourier modes, Fig. 6.25a) or works with fewer but larger blocks (more Chebyshev polynomials, Fig. 6.25b). Similarly to method E, this method performs slightly better for the less severe geometries and for smaller time step sizes, which result in a fewer iterations per time step.

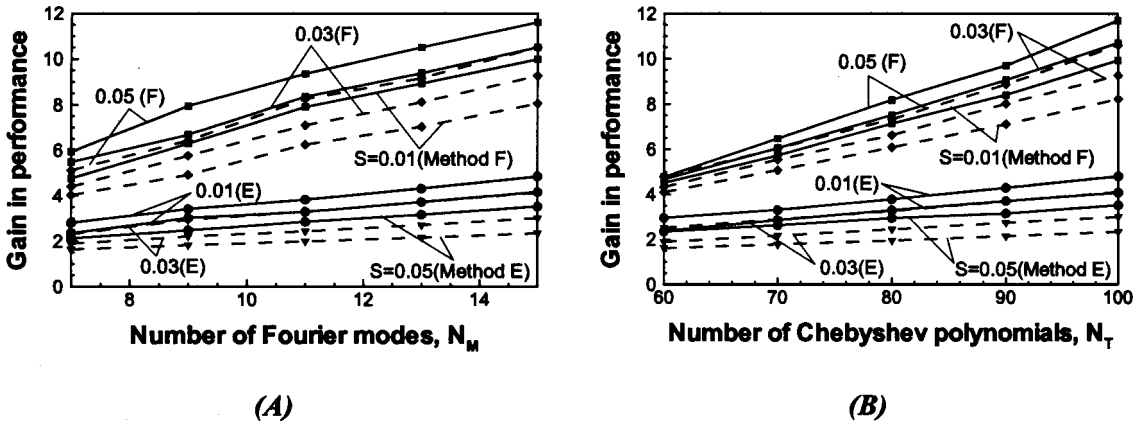


Figure 6.25. *Variations of the performance gains associated with the methods E (lower curves) and F (upper curves) as a function of the number of Fourier modes N_M (Fig. 6.25A, N_T=100) and as a function of the number of Chebyshev polynomials N_T (Fig. 6.25B, N_M=15) used 9in the computations. The tests have been carried out for the model problem #2 using the third-order implicit discretization with α=1.0, Re=100, $c=\pi$, and different values of S. The solid and dashed lines correspond to results obtained using time steps Δt=0.001 and Δt=0.01, respectively.*

It is of interest to summarize relative advantages of the IBC method as compared with the methods based on the dynamic grid adjustment represented in this discussion by the DT method. The IBC method requires the construction of the **H** matrix only once while the elements corresponding to the relatively smaller **K** matrix need to be computed at

every time step. In contrast, the complete coefficient matrix needs to be computed at each time step for the DT method at an additional cost already discussed at the beginning of this section. Here the reader should note that results presented in Fig. 6.22 compare costs of construction of the complete matrices for both methods while in the actual implementation a much smaller matrix K needs to be computed at each time step for the IBC method. Use of method F for solving the resulting linear system further increases the relative gains of the IBC method. The DT method leads to the full coefficient matrix and the cost of solution of the linear system is similar to the cost associated with method A of the IBC implementation; the relative gains of IBC methodology resulting from the use of specialized solver (method F) are illustrated in Fig. 6.25.

Memory requirement provides a second limitation for the applicability of the algorithm. The most common implementations of the IBC algorithm of the type discussed here rely on the construction of the complete matrix L [21-23] resulting in extreme memory requirements when a large number of Fourier modes and Chebyshev polynomials need to be used. Method F requires storage for matrices **B, C, D** and only for the diagonal blocks of matrix **A**. The memory requirement can be further reduced by noting that the information about the diagonal blocks can be overwritten with the information about the inverted blocks. Figure 6.26 illustrates variations of the ratio of the memory use by methods A and F. It can be seen that in the case of N_M=25 Fourier modes and N_T=120 Chebyshev polynomials method F uses about 20 times less memory than method A. The memory gains increase almost linearly and very rapidly when the number of Fourier modes N_M increases (resulting in a larger number of blocks). The memory gains change very little as the number of Chebyshev polynomials N_T increases (resulting in larger blocks - see Fig. 6.26).

The final comment deals with the memory use associated with the methods based on the dynamic grid adjustment represented in this discussion by the DT method. This method leads to the full matrix and offers effectively no potential for memory savings.

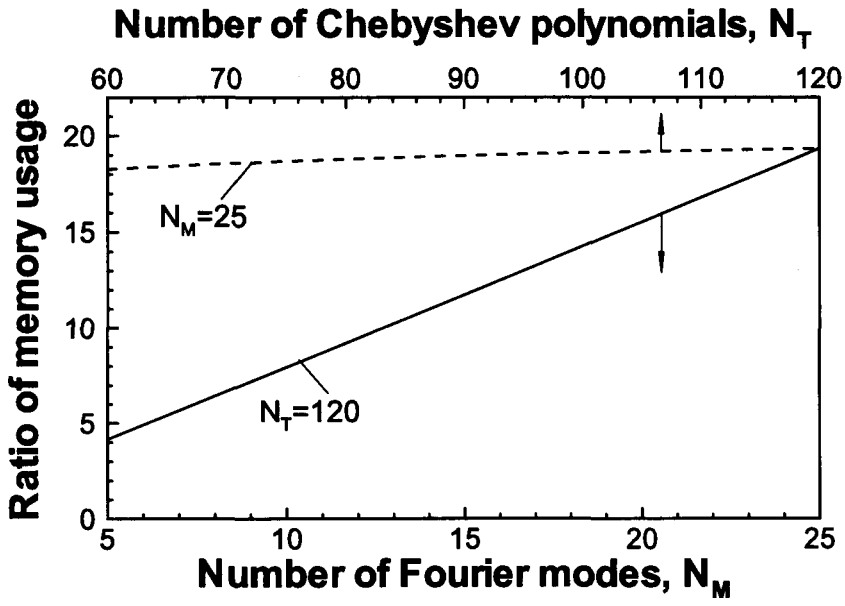


Figure 6.26. *Variations of the ratio of memory used by methods A and F as a function of the number of Chebyshev polynomials N_T (dash line, N_M=25) and as function of the number of Fourier modes N_M (dash line, N_T=120) used in the computations.*

6.5. Conclusions

A highly accurate algorithm to analyze unsteady flow problems associated with the presence of moving boundaries has been presented. The algorithm relies on the concept of immersed boundary conditions where the time-varying flow domain remains completely immersed inside a fixed computational domain during simulation time. The flow boundary conditions are imposed in the form of constraints. The algorithm uses Fourier expansions in the stream-wise direction, Chebyshev expansions in the wall-normal direction and fully-implicit time discretization with up to fourth-order accuracy. Various tests demonstrate that the algorithm delivers spectral accuracy in space and the theoretically predicted accuracy in time. The entries in the coefficient matrix corresponding to the field equations remain unchanged during boundary motions due to the use of the immersed boundaries concept. As a result, only a small part of the coefficient matrix corresponding to boundary constraints needs to be recomputed at each

time step resulting in a significant saving of computing resources. A very efficient linear solver that takes advantage of the structure of the coefficient matrix has been proposed. The proposed solver results in a significant acceleration of the computations as well as in a substantial reduction of memory requirements.

6.6. References

[1] Floryan, J.M. and Rasmussen, H., Numerical analysis of viscous flows with free surfaces, *Appl. Mech. Rev.*, vol. 42, pp. 323-341 (1989).

[2] Scardovelli, R. and Zaleski, S., Direct numerical simulation of free surface and interfacial flow, *Annu Rev. Fluid Mech.*, vol. 31, pp. 567-603 (1999).

[3] Hyman, J.M., Numerical methods for tracking of interfaces, *Physica,* vol. 12D, pp. 396-407 (1984).

[4] Glimm, J., Grove, J.W., Li, X.L., Shyue, K.M., Zeng, Y. and Zhang, Q., Three-dimensional front tracking, *SIAM J. Sci. Computing,* vol. 19, pp. 703-727 (1998).

[5] Hirt, C.W. and Nichols, B.D., Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *J.Comp.Phys.*, vol. 39, pp. 201-225 (1981).

[6] Harlow, F.H. and Welch, J.E., Numerical study of large amplitude free surface motions, *Phys.Fluids*, vol. 9, pp. 842-851 (1966).

[7] Osher, S.J. and Sethian, J.A., Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comp .Phys.*, vol. 79, pp. 12-49 (1988).

[8] Sethian, J.A. and Smereka, P., Level set methods for fluid interfaces, Annu. *Rev. Fluid Mech.*, vol. 35, pp. 341-372 (2003).

[9] Inculet, I., Floryan, J.M. and Haywood, R., Dynamics of water droplets break-up in electric fields, *IEEE Transactions on Industry Applications*, vol. 28, pp. 1203-1209 (1992).

[10] Hamed, M. and Floryan, J.M., Numerical simulation of unsteady nonisothermal capillary interfaces, *J.Comp.Phys.* vol. 145, pp. 110-140 (1998).

[11] Peskin, C.S., The fluid dynamics of heart valves: experimental, theoretical and computational methods, *Annu. Rev. Fluid Mech.*, vol. 14 pp. 235-59 (1982).

[12] Peskin, C.S., The Immersed Boundary Method, *Acta Numerica*, 2002; 479-517.

[13] Mittal, R. and Iaccarino, G., Immersed boundary methods, *Annu. Rev. Fluid Mech.*, vol. 37 pp. 239-261 (2005).

[14] Deng, J., Shao, X.M. and Ren, A. L., A new modification of the immersed-boundary method for simulating flows with complex moving boundaries, *Int. J. of Num. Meth. Fluids*, vol. 52, pp. 1195-1213 (2006).

[15] Taira, K. and Colonius, T., The immersed boundary method: A projection approach, *J. Comp. Phys.*, vol. 225, pp. 2118-2137 ((2007)).

[16] Kim, J., Kim, D. and Choi, H., An immersed-boundary finite-volume method for simulation of flows in complex geometries, *J. Comp. Phys.*, vol. 171, pp. 132-150 (2001).

[17] Marella, S., Krishnan, S., Liu, H., and Udaykumar, H.S., Sharp interface Cartesian grid method I: An easily implemented technique for 3d moving boundary computations, *J. Comp. Phys.*, vol. 210, pp. 1-31 (2005).

[18] Szumbarski, J. and Floryan, J.M., A direct spectral method for determination of flows over corrugated boundaries, *J. Comp. Phys.*, vol. 153, pp. 378-402 (1999).

[19] Floryan, J.M., Centrifugal instability of Couette flow over a wavy wall, *Phys. Fluids*, vol. 14, pp. 312-322 (2002).

[20] Floryan, J.M., Three-dimensional instabilities of laminar flow in a rough channel and the concept of hydraulically smooth wall, *Eur. J. Mech. B/Fluids*, vol. 26, pp. 305-329 (2007).

[21] Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for unsteady flow problems described by the Laplace operator, *Int. J. Num. Meth. Fluids*, vol. 56, pp. 1765-1786 (2007).

[22] Husain, S.Z. and Floryan, J.M., Implicit spectrally-accurate method for moving boundary problems using immersed boundary conditions concept, *J. Comp. Phys.*, vol. 227, pp. 4459-4477 (2008).

[23] Husain, S.Z. and Floryan, J.M, Gridless spectral algorithm for stokes flow with moving boundaries, *Comp. Meth. App. Mech. Eng.*, vol. 198, pp. 245-259 (2008).

[24] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., Spectral methods in fluid dynamics, *Springer* (1987).

[25] Husain, S.Z., Floryan, J.M. and Szumbarski, J., Over-determined formulation of the immersed boundary conditions method, *Comput. Methods Appl. Mech. Engrg.* doi: 10.1016/j.cma.2009.09.022 (2009).

[26] Husain, S.Z. and Floryan, J.M, An Efficient Linear Solver for Problems Arising from the Spectral Implementation of the Immersed Boundary Conditions Method, *Expert Systems in Fluid Dynamics Research Laboratory Report ESFD-2/2008*, Department of Mechanical and Materials Engineering, The University of Western Ontario, London, Ontario, N6A 5B9, Canada (2008).

CHAPTER 7

General discussion, applications and conclusions

7.1. Discussion

The research work presented in this dissertation has been motivated by the following three principal objectives:

i. Developing implicit, spectrally-accurate algorithms to simulate unsteady heat and fluid flow problems described by the most commonly encountered operators in the field of computational fluid dynamics in the presence of fixed and moving boundary irregularities.

ii. Exploring options to expand the range of applicability of the algorithm in terms of the geometric complexities that it can handle with an acceptable accuracy, as well as increasing the level of accuracy for a given boundary geometry.

iii. Enhancing the efficiency of the algorithm by improving the computational speed and reducing the memory requirements in order to render the IBC algorithm as a highly effective tool for analysis of two- and three-dimensional fixed and moving boundary problems.

Each of the above issues is discussed in sequence in Section 7.1.1, 7.1.2 and 7.1.3. The three-dimensional formulation is discussed in Section 7.2. The advantages provided by the IBC algorithm for analyses of physical problems are presented in Section 7.3 using

suitable examples. Finally, a short summary of the main conclusions arising from this dissertation is presented in Section 7.4.

7.1.1. Development of implicit, spectrally-accurate algorithms

The main goal of my doctoral research has been to develop an implicit spectrally-accurate algorithm for solving the Navier-Stokes equations in the presence fixed or moving boundary irregularities. The Laplace and the biharmonic operators are the core components for these partial differential equations. Therefore, investigations were first carried out in the context of flow problems involving the Laplace operator and the relevant results are presented in Chapter 2. The Laplace operator constitutes a convenient starting point due to its simplicity from both physical and numerical perspectives. However, the Laplace operator also governs several physically interesting flow problems, e.g., conductive heat flow, ground water hydrology, potential flow, etc. Later in Chapter 3, the IBC algorithm was extended to solving the Stokes flow with moving boundaries where the field equation is described by the biharmonic operator. In addition to being an important constituent of the Navier-Stokes problem, the biharmonic operator describes creeping flow, which is encountered in various micro-channel flows as well as a number of problems in the theory of elasticity. The investigation for two-dimensional flow problems were completed with the application of the algorithm for two-dimensional Navier-Stokes problem involving moving boundaries with the corresponding results and discussion presented in Chapter 6. Besides differences from the physical perspective, the model problem described in Chapter 2 has an important difference from the standpoint of boundary conditions with the problems presented in Chapters 3 and 6.

The two-dimensional Stokes and Navier-Stokes problems have been formulated using the stream function, thereby eliminating the pressure terms and for that reason any complication associated with the treatment of the pressure terms is removed. The pressure field is evaluated as a part of the post-processing once the solution for the stream function is obtained. The procedure for evaluating pressure is presented in

Appendices C and K in the context of the Stokes and Navier-Stokes problems, respectively. The completion of formulations for the Stokes and Navier-Stokes problem require additional constraints, either in the form of constant volume flux or constant pressure gradient. For convenience, the constant volume flux constraint has been selected for the problems presented in Chapters 3 to 6. However, the form of the boundary constraints for the constant pressure gradient condition is presented in Appendix D in the context of the Stokes problem. Similar constraint for the Navier-Stokes problem can be easily derived by comparing Appendix K with Appendix D.

Modeling of the boundary geometries is a core part of the immersed boundary conditions algorithm. The geometries, whether fixed or time-dependent, are modeled in the IBC algorithm using Fourier expansions. The algorithm only admits shapes of boundaries that can be expressed in terms of Fourier series thereby narrowing the type of geometries that are accessible. Nevertheless, a wide range of geometries can still be investigated and more importantly, the definitions of the geometries are reduced to defining the Fourier coefficients only thereby resulting in substantial simplification of the programming efforts. Therefore, the benefits associated with such definition of geometries significantly outweigh the limitations.

The moving boundaries have been illustrated in the model problems by considering elastic traveling waves and elastic standing waves (see Chapters 2, 3 and 6). The selection of the elastic traveling waves allowed solving the model problems as stationary problems in a moving frame of reference. The steady solutions computed in the moving frame of reference are free from any error associated with temporal discretization and provide means to identify the error associated with the various temporal schemes applied to solve the corresponding unsteady moving boundary problems in the fixed frame of reference. Besides this, the solution in the moving frame was useful in characterizing the spatial discretization dispositions of the IBC algorithm in definitive and unambiguous manners. The different model problems, when solved in the moving frame of reference, also permitted to characterize the performance of the algorithm for fixed boundary problems from a general perspective.

Certain characteristics of the IBC algorithm, as revealed from its application in the various model problems considered for the present study, deserve an overall recapitulation. The algorithm works by using Fourier and Chebyshev expansions in the stream-wise and normal-to-the-wall directions respectively. The order of Chebyshev polynomials used to discretize in the normal-to-the-wall direction is found to crucially dictate the behavior of error in the solution. It has been observed that with adequate number of Chebyshev polynomials used in the discretization, the maximum error in the solution field is shifted to locations around the irregular boundaries (see Fig. 6.3, 6.6 and 6.7). Therefore, a strict and definitive measure of accuracy for the IBC algorithm can be constructed by using the L_∞-norm of the error in the enforcement of the boundary conditions. This allowed for convenient studying of the error behavior for the algorithm by using the constructed error norm. One can conveniently estimate the adequate number of Chebyshev polynomials as any further increment in the order of the polynomials beyond this number will not change the solution field by any meaningful degree. The adequate number of Chebyshev polynomials required to discretize the field equation as well as the boundary conditions is dictated by the magnitude of the wave number of the model geometries. The distributions of the modal functions associated with the higher Fourier modes used in the discretization exhibit boundary layer like structure around the irregular boundaries (see Fig. 3.3 and 6.4). Increase of the corrugation wave number results in a decrease of the thickness of these boundary layers which, therefore, require the use of a larger number of Chebyshev polynomials for their proper resolution. Chapters 3 and 6 give estimates for the adequate number of Chebyshev polynomials that need to be used for different ranges of the corrugation wave numbers for the associated model problems.

Chebyshev norms are used to measure the contribution of the different Fourier modes for all the model problems. Exponential decrease of the Chebyshev norms of the truncated Fourier expansions, as observed for all the model problems, reveal the spectral convergence characteristic of the algorithm in the x-direction (see Fig. 3.4 and 6.5)

The algorithm in its classical form is found to provide spectrally-accurate solution when the geometric irregularities are below some critical threshold characterized by the wave number and the wave amplitude of the corrugated geometric shapes investigated in different Chapters (see Fig. 2.4, 2.5, 3.5, 3.6, 6.9 and 6.10). As expected, the critical combinations of the wave number and the wave amplitude are found to be the highest for the Laplace operator and the lowest for the Navier-Stokes operator. For the particular case of the Navier-Stokes operator the critical threshold is found to decrease with increase in the value of the Reynolds number. These observations are consistent with intuitive expectations.

The error in the enforcement of the boundary conditions is found to vary in an oscillatory manner with the maximum of the error found to coincide approximately with the location of the maximum channel/slot opening for all the model problems (see Fig. 2.6, 2.7, 2.13, 3.7, 3.8, 3.14, 6.8, 6.11 and 6.17). For the moving boundary problems, the algorithm reproduced this particular characteristic at any instant of time in accordance with the shape of the boundaries at that time. The periodic variations in the solution field as a function of time have been faithfully computed by the algorithm for all the model problems along with the associated temporal phase shift for solutions at points having different spatial locations (see Fig. 2.8, 3.9 and 6.12).

The spectral composition of the error in the enforcement of the boundary conditions for the classical formulation have shown that the solution vectors did not contain any harmonic of order lower than the number of Fourier modes used in the definition of the solution (see Fig. 2.14, 3.15 and 6.18). In order to check whether the algorithm produces any spurious spatial oscillation, tests have been designed that admit sub-harmonics for the different model problems by representing the boundary locations with different Fourier modes and accounting for the related changes in the length of the computational box. The solutions have been found to produce no sub-harmonics and the modes expected to contribute zero contributions behaved accordingly (see Fig. 2.14, 3.15 and 6.18).

Up to fourth-order accurate, implicit, backward-difference schemes have been investigated for temporal discretization purposes. The different temporal schemes have been found to provide the theoretically predicted accuracy for all the model problems considered (see Fig. 2.10, 2.15, 3.11, 3.16 and 6.20).

7.1.2. Improvement of accuracy and applicability of the algorithm

For the classical formulation of the IBC method, it has been observed that for any model problem there exists a critical combination of the wave number and the wave amplitude of the boundary geometry beyond which any further increment of the number of Fourier modes used to define the solution fails to improve the accuracy (see Fig. 4.3 and 5.13). The primary motivation behind the research work presented in Chapter 4 was to increase the critical wave number for a given value of the wave amplitude and vice versa. Another objective was to improve accuracy for a given number of Fourier modes used in the computations for any geometry. On the other hand, the part of the work presented in Chapter 5 associated with the over-determined formulation, was primarily motivated by the quest for improved the computational efficiency of the solution process.

Diagnosing the cause that limits the accuracy and applicability of the classical form of IBC algorithm is crucial for making any progress in improving the algorithm. As has been discussed in Chapter 5, the discretization of any model problem in the periodic direction in the context of the IBC algorithm is based on two different Fourier expansions. The first expansion is used to discretize the field equation of the model problem and the second one is related to the discretization of the boundary conditions. The discretized field equations do not offer any coupling among the different Fourier modes for the model problems except the Navier-Stokes equations where additional coupling comes from the nonlinear terms. Nevertheless, for all the model problems, the field equations in their discretized forms do not contain any information pertinent to the boundary geometries. Therefore, with increasing complexity of the boundary shapes for any model problem, the rate of convergence of the Fourier series related to the boundary

conditions slows down compared to the convergence of the Fourier series for the field equations. However, in the case of the Navier-Stokes problem, the rate of convergence of the Fourier expansions used to discretize the field equations may become an issue particularly for large values of the Reynolds number Re as the coupling influence of the nonlinear terms may overwhelm the coupling influence of the boundary conditions.

A possible remedy to the limitations of the classical IBC formulation, as investigated in Chapter 4 and parts of Chapter 5, is to increase the number of terms in the Fourier expansions used for discretizing the boundary conditions. The discretization of the boundary conditions by the IBC algorithm leads to a number of discretized boundary constraints that are far in excess than those required to form a close system of equations. In the classical formulations the boundary constraints are incorporated in a Tau like manner where only a certain numbers of discretized boundary constraints are retained that result in a closed system of equations. Use of additional boundary constraints implicitly implies using additional terms in the Fourier expansions for the boundary conditions, which is expected to contribute to better convergence characteristic of the truncated Fourier series for the boundary conditions. Incorporation of additional boundary constraints, however, leads to an over-determined formulation and the corresponding discretized set of algebraic equations can be solved only in the least squares sense.

The principal step in solving the over-determined problem is based on the computation of the pseudo-inverse of the over-determined coefficient matrix. In order to have a complete picture with regard to the effectiveness of the over-determined formulation, three model problems described by the Laplace operator, the biharmonic operator and the Navier-Stokes operator have been examined in Chapter 4. Only steady, fixed boundary versions of the model problems are considered for the convenience of testing. The solution approach in Chapter 4 is based on computing the pseudo inverse of the complete over-determined coefficient matrices. Three different methods for evaluating the pseudo-inverse have been examined in Chapter 4, namely, the method of normal equations, the

QR factorization and the singular value decomposition (SVD). The performances of all three methods were found to be equivalent, particularly around the critical zone.

The pseudo-inverses in Chapter 4 were computed for the complete coefficient matrix which for the Stokes and Navier-Stokes operator has been found to be prone to numerical rank-deficiency. Rank deficiency is found to originate from the difference in the order of magnitudes of the elements in the coefficient matrices originating from the discretized field equations and the boundary constraints. Such rank-deficiency problem suggested the use of SVD for computing the pseudo-inverse for its reliability. Besides this, a method of weighted residual has been introduced, where the discretized boundary conditions are multiplied by a weight factor, which permits to control the influence the boundary conditions can exert on the complete solution process. The best solution in this method corresponds to the optimum weight factor, which is defined as the smallest weight factor for which the numerical rank-deficiency of the coefficient matrix is eliminated.

The solution of the over-determined problem by computing the inverse of the complete coefficient matrix as pursued in Chapter 4 is found to improve the accuracy for Laplace and Stokes problem by at least two orders of magnitude for boundary shapes around the critical zone (see Fig. 4.3-4 and 4.11). The range of applicability of the algorithm is also found to be significantly expanded as compared with the classical formulation of the IBC method for these model problems. The investigation of the spectral composition of the boundary error have shown that unlike the classical method the error for the over-determined method is spread over all the Fourier modes involved in the computations resulting in lowering of the absolute value of the error and consequently improving the accuracy (see Fig. 4.6 and 4.13). However, the performance of the over-determined method in the case of the Navier-Stokes equations is found to depend on the Reynolds number Re (see Fig. 4.16). The performance of the algorithm for low Re is observed to be similar to the Stokes problem, while increase in Re is found to diminish the accuracy and applicability gain from the over-determined formulation although problems with numerical rank deficiency decreases (see Fig. 4.15). Such behavior of the algorithm for

the Navier-Stokes problem is attributable to the dominating role that the nonlinear terms start to play with increase of Re, which results in slower convergence rate of the Fourier expansions associated with discretization of the field equation as compared to the Fourier expansions related to the boundary conditions.

Although the IBC formulation gives access to a large pool of additional boundary constraints it has been observed that increasing the number of constraints arbitrarily does not improve the accuracy in the same manner (see Fig. 4.10 and 4.15). In fact, in some cases using too many additional constraints have been found to be disadvantageous for the solution approach considered in Chapter 4. The retention of additional projection equations led to another version of over-determined formulation in Chapter 4 that have been found to bring no meaningful improvement or degradation of the solution. Therefore, during the tests associated with Chapter 5 such version has not been attempted.

Chapter 5 primarily explores a different approach in solving the over-determined system of discretized algebraic equations resulting from the over-determined formulation in order to improve the computational performance of the algorithm. For convenience of investigation, only steady Stokes flow in the presence of fixed boundaries has been considered in the chapter. In the new method, the discretized field equations are satisfied exactly while the boundary constraints are solved in a least squares sense without relying on any weight factor. This approach saves a substantial amount of computations that is required by the direct method described in Chapter 5 which needs to determine the optimum weight factor. The SVD and QR factorization techniques have been used to compute the relevant pseudo-inverses in this approach. The solvers using this technique are referred to as the specialized over-determined solvers. The specialized solvers reduce the size of the effective matrix for which the pseudo-inverse needs to be evaluated and this results in a substantial reduction in computational time.

A brief discussion pertinent to the improvement of the computational efficiency observed for the specialized solvers is going to be presented in Section 7.1.3, while in the current

section we shall focus on the performance of the new approach from the accuracy point of view. For any given number of Fourier modes, the specialized over-determined solver based on SVD is found to be more accurate than any other classical or over-determined solver (see Fig. 5.16). Moreover, the specialized SVD based solver gives access to larger values of Chebyshev polynomials and Fourier modes as it requires substantially less amount of memory compared to the direct solvers and therefore, is found to significantly increase the range of applicability of the IBC method compared to the classical formulation of this method. Depending on the wave number of the boundary irregularities, this solver is found to extend the applicability of the algorithm by 50%-70% in terms of the increment in geometric severity defined by the maximum amplitude of the irregularities that can be accessed for different values of the wave numbers (see Fig. 5.21 and 5.22).

Spectral implementations of the domain transformation (DT) method have been used to compute machine accurate reference solutions for all the model problems presented in Chapter 4. The DT method works by analytically mapping an irregular physical domain into a regular computational domain and its implementations for the different model problems considered in Chapter 4 are shown in Appendices E, F and G. This method can be viewed as a representative of methods based on the construction of boundary conforming grids where the construction is done analytically rather than through a numerical grid generation.

7.1.3. Enhancement of computational efficiency

The computational efficiency of an algorithm is one of the most important factors that need to be considered when one is assigned to select an algorithm for numerical simulation of flow problems in the presence of fixed and moving irregularities. Chapter 5 presents a systematic study of the efficiency of the various solvers employed to solve the system of equations arising from the implementation of the classical and over-determined

implementations of the IBC algorithm for steady and unsteady problems in the presence of fixed boundary irregularities.

The model problem selected in Chapter 5 is the Stokes flow. Both steady and unsteady versions of the model problem have been considered. The steady-state model problem has been solved using the spectral implementation of the domain transformation (DT) method to compare the relative efficiencies of the various IBC solvers with respect to the DT method. Besides this, machine accurate solutions computed by the DT method were also tremendously useful in investigating the various over-determined IBC solvers. Gauss-Seidel and Jacobi iterative solvers have also been explored for the classical IBC formulation of the steady problem. The iterative solvers have been found to perform better than the standard direct solvers where the complete system of equations is solved directly (see Fig. 5.6). However, the efficiencies of the iterative solvers have been found to suffer significantly with increasing difficulty in the geometry of the boundaries eventually leading to failure to satisfy the desired convergence criterion (see Fig. 5.7).

A new class of solvers, termed as specialized direct solvers, has been developed that takes advantage of the special structure of the complete coefficient matrix resulting from the implementation of the IBC algorithm. The specialized solvers work by extracting the largest possible square matrix from the complete coefficient matrix. This largest square matrix happens to only contain information relevant to the discretized field equations and have a block diagonal structure that can be inverted in a block-by-block manner. This results in a substantial reduction in the computational overhead. The efficiency of the specialized solvers has been found to increase with the increasing severity of the boundary geometries as larger number of Fourier modes are needed for solving such problems. For example, the most efficient version of the specialized direct solver is found to be approximately 50 times faster than the standard direct solver for 30 Fourier modes and 120 Chebyshev polynomials used in the computations (see Fig. 5.9). It has been possible to reduce the computations for the specialized solvers further by considering the complex conjugate properties that exist among the different blocks.

Even the standard direct IBC solvers are found to be more efficient than the domain transformation (DT) method although the performance differences between them diminish with increasing geometric severity (see Fig. 5.5). This is so because the increase in geometric severity necessitates use of larger number of Fourier modes for the standard IBC solver as compared to the DT method in order to achieve the desired accuracy. In contrary, the efficiencies of the specialized IBC solvers are found to increase as compared with the DT method with increasingly difficult geometries irrespective of the level of desired accuracy in the solution (see Fig. 5.11). The concept behind the specialized solver cannot be extended to the DT method due to strong coupling that exist among the different Fourier modes in the discretized system of algebraic equations resulting from the transformed field equation.

The concept of specialized solvers is then extended to the over-determined formulation of the IBC algorithm. The performance gain obtained by the specialized direct over-determined solvers as compared against the standard direct over-determined solver is found to increase very rapidly and reaching the order of approximately 200 or more when the number of Fourier modes in the solution is set at 25 or beyond (see Fig. 5.17). This substantial reduction in the computational time as experienced by the specialized over-determined solvers make them tremendously useful in improving the accuracy and applicability of the IBC algorithm for practical purposes.

The concept of the specialized direct solvers is later applied to solve unsteady fixed boundary problems subject to constant as well as time-dependent boundary conditions in the Section 5.5. The specialized solvers are found to be substantially faster than the standard solvers for unsteady simulations (see Fig. 5.24).

The spectral methods are chosen widely for direct numerical simulation (DNS) of flows involving straight boundaries. The main reason behind the popularity of the spectral algorithms for DNS is the high spatial accuracy these algorithms can deliver. Another reason making the spectral methods feasible for DNS of flows through channels with straight walls is the fact that the discretized field equations along with the boundary

conditions for the different modes are decoupled with the only coupling originating from the nonlinear terms in the field equations that can be treated explicitly with different hybrid temporal discretization schemes [1]. The IBC algorithm coupled with specialized solvers has the potential for use in DNS of flow in channels with irregular shaped walls at a comparable cost.

The solution of the moving boundary problems through the application of any algorithm requires solving a system of equations at every time step. However, in the case of the IBC algorithm the system of equations originating form the discretized field equations remain unchanged and the corresponding part of the coefficient matrix need to be constructed only once. This inherent property of the IBC algorithm is extremely favorable for dealing with moving boundary problems as the cost of constructing the coefficient matrix at every time step is dictated only by the elements related to the boundary constraints. The concept of the specialized solvers as illustrated in Chapter 5 is extended to the moving boundary problems in Chapter 6 in the context of the two-dimensional Navier-Stokes equations. The specialized solvers as applied to the moving boundary problems enhanced the computational efficiency by a substantial margin (see Fig. 6.25).

In addition to the very high computational costs associated with the use of a large number of Fourier modes and Chebyshev polynomials, the related high memory requirements can restrict the severity of the geometric shapes that can be handled by the standard IBC solvers. For all model problems tested in the course of work for this dissertation the specialized solvers are found to reduce the memory requirements for the IBC algorithm by a substantial margin making the algorithm further attractive (see Fig. 5.12, 5.23 and 6.26).

7.2. Extension of the IBC algorithm to three-dimensional problems

Various model problems considered in the Chapters 2 through 6 dealt with two-

dimensional flow situations. Although significant physical information of practical interest can be extracted from the analyses of two-dimensional flows, in certain situations it may be essential to investigate three-dimensional flow problems for obtaining a complete picture.

7.2.1. Three-dimensional Navier-Stokes equations

The three-dimensional Navier-Stokes equations in terms of the wall-normal vorticity and wall-normal velocity derived using the methodology described in [2] have the form

$$\frac{\partial \eta}{\partial t} + u_0 \frac{\partial \eta}{\partial x} + \frac{\partial v_1}{\partial z}\frac{du_0}{dy} - \frac{1}{Re}\nabla^2 \eta = \langle N_1 \rangle, \tag{7.2.1.1}$$

$$\frac{\partial}{\partial t}\left(\nabla^2 v_1\right) - \frac{1}{Re}\nabla^4 \eta + u_0 \frac{\partial}{\partial x}\left(\nabla^2 v_1\right) + \frac{\partial v_1}{\partial x}\frac{d^2 u_0}{dy^2} = \langle N_2 \rangle, \tag{7.2.1.2}$$

where $\eta = \frac{\partial u_1}{\partial z} - \frac{\partial w_1}{\partial x}$ is the wall-normal component of vorticity and $\langle N_1 \rangle$ and $\langle N_2 \rangle$ are nonlinear terms whose explicit forms are given in Appendix M. The terms u_1, v_1 and w_1 correspond to the velocity modifications in the presence of surface irregularities and u_0 denotes the reference Poiseuille flow, i.e., $u_0(y)=1-y^2$.

A close look at the three-dimensional governing equations (7.2.1.1)-(7.2.1.2) reveals that the three-dimensional Laplace and biharmonic operators are the major constituents of the linear part of the equations. The two-dimensional versions of these operators have already been dealt with the IBC method in the preceding chapters and discretizations of the three-dimensional operators do not present any major conceptual difference. The IBC method for the three-dimensional problem proceeds by assuming the solution field to be periodic in the x- and z-directions and therefore, the unknown field variables η and v_1 are discretized using Fourier expansions in these directions. The resulting modal functions

should then be discretized in the y-direction using expansions based on Chebyshev polynomials. Later, use of different temporal discretization schemes along with the Galerkin projection technique will reduce the dimensionless field equations into a system of discretized algebraic equations in terms of the unknown coefficients of Chebyshev expansions for the different Fourier modes in the x- and z-directions.

The discretization of the no-slip and no-penetration conditions at the three dimensional bounding surfaces would also follow the same concept as described for the two-dimensional problems in the previous chapters. The first-step in this regard is to model the three-dimensional surface geometries. We shall assume the surface geometries to be periodic in the x- and z-directions and therefore, their shapes can be expressed using Fourier expansions as

$$y_L(x,z) = 1 + \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} H_L^{(n,m)} e^{i(n\alpha x + m\beta z)}, \qquad (7.2.1.3a)$$

$$y_L(x,z) = 1 + \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} H_L^{(n,m)} e^{i(n\alpha x + m\beta z)}, \qquad (7.2.1.3b)$$

where $H_{L,U}^{(n,m)} = H_{L,U}^{(-n,-m)^*}, H_{L,U}^{(n,-m)} = H_{L,U}^{(-n,m)^*}$, stars denote complex conjugates and subscripts L and U denote the lower and upper surfaces, respectively. Later the boundary conditions are discretized using the concept of immersed boundary conditions where distributions of u_1, v_1 and w_1 along the upper and lower surfaces are expressed in terms of the unknown coefficients of Chebyshev expansions of v_1 and η.

The nonlinear terms in the three-dimensional equations (7.2.1.1)-(7.2.1.2) have forms similar to those encountered in the two-dimensional problems (see Appendix M) and can be treated using the iterative technique described in Chapters 4 and 6. Other standard methods for treating the nonlinear terms are provided in [3].

In the following section we shall be discussing the different aspects of the three-dimensional implementation of the IBC algorithm in the context of a convenient model problem that only involves the Laplace operator. Later in Section 7.3.2, we shall see the application of the IBC method for solving three-dimensional velocity-vorticity formulation of the linear stability equations that involve both the three-dimensional Laplace and biharmonic operators.

7.2.2. Three-dimensional heat conduction problem

The materials to be presented in this section are extracted from the report [4]. The model problem selected in [4] deals with heat conduction in a three-dimensional medium. Some preliminary works relevant to [4] has been published in a peer-reviewed conference proceeding [5].

The physical process of conduction is described by the three-dimensional Laplace operator. The details on the modeling of three-dimensional geometries along with the discretization procedures for the field equation and the boundary conditions are presented in [4]. Section 7.2.2.1 presents discussion on the properties of the spatial and temporal discretization of the three-dimensional field equation and the boundary conditions with the use of the IBC method on the basis of the test results presented in [4]. Various issues related to the different solution techniques for the discretized system of equations as well as results on testing of their performances are discussed in Section 7.2.2.2.

7.2.2.1. Spatial and temporal discretization characteristics

Various test problems considered in [4] encompass all possible scenarios, including steady and unsteady problems in the presence fixed or moving boundary irregularities subject to constant and time-dependent boundary conditions. Characterization of the spatial discretization properties is achieved in [4] by considering a convenient steady

fixed boundary test problem having a flat lower surface and corrugated upper surface. The surface corrugation is described in terms of the corrugation amplitude as well as the corrugation wave numbers in the x- and z-directions. Various tests have shown that the error in the solution field is larger around the irregular upper surface. Error is shown to vary in an oscillatory manner along the upper surface and location of maximum error overlapped with the maximum slot openings. These observations with regard to the behavior of error for the three-dimensional problem are consistent with the characteristics of the IBC algorithm as observed in the context of different two-dimensional model problems presented in the preceding chapters. A L_∞-norm denoted by $\|\theta_{BC}\|_\infty$ measuring the maximum error in the enforcement of boundary conditions is used to study the accuracy characteristics of the algorithm. Tests have been conducted in [4] to illustrate the spectral convergence characteristics of the spatial discretizations and to substantiate the fact that the IBC algorithm for the three-dimensional problem is capable of enforcing the Fourier modes in the boundary constraints with appropriate values without producing any sub-harmonics or spurious error. For unsteady fixed boundary problems subject to time-dependent boundary conditions $\|\theta_{BC}\|_\infty$, i.e., the maximum error in the enforcement of the boundary conditions is observed to vary periodically as expected.

Test problems were also designed to investigate the IBC algorithm for conduction problem in slots with deforming upper surfaces and flat lower surfaces. Such test problems belong to the class of moving boundary problems and the deformations in the upper surface was simulated by considering traveling and standing elastic waves. The objectives of these test problems were to determine the ability and the efficiency of the algorithm to handle boundary motions. Besides solving the resulting problems directly as unsteady moving boundary problems in a fixed frame of reference the traveling wave problem is also solved as a steady problem in a moving frame of reference. The solution of the traveling wave problem in the moving frame of reference has been used to determine the error in the temporal discretization of the direct solution. Different tests have proved the capabilities of the algorithm to resolve the temporal and spatial variations in the solution field associated with boundary deformations with high accuracy and without any accumulation of error.

Up to fourth-order implicit temporal discretization schemes have been examined for all the test problems designed in the course of this work. The algorithm is found to deliver the theoretically predicted temporal accuracy in all the cases tested.

7.2.2.2. Linear solvers

The size of the system of discretized linear equations for a three-dimensional problem is considerably larger than its two-dimensional counterpart due to the presence of the additional spatial dimension which is discretized using Fourier expansions. Such large system of equations when attempted using a standard direct solver (as discussed in Chapter 5) results in very high memory requirement as well as extremely slow computational speed and, therefore, substantially limits the range of applications of the IBC algorithm for the three-dimensional problems in general. The structure of the complete coefficient matrix for the three-dimensional problem is similar to the structures observed for the two-dimensional problems. As a result, different solutions schemes including serial and parallel iterative solvers and two versions of the specialized direct solver have been developed in [4] and their relevant performances are examined by comparing against a 'reference solver' based on LU decomposition of the complete coefficient matrix.

The parallel Jacobi iterative solver based on 4 CPUs are found to perform slower than its serial version while the Gauss-Seidel iterative solver is found to be the fastest. Although the iterative solvers are seen to be fairly efficient compared to the reference solver they suffer from similar convergence issues as observed in the context of two-dimensional problems, which limit the range of geometries that can be accessed using such solvers.

Two versions of the specialized solver have been implemented to handle the three-dimensional problems, which are – i) *execution-efficient* implementation and ii) *memory-efficient* implementation. The *execution-efficient* version retains all the sub-matrices in

memory for re-usage while the *memory-efficient* version reconstructs all the sub-matrices whenever necessary and removes them from memory immediately after they are used. For steady fixed boundary problems the performance of both the versions has been found to be comparable although the *memory-efficient* method gives access to substantially larger number of Fourier modes required by the more difficult geometries. Comparative study on the performance of the serial Gauss-Seidel iterative solver and the *memory-efficient* specialized solver has been carried out for the steady fixed boundary problems. The test results presented in [4] reveal that the iterative solver is a bit more efficient for benign geometries (as defined by lower values of corrugation amplitude) while the specialized solver outperforms the iterative solver for the more extreme geometries where the iterative solver fails to converge. The advantage of the specialized solver increases with increasing number of Fourier modes without facing any issues similar to the convergence problem of the iterative solver. Therefore, specialized solvers are the best the options for the three-dimensional problems.

In the case of unsteady fixed boundary problems the *execution-efficient* specialized solver has shown clear advantage over the memory-efficient version when moderate number of Fourier modes (less than 10 modes) were used in the computations. However, for the moving boundary problems the *execution-efficient* method has shown little or no improvement in computational performance even for moderate number of Fourier modes. This is due to the fact that the effective computational time for memory intensive problems are dictated by the speed with which the CPUs are capable of accessing the physical memory as frequent retrieving of data from the memory becomes a major component of the total computational cost. The *execution-efficient* method can, therefore, eventually lead to slower computational performance for moving boundary problems with large enough number of Fourier modes used in the computations.

In the view of the aforementioned observations it can be concluded that both versions of the specialized solver provides excellent computational efficiency for the three-dimensional problems and the *memory-efficient* version is preferable to the *execution-*

efficient version for solving problems involving extreme geometries that require use of larger number of Fourier modes.

7.3. Applications of the IBC algorithm in solving problems of physical interest

The material presented in this dissertation has been primarily devoted to explaining the different concepts of the IBC algorithm and characterizing the algorithm as applied to different model problems in the presence of fixed and moving irregularities. We have focused more on the moving boundary problems as such problems are more difficult and challenging to deal with for any algorithm. The algorithm developed and described in the previous chapters can be applied to analyze a wide range of computational fluid dynamics problems, which in addition to being challenging have practical importance from physical perspective. With growing popularity of micro-electro-mechanical systems (MEMS) efficient and accurate algorithms to handle moving boundary problems are becoming very important as such problems are encountered in various engineering applications of MEMS [6]. Numerical simulation of flow problems in micro-conduits involving fixed or deforming surfaces are crucial in developing various micro-fluidic devices. In this regard, the algorithms presented in the context of Stokes flow in Chapter 3 and low Reynolds number viscous flows described by the Navier-Stokes equations presented in Chapter 6 would be particularly useful in simulating flows in micro-conduits involving fixed and moving irregularities. The presented moving boundary algorithms would be also very useful and computationally efficient in simulating various biomedical flow problems, e.g., pulsatile flows and peristaltic flows. For example, the algorithm will enable researchers to study peristaltic pumping in order to devise efficient peristaltic pumps. Identification of optimum boundary geometry that can lead to the most improvement in a desired functionality usually requires investigation on a large number of different geometries. The IBC algorithm is very suitable for such studies due to its efficient modeling of boundary geometries. Analyzing stability of shear layers in viscous flows in the presence of surface roughness refers to a class of problems which are very

interesting as well as extremely challenging and the IBC algorithm can be extremely useful in all the different stages of such analyses, i.e., determination of the basic state, solution of the linearized disturbance equations and studying the evolution of the disturbances by developing efficient DNS solvers.

The purpose of the current section is to show the advantages of the IBC algorithm when applied to investigate two problems of practical interest. In order to maintain the focus of the discussion centered on the advantageous characteristics of the IBC algorithm, details on the physical aspects of the chosen example problems are going to be kept at a minimum in Sections 7.3.1 and 7.3.2.

7.3.1. Investigation of the effects of distributed surface roughness on the friction factor

In this section we shall illustrate the application of the IBC algorithm to study the effects of distributed surface roughness on the friction factor f for steady two-dimensional channel flows. Friction factor is a dimensionless quantity that is related to the pressure loss in a flow system due to friction along the length of the system. Classical work by Moody [7] relating the friction factor with Reynolds number and 'relative roughness' for fully developed pipe flows is famously known as the 'Moody Chart' and is widely used to evaluate pressure loss in pipe flows.

The particular example problem selected for this section deals with roughness elements that have the forms of triangles. However, the algorithm as well as the methodology to be presented in this section is applicable to any Fourier transformable geometric shape.

Consider channels with flat uppers walls while the lower walls are described by triangular roughness elements as shown in Fig. 7.3.1.1. The boundaries of the channels are defined by the relations

$$y_U = 1, \qquad y_L = \sum_{n=-N_A}^{N_A} H_L^{(n)} e^{inax}, \qquad (7.3.1.1)$$

where the Fourier coefficients $H_L^{(n)}$ can be evaluated in terms of the parameters S, a, b, c and λ (see Fig. 7.3.1.1) and therefore, are treated as known.

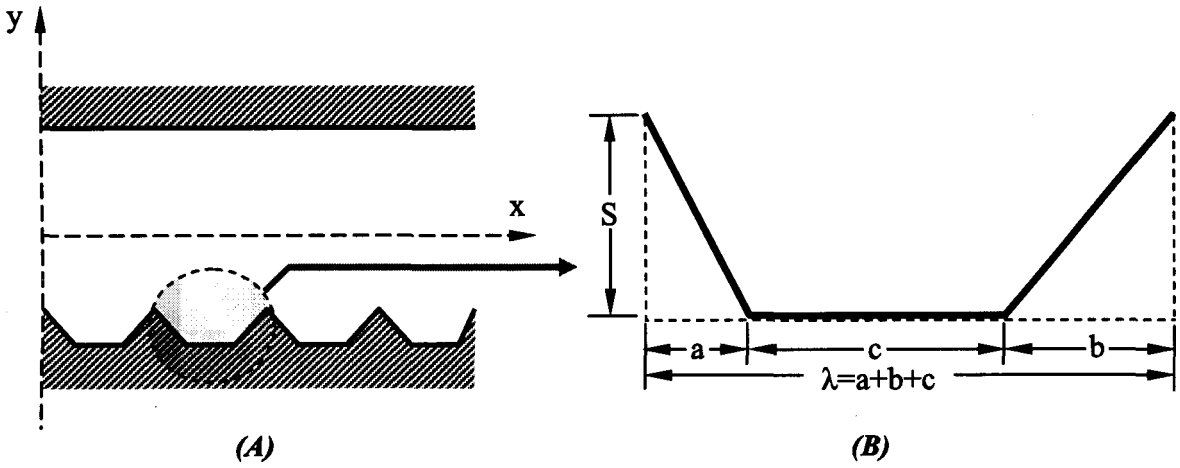


Figure 7.3.1.1. *Schematic of a channel with a flat upper wall while the lower wall has triangular roughness elements described by Eq.7.3.1.1 (Fig. 7.3.1.1A) and enlarged view of one wave length of the triangular roughness elements (Fig. 7.3.1.1B). The triangular geometries over one wave length λ are defined in terms of the parameters S, a, b and c as shown Fig. 7.3.1.1B. The wave number $\alpha=2\pi/\lambda$, where $\lambda=(a+b+c)$.*

The interest in such a problem could be motivated by different objectives that may include

- determining the optimum triangular shapes in terms of the (a/b) ratios corresponding to the lowest or highest friction factors,
- investigating the effects of the distance c between two triangular elements on the friction factor for shapes with different (a/b) ratios and different Re,
- studying effects of Re on friction factor for geometries defined by fixed (a/b) ratio while the values of S and/or c are variable, etc.

In order to perform any or all of the aforementioned analyses, we need to solve the steady version of the Navier-Stokes equations formulated in terms of the stream function given by Eq. (4.5.4) subject to the boundary conditions (4.4.5) using the discretization technique described in Section 4.4. We shall be working with the classical formulation of the IBC algorithm and refer to the computer program that computes the solution of (4.5.4, 4.4.5) as the 'mean flow solver'. The pressure field as well as the associated pressure loss and friction factor in the presence of the distributed roughness need to be evaluated as a part of post-processing and the details regarding the evaluations of these quantities are provided in Appendix K.

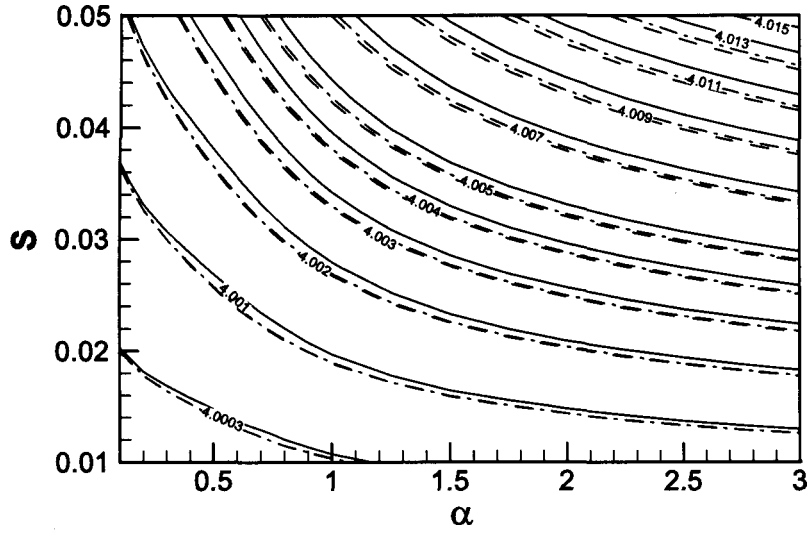


Figure 7.3.1.2. *Curves of constant* $f \times \mathrm{Re}$ *for channels with wall geometries defined by Eq. (7.3.1.1), where f is the non-dimensional friction factor (see Appendix K). Results are presented for Re=500 and three particular types of lower wall geometries. Type 1 (dash-dot lines): a/b=3, c=0; Type 2 (solid lines): a/b=1, c=0; Type 3 (dash lines): a/b=1/3, c=0.*

Figures 7.3.1.2 and 7.3.1.3 show two illustrative examples of friction factor contours in the α-S parameter space where α is the wave number of the triangular roughness elements and S is the base to peak height of the roughness. Figure 7.3.1.2 shows the contours of friction factor for triangular geometries with different (a/b) ratio while the Reynolds number is kept fixed at Re=500. On the other hand, Fig. 7.3.1.3 shows the

curves of constant friction factors for different values of Re for triangular roughness elements of a fixed (a/b) ratio.

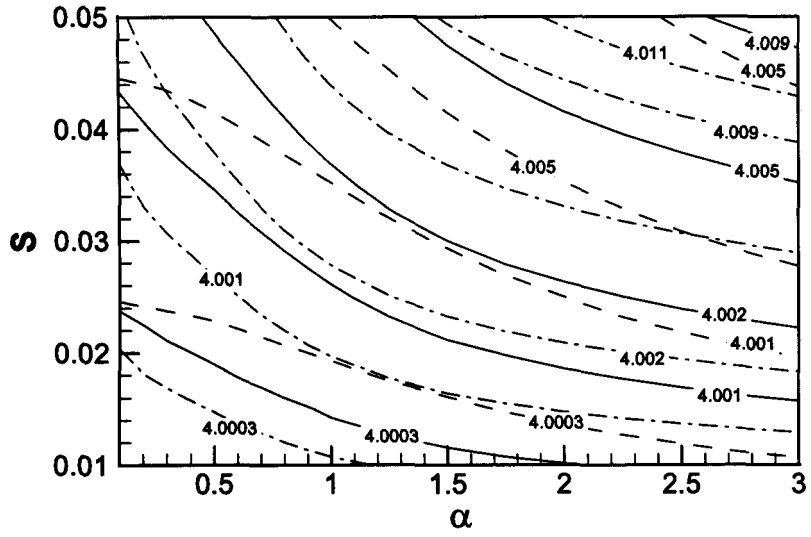


Figure 7.3.1.3. *Curves of constant* $f \times \mathrm{Re}$ *for channels with wall geometries defined by Eq. (7.3.1.1) for different values of Re, where f is the non-dimensional friction factor (see Appendix K). Results are presented for Re=1 (dash lines), Re=100 (dash-dot lines) and Re=500 (solid lines). In all cases the wall geometries belong to Type 2 (a/b=1, c=0) as defined in Fig. 7.3.1.2.*

Construction of contour plots of friction factor as shown in Figures 7.3.1.2 and 7.3.1.3 usually calls for investigation of a large number of geometries described by the variations of S and α. In our case, we have considered 26 and 18 different values of S and α, respectively, which lead to a total of 468 different geometries. Algorithms that require generation of body conforming grids would warrant substantial manual effort associated with the process of grid generation for all the different channel geometries. Whereas, in the case of the IBC algorithm the use of regular computational domain coupled with the use of very efficient modeling of the geometries eliminates the computational costs and manual efforts linked with grid generation. The different geometries in the IBC algorithm are defined in terms of the Fourier coefficients $H_L^{(n)}$ only, which are conveniently constructed by a separate function that just requires to take the values of a, b, α and S as its inputs. So, the manual effort associated with the investigation of 468 different

geometries is reduced to forming arrays of α and S for the different geometries. Afterwards, the mean-flow solver is called repetitively from loops that go through the different elements of the arrays of α and S.

In order to approximate the triangular geometries with sufficient accuracy, particularly in the cases of larger values of α and S, one needs to consider at least ten terms in the Fourier expansions defining the shape of the lower walls, i.e., $N_A \geq 10$. Consequently a significantly large number of Fourier modes N_M is required to define the solution. The data points for the Figures 7.3.1.2 and 7.3.1.3 were computed by setting the values of N_M and N_T to 75 and 150 in the mean flow solver, respectively. With the use of specialized linear solver the solution time for a single geometry has been observed to be approximately 17 seconds or less. The amount of memory required by the specialized solver for such combinations of N_M and N_T is little less than 200MB (Mega Bytes). On the other hand, the standard direct solvers would be prohibitively slow in computations for such large number Fourier modes and Chebyshev polynomials. Besides this, memory required by the standard solvers for such combinations of N_M and N_T is approximately 7GB (Giga Bytes), which is fairly high for most of the standard serial computing platforms.

One may decide to use commercial CFD codes for the type of investigations presented in this section. However, most of the commercially available CFD codes would require very fine grid resolution or equivalent to match the spatial accuracy achievable by the IBC method, which ultimately leads to larger computational costs. Moreover, manual efforts related to grid generation or equivalent operations for the different geometries will also add to the disadvantage for the commercial solvers.

7.3.2. Traveling wave instability in wavy channels

The second example on the application of the IBC algorithm is concerned with studying the stability of shear layers in wavy channels. Surface irregularities of various forms and

scales exist in most flow systems and contribute to spatial modulation of the flow fields. Although the application of the IBC algorithm in the current section is going to be presented in the context of wavy channels, the algorithm is equally applicable to any other form of distributed roughness that are expressible using Fourier expansions.

Analyzing possible responses of flows subject to spatial modulation of various forms are essential in devising various flow control strategies in order to alter particular properties of the flow field in favorable ways [8]. Wavy channel walls fall under the domain of distributed roughness. Surface roughness may be properly designed from various flow control standpoints, e.g., to accelerate or delay the laminar-turbulent transition process, to increase or reduce the skin friction drag, to enhance heat transfer by improving mixing, etc. Among the various aspects of the effects of surface roughness adequate comprehension of how it affects the laminar-turbulent transition process in shear layers is very fundamental. Flow instabilities as well as transition to turbulence can be investigated through various means including laboratory and numerical experiments as well as linear, weakly nonlinear, strongly nonlinear and bifurcation theories [9]. For our study we shall be using linear stability analysis which is based on the linear theory and is particularly suitable low-disturbance environments.

The walls of the wavy channel for the example problem are expressed as

$$y_L(x) = -1 + S[e^{i\alpha x} + e^{-i\alpha x}], \qquad y_U(x) = 1 + S[e^{i\alpha x} + e^{-i\alpha x}], \qquad (7.3.2.1)$$

where S and α are real quantities denoting the amplitude and wave number of the wall corrugation, respectively.

Before conducting the linear stability analysis we need to determine the form of the basic state or mean flow first by using the mean flow solver (described in Section 7.3.1) for channels bounded by the wavy walls described by Eq.(7.3.2.1). The basic state is computed in terms of the modification of stream function Ψ_1 (see Sections 4.4 and 4.5) in the presence of surface roughness. The associated modifications in the velocity fields u_1

and v_1 as well as the total mean flow velocity components u and v can be computed as a part of post-processing. For convenience, from this point onwards in this section, we shall use the subscript '2' to denote the total mean flow quantities, i.e.,

$$\mathbf{V}_2(\mathbf{x}) = [u_2(x,y), v_2(x,y)] = \mathbf{V}_0(\mathbf{x}) + \mathbf{V}_1(\mathbf{x}) = [u_0(y),0] + [u_1(x,y) + v_1(x,y)],$$

$$p_2(\mathbf{x}) = p_0(x) + p_1(x,y). \qquad (7.3.2.2)$$

Once the form of the basic state is computed, the next step is linear stability analysis where we shall work with the governing equations in the form of vorticity transport and continuity equations. The details of the procedure involved in linear stability analysis are given in [10]. In the present section we shall present the relevant formulations in a brief manner as the main objective of this section is to illustrate an application of the IBC algorithm. We shall superimpose unsteady three-dimensional disturbances on the mean flow quantities as

$$\boldsymbol{\Omega}(x,y,z,t) = \boldsymbol{\Omega}_2(x,y) + \boldsymbol{\Omega}_3(x,y,z,t), \qquad \mathbf{V}(x,y,z,t) = \mathbf{V}_2(x,y) + \mathbf{V}_3(x,y,z,t), \quad (7.3.2.3)$$

where $\boldsymbol{\Omega} = [\xi, \eta, \upsilon]$ denotes the vorticity vector. The subscripts 2 and 3 refer to the mean flow and disturbance fields, respectively. The mean flow velocity field has the form

$$\mathbf{V}_2(x,y) = [u_0(y),0,0] + \sum_{n=-N_M}^{N_M} [f_u^{(n)}(y), f_v^{(n)}(y), 0] e^{in\alpha x}, \qquad (7.3.2.4)$$

where $f_u^{(n)} = f_u^{(-n)^*}, f_v^{(n)} = f_v^{(-n)^*}$ and star denotes complex conjugate. Substitution of Eq. (7.3.2.3) into the governing vorticity and continuity equation followed by subtraction of the mean parts and subsequent linearization result in disturbance equations of the form

$$\frac{\partial \boldsymbol{\Omega}_3}{\partial t} + (\mathbf{V}_2.\nabla)\boldsymbol{\Omega}_3 - (\boldsymbol{\Omega}_3.\nabla)\mathbf{V}_2 + (\mathbf{V}_3.\nabla)\boldsymbol{\Omega}_2 - (\boldsymbol{\Omega}_2.\nabla)\mathbf{V}_3 = \mathrm{Re}^{-1}\nabla^2\boldsymbol{\Omega}_3,$$

$$\nabla.\mathbf{V}_3 = 0, \qquad \boldsymbol{\Omega}_3 = \nabla \times \mathbf{V}_3. \tag{7.3.2.5a-c}$$

The boundary conditions for the disturbance velocities take the homogeneous form

$$\mathbf{V}_3(x, y_L(x), z) = 0, \qquad \mathbf{V}_3(x, y_U(x), z) = 0. \tag{7.3.2.6a,b}$$

As we are interested in the asymptotic stability, therefore, the t and z dependence are taken care of by considering the solution in the form

$$\mathbf{V}_3(x, y, z, t) = [h_u(x, y), h_v(x, y), h_w(x, y)]e^{i(\delta x + \mu z - \sigma t)} + CC, \tag{7.3.2.7}$$

where δ and μ are real and indicate the wave numbers of the disturbance field in the stream-wise and span-wise directions, respectively. On the other hand, σ is a complex quantity, i.e, $\sigma = \sigma_r + i\sigma_i$, with its real ($\sigma_r$) and imaginary ($\sigma_i$) parts describing the frequency and the growth rate of the disturbances, respectively. In the case of stationary vortices the exponent σ is purely imaginary. In the above expression CC stands for complex conjugate. The functions h_u, h_v and h_w describe the amplitudes of the disturbance velocity vector which are modulated by the surface irregularities and, therefore, the disturbance velocity vector is further discretized as

$$\mathbf{V}_3(x, y, z, t) = \sum_{m=-N_D}^{N_D} [g_u(y), g_v(y), g_w(y)]e^{i[(\delta + m\alpha)x + \mu z - \sigma t]} + CC, \tag{7.3.2.8}$$

where $g_u^{(m)} = g_u^{(-m)^*}, g_v^{(m)} = g_v^{(-m)^*}, g_w^{(m)} = g_w^{(-m)^*}$, star denotes complex conjugate and N_D denotes the number of modes used to define the modulated disturbance amplitudes.

Substitution of (7.3.2.4) and (7.3.2.8) into the disturbance equations (7.3.2.5) and separation of the coefficients of the different Fourier modes result in a system of three ordinary differential equations in terms of the unknowns $g_u^{(m)}$, $g_v^{(m)}$ and $g_w^{(m)}$ for each

Fourier mode 'm'. For convenience, we introduce a new term Ξ, which is related to the y-component of the disturbance vorticity η_3 by the relation $\Xi = i\eta_3$. The solution for Ξ can be assumed in the same form given by (7.3.2.8) as

$$\Xi = \sum_{m=-N_D}^{N_D} g_{\Xi}^{(m)} e^{i[(\delta+m\alpha)x+\mu z-\sigma t]} + CC. \qquad (7.3.2.9)$$

The unknowns $g_u^{(m)}$ and $g_w^{(m)}$ then can be expressed in terms of $g_{\Xi}^{(m)}$ and $g_v^{(m)}$ as

$$g_u^{(m)} = [i\Gamma\gamma^{(m)} D g_v^{(m)} - \mu g_{\Xi}^{(m)}]/\{\kappa^{(m)}\}^2,$$

$$g_w^{(m)} = [\gamma^{(m)} g_{\Xi}^{(m)} + i\Gamma\mu D g_v^{(m)}]/\{\kappa^{(m)}\}^2, \qquad (7.3.2.10)$$

where $D = d/d\hat{y}$, $\gamma^{(m)} = \delta + m\alpha$, $\{\kappa^{(m)}\}^2 = \{\gamma^{(m)}\}^2 + \mu^2$ and Γ is constant resulting from the transformation of the y-coordinate into the $\hat{y}$-coordinate using Eq. (6.2.10). Introduction of $g_{\Xi}^{(m)}$ thus reduces the system of three ordinary differential equations for each mode in terms of $g_u^{(m)}$, $g_v^{(m)}$ and $g_w^{(m)}$ into a system of two ordinary differential equations in terms of $g_v^{(m)}$ and $g_{\Xi}^{(m)}$ which are

$$T^{(m)} g_{\Xi}^{(m)} + Re\,\mu\Gamma D u_0 g_v^{(m)} = Re \sum_{n=-N_D}^{N_D} [E_v^{(m,n)} g_v^{(m-n)} + E_{\Xi}^{(m,n)} g_{\Xi}^{(m-n)}],$$

$$S^{(m)} g_v^{(m)} = -Re \sum_{n=-N_D}^{N_D} [H_v^{(m,n)} g_v^{(m-n)} + H_{\Xi}^{(m,n)} g_{\Xi}^{(m-n)}], \qquad (7.3.2.11a,b)$$

where $S^{(m)}, T^{(m)}, E_v^{(m,n)}, E_{\Xi}^{(m,n)}, H_v^{(m,n)}$ and $H_{\Xi}^{(m,n)}$ are various differential operators and their explicit forms are given in Appendix L. The modal disturbance equations (7.3.2.11) are further discretized using expansions based on Chebyshev polynomials as

$$g_v^{(m)}(\hat{y}) = \sum_{j=0}^{N_T} G_j^{(m)} T_j(\hat{y}), \qquad g_\Xi^{(m)}(\hat{y}) = \sum_{j=0}^{N_T} Z_j^{(m)} T_j(\hat{y}). \qquad (7.3.2.12a,b)$$

Finally, (7.3.2.12) is substituted into (7.3.2.11) and the method Galerkin projection is applied to discretize the disturbance field equations into a system of algebraic equations.

The homogeneous boundary conditions (7.3.2.6) are discretized using the immersed boundary conditions method as described in the previous chapters. The final form of the discretized boundary constraints for the upper wall become

$$\sum_{n=-N_D}^{N_D} \sum_{k=0}^{N_T} \left[i\Gamma\gamma^{(n)} \{\kappa^{(n)}\}^{-2} (d_U)_k^{(m-n)} \right] G_k^{(n)} - \sum_{n=-N_D}^{N_D} \sum_{k=0}^{N_T} \left[\mu \{\kappa^{(n)}\}^{-2} (w_U)_k^{(m-n)} \right] Z_k^{(n)} = 0 \ ,$$

$$\sum_{n=-N_D}^{N_D} \sum_{k=0}^{N_T} \left[i\Gamma\gamma^{(n)} \mu \{\kappa^{(n)}\}^{-2} (d_U)_k^{(m-n)} \right] G_k^{(n)} + \sum_{n=-N_D}^{N_D} \sum_{k=0}^{N_T} \left[\gamma^{(n)} \{\kappa^{(n)}\}^{-2} (w_U)_k^{(m-n)} \right] Z_k^{(n)} = 0 \ ,$$

$$\sum_{n=-N_D}^{N_D} \sum_{k=0}^{N_T} \left[(w_U)_k^{(m-n)} \right] G_k^{(n)} = 0, \qquad (7.3.2.13a\text{-}c)$$

where the coefficients $(w_L)_k^{(m)}$, $(w_U)_k^{(m)}$, $(d_L)_k^{(m)}$ and $(d_U)_k^{(m)}$ are known and the processes for evaluating these coefficients are explained in Chapter 4. The sample results to be presented in this section have been obtained by using the constant volume flux constraint as the additional closing relation.

The complete system of homogeneous equations comprised of the discretized disturbance equations and boundary conditions will have non-trivial solutions only for certain combinations of parameters δ, μ and σ. For our analyses, we shall pose the problem as an eigenvalue problem for σ. If $\sigma_i>0$ the flow is unstable with the disturbances growing exponentially, whereas $\sigma_i<0$ indicates exponential decay of the disturbances and $\sigma_i=0$ refers to neutral stability.

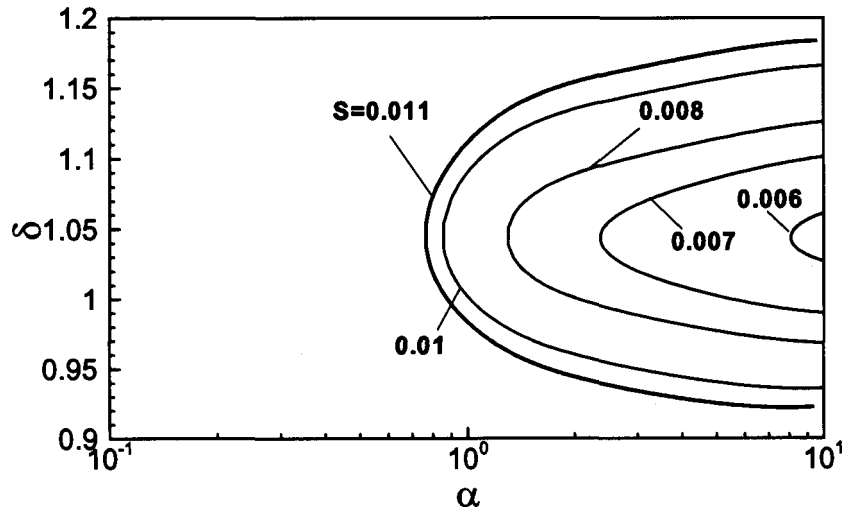


Figure 7.3.2.1. Neutral curves for Re=5000 and different values of S.

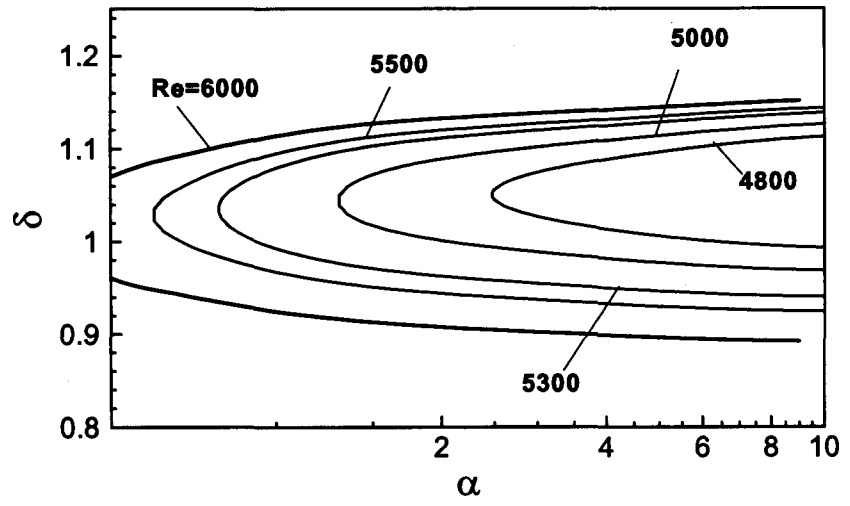


Figure 7.3.2.2. Neutral curves for S=0.008 and different values of Re.

A comprehensive study of the stability characteristics of flow in the presence of distributed surface roughness and for that matter in wavy channels requires among other things construction of neutral curves in different parameter. A neutral curve defines the boundary between the stable and unstable regimes in a parameter space. Figures 7.3.2.1 and 7.3.2.2 show neutral curves for flows in a wavy channel in the α-δ parameter space for fixed values of S and Re, respectively. The neutral curves provide information regarding the critical combinations of different flow and geometric parameters. A large

number of neutral curves are required to determine the global critical stability conditions as a function of the amplitude S of the waviness [11].

The linear stability equations are stiff in nature, which implies that the solution of these equations are extremely sensitive to the applied discretization scheme and are vulnerable to numerical instabilities. In order to overcome possible issues concerning numerical stability one needs to work with higher-order spatial discretization schemes. A comprehensive review of the various numerical methods applied to solve the classical Orr-Sommerfeld equation for plane Poiseuille flow is provided in [12, 13]. The experience of various researchers as reviewed in [12, 13] suggests the use of fourth-order or higher schemes for spatial discretization of the stability equations when discrete numerical methods based on finite-difference or finite-elements are to be applied. In the context of classical Orr-Sommerfeld equation for plane Poiseuille flow, Orszag [13] has shown that a fourth-order finite-difference scheme with 100 grid points is capable of predicting the most unstable eigenvalue with only three meaningful digits of accuracy. Application of finite-element method using a non-uniform grid and a fifth-order high precision Hermite element has been found to provide results comparable with the most accurate results available in the literature [12]. On the other hand, the spectral methods based on Chebyshev expansions are capable of evaluating the eigenvalues with very high accuracy and for less computational cost [13]. The steep gradients of flow properties near the boundaries are responsible for the requirements of higher-order schemes. Substantial dependence on the lower-order schemes or low-precision elements by most of the commercial CFD codes, therefore, makes them less suitable for analyzing stability problems. One can try to work around with lower-order schemes by using very fine grids, which will overwhelmingly increase the cost of grid-generation or equivalent operations.

The presence of boundary irregularities in our case results in modulated basic flows and consequently the stability operators for such problems are stiffer. Therefore, the level of difficulty to handle such problems using different discrete numerical techniques is increased even further. The DT method as presented in the previous chapters is capable of providing high spatial accuracy, however, the associated cost of mapping and

discretization of the stability equations will be enormous. The IBC algorithm presented in this dissertation relies on the spectral discretization on the spatial dimensions making it extremely suitable for linear stability analysis of spatially modulated flows. The algorithm is found to be secure against numerical instabilities and associated loss in numerical accuracy due to its use of spectral discretizations. The solution of the complete eigenvalue problem is computationally very expensive and the availability of very efficient of modeling of geometries by the IBC method makes it further attractive for the class of problem presented in this section. Besides these, it is well known that evaluation of the eigenvalues are fairly sensitive to the accuracy in the computed basic state [10, 11] and the spectral discretization used in the IBC algorithm ensures high accuracy in mean flow solution.

Two general approaches may be pursued to trace the eigenvalues for constructing the neutral curves, namely, global and local methods [8]. The global method works by solving the complete discretized stability equations using an eigenvalue solver that results in the spectrum eigenvalues at the expense of very high computational cost. A local method on the other hand works by converting the original eigenvalue problem into an initial value problem which requires an initial guess for the eigenvalue to start with. Although the global method is more reliable we opted for the local methods in our computations in order to avoid the excessive cost of solving eigenvalue problems. We have implemented two local methods for tracing of eigenvalues in the parameter space, which are the inverse-iteration and the Newton-Raphson search technique [11]. For both of the methods, repetitive solution of the mean flow is inevitable as each separate value of α indicates a different geometry. The problem of geometric variability is compounded for cases when both of the parameters related to the neutral curves affect the mean flow, i.e., when neutral curves are constructed in the α-S, α-Re, S-Re or similar parameter spaces. The variability of geometry and evaluation of associated mean flow thus act as the major costs in the computations behind the tracing of the neutral-curves for any algorithm and this is where the IBC method excels over various grid-dependent algorithms that suffer from major costs of grid generation. First of all, IBC method eliminates programming efforts associated with generating boundary conforming grids

for different geometries as all variations in geometry are reduced to setting the appropriate Fourier coefficients defining the geometries to correct values. Secondly, the IBC method with the use of specialized linear solvers developed in Chapter 5 reduces the computational time to solve the mean flow by a significant margin and adds to the usefulness of the IBC algorithm even further.

7.4. Reliability of the results produced by the IBC method

Verification of the reliability of the results produced by an algorithm is a very important part of the validation process. The simplest way to carry out validation is to compare the new results against the others available in the literature. For the class of problems that are of interest for the present research, there are extremely few results available in the literature. This difficulty has been overcome in the dissertation by solving the test problems using a different and more traditional method, i.e., the domain transformation (DT) method, and using the same forms of spatial and temporal discretizations. The spectral implementations of the DT method used in the dissertation (with details provided in the appendices) are capable of solving the test problems with machine accuracy. However, poor computational efficiency makes the DT method unsuitable for practical use, particularly for the moving boundary and geometry-optimization problems. Excellent agreement of the results obtained using the IBC and DT methods should make the reader confident about the accuracy and reliability of the IBC method. In the case of the moving boundary problems, elastic traveling waves have been chosen as a test problem to describe the deforming boundaries. Such model problem permitted determination of the solution by solving a stationary problem in a moving frame of reference and thus without any temporal discretization errors. Comparison of results obtained through a direct solution of the moving boundary problem in a fixed frame of reference with the same problem solved in the moving frame of reference showed that the IBC algorithm preserves the spatial and temporal variations of the solution field, including spatial and temporal periodicity as well as spatial phase shifts.

Table 7.4.1. *Growth rate (σ_i) of the disturbances in a wavy channel (see Eq. 7.3.2.1) for Re=5000, α=2.0, S=0.007, δ=0 and μ=2.0. Mean flow computations have been carried out using N_M=5 Fourier modes by both the DT and IBC methods while only N_S number of modes are retained in the representation of the mean flow for stability computations.*

No. of Chebyshev polynomials, N_T	No. of Fourier modes retained to represent mean flow, N_S	No. of Fourier modes used to represent disturbance fields, N_D	Growth rate ($\sigma_i \times 10^3$) obtained by	
			Cabal et al [16] using DT method	IBC method in the present work
60	1	3	0.215	0.153
	2		0.229	0.185
	3		N/A*	0.185
70	1	3	0.184	0.153
	2		0.187	0.185
	3		N/A	0.185
80	1	3	0.184	0.153
	2		0.185	0.185
	3		N/A	0.185
90	1	3	0.184	0.153
	2		0.184	0.185
	3		N/A	0.185

* *N/A denotes that results for such cases are not available in [16].*

There are a few works available in the literature that deal with triangular roughness [14, 15], however, none of them deals with configurations of interest in this dissertation (see Section 7.3.1). The available papers dealing with the hydrodynamic stability analyses are mostly based on spectral methods but are limited to problems involving regular physical domains. Cabal et al [16] present some results dealing with the stability of flow in a wavy channel which provides an opportunity to compare the results obtained by the IBC method. Cabal et al [16] has used similar definition of boundary irregularities while applied the DT method for solving the resulting problem. A comparison of the growth rates of disturbances, σ_i (see Eq. 7.3.2.8), obtained by Cabal et al [16] and in the present dissertation are presented in Table 7.4.1. Comparison of both sets of results shows good

agreement. The growth rates computed from the eigenvalue problem are extremely sensitive to the accuracy of the basic state. In this regard, the good agreement between the growth rates computed using the different methods, as shown in Table 7.4.1, demonstrates the effectiveness and reliability of the IBC method in computing the mean flow in the presence of boundary irregularities as well as predicting its linear stability characteristics.

7.5. Conclusions

Implicit spectrally-accurate algorithms based on the concept of immersed boundary conditions for two-dimensional flow problems described by the Laplace operator, biharmonic operator and the Navier-Stokes equations have been presented in Chapters 2, 3 and 6 of the dissertation. The algorithms are capable of handling flow problems involving fixed and moving boundary irregularities. The core algorithm works with a regular computational domain that at any instant of time completely surrounds the physical domain. In order to achieve spectral convergence characteristics the algorithm employs Fourier and Chebyshev expansions in the x- and y-directions respectively. The spatial and temporal boundary locations in the IBC algorithm are modeled using Fourier expansions and, therefore, the algorithm is suitable for physical domains that are periodic in the stream-wise direction. Various tests for the different model problems confirm spectral accuracy in the spatial dimensions. Up to fourth-order implicit temporal discretization schemes employed for the different model problems are found to deliver theoretically predicted accuracy.

The discretization of the boundary conditions as implemented in the IBC algorithm leads to substantially larger number of internal constraints than required to form a close system of equations. The classical formulation of the IBC method only retains a certain number of constraints that is required to form a close system of equations. However, the classical IBC algorithm fails to provide reasonable accuracy beyond some critical combination of the wave number and the wave amplitude irrespective of any further increment in the

number of Fourier modes used to define the solution. Use of additional boundary constraints leads to over-determined formulations of the IBC algorithm. Direct solution of the over-determined problem by evaluating the pseudo-inverse of the complete coefficient matrix is shown in Chapter 4 to improve accuracy for the Laplace and Stokes problems by at least two orders of magnitude. However, the improvement in accuracy obtained by the over-determined formulation for the Navier-Stokes problem is found to depend on Re and diminishes with increase in the value of Re.

Efficient specialized linear solvers for the classical and over-determined formulations of the IBC algorithm that take advantage of the special structure of the coefficient matrix have been developed and examined in Chapters 5 and 6. For steady fixed boundary problems investigated in Chapter 5, the specialized solvers deliver two orders of speed-up in computations when compared against the standard direct solvers. The specialized solvers were also observed to be at least one order faster than the iterative solvers for such problems. Tests have shown that even the standard IBC solvers perform faster than the domain transformation (DT) method while the specialized solvers demonstrated at least two orders of magnitude difference compared to the DT method. In the case of unsteady fixed boundary problems, the use of specialized solvers reduce the computational cost to a level comparable to the cost of DNS for flow problems involving straight walls. In addition to being substantially faster, the specialized over-determined SVD based solver is found to deliver better accuracy than the any version of standard over-determined solver particularly around the critical zone. The specialized over-determined solver is found to extend the level of severity in boundary geometry that can be accessed by the IBC algorithm by a substantial margin. The IBC algorithm is especially well suited for moving boundary problems as the part of the coefficient matrix originating from the field equation remain unchanged during the solution process. The use of specialized solvers adds to this inherent advantage of the IBC algorithm and is found to improve the computational speed significantly in Chapter 6.

The extension of the IBC algorithm to three-dimensional problems has been presented in Section 7.2. The concept of the algorithm for three-dimensional problems is explained in

the context of the velocity-vorticity formulation of the three-dimensional Navier-Stokes equations. The considered form of the Navier-Stokes equations is comprised of the Laplace and biharmonic operators which have been successfully discretized for the two-dimensional problems using the IBC method. It has been shown that the basic concepts of the IBC algorithm remains unchanged when applied to the three-dimensional problems as the solution field in the additional spatial dimension is assumed to be periodic. As in the case of the two-dimensional problems, the periodic dimensions are discretized using Fourier expansions while the normal-to-the-wall direction is discretized using expansions based on Chebyshev polynomials. The physical boundaries are modeled using Fourier expansions in the periodic directions. Various issues related to the implementation of the IBC algorithm for three-dimensional problems have been presented and discussed in the context of the heat conduction problem which is described by the Laplace operator. Up to fourth-order accurate temporal discretization schemes have been implemented for the conduction problem. Solutions in the spatial dimensions have been found to be spectrally-accurate for both fixed and moving boundary problems subject to constant and time-dependent boundary conditions. For all the test problems, the temporal schemes are found to deliver the expected accuracy. The dispositions of the algorithm for the three-dimensional problem have been found to be consistent with the observations made in the case of the two-dimensional problems. Serial and parallel iterative solvers as well as two versions of the specialized direct solver have been implemented in order to identify the most efficient solution approach. The *execution-efficient* specialized solver enhances computational speed by retaining certain blocks of the coefficient matrix in memory for repetitive usage while the *memory-efficient* version eliminates all information from memory immediately after their use and reconstructs the blocks of the coefficient matrix whenever required. Both versions of the specialized solvers are found to be computationally more efficient compared to the standard and iterative solution methods. However, for larger numbers of Fourier modes used in the computations, the *memory-efficient* specialized method has been found to be more efficient especially for the moving boundary problems. Therefore, the *memory-efficient* specialized method is recommended for handling problems with severe geometries where

larger numbers of Fourier modes are required to obtain solutions with acceptable accuracy.

The example applications presented in Section 7.3 have illustrated the benefits of the IBC algorithm in the context of two physically interesting and numerically challenging problems. The linear stability analyses presented in Section 7.3.2 also provided an example of the application of the IBC algorithm for solving three-dimensional problems involving surface irregularities. The elimination of coordinate mapping and grid generation or equivalent operations resulted in efficient performance of the IBC algorithm for the example problems both of which require investigation of a wide range of channel geometries. Spectral discretization of the spatial dimensions as implemented in the IBC algorithm makes it very useful for stability analysis. The specialized solver developed in Chapter 5 is found to add significantly to the intrinsic advantages of the IBC algorithm thereby making the algorithm more suitable than the discrete algorithms based on finite difference, finite elements or finite volumes for the type problems presented in Section 7.3. Very good agreement between the results obtained by the IBC method and the same based on DT method from available literature as presented in Section 7.4 demonstrates the reliability and effectiveness of the IBC method.

7.6. Recommendations for future work

Spectrally-accurate algorithms for fixed and moving boundary problems that are based on the immersed boundary conditions (IBC) concept have been presented in this dissertation. Flow problems described by the two-dimensional Laplace, biharmonic and Navier-Stokes equations have been investigated in the course of the present work using classical and over-determined formulations of the IBC method. In order to advance the research pertinent to this dissertation the following works may be undertaken in future.

i) It has been discussed in this dissertation that the extension of the IBC method to three-dimensional flow problems described by the Navier-Stokes equations

does not require any conceptual difficulty. However, the implementation of the algorithm to solve three-dimensional nonlinear moving boundary problems may require optimizing the specialized solvers for a balance between the computational speed and memory requirements. Therefore, the next logical step would be to implement the IBC algorithm for such problems to examine the ranges of surface geometries and motions that can be handled by the IBC algorithm in a practically feasible manner. The stability equations should also be reformulated to account for the three-dimensional mean flow modifications in order to analyze the stability characteristics of such flows.

ii) The most suitable temporal discretization methods for the direct numerical simulation of the Navier-Stokes problem are the hybrid schemes [1] that treat the nonlinear terms explicitly while use implicit methods for the viscous terms. However, the different hybrid schemes that have been tested in the course of the present research appeared to be numerically unstable when they are used in conjunction with the IBC method. In this regard, it would be extremely useful to devise a numerically stable hybrid scheme for the IBC method in order to eliminate the iterations on the nonlinear terms required by the fully-implicit schemes and would, therefore, result in an overall robust algorithm for performing DNS in irregular domains.

iii) Development of spectrally-accurate algorithms for compressible flows as well as for non-Newtonian fluids in the presence of fixed and moving boundary irregularities using the IBC method can be very interesting for different researchers.

iv) From physical perspective studying the effects of surface roughness on the heat transfer process in heat exchangers can be an interesting as well as a very important research work. Such a problem does not require any conceptual modification of the IBC algorithm. The main task from the numerical point-of-view would be to couple the energy equation with the

Navier-Stokes equations in the mean flow solver, linear stability analysis and the DNS solver. The fact, that convection is the dominant mode of heat transfer in a heat exchanger, provides opportunity for improvement of its efficiency by increasing the convective heat transfer coefficient. Properly structure roughness can be used to induce instabilities in the flow field to enhance the momentum transport in the flow of the circulating fluid to increase this coefficient. Surface roughness may also improve the efficiency by increasing the effective surface area available for the heat transfer process. However, instabilities caused by rough surfaces may necessitate more energy to drive the flow through the system. So, a research in this regard may need to be focused on optimizing heat transfer against the energy required to run the system.

7.7. References

[1] Jiang, Y. and Floryan, J.M., Finite-difference 4^{th}-order compact scheme for the direct numerical simulation of instabilities of shear layers, *Int. J. Num. Meth. Fluids*, vol. 48, pp. 1259-1281 (2005).

[2] Kim, J., Moin, P. and Moser, R., Turbulence statistics in fully developed channel flow at low Reynolds number, *J. Fluid. Mech.*, vol. 177, pp. 133-166 (1987).

[3] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., Spectral methods in fluid dynamics, *Springer* (1987).

[4] Fernandez, D.C.D.R., Husain, S.Z. and Floryan, J.M., Immersed boundary conditions method for three-dimensional heat diffusion problems, *Expert Systems in Fluid Dynamics Research Laboratory Report ESFD-1/2009*, Department of Mechanical and Materials Engineering, The University of Western Ontario, London, Ontario, N6A 5B9, Canada (2009).

[5] Fernandez, D. C. D. R., Husain, S.Z., and Floryan, J.M., Parallel Implementation of the Immersed Boundary Conditions Method for Three Dimensional Heat Conduction in a Medium with Moving Boundaries, *Proceedings of the 17^{th} Annual*

Conference of the CFD Society of Canada, May 3-5, Ottawa, Ontario, Canada (2009).

[6] Zhao, Y., Tai, J. and Ahmed, F., Simulation of micro flows with moving boundaries using higher-order upwind FV method on unstructured grids, *Comp. Mechanics,* vol. 28, pp. 66-75 (2002).

[7] Moody, L.F., Fiction factor for pipe flow, *Trans. ASME*, vol. 66, pp. 671-684 (1944).

[8] Schmid, P.J. and Henningson, D.S., Stability and transition in shear flows, *Springer* (2001).

[9] Drazin, P.G., Introduction to hydrodynamic stability, *Cambridge Univ. Press* (2002).

[10] Floryan, J.M., Stability of wall bounded shear layers in the presence of simulated distributed surface roughness, *J. Fluid Mech.*, vol. 335, pp. 29-55 (1997).

[11] Floryan, J.M., Centrifugal instability of Couette flow over a wavy wall, *Phys. Fluids*, vol. 14, pp. 312-322 (2002).

[12] Mamou, M. and Khalid, M., Finite element solution of the Orr–Sommerfeld equation using high precision Hermite elements: plane Poiseuille flow, *Int. J. Numer. Meth. Fluids*, vol. 44, pp. 721-735 (2004).

[13] Orszag, S.A., Accurate solution of the Orr-Sommerfeld stability equation, *J. Fluid Mech.*, vol. 50, pp. 689-703 (1971).

[14] Rawool, A.S., Mitra, S.K. and Kandlikar, S.G., Numerical simulation of flow through microchannels with designed roughness, *Microfluid. Nanofluid.*, vol. 2, pp. 215-221 (2006).

[15] Khadem, M.H., Shams, M. and Hossainpur, S., Numerical simulation of roughness effects on flow and heat transfer in microchannels at slip flow regime, *Int. Commu. Heat Mass Trans.*, vol. 36, pp. 69-77 (2009).

[16] Cabal, A., Szumbarski, J. and Florayn, J.M., Stability of flow in a wavy channel, *J. Fluid Mech.*, vol. 457, pp. 191-212 (2002).

APPENDIX A

Evaluation of coefficients $w_{k,\tau+1}^{(m)}$ and $d_{k,\tau+1}^{(m)}$ required in Eq.(3.3.14) and Eq.(6.3.4) and $r_{k,\tau+1}^{(m)}$ required in Appendix C.

Coefficients in (3.3.14a) and (6.3.4a) can be evaluated with the help of the recurrence relation $T_{k+1}(\hat{y}) = 2\hat{y}T_k(\hat{y}) - T_{k-1}(\hat{y})$ that leads to

$$w_{k+1,\tau+1}^{(m)} = 2\sum_{n=-\infty}^{n=+\infty} P_{\tau+1}^{(n)} w_{k,\tau+1}^{(m-n)} - w_{k-1,\tau+1}^{(m)} \qquad \text{for } k>1, \tag{A.1}$$

whose evaluation begins at k=0 giving the initial terms in the form

$$w_{0,\tau+1}^{(0)} = 1,\ w_{0,\tau+1}^{(m)} = 0 \ \text{ for } |m| \geq 1; \qquad w_{1,\tau+1}^{(m)} = P_{\tau+1}^{(m)} \ \text{ for } |m| \geq 0. \tag{A.1}$$

Coefficients in (3.3.14b) and (6.3.4b) can be evaluated with the help of the recurrence relation $DT_{k+1}(\hat{y}) = 2T_k(\hat{y}) + 2\hat{y}DT_k(\hat{y}) - DT_{k-1}(\hat{y})$ that leads to a relation

$$d_{k+1,\tau+1}^{(m)} = 2\sum_{n=-\infty}^{\infty} P_{\tau+1}^{(n)} d_{k,\tau+1}^{(m-n)} - d_{k-1,\tau+1}^{(m)} + 2w_{k,\tau+1}^{(m)} \qquad \text{for } k>2, \tag{A.3}$$

whose evaluation begins at k=0 giving the initial terms in the form

$$d_{0,\tau+1}^{(m)} = 0 \quad \text{for } |m| \geq 0;\ d_{1,\tau+1}^{(0)} = 1,\ d_{1,\tau+1}^{(m)} = 0 \ \text{ for } |m| \geq 1;\ \ d_{2,\tau+1}^{(m)} = 4P_{\tau+1}^{(m)} \ \text{ for } |m| \geq 0. \tag{A.4}$$

Coefficients $r_{k,\tau+1}^{(m)}$ required in Appendix E can be evaluated with the help of the recurrence relation $D^2T_{k+1}(\hat{y}) = 4DT_k(\hat{y}) + 2\hat{y}D^2T_k(\hat{y}) - D^2T_{k-1}(\hat{y})$ that leads to a relation

$$r_{k+1,\tau+1}^{(m)} = 2\sum_{n=-\infty}^{\infty} P_{\tau+1}^{(n)} d_{k,\tau+1}^{(m-n)} - r_{k-1,\tau+1}^{(m)} + 4d_{k,\tau+1}^{(m)} \qquad \text{for } k>3, \tag{A.5}$$

whose evaluation begins at k=0 giving the initial terms in the form

$$r_{0,\tau+1}^{(m)} = r_{1,\tau+1}^{(m)}\ 0 \quad \text{for } |m| \geq 0;\ r_{2,\tau+1}^{(0)} = 4,\ r_{2,\tau+1}^{(m)} = 0 \text{ for } |m| \geq 1;\ r_{3,\tau+1}^{(m)} = 24P_{\tau+1}^{(m)} \text{ for } |m| \geq 0. \tag{A.6}$$

APPENDIX B

Demonstration that $V_{\tau+1}^{(0)}$ in Eq.(3.3.17b) cannot be independently specified.

We can write the following relation along line l overlapping with the moving boundary $\hat{y} = f(x, \tau+1)$

$$\Psi_T(x+\lambda, \mathbf{f}(x+\lambda, \tau+1), \tau+1) - \Psi_T(x, \mathbf{f}(x, \tau+1), \tau+1) = \int_x^{x+\lambda} \frac{d\Psi_T(x, \mathbf{f}(x, \tau+1), \tau+1)}{dx} dx = 0 \tag{B.1}$$

which can be re-arranged as

$$\int_x^{x+\lambda} \left[\frac{\partial \Psi_T(x, f(x, \tau+1), \tau+1)}{\partial x} + \Gamma \frac{\partial \Psi_T(x, f(x, \tau+1), \tau+1)}{\partial \hat{y}} \frac{df}{dx} \right] dx = 0 \tag{B.2}$$

and expressed in terms of velocity components along the line l as

$$\int_x^{x+\lambda} \left[-v_l(x, \tau+1) + u_l(x, \tau+1) \frac{df}{dx} \right] dx = 0. \tag{B.3}$$

Substitution of (3.3.11) and (3.3.12) into (B.3) and integration gives

$$V_{\tau+1}^{(0)} = -i\alpha \sum_{n=-N_A}^{n=+N_A} n U_{\tau+1}^{(n)} P_{\tau+1}^{(n)^*} \tag{B.4}$$

and demonstrates that the mean value of $v_l(x, \tau+1)$, i.e., $V_{\tau+1}^{(0)}$, cannot be independently specified as it depends on the specification of $u_l(x, \tau+1)$ as well as on the wall geometry expressed by $f(x, \tau+1)$.

APPENDIX C

Evaluation of pressure field from the known stream function field for Stokes flow problem presented in Chapter 3.

The total pressure field is given by

$$p(x,y,t) = p_0(x) + p_1(x,y,t) \tag{C.1}$$

where $p_0(x)=-2x$ is the pressure field associated with the reference Poiseuille flow and $p_1(x,y,t)$ corresponds to the modification in the pressure field due to boundary motions. The dimensionless field equations have the form

$$\frac{\partial u_1}{\partial x} + \Gamma \frac{\partial v_1}{\partial \hat{y}} = 0, \tag{C.2a}$$

$$\frac{\partial u_1}{\partial t} + \frac{\partial p_1}{\partial x} - \frac{\partial u_1^2}{\partial x^2} - \Gamma^2 \frac{\partial u_1^2}{\partial \hat{y}^2} = 0, \tag{C.2b}$$

$$\frac{\partial v_1}{\partial t} + \Gamma \frac{\partial p_1}{\partial \hat{y}} - \frac{\partial v_1^2}{\partial x^2} - \Gamma^2 \frac{\partial v_1^2}{\partial \hat{y}^2} = 0. \tag{C.2c}$$

The velocity components u_1 and v_1 are known and can be expressed as

$$u_1(x,\hat{y},\tau+1) = \Gamma \sum_{n=-N_M}^{n=+N_M} D\Phi^{(n)}(\hat{y},\tau+1)e^{in\alpha x}, \quad v_1(x,\hat{y},\tau+1) = -i\alpha \sum_{n=-N_M}^{n=+N_M} n\Phi^{(n)}(\hat{y},\tau+1)e^{in\alpha x}. \tag{C.3}$$

The corresponding pressure field can be expressed as

$$p_1(x,\hat{y},\tau+1) = A_p(\tau+1)x + \sum_{n=-N_M}^{n=+N_M} p^{(n)}(\hat{y},\tau+1)e^{in\alpha x} \tag{C.4}$$

where $A_p(\tau+1)$ is a function of time and appears in the expression due to the fact that an x-periodic, time-dependent velocity field may be associated with a pressure field that has a time-dependent, linear component in x. The unknowns A_p and $p^{(n)}$ need to be evaluated from the momentum equations. We begin with the evaluation of A_p. Substitution of Eqs (C.3-C.4) into (C.2b) and use of the two-step time discretization result in

$$A_p(\tau+1) = \left[\Gamma^3 D^3 - 1.5\Gamma\Delta t^{-1} D\right]\Phi_{\tau+1}^{(0)} + 2\Delta t^{-1}\Gamma D\Phi_{\tau}^{(0)} - 0.5\Delta t^{-1}\Gamma D\Phi_{\tau-1}^{(0)}, \tag{C.5a}$$

$$\begin{aligned} p^{(n)}(\hat{y},\tau+1) = {} & \Gamma(in\alpha)^{-1}\left[\Gamma^2 D^2 - (n\alpha)^2 - 1.5\Gamma\Delta t^{-1}\right] D\Phi_{\tau+1}^{(n)} + 2\Gamma\Delta t^{-1} D\Phi_{\tau}^{(n)} \\ & - 0.5\Gamma\Delta t^{-1} D\Phi_{\tau-1}^{(n)} \quad \text{for } |n| \neq 0 \end{aligned} \tag{C.5b}$$

where $D = d/d\hat{y}$. Similar substitutions applied to Eq.(C.2c) lead to

$$\begin{aligned} Dp^{(n)}(\hat{y},\tau+1) = {} & in\alpha\Gamma^{-1}\left[\Gamma^2 D^2 - (n\alpha)^2 - 1.5\Gamma\Delta t^{-1}\right]\Phi_{\tau+1}^{(n)} - 2in\alpha\Delta t^{-1}\Phi_{\tau}^{(n)} \\ & + 0.5in\alpha\Delta t^{-1}\Phi_{\tau-1}^{(n)} \quad \text{for } |n| \geq 0 \end{aligned} \tag{C.6}$$

In the case of mode zero the right hand side of (C.6) becomes zero which results in

$$p^{(0)}(\hat{y},\tau+1) = \text{const}. \tag{C.7}$$

The final expression for the complete pressure field has the form

$$p(x,\hat{y},\tau+1) = \left[A_p(\tau+1) - 2\right]x + \sum_{n=-N_M, n\neq 0}^{n=+N_M} p^{(n)}(\hat{y},\tau+1)e^{in\alpha x} + \text{const} \tag{C.8}$$

where the constant can be selected conveniently.

APPENDIX D

Specification of the closing boundary conditions corresponding to the constant pressure gradient constraint for the Stokes flow problem presented in Chapter 3.

Equation (3.3.30a) represents boundary condition that can be selected arbitrarily. Equation (3.3.30b) needs to be replaced with condition expressing the desired pressure gradient. Eq. (E5a) can be rewritten as

$$\left[\Gamma^3 D^3 - 1.5\,\Gamma\Delta t^{-1} D\right]\Phi^{(0)}_{\tau+1} = A_p(\tau+1) - 2\Delta t^{-1}\Gamma D\Phi^{(0)}_{\tau} + 0.5\Delta t^{-1}\Gamma D\Phi^{(0)}_{\tau-1}. \tag{D.1}$$

Integration of the above equation across the channel gives

$$\begin{aligned}\left[\Gamma^3 D^2 - 1.5\Gamma\Delta t^{-1}\right]&\left[\Phi^{(0)}(\hat{y}_U(x,\tau+1),\tau+1) - \Phi^{(0)}(\hat{y}_L(x,\tau+1),\tau+1)\right] = \\ A_p(\tau+1)&\left[\hat{y}_U(x,\tau+1) - \hat{y}_L(x,\tau+1)\right] - 2\Delta t^{-1}\Gamma\left[\Phi^{(0)}(\hat{y}_U(x,\tau),\tau) - \Phi^{(0)}(\hat{y}_L(x,\tau),\tau)\right] + \\ &0.5\Delta t^{-1}\Gamma\left[\Phi^{(0)}(\hat{y}_U(x,\tau-1),\tau\text{-}1) - \Phi^{(0)}(\hat{y}_L(x,\tau-1),\tau\text{-}1)\right]\end{aligned} \tag{D.2}$$

which is the second condition required for the modal function $\Phi^{(0)}$. The pressure gradient A_p is assumed to be known at each time step. A computationally useful form of Eq. (D.2) is obtained by expressing $\Phi^{(0)}$ and $D^2\Phi^{(0)}$ in terms of Chebyshev expansions which are evaluated at the boundaries using the Immersed Boundary Conditions concept described in Section 3. If the boundary overlaps with the line l overlapping with the moving boundary $\hat{y} = f(x,\tau+1)$ the required relations take the forms

$$\Phi^{(0)}(f(x,\tau+1),\tau+1) = \sum_{k=0}^{k=N_T} G^{(0)}_{k,\tau+1} \sum_{s=-N_\theta}^{s=+N_\theta} w^{(s)}_{k,\tau+1}\, e^{is\alpha x}, \tag{D.3a}$$

$$D^2\Phi^{(0)}(f(x,\tau+1),\tau+1) = \sum_{k=0}^{k=N_T} G^{(0)}_{k,\tau+1} \sum_{s=-N_\theta}^{s=+N_\theta} r^{(s)}_{k,\tau+1}\, e^{is\alpha x} \tag{D.3b}$$

where,

$$D^2 T_K(f(x,\tau+1)) = \sum_{s=-N_\theta}^{s=+N_\theta} r^{(s)}_{k,\tau+1}\, e^{is\alpha x}\,. \tag{D.3c}$$

Method for evaluation of coefficients $r^{(s)}_{k,\tau+1}$ is explained in Appendix B. Substitution of (D.3) written for the upper and lower walls into Eq.(D.2) results in the final form of the condition being sought.

APPENDIX E

Domain transformation method for the Laplace equation presented in Chapter 4.

The irregular slot geometry in the physical domain (x, y) is mapped into a straight slot in the computational domain (ξ,η) using transformations in the form

$$\xi = x, \quad \eta = [2y - y_U(x) - y_L(x)]/[y_U(x) - y_L(x)]. \tag{E.1}$$

Equation (4.3.1) takes the form

$$\partial_{\eta\eta}\theta + B_1(\xi,\eta)\partial_\eta\theta + B_2(\xi,\eta)\partial_{\xi\eta}\theta + B_3(\xi,\eta)\partial_{\xi\xi}\theta = 0, \tag{E.2}$$

where $B_1(\xi,\eta) = \eta_{xx}/(\eta_x^2+\eta_y^2)$, $B_2(\xi,\eta) = 2\eta_x/(\eta_x^2+\eta_y^2)$ and $B_3(\xi,\eta) = 1/(\eta_x^2+\eta_y^2)$. Expressions for η_x, η_{xx} and η_y are given in Appendix H. Solution of (E.2) is assumed to be in the form of a Fourier expansion, i.e.,

$$\theta(\xi,\eta) = \sum_{n=-\infty}^{n=+\infty} \vartheta^{(n)}(\eta)e^{in\alpha\xi} \approx \sum_{n=-N_M}^{n=+N_M} \vartheta^{(n)}(\eta)e^{in\alpha\xi}, \tag{E.3}$$

where $\vartheta^{(n)} = \vartheta^{(-n)*}$. The known coefficients $B_N(\xi,\eta)$, N=1,2,3, are expressed in terms of Fourier expansions in the form

$$B_N(\xi,\eta) = \sum_{m=-2N_M}^{m=+2N_M} b_N^{(m)}(\eta)e^{im\alpha\xi}. \tag{E.4}$$

Substitution of (E.3) and (E.4) into (E.2) and separation of Fourier components lead to the modal equations in the form

$$D^2\vartheta^{(n)} + \sum_{m=-N_M}^{m=+N_M} [b_1^{(n-m)} + im\alpha b_2^{(n-m)}]D\vartheta^{(m)} - \sum_{m=-N_M}^{m=+N_M} [(m\alpha)^2 b_3^{(n-m)}]\vartheta^{(m)} = 0, \qquad (E.5)$$

requiring two boundary conditions each, where $D=d/d\eta$ and $n=0,\ldots,N_M$. The unknown $\vartheta^{(n)}(\eta)$ can be expressed with spectral accuracy using Chebyshev expansion in the form

$$\vartheta^{(n)}(\eta) = \sum_{k=0}^{k=\infty} E_k^{(n)} T_k(\eta) \approx \sum_{k=0}^{k=N_T} E_k^{(n)} T_k(\eta). \qquad (E.6)$$

Use of the Galerkin procedure described in Section 4.3 leads to N_T+1 algebraic equations for the expansion coefficients $E_k^{(n)}$ for each Fourier mode. Two equations corresponding to the largest k need to be replaced by the boundary conditions (tau procedure).

The boundary conditions in the (ξ,η) plane have the form

$$\theta(\xi,\eta=-1)=C_L, \quad \theta(\xi,\eta=+1)=C_U \qquad (E.7)$$

which lead to boundary conditions for the modal functions in the form

$$\begin{aligned} &\vartheta^{(0)}(-1)=C_L; \quad \vartheta^{(n)}(-1)=0,\ n\neq 0 \\ &\vartheta^{(0)}(+1)=C_U; \quad \vartheta^{(n)}(+1)=0,\ n\neq 0 \end{aligned} \qquad (E.8)$$

Introduction of Chebyshev expansion (E.6) leads to the final form of the boundary condition, i.e.,

$$\sum_{k=0}^{k=N_T} (-1)^k E_k^{(n)} = [\{C_L \text{ for } n=0\}, \{0 \text{ for } n\neq 0\}], \qquad (E.9a)$$

$$\sum_{k=0}^{k=N_T} E_k^{(n)} = [\{C_U \text{ for } n=0\}, \{0 \text{ for } n\neq 0\}]. \qquad (E.9b)$$

APPENDIX F

Domain transformation (DT) method for the biharmonic equation presented in Chapter 4.

Application of transformations given by Eq. (E.1) brings Eq. (4.4.4) or Eq. (5.2.2) into the form

$$\begin{aligned}&\partial_{\eta\eta\eta\eta}\Psi_1 + B_1(\xi,\eta)\partial_{\eta\eta\eta}\Psi_1 + B_2(\xi,\eta)\partial_{\eta\eta}\Psi_1 + B_3(\xi,\eta)\partial_{\eta}\Psi_1 + \\ &B_4(\xi,\eta)\partial_{\xi\eta\eta\eta}\Psi_1 + B_5(\xi,\eta)\partial_{\xi\eta\eta}\Psi_1 + B_6(\xi,\eta)\partial_{\xi\eta}\Psi_1 + B_7(\xi,\eta)\partial_{\xi\xi\eta}\Psi_1 + \\ &B_8(\xi,\eta)\partial_{\xi\xi\eta\eta}\Psi_1 + B_9(\xi,\eta)\partial_{\xi\xi\xi\eta}\Psi_1 + B_{10}(\xi,\eta)\partial_{\xi\xi\xi\xi}\Psi_1 = 0,\end{aligned} \tag{F.1}$$

where

$$B_1(\xi,\eta) = \left(6\eta_x^2\eta_{xx} + 2\eta_{xx}\eta_y^2 + 8\eta_x\eta_y\eta_{xy}\right)/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_2(\xi,\eta) = \left(3\eta_{xx}^2 + 4\eta_x\eta_{xxx} + 4\eta_{xxy}\eta_y + 4\eta_{xy}^2\right)/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_3(\xi,\eta) = \eta_{xxxx}/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_4(\xi,\eta) = \left(4\eta_x^3 + 4\eta_x\eta_y^2\right)/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_5(\xi,\eta) = \left(12\eta_x\eta_{xx} + 8\eta_{xy}\eta_y\right)/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_6(\xi,\eta) = 4\eta_{xxx}/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_7(\xi,\eta) = 6\eta_{xx}/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_8(\xi,\eta) = \left(6\eta_x^2 + 2\eta_y^2\right)/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_9(\xi,\eta) = 4\eta_x/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right),$$

$$B_{10}(\xi,\eta) = 1/\left(\eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4\right), \qquad \text{(F.2a-j)}$$

are known and the expressions for η_x, η_{xx}, η_{xxx}, η_{xxxx}, η_{xy}, η_{xxy} and η_y are given in Appendix H. Definition of the stream function given by Eq. (4.4.2) remains unchanged and thus the velocity components can be expressed as

$$u(\xi,\eta) = u_0(\xi,\eta) + u_1(\xi,\eta) = u_0(\xi,\eta) + \eta_y \partial_\eta \Psi_1,$$

$$v(\xi,\eta) = v_1(\xi,\eta) = -\partial_\xi \Psi_1 - \eta_x \partial_\eta \Psi_1, \qquad \text{(F.3a-b)}$$

where subscript '0' denotes the known reference flow (Poiseuille flow) and subscript '1' denotes modifications due to the presence of boundary irregularities. The boundary conditions can be expressed as

$$\eta_y \partial_\eta \Psi_1(\xi,-1) = -u_0(\xi,-1), \qquad \eta_y \partial_\eta \Psi_1(\xi,+1) = -u_0(\xi,+1),$$

$$\partial_\xi \Psi_1(\xi,-1) + \eta_x \partial_\eta \Psi_1(\xi,-1) = 0, \qquad \partial_\xi \Psi_1(\xi,+1) + \eta_x \partial_\eta \Psi_1(\xi,+1) = 0. \qquad \text{(F.4a-d)}$$

Solution of (F.1) is assumed in terms of Fourier expansion, i.e.,

$$\Psi_1(\xi,\eta) = \sum_{n=-\infty}^{n=+\infty} \phi^{(n)}(\eta) e^{in\alpha\xi} \approx \sum_{n=-N_M}^{n=+N_M} \phi^{(n)}(\eta) e^{in\alpha\xi} \qquad \text{(F.5)}$$

where $\phi^{(n)} = \phi^{(-n)^*}$, and the known coefficients $B_N(\xi,\eta)$, N=1,...,10, are replaced by the Fourier expansions in the form

$$B_N(\xi,\eta) = \sum_{m=-2N_M}^{m=+2N_M} b_N^{(m)}(\eta) e^{im\alpha\xi} . \tag{F.6}$$

Substitution of (F.5) and (F.6) into (F.1) and separation of Fourier components lead to the modal equations in the form

$$\begin{aligned} &D^4\varphi^{(n)} + \sum_{m=-N_M}^{m=+N_M}[b_1^{(n-m)} + im\alpha b_4^{(n-m)}]D^3\varphi^{(m)} + \sum_{m=-N_M}^{m=+N_M}[(m\alpha)^4 b_{10}^{(n-m)}]\varphi^{(m)} + \\ &\sum_{m=-N_M}^{m=+N_M}[b_2^{(n-m)} + im\alpha b_5^{(n-m)} - (m\alpha)^2 b_8^{(n-m)}]D^2\varphi^{(m)} + \\ &\sum_{m=-N_M}^{m=+N_M}[b_3^{(n-m)} + im\alpha b_6^{(n-m)} - (m\alpha)^2 b_7^{(n-m)} - i(m\alpha)^3 b_9^{(n-m)}]D\varphi^{(m)} = 0 \end{aligned} \tag{F.7}$$

where D=d/dη and n=0,...,N_M, with each equation requiring four boundary conditions.

Boundary conditions (F.4) need to be re-arranged into a more suitable form. The known reference flow at the boundaries $u_0(\xi,\pm1)$ can expressed as Fourier expansions in the form

$$u_0(\xi,-1) = \sum_{n=-N_M}^{n=+N_M} \hat{F}_L^{(n)} e^{in\alpha\xi}, \qquad u_0(\xi,+1) = \sum_{n=-N_M}^{n=+N_M} \hat{F}_U^{(n)} e^{in\alpha\xi} . \tag{F.8}$$

Substitution of (F.8) into (F.4a-b) and use of the resulting relations in (B4c-d) results in

$$\partial_\xi \Psi_1(\xi,-1) = (\eta_x/\eta_y)u_0(\xi,-1), \qquad \partial_\xi \Psi_1(\xi,+1) = (\eta_x/\eta_y)u_0(\xi,+1). \tag{F.9a-b}$$

All the terms in the right hand side of (F.9) are known and thus can be expressed using Fourier expansions, i.e.,

$$(\eta_x/\eta_y)u_0(\xi,-1)=\sum_{n=-N_M}^{n=+N_M}\hat{E}_L^{(n)}e^{in\alpha\xi}, \qquad (\eta_x/\eta_y)u_0(\xi,+1)=\sum_{n=-N_M}^{n=+N_M}\hat{E}_U^{(n)}e^{in\alpha\xi}. \quad \text{(F.10a-b)}$$

Boundary conditions for the modal functions $\phi^{(n)}(\eta)$ can now be expressed as

$$D\phi^{(n)}(-1)=-\hat{F}_L^{(n)} \text{ for } |n|\geq 0, \qquad D\phi^{(n)}(+1)=-\hat{F}_U^{(n)} \text{ for } |n|\geq 0,$$

$$\phi^{(n)}(-1)=\hat{E}_L^{(n)}/(in\alpha) \text{ for } n\neq 0, \qquad \phi^{(n)}(+1)=\hat{E}_U^{(n)}/(in\alpha) \text{ for } n\neq 0. \quad \text{(F.11a-d)}$$

The two remaining conditions are derived from the constant volume flux constraint. Volume flux Q along the slot can be evaluated by integrating the x-velocity component across the channel, i.e.,

$$Q(x)=\int_{y_L(x)}^{y_U(x)}u(x,y)dy, \qquad \text{(F.12)}$$

and can be expressed in the (ξ,η) coordinates as

$$Q(\xi)=\int_{-1}^{+1}(\partial_\eta\Psi)d\eta=\Psi_0(\xi,+1)+\Psi_1(\xi,+1)-\Psi_0(\xi,-1)-\Psi_1(\xi,-1) \qquad \text{(F.13)}$$

where the $\Psi_0(\xi,-1)$ and $\Psi_0(\xi,+1)$ are known and can be expressed in terms of Fourier expansions as

$$\Psi_0(\xi,-1)=\sum_{n=-N_M}^{n=+N_M}\Xi_L^{(n)}e^{in\alpha\xi}, \qquad \Psi_0(\xi,+1)=\sum_{n=-N_M}^{n=+N_M}\Xi_U^{(n)}e^{in\alpha\xi}. \qquad \text{(F.14)}$$

The volume flux can be represented in the form of another Fourier expansion

$$Q(\xi)=\sum_{n=-N_M}^{n=+N_M}\hat{Q}^{(n)}e^{in\alpha\xi}\,, \tag{F.15}$$

where the zero term, i.e., $\hat{Q}^{(0)}$, represents the net mass flux along the slot. Substituting (F.5), (F.14) and (F.15) into Eq. (F.13) and extracting mode zero results in

$$Q^{(0)}=\varphi^{(0)}(+1)-\varphi^{(0)}(-1)+\Xi_U^{(0)}-\Xi_L^{(0)}\,. \tag{F.16}$$

As the definition of the stream function is accurate up to a constant, we can select the value of Ψ on one wall arbitrarily. In the present analysis we assume the value of the stream function at the lower wall to be zero which leads to the two closing conditions in the form

$$\varphi^{(0)}(-1)=-\Xi_L^{(0)},\ \varphi^{(0)}(+1)=-\Xi_U^{(0)}+\hat{Q}^{(0)}\,. \tag{F.17a,b}$$

The unknowns $\phi^{(n)}(\eta)$ can be expressed with spectral accuracy using Chebyshev expansions in the form

$$\phi^{(n)}(\eta)=\sum_{k=0}^{k=\infty}S_k^{(n)}T_k(\eta)\approx\sum_{k=0}^{k=N_T}S_k^{(n)}T_k(\eta)\,. \tag{F.18}$$

Use of Galerkin procedure leads to N_T+1 equations for the expansion coefficients $S_k^{(n)}$ for each Fourier mode. Four equations for the highest index k are replaced by the discretized boundary conditions (tau procedure) resulting in a closed system of linear algebraic equations.

APPENDIX G

Domain transformation method for the Navier-Stokes equations presented in Chapter 4.

Application of transformations given in Eq. (E.1) brings Eq. (4.5.4) into the form

$$\begin{aligned}
&\partial_{\eta\eta\eta\eta}\Psi_1 + B_1(\xi,\eta)\partial_{\eta\eta\eta}\Psi_1 + B_2(\xi,\eta)\partial_{\eta\eta}\Psi_1 + B_3(\xi,\eta)\partial_{\eta}\Psi_1 + \\
&\quad B_4(\xi,\eta)\partial_{\xi\eta\eta\eta}\Psi_1 + B_5(\xi,\eta)\partial_{\xi\eta\eta}\Psi_1 + B_6(\xi,\eta)\partial_{\xi\eta}\Psi_1 + B_7(\xi,\eta)\partial_{\xi\xi\eta}\Psi_1 + \\
&\quad B_8(\xi,\eta)\partial_{\xi\xi\eta\eta}\Psi_1 + B_9(\xi,\eta)\partial_{\xi\xi\xi\eta}\Psi_1 + B_{10}(\xi,\eta)\partial_{\xi}\Psi_1 + B_{11}(\xi,\eta)\partial_{\xi\xi\xi}\Psi_1 + \\
&\quad B_{12}(\xi,\eta)\partial_{\xi\xi\xi\xi}\Psi_1 \;=\; L_1(\xi,\eta)\partial_{\xi\eta}\langle N_1\rangle + L_2(\xi,\eta)\partial_{\eta}\langle N_1\rangle + \\
&\qquad L_3(\xi,\eta)\partial_{\eta\eta}\langle N_1\rangle + L_4(\xi,\eta)\partial_{\eta\eta}\langle N_2\rangle + L_5(\xi,\eta)\partial_{\eta}\langle N_2\rangle + \\
&\qquad L_6(\xi,\eta)\partial_{\xi\eta}\langle N_2\rangle + L_5(\xi,\eta)\partial_{\xi\eta}\langle N_2\rangle\,,
\end{aligned} \tag{G.1}$$

where,

$$\langle N_1\rangle = \langle u_1u_1\rangle - \langle v_1v_1\rangle,\quad \langle N_2\rangle = \langle u_1v_1\rangle\,, \tag{G.2a-b}$$

$$B_0(\xi,\eta) = \eta_x^4 + 2\eta_x^2\eta_y^2 + \eta_y^4, \tag{G.3a}$$

$$B_1(\xi,\eta) = \left[6\eta_x^2\eta_{xx} + 2\eta_{xx}\eta_y^2 + 8\eta_x\eta_y\eta_{xy} - u_0 Re\left(\eta_x^3 - \eta_x\eta_y^2\right)\right]/B_0(\xi,\eta), \tag{G.3b}$$

$$B_2(\xi,\eta) = \left[3\eta_{xx}^2 + 4\eta_x\eta_{xxx} + 4\eta_{xxy}\eta_y + 4\eta_{xy}^2 - u_0 Re\left(3\eta_x\eta_{xx} + 2\eta_{xy}\eta_y\right)\right]/B_0(\xi,\eta), \tag{G.3c}$$

$$B_3(\xi,\eta) = \left[\eta_{xxxx} - Re\left(u_0\eta_{xxx} + 2\eta_x\right)\right]/B_0(\xi,\eta), \tag{G.3d}$$

$$B_4(\xi,\eta) = \left(4\eta_x^3 + 4\eta_x\eta_y^2\right)/B_0(\xi,\eta), \tag{G.3e}$$

$$B_5(\xi,\eta) = \left[12\eta_x\eta_{xx} + 8\eta_{xy}\eta_y - u_0 Re\left(3\eta_x^2 + \eta_y^2\right)\right]/B_0(\xi,\eta), \tag{G.3f}$$

$$B_6(\xi,\eta) = \left(4\eta_{xxx} - 3u_0 Re\eta_{xx}\right)/B_0(\xi,\eta), \tag{G.3g}$$

$$B_7(\xi,\eta) = \left(6\eta_{xx} - 3u_0 Re\eta_x\right)/B_0(\xi,\eta), \tag{G.3h}$$

$$B_8(\xi,\eta) = \left(6\eta_x^2 + 2\eta_y^2\right)/B_0(\xi,\eta), \tag{G.3i}$$

$$B_9(\xi,\eta) = 4\eta_x/B_0(\xi,\eta), \tag{G.3j}$$

$$B_{10}(\xi,\eta) = -2/B_0(\xi,\eta), \tag{G.3k}$$

$$B_{11}(\xi,\eta) = -u_0/B_0(\xi,\eta), \tag{G.3l}$$

$$B_{12}(\xi,\eta) = 1/B_0(\xi,\eta), \tag{G.3m}$$

$$L_1(\xi,\eta) = Re\eta_y/B_0(\xi,\eta), \tag{G.4a}$$

$$L_2(\xi,\eta) = Re\eta_{xy}/B_0(\xi,\eta), \tag{G.4b}$$

$$L_3(\xi,\eta) = Re\eta_x\eta_{xy}/B_0(\xi,\eta), \tag{G.4c}$$

$$L_4(\xi,\eta) = Re\left(\eta_y^2 - \eta_x^2\right)/B_0(\xi,\eta), \tag{G.4d}$$

$$L_5(\xi,\eta) = -Re\eta_{xx}/B_0(\xi,\eta), \tag{G.4e}$$

$$L_6(\xi,\eta) = -2Re\eta_x/B_0(\xi,\eta), \tag{G.4d}$$

$$L_7(\xi,\eta) = -Re/B_0(\xi,\eta). \tag{G.4e}$$

The corresponding boundary conditions are given in Eqs. (F.4). We assume solution of problem (G.1) in the form (F.5). Coefficients $B_N(\xi,\eta)$, N=1,...,12 and $L_M(\xi,\eta)$, M=1,...,7 are be replaced by the Fourier expansion in the form

$$B_N(\xi,\eta) = \sum_{m=-2N_M}^{m=+2N_M} b_N^{(m)}(\eta)e^{im\alpha\xi}, \qquad L_M(\xi,\eta) = \sum_{m=-2N_M}^{m=+2N_M} c_M^{(m)}(\eta)e^{im\alpha\xi}. \tag{G.5}$$

The nonlinear terms $\langle N_1\rangle$ and $\langle N_2\rangle$ are also expressed as Fourier expansions, i.e.,

$$\langle N_1\rangle(\xi,\eta) = \sum_{n=-N_M}^{n=+N_M} \gamma_1^{(n)}(\eta)e^{in\alpha\xi}, \qquad \langle N_2\rangle(\xi,\eta) = \sum_{n=-N_M}^{n=+N_M} \gamma_2^{(n)}(\eta)e^{in\alpha\xi}. \tag{G.6}$$

Substitution of (F.5) and (G.5) into (G.1) and separation of Fourier components lead to the modal equations in the form

$$\begin{aligned} D^4\phi^{(n)} + \sum_{m=-N_M}^{m=+N_M} &\left\{ \begin{array}{l} [b_1^{(n-m)} + (im\alpha)b_4^{(n-m)}]D^3\phi^{(m)} + \\ [b_2^{(n-m)} + (im\alpha)b_5^{(n-m)} - (m\alpha)^2 b_8^{(n-m)}]D^2\phi^{(m)} + \\ [b_3^{(n-m)} + (im\alpha)b_6^{(n-m)} - (m\alpha)^2 b_7^{(n-m)} - i(m\alpha)^3 b_9^{(n-m)}]D\phi^{(m)} + \\ [(im\alpha)b_{10}^{(n-m)} - i(m\alpha)^3 b_{11}^{(n-m)} + (m\alpha)^4 b_{12}^{(n-m)}]\phi^{(m)} \end{array} \right\} \\ = \sum_{m=-N_M}^{m=+N_M} &\left\{ \begin{array}{l} [c_2^{(n-m)} + (im\alpha)c_1^{(n-m)}]D\gamma_1^{(m)} + [c_3^{(n-m)}]D^2\gamma_1^{(m)} \\ [c_4^{(n-m)}]D^2\gamma_2^{(m)} + [c_5^{(n-m)} + (im\alpha)c_6^{(n-m)}]D\gamma_2^{(m)} + \\ [-(m\alpha)^2 c_7^{(n-m)}]\gamma_2^{(m)} \end{array} \right\}, \end{aligned} \tag{G.7}$$

where D=d/dη and n=0,...,N_M, with each equation requiring four boundary conditions. Substitution of (F.18) into (G.7) and use of the Galerkin procedure leads to N_T+1 equations for the expansion coefficients $S_k^{(n)}$ for each Fourier mode. The process of

construction of the complete system of equations is completed with the replacement of the four equations corresponding to the largest k by the discretized boundary conditions (tau procedure). The resultant complete system of equations is solved in an iterative manner.

APPENDIX H

Coefficients required for the transformation of the field equations in the domain transformation method for the different model problems described in Chapter 4.

$$\eta_x = -\left[(1+\eta)(y_U)_\xi + (1-\eta)(y_L)_\xi\right]/(y_U - y_L),$$

$$\eta_{xx} = -\left[2\eta_x\{(y_U)_\xi - (y_L)_\xi\} + (1+\eta)(y_U)_{\xi\xi} + (1-\eta)(y_L)_{\xi\xi}\right]/(y_U - y_L),$$

$$\eta_{xxx} = -\left[\begin{matrix}3\eta_{xx}\{(y_U)_\xi - (y_L)_\xi\} + 3\eta_x\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\} + \\ (1+\eta)(y_U)_{\xi\xi\xi} + (1-\eta)(y_L)_{\xi\xi\xi}\end{matrix}\right]/(y_U - y_L),$$

$$\eta_{xxxx} = -\left[\begin{matrix}4\eta_{xxx}\{(y_U)_\xi - (y_L)_\xi\} + 6\eta_{xx}\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\} + \\ 4\eta_x\{(y_U)_{\xi\xi\xi} - (y_L)_{\xi\xi\xi}\} + (1+\eta)(y_U)_{\xi\xi\xi\xi} + (1-\eta)(y_L)_{\xi\xi\xi\xi}\end{matrix}\right]/(y_U - y_L),$$

$$\eta_y = 2/(y_U - y_L),$$

$$\eta_{xy} = -\left[\eta_y\{(y_U)_\xi - (y_L)_\xi\}\right]/(y_U - y_L),$$

$$\eta_{xxy} = -\left[2\eta_{xy}\{(y_U)_\xi - (y_L)_\xi\} + \eta_y\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\}\right]/(y_U - y_L),$$

$$\eta_{xxxy} = -\left[3\eta_{xxy}\{(y_U)_\xi - (y_L)_\xi\} + 3\eta_{xy}\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\} + \eta_y\{(y_U)_{\xi\xi\xi} - (y_L)_{\xi\xi\xi}\}\right]/(y_U - y_L).$$

APPENDIX I

First-, second- and fourth-order implicit temporal discretization for the model problem described in Chapter 6.

Temporal discretization of (6.2.15) using the first-order implicit method results in the following equation

$$\begin{aligned}
&\left[-\mathrm{Re}^{-1}\Gamma^4\hat{D}^4+\left(\Gamma^2\mathrm{Re}^{-1}\Delta t^{-1}+2n^2\alpha^2\Gamma^2\mathrm{Re}^{-1}+in\alpha\Gamma^2u_0\right)\hat{D}^2+\right.\\
&\qquad\left.\left(-n^4\alpha^4\mathrm{Re}^{-1}-n^2\alpha^2\Gamma^2\Delta t^{-1}\mathrm{Re}^{-1}-in\alpha\Gamma^2\hat{D}^2u_0-in^3\alpha^3u_0\right)\right]\left(\Phi_{\tau+1}^{(n)}\right)^{J+1}=\\
&\qquad\left[-in\alpha\Gamma\hat{D}\left(R_1\right)_{\tau+1}^{(n)}-\Gamma^2\hat{D}^2\left(R_2\right)_{\tau+1}^{(n)}-(n\alpha)^2\left(R_2\right)_{\tau+1}^{(n)}+in\alpha\Gamma\hat{D}\left(R_3\right)_{\tau+1}^{(n)}\right]^J\\
&\qquad+\mathrm{Re}^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_\tau^{(n)},\qquad n\in\langle 0,N_M\rangle.
\end{aligned}\tag{I.1}$$

Temporal discretization of (6.2.15) using the second-order implicit method results in the following equation

$$\begin{aligned}
&\left[-\mathrm{Re}^{-1}\Gamma^4\hat{D}^4+\left(3\Gamma^2\mathrm{Re}^{-1}\Delta t^{-1}/2+2n^2\alpha^2\Gamma^2\mathrm{Re}^{-1}+in\alpha\Gamma^2u_0\right)\hat{D}^2+\right.\\
&\qquad\left.\left(-n^4\alpha^4\mathrm{Re}^{-1}-3n^2\alpha^2\Gamma^2\Delta t^{-1}\mathrm{Re}^{-1}/2-in\alpha\Gamma^2\hat{D}^2u_0-in^3\alpha^3u_0\right)\right]\left(\Phi_{\tau+1}^{(n)}\right)^{J+1}=\\
&\qquad\left[-in\alpha\Gamma\hat{D}\left(R_1\right)_{\tau+1}^{(n)}-\Gamma^2\hat{D}^2\left(R_2\right)_{\tau+1}^{(n)}-(n\alpha)^2\left(R_2\right)_{\tau+1}^{(n)}+in\alpha\Gamma\hat{D}\left(R_3\right)_{\tau+1}^{(n)}\right]^J\\
&\qquad+2\mathrm{Re}^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_\tau^{(n)}-0.5\mathrm{Re}^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau-1}^{(n)},\quad n\in\langle 0,N_M\rangle.
\end{aligned}\tag{I.2}$$

Temporal discretization of (6.2.15) using the fourth-order implicit method results in the following equation

$$\begin{aligned}
&\left[-\mathrm{Re}^{-1}\Gamma^4\hat{D}^4+\left(25\Gamma^2\,\mathrm{Re}^{-1}\Delta t^{-1}/12+2n^2\alpha^2\Gamma^2\,\mathrm{Re}^{-1}+in\alpha\Gamma^2u_0\right)\hat{D}^2+\right.\\
&\qquad\left.\left(-n^4\alpha^4\,\mathrm{Re}^{-1}-25n^2\alpha^2\Gamma^2\Delta t^{-1}\,\mathrm{Re}^{-1}/12-in\alpha\Gamma^2\hat{D}^2u_0-in^3\alpha^3u_0\right)\right]\left(\Phi_{\tau+1}^{(n)}\right)^{J+1}=\\
&\qquad\left[-in\alpha\Gamma\hat{D}\left(R_1\right)_{\tau+1}^{(n)}-\Gamma^2\hat{D}^2\left(R_2\right)_{\tau+1}^{(n)}-(n\alpha)^2\left(R_2\right)_{\tau+1}^{(n)}+in\alpha\Gamma\hat{D}\left(R_3\right)_{\tau+1}^{(n)}\right]^J\\
&\qquad+4\mathrm{Re}^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau}^{(n)}-3\mathrm{Re}^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau-1}^{(n)}\\
&\qquad+(4/3)\mathrm{Re}^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau-2}^{(n)}-(1/4)\mathrm{Re}^{-1}\Delta t^{-1}\left[\Gamma^2\hat{D}^2-(n\alpha)^2\right]\Phi_{\tau-2}^{(n)},\\
&\qquad\qquad n\in\langle 0,N_M\rangle.
\end{aligned}\tag{I.3}$$

APPENDIX J

Domain transformation method for the model problem#1 presented in Section 6.4.1 in the moving frame of reference.

The irregular conduit geometry in the physical domain (X,y) is mapped into a straight conduit in the computational domain (ξ,η) using transformations in the form

$$\xi = X, \quad \eta = [2y - y_U(X) - y_L(X)]/[y_U(X) - y_L(X)]. \tag{J.1}$$

Application of transformations given in Eq. (J.1) brings Eq. (6.4.4) into the form

$$\begin{aligned}
&\partial_{\eta\eta\eta\eta}\Psi_1 + B_1(\xi,\eta)\partial_{\eta\eta\eta}\Psi_1 + B_2(\xi,\eta)\partial_{\eta\eta}\Psi_1 + B_3(\xi,\eta)\partial_{\eta}\Psi_1 + \\
&\quad B_4(\xi,\eta)\partial_{\xi\eta\eta\eta}\Psi_1 + B_5(\xi,\eta)\partial_{\xi\eta\eta}\Psi_1 + B_6(\xi,\eta)\partial_{\xi\eta}\Psi_1 + B_7(\xi,\eta)\partial_{\xi\xi\eta}\Psi_1 + \\
&\quad B_8(\xi,\eta)\partial_{\xi\xi\eta\eta}\Psi_1 + B_9(\xi,\eta)\partial_{\xi\xi\xi\eta}\Psi_1 + B_{10}(\xi,\eta)\partial_{\xi}\Psi_1 + B_{11}(\xi,\eta)\partial_{\xi\xi\xi}\Psi_1 + \\
&\quad B_{12}(\xi,\eta)\partial_{\xi\xi\xi\xi}\Psi_1 \;=\; L_1(\xi,\eta)\partial_{\xi\eta}\langle N_1\rangle + L_2(\xi,\eta)\partial_{\eta}\langle N_1\rangle + \\
&\qquad L_3(\xi,\eta)\partial_{\eta\eta}\langle N_1\rangle + L_4(\xi,\eta)\partial_{\eta\eta}\langle N_2\rangle + L_5(\xi,\eta)\partial_{\eta}\langle N_2\rangle + \\
&\qquad L_6(\xi,\eta)\partial_{\xi\eta}\langle N_2\rangle + L_5(\xi,\eta)\partial_{\xi\eta}\langle N_2\rangle\,,
\end{aligned} \tag{J.2}$$

where

$$\langle N_1\rangle = \langle u_1u_1\rangle - \langle v_1v_1\rangle,\ \langle N_2\rangle = \langle u_1v_1\rangle\,, \tag{J.3a-b}$$

$$B_0(\xi,\eta) = \eta_X^4 + 2\eta_X^2\eta_y^2 + \eta_y^4, \tag{J.4a}$$

$$B_1(\xi,\eta) = \left[6\eta_X^2\eta_{XX} + 2\eta_{XX}\eta_y^2 + 8\eta_X\eta_y\eta_{Xy} + (c - u_0Re)\left(\eta_X^3 - \eta_X\eta_y^2\right)\right]/B_0(\xi,\eta), \tag{J.4b}$$

$$B_2(\xi,\eta) = \left[\begin{matrix}3\eta_{XX}^2 + 4\eta_X\eta_{XXX} + 4\eta_{XXy}\eta_y + 4\eta_{Xy}^2 + \\ (c - u_0Re)\left(3\eta_X\eta_{XX} + 2\eta_{Xy}\eta_y\right)\end{matrix}\right]/B_0(\xi,\eta), \tag{J.4c}$$

$$B_3(\xi,\eta) = \left[\eta_{xxxx} + c\eta_{xxx} - Re\left(u_0\eta_{xxx} + 2\eta_x\right)\right]/B_0(\xi,\eta), \tag{J.4d}$$

$$B_4(\xi,\eta) = \left(4\eta_x^3 + 4\eta_x\eta_y^2\right)/B_0(\xi,\eta), \tag{J.4e}$$

$$B_5(\xi,\eta) = \left[12\eta_x\eta_{xx} + 8\eta_{xy}\eta_y + (c - u_0 Re)\left(3\eta_x^2 + \eta_y^2\right)\right]/B_0(\xi,\eta), \tag{J.4f}$$

$$B_6(\xi,\eta) = \left(4\eta_{xxx} + 3(c - u_0\,Re)\eta_{xx}\right)/B_0(\xi,\eta), \tag{J.4g}$$

$$B_7(\xi,\eta) = \left(6\eta_{xx} + 3(c - u_0 Re)\eta_x\right)/B_0(\xi,\eta), \tag{J.4h}$$

$$B_8(\xi,\eta) = \left(6\eta_x^2 + 2\eta_y^2\right)/B_0(\xi,\eta), \tag{J.4i}$$

$$B_9(\xi,\eta) = 4\eta_x/B_0(\xi,\eta), \tag{J.4j}$$

$$B_{10}(\xi,\eta) = -2\,Re/B_0(\xi,\eta), \tag{J.4k}$$

$$B_{11}(\xi,\eta) = (c - u_0\,Re)/B_0(\xi,\eta), \tag{J.4l}$$

$$B_{12}(\xi,\eta) = 1/B_0(\xi,\eta) \quad , \tag{J.4m}$$

$$L_1(\xi,\eta) = Re\,\eta_y/B_0(\xi,\eta), \tag{J.5a}$$

$$L_2(\xi,\eta) = Re\,\eta_{xy}/B_0(\xi,\eta), \tag{J.5b}$$

$$L_3(\xi,\eta) = Re\,\eta_x\eta_{xy}/B_0(\xi,\eta), \tag{J.5c}$$

$$L_4(\xi,\eta) = Re\left(\eta_y^2 - \eta_x^2\right)/B_0(\xi,\eta), \tag{J.5d}$$

$$L_5(\xi,\eta) = -Re\,\eta_{XX}/B_0(\xi,\eta), \tag{J.5e}$$

$$L_6(\xi,\eta) = -2Re\,\eta_X/B_0(\xi,\eta), \tag{J.5d}$$

$$L_7(\xi,\eta) = -Re/B_0(\xi,\eta). \tag{J.5e}$$

In the above

$$\eta_X = -\left[(1+\eta)(y_U)_\xi + (1-\eta)(y_L)_\xi\right]/(y_U - y_L), \tag{J.6a}$$

$$\eta_{XX} = -\left[2\eta_x\{(y_U)_\xi - (y_L)_\xi\} + (1+\eta)(y_U)_{\xi\xi} + (1-\eta)(y_L)_{\xi\xi}\right]/(y_U - y_L), \tag{J.6b}$$

$$\eta_{XXX} = -\begin{bmatrix} 3\eta_{xx}\{(y_U)_\xi - (y_L)_\xi\} + 3\eta_x\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\} + \\ (1+\eta)(y_U)_{\xi\xi\xi} + (1-\eta)(y_L)_{\xi\xi\xi} \end{bmatrix}/(y_U - y_L), \tag{J.6c}$$

$$\eta_{XXXX} = -\begin{bmatrix} 4\eta_{xxx}\{(y_U)_\xi - (y_L)_\xi\} + 6\eta_{xx}\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\} + \\ 4\eta_x\{(y_U)_{\xi\xi\xi} - (y_L)_{\xi\xi\xi}\} + (1+\eta)(y_U)_{\xi\xi\xi\xi} + (1-\eta)(y_L)_{\xi\xi\xi\xi} \end{bmatrix}/(y_U - y_L), \tag{J.6d}$$

$$\eta_y = 2/(y_U - y_L), \tag{J.6e}$$

$$\eta_{Xy} = -\left[\eta_y\{(y_U)_\xi - (y_L)_\xi\}\right]/(y_U - y_L), \tag{J.6f}$$

$$\eta_{XXy} = -\left[2\eta_{xy}\{(y_U)_\xi - (y_L)_\xi\} + \eta_y\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\}\right]/(y_U - y_L), \tag{J.6g}$$

$$\eta_{XXXy} = -\begin{bmatrix} 3\eta_{xxy}\{(y_U)_\xi - (y_L)_\xi\} + 3\eta_{xy}\{(y_U)_{\xi\xi} - (y_L)_{\xi\xi}\} \\ + \eta_y\{(y_U)_{\xi\xi\xi} - (y_L)_{\xi\xi\xi}\} \end{bmatrix}/(y_U - y_L). \tag{J.6h}$$

Definition of the stream function given by Eq. (6.2.8) remains unchanged and thus the velocity components can be expressed as

$$u(\xi,\eta) = u_0(\xi,\eta) + u_1(\xi,\eta) = u_0(\xi,\eta) + \eta_y \partial_\eta \Psi_1 = \eta_y \partial_\eta \Psi_T ,$$

$$v(\xi,\eta) = v_1(\xi,\eta) = -\partial_\xi \Psi_1 - \eta_x \partial_\eta \Psi_1 = -\partial_\xi \Psi_T - \eta_x \partial_\eta \Psi_T . \qquad \text{(J.7a-b)}$$

The boundary conditions for the problem are given by the following relations

$$\eta_y \partial_\eta \Psi_1(\xi,-1) = -u_0(\xi,-1) , \qquad \eta_y \partial_\eta \Psi_1(\xi,+1) = -u_0(\xi,+1) ,$$

$$\partial_\xi \Psi_1(\xi,-1) + \eta_x \partial_\eta \Psi_1(\xi,-1) = (y_L)' \qquad \partial_\xi \Psi_1(\xi,+1) + \eta_x \partial_\eta \Psi_1(\xi,+1) = (y_U)' . \qquad \text{(J.8a-d)}$$

Solution of (J.2) is assumed in terms of a Fourier expansion, i.e.,

$$\Psi_1(\xi,\eta) = \sum_{n=-\infty}^{n=+\infty} \phi^{(n)}(\eta) e^{in\alpha\xi} \approx \sum_{n=-N_M}^{n=+N_M} \phi^{(n)}(\eta) e^{in\alpha\xi} , \qquad \text{(J.9)}$$

where $\phi^{(n)} = \phi^{(-n)^*}$ and star denotes complex conjugates. The coefficients $B_N(\xi,\eta)$, N=1,...,12 and $L_M(\xi,\eta)$, M=1,...,7 are be replaced by the Fourier expansions in the form

$$B_N(\xi,\eta) = \sum_{m=-2N_M}^{m=+2N_M} b_N^{(m)}(\eta) e^{im\alpha\xi} , \qquad L_M(\xi,\eta) = \sum_{m=-2N_M}^{m=+2N_M} g_M^{(m)}(\eta) e^{im\alpha\xi} . \qquad \text{(J.10)}$$

The nonlinear terms $\langle N_1 \rangle$ and $\langle N_2 \rangle$ are also expressed as Fourier expansions, i.e.,

$$\langle N_1 \rangle(\xi,\eta) = \sum_{n=-N_M}^{n=+N_M} \gamma_1^{(n)}(\eta) e^{in\alpha\xi} , \qquad \langle N_2 \rangle(\xi,\eta) = \sum_{n=-N_M}^{n=+N_M} \gamma_2^{(n)}(\eta) e^{in\alpha\xi} . \qquad \text{(J.11)}$$

Substitution of (J.9), (J.10) and (J.11) into (J.2) and separation of Fourier components lead to the modal equations in the form

$$D^4\phi^{(n)} + \sum_{m=-N_M}^{m=+N_M} \begin{Bmatrix} [b_1^{(n-m)} + (im\alpha)b_4^{(n-m)}]D^3\phi^{(m)} + \\ [b_2^{(n-m)} + (im\alpha)b_5^{(n-m)} - (m\alpha)^2 b_8^{(n-m)}]D^2\phi^{(m)} + \\ [b_3^{(n-m)} + (im\alpha)b_6^{(n-m)} - (m\alpha)^2 b_7^{(n-m)} - i(m\alpha)^3 b_9^{(n-m)}]D\phi^{(m)} + \\ [(im\alpha)b_{10}^{(n-m)} - i(m\alpha)^3 b_{11}^{(n-m)} + (m\alpha)^4 b_{12}^{(n-m)}]\phi^{(m)} \end{Bmatrix} \qquad (J.12)$$

$$= \sum_{m=-N_M}^{m=+N_M} \begin{Bmatrix} [g_2^{(n-m)} + (im\alpha)g_1^{(n-m)}]D\gamma_1^{(m)} + [g_3^{(n-m)}]D^2\gamma_1^{(m)} \\ [g_4^{(n-m)}]D^2\gamma_2^{(m)} + [g_5^{(n-m)} + (im\alpha)g_6^{(n-m)}]D\gamma_2^{(m)} + \\ [-(m\alpha)^2 g_7^{(n-m)}]\gamma_2^{(m)} \end{Bmatrix},$$

where $D=d/d\eta$ and $n=0,\ldots,N_M$, with each equation requiring four boundary conditions.

Boundary conditions (J.8) need to be re-arranged into a more suitable form. The known reference flow at the boundaries $u_0(\xi,\pm1)$ can expressed as Fourier expansions in the form

$$u_0(\xi,-1) = \sum_{n=-N_M}^{n=+N_M} \hat{F}_L^{(n)} e^{in\alpha\xi}, \qquad u_0(\xi,+1) = \sum_{n=-N_M}^{n=+N_M} \hat{F}_U^{(n)} e^{in\alpha\xi}. \qquad (J.13a,b)$$

Boundary conditions (J.8c-d) can be re-arranged with the help of (J.8a-b) into the following form

$$\partial_\xi \Psi(\xi,-1) = (\eta_x/\eta_y)u_0(\xi,-1) - c\sum_{n=-N_M,n\neq0}^{n=+N_M} in\alpha H_L^{(n)} e^{in\alpha\xi},$$

$$\partial_\xi \Psi(\xi,+1) = (\eta_x/\eta_y)u_0(\xi,+1) - c\sum_{n=-N_M,n\neq0}^{n=+N_M} in\alpha H_U^{(n)} e^{in\alpha\xi}. \qquad (J.14a-b)$$

All the terms in the right hand side of (J.14) are known and thus can be expanded using Fourier series with known coefficients as

$$(\eta_x/\eta_y)u_0(\xi,-1) - c\sum_{n=-N_M,n\neq 0}^{n=+N_M} in\alpha H_L^{(n)} e^{in\alpha\xi} = \sum_{n=-N_M}^{n=+N_M} \hat{E}_L^{(n)} e^{in\alpha\xi},$$

$$(\eta_x/\eta_y)u_0(\xi,+1) - c\sum_{n=-N_M,n\neq 0}^{n=+N_M} in\alpha H_U^{(n)} e^{in\alpha\xi} = \sum_{n=-N_M}^{n=+N_M} \hat{E}_U^{(n)} e^{in\alpha\xi}. \qquad \text{(J.15a-b)}$$

The boundary conditions can now be expressed in terms of the unknown modal functions $\phi^{(n)}(\eta)$ in the form

$$D\phi^{(n)}(-1) = -\hat{F}_L^{(n)} \text{ for } |n| \geq 0, \qquad D\phi^{(n)}(+1) = -\hat{F}_U^{(n)} \text{ for } |n| \geq 0,$$

$$\phi^{(n)}(-1) = \hat{E}_L^{(n)}/(in\alpha) \text{ for } n \neq 0, \qquad \phi^{(n)}(+1) = \hat{E}_U^{(n)}/(in\alpha) \text{ for } n \neq 0. \qquad \text{(J.16a-d)}$$

Two more conditions are required to close the problem formulation. One condition is arbitrary and is associated with the definition of the stream function. The constant mass flux constraint has been selected as the second condition for this model problem. Equations (J.7) and (J.8) lead to

$$\partial_\xi \Psi_T(\xi,-1) = c\sum_{n=-N_M,n\neq 0}^{n=+N_M} in\alpha H_L^{(n)} e^{in\alpha\xi}, \qquad \text{(J.17)}$$

where $\eta_x \partial_\eta \Psi_T(\xi,-1) = u(\xi,-1) = 0$. Integrating the above equation along ξ results in

$$\Psi_1(\xi,-1) = c\sum_{n=-N_M,n\neq 0}^{n=+N_M} H_L^{(n)} e^{in\alpha\xi} - \Psi_0(\xi,-1) + (\text{const})_L. \qquad \text{(J.18a)}$$

Similar expression for the upper wall takes the form

$$\Psi_1(\xi,+1) = c \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_U^{(n)} e^{in\alpha\xi} - \Psi_0(\xi,+1) + (\text{const})_U . \tag{J.18b}$$

As the definition of stream function is accurate up to a constant, we can arbitrarily assign the value of the stream function to be zero at $\xi = 0$, which results in

$$(\text{const})_L = -c \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_L^{(n)} . \tag{J.19}$$

Volume flux Q along the conduit can be evaluated by integrating the x-velocity component across the conduit, i.e.,

$$Q(X) = \int_{y_L}^{y_U} u(X, y)dy . \tag{J.20}$$

In the (ξ,η) coordinates the expression for the volume flux Q becomes

$$Q(\xi) = \int_{-1}^{+1} (\partial_\eta \Psi)d\eta = \Psi_0(\xi,+1) + \Psi_1(\xi,+1) - \Psi_0(\xi,-1) - \Psi_1(\xi,-1) \tag{J.21}$$

where the $\Psi_0(\xi,-1)$ and $\Psi_0(\xi,+1)$ are known from the solution of the reference Poiseuille flow and can be expressed in terms of Fourier series as

$$\Psi_0(\xi,-1) = \sum_{n=-N_M}^{n=+N_M} \mathcal{P}_L^{(n)} e^{in\alpha\xi} , \qquad \Psi_0(\xi,+1) = \sum_{n=-N_M}^{n=+N_M} \mathcal{P}_U^{(n)} e^{in\alpha\xi} . \tag{J.22}$$

The volume flux can be written in the form of Fourier expansion

$$Q(\xi) = \sum_{n=-N_M}^{n=+N_M} \hat{Q}^{(n)} e^{in\alpha\xi} , \tag{J.23}$$

where the zero term, i.e., $\hat{Q}^{(0)}$, represents the net mass flux along the conduit. Substitution of Eqs (J.22), (J.23) and (J.19) into Eq. (J.21) and extraction of mode zero results in

$$(\text{const})_U = \hat{Q}^{(0)} - c \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_L^{(n)} . \tag{J.24}$$

Substitution of Eqs (J.9), (J.19), (J.22) and (J.24) into Eq. (J.18) and separation of mode zero results in the two closing boundary conditions in the form

$$\phi^{(0)}(-1) = \mathcal{P}_L^{(0)} - c \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_L^{(n)} ,$$

$$\phi^{(0)}(+1) = \mathcal{P}_U^{(0)} - c \sum_{n=-N_M, n\neq 0}^{n=+N_M} H_L^{(n)} + \hat{Q}^{(0)} . \tag{J.25a,b}$$

The unknown $\phi^{(n)}(\eta)$ can be expressed with spectral accuracy using Chebyshev expansion in the form

$$\phi^{(n)}(\eta) = \sum_{k=0}^{k=\infty} S_k^{(n)} T_k(\eta) \approx \sum_{k=0}^{k=N_T} S_k^{(n)} T_k(\eta) . \tag{J.26}$$

Application of Galerkin procedure to Eq. (J.12), as described in Section 6.2, leads to N_T-3 algebraic equations in terms of the unknown coefficients $S_k^{(n)}$ for each Fourier mode. The remaining closing conditions come from discretization of the boundary conditions given by Eqs (J.16) and (J.25). Solution of the complete problem involves an iterative process.

APPENDIX K

Evaluation of expressions for pressure field, pressure loss and friction factor for two-dimensional steady Navier-Stokes problem presented in Sections 4.5 and 7.3.

The total pressure field is given by (4.5.2) as

$$p(x,y) = p_0(x) + p_1(x,y) \tag{K.1}$$

where $p_0(x)=-2x/Re$ is the pressure field associated with the reference Poiseuille flow and $p_1(x,y)$ corresponds to the modification in the pressure field due to boundary irregularities. The dimensionless field equations have the form

$$\frac{\partial u_1}{\partial x} + \Gamma \frac{\partial v_1}{\partial \hat{y}} = 0\,, \tag{K.2a}$$

$$u_1 \frac{\partial u_1}{\partial x} + \Gamma v_1 \frac{\partial u_1}{\partial \hat{y}} + u_0 \frac{\partial u_1}{\partial x} + \Gamma v_1 \frac{du_0}{d\hat{y}} = -\frac{\partial p_1}{\partial x} + \frac{1}{Re}\left[\frac{\partial u_1^2}{\partial x^2} + \Gamma^2 \frac{\partial u_1^2}{\partial \hat{y}^2}\right], \tag{K.2b}$$

$$u_1 \frac{\partial v_1}{\partial x} + \Gamma v_1 \frac{\partial v_1}{\partial \hat{y}} + u_0 \frac{\partial v_1}{\partial x} = -\Gamma \frac{\partial p_1}{\partial \hat{y}} + \frac{1}{Re}\left[\frac{\partial v_1^2}{\partial x^2} + \Gamma^2 \frac{\partial v_1^2}{\partial \hat{y}^2}\right]. \tag{K.2c}$$

The nonlinear terms in Eq. (K.2b) and Eq. (K.2c) can be rearranged by taking advantage of the continuity equation (K.2a) to obtain the following forms

$$\frac{\partial p_1}{\partial x} = \frac{1}{Re}\left[\frac{\partial u_1^2}{\partial x^2} + \Gamma^2 \frac{\partial u_1^2}{\partial \hat{y}^2}\right] - u_0 \frac{\partial u_1}{\partial x} - \Gamma v_1 \frac{du_0}{d\hat{y}} - \frac{\partial}{\partial x}\langle u_1 u_1 \rangle - \Gamma \frac{\partial}{\partial \hat{y}}\langle u_1 v_1 \rangle, \tag{K.3a}$$

$$\Gamma\frac{\partial p_1}{\partial \hat{y}} = \frac{1}{Re}\left[\frac{\partial v_1^2}{\partial x^2} + \Gamma^2\frac{\partial v_1^2}{\partial \hat{y}^2}\right] - u_0\frac{\partial v_1}{\partial x} - \frac{\partial}{\partial x}\langle u_1 v_1\rangle + \Gamma\frac{\partial}{\partial \hat{y}}\langle v_1 v_1\rangle. \qquad \text{(K.3b)}$$

The velocity components u_1 and v_1 are known and can be expressed as

$$u_1(x,\hat{y}) = \Gamma\sum_{n=-N_M}^{n=+N_M} D\Phi^{(n)}(\hat{y})e^{in\alpha x}, \qquad v_1(x,\hat{y}) = -i\alpha\sum_{n=-N_M}^{n=+N_M} n\Phi^{(n)}(\hat{y})e^{in\alpha x}. \qquad \text{(K.4a,b)}$$

The corresponding modification in the pressure field can be expressed as

$$p_1(x,\hat{y}) = A_p\, x + \sum_{n=-N_M}^{n=+N_M} p^{(n)}(\hat{y})e^{in\alpha x} \qquad \text{(K.5)}$$

where A_p represents the pressure correction factor or in other words the mean pressure gradient modification over one wave length of the channel due to the presence of boundary irregularities. A_p appears in the above expression due to the fact that an x-periodic velocity field may be associated with a pressure field that has a linear component in x-direction. The appropriate expression for A_p is unknown and needs to be evaluated.

The products on the right hand side of (K.3a,b) can be expressed as

$$\left[(u_1u_1),(u_1v_1),(v_1v_1)\right](x,\hat{y}) = \sum_{n=-N_M}^{n=+N_M}\left[R_1^{(n)}, R_2^{(n)}, R_3^{(n)}\right](\hat{y})e^{in\alpha x} \qquad \text{(K.6a,b,c)}$$

The unknowns A_p and $p^{(n)}$ need to be evaluated from the momentum equations. We begin with the evaluation of A_p. Substitution of Eqs (K.4-K.6) into the x-momentum equation (K.3a) result in

$$A_p = \frac{1}{Re}\left[\Gamma^3 D^3\ \Phi^{(0)}\right] - \Gamma D R_2^{(0)}, \qquad \text{(K.7a)}$$

$$p^{(n)}(\hat{y}) = (in\alpha)^{-1}\begin{bmatrix}(\Gamma^3 Re^{-1})D^3\Phi^{(n)} + \{-Re^{-1}(n\alpha)^2 - in\alpha\Gamma u_0\}D\Phi^{(n)} \\ +(in\alpha\Gamma\frac{du_0}{d\hat{y}})\Phi^{(n)} - (in\alpha)R_1^{(n)} - \Gamma DR_2^{(n)}\end{bmatrix} \quad \text{for } |n| \neq 0. \qquad \text{(K.7b)}$$

Similar substitutions applied to Eq.(K.3b) lead to

$$\Gamma Dp^{(n)}(\hat{y}) = Re^{-1}\left[-i(n\alpha)^3 - in\alpha\Gamma^2 D^2\right]\Phi^{(n)} - u_0(n\alpha)^2\Phi^{(n)} - in\alpha R_2^{(n)} - \Gamma DR_3^{(n)} \quad \text{for } |n| \geq 0 \qquad \text{(K.8)}$$

Substituting (4.4.6a), (K.4-K.6) into (4.5.4) and equating coefficients of mode zero result in

$$\frac{1}{Re}\left[\Gamma^3 D^4\ \Phi^{(0)}\right] - \Gamma D^2 R_2^{(0)} = 0. \qquad \text{(K.9)}$$

Integration of (K.9) lead to

$$\frac{1}{Re}\left[\Gamma^3 D^3\ \Phi^{(0)}\right] - \Gamma DR_2^{(0)} = \text{constant}. \qquad \text{(K.10)}$$

Comparison of (K.7a) with (K.10) suggests that the pressure correction factor A_p is a constant.

Now we are interested in evaluating the value of $p^{(0)}$. Equating the coefficients of mode zero in (K.8) and subsequent integration lead to

$$p^{(0)} = -R_3^{(0)} + \text{constant}. \qquad \text{(K.11)}$$

The final expression for the complete pressure field has the form

$$p(x,\hat{y}) = \left[A_p - 2/\mathrm{Re}\right]x + \sum_{n=-N_M}^{n=+N_M} p^{(n)}(\hat{y})e^{in\alpha x} + \text{constant} \qquad \text{(K.12)}$$

where the constant can be selected conveniently.

The expression for modification in the pressure field in the presence of surface irregularities is given by

$$p_1(x,\hat{y}) = A_p\, x + \sum_{n=-N_M}^{n=+N_M} p^{(n)}(\hat{y})e^{in\alpha x} + \text{constant}. \qquad \text{(K.13)}$$

The expression for effective pressure gradient over one wave length of the channel in the presence of boundary irregularities is equal to mode zero of the x-derivative of (K.12), i.e.,

$$\frac{\partial p}{\partial x} = (A_p - 2/\mathrm{Re}). \qquad \text{(K.14)}$$

Now, we are interested in evaluating friction factor. The expression for non-dimensional pressure loss is given by

$$\Delta p = \frac{\partial p}{\partial x}\,\Delta L \qquad \text{(K.15)}$$

where ΔL denotes the length of the channel. If L_s, U_s and ρU_s^2 are the length, velocity and pressure scales used for non-dimensionalization then we get,

$$\text{head loss, } h_L = \frac{\Delta\tilde{p}}{\rho} = \left(2\frac{\partial p}{\partial x}\right)\frac{\Delta\tilde{L}}{L_s}\,\frac{U_s^2}{2} \qquad \text{(K.16)}$$

where $\tilde{L}$ and $\tilde{p}$ are dimensional length and pressure. In the above expression the term $2\frac{\partial p}{\partial x}$ is known as the non-dimensional friction factor f corresponding to given L_S and U_s.

We have used half of the original channel height L as the length scale and the maximum velocity at the centre of the channel U_{max} as the velocity scale. For such length and velocity scales the expression for friction factor takes the form

$$f = 2A_p - 4/Re, \qquad \text{(K.17)}$$

where $Re = U_{max}L/\nu$ and ν is the kinematic viscosity of the fluid.

APPENDIX L

Expressions for the operators used in Eq.(7.3.2.11).

$$T^{(m)} = \Gamma^2 D^2 - \{\kappa^{(m)}\}^2 - iRe[\gamma^{(m)} u_0 - \sigma],$$

$$E_v^{(m,n)} = -\mu\Gamma D f_u^{(n)} + \Gamma^2 [in\alpha\mu\{\kappa^{(m-n)}\}^{-2}] f_v^{(n)} D,$$

$$E_\Xi^{(m,n)} = i\gamma^{(m)} f_u^{(n)} + \Gamma[1 + n\alpha\gamma^{(m-n)}\{\kappa^{(m-n)}\}^{-2}] f_v^{(n)} D,$$

$$S^{(m)} = [\Gamma^2 D^2 - \{\kappa^{(m)}\}^2]^2 - iRe[\gamma^{(m)} u_0 - \sigma][\Gamma^2 D^2 - \{\kappa^{(m)}\}^2] + iRe\gamma^{(m)}\Gamma^2 D^2 u_0,$$

$$\begin{aligned} H_v^{(m,n)} = {} & i\{\kappa^{(m)}\}^2 \gamma^{(m-2n)} f_u^{(n)} + i\gamma^{(m)}\Gamma^2 D^2 f_u^{(n)} \\ & + in\alpha\{\kappa^{(m-n)}\}^{-2}[\mu^2 - \gamma^{(m)}\gamma^{(m-n)}]\Gamma^2 D f_u^{(n)} D \\ & + \{\kappa^{(m)}\}^2\{\kappa^{(m-n)}\}^{-2}[\mu^2 + \gamma^{(m-n)}\gamma^{(m-2n)}]\Gamma f_v^{(n)} D \\ & + i\{\kappa^{(m-n)}\}^{-2}[-\{\kappa^{(m-n)}\}^2\gamma^{(m)} + 2n\alpha\mu^2]\Gamma^2 f_u^{(n)} D^2 \\ & + \{\kappa^{(m-n)}\}^{-2}[n\alpha\gamma^{(m)} - \{\kappa^{(m)}\}^2]\Gamma^3 f_v^{(n)} D^3, \end{aligned}$$

$$\begin{aligned} H_\Xi^{(m,n)} = {} & n\alpha\mu\{\kappa^{(m-n)}\}^{-2}[\gamma^{(m)} + \gamma^{(m-n)}]\Gamma D f_u^{(n)} \\ & - in\alpha\mu\{\kappa^{(m)}\}^2\{\kappa^{(m-n)}\}^{-2} f_v^{(n)} \\ & + 2n\alpha\mu\gamma^{(m-n)}\{\kappa^{(m-n)}\}^{-2}\Gamma f_u^{(n)} D \\ & - in\alpha\mu\{\kappa^{(m-n)}\}^{-2}\Gamma^2 f_v^{(n)} D^2. \end{aligned}$$

APPENDIX M

Explicit forms of the nonlinear terms in Eq.(7.2.1.1) and Eq.(7.2.1.2).

$$\langle N_1 \rangle = \frac{\partial H_w}{\partial x} - \frac{\partial H_u}{\partial z},$$

$$\langle N_2 \rangle = \frac{\partial}{\partial y}\left(\frac{\partial H_u}{\partial x} + \frac{\partial H_w}{\partial z}\right) - \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}\right) H_v,$$

where

$$H_u = \frac{\partial}{\partial x}\langle u_1 u_1 \rangle + \frac{\partial}{\partial y}\langle u_1 v_1 \rangle + \frac{\partial}{\partial z}\langle u_1 w_1 \rangle,$$

$$H_v = \frac{\partial}{\partial x}\langle u_1 v_1 \rangle + \frac{\partial}{\partial y}\langle v_1 v_1 \rangle + \frac{\partial}{\partial z}\langle v_1 w_1 \rangle,$$

$$H_w = \frac{\partial}{\partial x}\langle u_1 w_1 \rangle + \frac{\partial}{\partial y}\langle v_1 w_1 \rangle + \frac{\partial}{\partial z}\langle w_1 w_1 \rangle.$$