


Spring 2014

Numerical solutions for problems with complex physics in complex geometry

Yifan Wang

Follow this and additional works at: <https://digitalcommons.latech.edu/dissertations>

 Part of the [Applied Mathematics Commons](#), [Computer Sciences Commons](#), and the [Physics Commons](#)

NUMERICAL SOLUTIONS FOR PROBLEMS WITH
COMPLEX PHYSICS IN COMPLEX GEOMETRY

by

Yifan Wang, B.S., M.S.

A Dissertation Presented in Partial Fulfillment
of the Requirements of the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

May 2014

UMI Number: 3662219

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

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



UMI 3662219

Published by ProQuest LLC 2015. Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

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



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

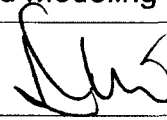
May 18, 2014

Date

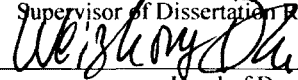
We hereby recommend that the dissertation prepared under our supervision
by Yifan Wang, M.S.

entitled Numerical Solutions for Problems with Complex Physics in Complex Geometry

be accepted in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy in Computational Analysis and Modeling



Supervisor of Dissertation Research

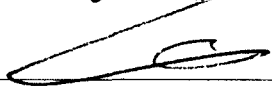
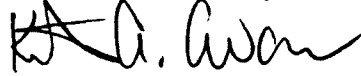
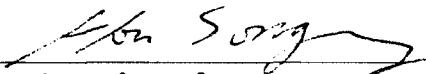


Head of Department

Computational Analysis and Modeling

Department

Recommendation concurred in:



Advisory Committee

Approved:



Director of Graduate Studies

Approved:



Dean of the Graduate School



Dean of the College

ABSTRACT

In this dissertation, two high order accurate numerical methods, Spectral Element Method (SEM) and Discontinuous Galerkin method (DG), are discussed and investigated. The advantages of both methods and their applicable areas are studied. Particular problems in complex geometry with complex physics are investigated and their high order accurate numerical solutions obtained by using either SEM or DG are presented. Furthermore, the Smoothed Particle Hydrodynamics (SPH) (a mesh-free weighted interpolation method) is implemented on graphics processing unit (GPU). Some numerical simulations of the complex flow with a free surface are presented and discussed to show the advantages of SPH method in handling rapid domain deformation.

In particular, four independent numerical examples are sequentially presented. A high-accurate SEM solution to the natural convection problem is provided. Up to the 6th order bases and the 4th order of the Runge-Kutta method are used in the simulation. Results show that our algorithm is more efficient than conventional methods, and the algorithm could obtain very detailed resolutions with moderate computational efforts (simply perform the *hp*-refinement). In another example, a more realistic and complete reaction model of simulating the reaction diffusion process in human neuromuscular junction (NMJ) is developed, and SEM is used to provide a high order accurate numerical solution for the model. Results have successfully predicted the distribution and amount of

open receipts during a normal action potential, which helps us gain a better understanding of this process.

Still, high order DG method is used intensively to study the fluid problems with moderately high Reynolds (Re) number such as: flow passing a vertical cylinder and lid-driven cavity flow in both two dimensional (2D) and three dimensional (3D).

Unstructured meshes (triangular element or tetrahedron) are adopted in our DG solver, which gives a greater ability than structured meshes (quadrilateral element or hexahedron) in solving particular problems with very complex geometry. By comparing our DG results with others obtained by conventional methods (Finite Difference Method, Finite Volume Method), high accuracy similar to other numerical results are obtained; however, the total number of degree of freedom in our simulation is greatly reduced due to the spectral accuracy of the DG method.

Lastly, the SPH method is implemented on GPU to generate 2D and 3D simulations of fluid problems. The SPH solver has an advantage for solving fluid problems with complex geometries, rapid deformations and even discontinuities (wave-break) without generating computational grids. A noticeable speedup of our GPU implementation over the serial version on CPU is achieved. The solver is capable of developing further researches in real engineering problems such as: dam breaks, landslides, and near shore wave propagation and wave-structure interaction.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author WANG YIFAN 王一帆

Date 5/9/2014

DEDICATION

To my beloved wife Qingqing Xu, my dearest daughter Jocelyn Anne Wang and my beloved parents Xuexian Xu, Jie Wang, Hanping Li, and Dongyuan Xu.

TABLE OF CONTENTS

ABSTRACT.....	iii
DEDICATION.....	vi
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
NOMENCLATURE.....	xv
ACKNOWLEDGMENTS.....	xvi
CHAPTER 1 INTRODUCTION.....	1
1.1 Motivation and Background.....	1
1.2 Objectives of the Research.....	3
1.3 Organization of Dissertation.....	4
CHAPTER 2 COMPUTATIONAL METHOD.....	5
2.1 Spectral Element Method.....	5
2.1.1 Basis Function in Quadrilateral/Hexahedron Domain.....	7
2.1.2 Constructing Elemental Matrices.....	9
2.2 Discontinuous Galerkin Method.....	11
2.2.1 Basis Function in Triangle/Tetrahedron Domain.....	11
2.2.2 Constructing Elemental Matrices.....	14
2.2.3 Numerical Flux.....	15
2.3 Smoothed Particle Hydrodynamics.....	16
2.3.1 Weighted Interpolation Representation.....	17
2.3.2 SPH Formulations.....	18

CHAPTER 3 SPECTRAL ELEMENT SIMULATION OF FREE HEAT AND MASS CONVECTION AROUND A CYLINDER WITH CHEMICAL REACTION	19
3.1 Introduction.....	19
3.2 Formulation.....	20
3.3 Nondimensionalization	21
3.4 Numerical Method	22
3.4.1 Spatial Discretization.....	22
3.4.2 Temporal Discretization.....	23
3.5 Numerical Results and Discussions.....	25
3.6 Conclusions.....	30
CHAPTER 4 SPECTRAL ELEMENT SIMULATION OF REACTION-DIFFUSION SYSTEM IN NEUROMUSCULAR JUNCTION	31
4.1 Introduction.....	31
4.2 Formulation.....	32
4.3 Numerical Method	36
4.4 Numerical Results and Discussions.....	40
4.5 Conclusions.....	46
CHAPTER 5 DISCONTINUOUS GALERKIN SIMULATION OF INCOMPRESSIBLE FLOW	48
5.1 Introduction.....	48
5.2 Numerical Method	48
5.2.1 Time Splitting Scheme.....	49
5.2.2 Internal Penalty Flux.....	50
5.2.3 Constructing Data for Visualization	51
5.3 Numerical Results and Discussions.....	52
5.3.1 Two Dimensional Simulation of Flow Passing Cylinder	52

5.3.2	Three Dimensional Simulation of Flow Passing Cylinder	54
5.3.3	Two Dimensional Simulation of Cavity Flow with High Reynolds Number	61
5.3.4	Three Dimensional Simulation of Cavity Flow with High Reynolds Number	67
5.4	Conclusions.....	70
CHAPTER 6 SMOOTHED PARTICLE HYDRODYNAMICS AND GPU COMPUTING.....		72
6.1	Introduction.....	72
6.2	Formulation of Smoothed Particle Hydrodynamics	72
6.2.1	Fluid Particles	72
6.2.2	Solid Particles	73
6.2.3	Particle Interactions	74
6.2.4	Velocity Evaluation	75
6.2.5	Time Evolution and Code Speedup	75
6.3	Numerical Results and Discussions.....	77
6.3.1	Two Dimensional Simulation of Unsteady Nozzle Flow	77
6.3.2	Two Dimensional Simulation of Vortex Shedding.....	79
6.3.3	Two Dimensional and Three Dimensional Simulation of Elastic Solid and Fluid Interaction	82
6.3.4	Simulation Execution Time	86
6.4	Conclusions.....	86
CHAPTER 7 CONCLUSIONS AND FUTURE WORK.....		88
REFERENCE.....		90

LIST OF TABLES

Table 4-1: Values of coefficients in NMJ simulation.....	35
Table 5-1: DG: 2D simulation results compared to reference values.....	54
Table 5-2: DG: 3D simulation results compared to reference values.....	60
Table 6-1: GPU SPH: Execution time (seconds) and specifics of GPU and CPU runs ...	86

LIST OF FIGURES

Figure 2-1: SEM: Solution of 2D Helmholtz equation in a complex domain	6
Figure 2-2: SEM: Exponential convergence vesus algebraic convergence	7
Figure 2-3: SEM: Legendre polynomial as the modal basis function in a standard quadrilateral element.....	8
Figure 2-4: SEM: Lagrangian polynomial as the nodal basis function in a standard quadrilateral element.....	8
Figure 2-5: DG: Jacobi polynomial as the modal basis function in the triangle element.....	12
Figure 2-6: DG: Generating interpolation points in the isosceles' right triangle	13
Figure 2-7: DG: Lagrangian polynomial as the nodal basis function in the triangle element.....	14
Figure 2-8: DG: Comparison of condition number of Vandemonde matrix	15
Figure 3-1: Natural convection simulation: Temporal convergence	26
Figure 3-2: Natural convection simulation: Spatial convergence	27
Figure 3-3: Natural convection simulation: Mesh and contour of velocity u.....	28
Figure 3-4: Natural convection simulation: Contour of temperature and concentration.....	29
Figure 3-5: Natural convection simulation: Local velocity profile	30
Figure 4-1: NMJ simulation: Neuromuscular junction.....	32
Figure 4-2: NMJ simulation: Mesh of the NMJ cleft	33
Figure 4-3: NMJ simulation: Computational mesh at different polynomial orders in the plane $Z = 0$ for NMJ.....	41
Figure 4-4: NMJ simulation: Computational mesh at different polynomial orders in the plane $Y = 0$ for NMJ.....	42

Figure 4-5: NMJ simulation: Time-evolution of Acetylcholine in the diffusion- Reaction system in the plane $Y = 0$	43
Figure 4-6: NMJ simulation: Time-evolution of number of molecules of A2Ropen in NMJ	44
Figure 4-7: NMJ simulation: Time-evolution of concentration of A2Ropen at post membrane, Part 1	45
Figure 4-8: NMJ simulation: Time-evolution of concentration of A2Ropen at post membrane, Part 2	46
Figure 5-1: DG: Generating Tecplot compatible data for visualization	51
Figure 5-2: DG simulation: Schematic of computational domain, 2D	52
Figure 5-3: DG simulation: 2D simulation results of 115 elements and a polynomial order of 5	53
Figure 5-4: DG simulation: 2D simulation results of 307 elements and a polynomial order of 5	53
Figure 5-5: DG simulation: 2D simulation results of 704 elements and a polynomial order of 5	54
Figure 5-6: DG simulation: Schematic of computational domain, 3D	55
Figure 5-7: DG simulation: 3D simulation of flow passing cylinder: Mesh of 2898 elements with a polynomial order of 4 and mesh of 3320 elements and a polynomial order of 3	56
Figure 5-8: DG simulation: 3D simulation of flow passing cylinder: Contour profile of velocity u in the x direction, with 5920 elements and a polynomial order of 3	57
Figure 5-9: DG simulation: 3D simulation of flow passing cylinder: Contour profile of velocity v in the y direction, with 5920 elements and a polynomial order of 3	58
Figure 5-10: DG simulation: 3D simulation of flow passing cylinder: Contour profile of velocity w in the z direction, with 5920 elements and a polynomial order of 3	59
Figure 5-11: DG simulation: 3D simulation of flow passing cylinder: Contour profile of Pressure, with 5920 elements and a polynomial order of 3	60
Figure 5-12: DG simulation: 2D lid-driven cavity flow: Mesh of 385 triangular elements	61

Figure 5-13: DG simulation: 2D lid-driven cavity flow: Streamline with Reynolds number 1000	62
Figure 5-14: DG simulation: 2D lid-driven cavity flow: Velocity v versus X , Reynolds number 1000	63
Figure 5-15: DG simulation: 2D lid-driven cavity flow: Velocity u versus Y , Reynolds number 1000	64
Figure 5-16: DG simulation: 2D lid-driven cavity flow: Streamline with Reynolds number 5000	65
Figure 5-17: DG simulation: 2D lid-driven cavity flow: Velocity v versus X , Reynolds number 5000	66
Figure 5-18: DG simulation: 2D lid-driven cavity flow: Velocity u versus Y , Reynolds number 5000	67
Figure 5-19: DG simulation: 3D lid-driven cavity flow: Mesh of 777 tetrahedron elements	68
Figure 5-20: DG simulation: 3D lid-driven cavity flow with Reynolds number 1000: Velocity contour of u in planes of $Y = -0.4, Y = 0$ and $Y = 0.4$	69
Figure 5-21: DG simulation: 3D lid-driven cavity flow with Reynolds number 1000: Velocity contour of w in planes of $Y = -0.4, Y = 0$, and $Y = 0.4$	70
Figure 6-1: GPU SPH simulation: Schematic of the implementation procedures	76
Figure 6-2: Unsteady nozzle flow: Flow ejected from a nozzle in a thin conical jet, photographed by N. Dombrowski	78
Figure 6-3: Unsteady nozzle flow: Five snapshots of a 2D unsteady nozzle flow	79
Figure 6-4: SPH simulation of Vortex Shedding: Von Kármán vortex street behind a circular cylinder at Reynolds number 140, photographed by Sadatoshi Taneda	80
Figure 6-5: SPH simulation of Vortex Shedding: Simulated laminar 2D vortex street at Reynolds number 140	81
Figure 6-6: SPH simulation of Vortex Shedding: Von Kármán vortex street behind a circular cylinder at Reynolds number 105, photographed by Sadatoshi Taneda	81
Figure 6-7: SPH simulation of Vortex Shedding: Simulated laminar 2D vortex street at Reynolds number 105	82

Figure 6-8: SPH simulation of imping flow: Snapshots of 2D solid-fluid interaction 84

Figure 6-9: SPH simulation of imping flow: Snapshots of 3D solid-fluid interaction 85

NOMENCLATURE

$\mathbf{u} \cdot \mathbf{v}$ or $\mathbf{u} * \mathbf{v}$ Inner Product $\sum_{i=1}^N u_i v_i$

$\mathbf{M} \otimes \mathbf{N}$ Tensor Product of Two Matrices

∇u Gradient (u_x, u_y, u_z)

$\nabla \cdot \mathbf{u}$ Divergence $u_x + v_y + w_z$

$\nabla \times \mathbf{u}$ Curl of Vector Field u

\mathbf{n} Unit Normal Vector $\frac{\nabla u}{|\nabla u|}$

Δt Time Step Size

Ω_{st} Standard Element

$P_n^{(\alpha, \beta)}(x)$ Jacobi Polynomial

$l(\mathbf{x})$ Lagrangian Polynomial

$L(\mathbf{x})$ Legendre Polynomial

f^* Numerical Flux

ACKNOWLEDGMENTS

On the completion of my dissertation, I would like to express my sincere thanks to my professors, family and colleagues.

My gratitude goes first to my advisor, Dr. Don Liu. His insightful guidance and constant encouragement has supported me throughout all aspects of my doctoral education and research. Without his advice and help, I would not have fulfilled my Ph.D. study.

I would like to thank my committee: Dr. Weizhong Dai, Dr. Songming Hou, Dr. Katie A. Evans, and Dr. Box Leangsuksun, and all others in the program of mathematics and statistics for their precious advice and kind help during the last five years.

I am grateful to my beloved family for their sacrifice and their love. They have supported me throughout these years. I also want to take this opportunity to thank all my colleagues and friends: Haibo Zhang and Wenyuan Ma, whom I have been working with closely and have built a deep friendship.

My Ph.D. study is supported by grants to Dr. Liu from the Board of Regents (LEQSF2007-10-RD-A-22: An Efficient Method for Simulating Diffusion Inside Nano-Tubes For Drug Delivery System) and National Science Foundation (DMS-1318988: An Efficient Model for Flows of Large Number of Particles at Moderate to High Reynolds Numbers and Its Applications; DMS-1115546: An Efficient Computational Approach for Wave and Surge Attenuation in Wetlands and Applications in Flood Risk Reduction).

CHAPTER 1

INTRODUCTION

1.1 Motivation and Background

There are many numerical methods available to solve partial differential equations. The most intuitive approach is the Finite Difference Method (FDM), because FDM is easy to derive a high order accuracy scheme and also it is very efficient to approximate high order derivatives. However, for high order accuracy schemes, FDM often requires a large size of stencil. The Compact Difference Method (CDM) [1, 2, 3, 4, 5] offers high order algebraic convergence schemes, which are built on smaller stencils than the Finite Difference Method. However, the linear system obtained via CDM is usually twice or even three times bigger than the one from FDM. Both FDM and CDM methods have a strong reliance on structured meshes, which means that they are not applicable to solve problems with complex geometry. Finite Element Method (FEM), derived from the variation method, is known for its advantage of handling geometrical complexity.

However, FEM is limited in acquiring a higher order of accuracy as the basis functions become mutually dependent once the order of the basis functions increases beyond the fifth and the system matrices become ill-conditioned. Although FEM could achieve a higher resolution by introducing finer elements, rounding errors would accumulate and eventually defeat accuracy at a certain point. Spectral Element Method

(SEM), first appeared in [6], could be a good alternative approach. SEM achieves very high accuracy by utilizing orthogonal polynomial bases and zeros of orthogonal polynomials as quadrature points. SEM is capable of hp -refinement (h stands for increasing the number of elements and p stands for increasing the order of bases), and especially the p -type refinement, which enhances resolution without an extra number of elements [7, 8]. Due to the capability of handling complex geometry and both hp -refinement for high accuracy, SEM has successfully appeared in solving problem involving complex physics and complex geometry, such as computational fluid dynamics [9, 10, 11, 12, 13, 14], simulating microfluidic devices [15, 16, 17], and so on.

However, SEM has a main weak point, and that is, it lacks an upwind treatment like FDM, which causes instable solutions for strong hyperbolic problems (strong advection problem). The Finite Volume Method (FVM) provides a good option by introducing the numerical flux between the elemental boundaries to retain conservation, but FVM has a limitation: it is unable to extend to higher order schemes on unstructured grids, due to the fact that FVM approximates the solution by cell average. To overcome this shortage, another relatively new approach, Discontinuous Galerkin method (DG), which incooperates the idea of numerical flux of FVM into SEM, has been successfully adopted to handle a strong hyperbolic problem [18, 19, 20, 21, 22, 23]. Using appropriate numerical fluxes at the boundaries between elements, DG-SEM is capable of capturing discontinuity in the solution without producing spurious oscillations near the discontinuity. Besides, the elements in DG are discontinuous and the mass matrix is of a block diagonal, which results in a highly parallelizable method.

For some particular problems which involve rapid domain deformations and complex multi-phase interactions, grid based methods become deficient. The particle-based method, Smoothed Particle Hydrodynamics (SPH), does not rely on fixed computation grids and offers a good alternative solution. SPH is capable of handling fluid mechanic problems involving free surface, wave breaking, and rapid geometry distortion, and has been adopted to mimic a variety of problems such as near shore wave-structure interaction [24, 25, 26], dam break, fragmentation or crack growth [27] in mechanical parts, material melting, and materials impact phenomena.

1.2 Objectives of the Research

Our objective is to implement those advanced numerical methods (SEM, DG and SPH) to solve real problems with complex physics and complex geometry. The main goals of this dissertation are addressed:

1. Systematically study the SEM and DG method and conclude their advantages and shortcomings. Develop the general solvers for both methods.
2. Implement the (Smoothed Particle Hydrodynamics) SPH method on the (Graphic Processing Unit) GPU, and develop a GPU based SPH solver.
3. Illustrate the availability and correctness of the SEM solver for particular parabolic and elliptic problems and provide high order numerical solutions to particular engineering problems.
4. Provide numerical examples to illustrate the availability and correctness of the DG solver for particular hyperbolic problems.
5. Provide numerical examples to illustrate the availability and correctness of the GPU based SPH solver for fluid problems.

1.3 Organization of Dissertation

Chapter 1 introduces the motivation and some background about this research work. The research goals and the organization of this dissertation are shown.

Chapter 2 gives a brief literature review covering the knowledge needed for this dissertation: Spectral Element Method, Discontinuous Galerkin Method, and Smoothed Particle Hydrodynamics.

Chapter 3 provides an accurate high order numerical solution to conjugate heat and mass transfer and chemical reaction process around a vertical cylinder in the cylindrical coordinates using SEM. Advantages of SEM over conventional methods are illustrated.

Chapter 4 describes 3D simulation of the reaction diffusion system in the neuromuscular junction using SEM. A full reaction diffusion model with realistic geometry for simulating the neuromuscular junction in the human body is present, and an accurate high order solution is given and discussed.

Chapter 5 demonstrates 2D and 3D DG simulations of fluid problems with a moderately high Reynolds (Re) number. Especially, the flow passing through the cylinder and cavity flow are considered.

Chapter 6 presents 2D and 3D simulations of particular fluidic problems which involve rapid domain deformation and discontinuity by GPU-based SPH method.

Chapter 7 concludes the results of the dissertation, with recommendation for some future works.

CHAPTER 2

COMPUTATIONAL METHOD

2.1 Spectral Element Method

The Spectral Element Method (SEM) was first introduced by [6], and has been further developed later by [7, 8, 28, 29, 30]. The Spectral Element Method, which combines the idea of the Finite Element Method (FEM) and the Galerkin Spectral Method [31], provides both geometry flexibility and spectral accuracy.

Convergence of SEM is achieved either by increasing the degree of the polynomials of the basis function (p -refinement) or by increasing the number of elements (h -refinement). Figure 2-1 shows the mesh and modal SEM solution of 2D Helmholtz equation given an exact solution ($\sin(2\pi x) \sin(2\pi y)$) in a complex domain. The solution is gained by using 32 elements and up to the 10th order of Legendre polynomials. Figure 2-2 illustrates the exponentially fast convergence (p -refinement) in accuracy achieved by SEM over algebraic convergence (h -refinement). Vertical axis represents the point-wise L_2 error in \log -scale, and the horizontal axis represents the polynomial order of the basis function or the total number of elements. We could see that point-wise L_2 error decreases fast in the way of p -refinement than h -refinement.

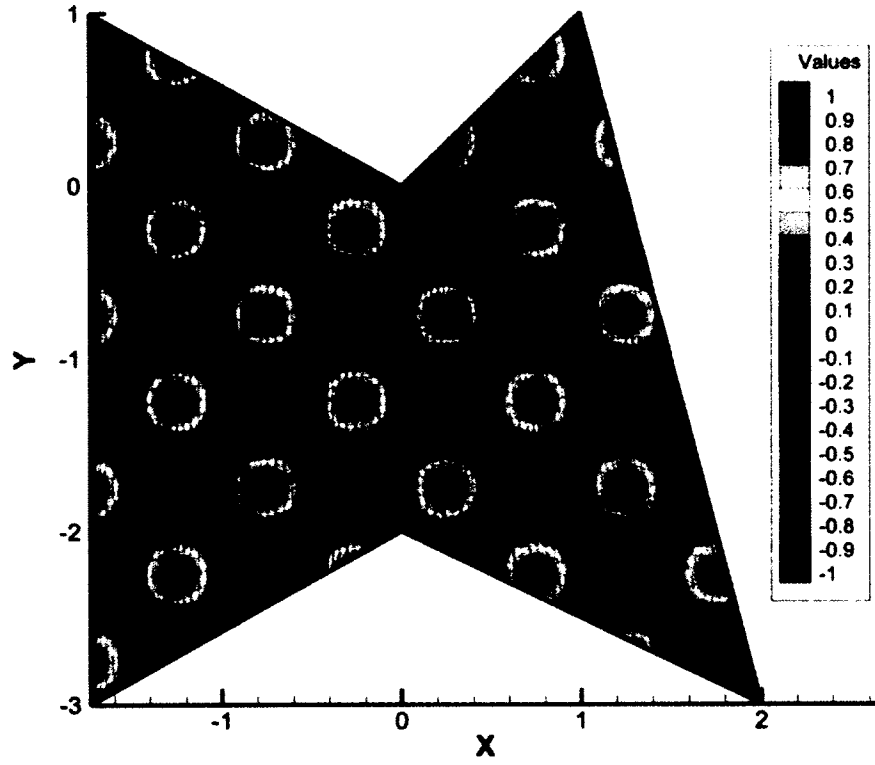


Figure 2-1: SEM: Solution of 2D Helmholtz equation in a complex domain.

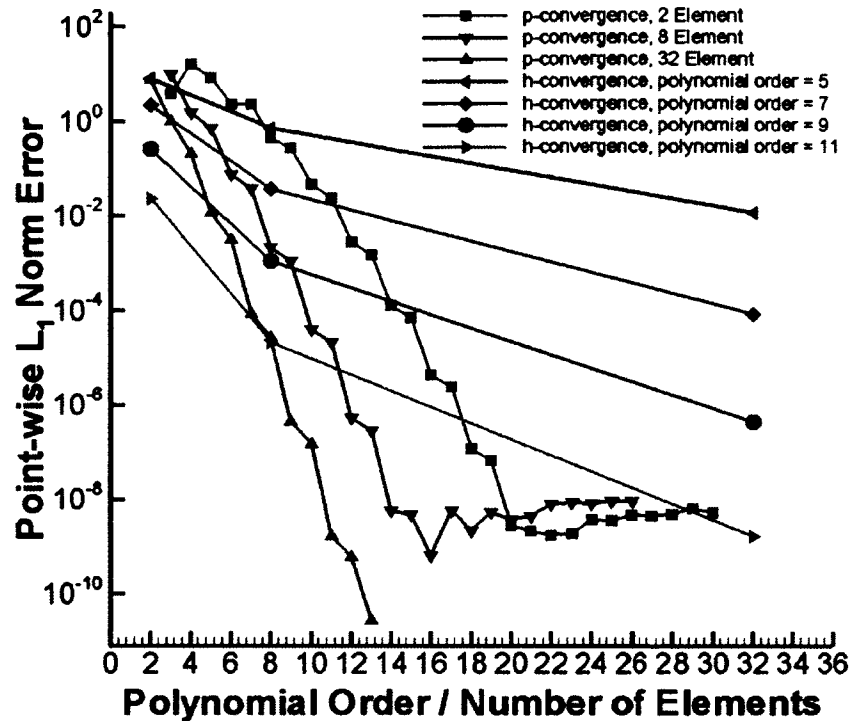


Figure 2-2: SEM: Exponential convergence versus algebraic convergence.

2.1.1 Basis Function in Quadrilateral/Hexahedron Domain

There are two types of bases available in SEM. One is modal bases, shown in Figure 2-3, which are obtained by the tensor product of one dimensional Legendre polynomial in the standard domain ($[-1, 1]$). At the corner of each element, the linear basis function is used as the lift function to maintain the C^0 continuity. The other one is nodal bases, as shown in Figure 2-4, which are obtained from the tensor product of one dimensional Lagrangian polynomial over the zeroes of Legendre polynomial. C^0 continuity is automatically satisfied by utilizing this mode.

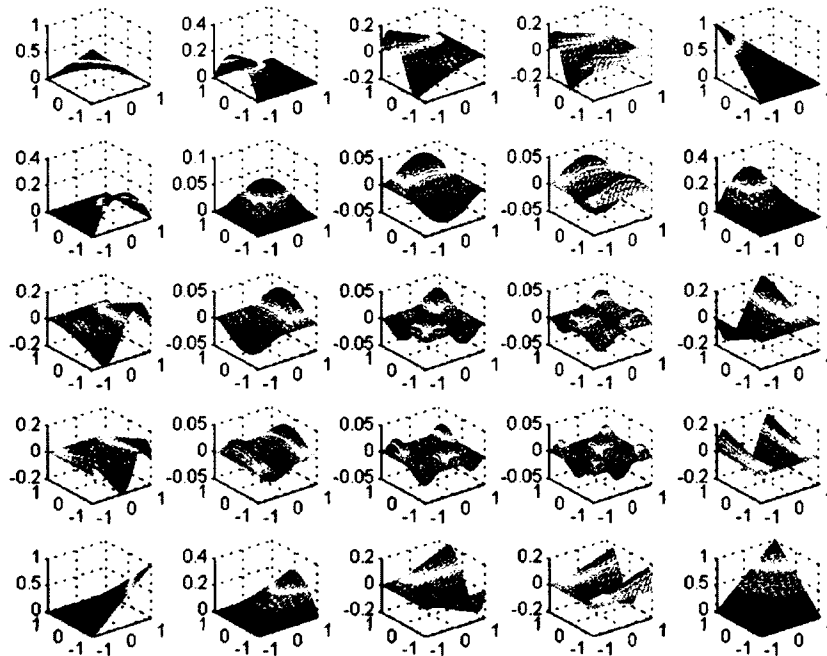


Figure 2-3: SEM: Legendre polynomial as the modal basis function in a standard quadrilateral element.

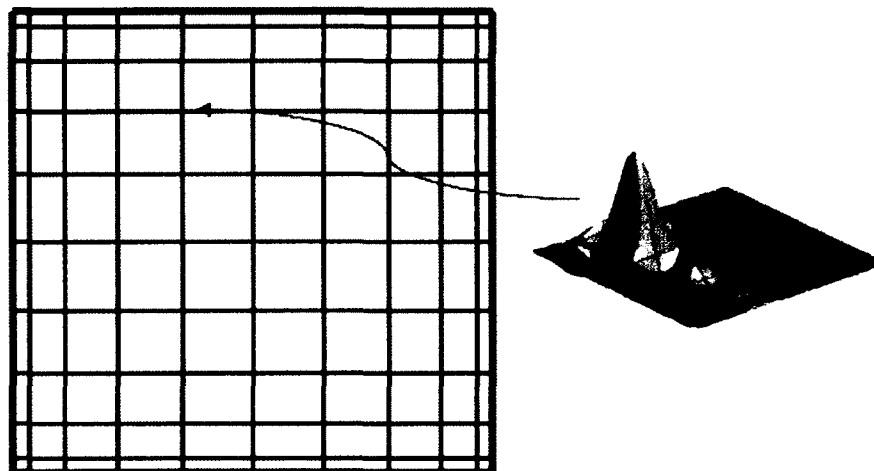


Figure 2-4: SEM: Lagrangian polynomial as the nodal basis function in a standard quadrilateral element.

2.1.2 Constructing Elemental Matrices

In this section, the method of computing mass and stiffness matrices are illustrated respectively. Gauss quadrature is used for integration. For instance, in the 1D case, if the nodal basis (“ $h(\xi)$ ”) is used as bases, then mass matrix is constructed as:

$$\begin{aligned} \underline{\underline{\mathbf{M}}} &= \int_{\Omega_e} \Phi_i(x)\Phi_j(x) dx = \int_{-1}^1 h_i(\xi)h_j(\xi)Jd\xi \\ &= \sum_{k=0}^N \left[w_k \left(\sum_{i=0}^N h_i(\xi_k) \sum_{j=0}^N h_j(\xi_k) \right) J_k \right] = \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k \delta_{ik} \delta_{jk} J_k \right], \end{aligned} \quad (1)$$

and the stiffness matrix is:

$$\begin{aligned} \underline{\underline{\mathbf{K}}} &= \int_{\Omega_e} \Phi'_i(x)\Phi'_j(x) dx = \int_{-1}^1 h'_i(\xi)h'_j(\xi)Jd\xi \\ &= \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k h'_i(\xi_k)h'_j(\xi_k) J_k \right]. \end{aligned} \quad (2)$$

In a quadrilateral element, if the nodal basis (Lagrangian polynomial “ $l(\xi, \eta)$ ”) is chosen as bases, then the bases are expressed as the tensor of 1D Lagrangian $h(\xi)$ and $h(\eta)$ and the local mass matrix is constructed as:

$$\begin{aligned} \underline{\underline{\mathbf{M}}} &= \int_{\Omega_e} \Phi_i(x, y)\Phi_j(x, y) dx dy = \int_{\Omega_{st}} l_i(\xi, \eta)l_j(\xi, \eta)Jd\xi d\eta \\ &= \sum_{k=0}^N \left[w_k \left(\sum_{i=0}^N h_i(\xi_k) \sum_{j=0}^N h_j(\xi_k) \right) J_{\xi, k} \right] \otimes \sum_{k=0}^N \left[w_k \left(\sum_{i=0}^N h_i(\eta_k) \sum_{j=0}^N h_j(\eta_k) \right) J_{\eta, k} \right] \\ &= \left\{ \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k \delta_{ik} \delta_{jk} J_{\xi, k} \right] \right\} \otimes \left\{ \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k \delta_{ik} \delta_{jk} J_{\eta, k} \right] \right\}, \end{aligned} \quad (3)$$

which results in a diagonal matrix by using Gauss integration over Gauss-Lobatto-Legendre quadrature points. ξ and η correspond to the values of x and y from an arbitrary quadrilateral element to the standard quadrilateral element. J stands for the Jacobian of the mapping. The local stiffness matrix is constructed as:

$$\begin{aligned} \underline{\underline{\mathbf{K}}} &= \int_{\Omega_e} \Phi'_i(x, y) \Phi'_j(x, y) dx dy = \int_{\Omega_{st}} l'_i(\xi, \eta) l'_j(\xi, \eta) J d\xi d\eta \\ &= \left\{ \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k h'_i(\xi_k) h'_j(\xi_k) J_{\xi, k} \right] \right\} \otimes \left\{ \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k h'_i(\eta_k) h'_j(\eta_k) J_{\eta, k} \right] \right\}, \end{aligned} \quad (4)$$

which is a full matrix.

If modal basis function “L(ξ, η)” (Tensor of 1D Orthonormalized Legendre Polynomial) is used, then, the local mass matrix is diagonal.

$$\begin{aligned} \underline{\underline{\mathbf{M}}} &= \int_{\Omega_e} \Phi_i(x, y) \Phi_j(x, y) dx dy \\ &= \int_{\Omega_{st}} \left(\frac{1+\xi}{2} \right)^2 \left(\frac{1-\xi}{2} \right)^2 \left(\frac{1+\eta}{2} \right)^2 \left(\frac{1-\eta}{2} \right)^2 L_{i-1}(\xi, \eta) L_{j-1}(\xi, \eta) J d\xi d\eta \\ &= \left\{ \sum_{j=1}^N \sum_{i=1}^N \left[\sum_{k=0}^N w_k \left(\frac{1+\xi_k}{2} \right)^2 \left(\frac{1-\xi_k}{2} \right)^2 \delta_{i-1, j-1} J_{\xi, k} \right] \right\} \otimes \\ &\quad \left\{ \sum_{j=1}^N \sum_{i=1}^N \left[\sum_{k=0}^N w_k \left(\frac{1+\eta_k}{2} \right)^2 \left(\frac{1-\eta_k}{2} \right)^2 \delta_{i-1, j-1} J_{\eta, k} \right] \right\}. \end{aligned} \quad (5)$$

The local stiffness matrix is computed using the derivative property of the Jacobi polynomial:

$$\begin{aligned}
\underline{\underline{\mathbf{K}}} &= \int_{\Omega_e} \Phi'_i(x, y) \Phi'_j(x, y) dx dy \\
&= \int_{\Omega_{st}} \left(\left[\left(\frac{1+\xi}{2} \right) \left(\frac{1-\xi}{2} \right) L_i(\xi) \right]' \left[\left(\frac{1+\xi}{2} \right) \left(\frac{1-\xi}{2} \right) L_j(\xi) \right]' \right) J d\xi d\eta \\
&= \left\{ \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k L'_i(\xi_k) L'_j(\xi_k) J_{\xi,k} \right] \right\} \otimes \left\{ \sum_{j=0}^N \sum_{i=0}^N \left[\sum_{k=0}^N w_k L'_i(\eta_k) L'_j(\eta_k) J_{\eta,k} \right] \right\}. \tag{6}
\end{aligned}$$

2.2 Discontinuous Galerkin Method

In many interesting applications, such as in aeroacoustics, modeling the shallow water, turbulent flows, gas dynamics, and many others which involve strong convection phenomena, SEM becomes less efficient and may fail to provide a good solution. For strong convective problems, discontinuities will eventually develop in the solution even with a smooth initial condition. Due to the lack of upwind treatment in SEM, the numerical solution may be unstable and contains spurious oscillation near the discontinuities. The Discontinuous Galerkin Method is an elementally conservative and high-order numerical method, which combines the idea of numerical flux, slope limiter [32, 33] and the filter of the Finite Volume Method with a Spectral Element Method frame work. It is capable of dealing with the problems with complex geometry over an unstructured mesh and has become the most powerful method of computational fluid dynamics [34, 21, 22, 35].

2.2.1 Basis Function in Triangle/Tetrahedron Domain

Similar to the SEM method, there are two types of bases. In order to handle the numerical fluxes near the elemental boundary easily, only nodal bases are considered in

this dissertation. However, we build the nodal bases with the help of modal bases. Within each triangle element, the following orthonormal polynomial bases are chosen [31, 36]:

$$\Phi_n(r, s) = \sqrt{2}P_i(a)P_j^{(2i+1,0)}(b) (1-b)^i, \quad (7)$$

in which, $a = 2\frac{1+r}{1-s} - 1$, $b = s$; r and s are the coordinates of two short sides of an

isosceles right triangle. $P_n^{(\alpha, \beta)}(x)$ is the n -th order Jacobi polynomial. Figure 2-5 shows the modes below the 5th order.

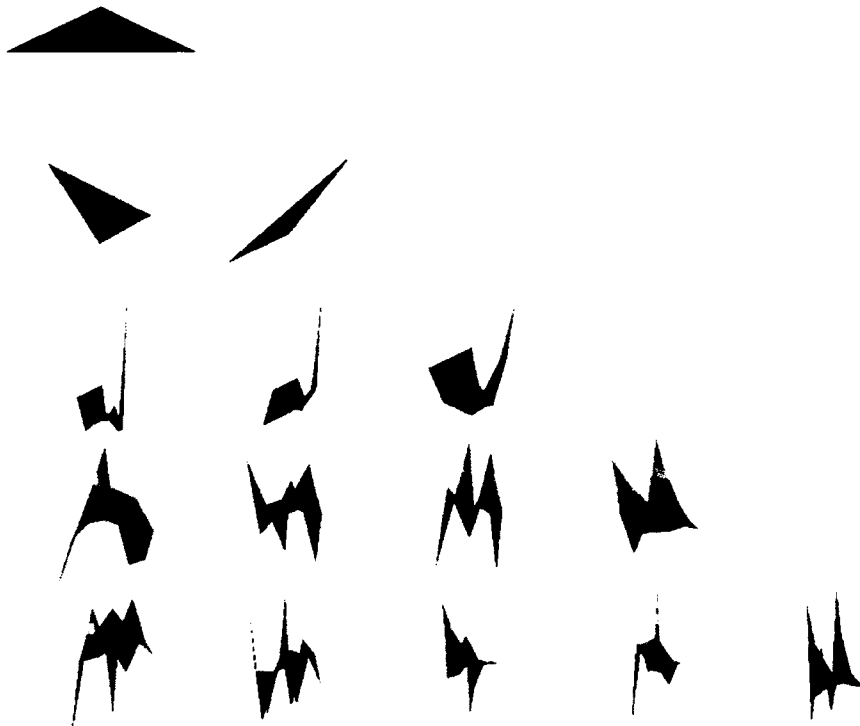


Figure 2-5: DG: Jacobi polynomial as the modal basis function in the triangle element.

To construct the nodal bases, we try to find a group of interpolation points with good interpolation property. Beginning from the barycentric coordinates in the equilateral triangle in Figure 2-6, we relocate those equally spaced points to “Gauss-Lobatto-Legendre” like points along three directions that are parallel to each side of the triangle.

Then we map those points to the isosceles' right triangle using the following relations:

$$(\lambda_1, \lambda_3) = \left(\frac{i}{N}, \frac{j}{N}\right), \lambda_2 = 1 - \lambda_1 - \lambda_3, (i, j) \geq 0, i + j \leq N, \quad (8)$$

$$\begin{pmatrix} r \\ s \end{pmatrix} = \lambda_2 \begin{pmatrix} -1 \\ -1 \end{pmatrix} + \lambda_3 \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \lambda_1 \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

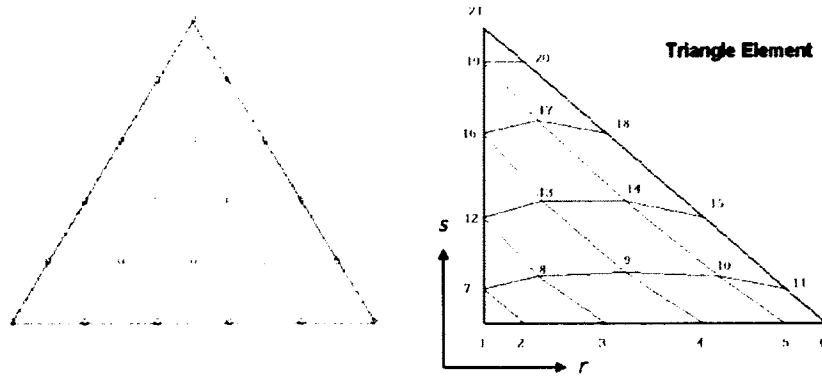


Figure 2-6: DG: Generating interpolation points in the isosceles' right triangle.

If the interpolation points are chosen to be the same points for constructing the Lagrangian polynomials, we could establish a useful connection between the node bases $l(r, s)$ and modal bases $\varphi(r, s)$ through the Vandermonde matrix V . These connections will eventually help us to construct the elemental matrices of the DG solver:

$$V_{ij} = \Phi_{j-1}(r, s)_i, \quad (9)$$

$$\underline{V}^T \underline{l}(r, s) = \underline{\Phi}(r, s).$$

In Vandermonde matrix, subscript $j - 1$ represents the order of the basis function, and i is the index of the interpolation points r and s .

Figure 2-7 shows the nodal modes of the 5th order. For each mode, it has a unit value at the interpolating point of that mode while zero at the rests.

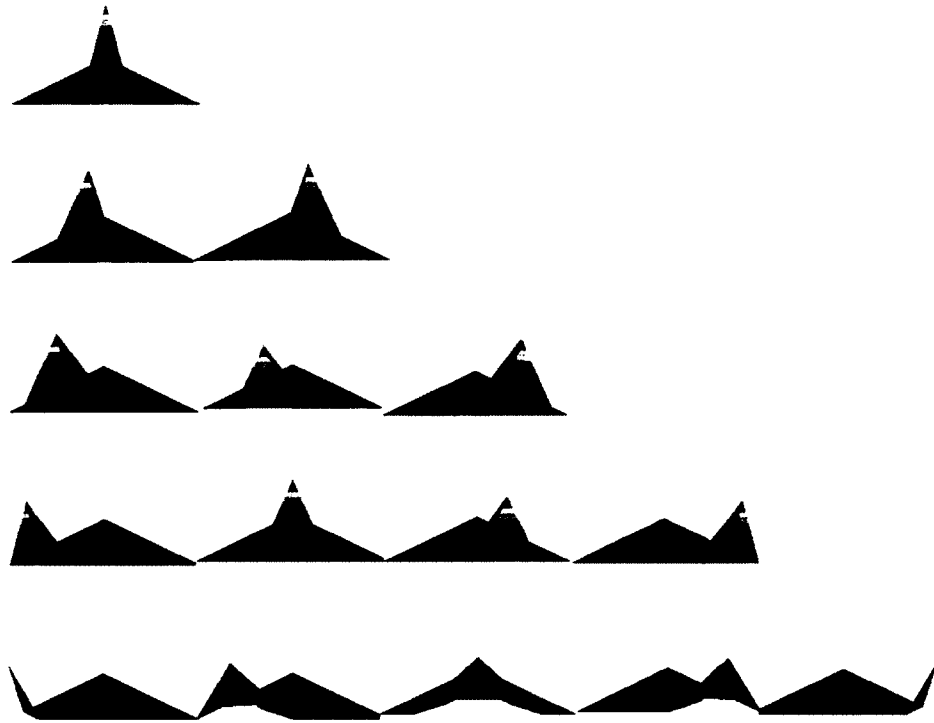


Figure 2-7: DG: Lagrangian polynomial as the nodal basis function in the triangle element.

2.2.2 Constructing Elemental Matrices

Since the nodal basis (Lagrangian polynomial “ l ”) is used as the bases in triangle element, the local mass matrix is constructed as:

$$\begin{aligned}
 \underline{\underline{\mathbf{M}}} &= \int_{\Omega_e} \Phi_i(r, s) \Phi_j(r, s) dr ds = \int_{\Omega_{st}} l_i(\xi, \eta) l_j(\xi, \eta) J d\xi d\eta \\
 &= \sum_{i=1}^N \sum_{j=1}^N \left[\int_{\Omega_{st}} \left(\sum_{n=1}^N (V^T)_{in}^{-1} \varphi_{n-1}(\xi, \eta) \sum_{m=1}^N (V^T)_{jm}^{-1} \varphi_{m-1}(\xi, \eta) \right) J d\xi d\eta \right] \quad (10) \\
 &= \sum_{i=1}^N \sum_{j=1}^N \left[\sum_{n=1}^N (V^T)_{in}^{-1} (V^T)_{jn}^{-1} J \right],
 \end{aligned}$$

which is a diagonal matrix by using Gauss integration over Gauss-Lobatto-Legendre quadrature points. Figure 2-8 compares the condition number of Vandemonde matrix constructed by Barycentric coordinates and “Gauss Lobatto Legendre” like points, which clearly shows the advantage of using those points.

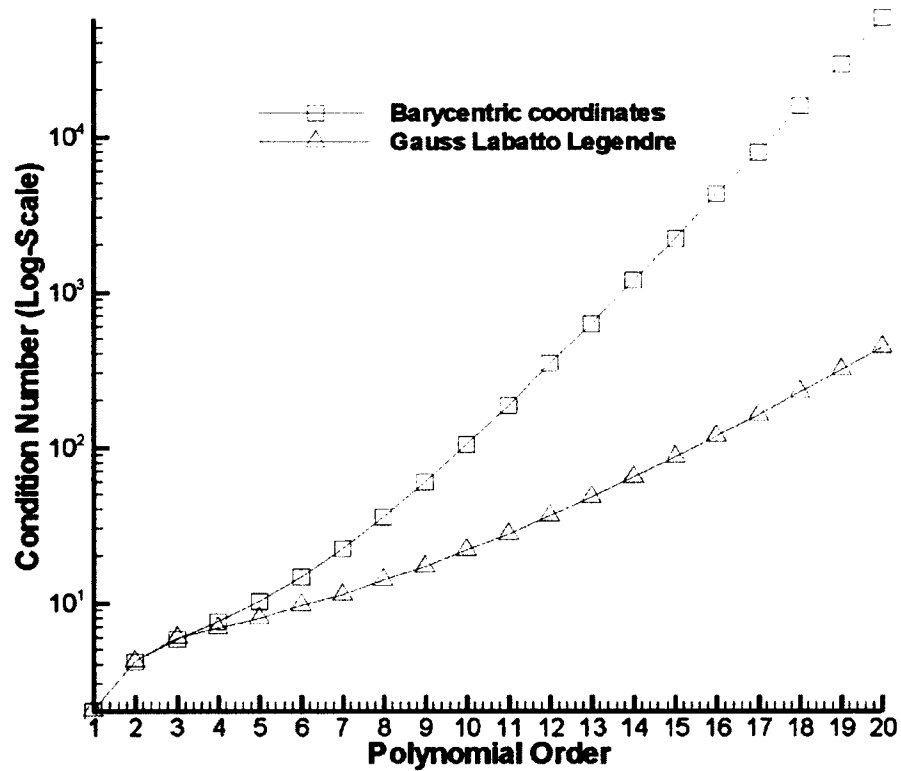


Figure 2-8: DG: Comparison of condition number of Vandemonde matrix.

2.2.3 Numerical Flux

There are various numerical fluxes [37] we can choose from, particularly in our DG solver, and the Lax-Friedrichs flux [36] is used to satisfy the stability:

$$\mathbf{f}^*(a, b) = \frac{\mathbf{f}(a) + \mathbf{f}(b)}{2} + \frac{K}{2} \hat{\mathbf{n}}(a - b), \quad (11)$$

where, $K = \max \left| \hat{n}_x \frac{\partial f_x}{\partial u} + \hat{n}_y \frac{\partial f_y}{\partial u} \right|$ and $\mathbf{f} = (f_x, f_y)$.

2.3 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) was first proposed by Gingold and Monaghan [38] and Lucy [39] for astrophysics problems, and later it was developed by Monaghan [40] and Liu et al. [41] for problems in fluid dynamics and solid mechanics. Originally, the method was mainly for solving compressible flows but had tensile instability and inconsistency. Over the last two decades, Monaghan [42, 43, 44] and Liu et al. [41] have further developed and improved SPH such that it has wider applications to incompressible flow, fluid-solid interaction, solid mechanics, and explosion simulation, etc.

Being a mesh-free weighted interpolation method, SPH is especially effective for complex problems with large domain distortion and complex physics. SPH has recently been extensively used to solve hydrodynamics and solid mechanics problems. Different from conventional mesh-based methods such as Finite Difference, Finite Volume, Finite Element, and Spectral Element Methods, SPH does not rely on any fixed computational grid. This gives rise to the capability of SPH in easily handling fluid mechanics problems involving free surface, wave breaking, and rapid geometry distortion. For solid mechanics problems, it is straightforward and intuitive for SPH to mimic atoms as particles. SPH is capable of handling large distortion in solid mechanics, and it has been adopted to mimic a variety of problems such as near shore wave-structure interaction [45, 25, 24, 26, 46], dam break, fragmentation or crack growth in mechanical parts [47, 48, 49], material melting, and materials impact phenomena [50, 51, 52].

2.3.1 Weighted Interpolation Representation

SPH is a weighted interpolation method. SPH represents all bulk properties of the fluid and solid body at a certain location in space with a discrete interpolation over a set of surrounding particles [40, 41]. The interpolating function used in SPH is called the kernel function and denoted as $W(\mathbf{r} - \mathbf{r}', h)$, where h is the radius of the influence region around the position \mathbf{r}' . The kernel function is similar to a delta function with the following properties:

$$\int W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' = 1, \quad (12)$$

$$\lim_{h \rightarrow 0} W(\mathbf{r} - \mathbf{r}', h) = \delta(\mathbf{r} - \mathbf{r}'). \quad (13)$$

Using a kernel function, a certain property of interest of the particle, such as I , at the location \mathbf{r} can be expressed as [40]:

$$\begin{aligned} I(\mathbf{r}) &= \int I(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' = \int \frac{I(\mathbf{r}')}{\rho(\mathbf{r}')} W(\mathbf{r} - \mathbf{r}', h) \rho(\mathbf{r}') d\mathbf{r}' \\ &\approx \sum_{\mathbf{r}'} m(\mathbf{r}') \frac{I(\mathbf{r}')}{\rho(\mathbf{r}')} W(\mathbf{r} - \mathbf{r}', h), \end{aligned} \quad (14)$$

in which, the summation is over all neighbor particles within the influence region. To compute the gradient of I at the location \mathbf{r} , we use the following approximation after performing the integration by parts:

$$\nabla I \approx \sum_{\mathbf{r}'} m(\mathbf{r}') \frac{I(\mathbf{r}')}{\rho(\mathbf{r}')} \nabla W(\mathbf{r} - \mathbf{r}', h). \quad (15)$$

The surface integral term is dropped, since the kernel function and I both go to zero by definition. Similarly, higher order spatial derivatives could always be presented as a weighted summation of derivatives of the kernel functions.

2.3.2 SPH Formulations

There are two types of SPH methods for fluids: the weakly compressible SPH (WCSPH) and the incompressible SPH (ISPH) [53, 27]. Compared with the weighted residual methods, SPH dramatically simplified the procedures of numerically solving Navier-Stokes equations. First, SPH simplifies the nature of the Navier-Stokes equations in that SPH is a Lagrangian method and thus the non-linear convective terms in Navier-Stokes equations disappear. Second, the momentum and energy equations could be solved explicitly in time, and there is no need to invert large linear systems any more. However, WCSPH and ISPH treat the density and pressure differently. Since the fluid is treated as a group of particles, WCSPH computes the fluid density with a weighted summation over all neighboring particles using certain kernel functions within the influence radius. Hence, the mass conservation is automatically satisfied as the total number of particles maintained. This saves the effort for solving the equation of mass conservation. Moreover, the pressure is directly computed from the equation of state for both gas and liquid, as the pressure depends on density only. In contrast, ISPH solves the mass conservation equation to obtain the density and solves a pressure Poisson equation to acquire the pressure. In this dissertation, WCSPH method is used for generating our solver.

CHAPTER 3

SPECTRAL ELEMENT SIMULATION OF FREE HEAT AND MASS CONVECTION AROUND A CYLINDER WITH CHEMICAL REACTION

3.1 Introduction

This chapter is based on my contribution to the publication titled "Spectral nodal element simulation of conjugate heat and mass transfer: Natural convection subject to chemical reaction along a circular cylinder [54]". I am the first author of this paper and its content is used in Chapter 3 with proper referencing.

Since the 1990s, heat and mass transfer through porous mediums has been of interest to researchers and widely researched and applied to the industry [55] such as migration of mass and energy through porous media [56, 57], and transfer of moisture through dehumidifying materials [58, 59, 60]. Conjugate heat and mass transfer involving vertical cylinders has occurred in various engineering areas [61, 62, 63] such as solar energy retrieving, food sciences [64], and biological materials [65].

In this chapter, nodal Spectral Element Method and the 4th order Runge-Kutta method are used to solve the conjugate heat and mass transfer problem coupled with chemical reaction within the free convection boundary layer in cylindrical coordinates. The numerical results show the profile of this natural convection phenomena and illustrate the efficiency of the algorithm.

3.2 Formulation

We consider the problem that a mass of quiescent fluid, in which the chemical reaction takes place, embraces a semi-infinite vertical cylinder with uneven distributed surface heat and mass fluxes. By the boundary layer approximation and Boussinesq approximation [66], we have the following governing equations for this boundary layer problem of free convection in cylindrical coordinates [63] as:

$$\frac{\partial(ru)}{\partial x} + \frac{\partial(rv)}{\partial r} = 0, \quad (16)$$

$$\frac{\partial u}{\partial t'} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial r} = g\beta_T(T' - T'_\infty) + g\beta_C(C' - C'_\infty) \quad (17)$$

$$+ \frac{\mu}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right),$$

$$\frac{\partial T'}{\partial t'} + u \frac{\partial T'}{\partial x} + v \frac{\partial T'}{\partial r} = \frac{\alpha}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T'}{\partial r} \right), \quad (18)$$

$$\frac{\partial C'}{\partial t'} + u \frac{\partial C'}{\partial x} + v \frac{\partial C'}{\partial r} = \frac{D}{r} \frac{\partial}{\partial r} \left(r \frac{\partial C'}{\partial r} \right) - KC'. \quad (19)$$

In which, μ is viscosity, α is thermal diffusivity, D is mass diffusivity, β_T and β_C are thermal expansion coefficient and concentration expansion coefficient, respectively, and K is reaction coefficient. For free convection, the pressure term is substituted by $-g\beta_T T'_\infty - g\beta_C C'_\infty$ from Boussinesq approximation [66], and the second order derivative terms with respect to x are dropped from boundary layer approximation.

By performing the nondimensionalization, we have:

$$R \frac{\partial U}{\partial X} + R \frac{\partial V}{\partial R} + V = 0, \quad (20)$$

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial X} + V \frac{\partial U}{\partial R} = \frac{T}{r_0} + \frac{NC}{r_0} + \frac{1}{R} \frac{\partial U}{\partial R} + \frac{\partial^2 U}{\partial R^2}, \quad (21)$$

$$\frac{\partial T}{\partial t} + U \frac{\partial T}{\partial X} + V \frac{\partial T}{\partial R} = \frac{1}{PrR} \frac{\partial T}{\partial R} + \frac{1}{Pr} \frac{\partial^2 T}{\partial R^2}, \quad (22)$$

$$\frac{\partial C}{\partial t} + U \frac{\partial C}{\partial X} + V \frac{\partial C}{\partial R} = \frac{1}{ScR} \frac{\partial C}{\partial R} + \frac{1}{Sc} \frac{\partial^2 C}{\partial R^2} - kC. \quad (23)$$

Where U , V , T , and C are the axial velocity of the cylinder, radial velocity of the cylinder, temperature and concentration, respectively. r_0 is the radius of the cylinder. Pr is the Prandtl number, Sc is the Schmidt number, and N is the coefficient of buoyancy ratio of the temperature over the concentration. The problem is well-posed with the following initial and boundary conditions:

1). All variables U , V , T , and C are set at zero initially.

2). On cylindrical surface $0 \leq X \leq 15$:

$$t > 0: U = 0, V = 0, q_T = \lambda \frac{\partial T}{\partial R} = -X^{0.5}, q_C = D \frac{\partial C}{\partial R} = -X^{0.5}.$$

3). $U = 0, T = 0, C = 0$ at $X = 0, R > R_0$.

4). $U \rightarrow 0, T \rightarrow 0, C \rightarrow 0$ as $R \rightarrow \infty$.

3.3 Nondimensionalization

Within the boundary layer, the radius of the cylinder r_0 is chosen as the length scale, and the diffusive velocity scale, $\mu/(\rho r_0)$ is chosen to be the velocity scale. q_T and q_C are the variable heat and mass fluxes. We summarize these scalings as follows:

$$X = \frac{x}{r_0}, R = \frac{r}{r_0}, U = \frac{\rho u r_0}{\mu}, V = \frac{\rho v r_0}{\mu}, t = \frac{\mu'}{\rho r_0^2}, T = \frac{(T' - T'_\infty)\lambda}{q_T},$$

$$T = \frac{(C' - C'_\infty)D}{q_C}, Pr = \frac{\mu}{\rho \alpha}, Sc = \frac{\mu}{\rho D}, N = \frac{\beta_C q_C \lambda}{\beta_T q_T D},$$

3.4 Numerical Method

3.4.1 Spatial Discretization

In space, Galerkin projections are implemented to Eqs. (20)-(23) to obtain the variational forms. Then spectral element discretization was employed. For an arbitrary element in the cylindrical coordinates, denoted as “ e ”, we map it to a standard element in ξ_0 and ξ_1 , which are bounded from -1 to 1 . Particularly, for solution of U , we expand it in terms of a tensor product of two one dimensional Lagrangian basis functions (denoted as h) on quadrature points, which are chosen to be zeros of Legendre polynomials:

$$U^e(\xi_0, \xi_1) = \sum_p^{P_0} \sum_q^{P_0} U_{pq}^e h_p^e(\xi_0) h_q^e(\xi_1) = \sum_{pq} U_{pq}^e \Phi_{pq}^e, \quad (24)$$

where P_0 is the highest polynomial order of the basis functions. Because of the collocation property of the nodal Spectral Element Method, coefficients U_{pq}^e in the above equation are the numerical solutions at those quadrature points. The test function in a standard element is chosen to be the same as the basis function:

$$\Phi_{rs}^e = \sum_r^{P_0} \sum_s^{P_0} h_r^e(\xi_0) h_s^e(\xi_1). \quad (25)$$

We set up the following short notations to particular matrices:

Global mass matrix:

$$\underline{\underline{M}} = \sum_e^{Total} \left[\int_{\Omega_e} \Phi_{pq}^e \Phi_{rs}^e |J|_e d\xi_0 d\xi_1 \right]. \quad (26)$$

Global Advection matrix:

$$\underline{\underline{Adv_X}} = \sum_e^{Total} \left[\int_{\Omega_e} (\Phi_{pq}^e)'_{\xi_0} \Phi_{rs}^e \left(\frac{2}{L_R} \right) |J|_e d\xi_0 d\xi_1 \right]. \quad (27)$$

$$\underline{\underline{Adv_R}} = \sum_e^{Total} \left[\int_{\Omega_e} (\Phi_{pq}^e)'_{\xi_1} (\Phi_{rs}^e)'_{\xi_1} \left(\frac{2}{L_X} \right) |J|_e d\xi_0 d\xi_1 \right]. \quad (28)$$

Global Laplacian matrix:

$$\underline{\underline{K}} = \sum_e^{Total} \left[\int_{\Omega_e} (\Phi_{pq}^e)'_{\xi_0} (\Phi_{rs}^e)'_{\xi_0} \frac{L_X}{L_R} d\xi_0 d\xi_1 + \int_{\Omega_e} (\Phi_{pq}^e)'_{\xi_1} (\Phi_{rs}^e)'_{\xi_1} \frac{L_R}{L_X} d\xi_0 d\xi_1 \right], \quad (29)$$

where J is the Jacobian for the transformation corresponding to element “e”. As in our simulation, each element is rectangular, and L_X, L_R represent the length in X and R direction of element “e”, respectively.

3.4.2 Temporal Discretization

At the initial time step, values of U_0, V_0, T_0 , and C_0 are known. For Eqs. (21)-(23), we could treat them as a system of initial-value problem by moving the convective terms on the left hand side to the right hand side of the equations, and then use the 4th order of the Runge-Kutta method to solve the system of partial differential equations. Let us assume k_{1U}, k_{1T} and k_{1C} stand for the first stage coefficients of the 4th order of the Runge-Kutta method, respectively. Then we have:

$$\underline{\underline{M}}k_{1U} = \Delta t \left[-U_0 \cdot (\underline{\underline{Adv_X}} U_0) - V_0 \cdot (\underline{\underline{Adv_R}} U_0) + \underline{\underline{M}} \frac{T_0}{r_0} + \underline{\underline{M}} \frac{NC_0}{r_0} + \frac{1}{R} (\underline{\underline{Adv_R}} U_0) + \underline{\underline{M}} U_X \Big|_{\Omega} - \underline{\underline{K}} U_0 \right], \quad (30)$$

$$\underline{\underline{M}}k_{1T} = \Delta t \left[-U_0 \cdot (\underline{\underline{Adv_X}} T_0) - V_0 \cdot (\underline{\underline{Adv_R}} T_0) + \frac{1}{PrR} (\underline{\underline{Adv_R}} T_0) + \frac{1}{Pr} \underline{\underline{M}} T_X \Big|_{\Omega} - \frac{1}{Pr} \underline{\underline{K}} T_0 \right], \quad (31)$$

$$\begin{aligned} \underline{\underline{Mk}}_{1C} = \Delta t \left[-U_0 \cdot \underline{\underline{Adv}}_X(C_0) - V_0 \cdot \underline{\underline{Adv}}_R(C_0) + \frac{1}{ScR} \underline{\underline{Adv}}_R(C_0) \right. \\ \left. + \frac{1}{Sc} \underline{\underline{MC}}_X \Big|_{\Omega} - \frac{1}{Sc} \underline{\underline{KC}}_0 - k \underline{\underline{MC}}_0 \right] \end{aligned} \quad (32)$$

where Ω represents the boundary of the problem. Intermediate value of V_i is computed from the following equation by substituting in the intermediate value of U_i , which is from $U_0 + k_{1U}$.

$$\underline{\underline{Adv}}_R + \frac{M}{R} V_i = \underline{\underline{Adv}}_X U_i. \quad (33)$$

Then, k_{1V} can be computed from $V_i - V_0$.

For computing the value of k_{2U} , k_{2T} and k_{2C} :

$$\underline{\underline{Mk}}_{2U} = \Delta t \left[\begin{aligned} & -(U_0 + \frac{1}{2} k_{1U}) \cdot \underline{\underline{Adv}}_X(U_0 + \frac{1}{2} k_{1U}) - (V_0 + \frac{1}{2} k_{1V}) \cdot \underline{\underline{Adv}}_R(U_0 + \frac{1}{2} k_{1U}) \\ & + \frac{M}{r_0} \frac{(T_0 + \frac{1}{2} k_{1T})}{r_0} + \frac{M}{r_0} \frac{N(C_0 + \frac{1}{2} k_{1C})}{r_0} + \frac{1}{R} \underline{\underline{Adv}}_R(U_0 + \frac{1}{2} k_{1U}) + \underline{\underline{MU}}_X \Big|_{\Omega} \\ & - \underline{\underline{K}}(U_0 + \frac{1}{2} k_{1U}) \end{aligned} \right] \quad (34)$$

$$\underline{\underline{Mk}}_{2T} = \Delta t \left[\begin{aligned} & -(U_0 + \frac{1}{2} k_{1U}) \cdot \underline{\underline{Adv}}_X(T_0 + \frac{1}{2} k_{1T}) - (V_0 + \frac{1}{2} k_{1V}) \cdot \underline{\underline{Adv}}_R(T_0 + \frac{1}{2} k_{1T}) \\ & + \frac{1}{PrR} \underline{\underline{Adv}}_R(T_0 + \frac{1}{2} k_{1T}) + \frac{1}{Pr} \underline{\underline{MT}}_X \Big|_{\Omega} - \frac{1}{Pr} \underline{\underline{K}}(T_0 + \frac{1}{2} k_{1T}) \end{aligned} \right] \quad (35)$$

$$\underline{\underline{Mk}}_{2C} = \Delta t \left[\begin{aligned} & -(U_0 + \frac{1}{2} k_{1U}) \cdot \underline{\underline{Adv}}_X(C_0 + \frac{1}{2} k_{1C}) - (V_0 + \frac{1}{2} k_{1V}) \cdot \underline{\underline{Adv}}_R(C_0 + \frac{1}{2} k_{1C}) \\ & + \frac{1}{ScR} \underline{\underline{Adv}}_R(C_0 + \frac{1}{2} k_{1C}) + \frac{1}{Sc} \underline{\underline{MC}}_X \Big|_{\Omega} - \frac{1}{Sc} \underline{\underline{K}}(C_0 + \frac{1}{2} k_{1C}) - k \underline{\underline{M}}(C_0 + \frac{1}{2} k_{1C}) \end{aligned} \right] \quad (36)$$

Similar procedures as Eqs. (34)-(36) are performed to compute k_{3U} , k_{3T} and k_{3C} ,

while k_{4U} , k_{4T} and k_{4C} are computed by following equations:

$$\underline{\underline{Mk}}_{4U} = \Delta t \left[\begin{array}{l} -(U_0 + k_{3U}) \cdot [\underline{\underline{Adv}}_X(U_0 + k_{3U})] - (V_0 + k_{3V}) \cdot [\underline{\underline{Adv}}_R(U_0 + k_{3U})] \\ + \frac{\underline{\underline{M}}(T_0 + k_{3T})}{r_0} + \frac{\underline{\underline{M}}N(C_0 + k_{3C})}{r_0} + \frac{1}{R} [\underline{\underline{Adv}}_R(U_0 + k_{3U})] + \underline{\underline{M}}U_X \Big|_{\Omega} \\ - \underline{\underline{K}}(U_0 + k_{1U}) \end{array} \right], \quad (37)$$

$$\underline{\underline{Mk}}_{4T} = \Delta t \left[\begin{array}{l} -(U_0 + k_{3U}) \cdot [\underline{\underline{Adv}}_X(T_0 + k_{3T})] - (V_0 + k_{3V}) \cdot [\underline{\underline{Adv}}_R(T_0 + k_{3T})] \\ + \frac{1}{PrR} [\underline{\underline{Adv}}_R(T_0 + k_{3T})] + \frac{1}{Pr} \underline{\underline{M}}T_X \Big|_{\Omega} - \frac{1}{Pr} \underline{\underline{K}}(T_0 + k_{3T}) \end{array} \right], \quad (38)$$

$$\underline{\underline{Mk}}_{4C} = \Delta t \left[\begin{array}{l} -(U_0 + k_{3U}) \cdot [\underline{\underline{Adv}}_X(C_0 + k_{3C})] - (V_0 + k_{3V}) \cdot [\underline{\underline{Adv}}_R(C_0 + k_{3C})] \\ + \frac{1}{ScR} [\underline{\underline{Adv}}_R(C_0 + k_{3C})] + \frac{1}{Sc} \underline{\underline{M}}C_X \Big|_{\Omega} - \frac{1}{Sc} \underline{\underline{K}}(C_0 + k_{3C}) - k \underline{\underline{M}}(C_0 + k_{3C}) \end{array} \right]. \quad (39)$$

After every coefficient for all four stages of Runge-Kutta scheme are determined, the value of U at the next time level is calculated by following the formula:

$$U_1 = U_0 + \frac{1}{6} (k_{1U} + 2k_{2U} + 2k_{3U} + k_{4U}). \quad (40)$$

Following the same formula, the values of T and C can be computed, where V is calculated by U from Eq. (33).

3.5 Numerical Results and Discussions

Before discussing the simulation results, we first perform temporal and spatial convergence tests to ensure the convergence and correctness of the algorithm. Figure 3-1 shows the results of the temporal convergence test. We discretize the computational domain into 20 elements, and use the 6th order Lagrangian polynomial as the basis function. A 4th order temporal convergence is achieved by varying the time step as 0.0000125, 0.000025, 0.00005, and 0.0001, respectively. The result agrees well with the theoretical expectation, because the 4th order of the Runge-Kutta method is used to treat the temporal derivative.

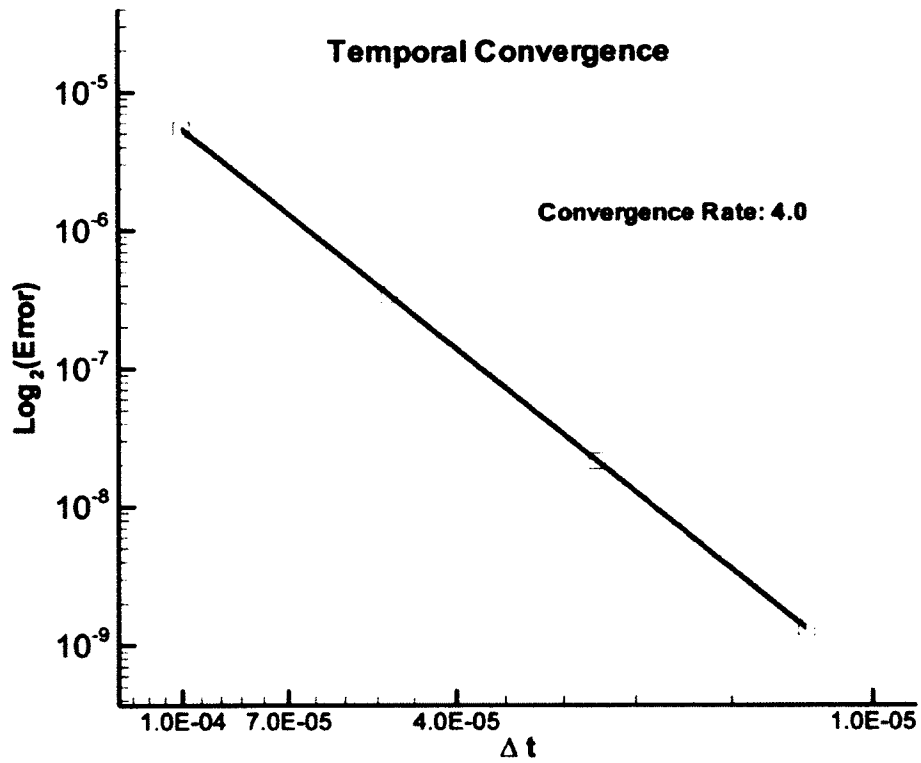


Figure 3-1: Natural convection simulation: Temporal convergence.

Figure 3-2 illustrates the exponential convergence in space. By increasing the polynomial order of the basis function, the point wise L_2 norm error decreases exponentially. Both spatial and temporal convergence indicate that the algorithm is compatible and correct.

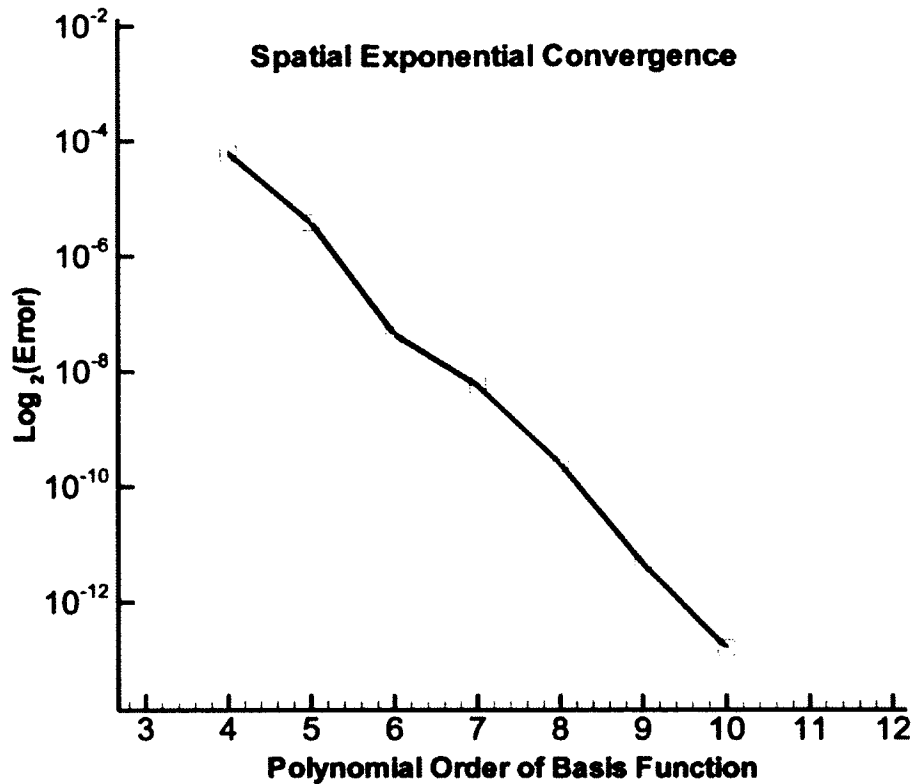


Figure 3-2: Natural convection simulation: Spatial convergence.

Figure 3-3 and Figure 3-4 present the numerical results of the free convection of conjugate heat and mass transfer with chemical reaction around a vertical cylinder. Sub graph of Figure 3-3 on the left shows the mesh we generated for this problem. A total of 20 elements are used and the 8th order of the basis function for each direction in each element is adopted. A graph of Figure 3-3 on the right is the velocity profile of U . At the cylinder's surface the velocity is equal to zero due to the non-slip boundary condition and a boundary layer is formed near the cylinder surface from free convection. Velocity increases in the boundary layer and then decreases away from the cylinder's surface. Figure 3-4 reveals the distributions of temperature and concentration from the cylinder's surface to the region away from the cylinder, respectively. We could see that the

distributions of temperature and concentration mainly located within the free convection boundary layer.

Outside the boundary layer the temperature and concentration drop to background temperature and concentration. Since in our simulation, we choose $Sc = 0.7$ (Methanol) and $Pr = 7.0$ (Water), which assumes the mass diffusion rate is larger than the thermal diffusion rate. Therefore, the thickness of the mass boundary layer is greater than the thickness of the thermal boundary layer. Figure 3-5 presents the local magnitude and direction of the velocity of the fluid. For better resolution, we zoom in the bottom portion of the whole domain with x chosen from 0 to 6. Figure 3-5 tells that the fluid is driven from the lower side to the upper side due to the uneven distribution of heat and mass flux.

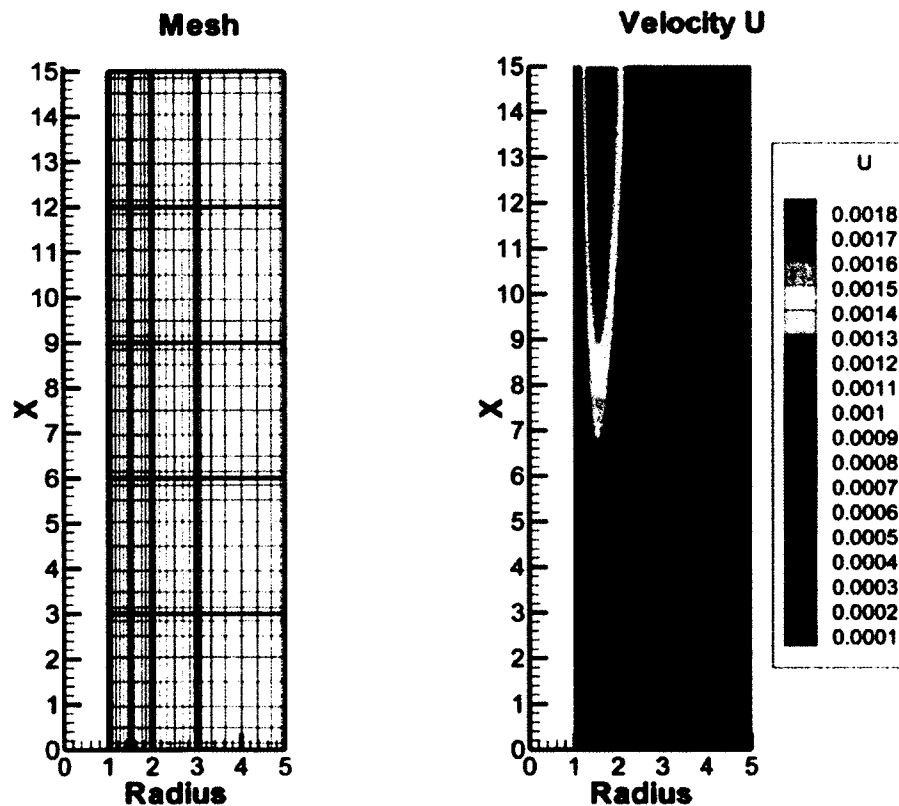


Figure 3-3: Natural convection simulation: Mesh and contour of velocity u .

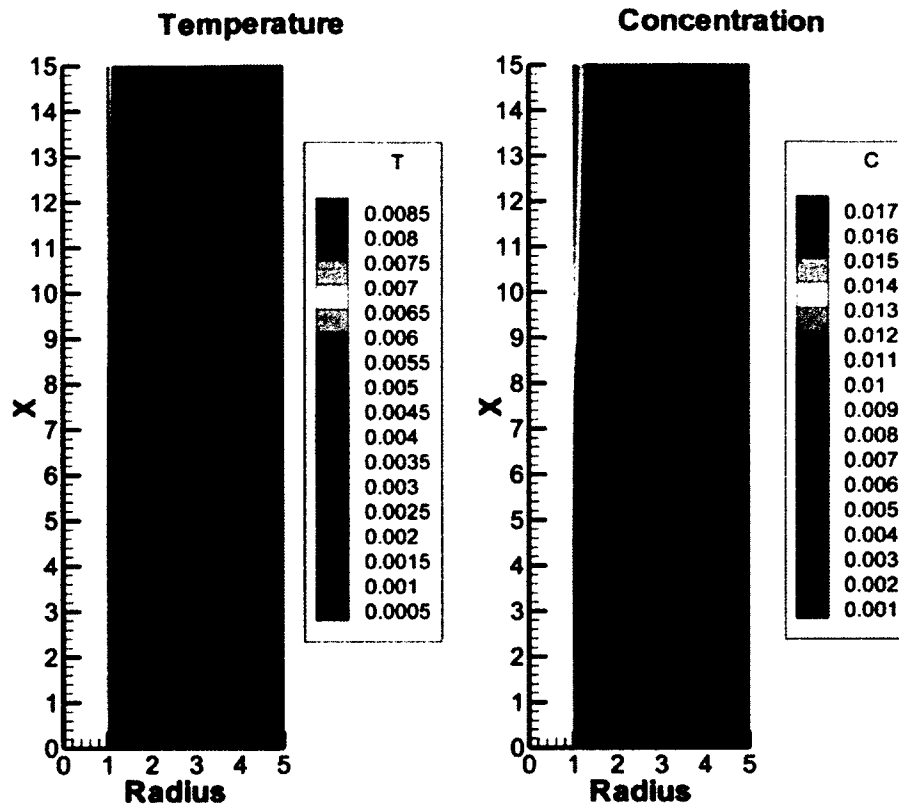


Figure 3-4: Natural convection simulation: Contour of temperature and concentration.

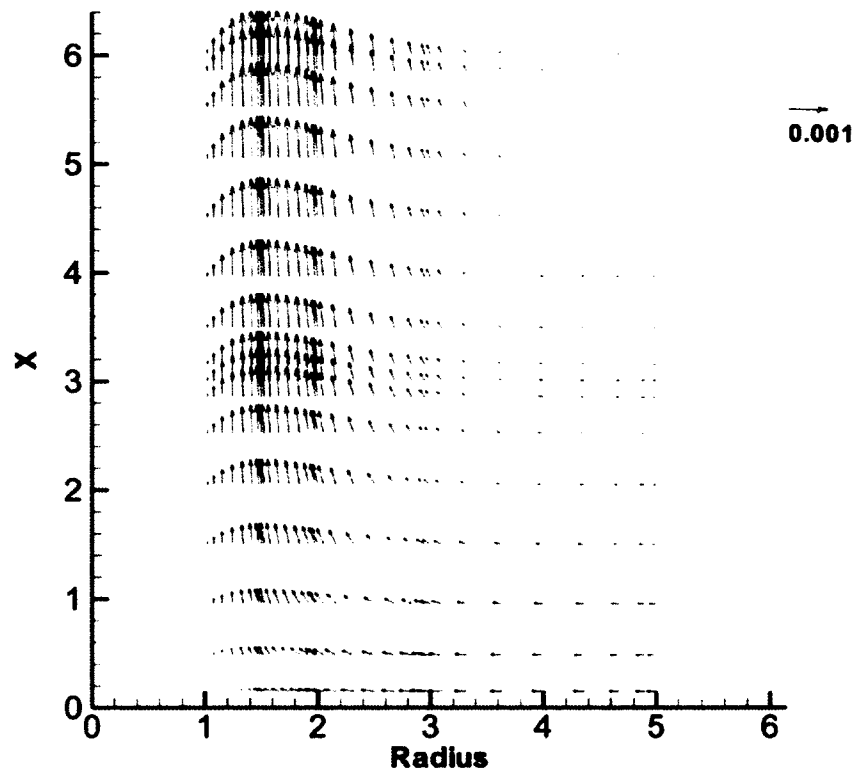


Figure 3-5: Natural convection simulation: Local velocity profile

3.6 Conclusions

Free heat and mass convection coupled with chemical reaction around a semi infinite vertical cylinder is examined in this chapter. The computational domain is simplified by using cylindrical coordinates, and the problem is solved by nodal spectral element method and the 4th order of the Runge-Kutta method. The results indicate the effectiveness and correctness of our algorithm for solving conjugate heat and mass transfer problem. This algorithm may be further used to help us understand phenomenon details of some engineering problem involving mass and heat transfer, such as in solar energy retrieving, food sciences and biological materials.

CHAPTER 4

SPECTRAL ELEMENT SIMULATION OF REACTION-DIFFUSION SYSTEM IN NEUROMUSCULAR JUNCTION

4.1 Introduction

This chapter is based on my contribution to the publication titled "Spectral element simulation of reaction-diffusion system in the neuromuscular junction [67]." I am the first author of this paper and its content is used in Chapter 4 with proper referencing.

Studying the synaptic signal transmission in the neuromuscular junction (NMJ) is central to the understanding of neuromuscular disorders such as myasthenia gravis disease. Investigating the dynamics of Acetylcholine and Acetylcholine receptors in the NMJ under the conditions of activated enzyme is an important step towards this mission. In this article, we further develop Khaliq's simulation model by adopting more realistic geometry to simulate the NMJ cleft and including new equations describing 3D reaction and the diffusion process with nonlinear reaction source terms and predicting the process rates of Acetylcholine with the receptor and the enzyme. The simulation analysis agrees with experimental measurements of the reported maximum number of open receptors during the course of a normal action potential. The population of the open receptor as a function of time are investigated and discussed.

4.2 Formulation

The NMJ is a three dimensional (3D) molecular diffusion-reaction system with a flat cylindrical fold shape. The synaptic gap is bounded by the pre-synaptic membrane at the top and post synaptic membrane or end-plate with a fork shape at the bottom, and it is accessible to the external environment at the edge, as shown in Figure 4-1. [68, 69, 70, 71, 72].

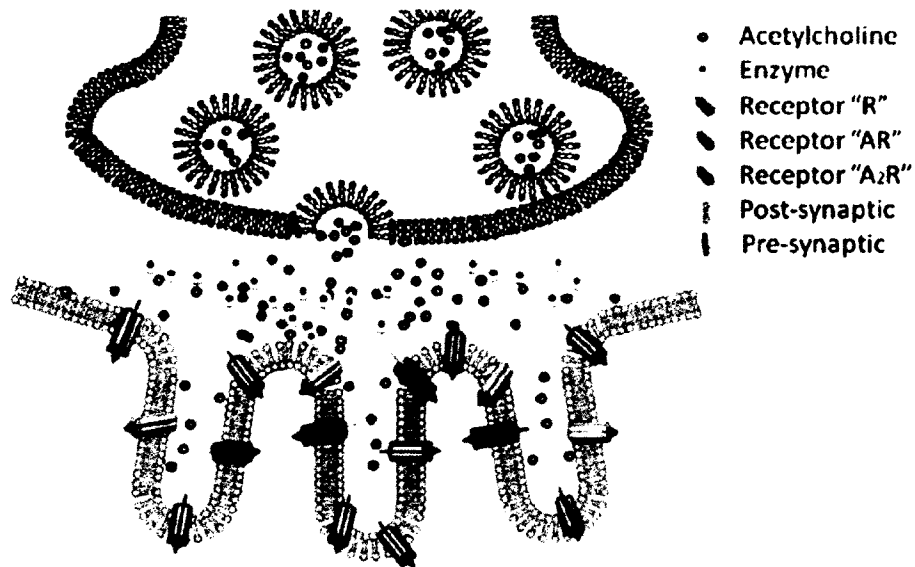


Figure 4-1: NMJ simulation: Neuromuscular junction.

We generate the corresponding 3D mesh for the NMJ cleft, shown in Figure 4-2, within which we turn the mesh upside down in order to demonstrate the shape of the post synaptic and the reactions of the Acetylcholine receptors.

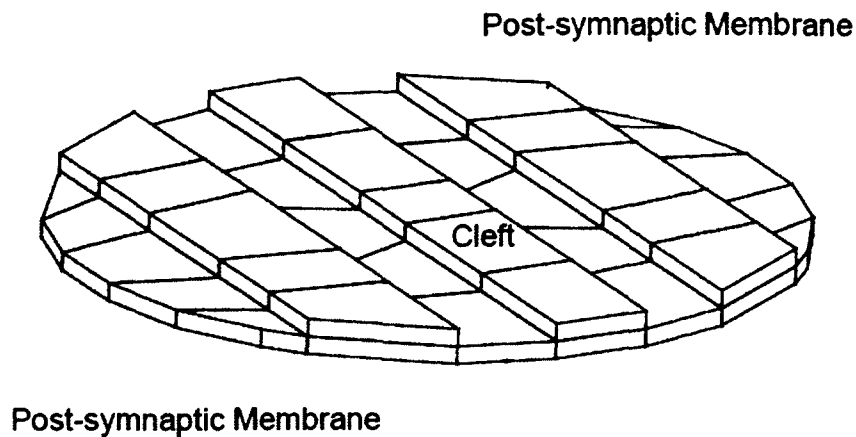


Figure 4-2: NMJ simulation: Mesh of the NMJ cleft.

The neurotransmitter Acetylcholine is located at the pre-synaptic membrane and diffuses across the NMJ cleft, while all types of neuromuscular receptors are located at the post-synaptic membrane and are immobile. Acetylcholinesterase, known as an enzyme which hydrolyzes the neurotransmitter Acetylcholine, fills in the cleft. Therefore, by further improving our previous model, we include the following chemical reaction partial differential equations (PDE) involving Acetylcholinesterase in the NMJ cleft. For consistency consideration, we adopt the same notations being used in Khaliq's paper [73].



where A , E , AE , and acE represent the Acetylcholine, Acetylcholinesterase, Michaelis ligand-substrate complex, and acylate enzyme, respectively, and k_{E1} , k_{-E1} , k_{E2} and k_{E3} are the forward and backward reaction constants for E , AE and acE , respectively. In a

normal NMJ activity, the enzymatic destruction of Acetylcholine by AchE is an important reaction. Thus, we have included the complete fundamental processes which constitute the production, transmission and attenuation of a neuromuscular action potential; that is, reactions Eqs. (41)-(43) represent the full kinetic cycle of Acetylcholine initially reacting with Acetylcholinesterase and proceeding to the final renewal of the enzyme.

The reaction rates of the chemical equations involving the enzyme given above can be expressed by the following equations:

$$\frac{\partial(E)}{\partial t} = -k_{E1}(A)(E) + k_{-E1}(AE) + k_{E3}(acE), \quad (44)$$

$$\frac{\partial(AE)}{\partial t} = k_{E1}(A)(E) - k_{-E1}(AE) - k_{E2}(AE), \quad (45)$$

$$\frac{\partial(acE)}{\partial t} = -k_{E3}(acE) + k_{E2}(AE). \quad (46)$$

The Acetylcholine is transported across the cleft and reacts with receptors located on the postsynaptic membrane. We expressed the rate change of concentration for Acetylcholine in a diffusion-reaction equation with source terms in the Cartesian coordinates:

$$\begin{aligned} \frac{\partial(A)}{\partial t} = & D_x \frac{\partial^2(A)}{\partial x^2} + D_y \frac{\partial^2(A)}{\partial y^2} + D_z \frac{\partial^2(A)}{\partial z^2} \\ & -2k_R(A)(R) + k_{-R}(AR) \\ & -k_{AR}(A)(AR) + 2k_{-AR}(A_2R) \\ & -k_{E1}(A)(E) + k_{-E1}(AE), \end{aligned} \quad (47)$$

where D_x , D_y , and D_z are diffusion coefficients along the x , y , z direction, respectively, and R , AR , and A_2R stand for unbound, single-bound and double-bound closed Acetylcholine receptors. The values of all the coefficients in our simulation is listed in

Table. 4-1. After a nondimensionalization process, we rescaled the geometry as $0 \leq x, y \leq 5$ and $0 \leq z \leq 0.5$.

Table 4-1: Values of coefficients in NMJ simulation.

Name	Value (units)
k_R	$3.0 \times 10^7 \text{ M}^{-1}\text{S}^{-1}$ ($3.0 \times 10^4 \text{ Kg}^{-1}\text{S}^{-1}$)
k_{-R}	$1.0 \times 10^4 \text{ S}^{-1}$
k_{AR}	$3.0 \times 10^7 \text{ M}^{-1}\text{S}^{-1}$ ($3.0 \times 10^4 \text{ Kg}^{-1}\text{S}^{-1}$)
k_{-AR}	$1.0 \times 10^4 \text{ S}^{-1}$
k_{E1}	$2.0 \times 10^8 \text{ M}^{-1}\text{S}^{-1}$ ($2.0 \times 10^5 \text{ Kg}^{-1}\text{S}^{-1}$)
k_{-E1}	$1.0 \times 10^3 \text{ S}^{-1}$
k_{E2}	$110.0 \times 10^3 \text{ S}^{-1}$
k_{E3}	$20.0 \times 10^3 \text{ S}^{-1}$
D_x, D_y, D_z	$1.0 \times 10^{-6} \text{ cm}^2/\text{s}$
R	$5.0 \times 10^{-5} \text{ cm}$
L	$5.0 \times 10^{-6} \text{ cm}$
<i>Num. of molecules per Mol</i>	$6.022 \times 10^{23} \text{ mol}^{-1}$

The boundary conditions for A are assumed to be:

$$\frac{\partial A(x, y, 0, t)}{\partial z} = 0, \quad (48)$$

$$\frac{\partial A(x, y, 0.5, t)}{\partial z} = 0, \quad (49)$$

$$A(x, y, z, t) = 0, \quad \text{where } x^2 + y^2 = 5^2. \quad (50)$$

The initial conditions for A , R , and E are assumed to be:

$$A(x, y, z, 0) = A_0, \quad (51)$$

$$R(x, y, z, 0) = R_0, \quad (52)$$

$$E(x, y, z, 0) = E_0, \quad \text{where } 0 \leq z \leq 0.5. \quad (53)$$

The parabolic Eq. (47) with an activated enzyme is to be solved numerically to predict the evolution of the concentration of Acetylcholine under normal neuromuscular operation.

4.3 Numerical Method

The ordinary differential Eqs. (44)-(46) are solved with the 4th order of the Runge-Kutta scheme for E , AE and acE . We demonstrate the procedure for E as an example below and the rest are similar:

$$E_g^{n+1} = E_g^n + \frac{E_{rk1} + 2E_{rk2} + 2E_{rk3} + E_{rk4}}{6}, \quad (54)$$

where the subscript g stands for the global value, the subscript rk_1 represents the first step in Runge-Kutta scheme, etc., and coefficients E_{rk1} , E_{rk2} , E_{rk3} , and E_{rk4} are determined as:

$$E_{rk1} = \Delta t \left[-k_{E1} \left(\frac{A_g^{n+1} + A_g^n}{2} \right) E_g^n + k_{-E1} (AE_g^n) + k_{E3} (acE_g^n) \right], \quad (55)$$

$$E_{rk2} = \Delta t \left[-k_{E1} \left(\frac{A_g^{n+1} + A_g^n}{2} \right) \left(E_g^n + \frac{E_{rk1}}{2} \right) + k_{-E1} \left(AE_g^n + \frac{AE_{rk1}}{2} \right) + k_{E3} \left(acE_g^n + \frac{acE_{rk1}}{2} \right) \right], \quad (56)$$

$$E_{rk3} = \Delta t \left[-k_{E1} \left(\frac{A_g^{n+1} + A_g^n}{2} \right) \left(E_g^n + \frac{E_{rk2}}{2} \right) + k_{-E1} \left(AE_g^n + \frac{AE_{rk2}}{2} \right) + k_{E3} \left(acE_g^n + \frac{acE_{rk2}}{2} \right) \right], \quad (57)$$

$$E_{rk4} = \Delta t \left[-k_{E1} \left(\frac{A_g^{n+1} + A_g^n}{2} \right) \left(E_g^n + \frac{E_{rk3}}{2} \right) + k_{-E1} \left(AE_g^n + \frac{AE_{rk3}}{2} \right) + k_{E3} \left(acE_g^n + \frac{acE_{rk3}}{2} \right) \right], \quad (58)$$

where AE_{rk} , acE_{rk} are chemical compound Michaelis ligand-substrate complex and Acylate enzyme in the corresponding steps in Runge-Kutta scheme.

For the parabolic differential Eq. (47), we first use the Spectral Element Method to discretize the spatial derivatives only and then adopt the second order Crank-Nicholson scheme for temporal derivatives. Within a typical element in the original x, y, z coordinates, we transfer it into a standard element in ξ_1, ξ_2, ξ_3 , and then describe the solution for A in Eq. (47) in a typical element in terms of a tensor product of three one dimensional Lagrange polynomials on quadrature points:

$$A^e(\xi_1, \xi_2, \xi_3) = \sum_p^{P_0} \sum_q^{P_0} \sum_r^{P_0} A_{pqr}^e h_p^e(\xi_1) h_q^e(\xi_2) h_r^e(\xi_3) = \sum_{pqr} A_{pqr}^e \Phi_{pqr}^e \quad (59)$$

where P_0 is the polynomial order of the basis function. For convenience, we choose the same order in x, y, z coordinates, although different orders could be implemented.

From the collocation property of nodal SEM, we know that the coefficients of the basis functions are the numerical solutions at quadrature points. Those reaction terms in Eq. (47) can be represented as:

$$[(A)(R)]^e(\xi_1, \xi_2, \xi_3) = \sum_{pqr} [(A_{pqr}) \cdot (R_{pqr})]^e \Phi_{pqr}^e \quad (60)$$

$$[(A)(AR)]^e(\xi_1, \xi_2, \xi_3) = \sum_{pqr} [(A_{pqr}) \cdot (AR_{pqr})]^e \Phi_{pqr}^e \quad (61)$$

$$[(A_2R)]^e(\xi_1, \xi_2, \xi_3) = \sum_{pqr} [(A_2R)_{pqr}]^e \Phi_{pqr}^e \quad (62)$$

$$[(A)(E)]^e(\xi_1, \xi_2, \xi_3) = \sum_{pqr} [(A_{pqr}) \cdot (E_{pqr})]^e \Phi_{pqr}^e \quad (63)$$

$$[(AE)]^e(\xi_1, \xi_2, \xi_3) = \sum_{pqr} [(AE)_{pqr}]^e \Phi_{pqr}^e \quad (64)$$

where superscript e stands for the elemental solutions. For nonlinear terms, the coefficients are calculated in the point-wise product of the values of two species on the quadrature points. We denoted this operation as “ \cdot ” in Eqs. (60)-(63) and (64). To be

specific, we choose Eq. (60) to illustrate this implementation. The nonlinear product $(A)(R)$ at a quadrature point ξ_{ijk} are computed as below:

$$[(A)(R)]^e(\xi_i, \xi_j, \xi_k) = [(A_{pqr})]^e \Phi_{pqr}^e(ijk) [(R_{lmn})]^e \Phi_{lmn}^e(ijk). \quad (65)$$

Due to the property of δ function:

$$\Phi_{pqr}^e(ijk) = \delta_{pqr,ijk} = \begin{cases} 1, & p = i, q = j, r = k \\ 0, & \text{Otherwise} \end{cases}, \quad (66)$$

and

$$\Phi_{lmn}^e(ijk) = \delta_{lmn,ijk} = \begin{cases} 1, & l = i, m = j, n = k \\ 0, & \text{Otherwise} \end{cases}. \quad (67)$$

Eq. (65) can be approximated as:

$$[(A)(R)]^e(\xi_i, \xi_j, \xi_k) = [(A_{pqr})]^e [(R_{pqr})]^e \Phi_{pqr}^e(ijk) = [(A_{pqr}) \cdot (AR_{pqr})]^e \Phi_{pqr}^e(ijk). \quad (68)$$

Galerkin projection is performed with the test functions chosen to be the same as the basis functions:

$$\Phi_{lmn}^e = \sum_l^{P_o} \sum_m^{P_o} \sum_n^{P_o} h_l^e(\xi_1) h_m^e(\xi_2) h_n^e(\xi_3) \quad (69)$$

We could acquire the weak form in a single element by performing a Galerkin projection.

After the global assembly, we get the matrix form of Eq. (47):

$$\begin{aligned} \underline{\underline{\mathbf{M}}} \frac{\partial A_g}{\partial t} &= -\underline{\underline{\mathbf{K}}} A_g - 2k_R \underline{\underline{\mathbf{M}}} [(A_g) \cdot (R_g)] + k_{-R} \underline{\underline{\mathbf{M}}} (AR)_g \\ &\quad - k_{AR} \underline{\underline{\mathbf{M}}} [(A_g) \cdot (AR)_g] + 2k_{-AR} \underline{\underline{\mathbf{M}}} (A_2R)_g \\ &\quad - k_{E1} \underline{\underline{\mathbf{M}}} [(A_g) \cdot (E_g)] + k_{-E1} \underline{\underline{\mathbf{M}}} (AE)_g. \end{aligned} \quad (70)$$

where $\underline{\underline{\mathbf{M}}}$ is the global mass matrix:

$$\underline{\underline{\mathbf{M}}} = \sum_e^N \left(\int_{\Omega_e} \Phi_{pqr}^e \Phi_{lmn}^e |J|_e d\xi_1 d\xi_2 d\xi_3 \right), \quad (71)$$

$|J|$ is the Jacobian of the element e , Ω_e is the integral domain, and N is the total

number of elements. The diffusion matrix $\underline{\underline{\mathbf{K}}}$ consists of contributions from all directions:

$$\underline{\underline{\mathbf{K}}} = \sum_e^N \left\{ \begin{array}{l} D_x \int_{\Omega_e} (\Phi_{pqr}^e)'_x (\Phi_{lmn}^e)'_x \left(\frac{\partial x_2}{\partial \xi_2} \frac{\partial x_3}{\partial \xi_3} - \frac{\partial x_2}{\partial \xi_3} \frac{\partial x_3}{\partial \xi_2} \right) |J|_e^{-1} d\xi_1 d\xi_2 d\xi_3 \\ + D_y \int_{\Omega_e} (\Phi_{pqr}^e)'_y (\Phi_{lmn}^e)'_y \left(\frac{\partial x_1}{\partial \xi_1} \frac{\partial x_3}{\partial \xi_3} - \frac{\partial x_1}{\partial \xi_3} \frac{\partial x_3}{\partial \xi_1} \right) |J|_e^{-1} d\xi_1 d\xi_2 d\xi_3 \\ + D_z \int_{\Omega_e} (\Phi_{pqr}^e)'_z (\Phi_{lmn}^e)'_z \left(\frac{\partial x_1}{\partial \xi_1} \frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2} \frac{\partial x_2}{\partial \xi_1} \right) |J|_e^{-1} d\xi_1 d\xi_2 d\xi_3 \end{array} \right\}. \quad (72)$$

By Crank-Nicholson, Eq. (68) becomes:

$$\begin{aligned} \underline{\underline{\mathbf{M}}} \frac{A_g^{n+1} - A_g^n}{\Delta t} &= -\underline{\underline{\mathbf{K}}} \frac{A_g^{n+1} + A_g^n}{2} - 2k_R \underline{\underline{\mathbf{M}}} \left[\left(\frac{A_g^{n+1} + A_g^n}{2} \right) .* \left(\frac{R_g^{n+1} + R_g^n}{2} \right) \right] \\ + k_{-R} \underline{\underline{\mathbf{M}}} \left(\frac{(AR)_g^{n+1} + (AR)_g^n}{2} \right) &- k_{AR} \underline{\underline{\mathbf{M}}} \left[\left(\frac{A_g^{n+1} + A_g^n}{2} \right) .* \left(\frac{(AR)_g^{n+1} + (AR)_g^n}{2} \right) \right] \\ + 2k_{-AR} \underline{\underline{\mathbf{M}}} \left(\frac{(A_2R)_g^{n+1} + (A_2R)_g^n}{2} \right) & \\ - k_{E1} \underline{\underline{\mathbf{M}}} \left[\left(\frac{A_g^{n+1} + A_g^n}{2} \right) .* \left(\frac{(E)_g^{n+1} + (E)_g^n}{2} \right) \right] &+ k_{-E1} \underline{\underline{\mathbf{M}}} \left(\frac{(AE)_g^{n+1} + (AE)_g^n}{2} \right) \end{aligned} \quad (73)$$

The computational procedures for values of chemical compounds at the next time level $n + 1$ from the present level n are as given below:

- 1: Initialize values of compounds A , R , AR , A_2R , A_2R_{open} , E , AE , and acE at time level $n + 1$ with the same values at time step n ;
- 2: Apply the Runge-Kutta scheme to obtain the values of A , R , AR , A_2R , A_2R_{open} , E , AE , and acE at time level $n + 1$;

3: Substitute updated values of A , R , AR , A_2R , A_2R_{open} , E , AE , and acE from step 2 into Eq. (71), then solve for the value of A at time level $n + 1$;

4: Repeat steps 1 to 3 until the overall difference of the two adjacent values of A is under a tolerance of 1.0×10^{-12} .

4.4 Numerical Results and Discussions

We use the nodal Spectral Element Method for numerical solutions. The resolution could be improved by introducing more quadrature points; hence, the polynomial order of the basis' expansions are increased as well.

Figure 4-3 shows the top view of the computational domain in the cross section of NMJ at plane $Z = 0$ with a polynomial order of 3, 4, and 5. The red lines are the boundaries of elements, and intersections of blue lines are actual quadrature points within this element. Quadrature points are selected to be zeros of Legendre polynomials and cluster around the boundaries in order to minimize the overall discretization error.

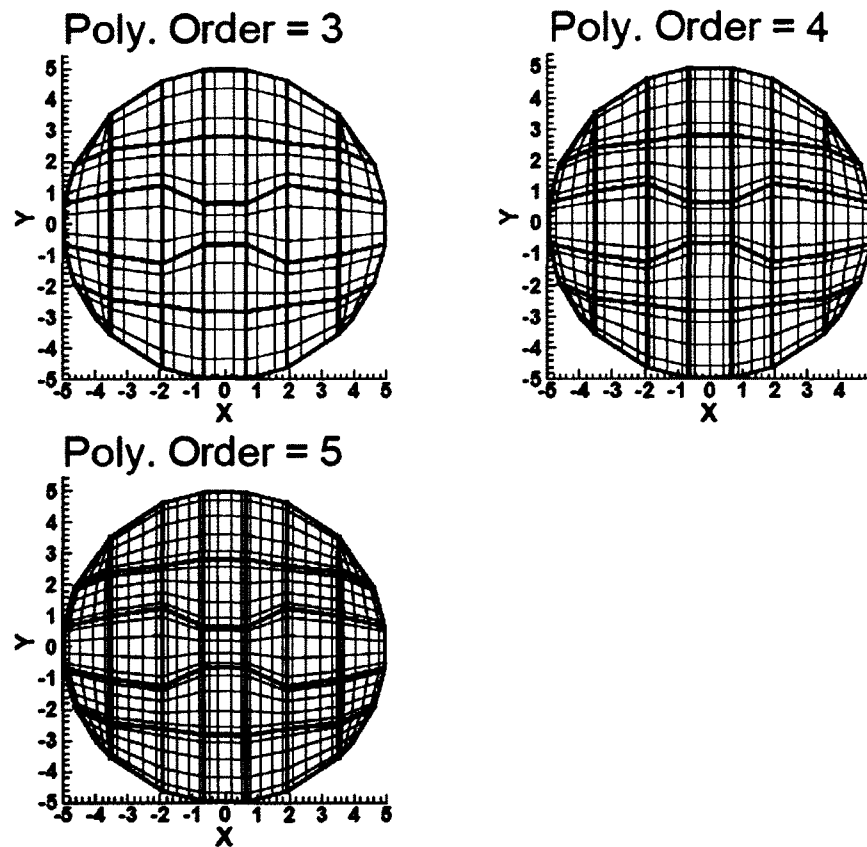


Figure 4-3: NMJ simulation: Computational mesh at different polynomial orders in the plane $Z = 0$ for NMJ.

Figure 4-4 illustrates the side view of computational domain in the middle cross section (plane $Y = 0$) of NMJ at the polynomial order being 3, 4 and 5. To better visualize the mesh, we exaggerated the ratio in the z direction for the cases of the 4th and 5th order expansion. This is because the synaptic cleft is very thin and when the polynomial order is above five, quadrature points are clustered together in the plot.

Poly. Order = 3 (x, z ratio 1: 1)



Poly. Order = 4 (x, z ratio 1: 0.5)



Poly. Order = 5 (x, z ratio 1: 0.5)



Figure 4-4: NMJ simulation: Computational mesh at different polynomial orders in the plane $Y = 0$ for NMJ.

Since the diffusion process of the neurotransmitter Acetylcholine happens in the NMJ cleft, we show the contour lines of concentration in the cross sectional plane of $Y = 0$ in NMJ in Figure 4-5. Four typical time snapshots are presented to show the change in the concentration of Acetylcholine in NMJ. Since the diffusion rate is almost the same in all directions (the effect of confinement is neglected here), we observe that speeds are the same in the x, y directions. To consider the effect of confined diffusion and any inhomogeneity, we could set different diffusion rates in each direction.

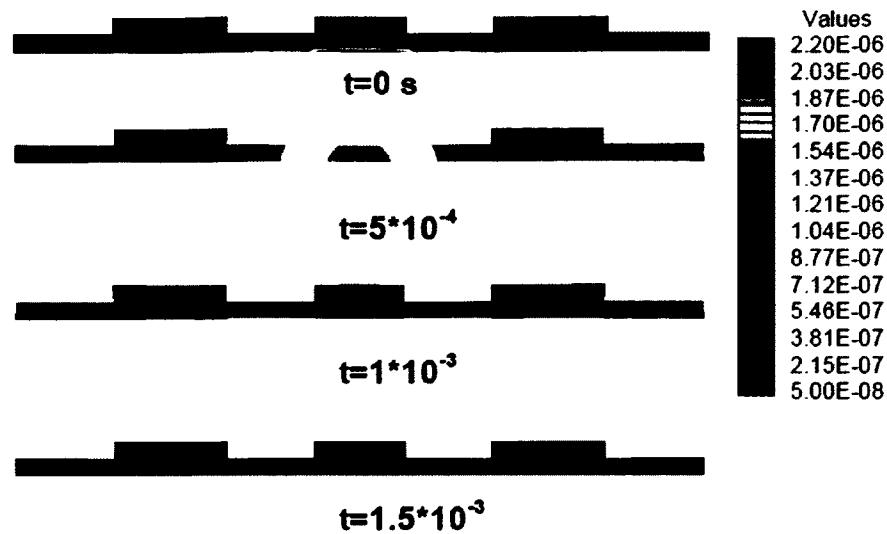


Figure 4-5: NMJ simulation: Time-evolution of Acetylcholine in the diffusion-Reaction system in the plane $Y = 0$.

An accurate model should be capable of predicting the number of open receptors versus time. Some references such as [74, 75, 76] have reported that the maximum number of open receptors during the course of a normal action potential is about 2000 at 0.3 ms. Our predicted time evolution of A_2R_{open} in the neuromuscular junction is shown in Figure 4-6. We could see the total number of A_2R_{open} increase significantly in a very short time and reaches its maximum around time $0.4 \times 10^{-3} s$ and then rapidly decreases after $3 \times 10^{-3} s$. This trend agrees with the experimental data [77, 69]. Table 4-1 lists the geometrical size parameters, reaction rate constants, diffusion coefficients, and other parameters in the model.

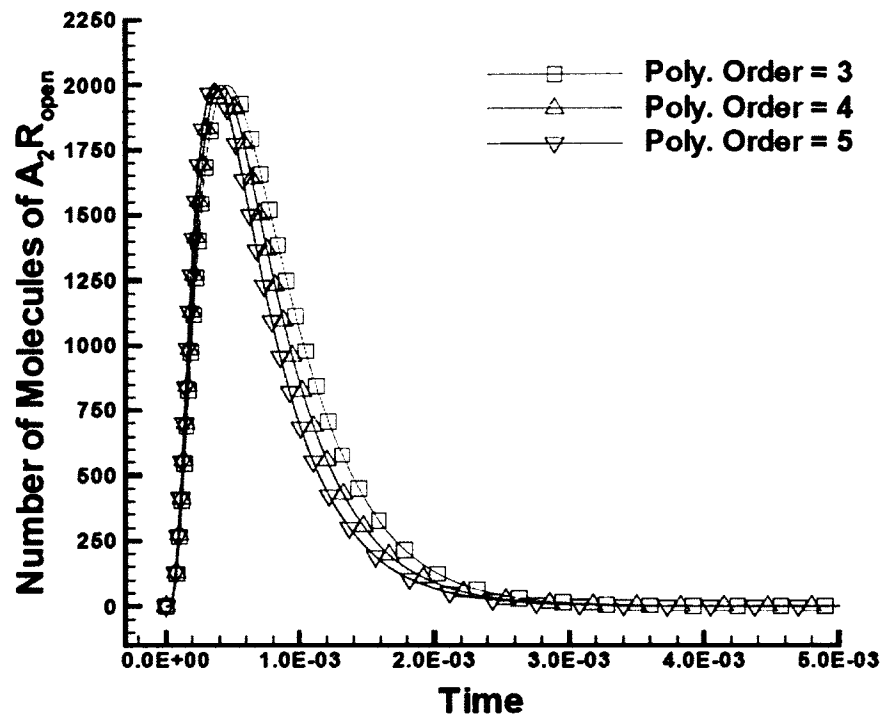


Figure 4-6: NMJ simulation: Time-evolution of number of molecules of A_2R_{open} in NMJ.

To further demonstrate the time evolution of double-bounded open Acetylcholine receptors (A_2R_{open}) located at the post synaptic membrane, we show the contour plots of the concentration of A_2R_{open} versus time. Because the most significant reaction and diffusion takes place from the beginning to 1.5×10^{-3} s, we focus on this frame in Figure 4-7 and Figure 4-8. The variation of concentration agrees with the results in Figure 4-6. We assume that the distribution of Acetylcholine receptors at the post-synaptic membrane is uniform. Our model allows us to study how a certain abnormal distribution affects the receptor dynamics during an action potential through changing the distribution of Acetylcholine receptors according to the actual situations.

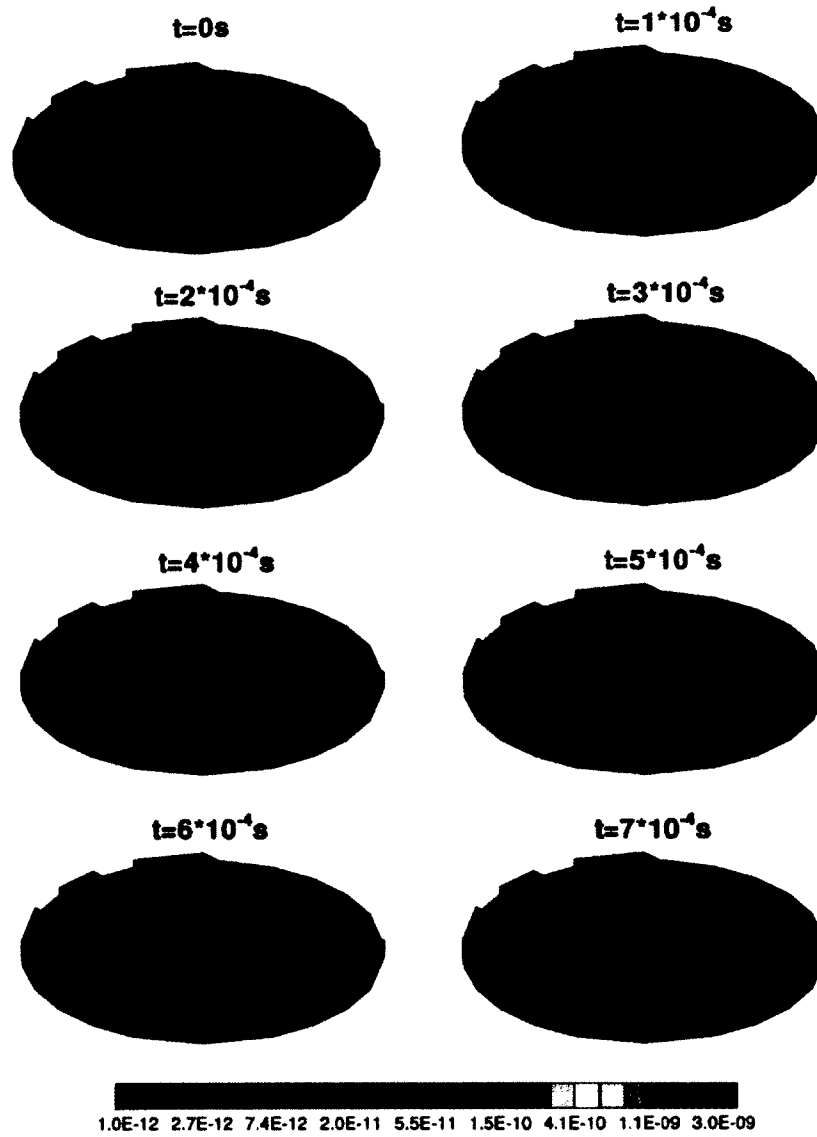


Figure 4-7: NMJ simulation: Time-evolution of concentration of A_2R_{open} at post membrane, Part 1.

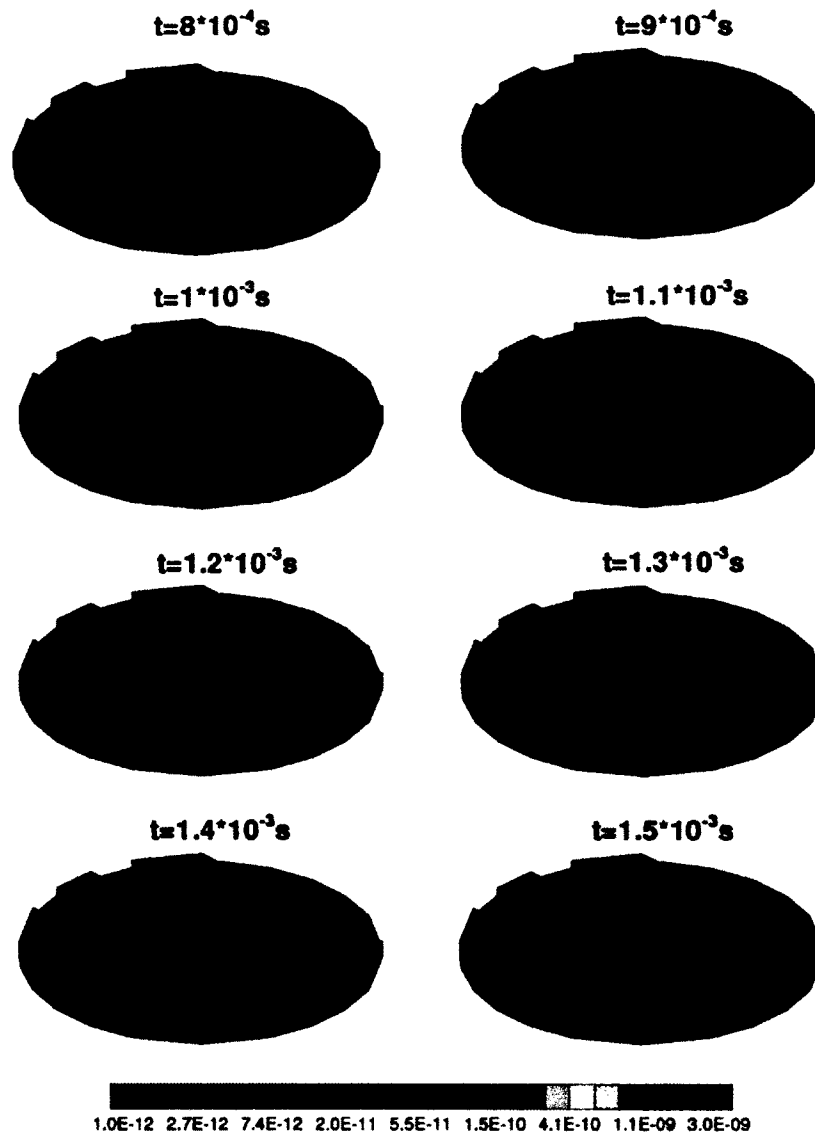


Figure 4-8: NMJ simulation: Time-evolution of concentration of A_2R_{open} at post membrane, Part 2.

4.5 Conclusions

We have presented a full 3D model with realistic geometry via Spectral Element Method for simulating the reaction-diffusion of Acetylcholine and Acetylcholine receptors' dynamics in the neuromuscular junction under conditions of activated enzyme. Results show agreements with other literatures [70, 72, 76] that the maximum number of

open receptors during the course of a normal action potential to be around 2000 after 0.3 msec. With high accuracy, our model predicted the maximum number of open receptors as time goes by. Aside from that, this model is fully capable of studying the sensitivity of the open receptors' dynamics to the changes in the anisotropic diffusion parameters, and it can also analyze the subsequent effects of open receptor distribution when Acetylcholine receptors are not uniformly distributed at the post membrane. Future investigations will focus on the study of an organophosphate neurotoxin entering the cleft from the outer periphery and enzyme regeneration with oxygen therapy.

CHAPTER 5

DISCONTINUOUS GALERKIN SIMULATION OF INCOMPRESSIBLE FLOW

5.1 Introduction

This chapter focuses on using the DG method to solve incompressible Navier-Stokes' equation. 2D and 3D simulations of the flow passing a cylinder and lid-driven cavity flow will be presented to show the accuracy and efficiency of the DG method. For the DG solver, unstructured triangle mesh and tetrahedron mesh are used for 2D and 3D simulations, respectively.

5.2 Numerical Method

Time-dependent incompressible Navier–Stokes equation is considered below:

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} + \mathbf{f} \quad \text{in } (0, T) \times \Omega, \quad (74)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } (0, T) \times \Omega, \quad (75)$$

where $\mathbf{u} = (u, v)$ and p are velocity in x, y directions, pressure, ν is kinematic viscosity and \mathbf{f} represents the external force.

We rewrite the nonlinear convection term into flux form \mathbf{F} :

$$\mathbf{F} = \begin{bmatrix} u^2 & uv \\ uv & v^2 \end{bmatrix}. \quad (76)$$

Then we have the conservative flux form of Eq. (74):

$$\mathbf{u}_t + \nabla \cdot \mathbf{F} = -\nabla p + \nu \Delta \mathbf{u} + \mathbf{f}, \quad (77)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (78)$$

5.2.1 Time Splitting Scheme

Second order accuracy time splitting scheme is used for marching time. Within each time step, the original Navier-Stokes equation is separated into three equations.

Firstly, we explicitly integrate the conservation component [78]. Adams-Bashforth second-order scheme is used:

$$\frac{\gamma \tilde{\mathbf{u}} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = -\beta_0 \nabla \cdot \mathbf{F}^n - \beta_1 \nabla \cdot \mathbf{F}^{n-1}. \quad (79)$$

Second, a pressure Poisson equation is solved by assuming:

$$\nabla \cdot \tilde{\mathbf{u}} = 0, \quad (80)$$

$$\nabla^2 \bar{p}^{n+1} = \frac{\gamma}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}, \quad (81)$$

with proper boundary condition for pressure. For the inflow and the wall boundaries, the Neumann boundary conditions are derived from the governing equation as:

$$\begin{aligned} \frac{\partial \bar{p}^{n+1}}{\partial \mathbf{n}} &= -\beta_0 \mathbf{n} \cdot (\mathbf{u}_t + \nabla \cdot \mathbf{F} - \nu \Delta \mathbf{u})^n \\ &\quad -\beta_1 \mathbf{n} \cdot (\mathbf{u}_t + \nabla \cdot \mathbf{F} - \nu \Delta \mathbf{u})^{n-1}. \end{aligned} \quad (82)$$

Then, the velocity field is updated by:

$$\gamma \frac{\tilde{\tilde{\mathbf{u}}} - \tilde{\mathbf{u}}}{\Delta t} = -\nabla \bar{p}^{n+1}. \quad (83)$$

Last, we implicitly integrate the viscous component:

$$\gamma \frac{\mathbf{u}^{n+1} - \tilde{\tilde{\mathbf{u}}}}{\Delta t} = \nu \Delta \mathbf{u}^{n+1}. \quad (84)$$

With the Sum of all three sub steps together, and we have a second order scheme in time, as shown is Eq. (83), where $\tilde{\tilde{\mathbf{u}}}$ and $\tilde{\mathbf{u}}$ are intermediate velocity field.

$$\begin{aligned} & \frac{\gamma \mathbf{u}^{n+1} - \gamma \tilde{\tilde{\mathbf{u}}} + \gamma \tilde{\tilde{\mathbf{u}}} - \gamma \tilde{\mathbf{u}} + \gamma \tilde{\mathbf{u}} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} \\ & = -\beta_0 \nabla \cdot \mathbf{F}^n - \beta_1 \nabla \cdot \mathbf{F}^{n-1} - \nabla \bar{p}^{n+1} + \nu \Delta \mathbf{u}^{n+1}. \end{aligned} \quad (85)$$

5.2.2 Internal Penalty Flux

Since the first step in the time splitting scheme is to solve a nonlinear convection equation, the discontinuous solution may be acquired in this step. In order to eliminate the jumps from the solution of the convection time step, for Poisson and Helmholtz equations, internal penalty fluxes are used to adjust the solution.

For example, consider the two dimensional Poisson equation,

$$\nabla^2 u = f. \quad (86)$$

We reconstruct it as a system of two first order equations [79, 36]:

$$\begin{cases} \nabla u = \mathbf{q} \\ \nabla \cdot \mathbf{q} = f \end{cases} \quad (87)$$

Assume we discretize (u, \mathbf{q}) as (u, q^x, q^y) , and we obtain the strong form for the first equation of (87):

$$\mathbf{M} \mathbf{q}_h^x = \mathbf{S}_x \mathbf{u}_h - \int_{\partial D_k} \hat{\mathbf{n}}_x (\mathbf{u}_h - \mathbf{u}_h^*) l(x, y) dx dy, \quad (88)$$

$$\mathbf{M} \mathbf{q}_h^y = \mathbf{S}_y \mathbf{u}_h - \int_{\partial D_k} \hat{\mathbf{n}}_y (\mathbf{u}_h - \mathbf{u}_h^*) l(x, y) dx dy. \quad (89)$$

So for the second equation of (87), we have:

$$\mathbf{S}_x \mathbf{q}_h^x + \mathbf{S}_y \mathbf{q}_h^y - \int_{\partial D_k} \hat{\mathbf{n}} \cdot (\mathbf{q}_h^x - \mathbf{q}_h^*) l(x, y) dx dy, \quad (90)$$

$$-\int_{\partial D_k} \hat{n} \cdot (q_h^y - q_h^*) l(x, y) dx dy = M f_h.$$

The internal penalty fluxes are:

$$q_h^* = \{ \{ \nabla u_h \} \} - \tau \llbracket u \rrbracket, u_h^* = \{ \{ u_h \} \}, \tag{91}$$

where $\{ \{ \nabla u_h \} \} = \frac{\nabla u_h^+ + \nabla u_h^-}{2}$ and $\llbracket u \rrbracket = n^- u^- + n^+ u^+$.

5.2.3 Constructing Data for Visualization

Unstructured mesh, such as triangular or tetrahedron element, is used in the DG simulation to visualize the results which usually requires additional information about connectivity. A new approach which is suitable for Tecplot format is used to generate the output data without providing the connective information. Tecplot zone data type is used for constructing the data. For instance, if the order of the basis function in each triangle element is chosen to be $P_{order} = 5$, then we have a total of $(P_{order} + 1) (P_{order} + 2)/2 = 21$ interpolation points in the triangle. In order to avoid constructing the connective data for these interpolation points, we expand to $(P_{order} + 1)^2$ points with extra points assigned the values using the rules shown in Figure 5-1.

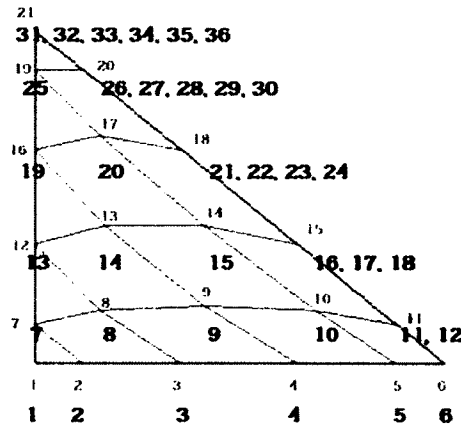


Figure 5-1: DG: Generating Tecplot compatible data for visualization

5.3 Numerical Results and Discussions

5.3.1 Two Dimensional Simulation of Flow Passing Cylinder

A 2D simulation of the flow passing a cylinder, as shown in Figure 5-2, is considered in this section, with zero external force and proper initial and boundary condition:

1. At wall and cylinder surface, non-slip boundary condition for \mathbf{u} and Neumann boundary condition $\frac{\partial p}{\partial \mathbf{n}}$ for pressure.
2. At the inflow, $\mathbf{u} = (\sin(\pi t/8)(6y(0.41-y)), 0)$, $0 \leq y \leq 0.41$, $0 \leq t \leq 8s$, and Neumann boundary condition $\frac{\partial p}{\partial \mathbf{n}}$ for pressure.
3. At the outflow, $\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0$, $p=0$.

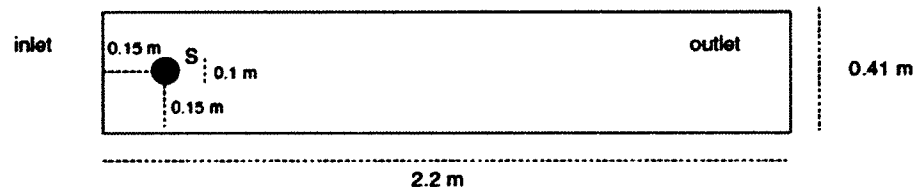


Figure 5-2: DG simulation: Schematic of computational domain [80], 2D.

The Reynolds number in this simulation is around 100. Figures. 5-3, 5-4, and 5-5 sequentially show the meshes and DG results at time $t = 8s$. We could see that the simulation results are consistent and not mesh-dependent. Vector fields and pressure contour are illustrated in each figure, and the vortex street is formed in the channel.

Six independent simulation runs are performed with different mesh and basis function settings, as shown in Table 5-1. The maximum drag, lift force and pressure difference are computed and compared with the reference value [80]. A good agreement

is shown in Table 5-1. The number of degree of freedoms in the DG simulation is only one tenth of the number required compared to standard Finite Element Method.

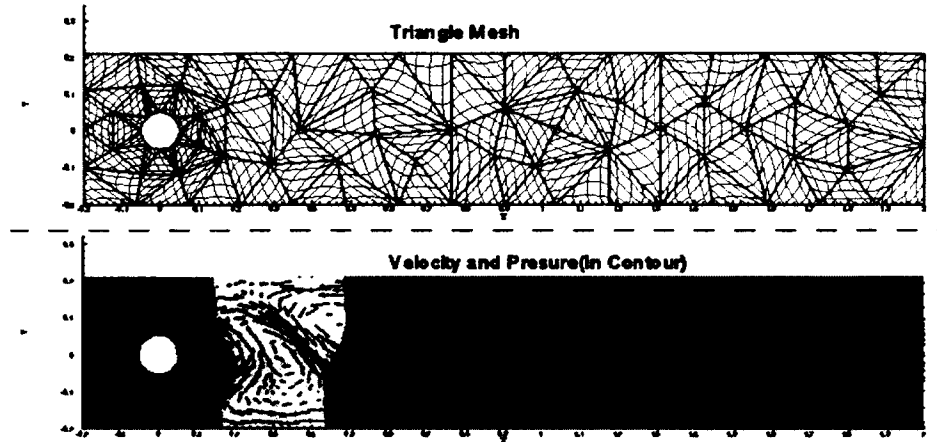


Figure 5-3: DG simulation: 2D simulation results of 115 elements and a polynomial order of 5.

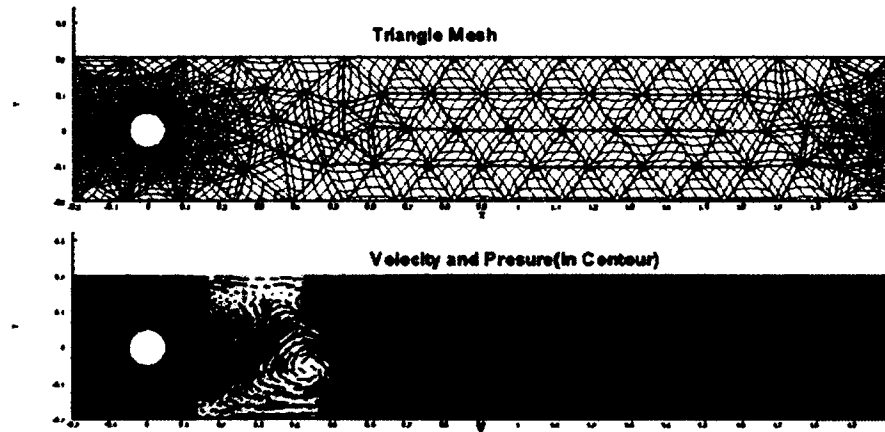


Figure 5-4: DG simulation: 2D simulation results of 307 elements and a polynomial order of 5.

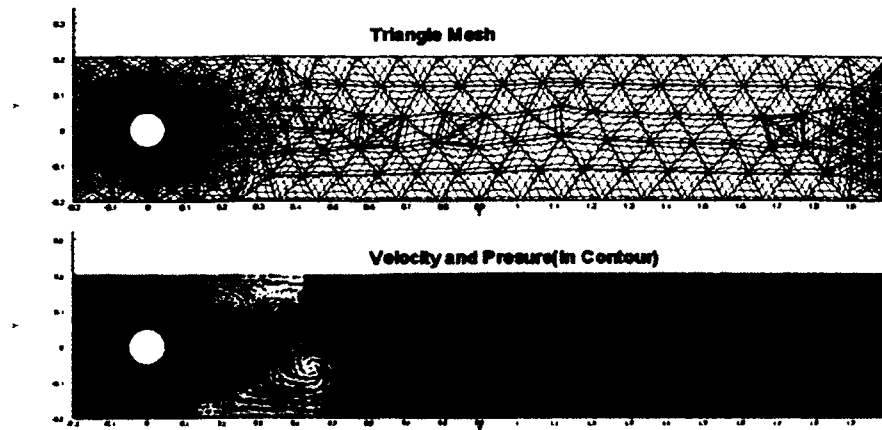


Figure 5-5: DG simulation: 2D simulation results of 704 elements and a polynomial order of 5.

Table 5-1: DG: 2D simulation results compared to reference values.

	Element Number	Polynomial order	Maximum Drag	Maximum Lift	Pressure Difference	Nodf
Numerical Results	115	5	2.9827	0.4749	0.1067	7245
	307	3	2.6332	0.4598	0.1266	9210
		4	3.0571	0.4238	0.1076	13815
		5	2.9598	0.4022	0.1080	19341
	704	4	2.9458	0.3784	0.1069	31680
5		2.9434	0.3888	0.1070	44352	
Reference Value	N/A	N/A	2.9505	0.3821	0.1113	449856

5.3.2 Three Dimensional Simulation of Flow Passing Cylinder

Similarly, a full 3D problem of flow passing a cylinder with a circular cross-section is considered. The fluid channel shown in Figure 5-6 [81, 82] is defined as $0 \leq$

$x \leq 1.5, -0.205 \leq y \leq 0.205, -0.205 \leq z \leq 0.205$ and with a cylinder center located at $x = 0.5, y = 0,$ and $z = 0$. The cylinder has a diameter of 0.1. Similar to the 2D simulation, non-slip velocity boundary conditions are set at the channel walls and cylindrical surface. At the inflow side of the channel, the fluid has a velocity of 0.45 m/s in the x direction. At the outflow, the fluid has Neumann boundary condition for velocity which is $\frac{\partial u}{\partial n} = 0$, and Dirichlet boundary condition for pressure $p = 0$. The Reynolds number of this simulation is around 20 with given inflow velocity.

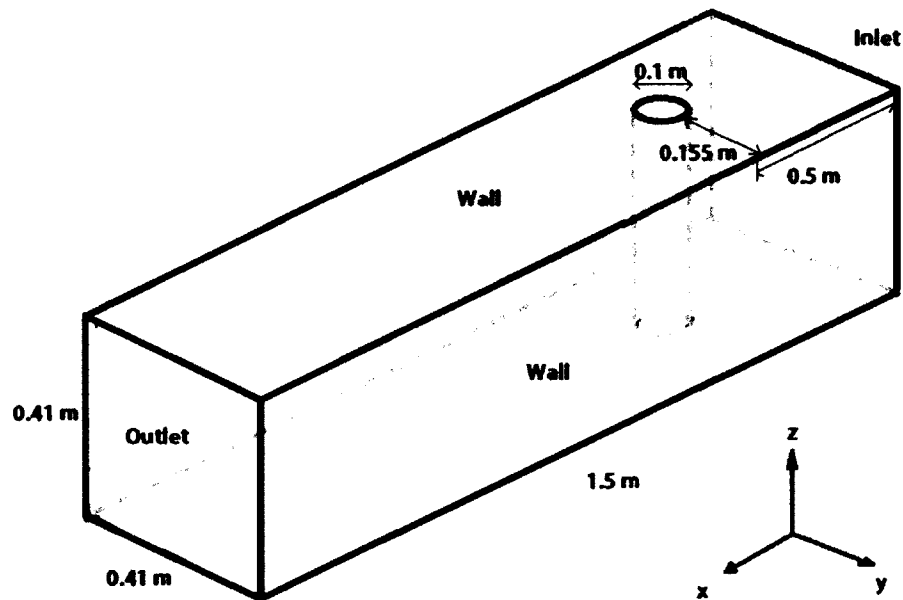


Figure 5-6: DG simulation: Schematic of computational domain, 3D.

Figure 5-7 shows the meshes we used for our 3D simulation. The tetrahedron elements are depicted by black lines, while the interpolation points are presented as intersections of red lines. Figure 5-8 shows the contour plots of velocity in the x direction obtained by using 5920 tetrahedron elements and third order bases. Different slices of the velocity fields are drawn.

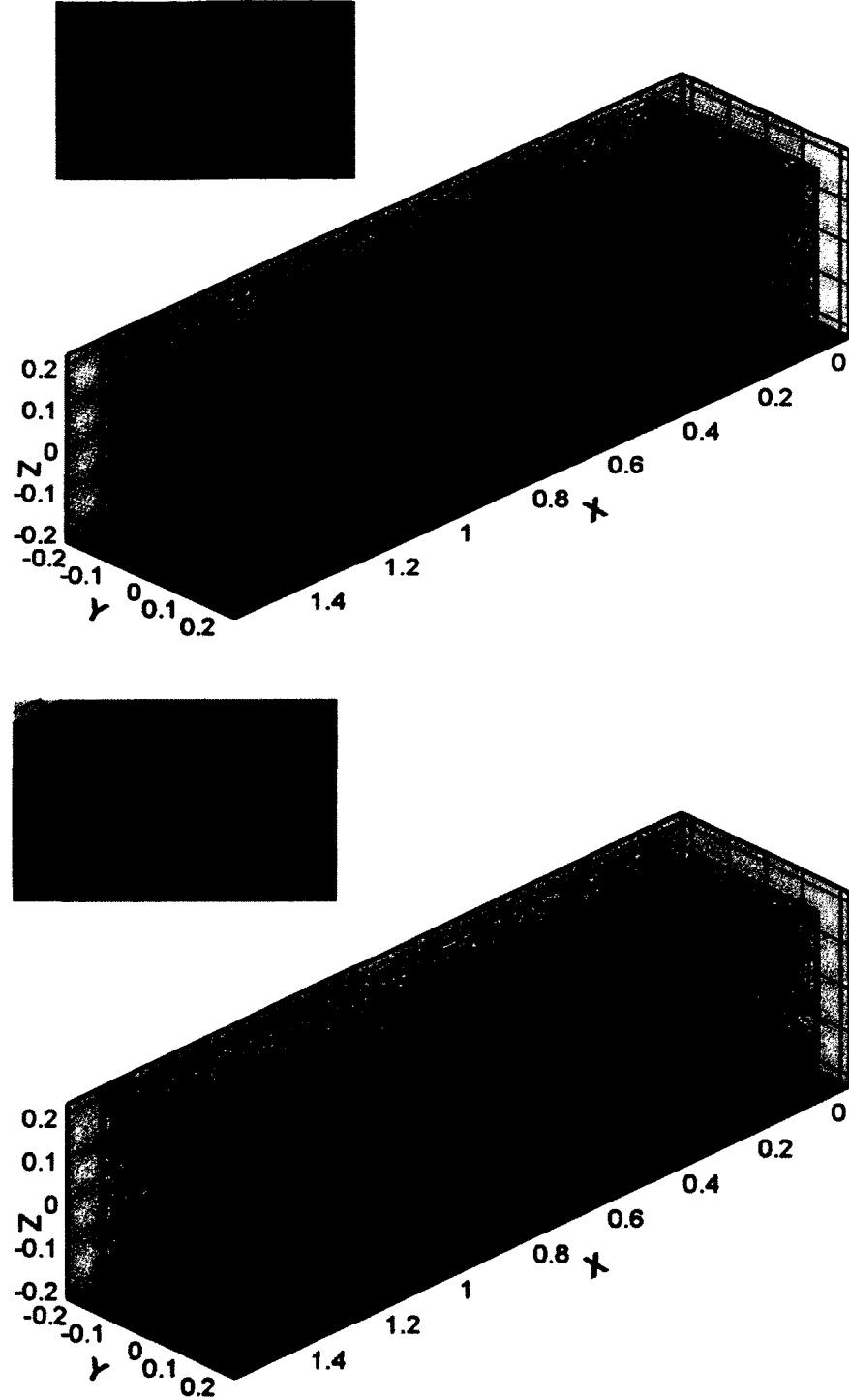


Figure 5-7: DG simulation: 3D simulation of flow passing cylinder: Mesh of 2898 elements with a polynomial order of 4 and mesh of 3320 elements and a polynomial order of 3.

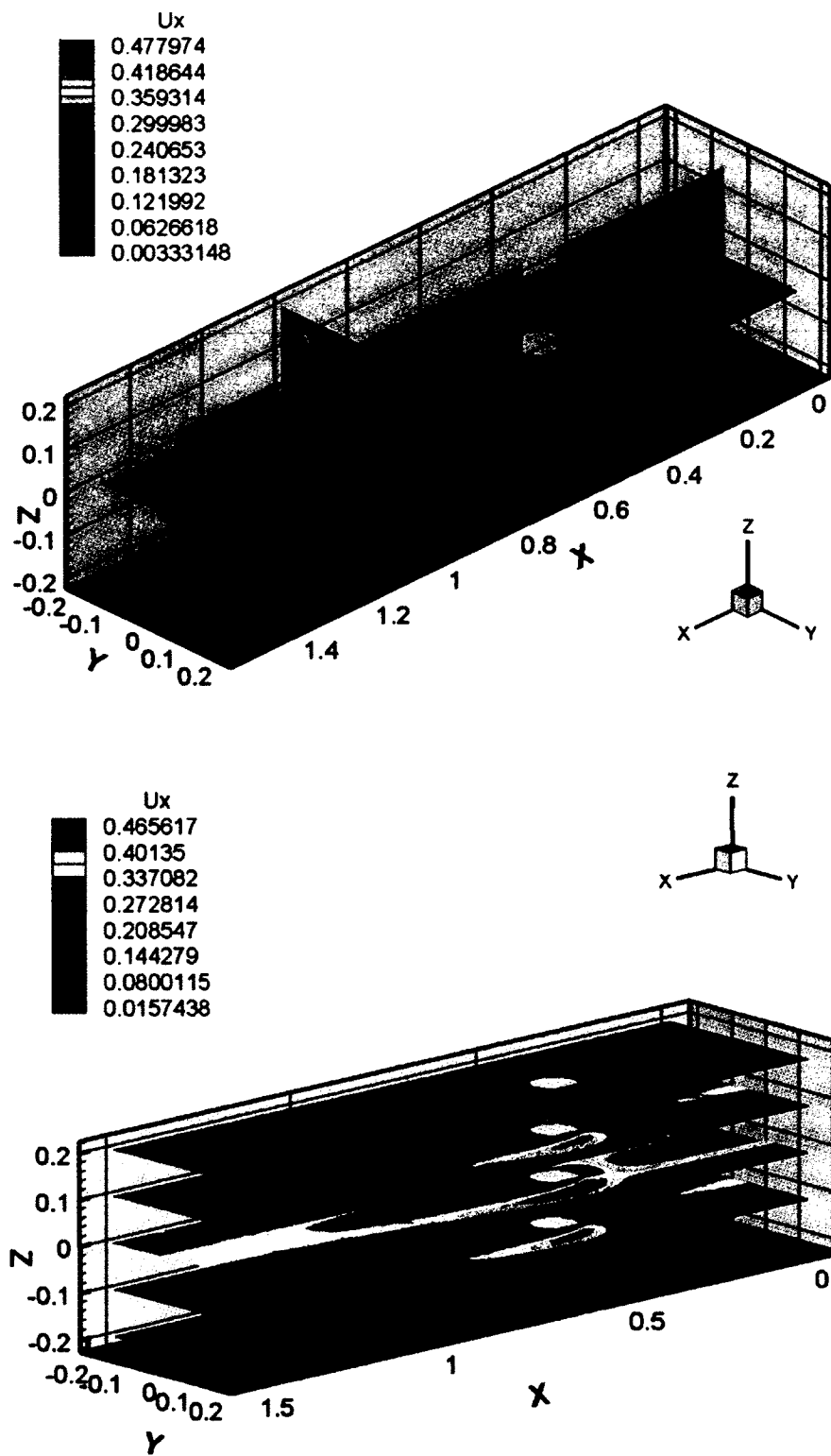


Figure 5-8: DG simulation: 3D simulation of flow passing cylinder: Contour profile of velocity u in the x direction, with 5920 elements and a polynomial order of 3.

Figures. 5-9, 5-10 and 5-11 present the contour plots of velocity in y and z directions and the pressure distribution, respectively. The view angles are adjusted individually for a better illustration.

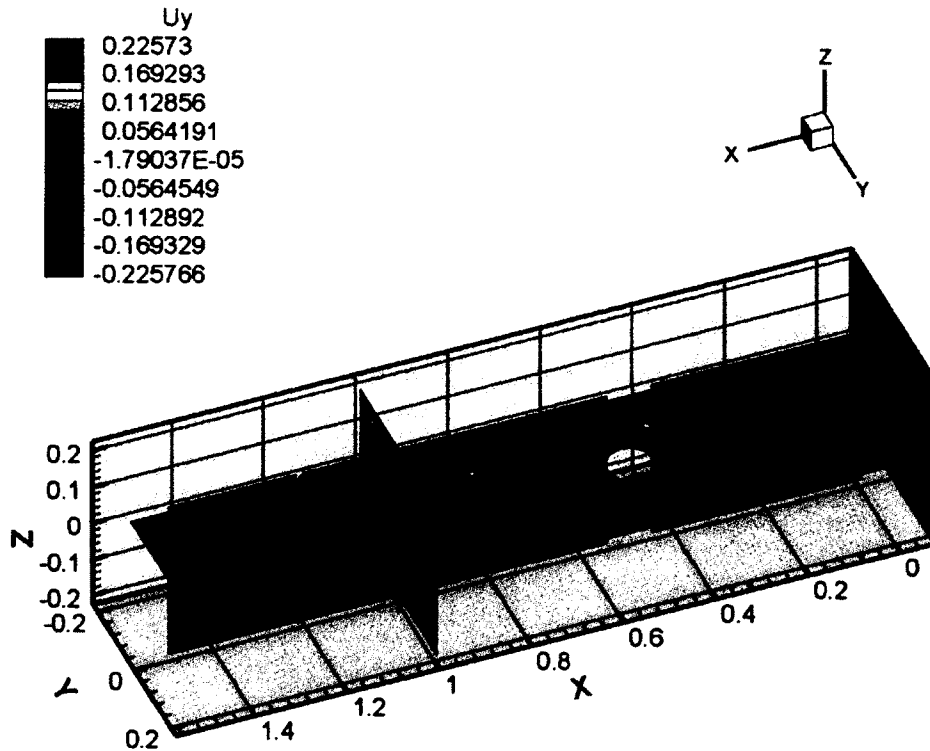


Figure 5-9: DG simulation: 3D simulation of flow passing cylinder: Contour profile of velocity v in the y direction, with 5920 elements and a polynomial order of 3.

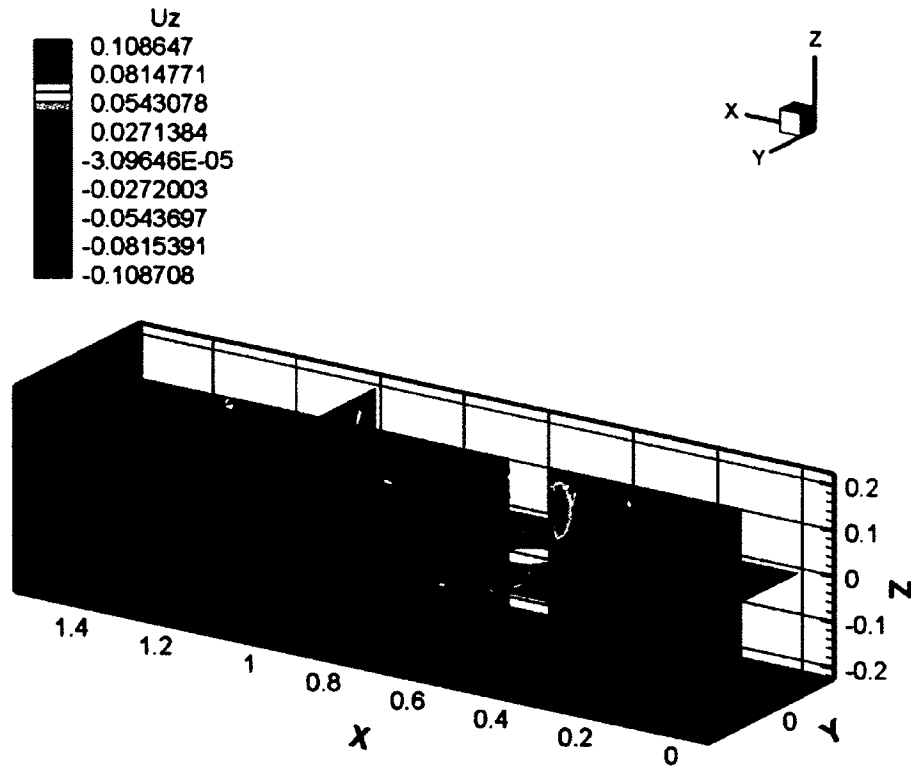


Figure 5-10: DG simulation: 3D simulation of flow passing cylinder: Contour profile of velocity w in the z direction, with 5920 elements and a polynomial order of 3.

Four independent simulations are performed with different settings (*hp*-refinement). Table 5-2 lists the computed drag force, the lift force and the pressure difference. Our simulation results have a good match with the reference values [82].

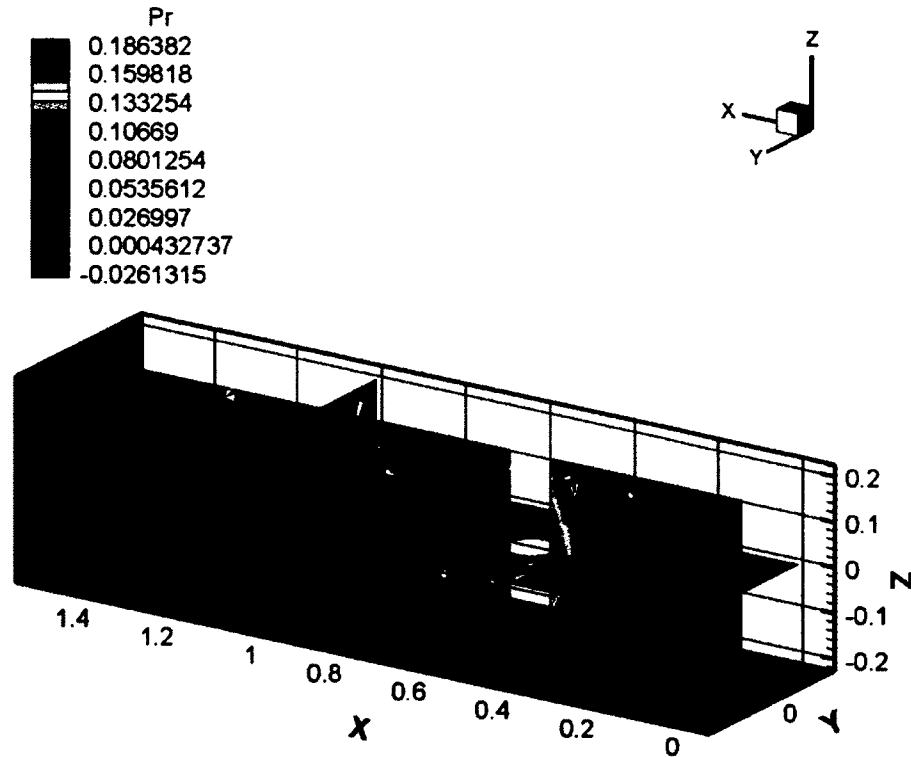


Figure 5-11: DG simulation: 3D simulation of flow passing cylinder: Contour profile of Pressure, with 5920 elements and a polynomial order of 3.

Table 5-2: DG: 3D simulation results compared to reference values

	Element Number	Polynomial order	Maximum Drag	Maximum Lift	Pressure Difference	Nodf
Numerical Results	2109	4	6.0044	0.0097	0.1577	295260
	2898	4	6.1059	0.0095	0.1611	405720
	3320	3	6.0977	0.0091	0.1546	265600
	5920	3	6.1105	0.0095	0.1613	473600
Reference Value	N/A	N/A	6.1295	0.0093	0.1693	2426292

5.3.3 Two Dimensional Simulation of Cavity Flow with High Reynolds Number

Two dimensional lid-driven cavity flow with Reynolds number of 1000 or 5000 is considered, respectively. Unstructured mesh consisted of 385 triangular elements is used in this simulation and the Lagrangian polynomial of the order 7 is chosen as the basis function. Since all the rich features of 2D cavity flow are mainly located at the three corners (right-upper corner excluded) of the domain, more dense meshes are put to these three corners while coarser meshes are at the other places, as shown in Figure 5-12. The intersections of red lines are interpolation points within each triangular elements.

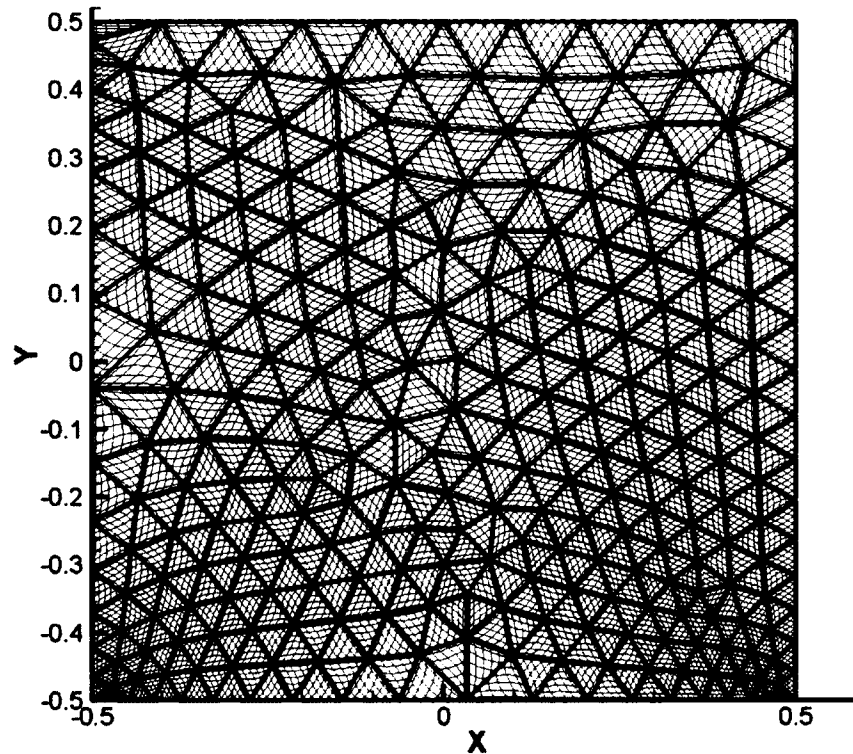


Figure 5-12: DG simulation: 2D lid-driven cavity flow: Mesh of 385 triangular elements.

Figure 5-13 illustrates the streamline of the 2D cavity flow with a Reynolds number of 1000 at a time of 40 seconds. We could see two separations locate at the bottom corners. Velocity profiles along the lines of $X = 0$ and $Y = 0$ are examined and

compared with the reference values from [83, 84]. Since an unsteady Navier-Stokes equation was considered, we choose a relatively steady velocity field at a time of 40 seconds for comparison. Comparisons are illustrated in Figures 5-14 and 5-15. The red line represents the reference values and the green squares are the numerical solutions.

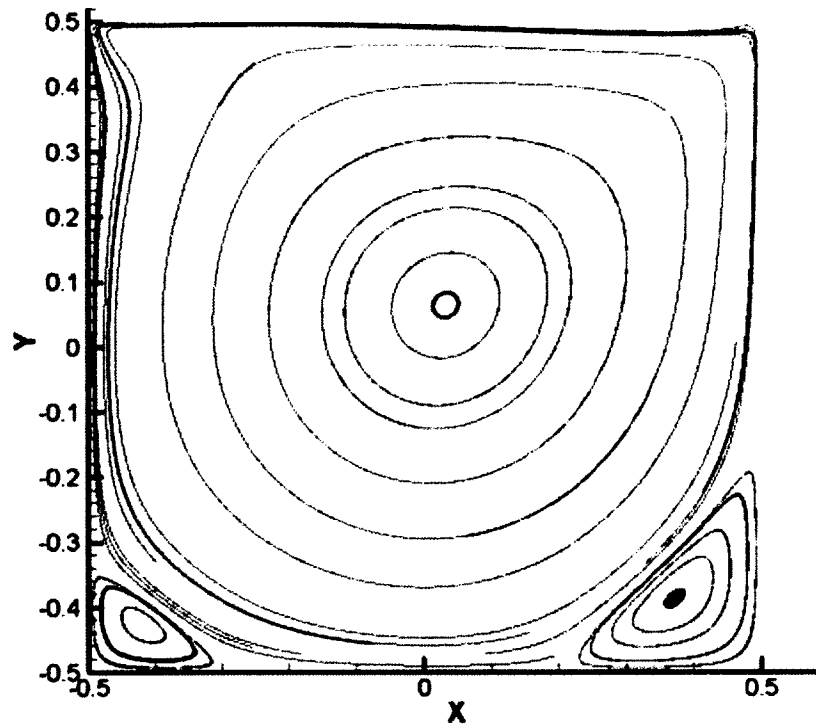


Figure 5-13: DG simulation: 2D lid-driven cavity flow: Streamline with Reynolds number 1000.

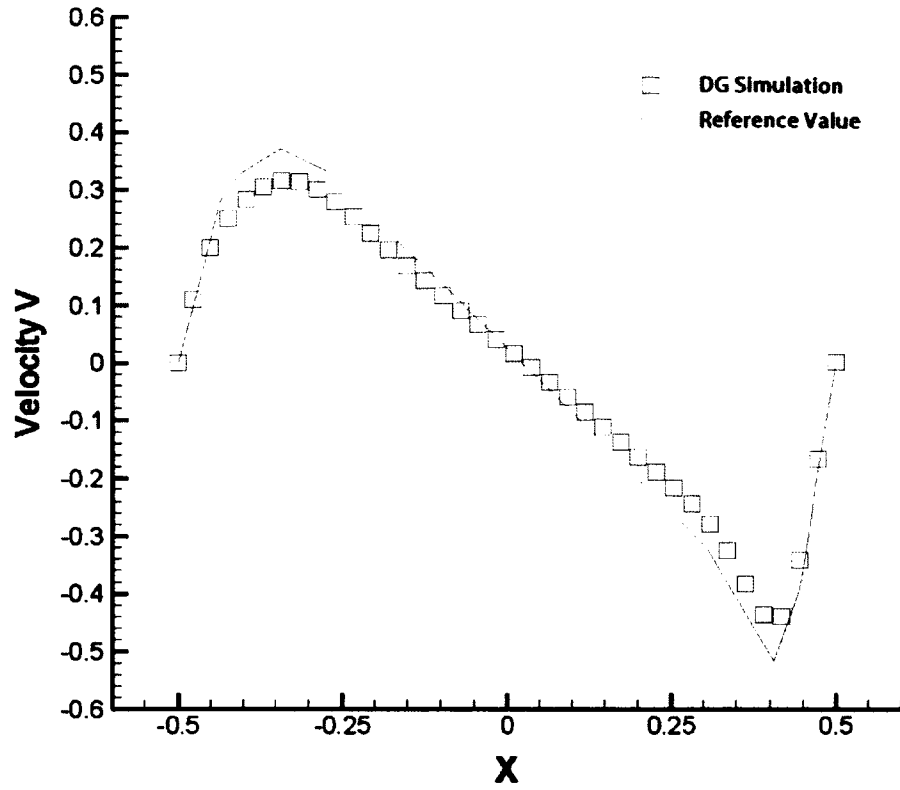


Figure 5-14: DG simulation: 2D lid-driven cavity flow: Velocity v versus X , Reynolds number 1000.

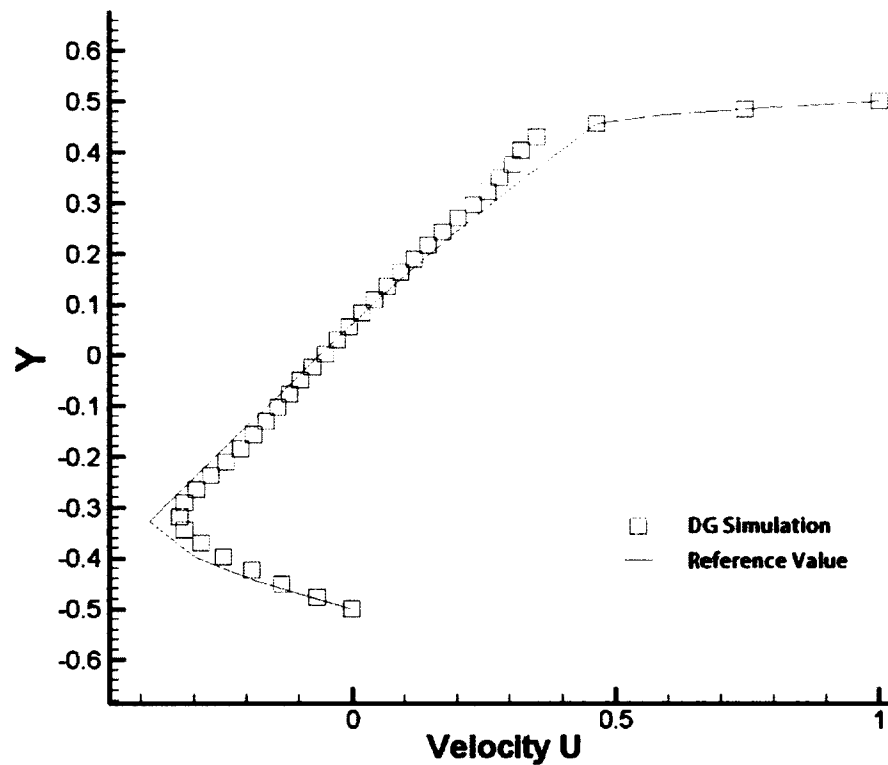


Figure 5-15: DG simulation: 2D lid-driven cavity flow: Velocity u versus Y , Reynolds number 1000.

Figure 5-16 shows the streamline of the 2D cavity flow of Reynolds number 5000 at a time of 80 seconds. More separations will occur at the three corners as the Reynolds number increases. Similarly, we examined the velocity profile at a time of 80 seconds located on lines $X = 0$ and $Y = 0$, respectively, and compared them to the reference value. As shown in Figures 5-17 and 5-18, simulations results are close to the values in reference papers [85, 84].

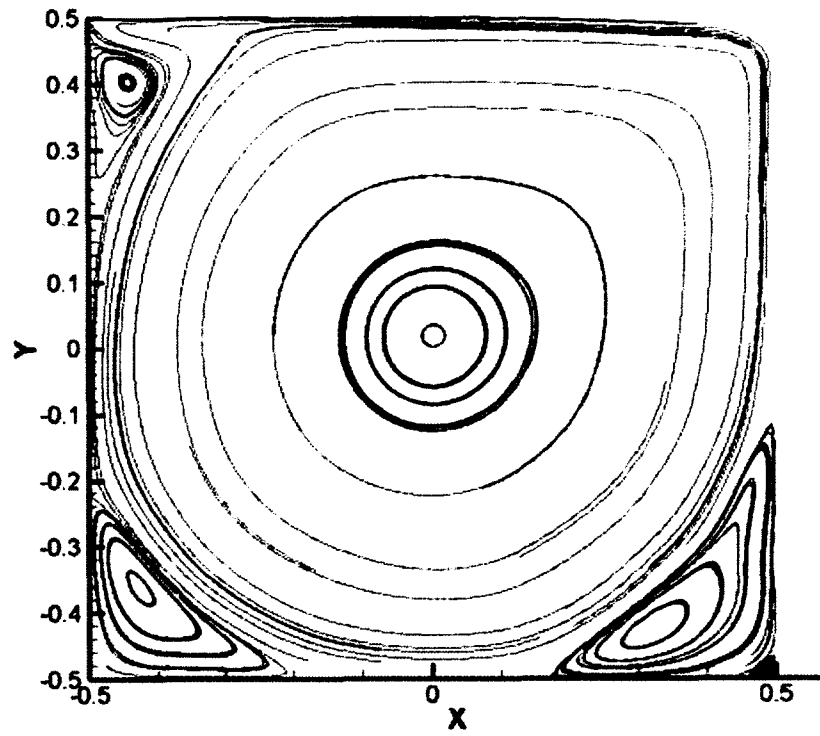


Figure 5-16: DG simulation: 2D lid-driven cavity flow: Streamline with Reynolds number 5000.

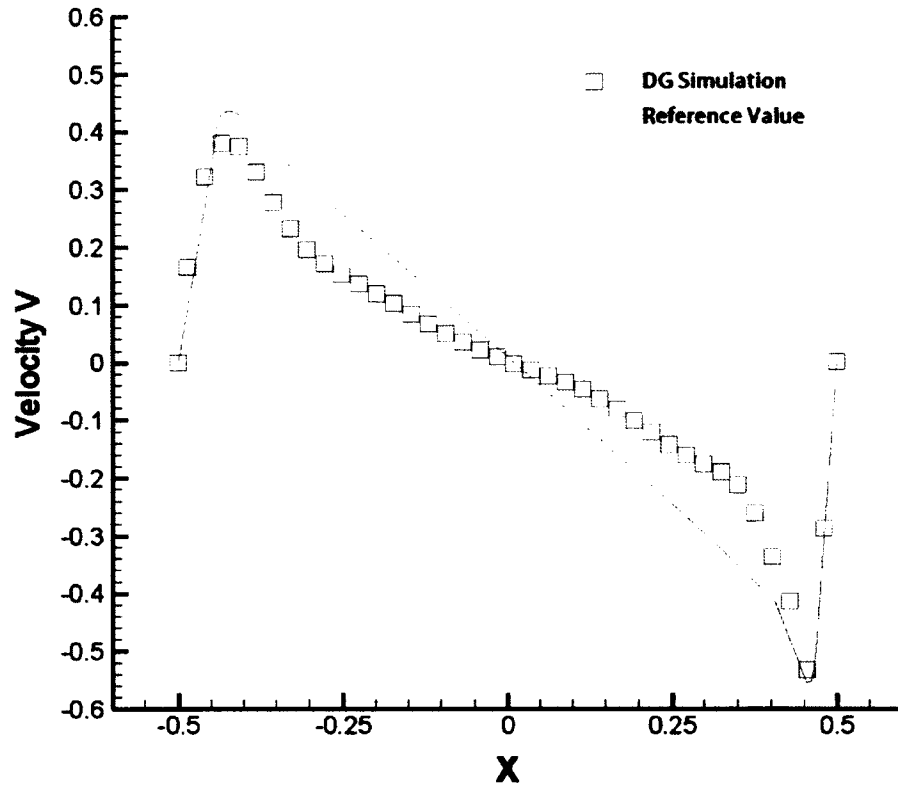


Figure 5-17: DG simulation: 2D lid-driven cavity flow: Velocity v versus X , Reynolds number 5000.

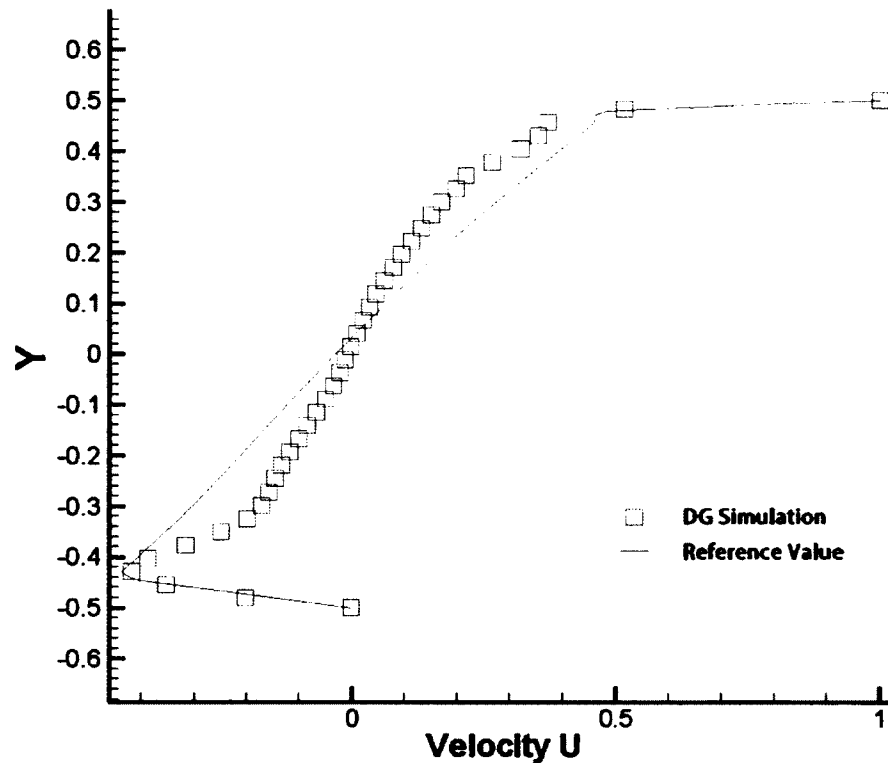


Figure 5-18: DG simulation: 2D lid-driven cavity flow: Velocity u versus Y , Reynolds number 5000.

5.3.4 Three Dimensional Simulation of Cavity Flow with High Reynolds Number

A 3D lid-driven cavity flow in a cube of unit length with Reynolds number of 1000 is considered. The cube is decomposed into 777 tetrahedron elements. Within each element, Lagrangian polynomial of an order of 4 is used as the basis function. The mesh adopted in this simulation is shown in Figure 5-19, in which the tetrahedron element is illustrated as black lines and inner interpolation points in each element are presented as the intersections of red lines. At the plane $Z = 0.5$, the flow is given a constant unit speed in the x direction, while other planes are assigned a non-slip boundary condition. We run the simulation till $T = 20$ seconds. Figure 5-20 and Figure 5-21 show the velocity contour of the 3D cavity flow in x and z directions (u and w), respectively. The slices as shown in

figure are in planes of $Y = -0.4$, $Y = 0$ and $Y = 0.4$. View angle are adjusted for a better viewpoint.

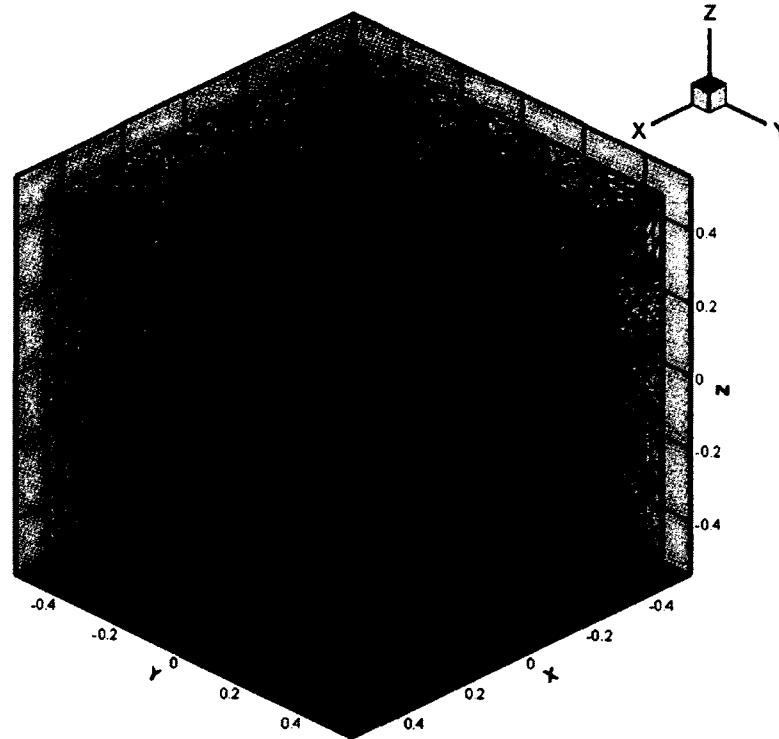


Figure 5-19: DG simulation: 3D lid-driven cavity flow: Mesh of 777 tetrahedron elements.

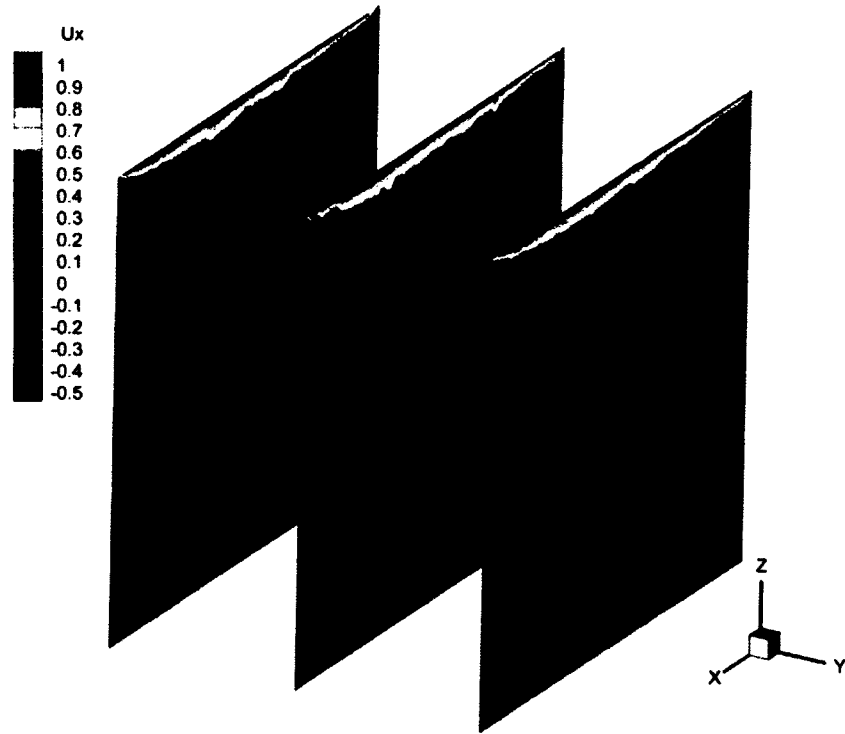


Figure 5-20: DG simulation: 3D lid-driven cavity flow with Reynolds number 1000: Velocity contour of u in planes of $Y = -0.4, Y = 0$ and $Y = 0.4$.

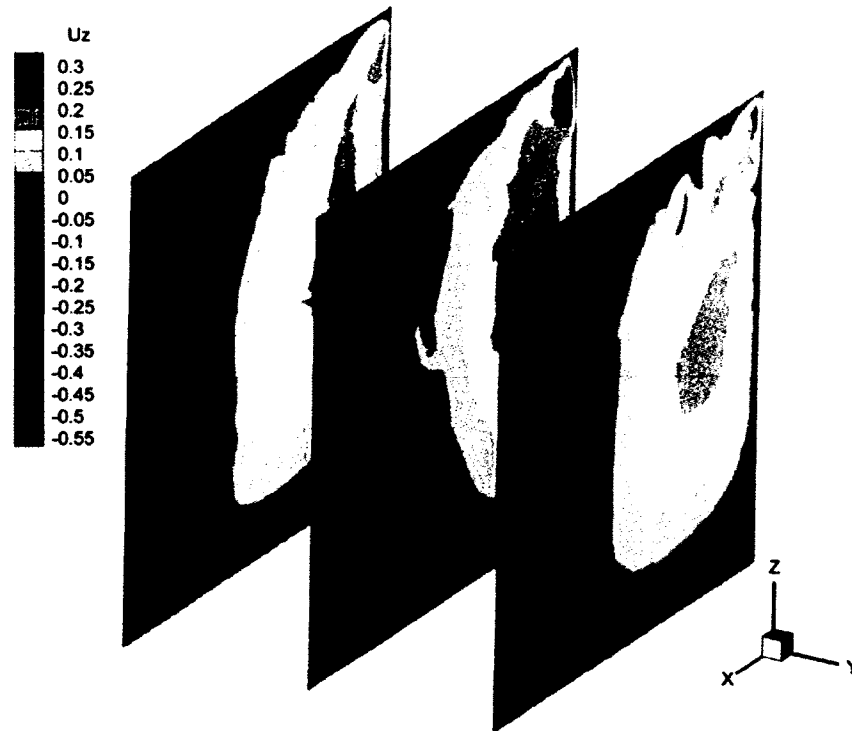


Figure 5-21: DG simulation: 3D lid-driven cavity flow with Reynolds number 1000: Velocity contour of w in planes of $Y = -0.4$, $Y = 0$, and $Y = 0.4$.

5.4 Conclusions

Two dimensional and three dimensional simulation results are presented in this chapter and have demonstrated that DG approach is advantageous to handle strong hyperbolic problems. Unstructured mesh such as using triangular element or tetrahedron element is used in our simulation to illustrate the flexibility of our DG solver for solving the problem with complex geometry. Therefore, Our DG solver can be further used to conduct research in electrodynamics, fluid mechanics and plasma physics.

For a larger scale of DG simulation (especially 3D simulation), parallel computing is necessary. For our 3D simulation, a single run of 5920 element and bases with a polynomial order of 3 took more than 38 hours for 8 seconds of real time simulation. Besides, the DG method strongly relies on numerical fluxes, and the

numerical fluxes are only applied at the interfaces of the adjacent elements or on the boundaries. By increasing the polynomial order of bases, the ratio of boundary points and interior points decreases, which means the proportion of total points with numerical fluxes decreases. Therefore, for gaining a better resolution, h -refinement is a must. Simply increasing the order of bases without introducing more elements will take longer time to gain a converged solution or, in some cases of simulating strong hyperbolic problem, even destroy the accuracy and fail to obtain a correct solution.

CHAPTER 6

SMOOTHED PARTICLE HYDRODYNAMICS AND GPU COMPUTING

6.1 Introduction

This chapter is based on my contribution to the submitted manuscript titled "Mesh-free GPU simulation of violent flows in rapidly distorting domains with Smoothed Particle Hydrodynamics." I am the first author of this paper and its content is used in Chapter 6 with proper referencing.

6.2 Formulation of Smoothed Particle Hydrodynamics

6.2.1 Fluid Particles

The momentum equation of fluid, in Lagrangian form, is as follows:

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\nabla P + \mu\Delta\mathbf{v} + \mathbf{f}_{ex}, \quad (92)$$

where P is the pressure, ρ is the density, and \mathbf{f}_{ex} stands for the external force. In order to conserve linear and angular momentum and stabilize the simulation, Gingold and Monaghan [38] reconstructed the original pressure term:

$$-\frac{1}{\rho}\nabla P + \mu\Delta\mathbf{v} = -\nabla\left(\frac{P}{\rho}\right) - \frac{P}{\rho^2}\nabla\rho + \mu\Delta\mathbf{v}. \quad (93)$$

By following the SPH formula [43], we have

$$\frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} + \mathbf{f}_{ex}, \quad (94)$$

in which, subscripts i and j denote the indices of particle of interest and the surrounding particles, respectively. For WCSPH, density ρ_i is given by [40, 42]:

$$\rho_i = \sum_j m_j W_{ij}, \quad (95)$$

and P_i is computed from equation of state [42], which depends on ρ_i only:

$$P_i = \frac{\rho_0 c_0^2}{7} \left[\left(\frac{\rho_i}{\rho_0} \right)^7 - 1 \right]. \quad (96)$$

Π_{ij} is the viscous pressure which has the following expression [86]:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c \tilde{\mu}_{ij} + \beta \tilde{\mu}_{ij}^2}{(\rho_i + \rho_j)/2}, & \text{if } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0, \\ 0, & \text{otherwise,} \end{cases} \quad (97)$$

in which, $\tilde{\mu}_{ij}$ is equal to $\frac{h \mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{r_{ij}^2 + 0.001 h^2}$ where h is the radius of influence region. The specific

values for α and β depend on the type of problems. For a low Mach number flow, $\alpha =$

0 and $0 < \beta < 1$; for a high Mach number flow, $\alpha = 1$ and $\beta > 1$.

6.2.2 Solid Particles

For an elastic body, since it involves stress tensor σ , we write the momentum equation in the component form [87]:

$$\frac{dv^a}{dt} = \frac{1}{\rho} \frac{\partial \sigma^{ab}}{\partial x^b} + f^a, \quad (98)$$

where f^a is the external force. To avoid the conflict in symbols, we use a and b to

represent Cartesian components. Tensor stress σ^{ab} is consisted of two components,

volumetric stress $P\delta^{ab}$ and deviatoric stress S^{ab} . To calculate the volumetric stress, it is

easy. However, for the deviator stress, we update it by adding the change of S^{ab} to the previous deviatoric stress. The rate of change of S^{ab} is given by:

$$\frac{dS^{ab}}{dt} = 2\mu \left(\dot{\epsilon}^{ab} - \frac{1}{3} \delta^{ab} \dot{\epsilon}^{ab} \right) + S^{ac} \Omega^{bc} + \Omega^{ac} S^{cb}, \quad (99)$$

where $\dot{\epsilon}^{ab}$ and Ω^{ab} are defined below:

$$\dot{\epsilon}^{ab} = \frac{1}{2} \left(\frac{\partial v^a}{\partial x^b} + \frac{\partial v^b}{\partial x^a} \right), \quad (100)$$

$$\Omega^{ab} = \frac{1}{2} \left(\frac{\partial v^a}{\partial x^b} - \frac{\partial v^b}{\partial x^a} \right). \quad (101)$$

To calculate the velocity derivatives, relative SPH scheme for velocity is adopted [88]:

$$\left(\frac{\partial v^a}{\partial x^b} \right)_i = - \sum_j \frac{m_j}{\frac{\rho_i + \rho_j}{2}} (v_i^a - v_j^a) \frac{\partial W_{ij}}{\partial x_i^b}. \quad (102)$$

For the momentum equation of the elastic body, the SPH scheme is:

$$\frac{dv_i^a}{dt} = \sum_j m_j \left(\frac{\sigma_i^{ab}}{\rho_i^2} + \frac{\sigma_j^{ab}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x_i^b} + f^a. \quad (103)$$

6.2.3 Particle Interactions

The interaction force exerted between particle "i" and particle "j" is described by the Lenard-Jones potential [41, 88]:

$$\mathbf{f}_{ij} = C_0 \left[\left(\frac{r_0}{|\mathbf{r}_{ij}|} \right)^{p_1} - \left(\frac{r_0}{|\mathbf{r}_{ij}|} \right)^{p_2} \right] \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}, \quad (104)$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, r_0 is the initial spacing between two particles, p_1 and p_2 are chosen to be 12 and 6, respectively. C_0 is an adjustable constant that depends on a particular problem.

6.2.4 Velocity Evaluation

Particle movement is computed using the XSPH scheme:

$$\frac{dr_i}{dt} = v_i + \varepsilon \sum_j m_j \left(\frac{v_j - v_i}{\bar{\rho}_{ij}} \right) W_{ij}, \quad (105)$$

with $\bar{\rho}_{ij} = \frac{\rho_i + \rho_j}{2}$ and $0 \leq \varepsilon \leq 1$ as a factor, which averages the velocity in the influence.

6.2.5 Time Evolution and Code Speedup

Although there are many options of high order time integration schemes, we choose the forward Euler method in this paper for simplicity. As an explicit method in time, the time step should be restricted by the CFL condition:

$$\Delta t < \frac{h}{c}, \quad (106)$$

where h is the characteristic length and c is the speed of propagation.

The procedures of the implementation for time evolution are shown in Figure 6-1. The values of velocity and positions are initialized at the beginning, and the densities of fluid and solid particles and pressure are computed with the position vector. Since the previous velocity of each particle is known, the deviatoric stress and viscous stress can be calculated. Next, using those known forces on particles, we could compute the acceleration of each particle. With the acceleration, we integrate to determine the values of the velocity and position of each particle at the next time level.

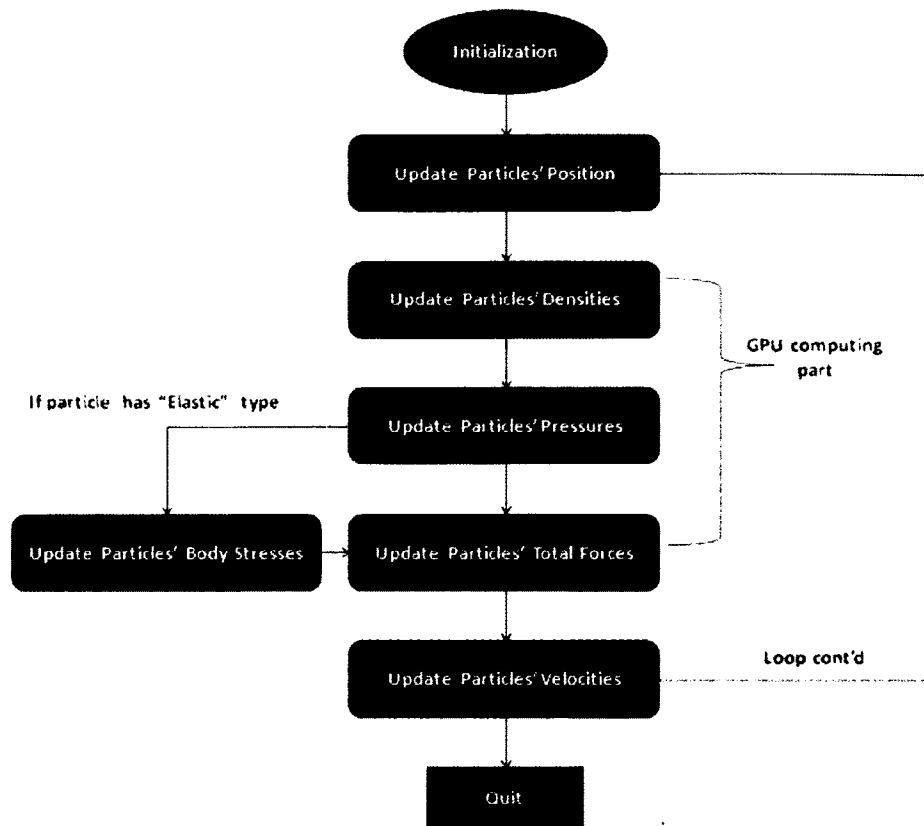


Figure 6-1: GPU SPH simulation: Schematic of the implementation procedures.

Several ways are applicable to speed up the SPH code. For instance, one approach is to improve serial searching algorithms, and another one is by parallel implementation. The former approach mainly is to improve the efficiency of searching for neighboring particles. An example is the tree search, which reduces the operation counts down to $O(N^2)$ to $O(N\log N)$. Another example is by introducing a link list, and only the neighboring particles within the list of neighbors of immediate neighbors are searched.

Perhaps the most intuitive approach is by using parallelization because SPH is ideal for parallelization and every particle is task-independent. We use NVIDIA CUDA C++ to write GPU codes and allow individual GPU thread to search for neighbor particles and compute forces. The flow chart of GPU version SPH is illustrated in Figure

6-1. The computational time is significantly reduced. However, since CUDA is still in its early stage of development, it does not support a composite or nested data structure, such as class or structure. Therefore, in order to utilize GPU to perform computations that use a complex data structure, we have to convert from a simple and raw data structure to the composite memory-access efficient data structure that we designed in our codes.

6.3 Numerical Results and Discussions

We present three test cases in both two dimensional and three dimensional spaces in order to illustrate better visual effects of results from GPU computing applications. Three different types of particles, fluid, elastic solid, and boundary particles are considered with different properties. The numerical simulations were performed on the desktop equipped with the Intel i5 processor with 16 GB memory and NVIDIA GTX 760 graphic card with 2 GB GDDR5 memory. Results are visually rendered immediately via the open graphic library (OpenGL).

6.3.1 Two Dimensional Simulation of Unsteady Nozzle Flow

A nozzle is a common fluid dynamics and fluid mechanics device to change the direction and rate of flow of the fluid through it. The “de Laval” nozzle is one of the most important one, since it is widely used in jet or rocket engines to obtain maximized kinetic energy in a certain direction.

To illustrate the characteristics of a transient jet flow ejected from a nozzle, we used GPU computing to simulate a 2D unsteady nozzle flow which adopts the WCSPH formulas (previously discussed in the Section Formulations of Smoothed Particle Hydrodynamics). In Figure 6-2, photographs of the flow ejected from a nozzle at a different flow rate in a thin conical jet by Dombrowski [89] was presented as a

comparison for our simulation. We set up a nozzle channel with the initial fluid velocity equal to a dimensionless speed of 20/second, ignoring the effect of the gravity force. In Figure 6-2, we can see that when the speed of outgoing flow increases beyond a critical point, the flow breaks into small droplets.

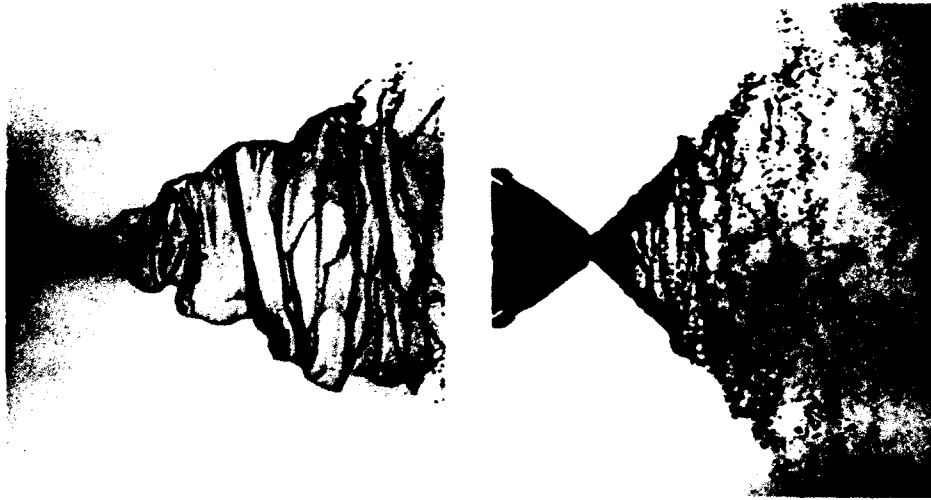


Figure 6-2: Unsteady nozzle flow: Flow ejected from a nozzle in a thin conical jet, photographed by N. Dombrowski [89].

Figure 6-3 sequentially shows the time evolution flow passing through this nozzle. The red color represents the high velocity particles, while, the blue color represents particles of the lower velocity. The total number of particles is 3,000. Although we did not use a lot of particles, the effect of the boundary layer on the wall of the nozzle was created. The second and third subfigures in Figure 6-3 show what happens in a Hagen–Poiseuille flow. The last three subfigures in Figure 6-3 show the process of droplet formation due to the effect of surface tension. The entire GPU simulation took only 40 seconds on a single GPU card as previously described.

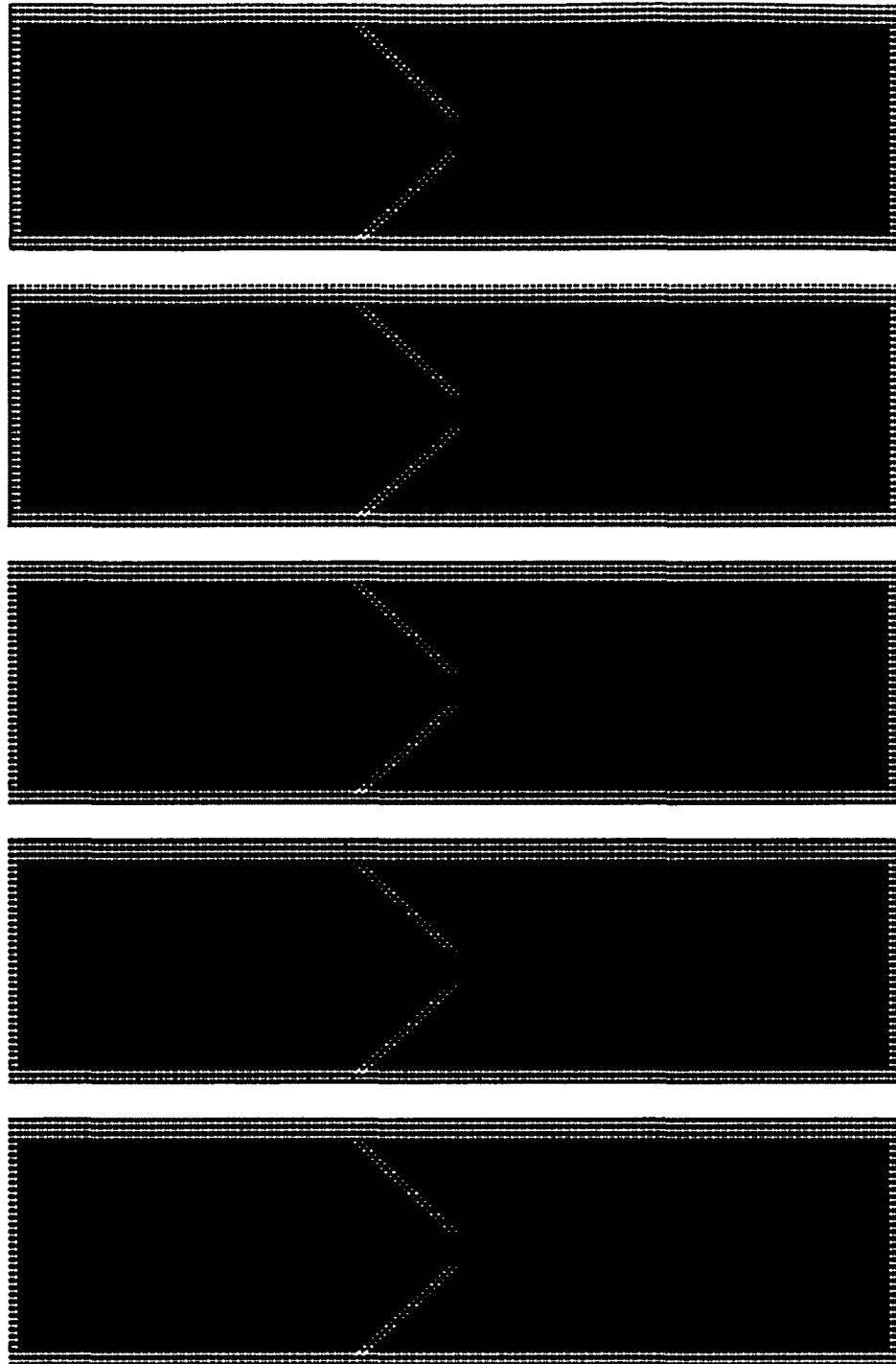


Figure 6-3: Unsteady nozzle flow: Five snapshots of a 2D unsteady nozzle flow.

6.3.2 Two Dimensional Simulation of Vortex Shedding

Vortex shedding formed by viscous fluid separated from a cylinder at Reynolds number between 40 and 150 is of engineering importance [90]. For example, in the

designing of bridges and offshore oil rigs, ignoring the effect of vortex shedding could result in serious safety issues and potential loss of equipment and lives [91]. Simulation of vortex shedding with SPH is rare in open literature [90, 92, 93, 94].

Figure 6-4 is a photo [89] for the experiment of the Von Kármán laminar vortex street [95]. We could see in Figure 6-4 that the vortices created behind the cylinder were separated from the boundary layer near the cylindrical surface which periodically detached from both sides of the cylinder due to the adverse pressure zone created by skin friction on the cylinder. For comparison, a 2D GPU simulation of a similar flow is illustrated in Figure 6-5. The flow has an initial dimensionless speed of 15 per seconds with a peak Reynolds number of 140. The only difference is that in our simulation, we use a narrow domain in order to use fewer particles and to speed up our GPU computing. The inflow and outflow are treated as periodic boundaries. This simulation used a total of 16,000 particles and it took 10 minutes and 22 seconds to spin the entire flow and to reach the stage of steady vortex street shedding. This is much faster than using a mesh-based conventional method.



Figure 6-4: SPH simulation of Vortex Shedding: Von Kármán vortex street behind a circular cylinder at Reynolds number 140, photographed by Sadatoshi Taneda [89].



Figure 6-5: SPH simulation of Vortex Shedding: Simulated laminar 2D vortex street at Reynolds number 140.

Figure 6-6 is a photo [89] for the experiment of the flow passing a cylinder with a unit radius at Reynolds number 105. Using the results from our self-developed GPU codes, a matching run with the same setting as in Figure 6-6 was performed and shown in Figure 6-7. The domain is narrower than Figure 6-6 so that this GPU computing could render more resolution with 34,000 particles. The total GPU computing time is 35 minutes. From the comparison between Figure 6-6 and Figure 6-7, we could claim that our GPU simulation captured the key characteristics of this vortex shedding phenomenon at the Reynolds number 105 with a relatively economical computational cost.



Figure 6-6: SPH simulation of Vortex Shedding: Von Kármán vortex street behind a circular cylinder at Reynolds number 105, photographed by Sadatoshi Taneda [89].



Figure 6-7: SPH simulation of Vortex Shedding: Simulated laminar 2D vortex street at Reynolds number 105.

6.3.3 Two Dimensional and Three Dimensional Simulation of Elastic Solid and Fluid Interaction

Many engineering problems involve a multi-phase media and a multi-phase flow, such as the dynamics of the weather development, the ocean circulations, and the near-shore sediment transportation. Under certain situations, SPH could be efficient in simulating some complex two-phase flow problems. In the following, we present two-phase interactions of elastic objects and the fluid in two- and three dimensional spaces.

A two dimensional simulation of the interaction of the fluid with a solid elastic cube and sphere at different times is shown in Figure 6-8. The size of the simulation domain is 40 by 65 in dimensionless units, the same as below, surrounded by boundaries which consist of solid particles. The side of the cube is 3 and the diameter of the sphere is 3 as well. The density of the fluid is scaled to 1 and the density of both the cube and sphere is 0.6. Both solids are free falling under gravity before colliding with the collapsed fluid beam on the left. Initially, the fluid is fixed still on the left side of the domain by a confinement immediately on its right. Once the simulation starts, the confinement is removed and the fluid is released from the left under gravity and then interacts with the elastic solids. Both objects are washed by the inertia of the fluid with some spinning. The color of the fluid particles indicates their magnitude of velocity. Zero velocity is in blue and the higher velocity is in red with purple and pink, which denote the intermediate

values. A total of 5,500 particles are used in this simulation and it takes 3 minutes to complete this GPU computing, as shown in Table 6-1, compared with 14 minutes for a serial run on a single i5 CPU. Therefore, the GPU run is about 4~5 times faster than the CPU run.

Similarly, a 3D simulation of one solid elastic sphere of diameter 5 interacting with the fluid is shown in Figure 6-9. The simulation domain, a little larger than the 2D run, is 40 by 75 by 16, exactly the same density settings as in the 2D case used here. Four snapshots indicate the free falling and interaction processes at different moments. The velocity magnitude of the fluid is indicated by the same color map as in the previous run. For a better visual effect, only the fluid and sphere are displayed and the boundary walls are skipped. A total of 28,000 particles are used to render the three dimensional effect. Compared with over two hours of CPU time with a serial algorithm without GPU computing, as shown in Table 6-1, the total run time for the GPU simulation is only less than 29 minutes. Therefore, the GPU run is at least four times faster than the CPU run.



Figure 6-8: SPH simulation of imping flow: Snapshots of 2D solid-fluid interaction.

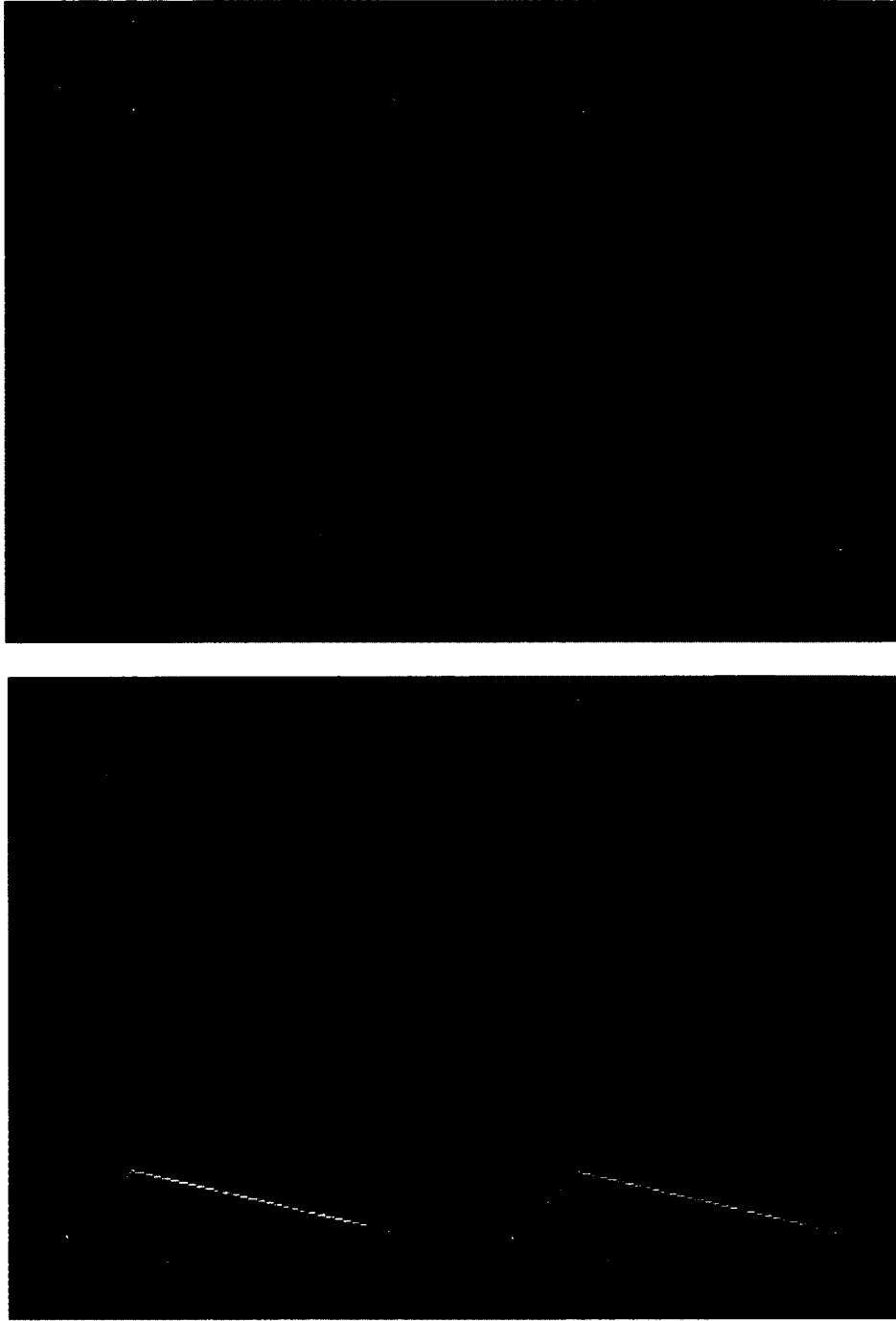


Figure 6-9: SPH simulation of impinging flow: Snapshots of 3D solid-fluid interaction.

Table 6-1: GPU SPH: Execution time (seconds) and specifics of GPU and CPU runs.

Simulations	Number of Particles	GPU Version time (seconds)	Serial Version time (seconds)	Time Ratio
Unsteady nozzle flow	3,000	40	255	6.38
Vortex shedding Ex. (1)	16,000	622	2,605	4.18
Vortex shedding Ex. (2)	34,000	2,100	8,176	3.89
2D impinging flow	5,500	180	840	4.67
3D impinging flow	28,000	1,731	6,840	3.95

6.3.4 Simulation Execution Time

In this section, the execution time for GPU simulation versus CPU simulation is listed in Table 6-1. For all these five runs with exactly the same number of particles and unknowns, the GPU computing which does have a parallel computing involved is 4 to 6 times faster than the corresponding CPU computation.

6.4 Conclusions

Simulation results have demonstrated that SPH, the mesh-free approach, is advantageous to handle problems involving free-surface flows, wave breaking, two-phase flows and polymorphic domain distortion. However, SPH is not an intrinsically accurate method due to its formulation and nature. To improve the resolution and accuracy, large numbers of particles are required for the simulation and parallel implementation is essential. A GPU-based multi-thread parallel approach could achieve a noticeable speedup comparing to a CPU serial algorithm.

Several limits restrict the GPU computing. First, the memory on a single GPU card is limited which limits the maximum number of particles allowed for a simulation. Second, the total number of CUDA cores is finite. For example, our GPU card has 1,024 cores. Third, GPU computing requires copying data back and forth between the CPU memory and GPU memory. This operation depends on the latency of the system, i.e., the bus bandwidths of both GPU and motherboard, and the cache size of the system. Therefore, for large simulations such as one involving 100,000 particles, distributing the total computational load to multiple GPUs is necessary to accelerate the speed of computation even further.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this dissertation, three different numerical methods are discussed individually and validations of our general solvers are tested by solving four different problems with both complex physics and complex geometry. The application areas for each method are discussed.

In Chapter 3 and Chapter 4, the Spectral Element Method (SEM) based on a structured mesh is used to provide high order accurate solutions for a natural convection problem and reaction-diffusion problem in neuromuscular junction (NMJ). In Chapter 5, Discontinuous Galerkin (DG) method based on an unstructured mesh is used to give high order accurate solutions for fluid problem with moderately high Reynolds number. In Chapter 6, the mesh-free Smoothed Particle Hydrodynamics method is used to provide reasonable solutions to the fluid problem with rapid domain deformation and discontinuity.

The focus of the future work will address the following topics:

1. To use the SEM solver to conduct further research in engineering simulations in areas of heat and mass transfer and computational fluid dynamics.
2. To develop a parallel version of DG solver based on MPI and test the code on supercomputers. Adapt the optimized parallel version of numeric libraries into

the DG solver, such as ScaLAPACK to improve the solver. Use the DG solver to conduct further research in engineering simulations.

3. SPH simulation based on GPU computing is an emerging area with a bright future. Further engineering applications in simulating near-shore wave breaking, micro-nano-fluids, and so on will be conducted in using this GPU solver. At the same time, a MPI version of the SPH will be developed and tested on supercomputers.

REFERENCE

- [1] W. Dai and R. Nassar, "A compact finite difference scheme for solving a one dimensional heat transport equation at the microscale," *J. Computational and Applied Mathematics*, vol. 132, pp. 431-441, 2001.
- [2] L. Gamet, F. Ducros, F. Nicoud and T. Poinso, "Compact finite difference schemes on non-uniform meshes. Application to direct numerical simulations of compressible flows," *Int. J. Numerical Methods in Fluids*, vol. 29, pp. 159-191, 1999.
- [3] S. Lele, "Compact finite difference schemes with spectral-like resolution," *Journal of Computational Physics*, pp. 16-42, 1992.
- [4] D. Liu, W. Kuang and A. Tangborn, "High-order compact implicit difference methods for parabolic equations in Geodynamo simulation," *Advances in Mathematical Physics*, vol. 56, pp. 82-96, 2009.
- [5] K. Mahesh, "A family of high order finite difference schemes with good spectral resolution," *Journal of Computational Physics*, vol. 145, pp. 332-358, 1989.
- [6] A. Patera, "A spectral element method for fluid dynamics - Laminar flow in a channel expansion," *Journal of Computational Physics*, vol. 54, pp. 468-488, 1984.
- [7] G. Karniadakis and S. Sherwin, *Spectral/hp element methods for computational fluid dynamics*, Oxford University Press, 2005.
- [8] C. Pozrikidis, *Introduction to finite and spectral element method using MATLAB*, Taylor and Francis Group, 2005.
- [9] A. Beskok and T. Warburton, "An unstructured hp Finite-Element scheme for fluid flow and heat transfer in moving domain," *Journal of Computational Physics*, vol. 174, no. 2, pp. 492-509, 2002.

- [10] S. Dong, D. Liu, M. Maxey and G. Karniadakis, "Spectral distributed Lagrange multiplier method: Algorithm and Benchmark test," *Journal of Computational Physics*, vol. 195, pp. 695-717, 2004.
- [11] F. Giraldo and T. Warburton, "A nodal triangle-based spectral element method for the shallow water equations on the sphere," *Journal of Computational Physics*, vol. 207, pp. 129-150, 2005.
- [12] J. Hesthaven and D. Gottlieb, "Stable spectral methods for conservation laws on triangles with unstructured grids," *J. Comput. Methods Appl. Mech. Engrg*, vol. 175, pp. 361-381, 1999.
- [13] D. Liu, Q. Chen and Y. Wang, "Spectral element modeling of sediment transport in shear flows," *J. Comput. Methods Appl. Mech. Engrg.*, vol. 200, pp. 1691-1707, 2011.
- [14] D. Liu, E. Keaveny, M. Maxey and G. Karniadakis, "Force-coupling method for flows with ellipsoidal particles," *Journal of Computational Physics*, vol. 228, pp. 3559-3581, 2009.
- [15] G. Karniadakis and A. Beskok, *Microflows: Fundamentals and simulation*, New York: Springer Verlag, 2002.
- [16] D. Liu, M. Maxey and G. Karniadakis, "Modeling and optimization of colloidal micro-pumps," *J. Micromechanics and Microengineering*, vol. 14, pp. 567-575, 2004.
- [17] D. Liu, M. Maxey and G. Karniadakis, "J. Micromechanics and Microengineering," *Simulations of dynamic self-assembly of paramagnetic microspheres in confined microgeometries*, vol. 15, pp. 2298-2308, 2005.
- [18] B. Cockburn, G. Karniadakis and C. Shu, *Discontinuous Galerkin methods. Theory, computation and applications*, Berlin: Springer Verlag, 2000.
- [19] B. Cockburn, S. Lin and C. Shu, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems.," *Journal of Computational Physics*, vol. 84, pp. 90-113, 1989.
- [20] B. Cockburn and C. Shu, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework," *Mathematics of Computation*, vol. 52, pp. 411-435, 1989.

- [21] A. Engsig-Karup, J. Hesthaven, H. Bingham and P. Madsen, "Nodal DG-FEM solutions of high-order Boussinesq-type equations," *J. Eng. Math.*, vol. 56, pp. 351-370, 2006.
- [22] C. Eskilsson and S. Sherwin, "Spectral/hp discontinuous Galerkin methods for modelling 2D Boussinesq equations," *Journal of Computational Physics*, vol. 212, pp. 566-589, 2006.
- [23] B. Li, *Discontinuous finite elements in fluid dynamics and heat transfer*, Springer, 2006.
- [24] T. Amada, M. Imura, Y. Yasumuro, Y. Manabe and K. Chihara, "Particle-based fluid simulation on the GPU," *Proc. ACM Workshop on General-purpose Computing on Graphics Processors*, 2003.
- [25] C. Antocj, M. Gallati and S. Sibilla, "Numerical simulation of fluid-structure interaction by SPH," *Computers and Structures*, vol. 85, pp. 879-890, 2007.
- [26] R. A. Dalrymple and B. D. Rogers, "Numerical modeling of water waves with SPH method," *Coastal Engng.*, vol. 53, pp. 141-147, 2006.
- [27] M. B. Liu and G. R. Liu, "Smoothed particle hydrodynamics (SPH): An overview and recent developments," *Archives of Computational Methods in Engineering*, vol. 17, pp. 25-76, 2010.
- [28] S. J. Sherwin and G. E. Karniadakis, "A triangular spectral element method: Applications to the incompressible Navier-Stokes equations," *Comp. Meth. Appl. Mech. Eng.*, vol. 123, pp. 189-229, 1995.
- [29] S. Sherwin and G. Karniadakis, "Tetrahedral hp finite elements: Algorithms and flow simulations," *Journal of Computational Physics*, vol. 124, pp. 14-45, 1996.
- [30] C. Schwab, *p and hp-Finite element methods*, Oxford University Press, 1998.
- [31] D. Gottlieb and S. A. Orszag, *Numerical analysis of spectral methods: Theory and applications*, Regional Conference Series in applied mathematics, 1977.
- [32] G. Chavent and B. Cockburn, "The local projection p_0 - p_1 discontinuous Galerkin finite element for scalar conservation laws," *M2AN*, vol. 23, pp. 565-592, 1989.

- [33] J. Qiu and C. W. Shu, "A comparison of troubled cell indicators for Runge-Kutta discontinuous Galerkin methods using WENO limiters," *SIAM J. Sci. Comput.*, vol. 27, pp. 995-1013, 2005.
- [34] H. Atkins and C. W. Shu, "Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations," *AIAA J.*, vol. 36, pp. 775-782, 1998.
- [35] F. Giraldo, J. Hesthaven and T. Warburton, "Nodal high-order discontinuous Galerkin method for the spherical shallow water equations," *Journal of Computational Physics*, vol. 181, pp. 499-525, 2002.
- [36] J. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: Algorithms, analysis, and applications*, Springer, 2008.
- [37] R. M. Kirby and G. E. Karniadakis, "Selecting the numerical flux in discontinuous Galerkin methods for diffusion problems," *Journal of Scientific Computing*, vol. 22, no. DOI: 10.1007/s10915-004-4145-5, 2005.
- [38] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: Theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, vol. 181, pp. 375-389, 1977.
- [39] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *Astron. J.*, vol. 82, pp. 1013-1024, 1977.
- [40] J. J. Monaghan, "An introduction to SPH," *Comp. Phys. Comm.*, vol. 48, pp. 89-96, 1988.
- [41] G. R. Liu and M. B. Liu, *Smoothed particle hydrodynamics: A mesh-free particle method*, Singapore: World Scientific, 2003.
- [42] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annual Review of Astronomy and Astrophysics*, vol. 30, pp. 543-574, 1992.
- [43] J. J. Monaghan, "Simulating free surface flows with SPH," *Journal of Computational Physics*, vol. 110, pp. 339-406, 1994.
- [44] J. J. Monaghan, "Smoothed particle hydrodynamics," *Rep. Prog. Phys.*, vol. 68, pp. 1703-1759, 2005.

- [45] S. J. Lind, R. Xu, P. K. Stansby and B. D. Rogers, "Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves," *Journal of Computational Physics*, vol. 231, pp. 1499-1523, 2012.
- [46] R. A. Dalrymple, M. Gómez-Gesteira, B. D. Rogers, A. Z. S. Panizzo, A. J. C. Crespo, G. Cuomo and M. Narayanaswamy, "Smoothed particle hydrodynamics for water waves, advances in numerical simulation of nonlinear water waves," *World Scientific*, 2009.
- [47] L. D. Libersky and A. G. Petschek, "Smooth particle hydrodynamics with strength of materials, advances in the free lagrange method," *Lecture Notes in Physics*, vol. 395, pp. 248-257, 1990.
- [48] T. Rabczuk, J. Eibl and L. Stempniewski, "Simulation of high velocity concrete fragmentation using SPH/MLSPH," *Int. J. Numer. Methods Eng.*, vol. 56, pp. 1421-1444, 2003.
- [49] S. Shao and E. Y. M. Lo, "Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface," *Adv. Water Resour.*, vol. 26, pp. 787-800, 2003.
- [50] P. W. Randles and L. D. Libersky, "Recent improvements in SPH modeling of hypervelocity impact," *Int. J. Impact Eng.*, vol. 20, pp. 525-532, 1997.
- [51] P. W. Randles and L. D. Libersky, "Smoothed particle hydrodynamics: some recent improvements and applications," *Comput. Methods Appl. Mech. Eng.*, vol. 139, pp. 375-408, 1996.
- [52] D. Deb and R. Pramanik, "Failure process of brittle rock using smoothed particle hydrodynamics," *Journal of Engineering Mechanics*, no. (ASCE) EM.1943-7889.0000592, 2013.
- [53] E. S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence and P. Stansby, "Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method," *Journal of Computational Physics*, vol. 227, pp. 8417-8436, 2008.
- [54] Y. Wang, D. Liu and H. Zhang, "Spectral nodal element simulation of conjugate heat and mass transfer: Natural convection subject to chemical reaction along a circular cylinder," *International Journal of Computer Science and Application*, vol. 3, pp. 51-55, 2014.
- [55] A. Bejan, *Convection heat transfer*, 2 ed., Wiley, 1993.

- [56] P. J. Lauriello, "Application of a convective heat source to the thermal fracturing of rock," *International Journal of Rock Mechanics and Mining Sciences & Geomechanics*, vol. 11, no. 2, pp. 75-81, 1974.
- [57] D. Nield and A. Bejan, *Convection in porous media*, 2 ed., New York: Springer, 1998.
- [58] L. S. Oliveira, M. Fortes and K. Haghighi, "Conjugate analysis of natural convective drying of biological materials," *Drying Technology*, vol. 12, no. 5, pp. 1167-1190, 1994.
- [59] L. Z. Zhang, S. M. Huang and L. X. Pei, "Conjugate heat and mass transfer in a cross-flow hollow fiber membrane contactor for liquid desiccant air dehumidification," *International Journal of Heat and Mass Transfer*, vol. 55, pp. 8061-8072, 2012.
- [60] T. Defraeya, B. Blocken and J. Carmeliet, "Analysis of convective heat and mass transfer coefficients for convective drying of a porous flat plate by conjugate modelling," *International Journal of Heat and Mass Transfer*, vol. 55, pp. 1-3, 2012.
- [61] A. S. Dorfman, "Methods of estimation of coefficients of heat transfer from nonisothermal walls," *Heat Transfer-Soviet Research*, vol. 15, no. 6, pp. 35-57, 1985.
- [62] A. Dorfman and Z. Renner, "Conjugate problems in convective heat transfer: Review," *Mathematical Problems in Engineering*, no. Article ID 927350, 2009.
- [63] A. M. Kawala and S. N. Odda, "Numerical investigation of unsteady free convection on a vertical cylinder with variable heat and mass flux in the presence of chemically reactive species," *Advances in Pure Mathematics*, vol. 3, pp. 183-189, 2013.
- [64] N. Nitin and M. V. Karwe, "Numerical simulation and experimental investigation of conjugate heat transfer between a turbulent hot air jet impinging on a cookie-shaped object," *Journal of Food Science*, vol. 69, pp. 59-65, 2004.
- [65] Z. Liu, R. Wan, K. Muldrew, S. Sawchuk and J. Rewcastle, "A level set variational formulation for coupled phase change/mass transfer problems: Application to freezing of biological systems," *Finite Elements in Analysis and Design*, vol. 40, no. 12, pp. 1641-1663, 2004.

- [66] W. M. Rohsenow, *Handbook of Heat Transfer*, McGraw-Hill, 1998.
- [67] Y. Wang, D. Liu and M. A. Decoster, "Spectral element simulation of reaction diffusion system in the Neuromuscular junction," *Journal of Applied and Computational Mathematics*, vol. 2, 2013.
- [68] D. Aidley, *The physiology of excitable cells*, 4 ed., Cambridge University Press, 1998.
- [69] E. Kandel, *Principles of neuroscience*, New York: McGraw-Hill, 2000.
- [70] D. Junge, "Nerve and muscle excitation," Sinauer Associates Inc, Sunderland, 1992.
- [71] G. Matthews, *Cellular physiology of nerve and muscle*, Blackwell Scientific Publications, 1991.
- [72] D. Purves, "Neuroscience," Sinauer Associates Inc, Sunderland, 2007.
- [73] A. Khaliq, F. Jenkins, M. Decoster and W. Dai, "A new 3D mass diffusion-reaction model in the neuromuscular junction," *Journal of Computational Neuroscience*, vol. 30, pp. 729-745, 2011.
- [74] H. Lester, "The response to Acetylcholine," *Scientific American*, vol. 236, pp. 106-117, 1977.
- [75] A. Friboulet and D. Thomas, "Reaction-diffusion coupling in a structured system: application to the quantitative simulation of endplate currents," *Journal of Theoretical Biology*, vol. 160, pp. 441-455, 1993.
- [76] J. Schild, "Afferent synaptic drive of rat medial nucleus tractus solitarius neurons: dynamic simulation of graded vesicular mobilization, release, and non-nmda receptor kinetics," *Journal of Neurophysiology*, vol. 74, pp. 1529-1548, 1995.
- [77] G. Hess and J. Andrews, "Functional Acetylcholine receptor-electroplax membrane microsacs: Purification and characterization," *Proceedings of the National Academy of Sciences*, vol. 74, p. 482-486, 1977.
- [78] G. Karniadakis, M. Israeli and S. Orszag, "High-order splitting methods for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 97, pp. 414-443, 1991.

- [79] B. Cockburn and C. W. Shu, "The local discontinuous Galerkin finite element method for convection-diffusion systems," *SIAM J. Numer. Anal.*, pp. 2440-2463, 1998.
- [80] V. John, "Reference values for drag and lift of a two dimensional time-dependent flow around a cylinder," *Int. J. Numer. Meth. Fluids*, vol. 44, pp. 777-788, 2004.
- [81] E. Bayraktar, O. Mierka and S. Turek, "Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FeatFlow," *Int. J. of Computational Science and Engineering*, vol. 7, pp. 253-266, 2012.
- [82] M. Schäfer, S. Turek, F. Durst, E. Krause and R. Rannacher, "Benchmark computations of laminar flow around a cylinder," *Vieweg+ Teubner Verlag*, pp. 547-566, 1996.
- [83] U. Ghia, K. N. Ghia and C. Shin, "High-Re solutions for incompressible flow using the navier stokes equations and a multigrid method," *Journal of Computational physics*, vol. 48, pp. 387-411, 1982.
- [84] E. Erturk and T. Corke, "Numerical solutions of 2D steady incompressible driven cavity flow at high reynolds numbers," *Int. J. Numer. Meth. Fluids 2005*, vol. 48, pp. 747-774, 2005.
- [85] K. Poochinapan, "Numerical Implementations for 2D Lid-Driven Cavity Flow in Stream Function Formulation," *International Scholarly Research Network*, no. doi:10.5402/2012/871538, pp. 17-35, 2012.
- [86] J. P. Gray and J. J. Monaghan, "SPH elastic dynamics," *Comput. Methods Appl. Mech. Engrg.*, vol. 190, pp. 6641-6662, 2001.
- [87] H. Takeda, S. M. Miyama and M. Sekiya, "Numerical simulation of viscous flow by smoothed particle hydrodynamics," *Progress of Theoretical Physics*, vol. 92, pp. 939-960, 1994.
- [88] J. J. Monaghan and A. Kocharyan, "SPH simulation of multi-phase flow," *Computer Physics Communications*, vol. 87, pp. 225-235, 1995.
- [89] M. V. Dyke, *An album of fluid motion*, The Parabolic Press, 1988.
- [90] D. J. Tritton, "Experiments on the flow past a circular cylinder at low Reynolds numbers," *J. Fluid Mech.*, vol. 6, pp. 547-567, 1959.

- [91] D. W. Holmes, J. R. Williams and P. Tilke, "Smooth particle hydrodynamics simulations of low Reynolds number flows through porous media," *Int. J. Numer. Anal. Methods Geomech.*, vol. 35, pp. 419-437, 2011.
- [92] A. Roulund, M. B. Sumer and J. Fredse, "Numerical and experimental investigation of flow and scour around a circle pile," *J. Fluid. Mech.*, vol. 534, pp. 351-401, 2005.
- [93] D. J. Price, "Resolving high Reynolds numbers in smoothed particle hydrodynamics simulations of subsonic turbulence," *Monthly Notices of the Royal Astronomical Society*, vol. 420, pp. 375-408, 2012.
- [94] S. Marrone, A. Colagrossi, M. Antuono, G. Colicchio and G. Graziani, "An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers," *Journal of Computational Physics*, vol. 245, pp. 456-475, 2013.
- [95] T. V. Kármán, *Aerodynamics*, McGraw-Hill, 1963.