Title	コンピュータグラフィックスにおける媒体中の剛体運動シミュレーション
Author(s)	謝，浩然
Citation	
Issue Date	2015-03
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/12755
Rights	
Description	Supervisor:宮田一乗，知識科学研究科，博士

Doctoral Dissertation

Immersed Rigid Body Dynamics in Computer Graphics

Haoran Xie

Supervisor: Professor Kazunori Miyata

School of Knowledge Science
Japan Advanced Institute of Science and Technology

March 2015

Keywords: real-time, immersed rigid body, free fall, phased diagram, turbulent flow, data-driven simulation, two-way coupling, aerodynamics, unsteady dynamics, model reduction, motion synthesis, motion patterns, precomputation, computer graphics

Be not immortal, since it is flame.
Be infinite, while it lasts.
— Sonnet of Fidelity, Vinicius de Moraes

Abstract

The real world is complex and specular: a leaf falling down from a tree side byside, a coin moving underwater swaying left and right, and the falling snowflakes dancing up and down in even still flow environments. Unfortunately, the virtual worlds under the conventional animation techniques utilize the ideal models with no consideration of the detailed effects of the flow environments, which take account of the inertial, viscous, and turbulent features in high Reynolds number flow. Although the physical simulations have dramatic success in 3D films, games and virtual reality applications in recent decades, the simulations of unsteady and turbulent dynamics frustrate researchers in both computation cost and simulation fidelity.

To solve these issues, this dissertation proposes a new topic, *immersed rigid body dynamics*, into the real-time computer graphics community. It is clearly different from the other traditional topics in computer graphics that the research aim of immersed rigid body dynamics is to simulate the motion of rigid body fully immersed or submerged inside real flows, and strongly coupled with the surrounding flows. This dissertation presents a family of algorithms for real-time simulations of immersed rigid body dynamics in computer animation. These algorithms are built on data-driven simulation methods to simulate the rigid body dynamics with the flow effects in computer environment. These approaches make it feasible to achieve realistic simulation results in low computation cost. In addition, a promising prior reduced model of dynamical systems is introduced for the parameter identification into computer animation.

The first contribution is a graph-based framework for synthesizing the motions of immersed rigid body, which are commonly lightweight. This framework is a first try to combine the motion graph technique in character animation field with the physics-based simulations. The typical motion patterns of immersed rigid-body dynamics are extracted in a phase diagram and verified from thousands of physical experiments to construct a precomputed trajectory database and the transition probabilities in Markov-chain model of the motion graph. Finally, an improved noise-based algorithm is proposed for integrating the wind field with the simulation results.

The second contribution is a stochastic model of immersed rigid body dynamics. This model first utilizes energy transport model of the surrounding turbulent flow to approximate the energy distributions due to the rigid body motions. Then, the proposed turbulent model is successfully introduced into a generalized Kirchhoff representation of the rigid body dynamics with Langevin model in a stochastic Wiener process. The proposed model adopts a new approach combining the precomputed simulation data of turbulent energy and the runtime simulations of rigid body solvers.

The third contribution focuses on a pattern-driven framework for immersed rigid body dynamics. This simulation framework first classifies the influences of parameter spaces of viscous force coefficients in a data training process, and then proposes a curvature-based motion planning method to represent the unsteady dynamics due to the vortex shedding. The proposed methods learns the knowledge of parameter subspaces of the rigid body dynamics from numerical experiments in a new dynamical model, which clarifies the viscous forces from the surrounding flow into three components and reveals the relationships among the force coefficients, the Reynolds number, and angle of attack of the body. In addition, the proposed framework combines the motion graph technique from the graph-based framework and the energy optimization in motion synthesis from the stochastic model.

Finally, a novel reduced model of dynamical systems is constructed, which can accelerate the parameter estimation of physical parameters in a dynamical model with low computation cost. In contrast to the conventional reduced order models, the proposed model is a prior meta-model of dynamical systems based on the separated representation in large domains including initial conditions, boundary conditions, and physical parameters. The proposed model does not depend on the preprocessed snapshots of the solutions from dynamics solvers. This model is successfully applied to the weakly coupled and nonlinear problems. The improvement of the proposed reduced model in strongly nonlinear problem, such as immersed rigid body dynamics, is worth being anticipated.

Acknowledgments

As I am finally summarizing this dissertation, I would like to thank and acknowledge many people for their support, help and encouragement over the years enrolling in the five-year-program of JAIST, Ishikawa, Japan, which is the most wonderful institute hovering in the blue sky over the mountains of imagination and innovation. First and foremost, my sincere thanks to my advisor, Prof. Kazunori Miyata for his vision and inspiration to visual simulations in computer graphics, for his guidance and commitment to pursue researches on persistence and think topics profoundly, and for his generosity of various research opportunities and encouragement to the qualified thinking of the challenging simulation problems.

During my graduate studies, it is great luck to have the chances to meet and discuss with excellent researchers in different research fields, including but not limited to computational fluid dynamics, computer vision, software engineering, complexity science, and computer graphics. Thanks to my great advisor and the sophisticated education system, I have been enjoyed three internship opportunities in the University of Sydney (2011), University of California (2012) and Kent state university (2013). I would like to thank all the friends among the internships who help me accommodate the cheerful life overseas.

I am also grateful to the members of my thesis committee: Prof. Kazunori Miyata, Prof. Taketoshi Yoshida, Prof. Huynh Nam Van, Prof. Ho Bao Tu in JAIST and Prof. Shigeo Morishima from Waseda university. Their professional suggestions and feedbacks help me to improve this dissertation greatly. I would also thank Dr. Ye Zhao, the advisor in Kent state university for conducting my minor research, and Prof. Takashi Hashimoto for his great inspiration on research problems, and Prof. Norishige Chiba for his constructive discussion on my research topic in the short visit to the computer graphics lab of Iwate university.

In addition, I would like to thank all the members in Miyata laboratory, especially, Mr. Xiaodong Du for accommodating my early life in Japan, and Dr. Ken Ishibashi for sharing his great research experiences and his help on the application of research fellow of Japan Society for the Promotion of Science, and Dr. Thao Ngoc Nguyen for her helpful discussions on computer vision problems.

Finally, I have been deeply indebted to my parents and elder brother for their unconditional support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since Dr. Ivan Sutherland invented the computer drawing system Sketchpad in 1962, computer graphics has been improved profoundly into various topics, including computational geometry and modelling, shading and rendering, animations and simulations. When time comes to the year of 1999, there are two important events playing significant roles in physical simulations to twenty-first century: NVIDIA created the world first graphics processing unit (GPU), and Jos Stam has published his famous paper "stable fluids", which has attracted around 1,500 citations until now. The traditional physical simulation in computer graphics includes the rigid body simulations, deformable simulations, and fluid simulations. The developments of GPU and semi-Lagrangian algorithm has ignited the enthusiasm of fluid simulations and the topics in other physical simulations. The researcher started to try more challenging topics, such as two-way coupling among fluids and bodies, turbulent flow, multiphase flow, and so on. The successful computer graphics techniques make the 3D films and games become popular to attract people worldwide. The applications of virtual reality make users immersed in a virtual world that has never happened before.

Even with the help of GPU computations, the simulation work always pursues a better trade-off between computation cost and simulation fidelity. The real-time simulations are urged in various applications, such as online games, mobile games, surgical simulations, and animation design systems, where an interactive system allows the computer to provide instantaneous feedback to the user's operations. Note that the physical simulations usually refer the simulations of physics-based dynamical systems, such as the visual simulations of natural phenomena and different bodies, e.g., rigid and soft bodies and particles [Par12]. There is another important field in computer animation, character animation, which covers various topics as human motion, facial animation, hand animation, and so on. Although it seems that the techniques of physical simulations and character animations are quite different, this thesis has a successful try to introduce the character animation technique into physical simulations with motion styles by a motion graph. The pursuit of both realistic and real-time simulations facilitates researchers to develop new algorithms, so that the state-of-art physical simulations in computer graphics have two main categories: physics-based methods and data-driven methods. All the proposed approaches in this thesis belong to data-driven simulations to achieve realistic simulations in low computation cost. The data-driven methods are specially useful to simulate complex dynamics in two ways: capturing the complex behaviours using data model and precomputating the heavy simulations in offline processes.

The physical simulations of the complex dynamical systems are greatly challenging and promising to enhance the realness of computer animations that cannot be achieved before. The complex systems arise from high degrees of freedom (DOFs) as particles and articulated bodies, high Reynolds numbers as turbulent flows, and nonsmooth dynamics as frictional contacts of cloths and rigid bodies in computer graphics. The main purpose of this thesis is to simulate the complex systems due to unsteady dynamics with high Reynolds numbers flow as turbulent flows, and to introduce a novel topic, immersed rigid body dynamics, into computer graphics.

1.1 A novel topic

The natural phenomena of the immersed bodies around us are prevalent in daily-life. The dynamics of an immersed body means that the motion of a body moving inside the flows, immersed in air or submerged underwater, is very sensitive affected by the surrounding flow. Precisely speaking, the immersed body undergoes the wake-induced path-instability in a strongly coupled process of vortex-structure coupling within high Reynolds number. As shown in Figure 1.1, these phenomena cover the motions of falling card, falling leaf, rising bubble, falling object underwater, swaying cloth, flying paper-airplane, snowflakes, dust, the swimming motions, and so on. In this thesis, we focuses on the fundamental immersed rigid body dynamics, and the simulation techniques of immersed rigid body can be extended to linked rigid bodies, spring-mass cloth, and particle systems straightforwardly. In contrast to the previous work in computer animation, the simulation of immersed rigid body dynamics is indicated as a novel topic in the following reasons:

- **Rigid body simulations:** The conventional rigid body solvers do not consider any flow effect from the surrounding flow that we can notice that a falling leaf would move downward vertically by current physics engines. The rigid body follows the classic Newton's law like in a vacuum environment, and it is obviously inconsistent with our daily experiences,
- **Two-way coupling:** The coupling motions among body and fluids have been studied extensively in computer graphics, there are two essential differences hindering

Figure 1.1: Examples of natural phenomena of the immersed bodies. Sources: Google image search.

the two-way coupling techniques to simulate the immersed rigid body successfully. First, they have different physics principles, the coupling motions are usually in low Reynolds number where the turbulent effect is out of account. Then, they have different research aims: the coupling simulations mainly focus on the fluid motions, such as splashing; while the simulations of immersed rigid body focus on the dynamical states of the rigid body. Due to the immersed nature, we cannot perceive the apparent movements of the surrounding air and water so that it is inadvisable to simulate the particles' motions for graphical applications, which are commonly in high computational costs.

- **Aerodynamics simulations.** The aerodynamics/hydrodynamics simulations usually utilize the approximations of drag and lift forces based on a quasi-steady assumption, such as the quadratic viscous forces from Kutta-Joukowski theory. In such approximations, the forces' coefficients play a significant role in the whole process of the aerodynamics simulations, which are usually designated to be as constants, functions of angle of attack or the Reynolds number. In the situation of an immersed rigid body, it is known that the force coefficients are instantaneously changed as a complex and unknown function of angle of attack, the Reynolds of number, the body geometry and the vortex-shedding periods. In this sense, the approximations of viscous forces are not sufficient to achieve realistic simulation results for the proposed topic.

- **Multiphase flow:** In computer animation, the multiphase flow simulations involve the simulations of bubble flow, dust simulations, snowfall simulations, leaves simulation, and so on. The participated bodies have different physical properties (e.g., density) from the surrounding flow particles. In these simulations, the immersed bodies are considered as sphere particles with 3 DOFs, where the non-spherical particle motions is still an absent issue which presents the same motion patterns with the proposed topic.

This novel topic is also related to the turbulent/fluid simulations and character animations. More details about the related previous work are described in next chapter. Finally, the techniques for simulating the immersed rigid body dynamics in computer graphics have to overcome the following issues: (1) The computation of coupling motions with fluid in small grids is too heavy for real-time graphical applications, (2) While simulating flow in high resolutions, the turbulent motions and their numerical dissipations are difficult to be analysed; (3) In order to achieve stable simulation results, the implementation involving boundary conditions requires infinitesimal timesteps; (4) The coupling problem among the translational and rotational velocities of the rigid body exists due to six DOFs states. Therefore, the simulation of immersed rigid body dynamics is a challenging topic in both fluid mechanics and computer graphics.

1.2 A challenging topic

This section explains why the proposed topic is considered to be challenging from the view of fluid mechanics.

- **Strongly coupling:** Coupling between a rigid body and the surrounding flow is an important subject in computer animation based on various disciplines of engineering and physical problems. Nevertheless, the community lacks of the computation technique that can simulate a strongly coupling problem of the moving boundary with unsteady flow due to its computationally challenging and expensive issues. In contrast to the enormous literature of the rigid body dynamics, particle dynamics and two-way coupling simulations, the strongly coupled fluid-body interactions involve the vortex shedding and flow instability. To make it clear, T_f and T_r present the characteristic timescales of fluid motion and rigid body motion, respectively. If $T_f \gg T_r$, then the rate of change of the surrounding flow can be neglected in comparison with the rate of change of the body, for example, a cup falling onto ground. If $T_f \ll T_r$, instead of solving a coupling dynamics, the quasi-steady approximations of drag and lift forces using force coefficients are adequate, which are the simplified representation of aerodynamics widely adopted in wind simulations. For a strongly coupled interaction, $T_f \approx T_r$, the steady dynamics based on quasi-steady approximations becomes invalid anymore. The numerical difficulty arises from the nonlinearity and unsteadiness of the coupled motions due to the generation and detachment of vortices from body's edge.

- **Wake instability:** The Reynolds number is a critical parameter to present the flow patterns with different dynamical similarities, and is a measurement of the relative importance of inertial and viscous effects from the surrounding flow. $Re = UL/\nu$, where U is the mean relative velocity of the body to the flow, L is the characteristic dimension of the rigid body, and ν is the kinematic viscosity of the flow. For low Reynolds number flow, the inertia of the flow is not important and the flow is smooth and straight forward; For higher Reynolds number, inertia begins to play an important role and vortices are generated behind the object. When the Reynolds number increases, the flow becomes unsteady from the steady state and undergoes several bifurcations. The first bifurcation leads the steady flow with low Reynolds number lose its axisymmetry at a critical number Re_{c1} ($Re_{c1} = 212$ for sphere [ERFM12], $Re_{c1} = 105$ for disk [ZLS+13]). The flow motion bifurcates into two branches in planar plane of the body center line. The second bifurcation occurs at Re_{c2} ($Re_{c2} = 273$ for sphere, $Re_{c2} = 160$ for disk) where the periodic hairpin-like vortices are shed from the symmetry plane of the body. The wake structures have the alternating sign and different magnitudes. And then, the third bifurcation occurs at Re_{c3} ($Re_{c3} = 355$ for sphere, $Re_{c3} = 200$ for disk) where the wake becomes irregular and chaotic, and the vortex shedding makes the motion be fully in three dimensional state. As the Reynolds number increases further, the secondary vortices and the counter-rotating vortex pair are observed at the leading-edge of the rigid body.
- **Path instability:** The effects of wake instability are related to the oscillations of the immersed body, which invoke the motion transitions among different motion patterns of the immersed rigid body. Besides of the viscous effects with the Reynolds number, the inertial effects due to the density ratio and the body geometry (aspect ratio) also play a important role for the three dimensional motion trajectories. For low Reynolds number flow, the body generates small horizontal oscillations due to the vortex shedding as Karman vortex street. While the Reynolds number increases, large amplitude oscillations were found [HW10]. The observed path could change largely with the environment noise at higher Reynolds number. When the viscous effects become predominant, the body moves side-by-side along a rectilinear path; when the inertial effects become predominant, the body falls in a planar plane. As density-ratio

increases, a zigzagging motion starts and then transfers to an autorotation motion. There are also some three dimensional motion paths observed, such as spiral and helical motions [ZCL11].

1.3 Global overview

This section describes the functional modules of this thesis in a global overview, which are adopted in different combinations for the proposed methods to simulate immersed rigid body dynamics. To clarify the functions of each module, we define the input and output of each module to construct data-driven methods.

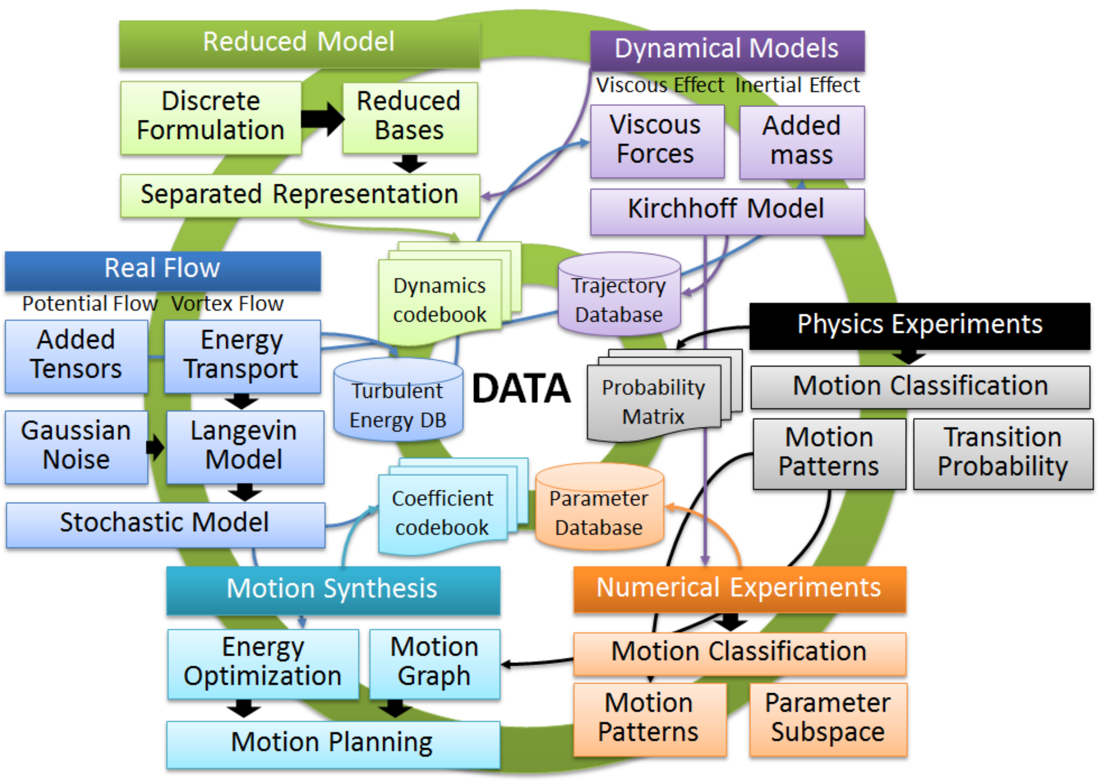


Figure 1.2: Functional modules defined in this thesis.

1.3.1 Functional modules

As illustrated in Figure 1.2, six functional modules are defined as follows.

Dynamical models provide the kinematic and dynamical equations of the states of the immersed rigid body. The states include the information about position, orientation, translational and angular velocities of the rigid body. The dynamical equations are commonly the ordinary differential equations which can be solved by traditional Runge-Kutta (RK) method. In contrast to RK method, the variational geometric integrator based on SE(3) [KCD09] is preferred for achieve stable numerical results. The dynamical model is designed in a generalized Kirchhoff model considering both viscous effect and inertial effect of the surrounding flow. For the purpose of motion synthesis, a dynamical model could create various motion segments to be stored in a trajectory database.

Physics experiments capture the motions of rigid bodies by high-speed cameras in the experiments of falling objects with different geometries and densities in air or water. In contrast to the motion capture of the body's state at each frame, this thesis proposes a high-level capturing system of motion patterns. In terms of the captured motions, the transition matrix among motion patterns can be obtained as discrete Markov-chain model for a motion graph in motion synthesis.

Numerical experiments obtain the precomputed simulation data by tuning control parameters. Although the control parameters are defined as drag and lift forces coefficients in this thesis, the other parameters can be added properly according to the simulation purposes, such as aspect-ratio, density, and release angle of the rigid body. All the motion trajectories are classified into different motion patterns as observed in the physical experiments. Finally, the database of parameter subspaces are constructed corresponding to each motion pattern.

Motion synthesis provides a motion planning process to connect the data of different motion patterns by a motion graph. This process can be executed in runtime for the purpose of online simulations or in precomputation for storing the instantaneous force coefficients as codebook. In addition, an energy optimization is proposed to reflect the turbulent features of energy dissipation in the surrounding flow.

Real flow defines the flow effects from the surrounding flow by considering both the potential flow and the vortex flow. For potential flow, a panel method or boundary element method can be utilized to obtain the added mass tensors. For vortex flow, there are various turbulent models, such as energy transport model, Reynolds averaged Navier-Stokes simulation, and large eddy simulation. In this thesis, the energy transport model is used by combining with a Langevin model. The Langevin model is a stochastic process with randomness sources including white noise or colored noises. To reduce the runtime computation cost, the turbulent model can be simulated offline with the turbulent energy database as output.

Reduced model represents the coherent features of a dynamical system. In this thesis, a prior reduced model of separated representation is proposed that does not depend on the precomputed simulation data. The computation process of reduced model can be executed in pre-computation steps.

1.3.2 Data-driven methods

All the proposed methods are data-driven simulation methods in this dissertation to achieve real time simulations. As shown in Table 1.1, the distinctions among the proposed methods are designed as follows:

Table 1.1: Combinations of different modules in all proposed methods.

Modules	Graph-based	Stochastic model	Pattern Driven	Reduced model
Dynamical models	√	√	√	√
Physics experiments	√		√	
Numerical experiments			√	
Motion synthesis	√		√	
Real flow		√	√	
Reduced model				√

Graph-based method utilizes a simplified *dynamical model* of immersed rigid dynamics to model each motion pattern from *physical experiments*, and synthesize the rigid body motions from the precomputed trajectory database in *motion synthesis*.

Stochastic model combines the *dynamical model* of immersed rigid body dynamics with a Langevin model in *real flow* using the precomputed turbulent energy database.

Pattern Driven method adopts a motion graph with the data of motion patterns from *physical experiments*, and an energy optimization in parameter subspaces from *numerical experiments* of a proposed *dynamical model*, which is combined with the turbulent energy from the turbulent model of *real flow*. A coefficients codebook is obtained in *motion synthesis* for the runtime simulations.

Reduced model proposes a *reduced model* of dynamical systems with the *dynamical model* in a formulation of ordinary or partial differential equations.

1.4 Contributions and outline

As illustrated in Figure 1.3, the contributions of this thesis stand in a family of novel algorithms for realistic and real-time simulations of immersed rigid body dynamics: graph-based method, stochastic model, and pattern-driven method. Note that a new reduced model is also proposed for all dynamical systems. This model can solve the weakly nonlinear and coupled systems currently, and can be improved in the simulations of strongly nonlinear dynamical systems in the proposed topic. The previous visual simulations in computer graphics utilized simplified approaches to handle viscous forces in steady or quasi-steady force approximations, such as the listed methods in the left-top of Figure 1.3. Also, two-way coupling techniques can handle weakly fluid-body coupling motions, where the timescale of the body dynamics is smaller than the timescale of the flow dynamics. It is difficult to achieve a realistic simulation by these approaches. The strongly coupling dynamical system involving unsteady aerodynamic forces can be resolved in high-Reynolds number and nonlinear problem in fluid mechanics by the proposed approaches, such as immersed boundary methods, vortex method, and spectral element method. All these approaches can achieve realistic simulation results but lose the efficiency as shown in bottom-down of Figure 1.3. This dissertation proposes new approaches with different ways to handle data information which are suitable for real-time applications.

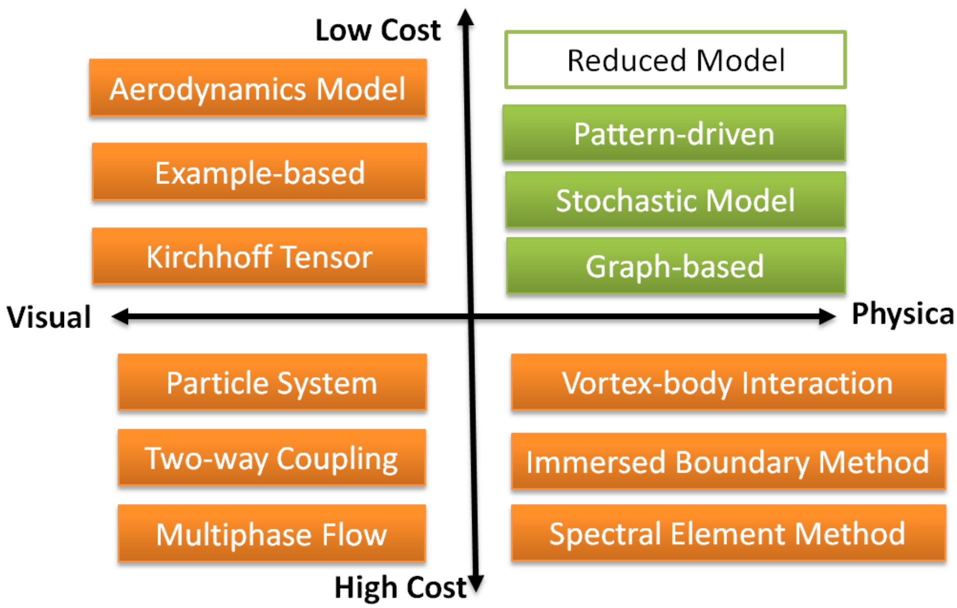


Figure 1.3: A family of algorithms proposed in this thesis for simulating immersed rigid body dynamics (Green color: the proposed methods; orange color: examples of the related methods). The abscissa denotes the simulation quality, and the ordinate represents the computation cost. The details of the related work are described in the next chapter.

The subsequent chapters are organized as follows:

Chapter 2 overviews the research background of immersed rigid body dynamics. This chapter first introduces the history of the proposed topic in physics research areas. Then, the recent related work from both computer graphics and fluid mechanics is summarized. Finally, the fundamental knowledge of the dynamical models and the representation of viscous forces are discussed.

Chapter 3 starts to present a motion synthesis method for simulating immersed rigid body dynamics, which is a graph-based framework with a proposed motion graph technique to capture the transitions among different motion patterns [XM11, XM12]. This work constructs a trajectory database for each motion segment from a dynamical model. Finally, the wind effect is added into this framework to create the realistic simulation results under different wind environments [XM14c].

Chapter 4 describes the second proposed method in the family of data-driven algorithms. A stochastic model is proposed to account for the fluid effects from the turbulent energy model [XM14a]. This work presents a Langevin approach to represent the velocity increments from a turbulent model in Wiener process. This work proposes a fractional-step algorithm to take both the inertial and viscous effects from the surrounding flow in account [XM13].

Chapter 5 switches the third proposed method for immersed rigid body dynamics. A pattern-driven method is proposed to estimate the force coefficients by considering the inertial, viscous and turbulent effects from the surround flow. A data training process is proposed to find out the parameter subspaces from numerical experiments. Four motion patterns are observed from the three dimensional numerical experiments, which are the utilized in a motion graph with the motion synthesis step to simulate various motion transitions. This work proposes an energy optimization method based on a turbulent model. Finally, the precomputed coefficient codebook is used in the runtime simulations of the proposed dynamical model to achieve real-time simulations.

Chapter 6 proposes a prior model reduction technique for dynamical systems [XM14b]. The reduced model utilizes separated representation of dynamical systems, which is a meta-model based on different variable domains including temporal and spacial domains, initial conditions, and physical parameters as extra-coordinates. In order to represent the dynamical model of immersed rigid body dynamics, this chapter offers a feasible approach to improve the algorithm for strong nonlinear dynamical system in future.

Chapter 7 presents the conclusions and the suggestions for future work. Furthermore, this chapter summarizes the contributions of the proposed topic to other research topics in computer graphics and knowledge science. The limitations of each proposed method are also analysed and evaluated in this chapter.

Chapter 2

Background

In this chapter, a review of the research development of immersed rigid body dynamics in physics is presented first. The freely falling motion is a foundational aspect of immersed dynamics where the body moves from rest under gravity. The other dynamical aspects, such as initial velocities, wind effects and body collisions, can be embedded into the dynamical system straightforwardly. Also, this chapter surveys the related simulation works from both computer graphics and fluid mechanics. Finally, the basic mathematical models of immersed rigid dynamics are analysed briefly.

2.1 A historical review

Hundreds years ago, Newton declaimed about immersed rigid body dynamics in the legendary book, Principia, as "the bladders did not always fall straight down, but sometimes flew about and oscillated to and fro while falling. And the times of falling were prolonged and increased by these motions, sometimes by one-half of one second, sometimes by a whole second." ([New87], page 759, 1687). Maxwell described the dynamics as "its motion, although undecided and wavering first, sometimes becomes regular"([Max90], page 115, 1853). Besides the qualitative descriptions of the phenomena, the quantitative scientific researches starts from the last century after the development of aerodynamics and hydrodynamics. The issue of immersed rigid body dynamics is relevant to different scientific and engineering problems, including meteorology, sedimentology, aerospace engineering, biological sciences, and chemistry problems. There are two booms of this research: U.S. military funded this study for military usage in 1960s and the research interest came back inspired by chaos theory in 1990s [Wei98]. Currently it may be the third booms of the research of the topic due to the growing computation power and the generation of data science. We can also infer these trends from the published papers listed in Table 2.1 and 2.2.

As shown in Table 2.1, scientist did numerous physical experiments of falling rigid body in fluid, and found out the hidden rules as motion patterns inside the daily-life phenomena. As explained previously, the unsteady dynamics in high Reynolds number flow make the motions seem to be unpredictable. [WHH64] firstly provided a phase diagram based on the Reynolds number Re and the dimensionless moment of inertia I^*. The stable and unstable oscillations of the body are determined by these two parameters. The tumbling motion happens at both larger Re and I^*. [SDG69] found the effects of the shape of various bodies and Re to the dynamics, and declaimed that the motion patterns of a disk are more unstable than the patterns of a three-dimensional sphere. [Smi71] measured the phase diagram for rectangular plates which is similar to the work [WHH64]. [FKMN97] started the further experiments on falling disks, and the apparent chaotic motion is proven to exist. Then, [BEM98] also observed the fluttering and tumbling motions, they discovered the fluttering dynamic of a falling object and the transition from fluttering to tumbling motions occurs at a special dimensionless quantity: the Froude number. [MRS99] studied the dependence of the angular velocity on the width of a tumbling card.

Recent experimental works attempts to uncover the motion patterns in three dimensions and the motion transitions among motion patterns. The experiments in [ZCL11, ZLS$^+$13] have found an additional three typical trajectories in a three-dimensional environment: zigzag, transitional helix, and spiral motions. The exhaust experimental work [Raz10] concerned about the transitions among different motion pattens of leaves based on more than six thousands three-dimensional experiments. In the recent work [VCW12] and [VCW13], the similar patterns of helical motions are observed for different shapes of rigid bodies. All the patterns and the experimental environments of previous work are listed in Table 2.1.

Table 2.1: Summary of previous investigations on immersed rigid body dynamics from physical experiments in the order of published years.

References	Rigid Body	Fluid	Re	Motion Patterns
[Ria35]	rectangular card	air		fluttering/tumbling
[WHH64]	circular disk	water/glycol	$10^2 \sim 10^3$	fluttering/descent/chaotic
[SDG69]	various bodies	water/glycerol	$10^2 \sim 10^5$	fluttering/ tumbling
[Smi71]	rectangular plate	air	$10^2 \sim 10^4$	tumbling
[FKMN97]	circular disk	water/glycol	$10^2 \sim 10^4$	fluttering/tumbling/chaotic
[BEM98]	thin strip	water/glycerol	$10^3 \sim 10^4$	fluttering/ tumbling
[MRS99]	rectangular card	air		tumbling
[APW05b]	rectangular plate	water	$10^2 \sim 10^3$	fluttering/tumbling/chaotic
[ZCL11]	circular disk	water	$10^2 \sim 10^4$	fluttering/helical/spiral
[VCW12]	maple seeds	air	10^3	helical
[WHXW13]	rectangular card	air	5×10^3	tumbling
[VCW13]	parallelograms card	air	10^3	tumbling/helical

2.2 Related work in physical simulations

In this section, the related works of physical simulations are presented from the research field of both computer graphics and fluid mechanics related to immersed rigid body dynamics. This thesis proposes a family of data-driven methods considering the real flow effects which the previous computer graphics work. The numerical methods in fluid mechanics are mainly in 2D case as shown in Table 2.2. Recent 3D physical simulations have the limitations to account for turbulent flow and the high computational cost, which are not feasible to satisfy the realistic and real-time requirements in this study.

2.2.1 Computer Graphics

Rigid body dynamics has a long history in computer graphics and is a significant starting points of other character and deformable body simulations [Bar93]. A recent work [BETC12] detailed the modern development of mechanics, complementarity problems, numerical methods in interactive rigid body simulations. The traditional rigid body solvers do not consider the influences from the surrounding flow, where any rigid body always falls down vertically.

Two-way Coupling simulations between rigid body and incompressible fluid has been studied extensively in computer graphics. Basically there are two types of schemes on this research. The first scheme handles fluid in Euler formulation and rigid bodies in Lagrangian formation [CMT04, GSLF05, BBB07, RMSG+08, CM10]. Guendelman et al. [GSLF05] proposed a robust ray casting algorithm for the coupling between fluid and cloths to avoid fluid leaking. Carlson et al. [CMT04] treated the rigid body as fluid grid by using distributed Langrange multiplier. The second scheme is the fully Langrangian meshless method [BTT09, CBP05, SSP07, HLW+12]. Becker et al. [BTT09] proposed a direct forcing method in a predictor-corrector scheme with SPH particles. Solenthaler et al. [SSP07] used a penalty method to analyze the forces on the immersed boundary. These two-way coupling approaches provide great simulation results for weakly coupling problems in low-Re conditions, where the rigid body does not exhibit chaotic behaviours. For the research purposes of immersed rigid-body dynamics in this thesis, it is trivial and infeasible to simulate high-Re two-way coupling with turbulent flow in computer graphics as explained in Chapter 1. It is too computationally heavy for immersed rigid body simulations where the motions of fluids are not visible.

Aerodynamics simulations are widely proposed in computer animation. [WZF+04] proposed Lattice-Boltzmann method for solving fluid simulation and Kutta-Joukowski theorem for body's dynamics, such as soap bubbles and feathers. This proposed method cannot produce a designated motion trajectory, and it has difficulty in simulating multiple objects, because this method requires a large computational cost (a single bubble: CPU 2.8 fps and GPU 11.5 fps; a single feather: CPU 0.76 fps and GPU 6.1 fps). The commercial CG tools, including LightWave^{TM} and Maya$^{@}$ nCloth^{TM}, do not embody the functions of the animation for immersed rigid body, instead, they provide particle systems to model an immersed body by adjusting the drag and lift parameters in a wind field. In all of these works, the motion paths are unpredictable, and it is infeasible to achieve realistic motion.

Unsteady dynamics The underwater simulation work [WP12] introduced a Kirchhoff tensor to represent inertial effects for underwater rigid body simulations. This approach is suitable for the inviscid and irrotational flow with low-Re number. [OKRC10] presented a fractional derivatives method for representing historic force of underwater cloth in low Reynolds number flow. This work proposes a Langevin model related to the turbulent flow for solving the vortical loads. Langevin model has been applied to enhance turbulent flow simulations [CZY11] and simulations of floating lightweight rigid body [YCZ11] in previous work. In these work, the rotational velocity and the coupling between translational and rotational velocities are not concerned. We resolve these issues by combing generalized Kirchhoff equations with Langevin model in this paper. There are also some interesting works about motions of snowflakes and dusts, such as particle system [CFW99, TLP06], spectral-particle method [LZK^{+}04]. All these approaches do not take into account the unsteady dynamics of the body, both inertial and viscous effects from the surrounding flow and the influences from the generated turbulences at the body's boundary layer.

Turbulent Flow simulations are different with direct numerical simulation of Navier-Stokes equations. First, from the view of fluid mechanics, there are some sophisticate approaches in this fields, including turbulent-viscosity models (k-ε equations), Reynolds-stress models, Probability Density Function methods (Langevin model) and large-eddy simulation. It is not apparent to adopt these approaches directly in computer animation, and there are some successful works [PTC^{+}10, PTSG09] in computer graphics community recently. Note that Langevin model is an empirical model based on k-ε equations [Pop83, Pop11] but an effective Langragian-stochastic approach to represent the dynamics of passive particles in turbulent flow [MD04]. Recent work shows that non-spherical particles moving in turbulent flow [MR10] exhibit the similar dynamics of immersed rigid bodies which has been discussed in previous work [XM13]. Therefore, it is physically reasonable to adopt Langevin model for simulating immersed rigid bodies in this work.

Data-driven methods have been proposed based on the Markov model [RP01], captured videos [AHN04], segments from fluid simulations [SYWC05], trajectories animated by Maya [VB08], and sketches by a designer [HQHQ10] to simulate the motions of falling leaves. All these simulations ignore the nature of motion of immersed rigid bodies and consider it a completely complex and unpredictable dynamic motion, which is modeled by stochastic processes or a simple particle representation.

Character animation method proposed motion synthesis combining the controllability of procedural and physically-based animation with the realistic appearance of a pre-recorded motion stream (e.g. motion capture). The motion graph [AF02] can automatically organize example motion clips into graphs for efficient motion synthesis. Later, Kovar et al. built an extended motion graph using local search with a branch and bound algorithm [KGP02]. Besides being used in human motion, motion graphs are also used in other physical simulations, such as tree animation [HFR06, ZZJ$^+$07]. This work includes the motion graph technique for synthesizing the motion of immersed rigid body.

2.2.2 Fluid mechanics

The developments of numerical simulations of immersed rigid body dynamics are listed in Table 2.2. The numerical work [TK94, APW05b, Umb05, PM11, ERFM12] corrected unsteady approximations of drag and lift forces. Tanabe et al. [TK94] built a simple phenomenological model of falling paper by solving ordinary differential equations (ODEs) based on the Kutta-Joukowski theorem. Andersen et al. [PW04] provided a solution of the 2D Navier-Stokes equations for the flow around the tumbling plate, which are solved in the formulation of vorticity stream function within a body-fixed elliptical coordinate system. This method utilized a conformal mapping to avoid singularities. [APW05b] conducted various numerical simulation in air, and discussed the motion transition between tumbling and fluttering motion patterns. This work observed the apparently chaotic motions are due to the high sensitivity of the dynamical system to experimental noise. [JX08] attempted to overcome the various discrepancies between experimental and numerical solutions encountered in [APW05b]. [KS10] developed a Fourier pseudo-spectral method to solve the 2D Navier-Stokes equations coupled with equations which govern free fall motion of a object, and the simulation results varied depending on the Reynolds numbers. [YS12] presented a direct forcing immersed boundary method for strongly coupling problems, including vortex-induced vibrations of a circular cylinder, transverse and rotational galloping of rectangular bodies, and fluttering and tumbling of rectangular plates. Another approach for solving strongly coupling motions is a point vortex method [MLS09]. This method is based on Brown-Michael equation via Kutta conditions for 2D coupled motion of a sharp-edged rigid body and the surrounding inviscid flow.

Table 2.2: Summary of previous investigations on immersed rigid body dynamics from numerical experiments in the order of published years.

References	Dimensions	Rigid Body	Re	Motion Patterns
[TK94]	2D	rectangular paper		fluttering/tumbling
[PW04]	2D	paper	10^3	fluttering/tumbling
[APW05a]	2D	Rectangular plate	$400 \sim 600$	fluttering/tumbling
[JX08]	2D	plates	$838 \sim 1100$	fluttering/tumbling
[KS10]	2D	leaves	$10^3 \sim 10^4$	tumbling
[YS12]	2D	rectangular plate	10^3	fluttering/tumbling
[AMF13]	3D	circular disk	$10^2 \sim 10^4$	fluttering/tumbling/chaotic/spiral
[CBD13]	3D	circular disk	$10^3 \sim 10^4$	fluttering/tumbling/spiral
[HW14]	2D	rectangular plate	10^3	fluttering/tumbling/chaotic

Recent progress studied the effects of mass distribution [HLW+13], motion transition of motion patterns [HW14]. These works are mainly discover the cases of quasi-two-dimensional setups. The recent work about three-dimensional dynamical motion patterns of immersed rigid body dynamics based on solid-fluid interaction simulations [AMF13, CBD13]. This thesis combines the research results of both experimental and numerical works to seek the motion patterns and their motion transition for an individual body in three dimensions. [CBD13] investigated the motion transitions among motion patterns based on the Galileo number and the non-dimensionalized mass. The numerical simulations of solid-fluid interaction utilized spectral element method with domain decomposition. [AMF13] studied the influence of the body density and the thickness of disk to the motion transitions by simulating the coupled Navier-Stokes equations with generalized Kirchhoff equations of rigid body dynamics [MM02].

2.3 Basic physics of immersed rigid body dynamics

2.3.1 Rigid body equations

Without the consideration of fluid effects, the dynamics of 3D rigid body follows the Newton-Euler equations in body-fixed frame.

$$F = ma_{cm} - mg \tag{2.1}$$

$$\tau = I\dot{\omega} + \omega \times I\omega \tag{2.2}$$

where F and τ are the aerodynamics forces and torques, respectively. m and I is the mass and moment of inertia of the rigid body. ω denotes angular velocity. a_{cm} is the translational acceleration at center of mass in body-fixed frame and the value becomes $a_{cm} = \dot{v} + \omega \times v$ by coordinate transformation from the world-frame, v is the translational velocity of the body. Finally, the rigid body equations can be described as:

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = \begin{pmatrix} mE & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} + \begin{pmatrix} \omega \times mEv \\ \omega \times I\omega \end{pmatrix} \tag{2.3}$$

where E is identity matrix.

2.3.2 Coupling equations of fluid-body system

The Navier-Stoke Equations are transformed into the following formulations in the reference frame moving with the rigid body.

$$\frac{\partial u}{\partial t} + \omega \times u + \nabla \cdot (u(u - w)) = \frac{1}{\rho_f} \nabla p + \nu \nabla^2 u \tag{2.4}$$

$$\nabla \cdot u = 0 \tag{2.5}$$

where u and p are the velocity field in the flow. ρ_f and ν are the density and viscosity of the flow. $w = v + \omega \times r$ is the body velocity in world-frame, and r denotes the orthogonal coordinate in body-fixed frame. On the body boundary S, the flow velocity satisfies the no-slip boundary condition.

$$u(x) = w(x) \qquad \forall x \in S \tag{2.6}$$

Then, the aerodynamic force and torque on the body are given as follows:

$$F = \int_S (\sigma \cdot n) ds \tag{2.7}$$

$$\tau = \int_S x \times (\sigma \cdot n) ds \tag{2.8}$$

where $\sigma = -pE + \rho_f \nu (\nabla v + \nabla^T v)$ is the stress tensor of the flow, n is the local normal to the solid boundary.

2.3.3 Motions in potential flow

In the case of the aforementioned coupling equation (Equation 2.5) in zero viscosity, the coupling problem can be converted a flow potential ϕ that satisfies the following Laplace problem if the flow is irrotational from rest.

$$\nabla^2\phi(x) = 0 \qquad \forall x \in \mathbb{R}^3 \setminus B \tag{2.9}$$

$$\frac{\partial\phi(x)}{\partial n} = w(x)\cdot n \qquad \forall x \in S \tag{2.10}$$

$$\phi(x) = 0 \qquad \forall \|x\| \to \infty \tag{2.11}$$

Here B represents the domain of rigid body. Note that the non-slip boundary conditions in Equation 2.6 should be replaced by Neumann conditions in the inviscid limit.

In terms of the inviscid theory [Lam75], the dynamics of rigid body inside potential flow is given as Kirchhoff Equations:

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = \begin{pmatrix} mE + K11 & K12 \\ K21 & I + K22 \end{pmatrix}\begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} + \begin{pmatrix} \omega \times (mE + K11)v \\ \omega \times (I + K22)\omega + v \times (K11v) \end{pmatrix} \tag{2.12}$$

where K is a symmetric added-mass tensor, i.e., $K21 = K12^T$.

$$K = \begin{pmatrix} K11 & K12 \\ K21 & K22 \end{pmatrix} \qquad K_{ij} = \rho_f \int_S \phi_i \frac{\partial \phi_j}{\partial n} ds \tag{2.13}$$

More details about the added-mass tensor can be found in [New77].

2.3.4 Experimental force model in 2D real flow

The real flow is viscous and vorticity exists around the immersed rigid body. The existence of vorticity in the flow makes the immersed rigid body dynamics complicated to analyse. For 2D situations, an approximation of the aerodynamic forces and torques is proposed by [APW05b, APW05a, Umb05]. The 2D dynamical model is given as follows:

$$(m + m11)\dot{v}_x = F_x - F_L \sin\alpha - F_D \cos\alpha - m_b g sin\theta \tag{2.14}$$

$$(m + m22)\dot{v}_y = F_y + F_L \cos\alpha - F_D \sin\alpha - m_b g cos\theta \tag{2.15}$$

$$(I + I_a)\dot{\omega} = M_a - M_s \tag{2.16}$$

$$\dot{\theta} = \omega \tag{2.17}$$

where m_b is the buoyancy-corrected gravitational force. $m11, m22$ and I_a are the diagonal components in added-mass tensor. $F_x = (m + m22)\omega v_y, F_y = -(m + m11)\omega v_x$ and $M_a = (m11 - m22)v_x v_y$ are the forces and torque due to added mass. θ and α are the angle and the angle of attack of the rigid body. F_D is the drag force in quadratic formulation.

$$F_D = 0.5\rho_f L\|v\|v(C_D(0)\cos^2\alpha + C_D(\pi/2)\sin^2\alpha) \tag{2.18}$$

where L is the width of the body while considering the cross-sectional area of the body in 2D model. $C_D(\alpha)$ is the drag coefficient at α angle of attack. This coefficient relation is proven to be valid at intermediate Reynolds number. F_L is the lift force defined as:

$$F_L = 0.5\rho_f L v^{-1}(\|v\|C_T\sin 2\alpha + LC_R\omega) \tag{2.19}$$

where C_T and C_R are the lift coefficients for translational and rotational lift forces. In contrast to classical Kutta-Joukowski model, this model is valid at high angle of attack for stall motion. Finally, M_s is the dissipative torque from drag and lift forces, which can be defined as follows:

$$M_s = \rho_f L^4\pi(U\mu_1/L + \mu_2|\omega|)\omega \tag{2.20}$$

where U is the characteristic velocity scale of the rigid motion. μ_1 and μ_2 are constant parameters.

2.3.5 Vortex effects on forces in real flow

Equation 2.7 described the aerodynamic force from the stress tensor of the flow. Sir James Lighthill [Lig86] pointed out that, the force on the body may be divided into a *potential-flow force* that depends linearly on the body velocity and can be accurately calculated as described previously; and a *vortex-flow force* that varies nonlinearly and is related in a definite way to the vortex shedding and the convection of shed vorticity. Equation 2.7 can be redefined as follows:

$$F = \rho_f \int_S \phi n ds - \frac{1}{2}\rho_f \int_V x \times \omega_a dV \tag{2.21}$$

Here, the two terms on the right-hand-side of the equation are the contributions from potential-flow force and vortex-flow force, respectively. ω_a denotes the additional vorticity in fluid domain. In real flow, there must be a thin boundary layer at the body surface to meet the no-slip boundary conditions of velocity field, and also the extra field where a turbulent is generated, both of these two effects are included in the additional vorticity.

Chapter 3

Graph-based immersed rigid body dynamics

This chapter first analyses the physical characteristics of free fall motions in quiescent flow, and proposes a new procedural motion synthesis method for modelling immersed rigid body dynamics like freely fall motions in interactive environments. Six motion patterns of immersed rigid bodies are defined in phase diagram and analysed separately using a trajectory search tree and a precomputed trajectory database. The global paths of the immersed body motion are synthesized on the basis of these motion patterns, using a specified motion graph whose edges are connected in the Markov chain model. Then, the proposed approach integrates with external forces (e.g. wind field) by an improved noise-based algorithm under different force magnitudes and object release heights. This approach exhibits not only realistic simulation results in various environments but also fast computation to satisfy real-time requirements.

3.1 Introduction

Rigid body simulations are widely used in applications ranging from films to engineering and games. For lightweight objects, not all objects fall straight down, for examples, a piece of paper and a leaf waver and flutter down in a seeming unpredictable motion in the daily life. Unlike common rigid bodies, the immersed rigid bodies have special and spectacular motions known as free fall motion, such as fluttering (oscillate from side to side) and tumbling (rotate and drift sideways). However, in computer graphics, the real-time simulation technique of simulating immersed rigid bodies in various environments is challenging. The main challenge in the real-time simulation of immersed rigid bodies comes from the need to handle the chaotic principles in an efficient way, which have not been completely resolved in physics, even this research has a rich history beginning with

James Maxwell as described in Chapter 2.

The complexity of the motions of immersed rigid bodies lies in the coupling of the forward motion of the object with lateral oscillations due to surrounding fluid and the production and influence of the vortices around object. The issue of the dynamics of lightweight object body involves multiple hydrodynamic effects(e.g. lift force, drag force and vortex shedding), which exhibit both regular and chaotic behaviors. This topic is challenging and promising in visual simulation of many phenomena, and it related to the researches of unsteady dynamics, such as flight aerodynamics, bubble rising and boiling, meteorology and hydrodynamics.

The common way to produce the motions of immersed rigid body is using key-frame control which requires the animator to exert much effort and expert ability. A reliable motion may be created in a physically-based way by modelling the dynamics of the surrounding flow. While this model involves inertial forces and vortex effects, the approach is not suitable for real-time applications due to its heavy computation cost. In this chapter, a procedural motion synthesis approach that includes the effects of lift and drag forces is proposed to simulate realistic motions of immersed rigid bodies efficiently. The proposed approach also provides proper simulation results under external forces(e.g. wind). The major contributions of the graph-based method for computer graphics are described as follows:

- This method proposes a data-driven approach of motion synthesis that uses a pre-computed trajectory database and a motion graph of motion patterns.
- This method applies a separate synthesis method that uses six motion patterns (Figure 3.1) of freely falling behavior, which are defined in a phase diagram of the dimensionless moment of inertia and Reynolds number.
- In order to connect each motion patterns, the Markov chain model is adopted based on motion groups of these motion patterns that allows an accurate estimation because of the apparent features of each motion pattern, and the estimation is in the terms of a hypothesis about global motion paths of immersed rigid bodies which is verified by experiments.

The rest of this chapter is organized as follows. An overview of the graph-based method is presented in Section 3.2. Section 3.3 describes modelling motion patterns on a phase diagram. Section 3.4 discusses how to synthesize the global motion paths of free fall in a still fluid. The implementation of the wind field in the noise-based method is discussed in Section 3.5. The simulation results of free fall motions in real-time under both wind and no wind conditions are presented in Section 3.6. The conclusion and possibilities for future work of the graph-based method are described in the last section.

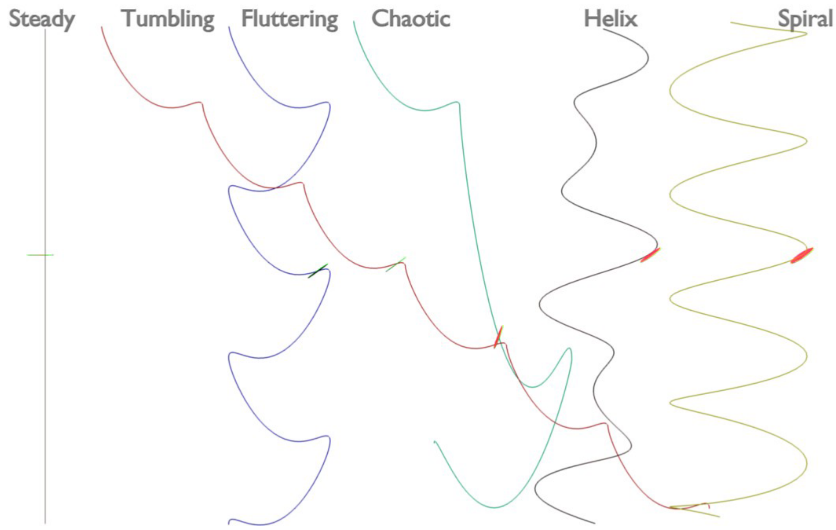


Figure 3.1: Six motion patterns are abstracted from experimental works [APW05b, ZCL11], from left to right: steady decent, tumbling, fluttering, chaotic, helix and spiral motions.

3.2 System overview

The graph-based method is illustrated in Figure 3.2. This method has two important steps: motion modeling and motion synthesis of immersed body's motion. In the motion modeling phase, the input parameters are introduced, including the physical characteristics of the object and the fluid in which it is released (release height, mass, etc.). This method transforms these parameters into two key non-dimensional numbers: the Reynolds number Re and the dimensionless moment of inertia I^*. Next, a phase diagram is utilized to obtain the main motion patterns in which the motion of the object is most stable.

To synthesize the motion patterns of immersed rigid body dynamics, the proposed method uses a trajectory search tree to represent chaotic, fluttering, and tumbling motions, and builds a precomputed trajectory database to provide featured motion segments. The global trajectory is achieved in the motion synthesis phase with an assumed hypothesis of motion classification verified by numerous experiments. In motion graph, motion pattern sequences are treated as nodes, and edges are connected with probabilities from the discrete-time Markov chain model. In addition, the wind field interactions with the falling object are calculated efficiently. In the end, the final simulation of immersed rigid body is achieved in real time.

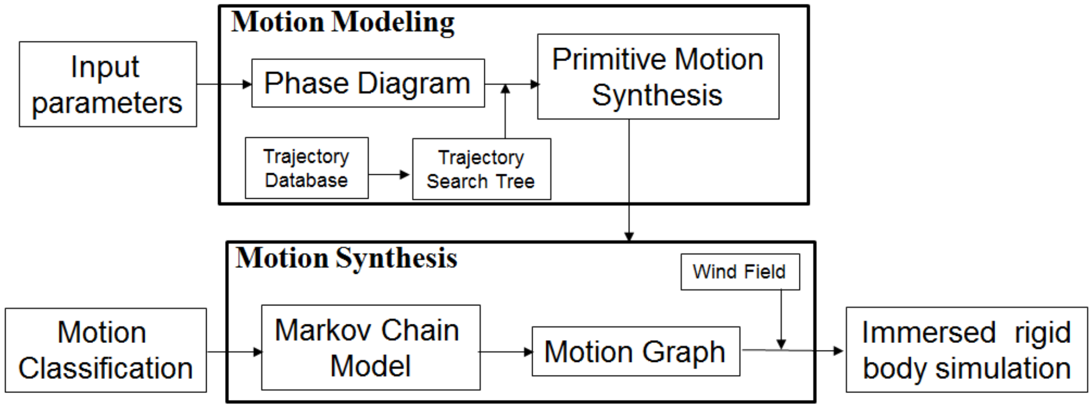


Figure 3.2: Overview of graph-based immersed rigid body dynamics.

3.3 Motion modeling

3.3.1 Input parameters

An immersed rigid body can be characterized by the following quantities:

- h: height of release
- L: length of the object
- a: length of the cross section of the object
- b: width of the cross section of the object
- ρ_s: density of the object
- ρ_f: density of the fluid
- ν: kinematic viscosity of the fluid
- g: gravity acceleration

From these parameters, three dimensionless quantities are derived: the Reynolds number (Re), the aspect ratio of the object ($\epsilon = b/a$), and the dimensionless moment of inertia (I^*). Re and I^* are the two key quantities for building a phase diagram of free fall motion (Figure 3.3). Here,

$$Re = \frac{UL}{\nu} \tag{3.1}$$

where U is the velocity scale of flow. In addition,

$$I^* = \int_V \frac{\rho(x,y,z)}{\rho_f a^5} \begin{bmatrix} y^2+z^2 & -xy & -xz \\ -xy & z^2+x^2 & -yz \\ -xz & -yz & x^2+y^2 \end{bmatrix} dxdydz \tag{3.2}$$

where $\rho(x, y, z)$ is the density function of the object. In special cases, $I^* = \frac{\pi \rho_s b}{64 \rho_f a}$ (disk) and $I^* = \frac{8\rho_s(a^2+b^2)b}{3\pi\rho_f a^3}$ (rectangle).Commonly, the velocity scale U is approximated by the average descent velocity of the moving object:

$$U \sim \sqrt{(\frac{\rho_s}{\rho_f} - 1)gb} \tag{3.3}$$

For an immersed rigid body, the object aspect ratio ϵ is usually so small that can be

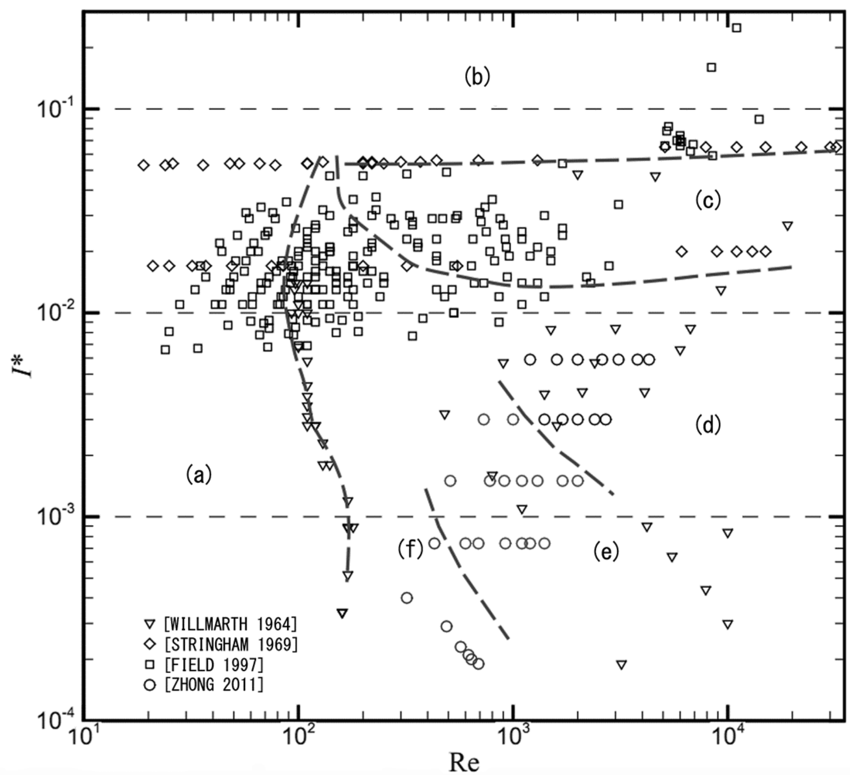


Figure 3.3: The *Re*-I^* phase diagram of immersed rigid body motions from [WHH64, SDG69, FKMN97, ZCL11], including six regimes:(a) steady descent, (b) tumbling, (c) chaotic, (d) fluttering, (e) helix, and (f) spiral motions. The symbols in the diagram represent experimental results from previous works.

omitted the its effect on the motions($\epsilon \ll 1$).

A *Re*-I^* phase diagram is illustrated as Figure 3.3. The regimes in the diagram represent different motion patterns. The tumbling, fluttering, and spiral motions almost appear periodic; the chaotic motion appears to be the transitional motion between the tumbling and fluttering motions; and the helix appears to be the transitional motion between the spiral and fluttering motions. The definitions of the motion patterns are as follows:

- (a) steady descent (SD): the object drops straight down in the vertical direction

- (b) periodic tumbling (PT): the object turns continuously end-over-end and drifts in one direction
- (c) transitional chaotic (TC): the object begins to oscillate with increasing amplitude, and the fluttering motion finally turns into a tumbling motion, and chaotic motion is observed
- (d) periodic fluttering (PF): the object oscillates from side to side with a well-defined period
- (e) transitional helix (TH): the object moves in a helical path at a constant speed
- (f) periodic spiral (PS): the object falls downward circularly in three-dimensional space.

The motion trajectories of these motion patterns are illustrated in Figure 3.1.

3.3.2 Precomputed trajectory database

It is not easy to build a trajectory database of immersed rigid body motions, because capturing accurate trajectories of a falling small object in the real world seems to be infeasible due to their chaotic motions and the short time interval. Using fluid simulations to track vortex particles from frame to frame by following velocity vectors is also not suitable for the following reasons: (1) they cannot detect all motion patterns; (2) they have difficulty capturing realistic motion trajectories; (3) various parameters adjustments make such simulations difficult to control.

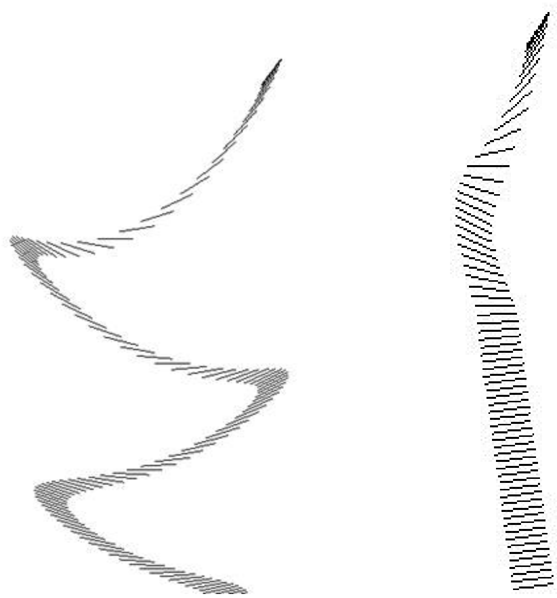
Another approach is to use the Kutta-Joukowski theorem [TK94] accounting for drag and lift forces, The dynamical model of the immersed body can defined in the following ODEs:

$$\begin{cases} \ddot{x} = -(A_{\perp} sin^2\theta + A_{\parallel} cos^2\theta)\dot{x} + (A_{\perp} - A_{\parallel}) sin\theta cos\theta \dot{y} \\ -kL\pi\rho_f V^2 cos\beta cos\alpha/m \\ \ddot{y} = -(A_{\perp} cos^2\theta + A_{\parallel} sin^2\theta)\dot{y} + (A_{\perp} - A_{\parallel}) sin\theta cos\theta \dot{x} \\ +kL\pi\rho_f V^2 cos\beta sin\alpha/m \\ \ddot{\theta} = -A_{\perp}\dot{\theta} - 3\pi\rho_f V^2 cos\beta sin\beta \end{cases} \tag{3.4}$$

where (x, y) and θ denote the position and angle of the center of mass of the object, respectively. In addition, (u, v) and ω are the linear and angular velocities of the object. Note that $u = \dot{x}, v = \dot{y}, \omega = \dot{\theta}$, and $V^2 = \dot{x}^2 + \dot{y}^2$. Moreover, m is the mass of the object, which is calculated from the object's density and aspect ratio parameters. The parameters $A_{\perp}$ and $A_{\parallel}$ are the drag coefficients in the directions perpendicular and parallel to the object, respectively. The angles α and β are defined as $\alpha = arctan(u/v), \beta = \alpha + \theta$. The parameter k is defined as follows,

$$k = \begin{cases} 1 : sign(v) sin\beta \geq 0 \\ -1 : sign(v) sin\beta < 0 \end{cases} \tag{3.5}$$

A standard fourth-order Runge-Kutta algorithm is utilized to solve this second-order ODEs in Equation (3.4), as shown in Figure 3.4 (a). Similar to a fluid simulation approach, it is difficult to control the ODEs model with the parameters $A_\perp$ and $A_\parallel$. For example, the calculated motion trajectory in Figure 3.4 (b) is useless, because it is not natural for an object to fall vertically after a fluttering motion. Nevertheless, because Equation(3.4) accounted for the effects of drag an d lift forces, the results of object orientations by solving Equation(3.4) are more accurate than other approaches.



(a) (b)

Figure 3.4: (a) Fluttering trajectory determined by solving the ODEs in Equation(3.4) with $A_\perp = 4.1, A_\parallel = 0.9$ (b) Useless trajectory by solving the ODEs in Equation(3.4) with $A_\perp = 4.6, A_\parallel = 0.15$.

There are two essential steps before building a trajectory database: motion segmentation of free fall trajectories and segment clustering. To obtain various motion segments, we use a harmonic functions to describe general fluttering motions:

$$\begin{cases} x_t = x_0 - \frac{A_x}{\Omega} sin(\Omega t) \\ y_t = y_0 - Ut - \frac{A_y}{2\Omega} cos(2\Omega t) \end{cases} \tag{3.6}$$

where A_x and A_y are the amplitudes of vertical and horizontal velocities of the falling object generated by oscillations due to the surrounding viscous flow, Ω describes the angular frequency of falling motion, and U is calculated from Equation (3.3). The segment breakpoints are chosen as the turning points of the trajectory given at time steps $t_i = \frac{2k+1}{2\Omega}\pi, k \in Z \geq 0$.

After the step of segmentation, there are numerous motion segments obtained by modifying the parameters in Equation (3.6) as shown in Figure 3.5. A motion segment set $(S_i|i = 1, 2...N)$, where N is the number of segments, is classified based on the value of the feature vector of each segment from the start point P_i^0 to the end point P_i^1. Feature vector sets $V\{V_i = P_i^1 - P_i^0, i = 1, 2...N\}$ are assigned into classes using the K-means algorithm.

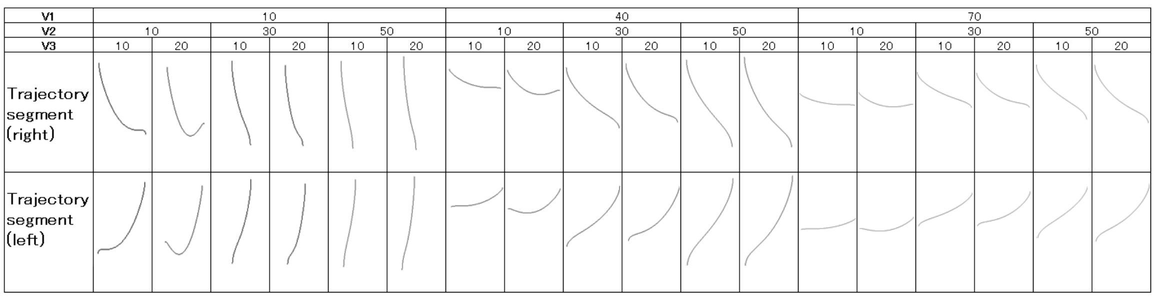


Figure 3.5: Trajectory segments obtained from Equation (3.6).

The orientation of the object in each frame of S_i is linearly interpolated by the angles calculated from Equation (3.4) as shown in Figure 3.6. Finally, the position and orientation data of segments are stored in a trajectory database.

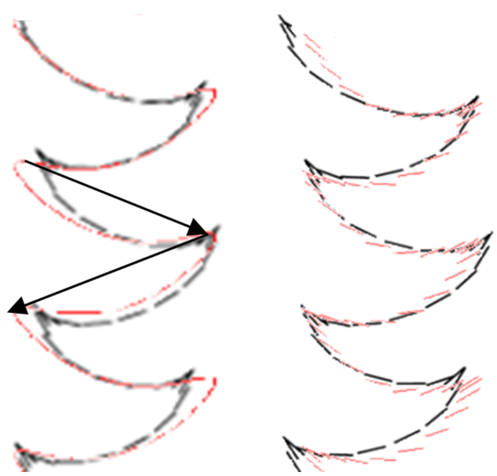


Figure 3.6: Comparing synthesized trajectory (red) and measured data (black) of fluttering motion, for no orientation (left) and interpolated orientations (right). Arrow lines represent feature vectors.

3.3.3 Trajectory search tree

This work compares the trajectories of chaotic motion, periodic fluttering and tumbling motions from experimental data. Because the airflow behind a falling object reveals vortex shedding and turbulence, the object comes to turning points, where the angular velocity becomes zero, and the velocity in the oscillation direction is also zero, but the velocity in the vertical direction is maximized. The object faces two alternatives of sliding left or sliding right (fluttering or tumbling). Simple structures of fluttering, chaotic, and tumbling motions are illustrated in Figure 3.7. In this tree structure, every child represents a motion segment derived from the precomputed trajectory database using the feature vector as the search key.

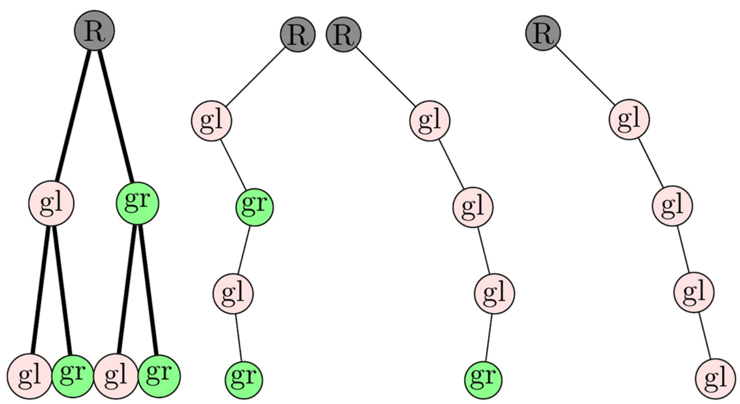


Figure 3.7: The first two levels of a small trajectory search tree (1) and the tree structures of fluttering (2), chaotic (3), and tumbling motions(4) created by traversal of four levels of the search tree. (gl: glide left; gr: glide right)

3.3.4 Unified trajectory functions

When projecting the motion paths of motion patterns onto XY plane, we notice that the curves of six motion patterns have characteristic shapes: the steady descent motion trajectory is one point; the fluttering, tumbling and chaotic motion trajectories are in a straight line; the spiral motion trajectory is a circle; the helix motion trajectory is similar to an eight-petal rose curve(Figure 3.8). These curves are all represented in the following equations:

$$\begin{cases} x_t = A_e cos(\Omega t)(1 + \epsilon_e sin(k\Omega t)) \\ y_t = A_e sin(\Omega t)(1 + \epsilon_e sin(k\Omega t)) \\ z_t = h - Ut \end{cases} \tag{3.7}$$

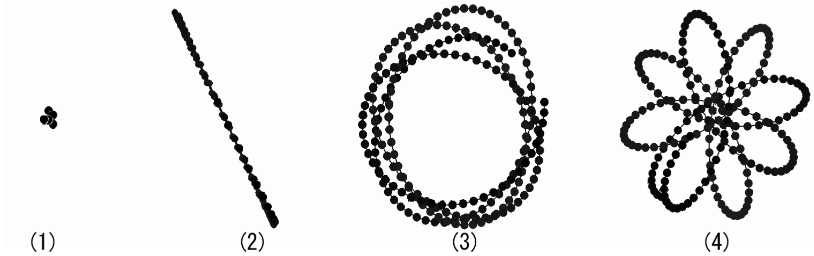


Figure 3.8: Measured curves projected onto theXY plane: (1) steady descent motion; (2) fluttering& chaotic& tumbling motions; (3) spiral motion; and (4) helix motion.

where A_e is the amplitude of the elliptical oscillation generated in the XY plane, ϵ_e is the aspect ratio of the minor axis and the major axis of the oscillation ellipse, k is the ratio of the period of elliptical oscillation to the period of rotation of the falling object, Ω is the angular frequency of the falling motion. As deduced from Equation (3.7), the simple form of trajectories is as follows:

$$\begin{cases} \epsilon_e \to 0, k = 1 : PS \\ A_e \to 0, k \to 0 : SD \\ \epsilon_e \neq 0, k = 4 : TH \end{cases} \tag{3.8}$$

3.3.5 Initialization of motion patterns

Because the nature of chaotic motion is more complex than that of periodic motions (PF, PT), we synthesize chaotic motion with a feature vector $V = rV_0$ and an amplitude of oscillation $A_e = rA_0$, where r is a random number between 0.1 and 10 calculated by the Box-Muller algorithm, V_0 is a feature vector whose orientation is the release angle of the falling object, and A_0 is the initial amplitude of object oscillation.

From experimental data [Raz10], we know that the deviations of motion patterns D_i from the release position are distributed in a normal distribution of Gaussian functions $A_i e^{-(\frac{r-B_i}{C_i})^2}$ and linearly related to the release height h.

$$D_i = \frac{kB_iL}{a} \tag{3.9}$$

where k is the deviation coefficient. Because the frequency of tumbling motion is given as $\Omega \sim \sqrt{b}/a$ [MRS99], the initial amplitude of oscillation A_0 is given by:

$$A_0 = \frac{D_iU}{h\Omega} \tag{3.10}$$

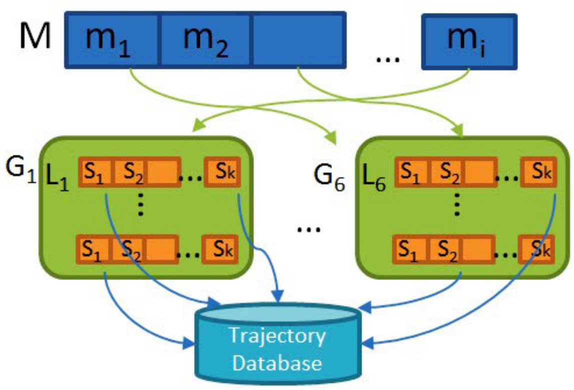
in which U is the average falling velocity from Equation (3.3). The final synthesized trajectories of motion patterns are shown in Figure 3.1.

3.4 Motion synthesis

3.4.1 Motion classification

The motion patterns $\{L_i|S_1, S_2, ...S_k\}$ (S_k is the k-th segment in motion pattern L_i) are synthesized by motion segments S_n from a precomputed trajectory database. Because motion patterns are the basic motions of immersed rigid bodies observed in various experimental works, the motion groups $\{G_i|1 \leq i \leq 6\}$ are used to represent this similar motion sets of motion patterns.

For a free fall motion M, $M\{M = m_1 \parallel m_2 \parallel ...m_i\}$ is annotated by a label, which is set to be L_i, where L_i is a motion pattern. We define L_i as follows: L_1: steady descent motion(SD);L_2: fluttering motion(PF); L_3: chaotic motion(TC); L_4: tumbling motion(PT); L_5: helix motion(TH); and L_6: spiral motion(PS), as shown in Figure 3.9 (a).



(a)

Class 1	L_4
Class 2	$L_3 \rightarrow L_6, L_3, L_4 \rightarrow L_6$
Class 3	$L_1 \rightarrow L_2, L_1 \rightarrow L_6$
Class 4	$L_4 \rightarrow L_6, L_3$
Class 5	$L_2 \rightarrow L_6$
Class 6	L_4, L_6
Class 7	L_1

(b)

Figure 3.9: (a) Motion classifications (blue: motion classes; green: motion groups; Brown: motion segments). (b) Motion classes observed in experiments.

In terms of thousands of experiments[Raz10], the trajectories are classified into seven motion classes (Figure 3.9 (b).) and we make the following hypothesis:

Hypothesis *If* $M\{M = m_1 \parallel m_2 \parallel ... \parallel m_i\}$ *represents the motion of immersed rigid body in three dimensions, then* $m_i = L_{j_i} \in \{L_j|1 \leq j \leq 6\}$*, and the subscript sequence* $\{j_1, j_2, ..j_i\}$ *should be an increasing sequence.*

This work analyzes this hypothesis qualitatively. When an object starts falling from a release point, the vortexes are gradually generated behind the object because of the vorticity of the surrounding flow. Then, the motion of immersed rigid body becomes increasingly sensitive to internal forces, including drag and lift forces.

In terms of this hypothesis, the number of potential motion classes of all motion patterns is determined by

$$N = \sum_{i=1}^{i\leq k} C_k^i, k \in Z, k \in [1, 6] \tag{3.11}$$

where k is the level of the main motion pattern by looking up phase diagram using calculated Re and I^* in Section 3.3.1.

3.4.2 Markov chain model

The proposed model focuses on how to determine m_i for motion M. A first-order discrete-time Markov chain model is proposed for solving this issue. Let us consider a discrete-time stochastic process $\{X_n\}$ with $N_0 \in Z \equiv i \in [1, 6]$ as the state space, which corresponds to motion groups $\{G_i | 1 \leq i \leq 6\}$ (Figure 3.9). The Markov property asserts that the distribution of the random variable X_{n+1} in the process $\{X_n\}$ depends only on the current state $X_n = i_n$, instead of depending on the whole history $\{X_0 = i_0, ...X_n = i_n\}$:

$$P[X_{n+1} = j | X_n = i_n] = P[X_{n+1} = j | X_0 = i_0, ...X_n = i_n] \tag{3.12}$$

where $j, i_0, ..., i_n \in N_0$. The stochastic process $\{X_n\}$ is a Markov chain.

Let the state space N_0 be the motion group G and let process $\{X_n\}$ on discrete time set $\{X_t\}$, then the transition probability $p_{ij} = P[X_{t+1} = L_j | X_t = L_i]$ is the conditional probability to transition from motion pattern L_i to motion pattern L_j. The transition matrix is given as $P = (p_{ij})$ (Figure 3.10). Because the process $\{X_t\}$ is stochastic, the matrix requires

$$p_{ij} \geq 0, and \quad \sum_j p_{ij} = 1 \quad i, j \in [1, 6]$$

According to the Hypothesis in Section 3.3.1, P has the following form

$$P = \begin{pmatrix} Q & R \\ 0 & T \end{pmatrix}_{i \times j} \tag{3.13}$$

Next, we discuss the realization of Markov chain $\{X_t\}$ and the transition matrix P calculation. To obtain the process $\{X_t\}$, the model starts with an initial state at time $t_0 = 0$. Then an iteration step is executed, for the state L_i at time t to state L_j at time $t + 1$, and the calculation depends on the probabilities at the i-th row of the transition

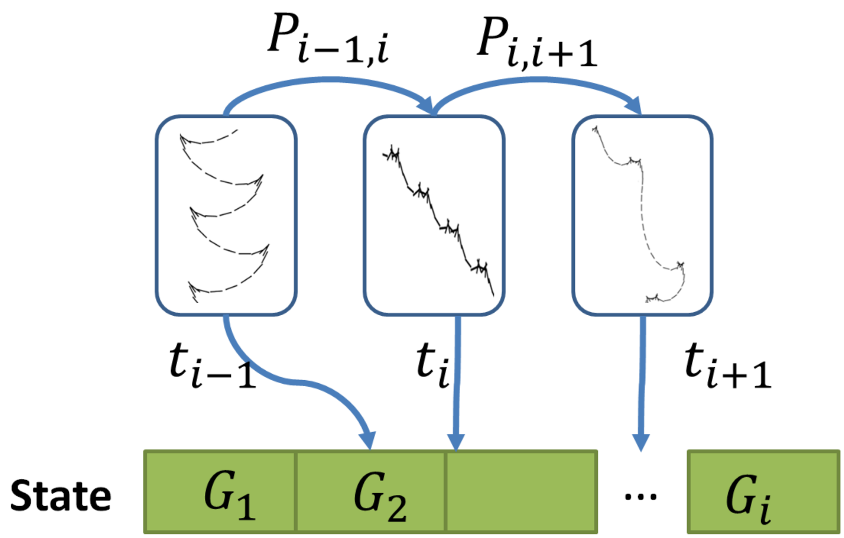


Figure 3.10: Markov chain model states and transition probabilities.

matrix P (i.e. $P_i = (p_{ij}|j = 1, 2..., 6)$).

The state transition probabilities for the transition matrix P are found by counting the state transitions that occurred in experimental data. Let us set N_i be the number of all transitions from state L_i in the experimental data, N_{ij} be the number of transitions from state L_i to state L_j, then the probability is given as $p_{ij} = \frac{N_{ij}}{N_i}$.

The advantages of using this first-order discrete-time Markov chain model are as follows:

- The next motion is only related to the current state in the case of motion patterns.
- The features of each motion pattern are so apparent that a valid transition matrix can be obtained from experimental data successfully.
- The computational cost of the model is low.

3.4.3 Graph construction

A special motion graph of immersed rigid body (Figure 3.11) is based on motion graph described in ref. [KGP02]. This graph is a complete directed graph: each node of the graph is connected to other nodes in the same graph. We use $G = (V, E)$ to represent a graph, where V is the node set, and E is the edge set. Every frame in a motion sequence of motion pattern appears as a node in the motion graph; a transition splice in the motion sequence appears as an edge between nodess. We search this graph in one direction (from top to down) in the order of m_i in the hypothesis presented in Section 3.4.1. Therefore, it is impossible for a motion to become tumbling motion after spiral motion.

Furthermore, the transition probability is attached to the edge between the nodes using

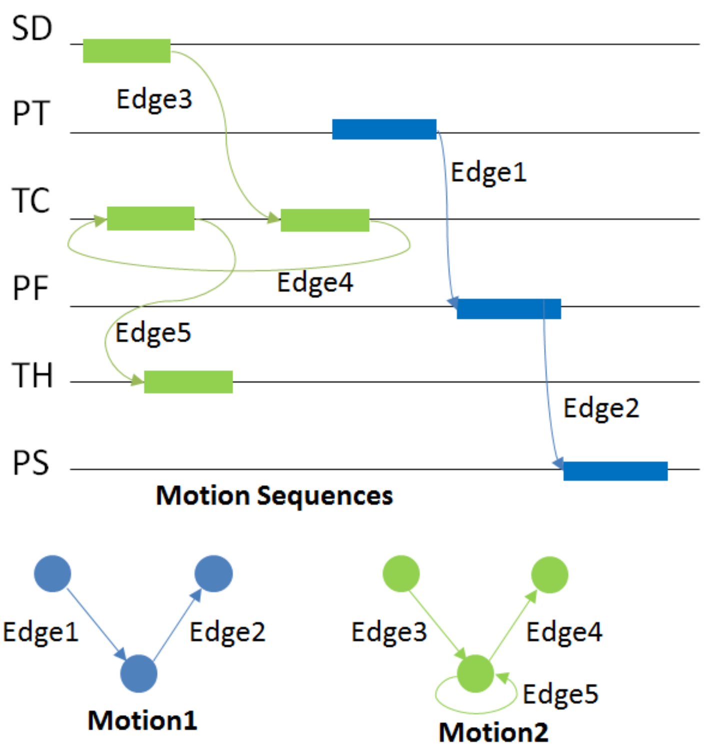


Figure 3.11: Free fall motion graph. A motion path represents a collection of splices between sequences. Here, two example motions are shown.

discrete-time Markov chain model (Section 3.4.2).

3.5 Motion in wind

3.5.1 Wind field

To obtain a wind field, a direct and straight method is to simulate the turbulent flow under a boundary condition by solving differential equations using a Fourier filter [SF92, SE93]. Another methods is to simulate the motion in a wind field using noise functions (fractional Brownian motion) [OTF$^+$04, KGC11]. Comparing the flow-based and noise-based methods, the flow-based method provides physically accurate and realistic results but requires a high computation cost. Whereas, the noise-based method is much simpler and more suitable for real-time simulation but at the cost of physical accuracy. To overcome the inaccuracy of the noise-based method, physically-based analysis of wind characteristics is to be essential.

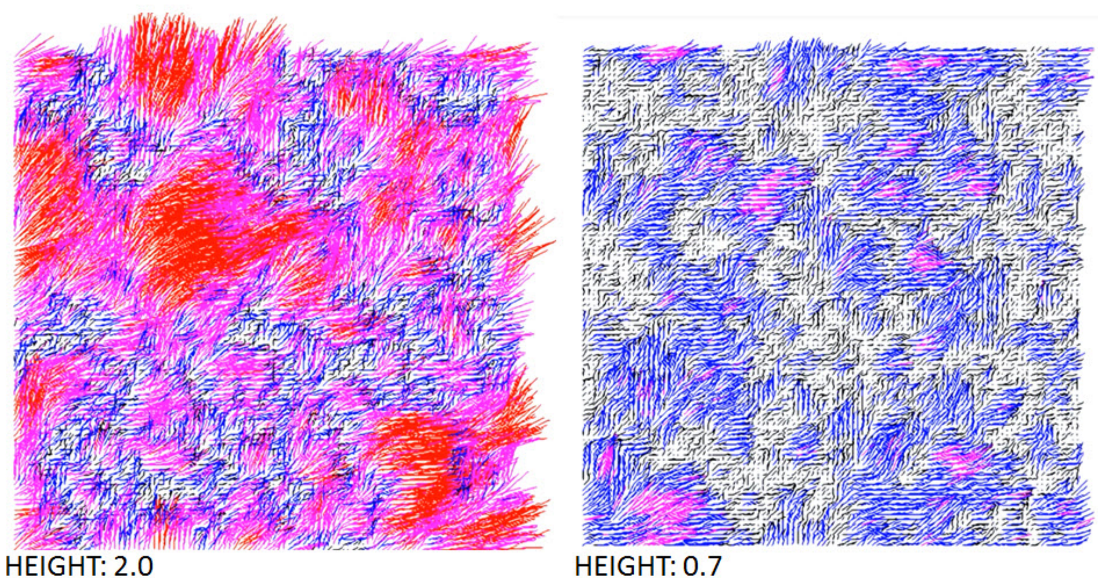


Figure 3.12: Two-dimensional wind field, HEIGHT represents the distance (m) above the ground. The color represents the velocity length compared to the mean wind velocity U(red: $V > 2U$; pink: $U < V \leq 2U$; blue: $U/2 < V \leq U$; black: $V \leq U/2$), the wind direction is along the x-axis.

Let the velocity of wind be $V = (V_u, V_v, V_w)$, where V_u, V_v, V_w describe wind velocity components along the x-, y- and z-axes of the coordinate system in the simulation. In addition, let $U(h)$ be the mean wind velocity at height h. According to the logarithmic wind law [TL64], $U(h)$ is given by

$$U(h) = \frac{u_*}{k} \ln(\frac{h}{z_0}) \tag{3.14}$$

where u_* is the friction velocity (m/s), k is the von Karman's constant ($k = 0.40$), and z_0 is the roughness parameter (conceptually it is the height where V goes to 0). The value of z_0 depends on the types of ground terrain (we choose $z_0 = 0.3$). The fBm method can suitably represent the wind [OLH84]. The spectral density function of the wind field is given as follows based on Kolmogoroff's law:

$$S_u(n) = u_*^2(\frac{U(h)\phi}{h})^{2/3}\frac{C}{n^{5/3}} \tag{3.15}$$

where $\phi = \epsilon kh/u_*^3$, ϵ is the dissipation rate according to Kolmogoroff's law, and C is a constant, $C = \alpha(2\pi k)^{-2/3}$, where α is determined experimentally to be 0.5. Therefore, $C = 0.3$ for the u wind direction and $C = 0.4$ for the v and w wind directions. To obtain the representation of fBm in the form of $S(f) = A/f^{\beta}$, where A is the amplitude in wind direction (u, v, w)), we adopt the same approximations of A_u, A_v, A_w from ref.[KGC11] as follows:

$$\begin{aligned} A_u &= u_*(\frac{U(h)}{h})^{2/3}, \beta = 5/3 \\ A_v &= 0.88A_u \\ A_w &= 0.55A_u \end{aligned} \tag{3.16}$$

where u_* is calculated from Equation (3.14). To obtain the wind velocities, we apply the inverse Fourier transform to the following equation:

$$S_p(f_1, ..., f_n) = \frac{A_p}{(\sqrt{\sum_{i=1}^{n} f_i^2})^{\beta+n-1}} \tag{3.17}$$

where n is the dimension number, and p is the wind direction (u, v, w).

Considering the computation costs of 2D and 3D wind fields (2D grid 100×100: 8 ms; 3D grid $100 \times 100 \times 100$: 1,363 ms), we use a 2D wind field to approximate a 3D wind field by using the release height of the falling object. The wind field $u(p, t)$ is represented as:

$$u(p, t) \equiv u(p', h, t) = \frac{\ln(h) - \ln(z_0)}{\ln(h_0) - \ln(z_0)} u(p', h_0, t) \tag{3.18}$$

where p' is the 2D position, and h_0 is the release height of the falling object. A 2D wind field for two different heights is illustrated in Figure 3.12 (mean wind velocity: $U = 4.0$

m/s; grid size: 100×100).

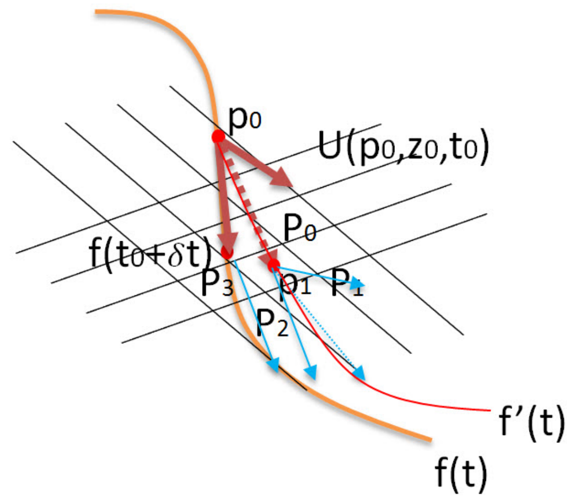


Figure 3.13: Trajectory of freely falling behaviour in wind field

3.5.2 Wind-object interaction

Let $u(p, h, t)$ be a 2D wind field at position p and height h, and let the wind velocity at point $p_0 = (x_0, y_0, z_0)$ be $u(p_0, z_0, t_0) = (u_x, u_y, u_z)$, where u_x, u_y, u_z correspond to the u,v,w components in Equation (3.16).

To represent the computed trajectory of a falling object in a still fluid, we set the trajectory to be a function $f(t)$, where $f(t)$ is a set of points per frame in the time domain. At time t_0, $f(t_0)$ is a quaternion (p_0, θ_0), which includes the position and orientation of the object.

After time step δt, $f(t_0 + \delta t)$ comes to point p'. The next point after p_0 is set to be p_1, $\widehat{p_0p_1} = u(\widehat{p_0, z_0}, t_0) + \widehat{p_0p'}$ (Figure 3.13). If p_1 does not coincide with any grid node of the wind field, assuming the neighboring 2D grid nodes around p_1 are $Pi(0 \leq i \leq 3)$, the wind velocity at p_1 is calculated from the linear interpolation of u_i at P_i. After the iterations, the new trajectory $f'(t)$ of the falling object in a wind field is synthesized using a Bezier curve to produce a smooth path with control points $p_i(i = 0, 1, ...)$ (Figure 3.14). In contrast to other curve-fitting methods, Bezier curve has advantage accounting for all control points.

Next, we consider the rotation of an object under the influence of wind. Note that a falling object, such as a leaf or a piece of paper, can change its orientation in a wind field. To achieve a realistic effect of wind, we apply a noise function into the orientation

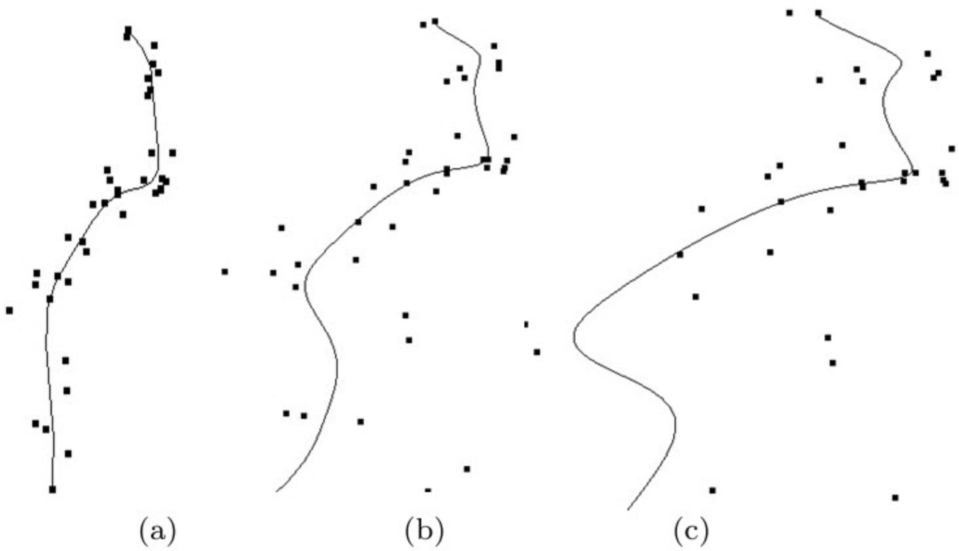


Figure 3.14: Motion paths for different values of the mean wind velocity U: (a) $U = 1.0$ m/s,(b) $U = 3.0$ m/s, (c) $U = 5.0$) m/s. The wind direction is from right to left.

calculation of the falling object:

$$\theta(t) = WN(t) \tag{3.19}$$

where $N(t) = \sqrt{u_x^2 + u_y^2 + u_z^2}/|U|$, where U is the mean velocity of wind field, u_x, u_y, u_z are the velocity components of wind at time t, and W is the maximum motion angle, which is designated by the animator.

3.6 Results

From the input parameters in Section 3.3.1, the main motion pattern L_i is determined by looking up the phase diagram using calculated I^* and Re. According to the hypothesis in Section 3.4.1 and the Markov chain model, the global path synthesis starts from a random motion pattern $L_j(i < j)$, and the next motion pattern is estimated by transition matrix in Section 3.4.2. The motion pattern segments are determined from the precomputed trajectory database in Section 3.3.2. To efficiently evaluate our simulations, we compare them with experimental videos of falling immersed rigid bodies.

The simulation results presented in Figure 3.15-3.18 suggest that our simulations are realistic and our approaches are applicable in various flows, such as water and air. Figure 3.15 shows an aluminium circular disk (radius: 1.0 cm; thickness: 0.15 cm) falls in still water from a height of 50 cm. A regular fluttering motion is observed. We use $Re - I^*$ phase diagram to determine the main motion pattern is fluttering for $I^* = 10^{-2}$ and $Re = 3.55 \times 10^3$. Note that both the simulation and the video show fluttering motion as

Figure 3.15: Comparison of the simulation with the ground truths of one-Japanese yen coin falling motion in water.

the global falling path.

Figure 3.16 shows an elliptical piece of paper (major axis: 8.0 cm; minor axis: 2.0 cm; thickness: 0.01 cm) falling in still air from a height of 3.1 m. Because $I^* = 2.2 \times 10^{-3}$ and $Re = 6.8 \times 10^3$, the $Re - I^*$ phase diagram indicates that the main motion pattern is a spiral motion. Note that the falling motion consists of tumbling and spiral motions, which is consistent with the hypothesis we presented in Sect. 3.4.1.

Figure 3.17 (a) shows a leaf (major axis: 7.3 cm; minor axis: 4.2 cm; thickness: 0.03 cm) falls in still air from a height of 2.0 m. Because $I^* = 6.3 \times 10^{-3}$ and $Re = 1.2 \times 10^4$, the $Re - I^*$ phase diagram indicates that the main motion pattern is a transitional helix motion. Note that the simulated result consists of steady descent, tumbling, and helix motions. Figure 3.17 (b) and (c)show the final simulated motions in two strong wind fields that corresponding to the condition in Figure 3.14 (b) and (c), respectively. We found that under a strong wind field, tumbling motion disappears and the object travels far.

Figure 3.18 shows an integrated scene of multiple leaves falling motion from a tree. Although the Re and I^* are the same values for these leaves, they have different motion trajectories under the Markov chain model. Note that the implementation of the simulation

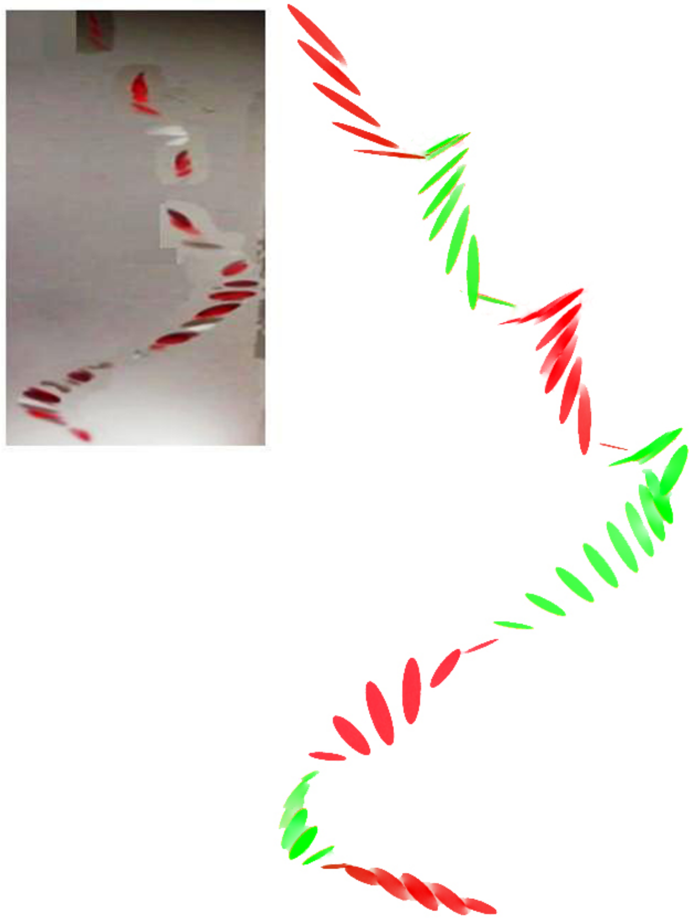


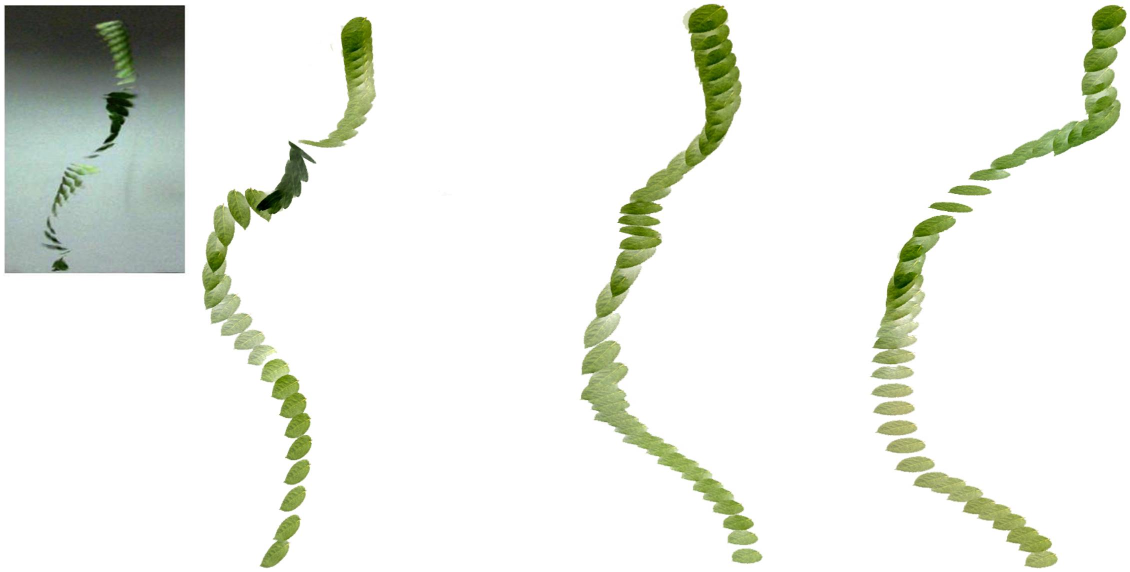
Figure 3.16: Comparison of the simulation with the ground truth of a paper falling motion in air. The figure of the ground truth is blurry because of the dim background in each frame of the captured video.

does not include collision detections among objects.

All simulations are implemented on an Intel Core i7 CPU 3.20 GHz and 12.0 GB RAM in real time (less than 3 ms per frame for one body). Because most of the computation was executed off-line, the runtime motion synthesis and optimization process were rarely memory consuming, therefore, our simulation is not only realistic but also feasible for interactive applications.

3.7 Discussion

This chapter presented a framework for simulating realistic motions of immersed rigid bodies in both still fluids and wind fields. In addition, this study presented the frontier research on the physical details of the dynamic of immersed rigid body. Furthermore,



(a) (b) (c)

Figure 3.17: (a) Comparison of the simulation with the ground truth of a leaf falling motion in air. The simulation of the motion in a wind field of mean wind velocity $U = 3.0$ m/s (b) and $U = 5.0$ m/s (c). The wind direction is from right to left in a screen space.

this work proposed an efficient motion synthesis method to achieve realistic simulations in real-time.

The limitation of this work is for rigid bodies with regular geometries (such as, rectangular, circular, and elliptical objects with constant densities). For rigid body with irregular geometry and uneven density distribution, it is difficult to determine the influences of the geometry and density distribution modifications on the motion. When a paper or plastic object falls freely, the deformation of shape would happen, which is not considered in this work. Furthermore, the orientations of rigid body are obtained from the precomputed trajectory database, but the rotational axis and angle of the rigid body are difficult to determine in the cases of spiral and helix motions, because they are related to its geometry and appear irregular observed in ref.[Raz10].

This chapter proposed the graph-based method mainly based on the observations from physical experiment. To solve the issues for complex geometries, this thesis proposes a new numerical method based on the dynamical model of the immersed dynamics in the next chapter.

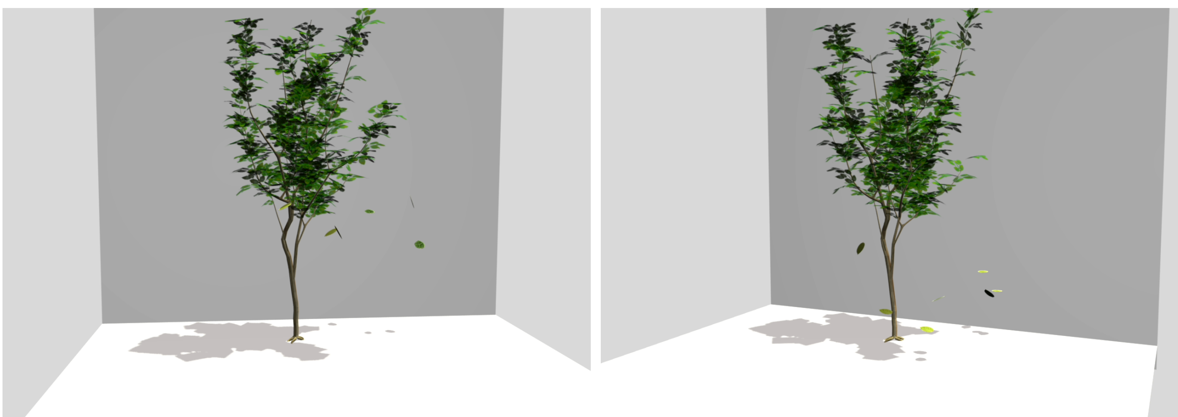


Figure 3.18: Simulation of multiple leaves falling from a tree.

Chapter 4

Stochastic model of immersed rigid body dynamics

This chapter presents a stochastic model for animating the dynamics of immersed rigid bodies in viscous incompressible fluid in real-time. This uses generalized Kirchhoff equations to ensure forces and torques from the surrounding fluid that create realistic motion of immersed rigid bodies. The proposed method utilizes the generalized Langevin equations to represent the effects of turbulent flow generated at the body surface. This approach precomputes added-mass effects and the vortical loads from turbulent model, and executes the rigid body solver in runtime, so that this method is straightforward and efficient to the interactive simulations. Many types of rigid bodies with lightweight mass (e.g. leaf or paper) can be simulated realistically in high-Reynolds-number flows.

4.1 Introduction

Rigid body simulations are the fundamental techniques in computer animation, which are ubiquitously used in various applications. Although current rigid-body solvers can handle the body dynamics and collisions sophisticatedly, it remains a challenging work to simulate immersed rigid-body dynamics, which considers the motion of rigid bodies fully immersed in air or submerged underwater. In our daily-life, we notice that a paper moving through air follows a beautiful but chaotic-like trajectory rather than a straightforward vertical path.

The motion of immersed rigid body can be characterized by a Reynolds number Re and its mean falling velocity U_0.

$$Re = \frac{U_0 d}{\nu}; \qquad U_0 = \sqrt{(\frac{\rho_b}{\rho_s} - 1)gb} \tag{4.1}$$

where ν is the kinematic viscosity of the surrounding fluid; d and b are the characteristic length and thickness of the rigid body; ρ_b and ρ_f are the densities of rigid body and fluid, respectively. For a common leaf moving in air, Re is greatly high at a magnitude of 10^4. In the case of a high-Reynolds-number flow, the vortices born around the body and then detach from body surface as vortex shedding. The immersed rigid-body dynamics is so unsteady with path instabilities that the simulations of immersed rigid bodies become notoriously difficult in the field of both fluid mechanics and computer graphics.

In contrast to the conventional two-way coupling simulations, high-Re two-way coupling is far complicated where the strong-coupled motions between rigid bodies and flow cannot be understood without fine details of fluid motion, i.e. vortex-body interaction. To the best of authors' knowledge, the technique for simulating high-Re two-way coupling of immersed rigid bodies is absent in computer animation. Therefore, it is difficult to simulate immersed rigid bodies by conventional coupling approaches. To resolve this issue, the generalized Kirchhoff equations are utilized with detailed analysis of the flow effects from the surrounding flow.

The research motivation of this work is to supply a plausible simulation approach of immersed rigid-body dynamics in real-time on CPU. Because this is a great challenging problem to account for turbulent flow for real-time simulations, this work assumes that the body is thought as a passive particle in the fluid flow with the mean falling velocity of the body whereas the real situation is a body moving through the still flows, so that the implementation of turbulent energy need not handle body's boundary conditions and can be executed in pre-processes. Due to the absence of boundary conditions, the aerodynamic drag and lift forces are resolved implicitly in this work. In contrast to the empirical model of aerodynamic forces based on quasi-steady assumptions, this approach can achieve visually plausible simulation results accounting for the viscous effects of unsteady forces from generated turbulence but lose physical accuracy as trade-off with computation cost.

In terms of the assumption, this work propose a stochastic model as a tradeoff between the computation costs and simulation accuracy to resolve the dynamics of immersed rigid bodies [XM14a]. As illustrated in Figure 4.1, this approach separates flow effect from the surrounding flow into inertial effect from potential flow and viscous effect from turbulent flow:

- For the inertial effect, this model precomputes the added-mass tensors due to both translational and rotational displacements of the surrounding flow.
- For the viscous effect, this model calculates the turbulent energy and its dissipation rate for obtaining vortical loads on the body in a pre-process stage.

The vortical loads of viscous effect are represented in the Langevin model as a stochastic process of the object velocity, and then substituted into Kirchhoff equations with added-

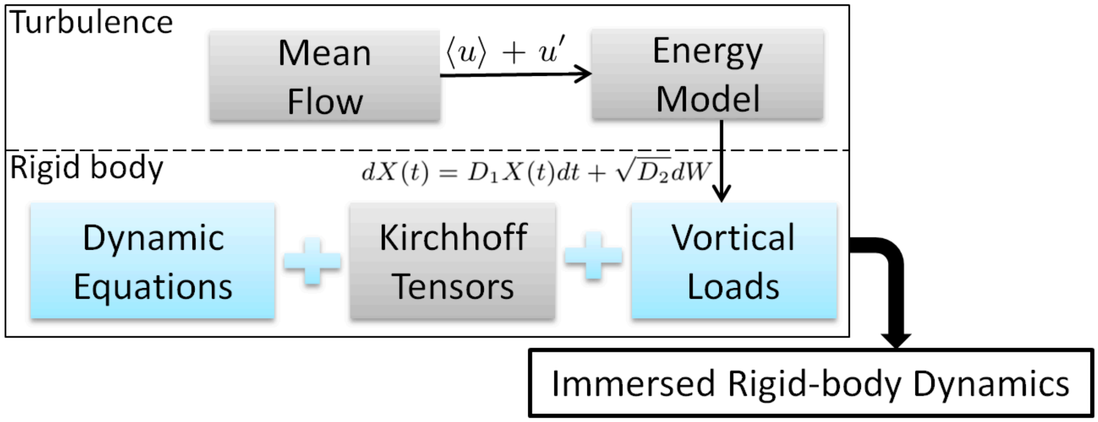


Figure 4.1: Overview of the proposed stochastic model. The steps in grey color run in precomputation steps. The runtime includes only two steps in blue color so that the computation cost is significantly reduced.

mass tensors. The proposed model runs a rigid body solver to solve generalized Kirchhoff equations in runtime process. Overall, the proposed approach makes it feasible to efficiently simulate immersed rigid bodies with arbitrary shapes in low computation costs. The major contributions of this work are summarized as follows:

- A new method based on generalized Langevin equations of both translational and rotational velocities to represent the characteristics of the surrounding flow whereas previous work [CZY11, YCZ11] did not account for the dynamics of rigid bodies and the coupling between translational and rotational velocities.
- A new representation of rigid body dynamics as generalized Kirchhoff equation in body-fixed frame to account for both inertial and viscous effects, which is different from previous work [WP12] where only inertial effect was considered.
- A two-stage framework includes pre-processes stage of added-mass effects and the k-ε turbulent model, and runtime stage of rigid body solver, which is shown to be efficient to simulate immersed body dynamics in real-time.

The rest of this chapter is organized as follows: Section 4.2 details the equations of rigid body by generalized Kirchhoff equations. Section 4.3 explains the Langevin model to capture the motion in a stochastic process way, and Section 4.4 describes the approach on how to obtain turbulent parameters from the k-ε turbulent model. Section 4.5 specifies the algorithms used in the implementation of the proposed approach. Section 4.6 shows the simulation results of different objects by the propose model. Finally, this chapter is concluded with a discussion of limitation of the proposed method in Section 4.7.

4.2 Equations of motion

Let us consider a rigid body of mass m, and center of mass O moving through a still fluid flow. The motion of the rigid body is described by $(R(t), \boldsymbol{x}(t))$. $R(t)$ represents the orientation of the body as a 3×3 orthogonal matrix rather than a quaternion form, and $\boldsymbol{x}(t)$ is the position of O at time step t in inertial reference frame. This work represents the dynamic equations of motion in body-fixed frame. All symbols used throughout this chapter can be found in Table 4.1.

4.2.1 Kinematic equations

The translational and angular velocities of the object $(\boldsymbol{v}, \boldsymbol{\omega}) \in \mathbb{R}^6$ are given in body-fixed frame as follows:

$$\dot{R} = R\hat{\boldsymbol{\omega}}, \qquad \dot{\boldsymbol{x}} = R\boldsymbol{v} \tag{4.2}$$

where the operator $\hat{}: \mathbb{R}^3 \to so(3)$ is defined as $\boldsymbol{s}\hat{\boldsymbol{\omega}} = \boldsymbol{s} \times \boldsymbol{\omega}, \forall \boldsymbol{s} \in \mathbb{R}^3$, where the space $so(3)$ is the Lie algebra of the Lie group $SO(3)$. $\hat{\omega}$ is defined as

$$\begin{pmatrix} 0 & -\omega(3) & \omega(2) \\ \omega(3) & 0 & -\omega(1) \\ -\omega(2) & \omega(1) & 0 \end{pmatrix} \tag{4.3}$$

where $\omega(n)$ is the n-th element of angular velocity $\boldsymbol{\omega}$.

4.2.2 Dynamic equations

The dynamics of a rigid body immersed in a viscous fluid results from the coupling between the body and the surrounding flow. The dynamical effects from the interaction of the fluid to a body displacement, including both translational and rotational transformations, are described as added-mass tensors M_f and J_f. M_f represents the force and torque due to the fluid coupling to a translational acceleration of the body and J_f is to a rotational acceleration. Therefore, the body dynamics is governed by the generalized Kirchhoff equations [FERM08]. The dynamic equation has the following form in body-fixed frame.

$$\begin{aligned} M \cdot \dot{\boldsymbol{v}} + \boldsymbol{v} \times (M \cdot \boldsymbol{\omega}) &= \boldsymbol{F}_t + \boldsymbol{F}_g \\ J \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (J \cdot \boldsymbol{\omega}) + \boldsymbol{v} \times (M_f \cdot \boldsymbol{v}) &= \boldsymbol{\Gamma}_t + \boldsymbol{\Gamma}_g \end{aligned} \tag{4.4}$$

where $M = mI + M_f$, $J = J_0 + J_f$, J_0 is the moment of inertia of the body and I is the 3×3 identity matrix. $\boldsymbol{F}_t$ and $\boldsymbol{\Gamma}_t$ are the resulted force and torque due to the turbulence generated at the body surface while the body moves in a viscous flow; $\boldsymbol{F}_g$ and $\boldsymbol{\Gamma}_g$ result

Table 4.1: Notation used through this chapter (Bold letters denote vector variables.).

symbol	description	symbol	description
R	rotation matrix of the body	M_f	added mass tensor
$\boldsymbol{x}$	position of body	M	mass tensor
$\boldsymbol{\omega}$	angular velocity	J_f	added inertia tensor
$\boldsymbol{v}$	translational velocity	J	inertia tensor
m	mass of body	J_0	moment of inertia
$\boldsymbol{r}$	center of buoyancy	U_0	mean falling velocity of body
$\boldsymbol{g}$	gravitational acceleration	k	turbulent kinetic energy
V	volume of body	ϵ	dissipation rate of turbulent energy
ρ_b	body density	χ	turbulent frequency $\chi = \varepsilon/k$
ρ_f	fluid density	$\boldsymbol{u}$	fluid velocity
ν	kinematic viscosity of fluid	$\langle \boldsymbol{u} \rangle$	mean flow velocity
Re	Reynolds number	α	relaxation rate coefficient
$\boldsymbol{F}_t$	force due to turbulence	β	diffusion coefficient
$\boldsymbol{F}_g$	force due to gravity	$\boldsymbol{W}$	Wiener process in $\mathbb{R}^3$
$\boldsymbol{\Gamma}_t$	torque due to turbulence	$\boldsymbol{\xi}$	normal Gaussian distributed variable
$\boldsymbol{\Gamma}_g$	torque due to gravity	Δt	time step of simulation

from the buoyancy-corrected gravity.

Because the added-mass tensors are only determined by the body's geometry and independent of the generated turbulence at body surface [How95], M_f and J_f can be computed in a precomputation step similar to the implementations of the mass tensor and rotational inertia tensor in [WP12].

The gravity and buoyancy act on the immersed rigid body with inverse directions. This work expresses them in body-fixed frame as follows:

$$\boldsymbol{F}_g = R^T(m - \rho_f V)\boldsymbol{g} \tag{4.5}$$

$$\boldsymbol{\Gamma}_g = \rho_f V \boldsymbol{r} \times R^T \boldsymbol{g} \tag{4.6}$$

where $V = m/\rho_b$ is the volume of the body and $\boldsymbol{r}$ is the vector from the center of mass to the center of buoyancy in body-fixed frame.

The challenge of solving Equation (4.4) is how to determine the force f_t and torque τ_t due to viscous effect of surrounding turbulent flow, which includes the drag and lift dynamics. Therefore, f_t has the following formation depending on aerodynamics.

$$f_t = f_{drag} + f_{lift} = -\frac{1}{2}\rho_f A|U|(C_d U + C_l U \times \frac{N \times U}{|N \times U|}) \tag{4.7}$$

where A is the surface area of the body, and U is an intermediate translational velocity in Section 4.3.2. C_d and C_l are the drag and lift coefficients, which can be given from experiments or advanced analysis. Here these coefficients are set to be user control parameters so that different dynamics of immersed rigid-body can be simulated. The torque due to drag and lift forces is $\tau_t = \vec{p} \times f_t$ where $\vec{p}$ is the vector from the center of mass to the center of pressure of the body.

4.3 Stochastic model

The stochastic model for the motion of suspended fluid particles is proposed by Langevin decades ago. The velocity increments of a particle in continuous time steps are in highly correlated process, which is called the Ornstein-Unlebeck process [UO30]. The model can be applied to describe the Brownian motion of lightweight objects undergoing the vortical loads from the surrounding turbulent flow [MD04, YCZ11].

For a statistically isotopic turbulence, the Langevin equation can be defined as the following stochastic differential equation:

$$d\boldsymbol{u}(t) = -\alpha \boldsymbol{u}(t)dt + \beta d\boldsymbol{W} \tag{4.8}$$

where $\boldsymbol{u}(t)$ is the translational velocity of a fluid particle; α and β are the relaxation rate and the diffusion coefficient, which reveal the properties of the turbulent flow; and $\boldsymbol{W}$ is a Wiener process which represents a Brownian motion with a continuous-time stochastic process. In this implementation, the process is calculated by a normal distribution with mean of zero and variance of the time interval Δt. Note that the numerical analysis of the stochastic process is out of the scope of this work, such as Ito calculus.

For a fluid particle with arbitrary shape, the relaxation term in Equation (4.8) has no effect to angular velocity increments of the body as a rotational Brownian motion [MD04]. The Langevin equation for angular velocity is given as:

$$d\boldsymbol{\omega}(t) = \beta d\boldsymbol{W} \tag{4.9}$$

4.3.1 Generalized Langevin equation

Pope [Pop83] described the generalized Langevin equation for the suspended particle in a turbulent flow. The equation gives the expressions of α and β having the following forms:

$$\alpha = (\frac{1}{2} + \frac{3}{4}C_0)\frac{\varepsilon}{k}, \qquad \beta = (C_0\varepsilon)^{\frac{1}{2}} \tag{4.10}$$

where k and ε are kinetic energy and its dissipation rate of the surrounding turbulent flow; C_0 is a Kolmogorov coefficient. According to the Kolmogorov hypothesis, C_0 is related to the Reynolds number Re of the flow [Pop11].

$$C_0(Re) = 6.5(1 + 140Re^{-\frac{4}{3}})^{-\frac{3}{4}} \tag{4.11}$$

For high-Reynolds-number flow ($Re > 10^3$), this relation is empirically fitted.

Finally, the dynamic equations of immersed body are discretized through finite-difference scheme by substituting Equations (4.8)(4.14)(4.10) into Equation (4.4).

$$\begin{aligned}\boldsymbol{v}(t+\Delta t) - \boldsymbol{v}(t) = M^{-1}(&-\boldsymbol{v}(t) \times (M\boldsymbol{\omega}(t))\Delta t \\ &-\chi(\tfrac{1}{2} + \tfrac{3}{4}C_0)\boldsymbol{v}(t)\Delta t \\ &+(C_0\varepsilon\Delta t)^{\frac{1}{2}}\boldsymbol{\xi}_1 + \boldsymbol{F}_g(t)\Delta t)\end{aligned} \tag{4.12}$$

$$\begin{aligned}\boldsymbol{\omega}(t+\Delta t) - \boldsymbol{\omega}(t) = J^{-1}(&-\boldsymbol{\omega}(t) \times (J\boldsymbol{\omega}(t))\Delta t \\ &-v(t) \times (M_f\boldsymbol{v}(t))\Delta t \\ &+(C_0\varepsilon\Delta t)^{\frac{1}{2}}\boldsymbol{\xi}_2 + \boldsymbol{\Gamma}_g(t)\Delta t)\end{aligned} \tag{4.13}$$

4.3.2 Time integration

This work proposes a fractional-step method for solving the velocity $u = (\omega, v) \in R^6$ from Equation (4.4) which includes three steps as follows:

$$u \xrightarrow{(f_g,\tau_g)} \dot{u} \xrightarrow{Langevin} \ddot{u} \xrightarrow{(f_t,\tau_t)} u^{new}$$

The first step is to numerically solve for a guess velocity $\dot{u}$ while not considering the vortical loads in Equation (4.4). Then, this work utilizes the Langevin equations of both translational and rotational velocities to obtain an intermediate velocity $\ddot{u}$.

$$\ddot{u} = \dot{u} - \begin{pmatrix} -\sqrt{C_k\varepsilon\Delta t}\vec{\xi_1} \\ \chi(\frac{1}{2} + \frac{3}{4}C_k)\dot{v}\Delta t - \sqrt{C_k\varepsilon\Delta t}\vec{\xi_2} \end{pmatrix} \tag{4.14}$$

Where $\chi = \varepsilon/k$, $\vec{\xi_1}$ and $\vec{\xi_2}$ are the vectors of normal Gaussian distributed variables $Norm(0,1)$ with mean zero and unit deviation. The vectors are generated using the Box-Muller algorithm [PTVF92].

After calculating the vortical loads (f_t, τ_t) by Equation (4.7) using $U = \ddot{v} \in \ddot{u}$, the last step is to solve Equation (4.4) to obtain u^{new} for next time step. The parameters (χ, ε) measure the characteristics of the surrounding turbulent flow. This work pre-generates these parameters $(\chi(t), \varepsilon(t))$ by two-equation k-ε turbulent model in the next section.

4.4 Turbulence model

In a turbulent flow, the fluid velocity $\boldsymbol{u}$ can be represented by Reynolds decomposition with the mean flow $\langle \boldsymbol{u} \rangle$ and fluctuating velocity $\boldsymbol{u}'$ ($\boldsymbol{u} = \langle \boldsymbol{u} \rangle + \boldsymbol{u}'$). The common approach for solving the fluid-rigid coupling problem based on the three dimensional Navier-Stokes equations are extremely computationally expensive, because the fluctuations of turbulence would be of small scale and high frequency. It is obvious not suitable for an interactive application. The most widely used turbulence model is the k-ε turbulent model proposed in [LS74], which requires low computational cost. The k-ε model is a semi-empirical model based on the transport equations, which consist of two coupled equations of the turbulent kinetic energy k and its dissipation rate ε. This energy transport equations are defined as follows:

$$\begin{aligned} D_t k &= \nabla \cdot ((\nu + \tfrac{v_T}{\sigma_k})\nabla k) + G - \varepsilon \\ D_t \varepsilon &= \nabla \cdot ((\nu + \tfrac{v_T}{\sigma_\varepsilon})\nabla \varepsilon) + \chi(C_1 G - C_2 \varepsilon) \end{aligned} \tag{4.15}$$

where D_t denotes a Lagrangian derivative; σ_k and σ_ε are the turbulent Prandtl numbers for k and ε; C_1 and C_2 are empirical constants. The values of these parameters are given empirically as: $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$, $C_1 = 1.44$ and $C_2 = 1.92$ [LS74].

The turbulent viscosity v_T in Equation (4.15) describes the small scale turbulent motion as a viscous diffusion scale in the turbulent model. Turbulent viscosity v_T is defined as:

$$v_T = C_\mu \frac{k^2}{\varepsilon} \tag{4.16}$$

where $C_\mu = 0.09$ is an empirical constant.

The term G in Equation (4.15) represents the generation of turbulent kinetic energy due to the mean velocity gradients and can be defined in terms of the strain tensor of the flow:

$$G = 2v_T \sum_{ij} S_{ij}^2 \tag{4.17}$$

where $S_{ij} = \frac{1}{2}(\frac{\partial \langle u \rangle_i}{\partial x_j} + \frac{\partial \langle u \rangle_j}{\partial x_i})$.

In the implementation of this model, Equation (4.15) is simplified by avoiding the calculations of the incorporated diffusion terms, which are proven to be visually unnecessary in previous work [PTC+10]. The transport equations have the following simplified formulations:

$$D_t k = G - \varepsilon \tag{4.18}$$

$$D_t \varepsilon = \chi(C_1 G - C_2 \varepsilon) \tag{4.19}$$

In cases of high turbulent flows with high Reynolds numbers, the initial state (k_0, ε_0) is defined in terms of the mean falling velocity U_0 as described in Equation (4.1) to estimate the information about the history of the moving body. The initial conditions for energy transport equations are given as follows:

$$k_0 = \frac{3}{2}U_0^2; \qquad \varepsilon_0 = C_\mu^{\frac{3}{4}} k_0^{\frac{3}{2}} l^{-1} \tag{4.20}$$

where l is the length scale of the MAC grid cell in the mean flow simulation.

The turbulent parameters (χ, ε) are explicitly solved with finite difference scheme from Equations (4.17)(4.18)(4.19) as shown in Figure 4.2, where a standard fluid solver is applied to obtain the mean velocities $\langle \boldsymbol{u} \rangle$ of the base flow. According to the Kolmogorov theory, for high Reynolds number, the initial turbulence is unstable and the kinetic energy is divided into smaller scales. After reaching a critical scale value, turbulent energy dissipates due to viscosity, creating an energy cascade [PTSG09]. Figure 4.2 shows the varying dissipate rate accompanying the kinetic energy in the calculated result.

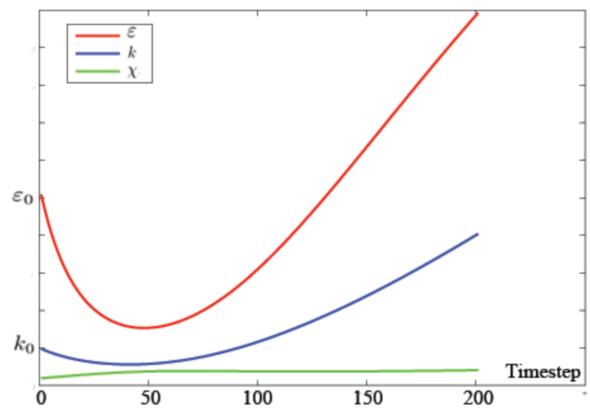


Figure 4.2: Turbulent parameters (χ, ε) in time steps with $Re = 3.8 \times 10^3$ and $32 \times 32 \times 16$ MAC grids.

4.5 Implementation

The implementation of this model consists of two computation stages: pre-computations of added-mass tensors and turbulent flow; and runtime simulation of a rigid body solver as shown in Figure 4.1.

Turbulent flow The turbulent model is based on a standard fluid solver to resolve the mean flow around the body. However, the complicated solver of Reynolds-averaged Navier-Stokes equations is usually applied to the k-ε model for accurate solutions, the standard solver can be more visually plausible and efficient in computer graphics [PTSG09].

This work utilizes a typical MAC staggered grid with semi-Langragian advection to obtain the base mean flow as described in Algorithm 1, which is similar to previous work [CZY11, PTSG09]. The turbulent energy k and its dissipation rate ε are computed at each grid node. This model considered the viscous effect from the surrounding flow by Lagrangian tracking a passive particle with the same position of rigid body in the fluid field, i.e. the implementation of boundary conditions of rigid body is not necessary, and the inflow velocity is chosen as the mean falling velocity of Eq.(4.1) and defined as:

$$U_{in} = U_0 \tag{4.21}$$

Although the implementation of the mean flow and turbulent energy could be in real-time by GPU for low-resolution, this computation should be executed offline because: 1) it only need to be computed once; 2) the solver should guarantee the runtime computation of rigid body solver in real-time; 3) Other turbulent solver or high-resolution simulations are also acceptable in the framework.

Algorithm 1 Pseudo-code for pre-generated turbulent model.

Boundary conditions $\leftarrow$Equations (4.20)(4.21)
Timestep $t = 0$
while not stopped **do**
    // Mean flow $\langle u \rangle$
    Convection by semi-Lagrangian
    Pressure projection by Poisson solver

    // Energy transport
    Get turbulent viscosity v_T $\leftarrow$Equation (4.16)
    Get strain tensor term G $\leftarrow$Equation (4.17)
    Integrate turbulent energy k $\leftarrow$Equation (5.23)
    Integrate dissipation rate ε $\leftarrow$Equation (4.19)

    $t = t + \Delta t$
end while
Output: (χ, ε)

Rigid body solver The proposed stochastic model is relatively efficient for real-time simulations, because the computation burdens involving turbulent flow effects are executed in pre-computation steps. The most runtime computation is for the rigid body solver, which is described in Algorithm 2. This work adopts a standard Runge-Kutta scheme for resolving the coupling dynamic equations, Equation (4.12) and Equation (4.13). In the work of [KCD09], a lie group integrator of Euclidean motions is shown to be more robust than the Runge-Kutta scheme for large timesteps. Because this work focuses on the falling motion of immersed rigid bodies, a quite small scale of timestep is used for the rigid body solver. The Runge-Kutta scheme is efficient enough for the simulations in the proposed model.

Algorithm 2 Pseudo-code for the runtime computation.

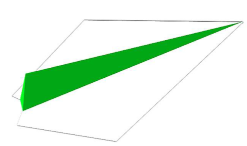
```
Precomputation of added-mass tensors
Initialization of rigid body
Timestep t = 0
while not arrive ground do
    Calculate gravity force←Equation (4.5)
    Query χ_t and ε_t (Algorithm 1)
    Update translational velocity v ←Equation (4.12)
    Update angular velocity ω ←Equation (4.13)
    Integrate (R, x) ←Equation (4.2)
    Render data
    t = t + Δt
end while
```

4.6 Results

This section describes the simulation results using the proposed stochastic model.

A piece of paper released in air with different release angles is simulated by the proposed model as shown in Figure 4.3. The cross section of the leaf model used in the simulation is elliptical (semi-major axis and minor axis are 4.0 and 1.0 cm respectively). The thickness is set to be 0.01 cm and the density number is 0.8. The Reynolds number (4.3×10^4) is so large that the turbulence can be generated at the paper surface. The paper falls down following a helical trajectory in Figure 4.3(a) and a side-to-side tumbling motion in Figure 4.3(b) which are in compliance with the analysis result in [XM14c]. The trajectories have the secondary motions that the paper rotates around the major-axis while falling, which usually happens in reality. Note that we can not understand the simulation results in prior by the proposed approach while the initial conditions are modified, e.g. release angle.

Figure 4.4 shows a comparison between the simulation result and a video of a flying paper airplane. The paper airplane is made by a $8.3 \times 8.3 \times 0.01$ (cm) piece of paper. The added-mass tensors and moment of inertia of the body depend on the geometries with closed shape, where the fold part of the paper airplane is constructed as volume as shown in the right figure. The simulation begins with an initial velocity of 20 cm/s in the horizontal direction, and the simulated result shows two turning motions (turning front and turning sideways) which are caused by the surrounding airflow. The



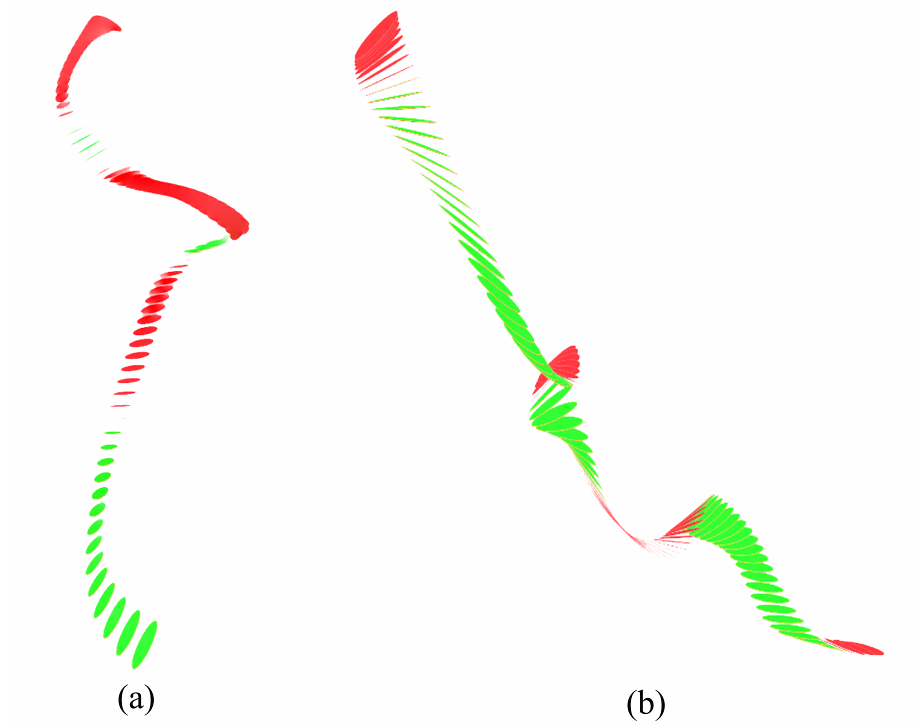


Figure 4.3: Simulation results of a piece of paper falling in air. (a) initial release angle = 75°; (b) initial release angle = 30°.

turning motions are similar to the observation from ground truth in Figure 4.4.

Figure 4.5 shows the discrete frames from the animation of a rubber ellipsoid falling in water, and the time interval is 50 ms. The rubber ellipsoid with semi-principal axes of length 1 cm, 2 cm and 4 cm falls down in a quiescent water flow. A small scale of fluttering motion can be found in Figure 4.5(a) using the simulation method of previous work [WP12]. In contrast to this previous work, the coupling between forces and torques due to the surrounding turbulent flow can be indicated properly using the proposed model. The oscillations of rigid body in differential directions from the video of falling experiment (Figure 4.5(c)) are successfully captured by the proposed method.

The precomputation time of added-mass tensors depends on the amount of the body meshes; and the precomputation time of turbulent model depends on the grid solutions of the base flow. In the case of 1280 meshes and $32 \times 32 \times 8$ MAC grid, the precomputation times are 53 ms and 182 ms, respectively. All simulations were implemented on an Intel Core i7 CPU with 3.20 GHz and 12.0 GB RAM. The simulation time for a single loop of runtime computation is not more than 2.0 ms. As shown in Table 4.2, the runtime computation time is independent of the body triangular meshes and time step, and it is suitable for real-time simulations.

Figure 4.4: Comparison between the simulation and the ground truth of a flying paper airplane.

4.7 Discussion

This chapter presented the stochastic model for realistic simulations of rigid bodies in viscous, high-Reynolds-number flows. The main strength of the this stochastic method lies in combining Kirchhoff equations and Langevin model to represent the chaotic motions of immersed rigid bodies. The method allows a real-time simulation for interactive applications, such as virtual reality, online or mobile graphical applications.

Limitations exist in the proposed approach. Because the simulation results are sensitive to the initial conditions as discussed in Section 4.6, including release angle, velocities etc., the appropriate variables should be chosen to meet the ground truth in the simulation results. Some characteristic motions like fluttering and tumbling motions, are not apparently captured by the approximated turbulent model.

To solve these issues, this thesis proposes a new approach combining the proposed stochastic with motion synthesis of immersed rigid body dynamics in next chapter.

Table 4.2: Computation cost of the simulation results on runtime.

bodies	meshes	timestep	avg cost
ellipsoid 1	320	1 ms	1.59 ms
ellipsoid 1	320	5 ms	1.63 ms
ellipsoid 1	320	10 ms	1.65 ms
ellipsoid 2	1280	5 ms	1.71 ms
piece of paper	1024	5 ms	1.73 ms
paper airplane	288	5 ms	1.86 ms

(a) (b) (c)

Figure 4.5: Comparison among (a) ground truth,(b) previous work and (c) the proposed approach. Ground truth shows oscillations generated in different directions (the shorter silhouettes manifest the oscillation in three dimensions.), previous work takes account only Kirchhoff tensor, whereas the proposed model has concerned the vortical loads.

Chapter 5

Pattern-driven immersed rigid body dynamics

This chapter introduces novel pattern-driven techniques for immersed rigid-body simulations which concern the unsteady dynamics of rigid body fully immersed or submerged in a still flow. For real flow situations, an integrated representation of immersed rigid body dynamics that considers aerodynamic properties of inertial effects, viscous effects, and the influence of turbulence from the surrounding flow is presented. Numerical experiments in parameter space based on a parametric dynamical model are conducted. The experimental system is more efficient and simpler than two-way coupling simulations. The motion patterns and corresponding parameter subspaces are classified. A curvature-based motion synthesis based on a motion graph and turbulent energy optimization is developed to determine instantaneous force coefficients in unsteady dynamics of immersed rigid body dynamics. The proposed approach achieves efficient and life-like immersed rigid body simulation results, and these results are relevant to the animation of strongly coupled objects and flows.

Note that the coefficients of drag and lift forces play an important role in aerodynamics and hydrodynamics simulations. In previous work, these coefficients have been considered heuristic constants [HLYK08, SJ13], functions of a Reynold number [MMS09, NO13] and functions of an angle of attack [WP03, OFM09, JWL$^+$13]. In this work, such coefficients are handled as functions of both angle of attack and Reynolds number to represent the unsteady dynamics of an immersed rigid body. A similar idea is the parameter-fitting method [UKSI14] to determine the relationship between coefficients and these two factors. However, it is impossible for animators to determine proper coefficients for chaotic motions with different motion patterns; therefore, parameter subspaces are classified and utilized in a motion synthesis approach to construct a user-friendly framework in this work.

5.1 Introduction

Plausible simulation of immersed rigid bodies moving through still fluid flows requires a combination of quantitative experiments and numerical models that characterize the state of the passive dynamics of the rigid bodies. This subject has attracted researchers' attention for over one hundred years, because of the intricate interaction between the motion of rigid bodies and the induced fluid reaction, and it remains an open issue in computer graphics. Our goal is to develop pattern-driven simulation techniques that are well suited for immersed rigid bodies. Examples of such bodies are abundant in everyday phenomena and computer animations, e.g., the motions of leaves, snowflakes, confetti, and petals.

This work introduces immersed rigid body dynamics to represent the motion of a rigid body fully immersed or submerged under real flow of water or air, that is strongly coupled with the surrounding flow. There appears to be a paradox for the simulation of the immersed rigid body dynamics in computer graphics, i.e., fluid effects on the body cannot be understood without solving the turbulent motions at a similar timescale with the body, and fluid motion is trivial when rendering a simulation scene in a graphical application. In order to solve this problem and efficiently simulate such dynamics, we propose a new solution that avoids computation of fluid motion while considering the fluid effects. The proposed simulation technique of immersed rigid body dynamics is significant and can achieve realistic motions in animations, especially for thin, sharp-edged, or wing structure bodies. In addition, it is a promising and fundamental topic for immersed body simulations involving cloth and character simulations such as the locomotion of swaying cloth, flying birds, and swimming fish.

This work presents a unique representation of immersed rigid body dynamics that considers inertial and viscous flow effects. For inertial effects in an inviscid flow, we propose a proxy geometry approximation method of thin and sharp-edged bodies to compute added-mass tensors to alleviate the limitations of the previous numerical methods. For viscous effects in a viscous flow, we decompose the total viscous forces into three components, i.e., drag, translational lift, and rotational lift forces with instantaneous coefficients as the function of angle of attack and Reynolds number, which make unsteady dynamics possible in our simulation. Based on our dynamical model, we herein present an initial analysis of the three-dimensional (3D) motion patterns and their parameter subspaces for an individual body.

This work presents a motion synthesis technique that reflects the influences of turbulent effects in high Reynolds number conditions. We propose a curvature-based motion synthesis based on a specified motion graph to represent motion transitions among motion patterns, which are found in our experiments of falling rigid bodies. We also constructed

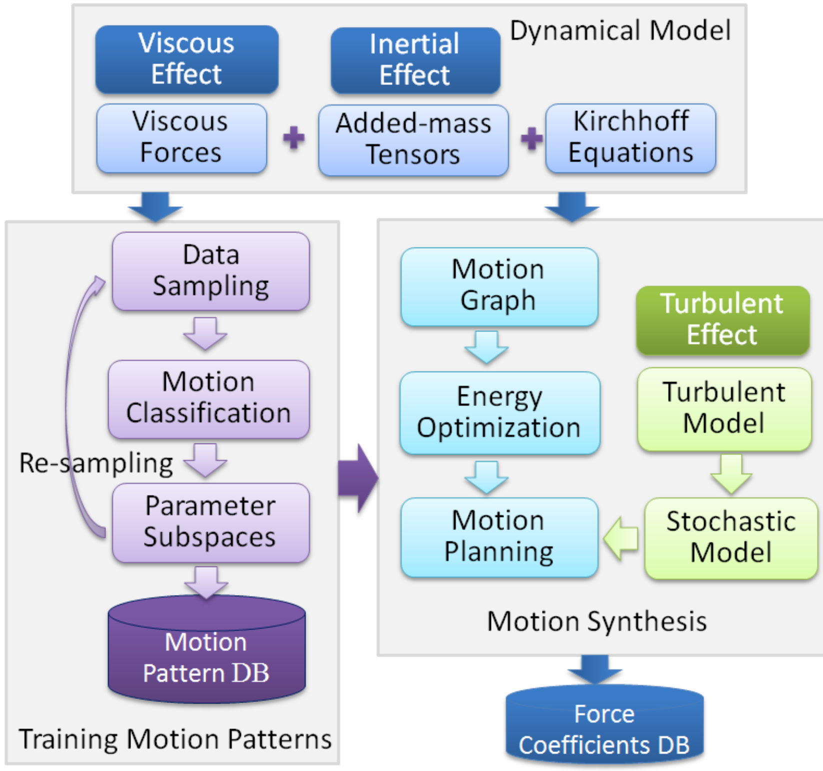


Figure 5.1: The proposed pattern-driven framework consists of a dynamical model, data training of motion patterns, and motion synthesis of immersed rigid body dynamics.

an optimization of turbulent kinetic energy to determine the instantaneous coefficients implicitly related to Reynolds numbers.

In summary, the contributions of this work include: (1) a dynamical model with instantaneous coefficients that describes the dynamics of an immersed rigid body in a real flow, (2) a pattern-driven framework to reveal parameter subspaces of typical motion patterns, and (3) a turbulent-energy optimization-based motion synthesis solver.

5.2 System overview

The proposed pattern-driven framework consists of three main components (Figure 5.1): a dynamical model, data training of motion patterns, and motion synthesis. Note that a force coefficients database that reflects motion transitions among different motion patterns of immersed rigid body dynamics is also created by the proposed framework. This database can be adopted in runtime simulations by our dynamical model to meet real-time simulations in interactive computational environments.

5.3 Dynamical model

The motion of an immersed rigid body is represented in Euclidean group SE(3) with $(R(t), x(t))$, where $R(t)$ and $x(t)$ are the orientation matrix and position of the body's center of mass (COM) in the world frame, respectively. The generalized vector is defined as $X = (u, \omega) \in \mathbb{R}^6$, with translational velocity u and angular velocity ω in a body-fixed frame. The kinematic equations of the immersed rigid body are given as:

$$\begin{pmatrix} \dot{R} \\ \dot{x} \end{pmatrix} = \begin{pmatrix} R\hat{\omega} \\ Ru \end{pmatrix} \tag{5.1}$$

where $\hat{\omega}$ is the skew matrix of ω. In order to obtain the dynamic equations of the immersed rigid-body in real flow, this work expands the situation from inviscid flow to clarify the fluid effects generated by the surrounding flow.

5.3.1 Inertial effect

Assuming a rigid body B with mass m moving in an inviscid flow domain Ω with fluid density ρ_f at rest at infinity, the flow velocity field v has a potential field ϕ in flow domain due to the irrotational flow condition, and ϕ satisfies the following Laplace equation with no-slip boundary conditions:

$$\Delta\phi(z) = 0 \quad z \in \Omega \tag{5.2}$$

$$\nabla\phi(z) \cdot n = u_n(z) \quad z \in \partial B \tag{5.3}$$

$$\phi(z) = 0 \quad \|z\| \to \infty \tag{5.4}$$

where n is the surface normal vector and $u_n(z)$ is the normal velocity of the rigid body at the surface point. The fundamental solution of the Laplace equation is given as

$$G(x, y) = \frac{1}{4\pi r} \tag{5.5}$$

where $r = \|x - y\|$ denotes the distance from a source point x to a collocation point y on the body surface. Then, the solution of Laplace equation can be given in the following formulation as boundary integral equation:

$$\phi(x) = 2 \int_{\partial B} [G(x, y)\frac{\partial \phi(y)}{\partial n} - \frac{\partial G(x, y)}{\partial n_y}\phi(y)]ds(y) \tag{5.6}$$

for $\forall x \in \partial B$. This boundary integral equation can be computed by boundary element method [HS67, WP12], which subdivides the body surface into various flat panels with a uniform source strength distribution. Then, the kinetic energy of the surrounding flow is expressed as follows:

$$E = \frac{1}{2}u \cdot (Mu) + \frac{1}{2}\omega \cdot (J\omega) \tag{5.7}$$

where M and J are the second-order tensors of the added mass and the added moment of inertia of flow due to the translational and rotational motions of a body, respectively. The value of these tensors are determined from the velocity potential value $\phi_i, 1 \leq i \leq 6$ in the following formulation:

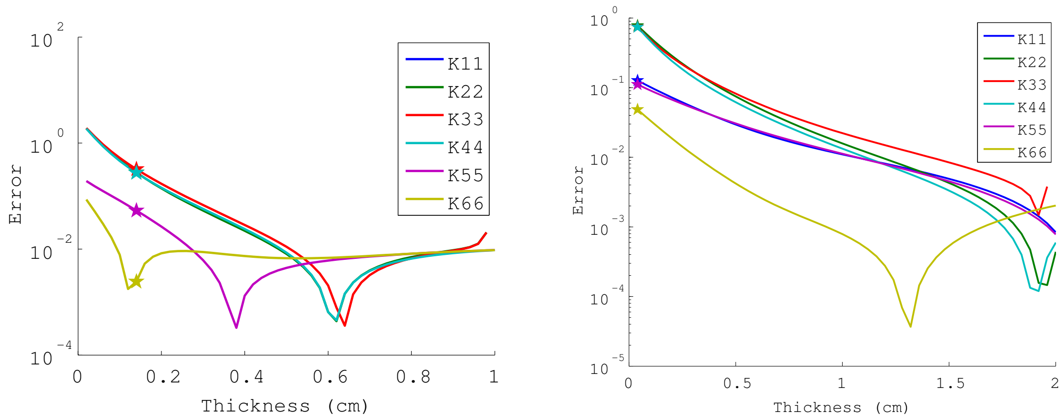
$$K_{ij} = \rho_f \int_{\partial B} \phi_i \frac{\partial \phi_j}{\partial n} ds \tag{5.8}$$

Note that K is symmetric in terms of Green's theorem, i.e., $K_{ij} = K_{ji}$. M and J are corresponding to the value of K when $1 \leq i, j \leq 3$ and $4 \leq i, j \leq 6$, respectively. According to the expression of Equation (5.8), the added tensors do not depend on the velocities of the body, and they are functions of body's geometry. Therefore, the implementation of added mass tensors can be executed in an offline process. For more details about the added mass tensors, please refer to [New77].

In the computation of added tensors, there are two limitations of previous approaches [WP12]: (1) lack of efficiency for dense meshes that the numerical solution requires $O(N^3)$ operations where N is the number of boundary elements, i.e., the number of triangular meshes; (2) lack of accuracy for thin structures that the integral equation (5.6) becomes nearly singular when the distance r between the source point and the collocation point are very small. In engineering literature, a fast multipole accelerated method [Nis02] can reduce the computation operations to $O(N)$, and a sinh transformation [JJE13] can weaken near singularity. In this work, we adopt an analytic solution of Laplace equation using proxy geometry approximation for simplicity. As shown in Figure 5.2, the relative error $\|K - K_a\|/\|K_a\|$ (K: solution by the previous approach; K_a: analytic solution) of traditional boundary element method becomes inaccurate when the body is very thin (smaller than 0.6 cm in the figure). The thickness of a leaf is around 0.03 cm and the relative computation error is close to 90% as shown in Figure 5.2 (b).

A rigid body exhibiting sensitive viscous effects from the surrounding flow has the following features: low body-to-fluid density ratio and airfoil structure with a large frontal area, such as leaves, pieces of paper, and so on. In terms of these features, a proxy geometry as ellipsoid is efficient and sufficient for thin-structure rigid body in this work. For complex models, this work suggests to adopt a sinh transformation.

Proxy geometry approximation. To obtain the axes of an input model, the algorithm is similar to a bounding ellipsoid construction method. For N mesh vertices P_i , $\max\limits_{1 \leq i \leq N} \{P_i \cdot r_j\} - \min\limits_{1 \leq i \leq N} \{P_i \cdot r_j\}$ denote three axes (a, b, c) of the proxy geometry where vectors r_j are the natural axes of the set of vertices P_i ($j \in [1, 3]$, and their corresponding eigenvalues are given as $\lambda_1 \geq \lambda_2 \geq \lambda_3$). Considering that a model might not happen to be an ellipsoid, the proxy geometry of the model is a mixture of the body and its surrounding flow, and the average density $\bar{\rho}$ is given by $(\rho_r * V_r + \rho_f * V_f)/(V_r + V_f)$, where V_r and V_f are the volumes of the body and fluid in the proxy geometry, respectively. Here, ρ_r is the body



(a) cylinder coin (thickness = 0.15 cm) (b) ellipsoid leaf (thickness = 0.03 cm)

Figure 5.2: Computation errors between previous approach and the analytic solution for different bodies. For both cases , the relative error becomes larger with smaller thickness. The previous approach fails (error > 1.0%) when the thickness is smaller than 1.3 cm for ellipsoid and cylinder structures.

density. $\bar{\rho}$ is used for calculating gravitational force.

Analytic added tensors. Only for few regular geometries, the analytic solutions of the Laplace equation exist. In terms of the three axes of the approximated ellipsoid a, b and c, the translational kinetic energy of the surrounding flow along a-axis is given as follows:

$$E_a = \frac{1}{2} m_f \frac{a_0 u^2}{2 - a_0} \tag{5.9}$$

where $m_f = \frac{4}{3}\rho_f \pi abc$ is the fluid mass occupied by an immersed body. a_0 is defined by the following formulation in elliptic integrals:

$$a_0 \equiv abc \int_0^\infty \frac{d\lambda}{(a^2 + \lambda)\sigma} \tag{5.10}$$

where $\sigma = \sqrt{(a^2 + \lambda)(b^2 + \lambda)(c^2 + \lambda)}$. Then, the rotational kinetic energy is given as follows:

$$E_a^r = \frac{1}{2} m_f \frac{(b^2 - c^2)^2 (c_0 - b_0)\omega^2}{10(b^2 - c^2) - 5(b^2 + c^2)^2 (c_0 - b_0)} \tag{5.11}$$

where b_0 and c_0 are defined in the same form with a_0. Because of the symmetrical expressions for $b-$ and $c-$axis, E_b, E_b^r, E_c and E_c^r are calculated in the same procedure. From the kinetic energy of the surrounding flow in different directions, M and J are obtained directly, which are explicitly diagonal matrix for an ellipsoid. The analytic solutions of added tensors are adaptive to the different structures, such as, cylinder, spheroid, sphere, and the other structures related to the aforementioned geometries.

Equations of motion. The fluid effect in an inviscid flow is inertial, and the equation of motion is given as Kirchhoff equation from the Kirchhoff's theory [Lam75] which describes the linear and angular dynamics for a rigid-fluid system.

$$
\begin{aligned}
M_a \frac{du}{dt} &= (M_a u) \times \omega && (5.12)\\
I_a \frac{d\omega}{dt} &= (I_a \omega) \times \omega + (M u) \times u && (5.13)
\end{aligned}
$$

where $M_a = mE + M$ and E is a unit tensor. $I_a = J + I$, and I is the moment of inertia of the body. For an ellipsoid structure, $I = diag(\frac{1}{5}m(b^2 + c^2), \frac{1}{5}m(a^2 + c^2), \frac{1}{5}m(a^2 + b^2))$.

5.3.2 Viscous effect

In a real situations the surrounding flow is commonly viscous, and the vortex shedding happens around the body surface. The model for the motion in an inviscid flow lacks of two significant ingredients in real situation: viscous forces and vorticity. The fluid effects on a immersed rigid body have the following contributions: the inertial effect force F_i that can be accurately calculated, the viscous effect force F_v due to the flow viscosity, and the turbulence effect related to the vortex shedding [BH06], which we will consider in motion synthesis steps. The added-mass contributions do not change by the rotational effects of viscous flow [MM02, ERFM12], i.e., the contributions are the same as in an inviscid flow. Therefore, the Kirchhoff equation is generalized as follows:

$$
\frac{dX}{dt} = F_i(X) + F_v(X) \tag{5.14}
$$

where $X = (u, \omega)$ generalized vector. F_i and F_v are the contributions in the inviscid and viscous cases, respectively.

Viscous forces. The viscous fluid effect cannot be expressed in quasi-steady assumptions based on the Kutta-Joukowski lift theorem. From the recent numerical analysis [Umb05, VBL09] and the experimental observations [VCW13, HLW+13], the viscous force has following three components: drag F_D, rotational lift F_{L1}, and translational lift F_{L2} forces as illustrated in Figure 5.3.

$$
\begin{pmatrix} F_D \\ F_{L1} \\ F_{L2} \end{pmatrix} = \frac{1}{2}\rho_f \|u\|^2 A \begin{pmatrix} -C_d e_1 \\ C_{l1} e_2 \\ C_{l2} e_3 \end{pmatrix} \tag{5.15}
$$

where A is the frontal area of the body with an approximation of $ab\pi$. (e_1, e_2, e_3) is a local coordinate frame which is similar to a Frenet reference frame [ZGB+11] and defined by the directions of linear velocity $\vec{u}$ and angular velocity $\vec{\omega}$.

$$
(e_1, e_2, e_3) = (\vec{u}, \vec{\omega} \times \vec{u}, (\vec{\omega} \times \vec{u}) \times \vec{u}) \tag{5.16}
$$

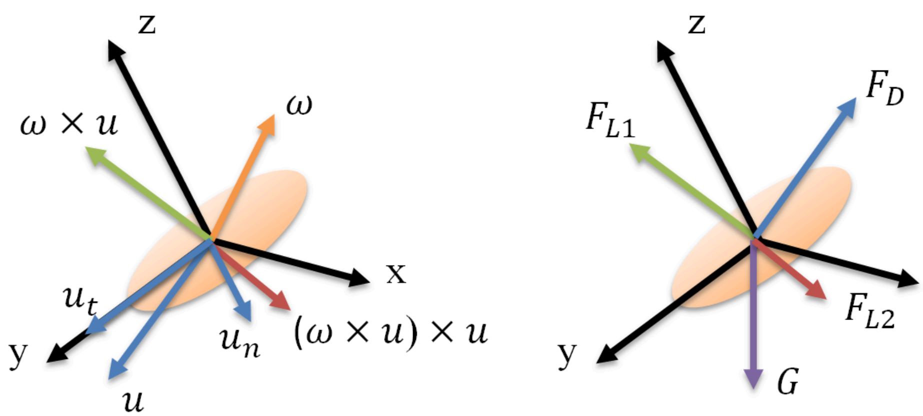


Figure 5.3: The illustration of the drag, rotational lift, and translational lift forces in body-fixed frame, which are corresponding with the translational velocity (blue) and angular velocity (yellow) and their cross products in the left subfigure.

Instantaneous coefficients. The fore coefficients $C = \{C_d, C_{l1}, C_{l2}\}$ are the unknown coefficients of the forces and torques, which are instantaneously modified with the body states. The complicated coefficients are significant to decide the sensitive motions of an immersed rigid body. The coefficients are related to the body geometry, the angle of attack of the body α, and the Reynolds number of flow Re [ZJW04, Umb05, Kel11, ZMZvW12], i.e., $C = C(\alpha, Re)$. First, we consider the relationships between the drag and lift coefficients and angle of attack. According to the experimental observations in previous work, a generalized parameter model is proposed as follows:

$$(C_d, C_{l1}, C_{l2}) = (C_D \sin^2 \alpha, C_{L1} \sin(2\alpha), C_{L2} \cos(2\alpha)) \tag{5.17}$$

where $\tilde{C} = \{C_D, C_{L1}, C_{L2\}}$ are the control parameters of the force coefficients. α is the angle of attack calculated in the formulation $\alpha = \tan^{-1}(\|u_n\|/\|u_t\|)$, and u_n and u_t are the normal and tangent components of the translation velocity u. The proposed parameter model is in compliance with the physical observations that all force coefficients have the same oscillating frequencies [VCW13]. Note that the parameter C_D has the same formulation with [ZJW04, ZMZvW12], which concern the drag coefficient at $\alpha = \pi/2$ [APW05b, HW14], i.e., $C_D = C_d(\pi/2)$ where $C_d(0)$ is omitted because of less effects on the force computation in the implementation.

In contrast to the drag and lift coefficients adopted in [WP03, OFM09, JWL$^+$13], the proposed parameter model is generalized with the control parameters $\tilde{C} = \{C_D, C_{L1}, C_{L2}\}$ as shown in Figure 5.4. The control parameters are considered to be an implicit function of the Reynolds number, i.e., $\tilde{C} = \tilde{C}(Re)$. The drag coefficients are fitted perfectly (solid lines) and the lift coefficients are similar (dashed lines) by the proposed parameter model, because the lift coefficients are separated into two components here, and the previous work did not consider this apparently.

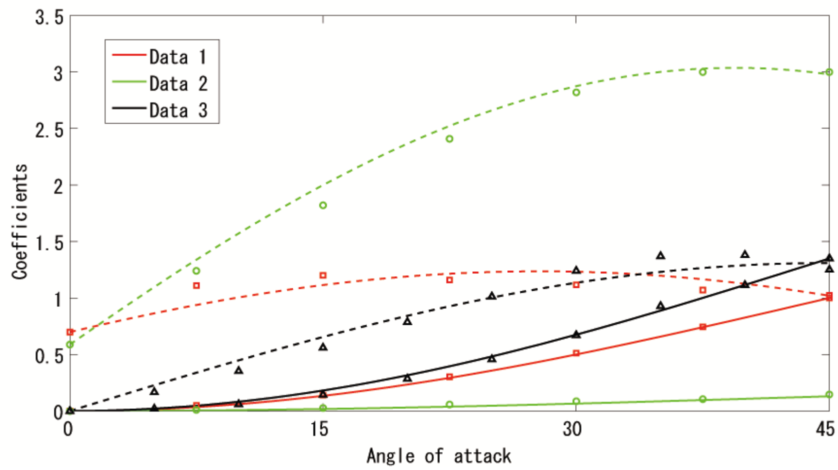


Figure 5.4: Drag and lift coefficients related to the angle of attack. The dashed lines represent the lift coefficients and the solid lines for drag coefficients by the proposed parameter model. The circle-marked data [JWL+13] is fitted by Data 1 ($C_D = 0.26, C_{L1} = 2.98, C_{L2} = 0.59$); the square-marked data [WP03] by Data 2 ($C_D = 2.0, C_{L1} = 1.02, C_{L2} = 0.70$); the triangle-marked data [OFM09] by Data 3 ($C_D = 2.70, C_{L1} = 1.31, C_{L2} = 0$) in the proposed model.

Then, the torque induced by the vortex-flow force is given as follows:

$$\Gamma_M = \vec{p} \times (F_D + F_{L1} + F_{L2}) \tag{5.18}$$

where the vector $\vec{p}$ is from COM to the center of pressure of the proxy geometry. In Kutta condition, $\vec{p}$ is a quarter of the major axis for an ideal flow. In this work, $\|\vec{p}\| = (1 - \sin^3 \alpha)a/4$ is used to account for the viscous effect [ZMZvW12].

For the sake of completeness, the buoyancy-corrected gravity F_G and its torque Γ_G are added into the dynamics equations. Finally, $F_v(X)$ can be expressed as follows:

$$\begin{aligned} F_v(X) &= \begin{pmatrix} F_D + F_{L1} + F_{L2} + F_G \\ \Gamma_M + \Gamma_G \end{pmatrix} \\ &= \begin{pmatrix} F_D + F_{L1} + F_{L2} \\ \Gamma_M \end{pmatrix}(X) + m_f \begin{pmatrix} (\bar{\rho}/\rho_f - 1)R^T g \\ \vec{r} \times R^T g \end{pmatrix} \end{aligned} \tag{5.19}$$

where g is the gravitational acceleration, and $\vec{r}$ represents the vector from the position of COM to the center of buoyancy, and R is the orientation matrix of the immersed rigid body.

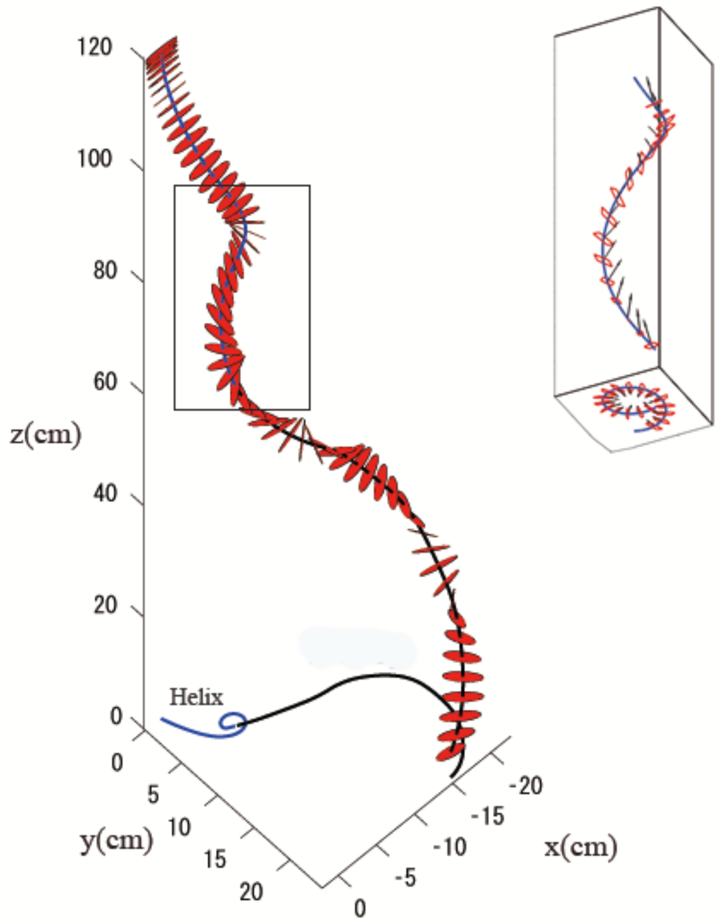


Figure 5.5: Comparison between the simulation result of the proposed dynamical model (rectangle area in the left subfigure) and the captured trajectory from [VCW13] in right-top subfigure.

Validation of dynamical model

In contrast to the dynamical model of previous work [VCW13], the proposed model takes account of the inertial effect from the surrounding flow and the relationships between the force coefficients and angle of attack. As shown in Figure 5.5, the simulation result shows the trajectory of falling card with $\rho_r = 1.2g/cm^3, a = 2.5cm, b = 1.5cm, c = 0.01cm$, and the release angle of 45°. The calculated trajectory exhibits a helical motion with the autorotation motion as reported in previous work.

Table 5.1: Summary of control parameters in previous experimental work.

References	Re	C_D	C_{L1}	C_{L2}
[APW05b]	$10^2 \sim 10^3$	$1.72 \sim 1.92$	1.0	$1.1 \sim 1.4$
[PM11]	10^3	0.4	1.2	π
[VCW13]	10^3	1.06	0.92	0.32
[HLW+13]	1.5×10^3	0.3	4.5	1.8
[HW14]	10^3	$1.4 \sim 2.4$	π	$0 \sim \pi$

5.4 Training motion patterns

The simulation function of an immersed rigid body is defined as $S(t, q, \tilde{c})$ based on the the proposed dynamical model, where t is the time series, and $q = (x, R, u, w)$ is the state of the body including the body's position and orientation, translational and rotational velocities. $\tilde{c} \in \tilde{C}$ represents the control parameters in the simulation of immersed rigid body dynamics.

5.4.1 Parameter spaces

In a phase diagram [FKMN97, ZCL11, AMF13], there are two key parameters: the dimensionless moment of inertia I^* and the Reynolds number Re, which reveal the inertial and viscous effects from the surrounding flow, respectively.

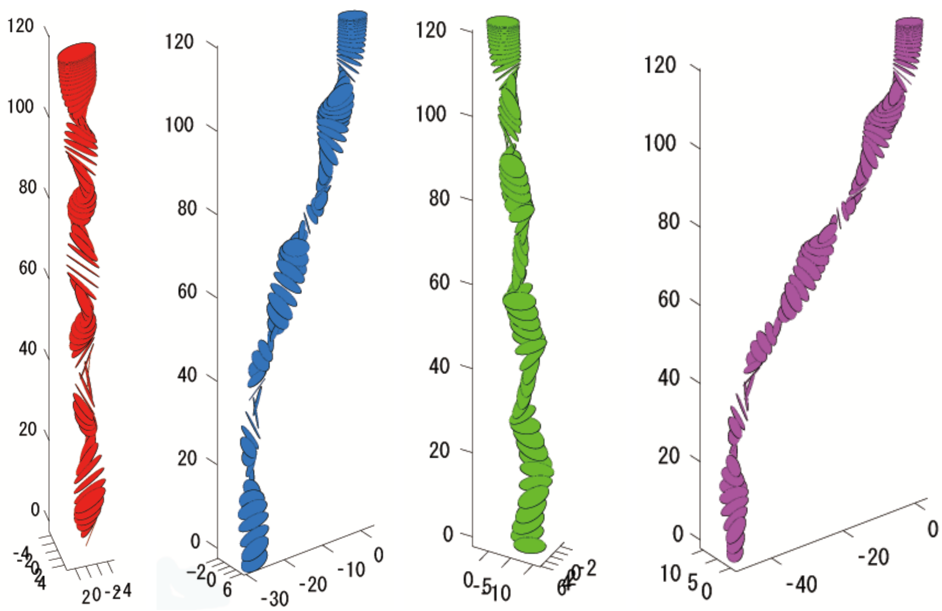
$$Re = \frac{a}{\nu}\sqrt{(\frac{\rho_r}{\rho_f} - 1)gb}, \quad I^* = \frac{8\rho_r(a^2 + b^2)}{3\pi\rho_f a^3} \tag{5.20}$$

where ν is the kinematic viscosity of the surrounding flow. (Re, I^*) cannot directly used as the control parameters in the simulation because they are invariants for the given model of an individual body. Although the function of drag coefficient with Re is analysed elaborately in [Kel11], the relationship with the control parameters and Re with instantaneous body states remains unsolved. As shown in Table 5.1, the different values of control parameters are adopted in the literature (Note that $C_D = C_d(\pi/2) - C_d(0)$). In terms of Equation 5.20, the Reynolds number of immersed rigid body motions is so high that the unsteady dynamics becomes predominant. For a leaf motion in air, $Re = 1.2 \times 10^4$ with $a = 8cm, b = 2.0cm$ and $c = 0.01cm$.

The analysis of the effects of control parameters $\tilde{c}$ is similar to the previous work [TK94, HW14] to find out the motion patterns by sampling parameter space. The key difference between this work and previous works is that this work is the first work on the motion patterns of an immersed rigid body in fully three dimensions. This analysis is also different with the previous work [AMF13], which investigates the three dimensional motion patterns with various aspect ratios by two-way coupling simulations. This work considers the motion patterns of an individual body with various control parameters and the same initial conditions, such as, released angle, velocities (set to be 0 from rest), aspect ratio and body's density. From Table 5.1, we set $\tilde{c} \in (0, \pi) \times (0, \pi) \times (0, \pi)$, and we will explain that the upper limitation of π is sufficient later.

5.4.2 Motion classification

From the numerical simulations in the parameter space $\tilde{C}$, we observed four typical motion patterns in three dimensional space as shown in Figure 5.6: (1) **SD** (steady descent), a body falls down vertically; (2) **ZZ** (zigzag), a fluttering motion that the body turns left and right; (3) **AR** (autorotation), a tumbling motion that the body falls broadside to one side; (4) **AZ** (autorotation and zigzag), a chaotic motion with the transition between autorotation and zigzag motions. These patterns agree with previous two-dimensional and quasi-two-dimensional experiments ([TK94, FKMN97, APW05b, ZCL11]). Furthermore, the similar patterns are also reported in the numerical experiments by the fluid-structure simulations [AMF13].



(a) **SD** (b) **AZ** (c) **ZZ** (d) **AR**

Figure 5.6: Four motion patterns of a falling card from rest observed in this work, where the release angle is 10°, $a = 3.5cm$, and $b = 2cm$: (a) SD ($\tilde{c} = (1.81, 0.31, 0.31)$); (b) AZ ($\tilde{c} = (0.61, 1.81, 1.21)$); (c) ZZ ($\tilde{c} = (0.31, 0.91, 0.61)$); (d) AR ($\tilde{c} = (0.61, 2.71, 0.91)$). The unit of the spacial coordinates is cm.

To give the details of the effects from the surrounding flow, the inertial effect of added-mass tensors causes the planar oscillation in the horizontal plane as shown in Figure 5.6(c).

The amplitude of the oscillation increases with the increased inertial effect. The motion of a coin underwater has a larger fluttering amplitude of ZZ motion as reported in [ZCL11, XM14c]. The viscous effect of drag and lift forces makes the immersed rigid body move far away from the vertical direction to an autorotate motion with increased viscous effect as shown in Figure 5.6(d). As a trade-off among these two effects, the body exhibits chaotic motions with both AR and ZZ motions. AR becomes ZZ at the end of the example in Figure 5.6(c). However, the motions of immersed rigid body in real situations are more complicated due to the turbulence structures of the counter-rotating vortex pair, the leading-edge, and the trailing-edge vortices [ZLS+13], which we will consider in the next section.

To classify the motion patterns from numerical experiments of the proposed dynamical model, the three dimensional trajectories are projected on the horizontal XY plane. Although the orientation information of the body is absent in this process, the motion patterns can be discriminated correctly as reported in previous work [ZCL11, XM14c]. Based on the scattered 2D points, the principal components analysis (PCA) is utilized to obtain the characteristic features of the motions. Here, the eigenvalues and eigenvectors of the data are represented as the axes of an eclipse shape. The center of the eclipse is calculated by the mean value of the data. Then, we adopt a pair value (d, e) to label each motion, where $d \geq 0$ is the distance from the eclipse center to the original release point and $e \in (0, 1)$ is the eccentricity of the eclipse as shown in Figure 5.7.

After defining four motion patterns by the numerical experiments and previous experimental work, the values (d, e) of all sampling data are clustered by k-means method with four clusters as input parameter. As shown in Figure 5.8, the trajectories of elliptical card and leaf are clustered into each patterns in different parameter space, $\tilde{C} = (0, \pi) \times (0, \pi) \times (0, \pi)$ in Figure 5.8(a), and $(0, 2\pi) \times (0, 2\pi) \times (0, 2\pi)$ in Figure 5.8(b), respectively. The classification results are similar in the both case based on (d, e), so that the parameter space $(0, \pi) \times (0, \pi) \times (0, \pi)$ is sufficient in this work. When d is small, the trajectory is close to the original location that the motion pattern is SD. When d increases, we observed that e decreases where ZZ and AZ motions are found. When d becomes large, $e \to 1$. The motion becomes planar and moves far away, which is shown to be AR. There are more data in ZZ zone than the others that the classification results are compliance to the observation in previous work [FKMN97]. According to the initial parameters of the numerical experiments, we obtain $Re = 6 \times 10^3$ and $I^* = 2 \times 10^{-3}$ (Equation 5.20) that the motion of the body is categorized into a periodic motion as ZZ in the phase diagram. Note that the spiral and helical motions observed in [ZCL11, AMF13] are also periodic motions, which are included in ZZ motion pattern.

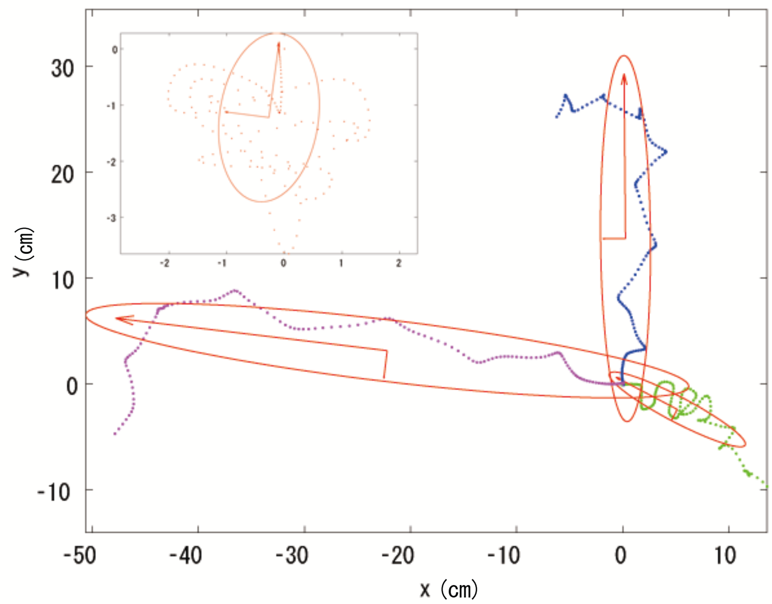
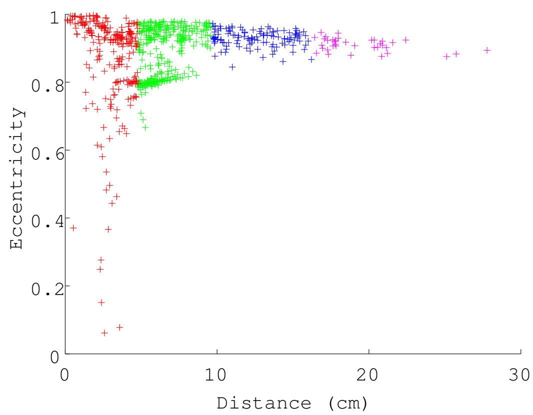


Figure 5.7: The distinction among four motion patterns. SD (red points, left-top subfigure): $(d = 1.25, e = 0.72)$; ZZ (green points): $(d = 5.69, e = 0.79)$; AZ (blue points): $(d = 13.72, e = 0.93)$; AR (magenta points): $(d = 22.43, e = 0.92)$. The projected points are corresponding to the trajectories in Figure 5.6 with the same colors. For better appearance, the points of AZ and ZZ are rotated in $\pi/2$ and π, respectively.

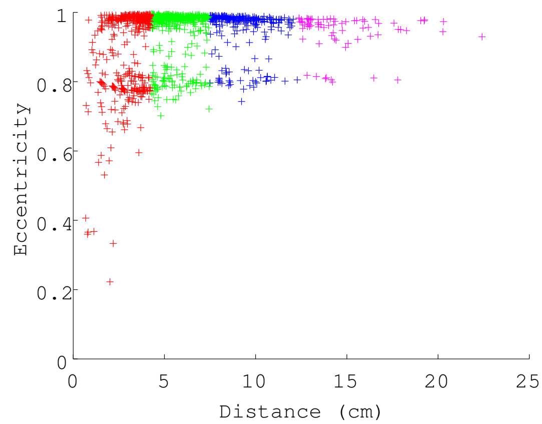
5.4.3 Subspaces construction

According to the clustering results of the numerical experiments by unified sampling in the parameter space, the whole parameter space can be divided into different parameter subspaces corresponding to each motion patterns. This process is finished by indexing data (d, e) with $\tilde{C}$.

To reduce the computational cost of numerical experiments in the parameter space, a resampling process is necessary to obtain the smooth boundaries among the parameter subspaces. In this work, a simple sampling method is proposed as illustrated in Figure 5.9. First, the low resolution sampling grids are subdivided into a high resolution grids, and the motion patterns of the sampling points are decided by their nearest neighbours (6 neighbours in three dimensions). If all the neighbours are in the same motion patterns, the point belongs to this pattern. Here, the process does not account for the points without motion pattern info for simplicity. If the patterns are different, the points should be resimulated again to obtain the pattern info, such as black-triangle marked points in Figure 5.9. Finally, the parameter subspaces are constructed based on the whole parameter space $\tilde{C}$. The computation complexity of the sampling process is reduced from $O(N^3)$ to $O(N^2)$ simulations by the proposed method. As shown in Figure 5.10, the parameter



(a) Card simulations from 729 samples in parameter space.



(b) Leaf simulation from 1,331 samples in parameter space.

Figure 5.8: Clustering results of falling card and falling leaf motions. The different colors represent different motion patterns. red: SD; green: ZZ; blue: AZ; magenta: AR.

space is divided into five regions: SD pattern occupies two regions, and the others have one regions. Note that a very smooth boundary can be achieved by resampling iteratively as shown in Figure 5.9(b), but we found that the subspaces sampling after few steps are sufficient for motion synthesis.

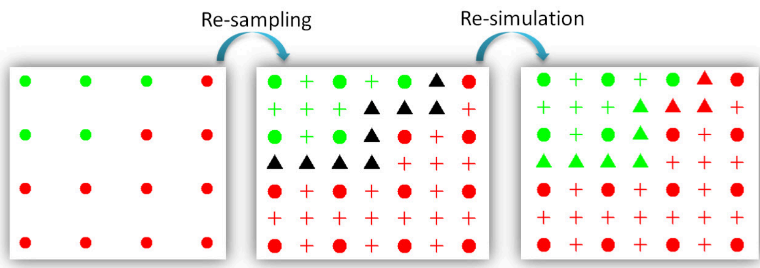
5.5 Motion synthesis

For an immersed rigid body moving in real flow environments, there is a significant effect which is different with both the inertial and viscous effects as discussed before, i.e., the effect due to the coupling of the body with the generated turbulence [APW05b, ZLS+13]. This effect is apparent at the turning points of the motion trajectory. In this work, the motion synthesis of the realistic motions of immersed rigid body is proposed based on a curvature-based motion planning method using a motion graph with turbulent kinetic energy (TKE) optimization.

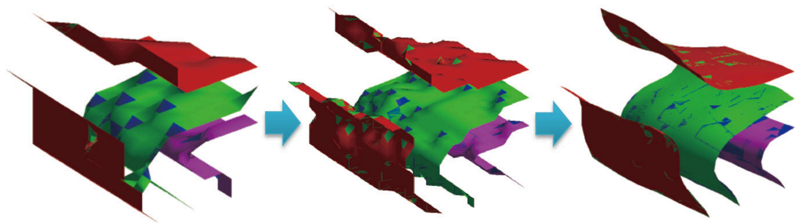
5.5.1 Turbulent kinetic energy

We model TKE by a stochastic approach based on the synthetic turbulence method to approximate the energy transfer among the body and the surrounding flow, which accounts for the viscous effects due to turbulence. The assumption of the proposed method is to treat the objects as suspended particles inside fluid domain as [YCZ11, XM13, SJ13].

Stochastic model. For a statistically isotopic turbulence, a first-order stochastic dif-



(a)



(b)

Figure 5.9: Sampling process from low resolution to high resolution sampling. (a) 2D illustration. The red and green colors denote different patterns. In the middle figure, the data (+) represent the new sampled data, and the black triangles represent the data on the boundary to be resimulated. (b) The boundaries among parameter subspaces after resampling.

ferential equation can be adopted to represent the velocity increments:

$$du^*(t) = D_1 u^*(t)dt + \sqrt{D_2}dW(t) \tag{5.21}$$

This equation is also known as Langevin model [Pop83], where $u^*(t)$ is the translational velocity of the immersed rigid body due to flow viscosity and vorticity; D_1 and D_2 are the drift and diffusion functions, respectively, which reveal the properties of the turbulent flow. $W(t)$ is a continuous-time stochastic process of Brownian motion called Wiener process. In the implementation, the Wiener process is calculated as a normal distribution with mean of zero and variance of the time interval Δt. D_1 and D_2 are the functions of the turbulent kinetic energy k and its dissipation rate ε in a turbulent flow. They are defined as follows:

$$D_1 = -(0.5 + 0.75C_k)\lambda, \qquad D_2 = C_k\varepsilon \tag{5.22}$$

where $\lambda = \varepsilon/k$ is the turbulent frequency. C_k is a Kolmogorov coefficient, which is a function of Re according to the Kolmogorov hypothesis for high-Re flow [Pop11], $C_k(Re) = 6.5 \times (1 + 140 \times Re^{-1.33})^{-0.75}$.

Turbulent model. Considering the turbulent flow generated around the moving rigid

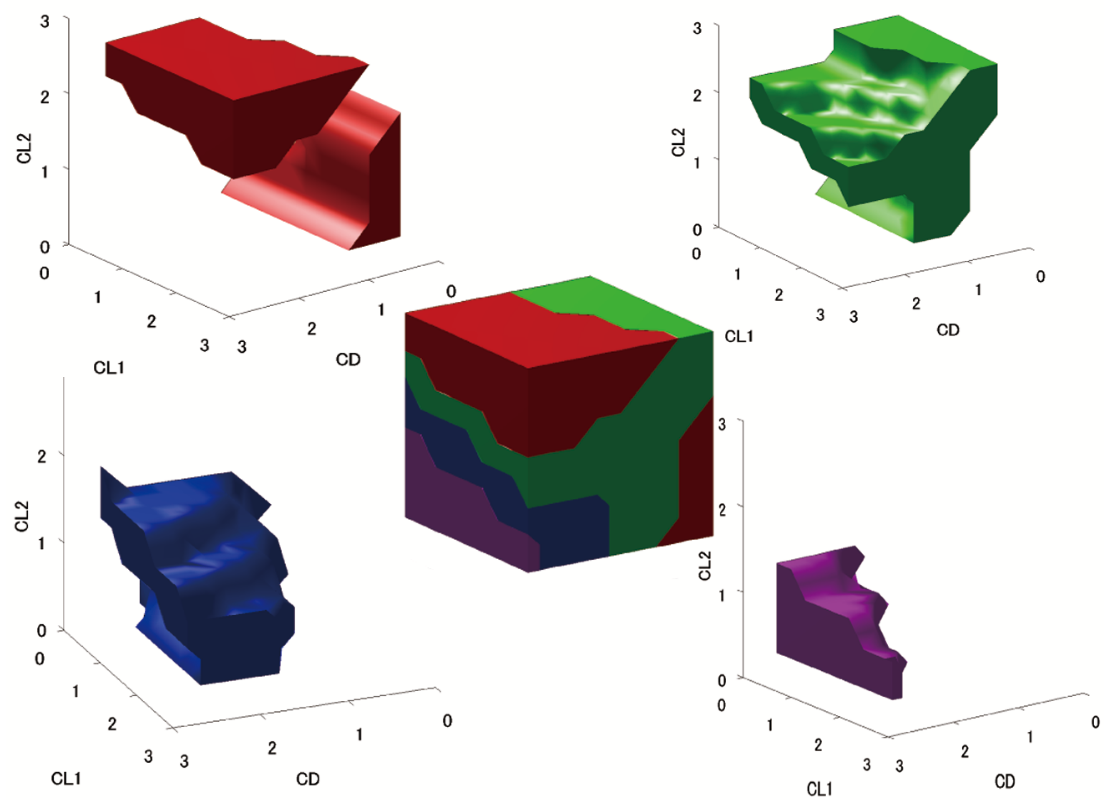


Figure 5.10: Parameter subspaces of different motion patterns of falling card motions. Red: SD; green: ZZ; blue: AZ; magenta: AR.

body, the fluid velocity U of the turbulent flow is decomposed into a mean flow $\langle U \rangle$ and a fluctuating velocity U' ($U = \langle U \rangle + U'$) by Reynolds decomposition. A k-ε turbulent model is adopted in this work. This model is a semi-empirical model based on the energy transport equations [PTC+10], which consists of two coupled equations k and ε as follows:

$$D_t k = G - \varepsilon \tag{5.23}$$

$$D_t \varepsilon = \lambda(C_1 G - C_2 \varepsilon) \tag{5.24}$$

where C_1 and C_2 are the empirical constants with values $C_1 = 1.44$ and $C_2 = 1.92$ [LS74]. G represents the generation of turbulent kinetic energy due to the mean velocity gradients, and is defined in terms of the strain tensor $S_{ij} = \frac{1}{2}(\frac{\partial \langle U \rangle_i}{\partial x_j} + \frac{\partial \langle U \rangle_j}{\partial x_i})$ of the flow: $G = 2 v_T \sum_{ij} S_{ij}^2$, where $v_T = C_\mu k^2 / \varepsilon$ is the turbulent viscosity which describes the small-scale turbulent motion as a viscous diffusion scale in the turbulent model, and $C_\mu = 0.09$ is an empirical constant. In this work, the mean flow simulation of $\langle U \rangle$ is defined on a MAC grid with the mean falling velocity of the body $U_0 = \sqrt{(\rho_s/\rho_f - 1)gb}$ as the inflow velocity.

Note that the implementation of the turbulent model needs to be computed only once in offline processes due to λ and ε do not depend on the body's state. Finally, we obtain velocity increment $du^*(t)$ for the energy optimization in the motion planning with $du^*(t)/dt \sim F_D(t) + F_{L1}(t) + F_{L2}(t)$ in the proposed dynamical model.

5.5.2 Motion graph

Motion transition. Physical experiments of falling immersed rigid bodies are executed to find out the motion patterns, which are recorded by a high-speed camera. The experiments include piece of papers and leaves with different shapes moving in air, coins and plastic materials moving in water. According to our experiment results, the motions of the body transfer in the turn of SD→AR→AZ→ZZ as shown in Figure 5.11. These observations also meets the results by thousands of experiments in the work [Raz10, XM14c]. The analysis of the experimental results is based on the phase diagram found in [FKMN97]. When the object falls down in real flow by the gravity, the instantaneous Reynolds number increases due to the increasing body's velocity. Then, the surrounding flow becomes unsteady to generate turbulences, and the body's motion becomes sensitive to the flow. In this sense, the motion cannot transfer from ZZ to SD motion.

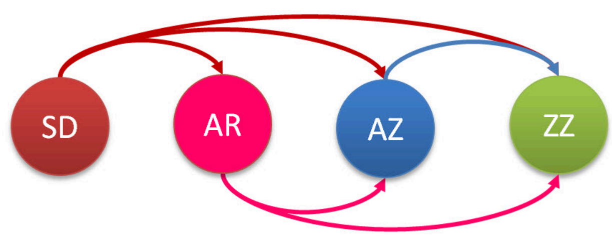


Figure 5.11: Motion transitions among motion patterns.

Graph structure. We model the unsteady dynamics of immersed rigid body using motion graph [KGP02], which is a finite directed graph of the motion patterns: $G = (V, E)$. A node $i \in V$ in this graph corresponds to a motion pattern with a function $q_i = S(t, q_{i-1}, \tilde{c}_i)$ and a prior distribution function $p(X_i)$. An arc $(i, j) \in E$ represents the transition from node i to j with a transition probability $p(q_j|q_i)$ in the given state q_i at node i. Based on the three dimensional experimental results by labelling motion patterns in each video of falling experiments, the transition matrix P of transition probabilities is given as follows:

$$\{P_{ij}\} = \{\frac{N_{ij}}{N_i}\} = \begin{pmatrix} 0.154 & 0.385 & 0.038 & 0.423 \\ 0 & 0.383 & 0.086 & 0.531 \\ 0 & 0 & 0.652 & 0.348 \\ 0 & 0 & 0 & 1.0 \end{pmatrix} \tag{5.25}$$

where N_{ij} and N_i denote the amount of patterns transferred from state q_i to state q_j and the total amount of states transferred from state q_i, respectively. The prior distribution function is given as a normal distribution of motion patterns $\{SD, AR, AZ, ZZ\}$. From the measured P, the transitions among motion patterns are unidirectional, which make the motion planning step effective.

5.5.3 Motion planning

For an immersed rigid body moving in a time interval $[t_0, t_f]$ with initial state q_0, the motion $q(t)$ is defined by n functions:

$$q(t) = \begin{cases} S(t, q_0, \tilde{c}_1) & t \in [t_0, t_1] \\ \cdots & \\ S(t, q_{n-1}, \tilde{c}_n) & t \in [t_{n-1}, t_n] \end{cases} \tag{5.26}$$

where $t_n = t_f$. To guarantee the continuity of the motion $q(t)$, the initial state q_i is set to be $q_i = S(t, q_{i-1}, \tilde{c}_i), i \in (0, n)$. The parameter estimation of control parameters $\{\tilde{c}_1, ..., \tilde{c}_n\}$ at time series $\{t_0, t_1, ..., t_{n-1}\}$ is the key issue for the motion planning in this work, which exhibits the feature of force coefficients as implicit functions of the Reynolds number, i.e., $\tilde{C} = \tilde{C}(Re)$. An efficient curvature-based motion planning solver is proposed to realize the motion graph and the turbulent energy transition of immersed rigid body dynamics.

Energy optimization Because the force coefficients of the dynamical system are instantaneous coefficients related to the angle of attack and Reynolds number, we assume the control parameters are time-varying at each time step. Therefore, the energy transition among the body and the surround flow is calculated by the minimum of time derivatives of the drag and lift forces energy consume of the body E_1, and the approximated TKE of the flow E_2. The following object function of energy optimization is defined from the Equations 5.15 and 5.21:

$$\min_{\tilde{c}} E_i, E_i = \sum_{i=1}^{n} \|dE_1(t_i) - dE_2(t_i)\|^2 \tag{5.27}$$

$$E_1(t) = (F_D + F_{L1} + F_{L2})(X^*, \tilde{c}) \tag{5.28}$$

$$E_2(t) = \sqrt{C_k \varepsilon(t)}\vec{\xi} - \frac{2 + 3C_k}{4}\lambda(t)u^* \tag{5.29}$$

where X^* and u^* are intermediate general velocity and translational velocity of the body, which are calculated from the proposed dynamical model with the absent viscous forces. $\vec{\xi}$ is a random vector defined by $Norm(0, 1)$ as the Wiener process using Box-Muller method. In contrast to the path planning techniques of rigid body simulations [PSE03], the proposed method is a high-level planning approach to estimate the control parameters of force coefficients in dynamical model, and reflects the statistical description of turbulent flow [Pop83].

Curvature-based motion planning The state q_i of each function $S(t, q_{i-1}, \tilde{c}_i), i = 1, ..., n$ is determined by the control parameters $\tilde{c}_i$ for the initial state q_{i-1}. From the data training step (Section 5.4), the motion pattern database of parameter subspaces is constructed for all motion patterns $\{\tilde{C}_i, i \in [1, 4] | (SD, AR, AZ, ZZ)\}$ in the whole parameter space $\tilde{C}$. The parameter subspaces $\{P_i\}$ are disjointed, and $\bigcup \tilde{C}_i = \tilde{C}$. We set a motion graph $G = (V, E), V \in \tilde{C}$ and determine the motion pattern of each state q_i from the prior distribution function and the transition probability. In a special case, a heavy brick falling in air has only one motion pattern (SD) which belongs to the parameter subspace $\tilde{C}_1$.

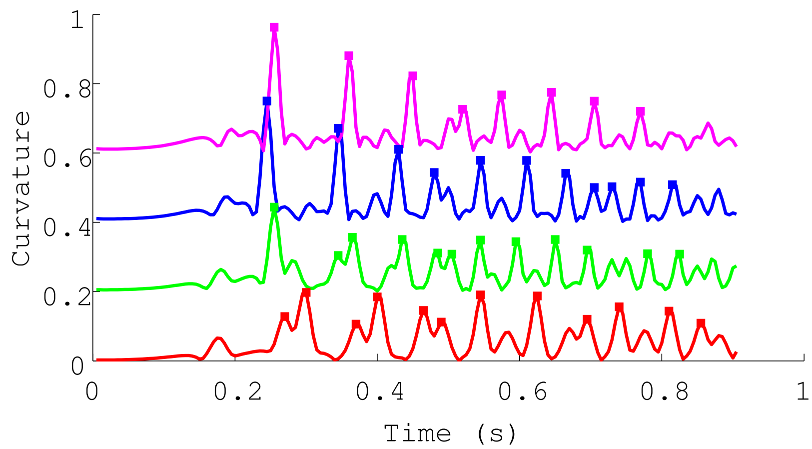


Figure 5.12: Curvature curves corresponding to the motion patterns in Figure 5.6 (Red: SD; green: ZZ; blue: AZ; magenta: AR). The square points on the motion curves denote the maximum values detected on the curvature curves with a tolerance 0.1. For better appearance, the curvature values of ZZ, AZ, and AR are added by 0.2, 0.4, and 0.6, respectively.

Planning algorithm. The motion of an immersed rigid body becomes most sensitive when the body arrives the turning point of the motion trajectory [APW05b]. In this sense, the motion transitions happen at the maximum curvature points as shown in Figure 5.12. The realization algorithm of the motion transition and the motion graph is outlined as follows:

- Start with a random chosen motion pattern $p_0 \in \tilde{C}_i$ in the motion graph at time t_0 and the initial state q_0.
- For each iteration step at time step t_i:
    - If the previous motion pattern is p_{i-1} and the curvature of trajectory form t_0 to t_{i-1} arrives the maximum value, the next pattern p_i is determined by the transition probabilities P in the motion graph. Otherwise, $p_i = p_{i-1}$.
    - Estimating the control parameters $\tilde{c}_i$ in the parameter subspace of motion pat-

tern p_i. After calculating the intermediate velocities, a global simulated annealing optimization [KGV83] is performed that the minimal cost of energy optimization (Equation 5.27) in the parameter subspace is found.

  - Based on $\tilde{c}_i$ and the previous state q_{i-1}, q_i is implemented by the proposed dynamical model.

- The series of control parameters $\{\tilde{c}_1, ..., \tilde{c}_n\}$ of all time steps are obtained by concatenating $\tilde{c}_i$.

Note that various control parameter series can be achieved after repeating the planning algorithm due to the stochastic features of the proposed method. It is also observed experimentally that a body exhibits different motions even with the same initial conditions. Although the optimization steps are time consuming, the motion planning is executed in precomputation steps, and the control parameter series are saved for the runtime simulations, where real-time simulations become possible.

Table 5.2: The configurations of all the simulations in our implementations.

rigid body	release angle	a(cm)	b(cm)	c(cm)	motion patterns
Falling card, Figure 5.5	45°	2.5	1.5	0.01	AZ
Falling card, Figure 5.13	30°	4.0	2.0	0.04	ZZ
Falling leaf, Figure 5.14	10°	7.3	4.2	0.03	SD,AR
Falling leaves, Figure 5.16	6°	7.3	4.2	0.03	SD,AR,AZ,ZZ

5.6 Results

We implemented the proposed methods using MATLAB R2012a on a standard PC with a Core i7 CPU (3.20 GHz) and 12 GB RAM as listed in Table 5.2. The rigid body simulator utilizes a geometric Lie group integrator [KCD09], which enables stable numerical results for various time steps. The average computation time is approximately 20 ms per time step. The turbulent model utilized a typical 32×8 staggered grid, and a semi-Lagrangian method was used to obtain the mean flow. The training data was obtained by the rigid body simulator with different sampling in parameter spaces (approximately 5.7 h for 729 uniform sampling points and 9.3 h for 1311 samples (Figure 5.8)). With the proposed resampling process, 4913 sampling points were obtained in 4.0 h of resimulations.

A comparison of simulation results with different flow effects is presented in Figure 5.13. The experimental object is an elliptical and planar card (4.0 cm major axis, 2.0 cm minor axis, 0.04 cm thickness, and 30° initial release angle). The standard rigid body solver handles the body in a vacuum environment without influence from the surrounding flow such that the body moves straight down. To address the nearly singular problem

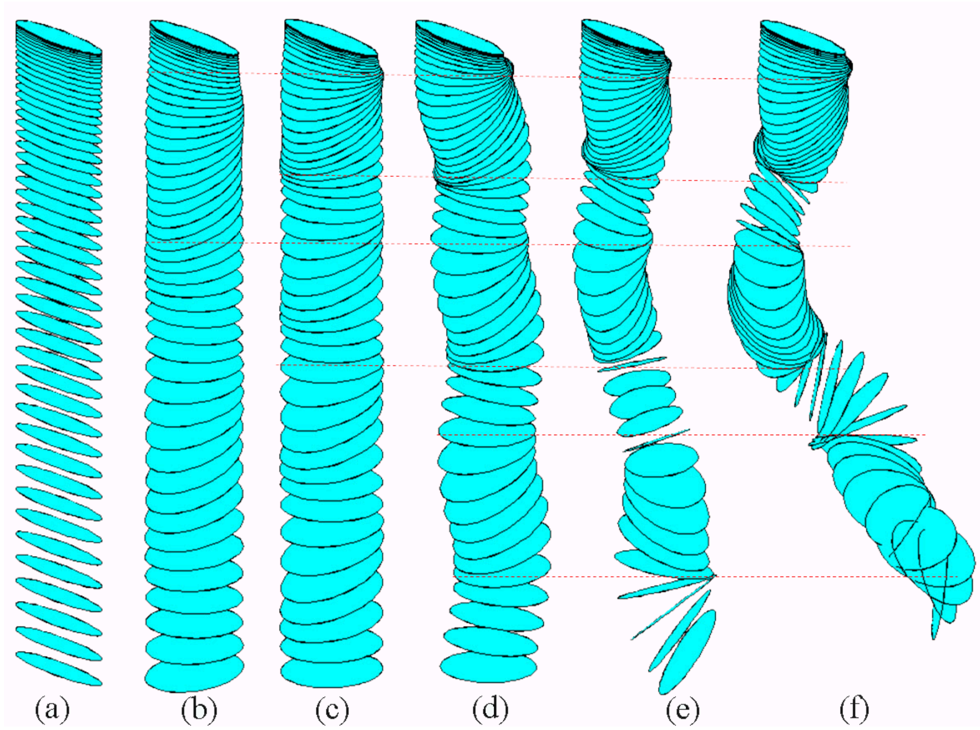


Figure 5.13: Simulation results with different flow effects: (a) rigid body solver without flow effects; (b) rigid body solver with inertial effect (previous method); (c) rigid body solver with inertial effect (proposed method); (d) rigid body solver with inertial effect (proposed method) and viscous effect (previous method); (e) proposed dynamical model with both effects; and (f) proposed immersed rigid body solver with inertial, viscous, and turbulent effects.

in the previous method [WP12], the proposed analytic added tensors capture vertical oscillations in potential flow, as shown in Figures 5.13(b) and 5.13(c). In contrast to the drag and lift forces, for which there is a relationship between their coefficients and angle of attack [WP03, JWL+13], our dynamical model proposes a generalized model of parametric coefficients and distinguishes the translation lift force with rotational lift. Consequently, elevation of the body's COM occurs at the turning points of the motion trajectory (red dashed lines, Figure 5.13). This phenomenon has been observed for unsteady dynamics of plates and insect wings [Umb05]. Figure 5.13(f) shows the simulation results considering inertial, viscous, and turbulent effects from the surrounding flow. Evidently, the motion becomes sensitive to the flow environment and maintains its inertial property.

Figure 5.14 shows a comparison between the simulation results and the captured video of the falling motion of a planar leaf without deformation. The major axis, minor axis, and thickness are 7.3 cm, 4.2 cm, and 0.03 cm, respectively. The captured video suggests that the leaf initially falls straight down and then tumbles from side to side, i.e., $SD \rightarrow AR$. In

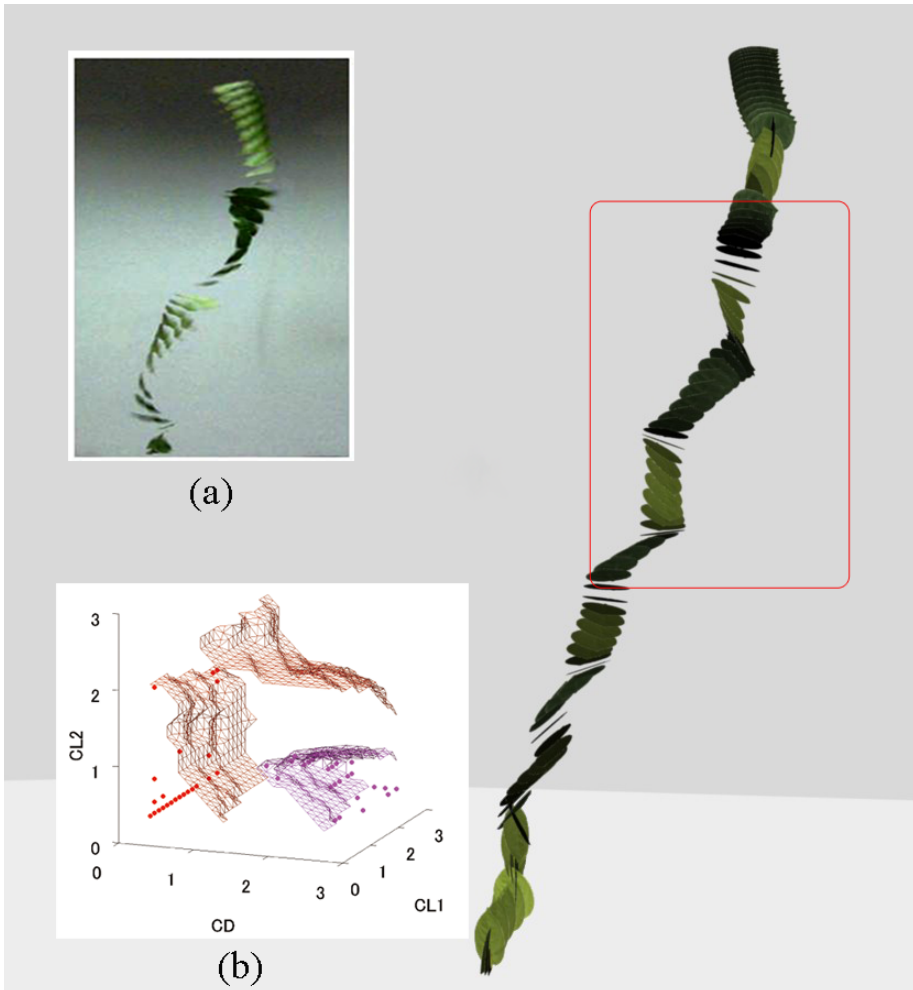


Figure 5.14: Comparison of simulation results and ground truth of a leaf falling in air: (a) ground truth from captured video; (b) optimized control parameters in parameter subspaces (SD, red; AR, magenta). All grid surfaces denote boundary surfaces among parameter subspaces.

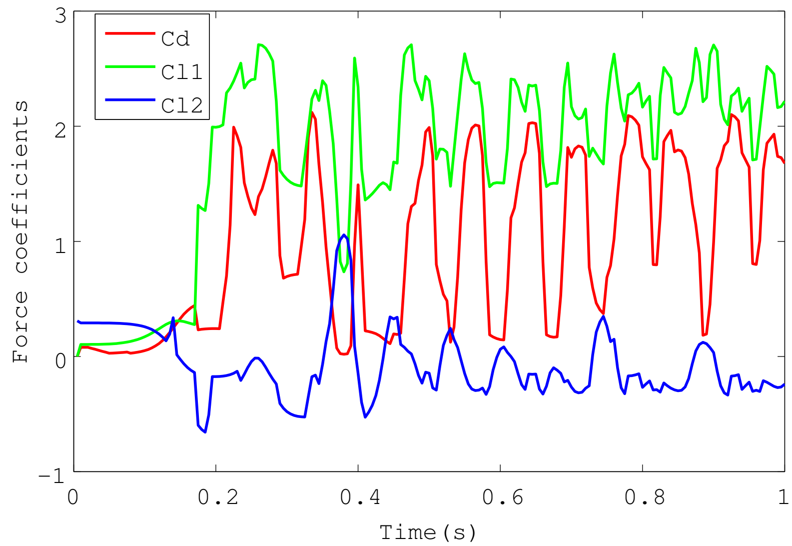


Figure 5.15: Force coefficients C_d, C_{l1} and C_{l2} of drag, rotational lift, and translational lift forces, respectively.

the motion synthesis process, the motion transition is determined by the proposed motion graph. The proposed method executes energy optimizations in the parameter subspaces of both SD and AR motion patterns as shown in Figure 5.14(b). The tumbling motions are simulated correctly by the proposed approach, where double tumbling motions (red rectangle, Figure 5.14) are found by considering the turbulent effects, which is clear from the analysis of the periodic oscillations of force coefficients as shown in Figure 5.15. In contrast to the harmonic oscillations, there are period-two structures, which are considered to correspond to the vortex-shedding period.

The falling motions of falling leave in air are shown in Figure 5.16. All force coefficients from the motion synthesis process have the same initial conditions. Owing to the stochastic features of the proposed methods, the different paths exhibit varied motion transitions among the motion patterns. Because of the influences of turbulence, more complex motions are generated as compared with the basic motion patterns shown in Figure 5.6. In contrast to the ground truth of captured motions from high-speed camera (source: YouTube[1]), the tumbling, fluttering motions and the motion transitions among motion patterns are realistically simulated.

[1] www.youtube.com/watch?v=kLp8Q1OU_w4

Figure 5.16: Comparison between different simulation results of leaves falling from a tree in air under the same initial conditions and the ground truth (top-left subfigure).

5.7 Discussion

This chapter proposed a pattern-driven framework for simulating immersed rigid body dynamics by taking into considerations of inertial, viscous and turbulent effects of the surrounding flow. Four motion patterns of 3D motions are identified in numerical experiments. The relationships among these motion patterns and the control parameter subspaces are clarified. For the parameter estimation of the force coefficients in simulations, we proposed a curvature-based motion-planning method based on a motion graph to represent the effect due to generated turbulence. The proposed method defines the control parameters and parameter spaces for realistic and real-time simulations in graphical applications.

It is very challenging to simulate strong coupling motions between a body and its surrounding flow. To the best of authors' knowledge, a 3D solution for this problem has not been addressed in both physics and graphics research areas. This work employs a methodology to avoid time-consuming and inefficient simulations of vortex-structure interaction. Observations from physical experiments suggest that parameter estimation of force coefficients with the transitions of motion patterns is feasible. By combining experimental and theoretical work, the proposed framework offers a fresh approach to this topic.

In the proposed motion synthesis, we consider only the influences of control parameters of force coefficients with equal initial conditions. Other relevant parameters may include aspect ratio [WHXW13], mass distribution [HLW+13], initial body orientation, initial velocities, and external forces. Our numerical experiments demonstrate similar motions with different aspect ratio (Figure 5.8). The proposed framework can be extended to simulations under these conditions because equal motion patterns are evident in the motion graph. External forces, such as wind forces, and collision detection among bodies can be considered directly in the proposed dynamical model. This work has only considered sharp-edged thin bodies because viscous effects may not be apparent for some complex geometry bodies such as the dragons and Stanford bunny. The proposed framework can be easily embedded into graphical development tools and game engines to enhance simulation realness.

Like other data-driven methods, the proposed pattern-driven framework strongly depends on the precomputed data, the initial conditions in numerical experiments. To resolve this issue, the next chapter provides a novel model reduction technique that does not rely on the simulated data.

Chapter 6

Reduced model of dynamics systems

This chapter describes a reduced model technique for simulating dynamical systems in computer graphics. Most procedural models of physics-based simulations consist of control parameters in a highly dimensional domain in which the real-time controllability of simulations is an ongoing issue. Therefore, this model adopts a separated representation of the model solutions that can be preprocessed offline without relying on the knowledge of the complete solutions. To achieve the functional products in this representation, this work utilizes an iterative method involving enrichment and projection steps in a tensor formulation. The proposed approaches are successfully applied to different parametric and coupled models.

6.1 Introduction

The simulation of dynamical systems in computer graphics (CG) can be divided into two main categories, i.e., physics-based and data-driven methods. Physics-based methods follow physical principles and have seen remarkable progress recently. The main disadvantages of these methods include high computational cost, low simulation controllability due to numerous control parameters, and the reliance on the development of related knowledge. Alternatively, data-driven methods are more efficient and adaptable to complex dynamical systems, where prerecorded data are largely consumed. One of the limitations of these methods is that the simulation results are highly restricted in the prior database or the training data.

A reduced model is a spectacular strategy in data-driven methods, which has been applied successfully to the simulations of deformable bodies [JK03] and fluids [TLP06]. Most reduced models are based on proper orthogonal decomposition, also known as principal component analysis, which is a posterior method built on a precomputed data field to determine coherent features and reduced basis. Our goal is to introduce a prior reduced model that does not rely on the preprocessed solutions of the problems. The prior reduced model [CKL14] is a developing technique based on separated representations. It has recently been used in different engineering research, including fluids [DAA11, DAA13] and soft tissue [NO13]. In contrast to the previous studies [DAA11, DAA13, CKL14], this study describes separated representation in discrete and tensor formulations with high-dimensional dynamical systems and proposes a decoupling approach for coupled problems. In other words, the contribution of this study is the first reported attempt to construct a practical framework for separated representations that can be used in CG applications to achieve realistic simulations at low computational cost.

6.2 Reduced model

6.2.1 Problem description

Given a dynamical system $D(U) = G(U, P)$ with unknown state field $U(x_1, x_2, ..., x_d)$ where $U = U(t) \in \mathbb{R}^N, t \in [0, T]$, N denotes the degree of freedom (DOF) of the system, G is a source term related to the state U and parameter set $P(p_1.p_2, ...)$ and D is represented as a differential operator from the time or parameter dependent ordinary or partial differential equations. The solution of the dynamical systems can be approximated in a high-dimensional domain $(x_1, x_2, ..., x_d) \in \Omega_1 \times \Omega_2 \times ... \times \Omega_d$ as follows:

$$U(x_1, x_2, ..., x_d) = \sum_{i=1}^{N} \alpha_i \prod_{j=1}^{d} U_i^j(x_j) \tag{6.1}$$

This is also known as a separated representation of the solution [BM05, CM10]. The representation is a sum of N functional products of prior unknown functions $U_i^j(x_j)$ and the normalization coefficients α_i ($j = 1, 2, ..., d$ in the following sections), which are constructed by enrichment steps in an iterative manner. As soon as this representation becomes available, the approximated solution with different domains is obtained, i.e., temporal and spatial domains, physical parameters, and initial and boundary conditions as extra coordinates as shown in Figure 6.1. Assuming to discretize each domain in M nodes, then the representation involves $N \times d \times M$ rather than M^d DOFs in the original problem. For example, if $d = 6, M = 300$, and $N = 15$ (usually, $N \ll M$), the separated representation reduces the DOFs of the dynamical model at a magnitude of 10^{10} . In this sense, the separated representation is a model reduction technique, also known as proper generalized decomposition. In contrast to other reduced models, such as proper orthogonal decomposition, it is a priori model that does not depend on fully precomputed snapshots of the solution. In a two-dimensional problem, the separated representation is similar to singular value decomposition; however, this approach is efficient in high-dimensional dynamical system problems.

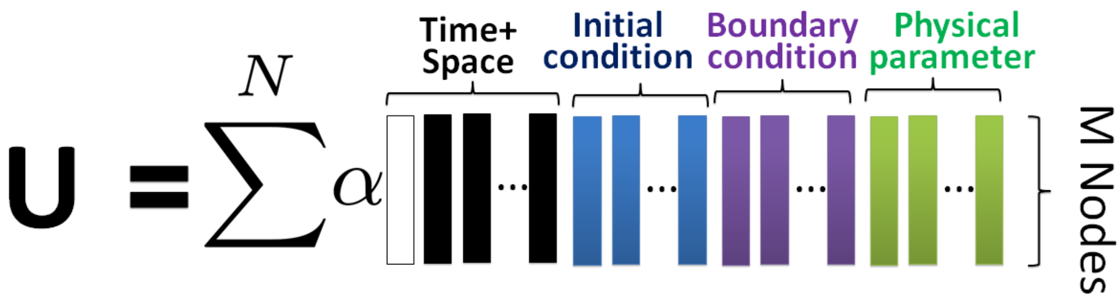


Figure 6.1: Illustration of the separated representation.

6.2.2 Reduction solver

To determine the functions $U_i^j(x_j)$ and coefficient α_i in the representation Equation (6.1), the first $n-1$ separated representation has been obtained at step n. It is straightforward to utilize an iterative process to calculate each $U_n^j(x_j)$. First, the algorithm starts from $\alpha_n = 1$, which is then recalculated from a projection process. The solution of the representation at step n is defined as follows:

$$U = \sum_{i=1}^{n-1} \alpha_i \prod_{j=1}^{d} U_i^j(x_j) + \prod_{j=1}^{d} U_n^j(x_j) \tag{6.2}$$

where $U_n = \{U_n^1(x_1), U_n^2(x_2), ..., U_n^d(x_d)\}$ are the test functions that need to be solved next. Then, each term of U_n is projected on the weak form of the dynamical model $D(U) = G$.

$$\langle D(U), U_n^j \rangle_{\Omega_j} = \langle G, U_n^j \rangle_{\Omega_j} \tag{6.3}$$

where $\langle , \rangle_{\Omega_j}$ represents the scalar product in L^2 norm on the domain Ω_j. Note that the following residual term R^n is omitted in the weak form which can be used to check the process convergence.

$$R^n = D(\sum_{i=1}^{n-1} \alpha_i \prod_{j=1}^{d} U_i^j(x_j) + \prod_{j=1}^{d} U_n^j(x_j)) - G \tag{6.4}$$

To solve each $U_n, 1 \leq n \leq N$, a simple choice to obtain the enrich term U_n is an iterative method as an alternating directions fixed-point algorithm to solve Equation (6.3) simultaneously. The idea at p-th iteration for U_n is described as follows. First, u_p^1 is computed with the previously obtained values $(u_{p-1}^2, u_{p-1}^3, ..., u_{p-1}^d)$ (Small letter u is distinguished from capital letter U for U_n in a fixed-point iterative process.). Then, for the term $u_p^k, k \in (1, d]$, the updated values and previous values $(u_p^1, ..., u_p^{k-1}, u_{p-1}^{k+1}, ..., u_{p-1}^d)$ are utilized. After reaching convergence, the U_n values are updated from u.

As per the obtained n functional products $\prod_{j=1}^{d} U_i^j(x_j), 1 \leq i \leq n$, the coefficients α_i

is computed by projection of $D(U)$ to each functional product.

$$\langle D(U), \prod_{j=1}^{d} U_i^j(x_j)\rangle = \langle G, \prod_{j=1}^{d} U_i^j(x_j)\rangle \tag{6.5}$$

Finally, if the residual term $R^n < \epsilon$, ϵ is a designated threshold value, the entire process is in convergence; otherwise, the computation process returns to the enrichment step in the fixed-point algorithm.

6.2.3 Discrete formulation

Here, the scalar products in L^2 are clarified using a discrete formulation. For simplicity, a linear differential operator is considered as follows:

$$\frac{du}{dt} + ku = 0 \tag{6.6}$$

where $u(t,k) = \sum_{i=1}^{N} T_i(t)K_i(k)$ on the domain $\Omega_t \times \Omega_k$ where the normalization coefficients α_i are omitted for simplicity. From Equation (6.3), the formulation is substituted as follows:

$$\langle \frac{dT_n}{dt}, T_n\rangle\langle K_n, K_n\rangle + \langle T_n, T_n\rangle\langle kK_n, K_n\rangle = -\sum_{i=1}^{n-1}(\langle \frac{dT_i}{dt}, T_n\rangle\langle K_i, K_n\rangle + \langle T_i(t), T_n\rangle\langle kK_i(k), K_n\rangle) \tag{6.7}$$

By adopting finite element discretization techniques in each domain mesh, the equation has the following matrix form (for simplicity, the subscript n is omitted):

$$T^T PT \cdot K^T M_k K + T^T M_t T \cdot K^T NK = -\sum_{i=1}^{n-1}(T_i^T PT \cdot K_i^T M_k K + T_i M_t T \cdot K_i^T NK) \tag{6.8}$$

The definitions of P, N, M_t,and M_k are as follows:

$$\begin{cases} P_{ij} = \int_{\Omega_t} \frac{dN_i}{dt} N_j dt \\ N_{ij} = \int_{\Omega_k} N_i k N_j dk \\ M_t = \int_{\Omega_t} N_i N_j dt \\ M_k = \int_{\Omega_k} N_i N_j dk \end{cases} \tag{6.9}$$

where N_i and N_j are shape functions associated with meshes on Ω_t and Ω_k. Note that the discrete formulation is commonly available for other differential operators, such as gradient and Laplacian etc..

Algorithm 3 Pseudo-code for the prior reduced model of the separated representation.

Initialize D, G,and U from Equations (6.8)(6.10)
for $E = 1$ to E_{max} **do** // number of coupled equations
  for $N = 1$ to N_{max} **do** // number of enrichments
    for $p = 1$ to p_{max} **do** // fixed-point iteration
      Compute R_j^p from Equation (6.12) // enrichment step
      Check convergence from Equation (6.13)
    end for
    Normalize U_j^n from Equation (6.14)
    Compute coefficients α_i from Equation (6.15) // projection step
    Update U_n
    Check convergence from Equation (6.16)
  end for
  Update G
end for

6.2.4 Tensor formulation

From the discrete form of a dynamical model, the separated representation can be described in algebraic form with tensor products. For $D(U) = G$:

$$D = \sum_{i=1}^{N_D} D_1^i \otimes D_2^i ... \otimes D_d^i, \quad G = \sum_{i=1}^{N_G} G_1^i \otimes G_2^i ... \otimes G_d^i, \quad U = \sum_{i=1}^{N} \alpha_i U_1^i \otimes U_2^i ... \otimes U_d^i \tag{6.10}$$

where $D_j^i, j = 1, 2, ..., d$ is an $N_j \times N_j$ matrix, and N_j is the number of nodes in domain mesh Ω_j. The sizes of G_j^i and U_j^i are N_j, and they can be obtained directly from the discrete formulation of the dynamical system. The implementation details of the algorithm (Section 6.2.2) is described by utilizing a tensor formulation in the next section.

6.3 Implementation

In this section, the proposed algorithm is discussed comprehensively. The pseudo code is shown in Algorithm 3.

6.3.1 Enrichment step

As mentioned in Section 6.2.2, the fixed-point algorithm is adopted to search for an enrichment $U_n = \alpha_n R_1 \otimes R_2 ... \otimes R_d$ using iterative processes. In the dynamical model, the following formulation of U_n from Equation (6.2) is achieved.

$$\sum_{i=1}^{N_D} D_1^i R_1 \otimes D_2^i R_2 ... \otimes D_d^i R_d = G - \sum_{i=1}^{N_D} \sum_{k=1}^{n-1} \alpha_i D_1^i U_1^k \otimes D_2^i U_2^k ... \otimes D_d^i U_d^k \tag{6.11}$$

Note that $\alpha_n = 1$. At the p-th step of the fixed-point iteration, $R_j^p, 1 \leq j \leq d$ is obtained from $(R_1^p, .., R_{j-1}^p R_{j+1}^{p-1}, ...R_d^{p-1})$. Thus, the following formulation is obtained:

$$ER_j = \sum_{i=1}^{N_G}(\prod_{k=1,k\neq j}^{d} R_k^T G_k^i)G_j^i - \sum_{i=1}^{N_D}\sum_{k=1}^{n-1}(\prod_{m=1,m\neq j}^{d} R_m^T D_m^i U_m^i)\alpha^k D_j^i U_j^k \tag{6.12}$$

Here, the matrix $E = \sum_{i=1}^{N_D}(\prod_{k=1,k\neq j}^{d} R_k^T D_k^i R_k)D_j^i$. After all $(R_1, R_2, ..., R_d)$ are obtained at the p-th step. The convergence condition is defined as follows:

$$\|R_1^p \otimes R_2^p ... \otimes R_d^p - R_1^{p-1} \otimes R_2^{p-1} ... \otimes R_d^{p-1}\| < \epsilon \tag{6.13}$$

Here, ϵ is set by the user and $\|\cdot\|$ represents L^2-norm. Finally, U_j^n in the formulation of U (Equation 6.10) is obtained by the normalization of each R_j.

$$U_j^n = \frac{R_j}{\|R_j\|}, \qquad j = 1, 2, ..., d \tag{6.14}$$

6.3.2 Projection step

From the Equation 6.5, the formulation is modified as follows by using the value of G.

$$BA = H, \quad B_{ij} = \sum_{k=1}^{N_D}[\prod_{e=1}^{d}(U_e^i)^T D_e^k U_e^j] \quad H_i = \sum_{m=1}^{N_G}[\prod_{k=1}^{d}(U_k^i)^T G_k^m] \tag{6.15}$$

where $A = [\alpha_1 \alpha_2 ... \alpha_n]^T$ and $1 \leq i, j \leq n$. Finally, U_n is updated and the residual term R^n is given as follows:

$$R^n = \sum_{i=1}^{N_D}\sum_{k=1}^{n}\alpha_i D_1^i U_1^k \otimes D_2^i U_2^k ... \otimes D_d^i U_d^k - G \tag{6.16}$$

6.3.3 Coupled terms

For simultaneous differential equations, their solutions benefit from utilizing a decoupling strategy to reduce computational complexity. If another variable $W = \sum_{i=1}^{N} \beta_i W_1^i \otimes W_2^i ... \otimes W_d^i + S_1 \otimes S_2 ... \otimes S_d$ exists, then the coupled equations can be decoupled as follows:

$$\begin{cases} D_U(U, W, R, S = 0) = G_U \\ D_W(U, W, R = 0, S) = G_W \end{cases} \tag{6.17}$$

where D_U and D_W are different operators for U and W, respectively. Therefore, there is $(n-1) \times (n-1)$ terms when a multiple term $U \cdot W$ is computed using their previously known values $U_{n-1} \cdot W_{n-1}$. The value of the coupled term is known; therefore, all the terms in G are treated as source terms, as shown in Line 13 of Algorithm 3.

6.4 Numerical results

This section provides numerical examples to verify the efficiency and the accuracy of the proposed prior reduced model of dynamical systems. The examples include a parametric model and two coupled models with unknown initial values. All examples were implemented on a standard PC (Intel Core i7 CPU 2.10 GHz and 8.0 GB RAM), and their reference solutions were obtained from MATLAB ODE solvers.

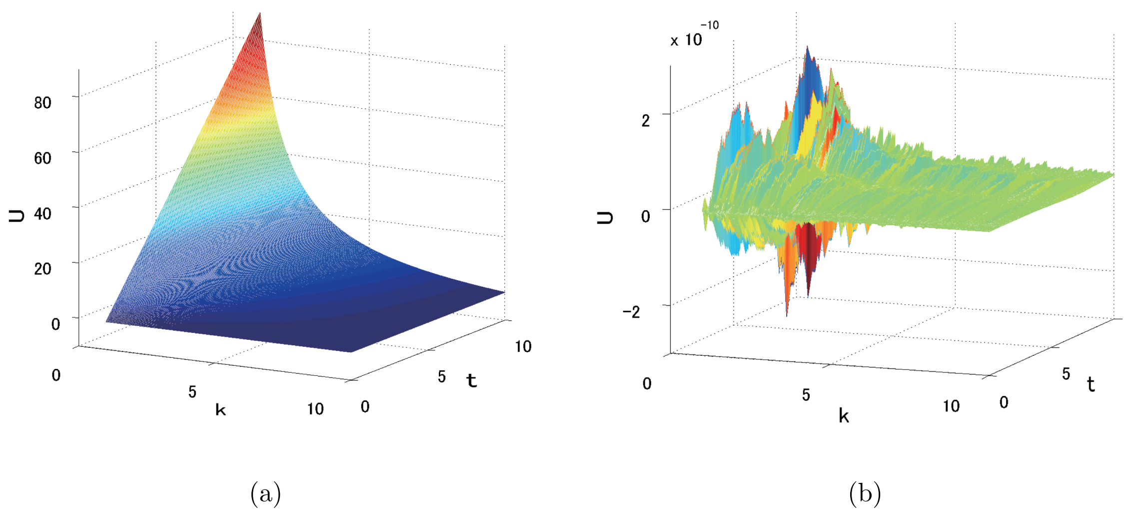


Figure 6.2: (a) Numerical result of the separated representation. (b) Computation error compared with the reference solution.

6.4.1 Parametric linear model

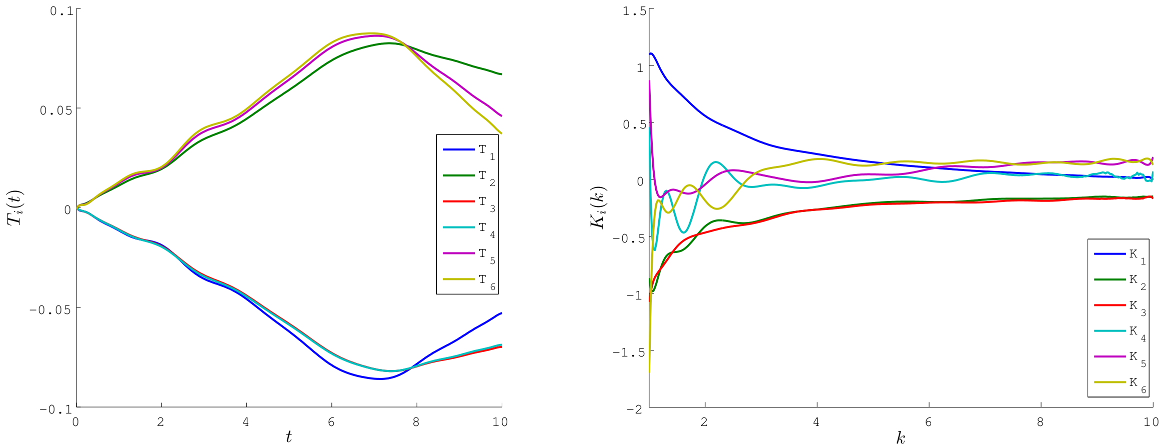
In a separated representation, the control parameters can be introduced into the representation as extra coordinates. The following differential equation is considered as an example:

$$k(\frac{du}{dt} + 1) = 10 \tag{6.18}$$

with initial condition $u(t = 0) = 0$. The separated representation of this parametric model is given in the following formulation:

$$u(t, k) = \sum_{i=1}^{n} \alpha_i T_i(t) K_i(k) \tag{6.19}$$

where $t \in [0, 10]$ and $k \in [1, 10]$. Figure 6.2 (a) shows the simulation results obtained using the proposed approach, which is sufficiently accurate with an error level of 10^{-11} (Figure 6.2 (b)). The reference solutions in the examples are calculated by an ODE solver, e.g.,



(a) (b)

Figure 6.3: First six functions of $T_i(t)$ and $K_i(k)$. (Note that the values of functions have been curve fitted by the polynomial curves)

the Runge-Kutta method. Figure 6.3 shows the first six functions in the representation, which are obtained by the enrichment steps in Algorithm 3.

6.4.2 Parametric nonlinear model

This example illustrates the proposed approach to a nonliear model of one-dimensional free fall problem. Considering one body with mass m and cross-section area A falling from rest, the dynamical equation has the following form.

$$\frac{du}{dt} = \frac{1}{2m}\rho_f C_D A u^2 - g \tag{6.20}$$

where ρ_f denotes the density of the surrounding fluid; C_D is the drag coefficient and g is the gravitational acceleration. The falling velocity is set to related to time and drag coefficient with the following representation.

$$u(t, C_D) = \sum_{j=1}^{n} \alpha_j T_j(t) D_j(C_D) \tag{6.21}$$

The exact solution of the equation is given as $u(t, C_D) = -u_T \tanh(\frac{gt}{u_T})$, where u_T represents the terminal velocity $(\frac{2mg}{\rho_f C_D A})^{\frac{1}{2}}$. The nonlinear term is evaluated by Newton iteration. The proposed reduced model achieved the accurate approximation results as shown in Figure 6.4. The approach is converged after only 5 iterations as illustrated in Figure 6.5. Note that the error level here is calculated by $\|u - u_{exact}\|/\|u_{exact}\|$.

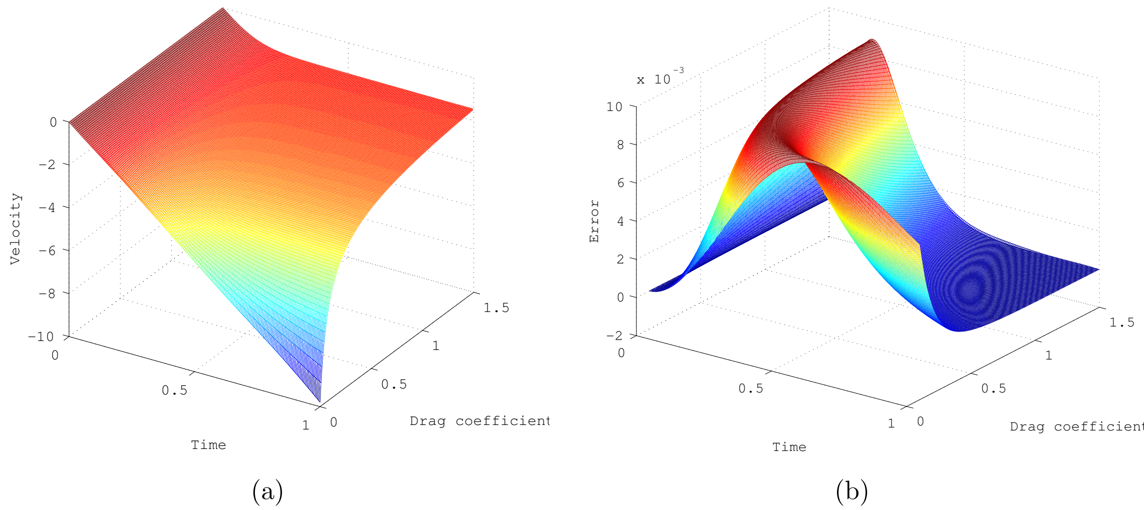


Figure 6.4: Simulation result and its error compared with exact solution. The parameters are $\rho_f = 1.23, g = 9.81, A = 0.5, m = 0.1$.

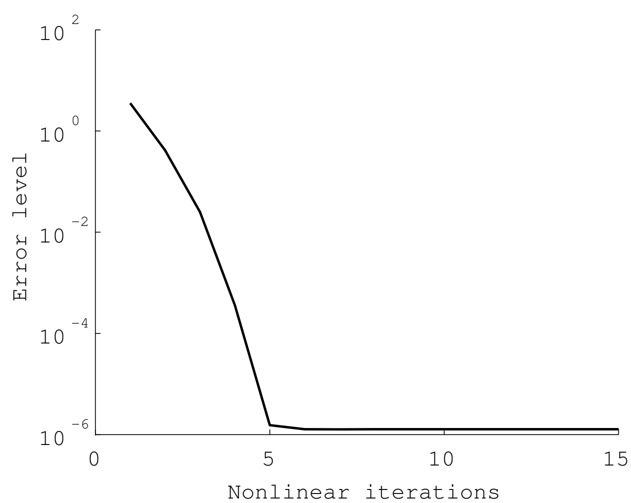


Figure 6.5: Convergence of our approach toward the exact solution as iteration increased.

6.4.3 Coupled model

This example evaluates the proposed method for dynamical systems with coupled terms using the following differential equations:

$$\begin{cases} \frac{du_1}{dt} + u_2 u_3 = 1 \\ \frac{du_2}{dt} + u_1 u_3 = 2 \\ \frac{du_3}{dt} + u_1 u_2 = 3 \end{cases} \tag{6.22}$$

The initial conditions $u_i(t=0) = u_i^0, i = 1, 2, 3$ are considered unknown in this example. To solve these equations, new variables $\hat{u}_i = u_i - u_i^0$ are introduced. Then, Equation (6.22) becomes a system of the variables $\hat{u}_i$, where the initial conditions are considered as new model coordinates in the separated representation of $\hat{u}_i$.

$$\hat{u}_i(t, u_i^0) = \sum_{j=1}^{n} \alpha_j T_j(t) U_j^1(u_1^0) U_j^2(u_2^0) U_j^3(u_3^0) \tag{6.23}$$

where the domains are $\Omega_t(t) \times \Omega_1(u_1^0) \times \Omega_2(u_2^0) \times \Omega_3(u_3^0) = [0, 1] \times [0, 1] \times [0, 1] \times [0, 1]$. From Figure 6.6, our simulation results have good compliance with the reference solutions in this coupled model. The coupled terms in this example are solved by the proposed approach, and the convergence of the solutions can be achieved after eight iterations, as shown in Figure 6.7 (a).

As per Figure 6.7 (b), the computation speed of the proposed reduced model is stable and fast compared to a simple iterative procedure for parameter identification. The computation cost of the reference ODE solver increases exponentially, which is known as *curse of dimensionality* [GMP+12]. For example, the computation cost of the proposed method is only 17 ms, i.e., 50 times faster than the reference approach when the node numbers of $u_i^0, i = 1, 2, 3$ are set to 20, as shown in Figure 6.7 (b). The computation cost can be reduced remarkably with high DOFs of the dynamical system.

6.4.4 Complex model

This example considers six DOFs rigid body dynamics in potential flow [MM02, XM13], where the non-linear viscous forces are omitted. The dynamical equations are as follows:

$$\begin{cases} (mE + M)\frac{du}{dt} = (mE + M)u \times \omega + f_g \\ (J + I)\frac{d\omega}{dt} = (J + I)\omega \times \omega + (Mu) \times u + \tau_g \end{cases} \tag{6.24}$$

where E is a unit tensor, I is the moment of inertia of the body, M and J are added mass and added moment of inertia due to the accelerations from the surrounding flow, respectively, and f_g and τ_g are the force and its torque from the buoyancy-corrected gravity

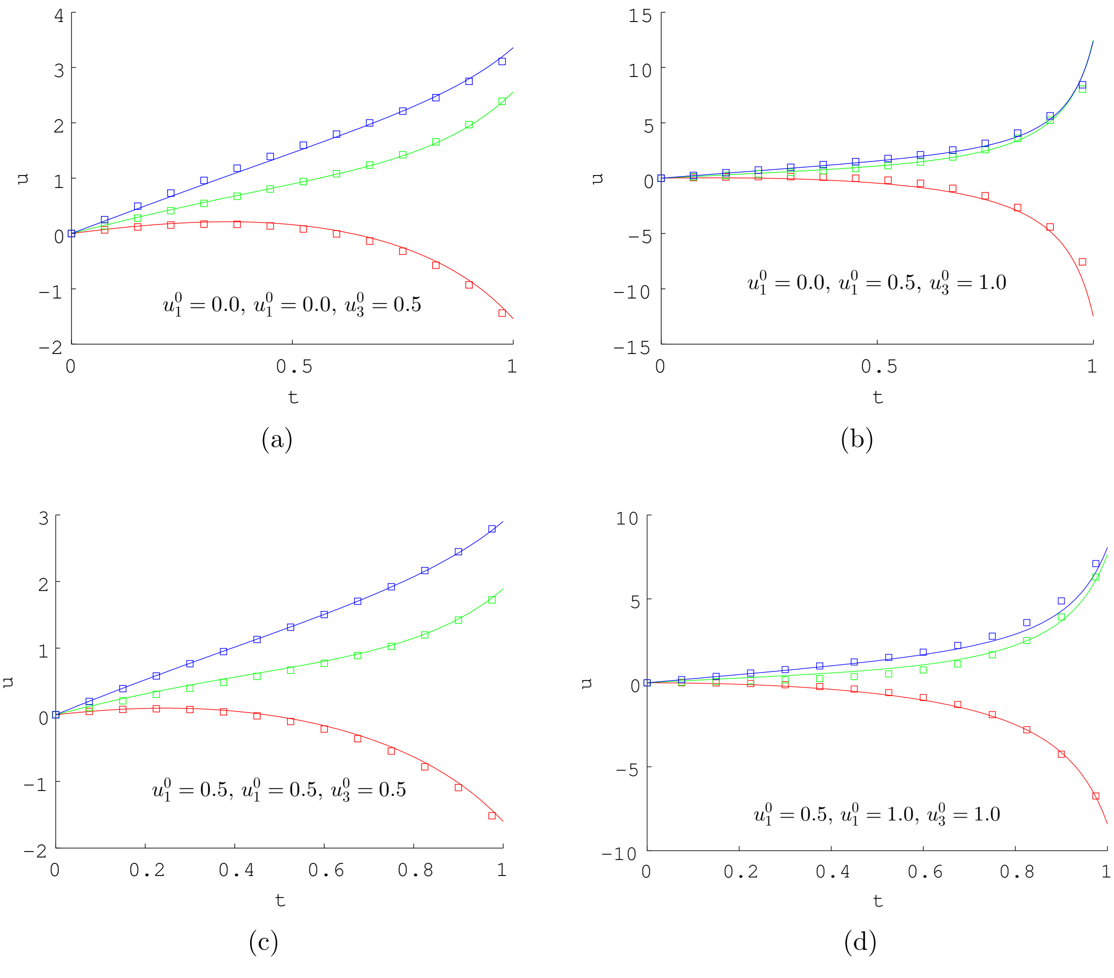
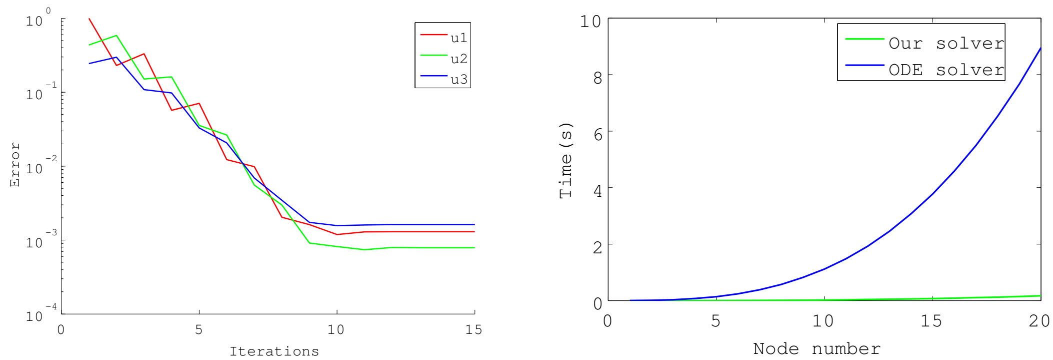


Figure 6.6: Comparison with reference results for different values of initial conditions (lines represent reference results; empty squares represent the computation results of our separated representation (red: u_1; blue: u_2; green: u_3))



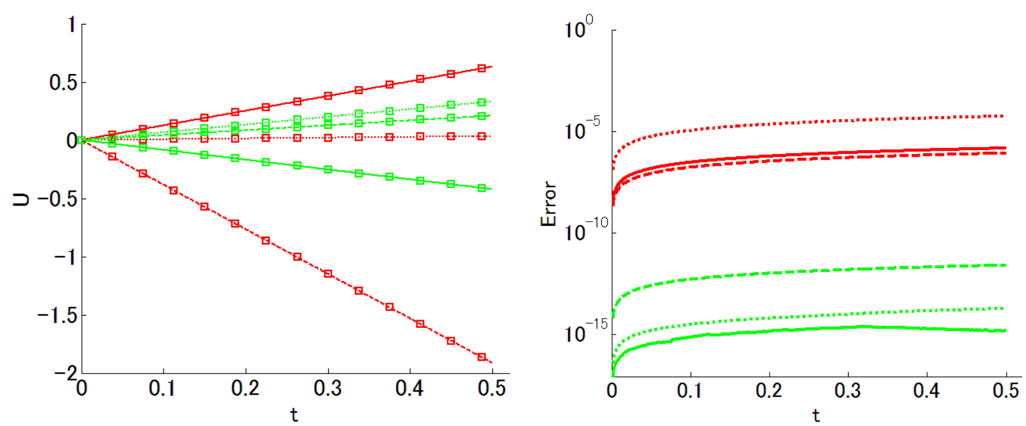
(a) (b)

Figure 6.7: (a) Convergence of simulation results after iterations in the coupled model with initial conditions corresponding to the case of Figure 3 (d). (b) Computation times compared with a simple iterative procedure.

in terms of initial velocity state, respectively. This example does not consider the strongly coupled terms due to translational and angular velocities (u, ω). However, it is helpful to evaluate the strongly coupled terms due to initial conditions, where six initial values of $U_0 = (u_0, \omega_0) \in \mathbb{R}^6$ are introduced in the following separated representation as new coordinates.

$$U_k(t, u_k^0) = \sum_{i=1}^{n} \alpha_i T_i(t) \prod_{j=1}^{6} U_i^j(u_j^0) \tag{6.25}$$

where $1 \leq k \leq 6$. The simulation results of the separated representation match the reference solution as shown in Figure 6.8.



(a) (b)

Figure 6.8: (a) Simulation result with initial velocities $[1.0, 1.0, 1.0, 1.0, 1.0, 1.0]$ and (b) its computation error (red: translational velocity; green: angular velocity)

6.5 Towards immersed rigid body dynamics

The main issue in solving separated representation concerns the strong non-linearity of coupled dynamics in immersed rigid-body dynamics. The treatment of nonlinear term in $f(X, c_1, ..., c_d)$ is significant to obtain accuracy and convergence in the proposed model. The linearization strategies in previous work perform good in linear or weakly-nonlinear model reduction system. First, the limitations of previous work [PCA10, CLB+13a, CLB+13b] accounting for a nonlinear and non-parameterized term $f(x)$ (for simplicity) are analyzed as follows.

- **Incremental linearization (IL).** In numerical simulation, the incremental linearization utilizes the solution of the representation in previous step.

$$f(x) \approx f(\sum_{i=1}^{n-1} T_i(t) \circ \prod_{j=1}^{d} P_i^j(c_j)) \tag{6.26}$$

- **Polynomial linearization (PL).** Based on a polynomial expansion of the nonlinearity about some initial or equilibrium state x_0, $f(x)$ can be expanded as

$$f(x) \approx f(x_0) + J_0(x - x_0) + \frac{1}{2} H_0(x - x_0) \otimes (x - x_0) \tag{6.27}$$

  where J_0 and H_0 are the Jacobian and Hessian of $f(x)$ at point x_0, respectively. The symbol $\otimes$ stands for the Kronecker product.

- **Asymptotic Numerical Method (ANM).** The unknown field x has an asymptotic expansion with a expansion parameter λ.

$$x = x_0 + \lambda x_1 + \lambda^2 x_2 + ... \tag{6.28}$$

  Then, the nonlinear term can be approximated as

$$f(x) \approx f_0 + \lambda f_1 + \lambda^2 f_2 \tag{6.29}$$

- **Discrete Empirical Interpolation Method (DEIM).** From a training input, N_k interpolation points of $(t_i^k, c_{1i}^k, ..., c_{di}^k), i = 1, 2, ..., N_k$,$N_k$ is the number of function products at iteration k. Then, we have

$$f(x^k) \approx \sum_{j=1}^{N_k} \mu^k T_i^k(t) \circ \prod_{j=1}^{d} P_i^{kj}(c_j) \tag{6.30}$$

  where μ^k is solved from a linear system at interpolation points. In **DEIM** method, the procedure of nonlinear approximation depending on training input is repeated until reaching convergence.

The standard **IL** method can handle limited types of non-linear term, such as p-order term x^p. Because it is necessary to evaluate non-linear term in a separated form, i.e. $x^p = (\sum_{i=1}^{n-1} T_i(t) \circ \prod_{j=1}^{d} P_i^j(c_j))^p$, the approximation will introduce $(n-1)^p$ terms which increase exponentially, and becomes inefficient. Although **ANM** method can resolve the computation to approximate high-order terms, the computation complexity increase greatly in multi-parametric case. In contrast to **IL** and **ANM** method, **DEIM** approach can handle any types of nonlinearity in principle but also suffer much complexity that it is not straightforward to handle strong nonlinearities. **PL** method is valid around the neighbor of the expansion point x_0, an equilibrium point. It becomes very inaccurate when the current point varies largely from x_0, so that it is limited to weakly nonlinear systems. Therefore, the nonlinear model reduction of the immersed dynamical system requires a nonlinear approximation method that can handle any type of nonlinearity efficiently.

To resolve this problem, a parameterized piecewise-linear approximation (PPWL) for the separated representation of immersed rigid-body dynamics is anticipated. In contrast to PL approximation, the PPWL method adopts a form of a weighted combination of linearized models at different expansion points. Furthermore, different with previous trajectory piecewise-linear approach (TPWL) for POD model reduction [RW06, DR08], this technique is proposed for parameterized PGD nonlinear model reduction. In the approach, the nonlinear function f in the dynamical model has the following piecewise-linear approximation.

$$\begin{aligned} f(x, c_1, ..., c_d) &= \sum_{j=0}^{d} c_j f_j(x) \\ &\approx \sum_{j=0}^{d} c_j (\sum_{i=0}^{s-1} w_{ij}(x)(f_j(x_i) + J_{ij}(x - x_i))) \end{aligned} \tag{6.31}$$

where x denotes $X(t)$. Considering the function f_j individually, x_i are s linearization points along a training input as shown in Figure 6.9. $w_i(x)$ are state-dependent weights $(w_i(x) \geq 0, \sum_{i=0}^{s-1} w_i(x) = 1)$. J_i are the Jacobians of f evaluated at linearization points x_i.

Applying the model reduction of separated representation, the above approximation becomes a reduced form of individual $f_j(x)$:

$$f(x) = \sum_{i=0}^{s-1} w_i(X)(f_j(X_i) + J_i(X - X_i)) \tag{6.32}$$

where $X_i = X(t^i, c_1^i, .., c_d^i)$, the definition of X is given in Equation 6.1. For a reduced state X, the linearization points X_i are specified by the points $(t^i, c_1^i, .., c_d^i), i = 0, 1, ..., s$. In Equation 6.31, $w_i(X)$ are weights depending on the reduced order state X $(\sum_{i=0}^{s-1} w_i(X) =$

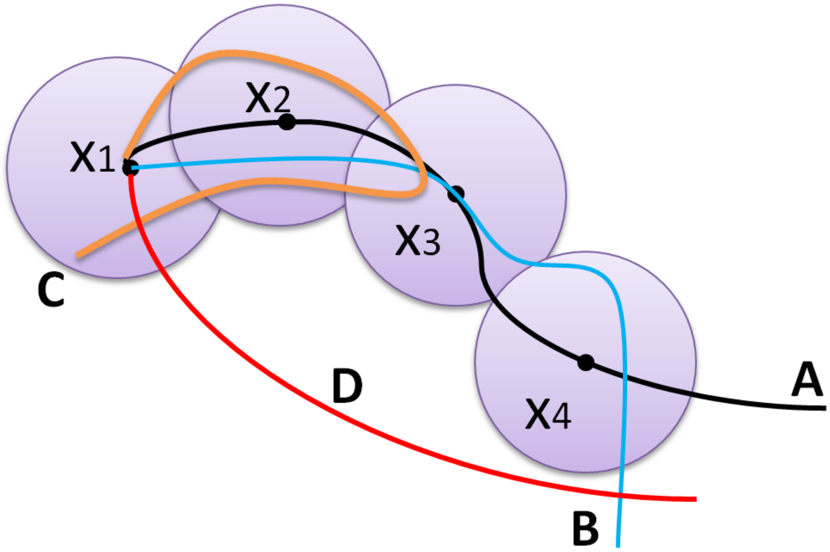


Figure 6.9: Weighted combination of linearized model at expansion points x_1, x_2, x_3 and x_4 in a 2D state space. The range of the blue circles present the valid ranges around expansion points. Line A denote the sampling points from a training input. Note that line B and C can be approximated well because they locate inside the weighted regions. In the contrary, red line D is badly represented.

1). There are three main issues to be resolved in the above reduced model: (1) Generation of the separated representation of X; (2) Selecting the linearization points X_i; (3) Computing the corresponding weights $w_i(X)$.

6.6 Discussion

This chapter introduced a new prior reduced model based on separated representations that do not require snapshots of complete solutions for dynamical systems. This method can reduce high dimensional problems and tackle different domains, i.e., temporal and spatial domains, physical parameters, and initial and boundary conditions as extra coordinates. This work has proposed a framework for separated representation on discrete and tensor formulations and a method to account for coupled terms in the proposed framework. The proposed method utilizes a fixed-point algorithm in an iterative process to control the desired accuracy of the problems under convergence.

The limitation of the proposed method is the difficulty in accounting for nonlinear and coupled terms. For complex models, especially strongly coupled problems, the proposed approach may fail because of the large amount of terms generated in the iterative process, i.e., $O(N^2)$. Other approaches, such as an asymptotic numerical method and discrete empirical interpolation method [CLB$^+$13a, CLB$^+$13b, PCA10] also suffer the same limitations. A promising solution for this issue is discussed in this chapter for nonlinear model reduction of immersed rigid body dynamics.

The proposed approach is efficient because the computation of the reduced model is only executed in a precomputed process. The preprocessed data can be saved as a codebook to search solutions for different control parameters and initial and boundary conditions, and the computation cost is only a few milliseconds. The proposed approaches can be applied to motion control, inverse identification, and parameter estimations for various physical simulations in real-time CG applications. The challenge of the physical simulations and their control problem for complex dynamics in CG is related to the physical control parameters, such as the coefficients of restitution and surface normals for rigid bodies [PSE03], stiffness and friction coefficients for deformable bodies [MTB+13], drag and lift coefficients for aerodynamics simulations [JWL+13], and the Reynolds number for flow simulations, which are usually designated by measured data in constant or curve forms. In the proposed method, all these parameters would be embedded in separated representations as extra coordinates to achieve realistic simulation results at low computational cost.

Chapter 7

Conclusion and future work

This chapter presents the conclusion and contributions of all the proposed methods in this dissertation. Then, the limitations of this study are analysed by comparing the proposed methods and evaluated quantitatively. Finally, the suggestion of the future work of immersed rigid body dynamics is discussed.

7.1 Conclusion

The main purpose of this study was to simulate immersed rigid body dynamics realistically and interactively. To achieve this purpose, this study proposed four different approaches and six functional modules. The graph-based method proposed a motion synthesis approach based on motion graph, which combines the motion patterns in physical experiments by a Markov chain model. The stochastic model considered the inertial and viscous effects of the surrounding flow numerically, where a Langevin model was successfully adopted in the dynamical model to represent the turbulent characteristic. The pattern-driven method combined the motion graph technique in graph-based method and the turbulent energy computation in stochastic model. This method proposed an inverse parameter estimation of force coefficients from the motion synthesis in parameter subspaces. Finally, a reduced model was proposed to represent the dynamical systems, which did not depend on the precomputed simulation data. This reduction algorithm is anticipated to solve strongly nonlinear problems.

This thesis has the following contributions to computer graphics and knowledge science.

Computer Graphics This study introduces a new and challenging topic into computer graphics community. To solve the trade-off issue for the simulations of immersed rigid-body dynamics in computer graphics: fluid effects on the body cannot be understood without solving the turbulent motions at the similar timescale with body; the task of simulating fluid motion is trivial while rendering the simulation scene in graphical application. This work proposes novel approaches to avoid the computation of fluid motion while accounting for the fluid effects.

For efficiently simulating and controlling the simulations of the proposed topics, the brand-new and different data-driven simulation approaches are proposed in this work as follows: graph-based method based on the phase change of the complex physical system; stochastic method based on turbulent model; pattern-driven method based on the properties of different parameter subspaces; a prior reduced order model of the body dynamics based on separated representation;.

The simulation techniques of immersed rigid-body dynamics are significant to achieve realistic physical simulations in computer animation, especially for the thin, sharp-edged or wing structure bodies, such as the falling motions of leaves, papers, petals, snowflakes, paper airplanes, and so on. Furthermore, it is a promising and fundamental topic for immersed body simulations of cloth and character simulations, such as the locomotion of moving cloth, flying birds and swimming fishes.

Knowledge Science As shown in Figure 7.1, the contributions of this study to knowledge science are summarized as follows:

Knowledge presentation combining physics and simulations Since the intricate complexity of immersed rigid dynamics, the implicit knowledge of the chaotic system includes the explicit knowledge of concrete dynamical models and the fuzzy knowledge of uncertain stochastic models. The simulation work in thesis combines both of them to account for fluid effects.

Knowledge discover from experiments and simulations This study discovers the knowledge of motion patterns from the domain knowledge of dynamical model in parameter spaces. By combining both experimental and numerical works, this study extracts a new problem-solving methodology for the traditionally unsolvable problems. The proposed methodology is helpful in the real-time simulation of complex dynamics using the computations from both online and offline processes.

Knowledge creation combining data and simulations This study aims at resolving the problems that arise from a nonlinear dynamical system, where different data types are preprocessed for pattern recognition and fast simulations, such as physical rules, motion patterns, and the reduced states. The new knowledge of the body dynamics is created from motion synthesis based on existed motion patterns.

Knowledge representation of dynamical systems A meta-model of dynamical systems is proposed to represent the existing knowledge of dynamical models. From a dynamical system in the formulations of ordinary or partial differential equations, the new model utilizes a separated representation to represent the solutions in temporal, spatial and parametric domains.

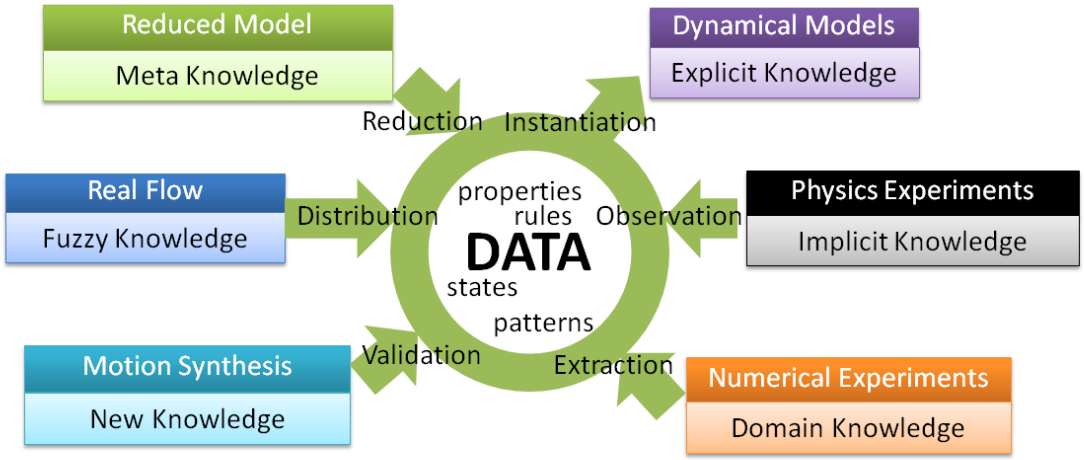


Figure 7.1: Global review from the view of knowledge science. The functional modules are corresponding to the explanations in Section 1.3.1.

7.2 Evaluation and limitation

As listed in Table 7.1, seven criteria are defined to evaluate all the proposed methods in this thesis: (1) *Adaptive Geometries* to evaluate whether the method can simulate complex objects; (2) *Data-independence* denotes the dependence on precomputed simulation data;(3) *Computation Cost* denotes whether a real-time simulation is feasible by the proposed method; (4) *Simulation Fidelity* for whether a realistic simulation is feasible by the proposed method;(5) *Controllability* indicates the proposed method can be adopted in controllable simulations; (6) *Simulation Stability* for the stable simulation outputs; (7) *Extensibility* declaims whether the proposed method can be adopted to the simulations of other bodies, such as cloth and articulated bodies. These criteria are given based on three aspects of one simulation approach: input (1)(2), output (3)(4), and algorithm (5)(6)(7).

Table 7.1: Comparisons of proposed methods in different criteria.

Functions	Graph-based	Stochastic model	Pattern Driven	Reduced model
Adaptive Geometries		√	√	√
Data-independence				√
Computation Cost	√	√	√	√
Simulation Fidelity	√		√	–
Controllability	√		√	–
Simulation Stability	√		√	√
Extensibility		√	√	√

The main limitation of graph-based method is the high dependence on the physical experimental results, so that it is difficult to adopt this method to simulate complex objects and other bodies that do not exist in the captured database. The stochastic method is a numerical method based on the precomputed turbulent energy. Due to the assumption in the turbulent model and its stochastic feature, the simulation results are not stable enough to capture different motion patterns observed in physical experiments. To overcome the limitations of these two methods, a pattern-driven method was proposed. This method also has a high dependence on the data from numerical experiments. To solve the data-dependence problems, a prior reduced model is proposed in a formulation of separated representation that does not rely on the results of numerical simulations.

To evaluate the simulation results by the proposed methods in this thesis, a questionnaire is executed to analyse the difficulty, similarity, and regularity of the proposed immersed rigid body simulations quantitatively. The participants include graduate students with computer science background and the experts in computer animations. We collected $N = 19$ valid samples from totally 21 samples where two samples are discarded due to their singularities among all the samples. All questions in the questionnaire are designed into two parts: fundamental questions and evaluation questions.

Fundamental questions are designed to discriminate the participants' knowledge background and their perceptual knowledge of the motion patterns of immersed rigid body dynamics. The knowledge background is assessed by five-stage strategy with a question "Are you familiar with graphical programming of game development (animation, physics engines, etc.), fluid mechanics (aerodynamics, etc.), or graphics tools (Maya, Autodesk, Blender, etc.)" where 1 for not familiar and 5 for very sophisticated. The value of this assessment is calculated as the weighting coefficient w_i for each participant p_i in our evaluation. To obtain the perceptual knowledge of the immersed rigid body dynamics of falling leaves, the other question is designed by multiple checkboxes of six motion patterns corresponding to Figure 3.1 as shown in Figure 7.2. Among all valid samples, 31.6% samples recognize the motion transitions among motion patterns where their values are calculated by fractions for each motion patterns. The proportions for steady, tumbling, chaotic, fluttering, helix, and spiral motions are 7.7%, 23.1%, 15.4%, 30.8%, 7.7%, and 15.3%, respectively.

According the phase diagram of immersed rigid body dynamics (Figure 3.3), the most stable motion of a falling leaf is fluttering by calculated Reynolds number and dimensionless moment of inertia as described in Section 5.4.2, which is corresponding the largest proportion of fluttering in Figure 7.2. From the explanations in Section 3.3.1, we know that the chaotic and helix motions are transitional motions which are difficult to be observable. For this reason, the proportions of chaotic and helix motions are less than their neighbours, i.e., tumbling, fluttering and spiral motions. The steady descent motion occupies the least proportion which is compliance with

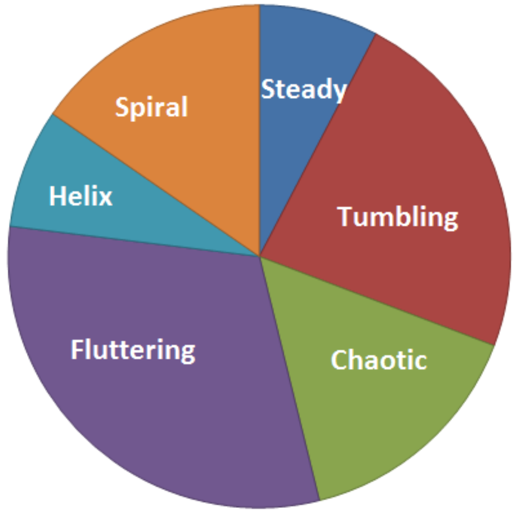


Figure 7.2: Perceptual knowledge of immersed rigid body dynamics.

the phase diagram.

Furthermore, the discovery of the perceptual knowledge of immersed rigid body dynamics from participants is compliance with the transition probability from our physical experiments (Section 3.4.2 and Section 5.5.2) and the sampling results from our numerical experiments (Section 5.4.3). In other words, the participants have the ability to distinguish different motion patterns in evaluation questions, and the common user can notice the artefacts of the simulation of an immersed rigid body that the proposed approaches in this thesis are significant to improve the simulation quality for graphical applications.

Evaluation questions are designed for evaluating the simulation results by the proposed approaches in this thesis. The evaluations are assessed by comparing the capture videos and simulation results in five-stage strategy. As shown in Figure 7.3, three criteria are evaluated for the proposed graph-based, stochastic model and pattern-driven approaches: difficulty, similarity, and regularity. Difficulty is evaluated by the question "do you think it is easy to simulate the motion of lightweight objects (leaf, paper, etc.)?" for participants with knowledge background, similarity by the question "is the simulation result similar to captured video?", and regularity by the question "are the motion patterns in the simulation result apparent?". All the evaluated value E is calculated in the following equation with weighting coefficients.

$$E = \frac{\sum_i^N w_i E_i}{\sum_i^N w_i} \tag{7.1}$$

The value of difficulty is 4.49 (5 for very difficult) in this assessment that means the immersed rigid dynamics simulation is not easy to be simulated in common sense for the reasons analysed in Section 1.2. For similarity and regularity, the simulation results of stochastic model are assessed in lower scores (mean similarity 3.64 and mean regularity 3.4) than other approaches in Figure 7.3 because no motion patterns is considered in this method, meanwhile, the pattern-driven approach (mean similarity 4.11 and mean regularity 3.9) has the similar score of graph-based approach (mean similarity 4.24 and mean regularity 4.0) due to the usage of motion graph technique. Note that G1 has the highest scores (similarity 4.53 and regularity 4.67, and 5 for very similar and very apparent) because there is only one motion pattern (fluttering) for coin moving underwater in our simulation result. The evaluation results are compliance with the analysis in Table 7.1.

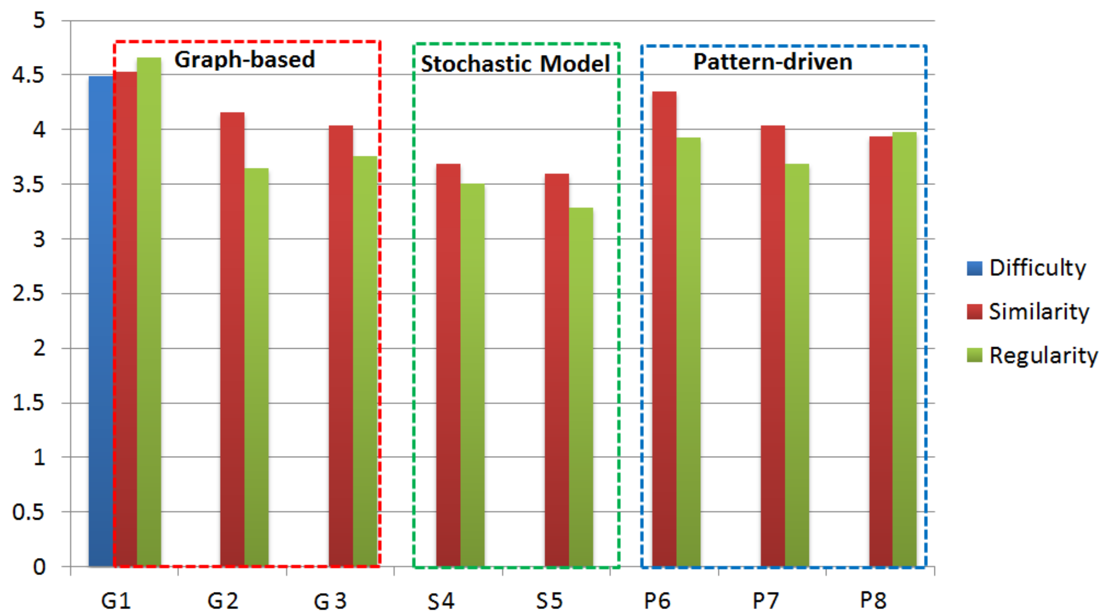


Figure 7.3: Evaluations of the simulations results by the proposed approaches. G1, G2, G3, S4, S5, S6, P7, and P8 are corresponding to the simulation results in Figures 3.15, 3.16, 3.17, 4.4, 4.5, 5.5, 5.14, and 5.16, respectively.

7.3 Future work

The proposed methods of immersed rigid body dynamics could be improved in following aspects as future work.

Model-driven simulation A model-driven simulation is to combine the proposed reduced model with the dynamical models in the other proposed methods. In contrast to the complete data-driven methods, this simulation method depends on the accurate model definition rather that the simulated data.

Controllable simulation A designer-friendly controllable simulation that the motions of the body follow a sketch of designed trajectory is an interesting application. The control of immersed rigid body simulations could be realized by combining the proposed high-level energy optimization with multiple-shooting optimization on the dynamical states of the body. In contrast to the discontinuities due to collisions in motion sketching method [PSE03], the motion transitions at high curvature points initiate the motion discontinuity similarly.

Multibody simulation The collision detection problem was omitted in all the proposed methods. It is promising to combining the proposed algorithms with particle systems to improve the simulation levels of suspension flows, such as dust, snowfall, confetti, and so on.

Besides of rigid body dynamics, it is challenging and exciting to resolve the topic of *immersed body dynamics* of different objects following the methodology in this thesis.

Bibliography

[AF02] Okan Arikan and D. A. Forsyth. Interactive motion generation from examples. *ACM Trans. Graph.*, 21:483–490, July 2002.

[AHN04] Kohta Aoki, Osamu Hasegawa, and Hiroshi Nagahashi. Behavior learning and animation synthesis of falling flat objects. *JACIII*, 8(2):223–230, 2004.

[AMF13] Franck Auguste, Jacques Magnaudet, and David Fabre. Falling styles of disks. *Journal of Fluid Mechanics*, 719:388–405, 2 2013.

[APW05a] A. Andersen, U. Pesavento, and Z. J. Wang. Analysis of transitions between fluttering, tumbling and steady descent of falling cards. *Journal of Fluid Mechanics*, 541:91–104, October 2005.

[APW05b] A. Andersen, U. Pesavento, and Z. J. Wang. Unsteady aerodynamics of fluttering and tumbling plates. *Journal of Fluid Mechanics*, 541:65–90, October 2005.

[Bar93] David Baraff. Non-penetrating rigid body simulation. In *EUROGRAPHICS State of the Art Reports*. Eurographics Association, 1993.

[BBB07] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid fluid coupling. *ACM Trans. Graph.*, 26(3), July 2007.

[BEM98] A. Belmonte, H. Eisenberg, and E. Moses. From flutter to tumble: Inertial drag and froude similarity in falling paper. *Physical Review Letters*, 81:345–348, July 1998.

[BETC12] Jan Bender, Kenny Erleben, Jeff Trinkle, and Erwin Coumans. Interactive simulation of rigid body dynamics in computer graphics. In *EUROGRAPHICS 2012 State of the Art Reports*. Eurographics Association, 2012.

[BH06] Arie Biesheuvel and Rob Hagmeijer. On the force on a body moving in a fluid. *Fluid Dynamics Research*, 38(10):716 – 742, 2006.

[BM05] G. Beylkin and M. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

[BTT09] Markus Becker, Hendrik Tessendorf, and Matthias Teschner. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):493–503, May 2009.

[CBD13] Marcin Chrust, Gilles Bouchet, and Jan Dusek. Numerical simulation of the dynamics of freely falling discs. *Physics of Fluids*, 25(4):044102:1–24, 2013.

[CBP05] Simon Clavet, Philippe Beaudoin, and Pierre Poulin. Particle-based viscoelastic fluid simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pages 219–228, New York, NY, USA, 2005. ACM.

[CFW99] Jim X. Chen, Xiadong Fu, and J. Wegman. Real-time simulation of dust behavior generated by a fast traveling vehicle. *ACM Trans. Model. Comput. Simul.*, 9(2):81–104, April 1999.

[CKL14] F. Chinesta, R. Keunings, and A. Leygue. The proper generalized decomposition for advanced numerical simulations:a primer. *Springer International Publishing*, 2014.

[CLB+13a] F. Chinesta, A. Leygue, M. Beringhier, L. Nguyen, J. Grandidier, B. Schrefler, and F. Pesavento. Towards a framework for non-linear thermal models in shell domains. *International Journal of Numerical Methods for Heat and Fluid Flow*, 23(1):55–73, 2013.

[CLB+13b] F. Chinesta, A. Leygue, F. Bordeu, J.V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta. Pgd-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20(1):31–59, 2013.

[CM10] Nuttapong Chentanez and Matthias Müller. Real-time simulation of large bodies of water with small scale details. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 197–206, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[CMT04] Mark Carlson, Peter J. Mucha, and Greg Turk. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Trans. Graph.*, 23(3):377–384, August 2004.

[CZY11] Fan Chen, Ye Zhao, and Zhi Yuan. Langevin particle: A self-adaptive lagrangian primitive for flow simulation enhancement. *Computer Graphics Forum*, 30(2):435–444, 2011.

[DAA11] A. Dumon, C. Allery, and A. Ammar. Proper general decomposition for the resolution of navier stokes equations. *Journal of Computational Physics*, 230(4):1387 – 1407, 2011.

[DAA13] A. Dumon, C. Allery, and A. Ammar. Proper generalized decomposition method for incompressible navier-stokes equations with a spectral discretization. *Applied Mathematics and Computation*, 219(15):8145 – 8162, 2013.

[DR08] Ning Dong and J. Roychowdhury. General-purpose nonlinear model-order reduction using piecewise-polynomial representations. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(2):249–264, Feb 2008.

[ERFM12] Patricia Ern, Frederic Risso, David Fabre, and Jacques Magnaudet. Wake-induced oscillatory paths of bodies freely rising or falling in fluids. *Annual Review of Fluid Mechanics*, 44(1):97–121, 2012.

[FERM08] P.C. Fernandes, P. Ern, F. Risso, and J. Magnaudet. Dynamics of axisymmetric bodies rising along a zigzag path. *Journal of Fluid Mechanics*, 606:209–223, 6 2008.

[FKMN97] S. B. Field, M. Klaus, M. G. Moore, and F. Nori. Chaotic dynamics of falling disks. *Nature*, 388:252–254, July 1997.

[GMP+12] D. Gonzalez, F. Masson, F. Poulhaon, A. Leygue, E. Cueto, and F. Chinesta. Proper generalized decomposition based dynamic data driven inverse identification. *Mathematics and Computers in Simulation*, 82(9):1677 – 1695, 2012.

[GSLF05] Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph.*, 24(3):973–981, July 2005.

[HFR06] William Van Haevre, Fabian Di Fiore, and Frank Van Reeth. Physically-based driven tree animations. In *NPH*, pages 75–82, 2006.

[HLW+12] Xiaowei He, Ning Liu, Guoping Wang, Fengjun Zhang, Sheng Li, Songdong Shao, and Hongan Wang. Staggered meshless solid-fluid coupling. *ACM Trans. Graph.*, 31(6):149:1–149:12, November 2012.

[HLW+13] Wentao Huang, Hong Liu, Fuxin Wang, Junqi Wu, and H. P. Zhang. Experimetal study of a freely falling plate with an inhomogeneous mass distribution. *Phys. Rev. E*, 88:053008, Nov 2013.

[HLYK08] Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. Bubbles alive. *ACM Trans. Graph.*, 27(3):48:1–48:4, August 2008.

[How95] M. S. Howe. On the force and moment on a body in an incompressible fluid, with application to rigid bodies and bubbles at high and low reynolds numbers.

The Quarterly Journal of Mechanics and Applied Mathematics, 48(3):401–426, 1995.

[HQHQ10] Li Haiyan, Sun Qingjie, Zhang Hui, and Zhang Qin. Example-based motion generation of falling leaf. In *Computer Design and Applications (ICCDA), 2010 International Conference on*, volume 5, pages V5–217 –V5–221, june 2010.

[HS67] J.L. Hess and A.M.O. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aerospace Sciences*, 8(0):1 – 138, 1967.

[HW10] M. HOROWITZ and C. H. K. WILLIAMSON. The effect of reynolds number on the dynamics and wakes of freely rising and falling spheres. *Journal of Fluid Mechanics*, 651:251–294, March 2010.

[HW14] Ruifeng Hu and Lifeng Wang. Motion transitions of falling plates via quasisteady aerodynamics. *Phys. Rev. E*, 90:013020, Jul 2014.

[JJE13] Barbara M. Johnston, Peter R. Johnston, and David Elliott. A new method for the numerical evaluation of nearly singular integrals on triangular elements in the 3d boundary element method. *J. Comput. Appl. Math.*, 245:148–161, June 2013.

[JK03] D. Jame and F. Kayvon. Precomputing interactive dynamic deformable scenes. *ACM Trans. Graph.*, 22(3):879–887, July 2003.

[JWL$^+$13] Eunjung Ju, Jungdam Won, Jehee Lee, Byungkuk Choi, Junyong Noh, and Min Gyu Choi. Data-driven control of flapping flight. *ACM Trans. Graph.*, 32(5):151:1–151:12, October 2013.

[JX08] Changqiu Jin and Kun Xu. Numerical study of the unsteady aerodynamics of freely falling plates. *Computer Physics Communications*, 3:834–851, 2008.

[KCD09] Marin Kobilarov, Keenan Crane, and Mathieu Desbrun. Lie group integrators for animation and control of vehicles. *ACM Trans. Graph.*, 28(2):16:1–16:14, May 2009.

[Kel11] G.I. Kelbaliyev. Drag coefficients of variously shaped solid particles, drops, and bubbles. *Theoretical Foundations of Chemical Engineering*, 45(3):248–266, 2011.

[KGC11] O. Khorloo, Z. Gunjee, and N. Chiba. Wind field synthesis for animating wind-induced vibration. *The International Journal of Virtual Reality*, 10:53–60, 2011.

[KGP02] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Trans. Graph.*, 21:473–482, July 2002.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KS10] Dmitry Kolomenskiy and Kai Schneider. Numerical simulations of falling leaves using a pseudo-spectral method with volume penalization. *Theoretical and Computational Fluid Dynamics*, 24(1-4):169–173, 2010.

[Lam75] Horace Sir Lamb. *Hydrodynamics*. Cambridge University Press, 1975.

[Lig86] James Lighthill. An informal introduction to theoretical fluid mechanics. 1986.

[LS74] B. E. Launder and B. I. Sharma. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters Heat Mass Transfer*, 1:131–137, December 1974.

[LZK+04] M. S. Langer, L. Zhang, A. W. Klein, A. Bhatia, J. Pereira, and D. Rekhi. A spectral-particle hybrid method for rendering falling snow. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*, EGSR'04, pages 217–226, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[Max90] James Clerk Maxwell. *The scientific papers of James Clerk Maxwell*. Dover publication, New York, 1890.

[MD04] M. Makino and M. Doi. Brownian Motion of a Particle of General Shape in Newtonian Fluid. *Journal of the Physical Society of Japan*, 73:2739, October 2004.

[MLS09] Sebastien Michelin and StefanG. Llewellyn Smith. An unsteady point vortex method for coupled fluid solid problems. *Theoretical and Computational Fluid Dynamics*, 23(2):127–153, 2009.

[MM02] G Mougin and J Magnaudet. The generalized kirchhoff equations and their application to the interaction between a rigid body and an arbitrary time-dependent viscous flow. *International Journal of Multiphase Flow*, 28(11):1837 – 1851, 2002.

[MMS09] Viorel Mihalef, Dimitris Metaxas, and Mark Sussman. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Computer Graphics Forum*, 28(2):229–238, 2009.

[MR10] Matthias Mando and Lasse Rosendahl. On the motion of non-spherical particles at high reynolds number. *Powder Technology*, 202(1):1 – 13, 2010.

[MRS99] L. Mahadevan, W. S. Ryu, and A. D. T. Samuel. Tumbling cards. *Physics of Fluids*, 11:1–3, January 1999.

[MTB+13] E. Miguel, R. Tamstorf, D. Bradley, Sara C. Schvartzman, B. Thomaszewski,

B. Bickel, W. Matusik, S. Marschner, and Miguel A. Otaduy. Modeling and estimation of internal friction in cloth. *ACM Trans. Graph.*, 32(6):212:1–212:10, November 2013.

[New87] I. Newton. *Mathematical Principles of Natural Philosophy.* transl. B. Cohen & A. Whitman, University of California Press, this edition published 1999, Berkeley, USA, 1687.

[New77] John Nicholas Newman. *Marine hydrodynamics.* MIT Press, 1977.

[Nis02] N. Nishimura. Fast multipole accelerated boundary integral equation methods. *Applied Mechanics Reviews*, 55(4):299–324, 2002.

[NO13] Michael B. Nielsen and Ole Osterby. A two-continua approach to eulerian simulation of water spray. *ACM Trans. Graph.*, 32(4):67:1–67:10, July 2013.

[OFM09] Taichi Okamoto, Makoto Fujisawa, and Kenjiro T. Miura. An interactive simulation system for flying japanese kites. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, Sandbox '09, pages 47–53, New York, NY, USA, 2009. ACM.

[OKRC10] Oktar Ozgen, Marcelo Kallmann, Lynnette Es Ramirez, and Carlos Fm Coimbra. Underwater cloth simulation with fractional derivatives. *ACM Trans. Graph.*, 29(3):23:1–23:9, July 2010.

[OLH84] H. R. Olesen, S. E. Larsen, and J. Hojstrup. Modelling velocity spectra in the lower part of the planetary boundary layer. *Boundary-Layer Meteorology*, 29:285–312, 1984.

[OTF+04] Shin Ota, Machiko Tamura, Tadahiro Fujimoto, Kazunobu Muraoka, and Norishige Chiba. A hybrid method for real-time animation of trees swaying in wind fields. *The Visual Computer*, 20:613–623, 2004.

[Par12] R. Parent. Computer animation: algorithms and techniques, third edition. 2012.

[PCA10] E. Pruliere, F. Chinesta, and A. Ammar. On the deterministic solution of multidimensional parametric models using the proper generalized decomposition. *Mathematics and Computers in Simulation*, 81(4):791 – 810, 2010.

[PM11] P. Paolettia and L. Mahadevan. Planar controlled gliding, tumbling and descent. *Journal of Fluid Mechanics*, 689:489–516, Dec 2011.

[Pop83] S. B. Pope. A lagrangian two-time probability density function equation for inhomogeneous turbulent flows. *Physics of Fluids*, 26(12):3448–3450, 1983.

[Pop11] Stephen B. Pope. Simple models of turbulent flows. *Physics of Fluids*, 23(1):011301, 2011.

[PSE03] Jovan Popović, Steven M. Seitz, and Michael Erdmann. Motion sketching for control of rigid-body simulations. *ACM Trans. Graph.*, 22(4):1034–1054, October 2003.

[PTC+10] Tobias Pfaff, Nils Thuerey, Jonathan Cohen, Sarah Tariq, and Markus Gross. Scalable fluid simulation using anisotropic turbulence particles. *ACM Trans. Graph.*, 29(6):174:1–174:8, December 2010.

[PTSG09] Tobias Pfaff, Nils Thuerey, Andrew Selle, and Markus Gross. Synthetic turbulence using artificial boundary layers. *ACM Trans. Graph.*, 28(5):121:1–121:10, December 2009.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd Edition)*. 1992.

[PW04] Umberto Pesavento and Z. Jane Wang. Falling paper: Navier-stokes solutions, model of fluid forces, and center of mass elevation. *Phys. Rev. Lett.*, 93:144501, Sep 2004.

[Raz10] P. Razavi. On the motion of falling leaves. *ArXiv e-prints*, July 2010.

[Ria35] Dimitri Pavlovitch Riabouchinsky. Thirty Years of Theoretical and Experimental Research in Fluid Mechanics. 1935.

[RMSG+08] Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph.*, 27(3):46:1–46:9, August 2008.

[RP01] L.-M. Reissell and Dinesh K. Pai. Modeling stochastic dynamical systems for interactive simulation. *Comput. Graph. Forum*, 20(3), 2001.

[RW06] Michal Rewienski and Jacob White. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear Algebra and its Applications*, 415(2?3):426 – 454, 2006. Special Issue on Order Reduction of Large-Scale Systems.

[SDG69] G. E. Stringham, Simons Daryl, and Harold Guy. The behavior of large particles falling in quiescent liquids. *GEOL SURV PROF PAP 562-C*, 562:C1–C36, 1969.

[SE93] Jos Stam and Fiu Eugene. Turbulent wind fields for gaseous phenomena. SIGGRAPH '93, pages 369–376, New York, NY, USA, 1993. ACM.

[SF92] Mikio Shinya and Alain Fournier. Stochastic motion-motion under the influence of wind. *Computer Graphics Forum*, 11(3):119–128, 1992.

[SJ13] A. Selino and M. D. Jones. Large and small eddies matter: Animating trees in wind using coarse fluid simulation and synthetic turbulence. *Computer Graphics Forum*, 32(1):75–84, 2013.

[Smi71] E. H. Smith. Autorotating wings: an experimental investigation. *Journal of Fluid Mechanics*, 50:513–534, 1971.

[SSP07] Barbara Solenthaler, Jurg Schlafli, and Renato Pajarola. A unified particle model for fluid solid interactions. *Comput. Animat. Virtual Worlds*, 18(1):69–82, February 2007.

[SYWC05] Lin Shi, Yizhou Yu, Christopher Wojtan, and Stephen Chenney. Controllable motion synthesis in a gaseous medium. *The Visual Computer*, 21(7):474–487, 2005.

[TK94] Yoshihiro Tanabe and Kunihiko Kaneko. Behavior of a falling paper. *Phys. Rev. Lett.*, 73:1372–1375, Sep 1994.

[TL64] R. H. Thuillier and U. O. Lappe. Wind and temperature profile characteristics from observations on a 1400 ft tower. *Journal of Applied Meteorology*, 3:299–306, June 1964.

[TLP06] Adrien Treuille, Andrew Lewis, and Zoran Popović. Model reduction for real-time fluids. *ACM Trans. Graph.*, 25(3):826–834, July 2006.

[UKSI14] Nobuyuki Umetani, Yuki Koyama, Ryan Schmidt, and Takeo Igarashi. Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph.*, 33(4):65:1–65:10, July 2014.

[Umb05] Pesavento Umberto. *Unsteady aerodynamics of falling plates.* PhD thesis, Cornell University, 2005.

[UO30] G. E. Uhlenbeck and L. S. Ornstein. On the theory of the brownian motion. *Phys. Rev.*, 36:823–841, Sep 1930.

[VB08] Pere-Pau Vázquez and Marcos Balsa. Rendering falling leaves on graphics hardware. *Journal of Virtual Reality and Broadcasting*, 5(2), March 2008.

[VBL09] C.H.J. Veldhuis, A. Biesheuvel, and D. Lohse. Freely rising light solid spheres. *International Journal of Multiphase Flow*, 35(4):312 – 322, 2009.

[VCW12] Kapil Varshney, Song Chang, and Z Jane Wang. The kinematics of falling maple seeds and the initial transition to a helical motion. *Nonlinearity*, 25(1):C1, 2012.

[VCW13] Kapil Varshney, Song Chang, and Z. Jane Wang. Unsteady aerodynamic forces and torques on falling parallelograms in coupled tumbling-helical motions. *Phys. Rev. E*, 87:053021, May 2013.

[Wei98] Peter Weiss. The puzzle of flutter and tumble: Physicists reconsider the fall of leaves. *Science News*, 154(18):285–287, 1998.

[WHH64] W. W. Willmarth, N. E. Hawk, and R. L. Harvey. Steady and Unsteady Motions and Wakes of Freely Falling Disks. *Physics of Fluids*, 7:197–208, February 1964.

[WHXW13] W. B. Wang, R. F. Hu, S. J. Xu, and Z. N. Wu. Influence of aspect ratio on tumbling plates. *Journal of Fluid Mechanics*, 733:650–679, 2013.

[WP03] Jia-chi Wu and Zoran Popović. Realistic modeling of bird flight animations. *ACM Trans. Graph.*, 22(3):888–895, July 2003.

[WP12] Steffen Weissmann and Ulrich Pinkall. Underwater rigid body dynamics. *ACM Trans. Graph.*, 31(4):104:1–104:7, July 2012.

[WZF+04] Xiaoming Wei, Ye Zhao, Zhe Fan, Wei Li, Feng Qiu, Suzanne Yoakum-Stover, and Arie E. Kaufman. Lattice-based flow field modeling. *IEEE Transactions on Visualization and Computer Graphics*, 10:719–729, November 2004.

[XM11] Haoran Xie and Kazunori Miyata. Free fall motion synthesis. In *SIGGRAPH Asia 2011 Sketches*, SA '11, pages 6:1–6:2, New York, NY, USA, 2011. ACM.

[XM12] Haoran Xie and Kazunori Miyata. Realistic motion simulations of objects in free fall. In *Computer Graphics International 2012*, pages 1–11, 2012.

[XM13] Haoran Xie and Kazunori Miyata. Stochastic modeling of immersed rigid-body dynamics. In *SIGGRAPH Asia 2013 Technical Briefs*, SA '13, pages 12:1–12:4, New York, NY, USA, 2013. ACM.

[XM14a] Haoran Xie and Kazunori Miyata. Langevin rigid: Animating immersed rigid bodies in real-time. *Journal of the Society for Art and Science*, 13:1–10, 2014.

[XM14b] Haoran Xie and Kazunori Miyata. A prior reduced model of dynamical systems. In *Mathematical Progress in Expressive Image Synthesis 2014*, pages 55–64, 2014.

[XM14c] Haoran Xie and Kazunori Miyata. Real-time simulation of lightweight rigid bodies. *The Visual Computer*, 30(1):81–92, 2014.

[YCZ11] Zhi Yuan, Fan Chen, and Ye Zhao. Stochastic modeling of lightweight floating objects. In *Digital Media and Digital Content Management (DMDCM), 2011 Workshop on*, pages 190–193, May 2011.

[YS12] Jianming Yang and Frederick Stern. A simple and efficient direct forcing immersed boundary framework for fluid structure interactions. *Journal of Computational Physics*, 231(15):5029 – 5061, 2012.

[ZCL11] Hongjie Zhong, Shiyi Chen, and Cunbiao Lee. Experimental study of freely falling thin disks: Transition from planar zigzag to spiral. *Physics of Fluids*, 23(1), 2011.

[ZGB+11] Robert Zimmermann, Yoann Gasteuil, Mickael Bourgoin, Romain Volk, Alain Pumir, and Jean-Fran çois Pinton. Rotational intermittency and turbulence induced lift experienced by large particles in a turbulent flow. *Phys. Rev. Lett.*, 106:154501, Apr 2011.

[ZJW04] Michael H. Dickinson Z. J. Wang, James M. Birch. Unsteady forces and flows in low Reynolds number hovering flight: two-dimensional computations vs robotic wing experiments. *Journal of Experimental Biology*, 207:449–460, 2004.

[ZLS+13] Hongjie Zhong, Cunbiao Lee, Zhuang Su, Shiyi Chen, Mingde Zhou, and Jiezhi Wu. Experimental investigation of freely falling thin disks. part 1. the flow structures and reynolds number effects on the zigzag motion. *Journal of Fluid Mechanics*, 716:228–250, Feb 2013.

[ZMZvW12] Marian Zastawny, George Mallouppas, Fan Zhao, and Berend van Wachem. Derivation of drag and lift force and torque coefficients for non-spherical particles in flows. *International Journal of Multiphase Flow*, 39(0):227 – 239, 2012.

[ZZJ+07] Long Zhang, Yubo Zhang, Zhongding Jiang, Luying Li, Wei Chen, and Qunsheng Peng. Precomputing data-driven tree animation. *Journal of Visualization and Computer Animation*, 18(4-5):371–382, 2007.