**Title**
Monolithic Methods and Versatile Applications for Numerical Fluid--Structure Interaction Studies



**Author**
Yang, Ruizhi

**Publication Date**
2020

UNIVERSITY OF CALIFORNIA

Los Angeles

# Monolithic Methods and Versatile Applications for Numerical Fluid–Structure Interaction Studies

A dissertation submitted in partial satisfaction

of the requirements for the degree

Degree of Philosophy in Mechanical and Aerospace Engineering

by

Ruizhi Yang

2020

ABSTRACT OF THE DISSERTATION

**Monolithic Methods and Versatile Applications for Numerical Fluid–Structure Interaction Studies**

by

Ruizhi Yang
Degree of Philosophy in Mechanical and Aerospace Engineering
University of California, Los Angeles, 2020
Professor Jeffrey D. Eldredge, Chair

A monolithic method for numerically solving fluid–structure interactions between viscous flow on an unbounded domain and rigid body system is presented. The vorticity form of incompressible Navier–Stokes equations is solved on a uniform Cartesian grid with immersed boundary projection method. The Lattice Green's function based integrating factor technique developed by Liska and Colonius is used to analytically evaluate the viscous diffusion term. Half-explicit Runge–Kutta method is applied for both fluid and rigid body system in time marching. The overall fluid–body saddle point system is designed to be arranged in a way to reveal an added-mass term for viscous flow in the solving procedure of block LU decomposition. The added-mass term is inherently embedded into the coupled system, compared to its explicit use in previous fluid–structure interaction studies. Because the added-mass augments inertia of the body system, the algorithm is proved to be stable for arbitrary small density ratios including zero mass case. Interaction force and boundary conditions on the interface are kept implicitly and solved as internal variables , and no iteration process or parameter tuning is needed. Both passive and active motions of rigid body system are allowed, with joint constraint and active motions enforced through Lagrange multipliers. With a unified treatment, the interaction force on the immersed surface is also enforced through Lagrange multipliers. Multiple applications are demonstrated, including cylinder rising/falling under gravity with different mass ratios, actively oscillating airfoil in a free stream, free swimming of an articulated fish, and self-excited oscillation of an articulated flag. Comparisons are made with previous studies and good agreement is found. Another application of a two-component linked plates model is studies for physical behaviour of bifurcation, hysteresis and energy harvesting. The algorithm is versatile in the sense of implementation. All aforementioned applications can be set up with very little change in the code, benefit from the use of rigid body dynamics in spatial vector form and numerical techniques in this work.

The dissertation of Ruizhi Yang is approved.

Yongho Sungtaek Ju

Marcelo Chamecki

Xiaolin Zhong

Jeffrey D. Eldredge, Committee Chair

University of California, Los Angeles

2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# ACKNOWLEDGMENTS

No doubt I need to express my appreciation first to my advisor Prof. Jeff Eldredge for all the help he kindly offered, both academical and educational. I would have not persisted with my Ph.D study to this day without him. Words are too pale at the point to illustrate hardships all the way along. Secondly I would like to dedicate this thesis to two important women in my life: my mother Haibin, who has absolute faith in me all the time, and my significant other Jin, who accompanies me through this five years long journey with joy. Finally I want to give thanks to all my family members and friends, for their support and assistance from all aspects of life.

# VITA

| | |
|---|---|
| 2014 | B.S. (Aerospace Engineering) and Minor (Applied Mathematics), Beijing University of Aeronautics and Astronautics, Beijing, China |
| 2015 | M.S. (Mechanical Engineering), UCLA, Los Angeles, California. |
| 2015-2019 | Teaching Assistant, Mechanical and Aerospace Engineering Department, UCLA, Los Angeles, California. |

# CHAPTER 1

# Introduction

Fluid–structure interaction (FSI) problems, generally defined as solid structures interacting dynamically with internal or external fluids, are known for their strongly nonlinear multidisciplinary nature. Applications of FSI problems are investigated in a variety of fields including internal fluids for pipe system [14, 15], physiological flows in deformable boundaries [16, 17, 18], biological locomotion [19], aerodynamics stability and response [20], vibration of turbines [21], particle transportation and sedimentation [22, 23] etc. Theoretical analysis on these problems are not always possible while experimental analysis are usually limited to certain parameter range. As a remedy, numerous numerical methods are developed to simulate the multidisciplinary physics and their interactions. Solvers that are both accurate and stable are desired, in the sense of describing physics of each system and their coupling precisely with low and controllable errors in time marching, and reliable performance when structures are under large scale motion or structure density is either large or small compared to fluid. It is also notable that computation speed is important, for that high computation cost would make physical analysis in a parametric space search unaffordable. Besides the aforementioned features, we would like the solver to be flexible for applications and easy to implement. To solve these problems, it is of our interest to develop a versatile open source numerical solver using a monolithic coupling scheme. In the broad FSI topics, we constrain our interest for incompressible laminar viscous external flow of low to moderate Reynolds number, interacting with rigid body systems connected by joints. Meaningful physical problems still apply in this scope, such as free swimming of an articulated fish and vortex shedding by an airfoil, shown later in chapter 3.

One of the classification of FSI algorithms is based on the treatment of meshes, i.e. body conforming meshes or non-conforming meshes. Literally, body conforming mesh requires constructing fluid–body interface as part of the mesh, so that boundary conditions can be applied directly. For moving or deforming structures, re-meshing is then required every one or several time steps. On the other hand for non-conforming mesh, fluid computation is conveniently done on a fixed Cartesian gird. However, fluid–body interface needs to be handled with extra care. The difference in the ways of enforcing no-slip and no-flow-through boundary conditions on the interface results in different methods under the category of immersed boundary method, including the original immersed boundary method by Peskin [24], the direct forcing method [25], immersed interface method [26], and Lagrange multiplier based immersed boundary projection method [27] etc. Our method follows Taira and Colonius [27] in treating the immersed surface forcing term as Lagrange multipliers.

Another way to categorize FSI algorithms is by the ways of coupling, i.e. monolithic methods or partitioned methods. Monolithic methods accommodate both fluid and structure in the same mathematical form, advance both systems with the same time marching scheme and treat interaction terms implicitly. On the other hand, partitioned methods advance each system separately while bonding force and kinematic constraints on the interface explicitly. Partitioned methods are further divided into weakly-coupled method and strongly-coupled method, depending on the difference in accuracy and time stage to enforce interface conditions. We provide a more detailed review in section 1.1 for these numerical methods used in FSI studies. Besides these direct numerical simulation based methods, there also exists sets of modelling methods for fluid–structure interaction studies [28]. For reduced-order models like vortex particle methods [29], modal decomposition based methods [30] and smoothed particle hydrodynamics methods [31], we just provide some useful reference here without discussing details.

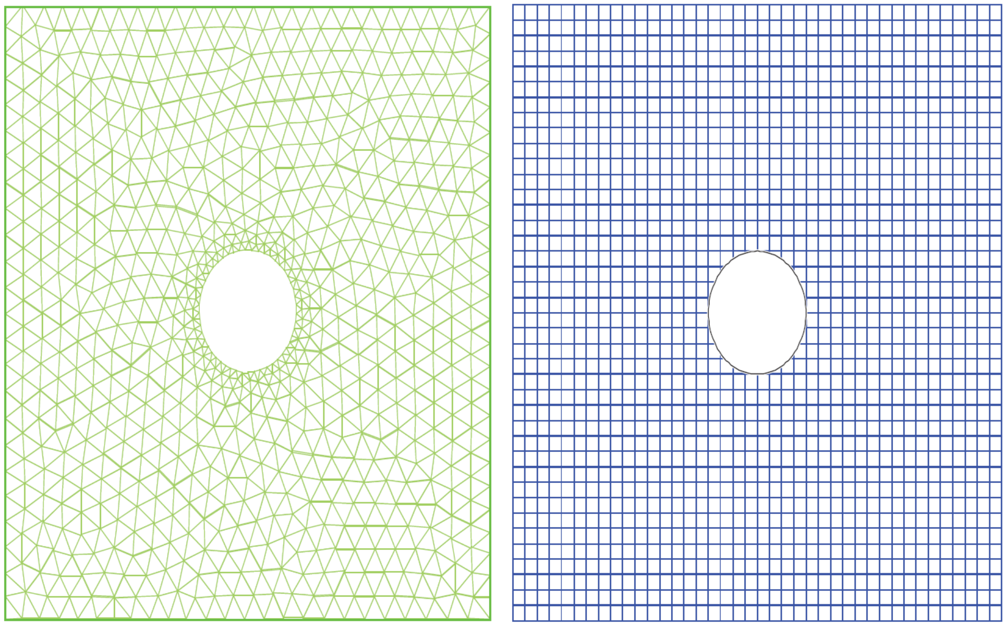
With the tools to compute fluid–structure interaction, we are now able to dig further into the physical aspect of vortex-induced vibration (VIV) problems. Rich dynamical behaviour of VIV problems, like the most well known von Kármán vortex street, gets reflected in many different systems (bluff bodies, flapping flags, oscillating airfoils etc.). Through fluid–structure interaction, shear layer separation and interaction evolves to form different vortex wakes, which is a good reveal of change in dynamical behaviours [2]. Characteristics such as bifurcation, lock-in and hysteresis are widely studied in either free oscillation or forced oscillation. The self-sustained body oscillation in flow also enlightens us of energy harvesting. Using model as simple as a mass-spring-damper system, mechanical energy from fluid can be transformed into other types of energy. These topics are reviewed in section 1.2 for more details.

## 1.1 Numerical methods for fluid–structure interaction

### 1.1.1 Conforming and non-conforming mesh methods

Figure 1.1 included in Hou et al. [1] is shown at the beginning of this section for an intuitive impression on conforming mesh and non-conforming mesh methods. For conforming mesh methods, there's no guarantee that meshes for the fluid domain and meshes for the structure are aligned. Usually fluid mesh at a large geometry curvature area or on the structure surface needs to be refined, while structure mesh needs to be refined at locations where stresses are potentially high, which usually is not on the surface. Also for moving structure, the interface conditions need to be in track to provide boundary conditions and exchange traction force information, thus mesh updating (re-meshing) is essential in the process. Data transfer between the two sets of meshes and mesh updating can be the key sources of error and ambiguity. To resolve these issues, compensations are made by the point match method, which picks common points from fluid mesh and body mesh as pivot, then determine the matching hierarchy of other points based on either shortest distance [32] or normal projection [33]. This relatively naive method suffers the problem of not conserving load transferred between the two physical systems, which is later fixed in Cebral and Lohner [34]. The other more complicated mesh matching technique is called the mortar method, which introduces an artificial thin shell containing fluid–structure interface for data transformation in Hou and Satyanarayana [35].

Between purely conforming and non-conforming mesh methods, there also exists a group



(a) (b)

Figure 1.1: Illustration of (a) conforming mesh (b) non-conforming mesh in Hou et al. [1]

of methods known as arbitrary Lagrangian-Eulerian (ALE) methods [36, 37], trying to combine benefits of both conforming and non-conforming meshes in the spatial discretization sense. Basically computation meshes in ALE methods don't have to be in line with fluid or structure. It is allowed to be held fixed in Eulerian manner, or move with material point in Lagrangian manner, with the introduction of a reference domain. In ALE methods, a relative velocity between the material point and the fixed mesh is computed to acquire mapping between both meshes. As a result, the material derivative needs to be modified to account for the mesh movement velocity $\boldsymbol{U}$.

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = \frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} - \boldsymbol{U}) \cdot \nabla \boldsymbol{v} \tag{1.1}$$

With features of both views combined, ALE methods perform well in the way of allowing greater distortions of continuous material than purely conforming meshes, meanwhile have better resolution near the surface compared to non-conforming meshes. It is also free of conservation violation problem, since governing equations for moving mesh are derived in a systematic way. With larger freedom for self-determined mesh, mesh generation algorithms are required in ALE formulation. One basic idea is to keep the mesh as regular as possible to reduce computation cost and avoid mesh entanglement [38], namely the mesh regularization technique. The other basic algorithm is called mesh adaptation technique [39], relying on the concentration of elements in solution zone with steep gradient. Since ALE methods still rely partially on body-conforming meshes, it can be categorized into conforming mesh methods as well. Conservation criteria for moving boundaries in ALE formulation has been discussed in Farhat et al. [40].

As pointed out in Hou et al. [1], coupling for most FSI methods involving conforming mesh aforementioned are based on the generalized Gauss–Seidel (GGS) approach [41], which is a partitioned method. Concretely speaking, an estimated interface location is first applied in solving for the fluid to get fluid pressure and stress on the structure. The structure solver then returns new kinematics, serving as new interface conditions. Iteration process is needed from here for convergence.

Generally speaking, conforming mesh methods endure with the complexity of meshes for complex body shape, together with the need to re-mesh frequently for moving or deforming bodies. The re-meshing algorithms can be complicated by themselves and are lack of common ground for different problems [42]. As an alternative, immersed boundary (IB) methods take the advantage of a non-conforming fixed Cartesian grid, regardless of body shape. IB method is first proposed in Peskin [24] to simulate blood flow in elastic heart valves, from then on numerous ramifications are developed. The difference in all the variations is associated with the imposition of boundary conditions by modifying the governing equations with a forcing term near the interface, since fluid domain are cut through by the interface [43]. Depending on the level to impose boundary conditions, IB methods are classified into continuous forcing methods and discrete forcing methods. Continuous forcing methods incorporate the forcing term in the continuous formulation of Navier–Stokes equation before spatial discretization, the original IB method [24] is in this category. By contrast, discrete forcing methods require spatial discretization first, then the forcing term is introduced in the spatial discretized formulation. Generally speaking, continuous forcing does not depend on spatial discretization, which is a big advantage, while discrete forcing can manipulate the numerical accuracy near the interface more directly.

The original IB method [24] explicitly modelled the continuous forcing term to be a constitutive relation with Hooke's law for the elastic boundary. Considering large stiffness for the elastic boundary to be the rigid limit, extension is made to rigid body [44, 45]. However this extension suffers from the stiff dynamical system brought by high spring stiffness [46]. Another extension to the original IB is to replace the Hooke's law by a damped oscillator to provide feedback control of surface velocity [47, 48]. These constitutive law related methods have solid background when dealing with elastic boundaries. But they involve in a parameters tuning process to avoid extreme stiffness, which is not very ideal. Also the use of constitutive laws are simplified models, and there's no promise of enforcing arbitrary desired boundary conditions with these models. Therefore, direct forcing method is proposed in [25, 49] as a subset of discrete forcing methods to implicitly impose the boundary condition. This is achieved by approximating the forcing term from an intermediate fluid velocity field, which can be interpreted by the penalization of slip velocity on the surface.

Another method in the category of discrete forcing is the immersed interface method (IIM) [26], where boundary force is computed directly from structural configuration. The boundary force is decomposed in normal and tangential directions, where the normal component is grouped into the pressure Poisson equation. Second order accuracy near the interface is achieved for IIM. With the desire for higher order accuracy near the interface for higher Reynolds numbers, other discrete forcing methods directly impose boundary conditions by modifying the stencil near the interface, with examples of sharp interface methods, i.e. ghost-cell finite-difference approach and cut-cell finite-volume approach. Ghost-cell method extends the fluid domain inside the body by one layer of ghost cells, and interpolates the interface condition onto that layer [50]. Simple interpolation can be in a bilinear (two-dimensional) or trilinear (three-dimensional) way, which is acceptable for laminar flows [51]. A higher order interpolation scheme is needed when the viscous sublayer is thinner at higher Reynolds numbers [50]. Cut-cell method is inherently secured on global and local conservation, thus serves as a good alternative to enforce boundary conditions. Basically cut-cell method reshapes a cell into either fluid part or solid part near the interface depending on the area ratio, and form control volumes from that. Overall, discrete forcing methods allow for a sharp solution in the vicinity of the boundary, which is desirable for high Reynolds number flows. However, methods like cut-cell method are difficult to extend to three-dimensional space, and prior knowledge of forcing term is usually needed to be provided.

A variation of continuous forcing becomes popular in more recently years, referred to as the immersed boundary projection method (IBPM) by Taira and Colonius [27, 52]. Adapting the idea to consider the boundary force as a Lagrange multiplier to enforce the boundary conditions [53], the boundary conditions are enforced in an implicit way. Remember that for incompressible flow, pressure serves as the Lagrange multiplier to enforce conservation of mass. Thus the concept of combining the boundary forcing and pressure together to be the general Lagrange multiplier unifies the solving procedure. If the vorticity-streamfunction form of governing equation is used, the divergence free condition is automatically satisfied, further simplifies the problem [52]. There also exists other extensions to the immersed boundary method, for example the fictitious domain method [53, 54]. The fictitious domain method is designed specifically to deal with body with volume, with Lagrange multipliers distributed at every grid point inside body with finite volume.

If we compare advantages and disadvantages of conforming methods and immersed boundary methods, each has its pros and cons. Conforming mesh methods near the interface generally have higher order of accuracy due to the ability to control grid resolution. Discrete forcing methods are also able to improve the accuracy near the interface by changing nearby stencil. Most continuous IB methods on the other hand, is only first order accurate near the vicinity of the interface. This can not be improved by using a higher order spatial discretization scheme or a smoother discrete delta function for regularization. It is related to the nature of discontinuous solution on the immersed surface [42]. The first order accuracy is durable for moderate Reynolds number flows, but not for higher Reynolds number flows. The more recent study in Stein et al. [55] proposed an immersed boundary smooth extension method based on the direct forcing IB method, and achieved higher order accuracy near the surface. The major advantage of IB methods lies in the fact of no need to generate conforming mesh with complicated process. Conforming grid generation even for a simple geometry might not be trivial, due to the balance of sufficient resolution and small number of total grid points. The influence of grid generation of a complex shape becomes obvious for stability and accuracy. Also there's no doubt that IB methods trumps conforming methods when it comes to moving bodies. The need of re-meshing and mesh projection from step to step greatly increases the computation cost and concern for accuracy problems. When comparing continuous forcing methods to discrete forcing methods, usually the movement of boundary makes it difficult for discrete forcing methods. For example, moving boundary in ghost-cell methods would have the cells previously inside body encounter fluid cells, which creates complexity. Based on our need to simulate fluid coupling with moving rigid body systems, continuous IB method with Lagrange multiplier representation should be a good choice.

### 1.1.2 Monolithic and partitioned methods

By the ways of coupling between fluid and body, FSI problems can be classified into monolithic methods (fully-coupled methods) and partitioned methods. Partitioned methods can be further classified into weakly-coupled methods and strong-coupled methods. A schematic illustration is shown in figure 1.2 for a quick impression. Monolithic methods are more ideal in the analytical sense, not only because multidisciplinary physics are described in the same mathematical framework, but also the whole system is solved using unified numerical schemes. Forces and kinematics on the fluid–body interface are treated implicitly in the monolithic coupling procedure, which means no estimation about interface is needed. Time marching schemes for both systems are kept consistent, which preserves the overall time accuracy in a rigorous way. The downside of monolithic methods is in the implementation level. It's not trivial to contain different physical systems in the same framework, since we should first find out ways to describe different systems as the same category of problems. In solving the monolithic problem, there could be issues due to scale difference, which might lead to poor matrix condition. Partitioned methods on the other hand, are easier to implement in this sense, since we are capable of keeping the two systems relatively separate except for information change at the interface. With legacy codes from both systems available, we only need to develop an additional interface model for the coupling. Concretely for fluid–structure interaction problems, fluid–body interaction force and body kinematics on the interface need to be enforced consistently in the ways of either sequentially (weakly-coupled) or iteratively (strongly-coupled). We will discuss applications of all aforementioned coupling schemes in more details.

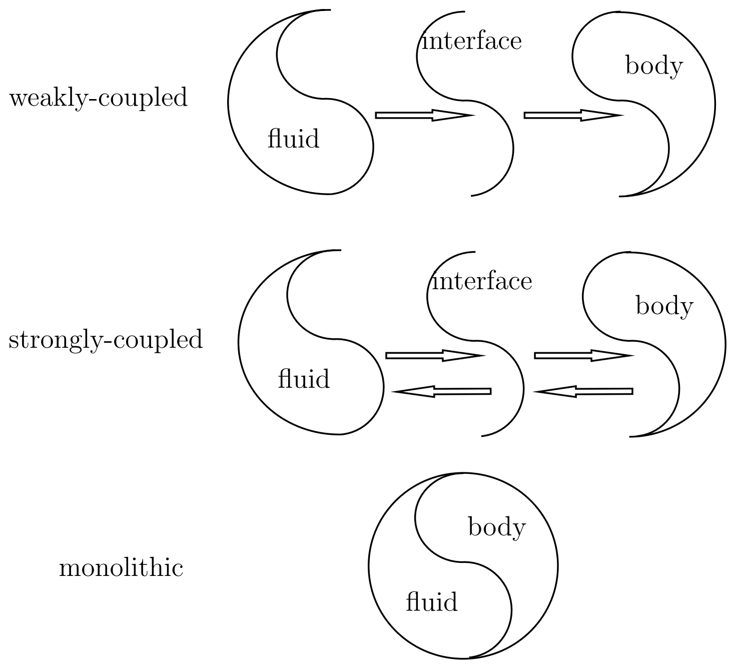



Figure 1.2: A schematic illustration for weakly-coupled, strongly-coupled and monolithic methods

Weakly-coupled method is first applied in many FSI problems due to its simplicity [56, 57]. This method solves both fluid and body equations in a one-pass manner for each time step. For example in figure 1.2, fluid system is solved first to get the fluid–body interaction force. The body solver then use this as an external force to solve for body kinematics. Due to the lack of iterations, there is actually no guarantee of convergence, especially shown in [58, 59] that added-mass effect would cause the instability in weakly-coupled approach. However weakly-coupled methods work surprisingly well for aeroelastic problems, where the structural deformation in one time step is relatively small. Typically, weakly-coupled methods are able to apply small time steps, since the computation load is low. Techniques called subcylcing are sometimes used, which allows the fluid solver to have smaller time steps than the body solver [60].

Compared to weakly-coupled methods, strongly-coupled methods involve in an iteration process to ensure convergence. However the iteration can be either computationally expensive if Newton–Raphson approach is used for fast convergence, or converging slowly if less expensive methods like Gauss–Seidel (fixed-point) iterations are used [61]. If the coupling between fluid and structure is strong, the iteration can still diverge even if a good initial guess is available. Modifying the body inertia matrix by manually considering the added-mass is proven to be effective in accelerating the convergence in Gauss–Seidel methods [58, 12]. Convergence rate can also be improved with relaxation schemes presented in [62, 13] for example. The coupling scheme generally achieves satisfying stability and convergence, meanwhile has easier implementation compared to monolithic methods. Thus it is widely used across multiple FSI problems [63, 64, 65, 66], involving in different techniques on treating meshes and different body systems. Particularly for the use of immersed boundary methods, Wang and Eldredge [13] applied a strongly-coupled scheme between viscous flow and rigid body systems. Goza and Colonius [67] also applied a strongly-coupled scheme, but for viscous flow coupling with elastic structures.

Although strongly-coupled methods use iterations for the convergence on the interface, it is still staggered in time level. Usually body kinematics are estimated to provide boundary conditions for the fluid, then fluid solver returns dynamical loading on the body, and iteration starts until convergence to a certain degree. It is believed that monolithic methods are free of this time lag problem, as presented in Blom [68] with comparison made between monolithic and strongly-coupled methods using a simple example: moving piston in an inviscid flow. In this paper monolithic methods show unconditional stability while strongly-coupled method is conditional stable within certain range of CFL numbers. Hübner et al. [69] presented a monolithic solver of elastic beam coupled with viscous flow. The fluid domain used a conforming mesh moving with the structure during interaction, as a result, the mesh conforming is considered as another constraint. To reduce computation cost, the set of nonlinear equations are linearized before a uniform time marching scheme is applied for both fluid and structure. Monolithic methods are applied in a set of coupling problems between elastic structures and viscous flow with ALE description [70, 61]. With the monolithic formulation in Heil [61], the coupled block system consists of diagonal fluid and body blocks, together with off-diagonal interaction block. The matrix is solved as a whole using Newton–Raphson iteration with preconditioners constructed by neglecting the fluid–structure interaction block. The block matrix is then solved with generalized minimal residual (GMRES) method. The difference between the method used in Heil [61] and current study in the matrix solving aspect is that we do not solve the block matrix as a whole. The utilization of block LU decomposition equips us with the benefit of easy physical interpolation of the solution procedure, also does not require an iterative matrix solver. In reviewing the use of monolithic methods for FSI problems, we do observe a lack of studies using immersed boundary methods. Our study could be a good supplementary material in this field.

## 1.2 Vortex induced vibrations

With the computation tool in hand for FSI problems, we are now ready to look into physical problems. Although our interest is limited to incompressible viscous laminar flow, there are still a set of meaningful physical studies revealing rich physical behaviours in this scope — the vortex induced vibrations. In this part of the introduction, we aim to review various kinds of vortex induced vibration (VIV) problems by different dynamical systems. Focus is put on dynamical response of the system interacting with fluid, and energy harvesting characteristics associated with it. Introduction to dynamical response is ordered by different types of dynamical systems (cylinder and bluff bodies, airfoil and flapping flags). Research on energy harvesting behaviour is also briefly mentioned.

### 1.2.1 Cylinder and Bluff Bodies

As one type of flow induced vibrations, vortex shedding leads to more severe oscillations than turbulence produced in the flow around a bluff body. Bearman [71] made a thorough review on experimental studies of vortex shedding by oscillating bluff bodies as a summary

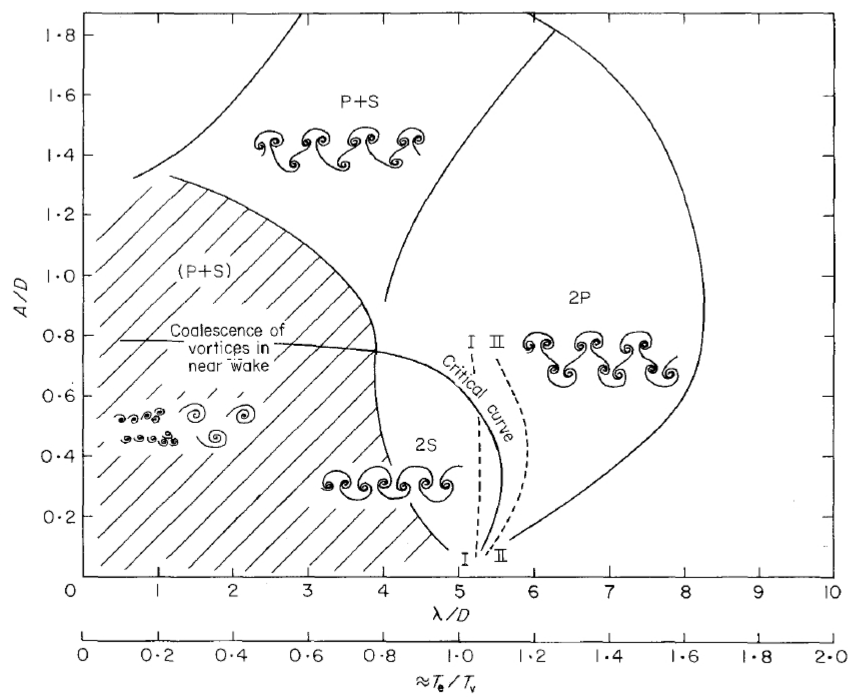



Figure 1.3: Vortex formation modes in the wake of a cylinder in Williamson and Roshko [2]

for early research. Bearman summarized the early understanding on both fixed and oscillating bluff bodies, including free oscillation, forced oscillation and flexibly mounted bluff body oscillation. With reduced velocity $U/fD$ defined out of uniform flow speed $U$, body oscillation frequency $f$ and cylinder diameter $D$, as the governing parameter, characteristics including hysteresis and lock-in are observed.

#### 1.2.1.1 Forced oscillation of cylinder

Williamson and Roshko [2] explored the vortex formation using forced oscillation of a cylinder in the free stream. Controlling parameters are chosen to be non-dimensional cylinder oscillation amplitude and wavelength ratio to fully specify the motion of the cylinder. Wavelength ratio is defined as $UT_e/D = \lambda/D$, equivalent to the definition of reduced velocity. Here $T_e$ represents the period of cylinder oscillation in the cross-flow direction and $\lambda$ represents the wavelength. Different vortex modes in the amplitude-wavelength map are reported in [2], shown in figure 1.3.

When oscillation amplitude and wavelength are both small, shed vortex coalesces in the near wake. As amplitude increases, vortex size becomes comparable to cylinder radius $D$, leading to the vortex shedding of 2S mode. With wavelength further increases beyond 2S mode, cylinder oscillation period increment gives extra time for a second vortex to shed in one half period. This is the transition from 2S mode to 2P mode. Lift on the cylinder could experience an abrupt change when experiencing this mode change, and both force phase and body motion could also jump when the mode change occurs. This hypothesis was used to explain hysteresis and confirmed by other studies later on. It also shows that 2P and 2S mode may exist at the same time corresponding to the same amplitude and wavelength.

#### 1.2.1.2 Free oscillation of an elastically mounted cylinder

Brika and Laneville [3] first proved the existence of 2P vortex wake mode by conducting experiments on cylinders with free oscillation in water. By plotting non-dimensional vibration amplitude versus reduced velocity, two branches revealing the hysteresis loop are shown in figure 1.4. The upper branch (later being called initial branch in Williamson and Govardhan [5]) is achieved by small progressive increment of flow velocity from cylinder staying at rest. Vortex mode according to this branch is the 2S mode, starting from the onset of synchronization ($U = 0.78$) to the upper critical velocity ($U = 1.01$), where an abrupt vortex wake mode change to 2P mode happens. The lower branch is achieved either by gradually decrement of flow velocity from a self-sustained state, or by releasing a stationary cylinder in the flow field with relatively high flow speed, covering the flow speed extending from the lower critical velocity ($U = 0.88$) to the end of synchronization ($U = 1.20$). The flow speed range for bi-stability to happen is pretty large with respect to later numerical simulation.

Khalak and Williamson [4] discovered the vortex wake mode of cylinder with free oscillations at low mass-damping coefficient $m^*\zeta$, where $m^*$ is body mass ratio to the fluid and $\zeta$ is a combination of body mass, spring stiffness and spring damping. When $m^*\zeta$ is small enough, synchronization range is larger and oscillation amplitude is higher compared to medium $m^*\zeta$ values. Furthermore, a new branch called the upper branch appears besides

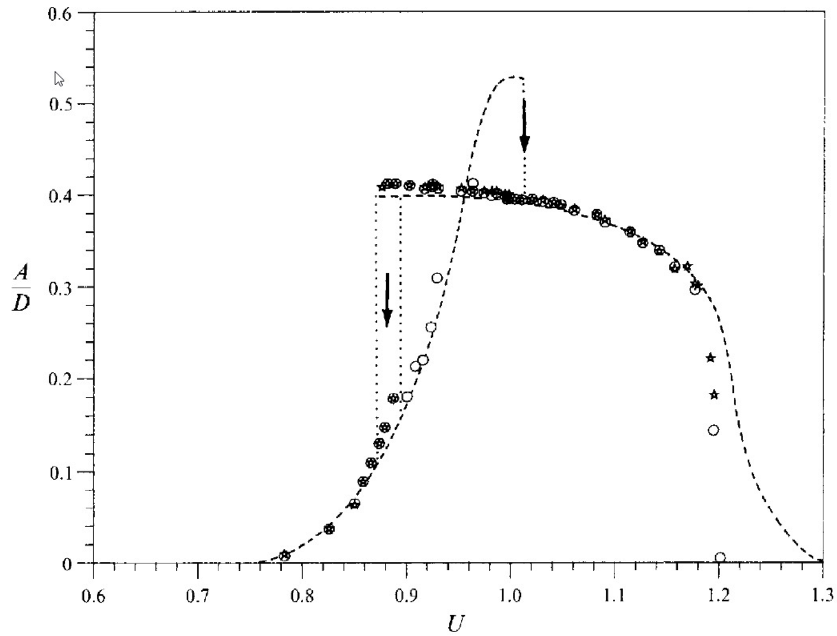



Figure 1.4: Relative vibration amplitude versus relative velocity in Brika and Laneville [3]

the initial and lower branch (shown in figure 1.5), for which the vortex wake mode is still comprised of 2P mode, but the second vortex of each pair is much weaker than the first one.

It is found that the oscillation amplitude relies on $m^*\zeta$, however the range of $U$ for synchronization primarily depends on $m^*$ only. One thing to mention here is that the tradition definition of synchronization (or lock-in) means the ratio between oscillation frequency and natural frequency of the fluid approaching unity, i.e. $f^* = f/f_N = 1$. But for different range of mass ratio, this ratio may deviate from one. For light bodies in water, the ratio $f^*$ departure from 1 is shown in figure 1.6. So the modified definition for lock-in should be the consistency between oscillation frequency and wake vortex frequency, or the consistency between oscillation frequency and body force frequency. By using Hilbert transformation, it is shown that the transition between initial and upper branch is hysteretic, while the switch between upper and lower branch is intermittent (can co-exist at the same time).

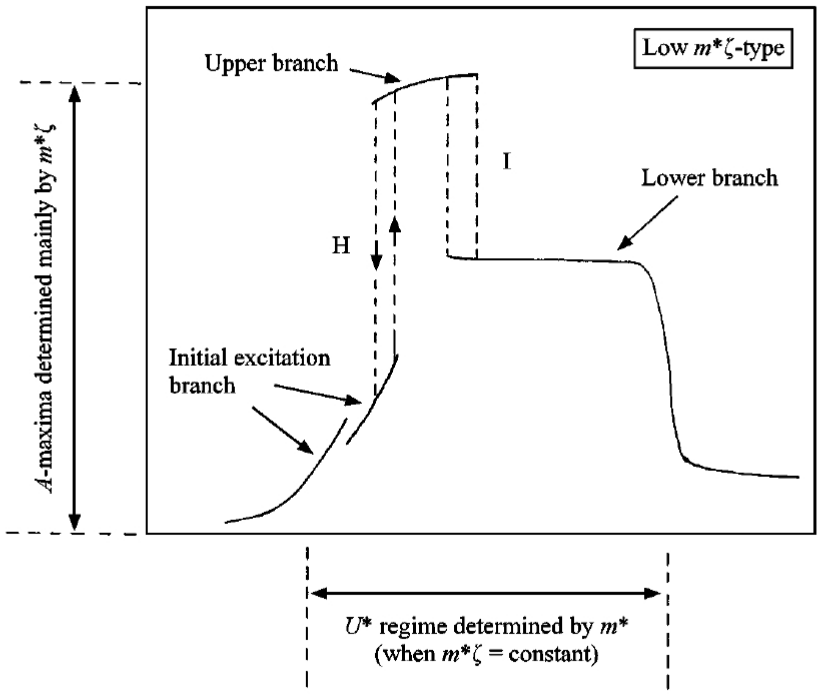



Figure 1.5: Amplitude response involving initial, upper and lower branches in Khalak and Williamson [4]

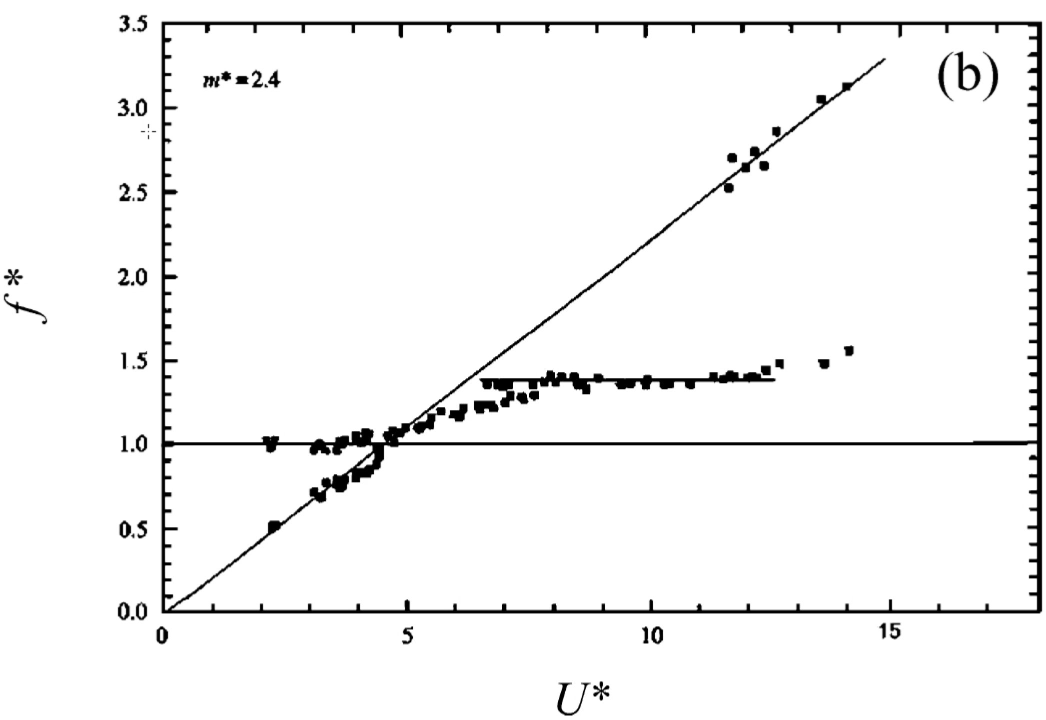



Figure 1.6: Lock-in behavior in Williamson and Govardhan [5]

#### 1.2.1.3 Tandem cylinder system

Carmo et al. [6] conducted numerical research on flow-induced vibrations of a 2-tandem-cylinder system, for which the upstream cylinder is fixed and the downstream one has the freedom to move in the transverse direction at moderate Reynolds number (150 to 300). Reduced velocity is chosen as the control variable and it is varied by changing spring stiffness. Flow induced vibrations of bluff body subject to wake interference (two cylinders) has been widely studied since King and Johns investigated the vibration of two flexible cylinders by experiments in water channel [72]. From studies in tandem cylinder systems, general consensus is that the wake inference results in a larger range of reduced velocity for synchronization, with possibly no upper limit for the vibration to stop. Also oscillation amplitude is larger comparing to the single cylinder case. It is physically understandable because uniform flow in single cylinder case can have a stabilizing effect when reduced velocity is large. However in the wake inference, the larger upstream velocity will result in stronger oscillatory wake for the downstream wake, which will keep the vibration synchronization. In [6], a weakly-coupled method is applied to arbitrary Lagrangian-Eulerian formulation for fluid solver and mass-spring-damper model for structure solver. It is shown in figure 1.7 that both oscillation amplitude and lock-in regime has increased for the two cylinder case. It is proposed that the oscillatory flow between the gap of two cylinders would cause the front stagnation point of the downstream cylinder to shift along the upstream portion of its surface, which results in the difference from single cylinder case.

### 1.2.2 Airfoil

Vortex shedding of airfoil and associated boundary layer behavior is widely studied for both incompressible and compressible regime at various Reynolds number using experimental and computational tools. Moderate Reynolds number flows are of main interest in this section to explore the vortex shedding of oscillating airfoil, different from what's normally related to airfoil study — high Reynolds number and compressible flows. Svavcek et al. [73] performed a numerical simulation for flow induced airfoil vibrations with large amplitude based on

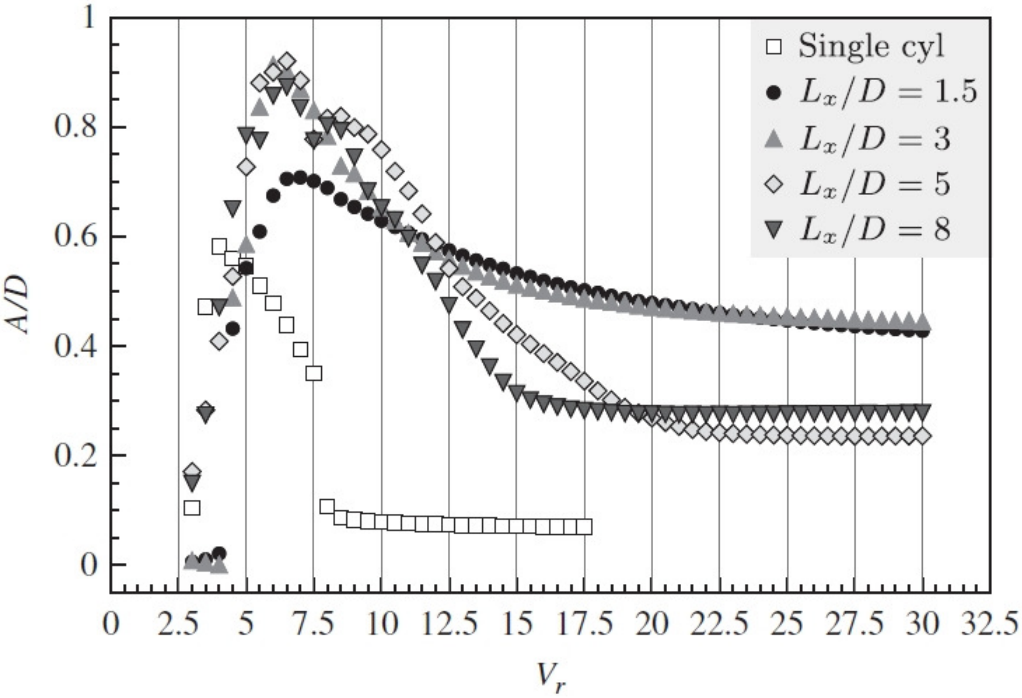



Figure 1.7: Comparison on oscillation amplitude of single cylinder with 2-tandem-cylinder system in Carmo et al. [6]

an incompressible laminar model in relatively low Reynolds number. Young and Lai [74] conducted numerical simulation based on a two-dimensional laminar compressible model to explore oscillation frequency and amplitude effects on the wake of a plunging airfoil. Yarusevych and Boutilier [75] did experiments on shedding frequency of a static airfoil by varying Reynolds number and airfoil thickness in the low Reynolds number regime. We focus on vortex shedding modes from forced oscillation of an airfoil for now.

Schnipper et al. [7] experimentally discussed vortex shedding modes with forced oscillation of an pitching airfoil in a vertically two-dimensional soap film. Except from the basic kármán vortex mode behind a stationary cylinder which produces a pair of vortex of opposite sign at each oscillation period - called 2S mode, this paper found and analyzed other modes of vortex shedding when making a variation of airfoil oscillation amplitude and frequency. One is called inverted kármán vortex street when frequency is relatively low but magnitude is large. With large oscillation magnitude, vortex produced at the leading edge is strong. As it convects along the stream, big strain rates tear every single vortex apart to be the inverted

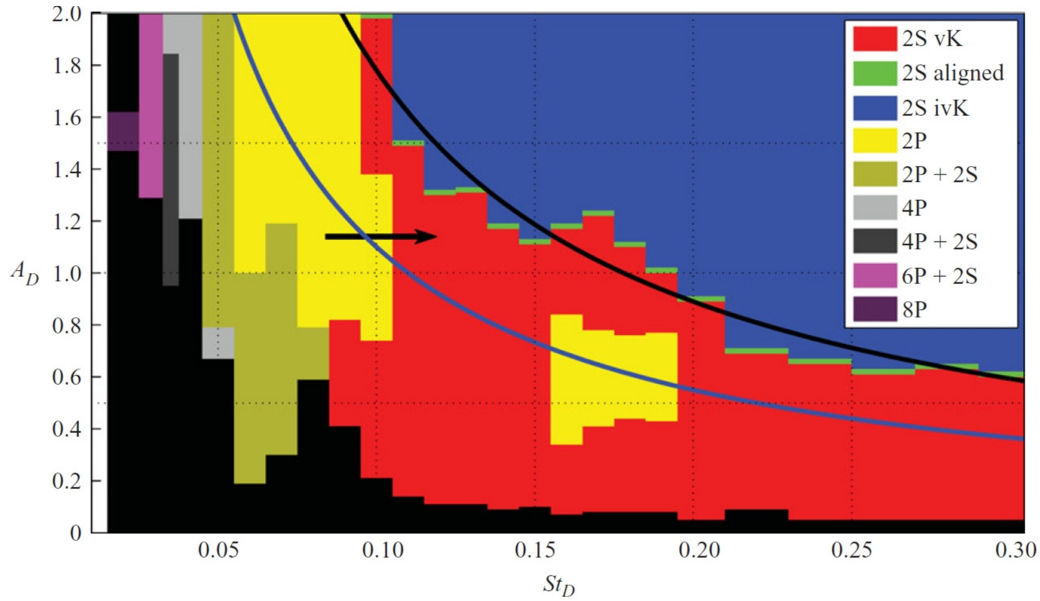



Figure 1.8: Phase diagram of vortex shedding regions in Schnipper et al. [7]

kármán vortex. The other basic mode is called 2P mode, which represents the shedding of two vortex pairs of different sign in one oscillation period. Similarly 4P, 6P, 8P and their combination with 2S are also observed. Stating that Reynolds number is not decisive in the intermediate rage, vortex shedding modes are plotted in a map of the nondimensional Strouhal number $St_D = Df/U$ versus the nondimensional oscillation amplitude $A_D = 2A/D$ shown in figure 1.8.

In n Schnipper et al. [7], different wake modes are observed in different regimes of the $St_D - A_D$ map. In chapter 3, we also compared our numerical results with [7] for vortex modes. One thing to note here is that no hysteresis or lock-in is observed under this experiment setup, possibly due to the use of forced oscillation. The difference between actively coupled fluid–body interaction and passively coupled case should be stressed. Schnipper et al. in [7] also decently explained the formation mechanism for 2P mode in detail with figure 1.9. The two vortex in a pair have different source of generation. One of them is shed from the leading edge and the other is shed at the airfoil tip whenever the airfoil changes the direction of oscillation. When the airfoil oscillation period is an integer times of leading vortex traveling period though airfoil, 2P mode could exist.

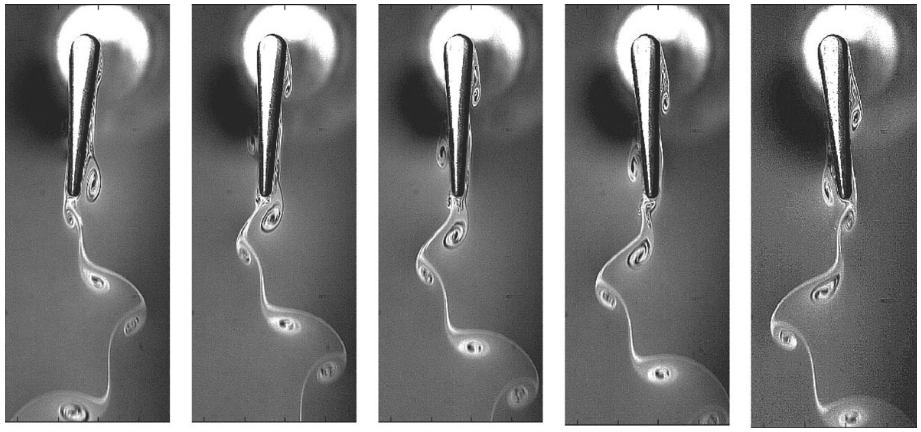


Figure 1.9: Vortex shedding of 2P mode in half oscillation period in Schnipper et al. [7]

### 1.2.3 Flapping flags

Vortex induced vibrations are also widely studies with elastic flags. One example of flapping flag problems with experimental setup is the flexible filaments in a flowing soap film shown in Zhang et al. [8]. This simple setup was then used in numerical simulations in Farnell et al. [76] to compare with experiments. Both of these two papers showed the bi-stability of filament stretched-straight state and filament flapping state. Further, using the filament length as a variable, hysteresis in flapping frequency and amplitude was shown. It is also proved in Zhu and Peskin [77] that if the filament length is short enough, the stretched-straight state is the only stable state. The effect of varying body to fluid mass ratio and flow speed is also discussed in Shelley et al. [78].

Improved from this two-dimensional model, three-dimensional effects are also discussed in Banerjee et al. [79] using an infinite third dimension flag in a viscous flow. Influence of the third dimension is characterized by spanwise wavelength, showing that stability increases with spanwise wavelength increasing, with the two-dimensional case being most unstable. In the process of reaching the limit cycle stability, three different instability modes are identified for governing the growth of instabilities. The effect of flapping flag with a finite third dimension in an inviscid flow is studied in Eloy et al. [80], drawing opposite conclusion on the role of third dimension: flag with a finite span is more stable than an infinite span.

Except from the analysis on single filament immersed in uniform flow, many other sce-

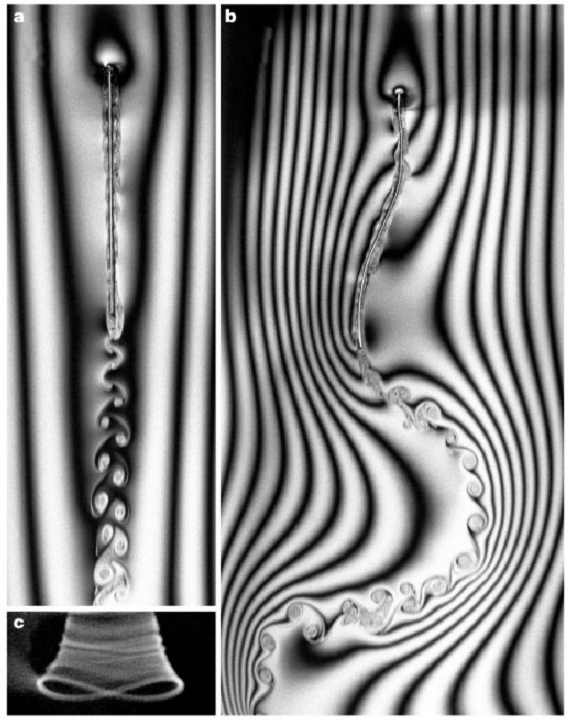



Figure 1.10: Bi-stability of the filament in Zhang et al. [8]

narios are also studied, such as interaction of two parallel flapping filaments in a flowing soap film in Zhu and Peskin [81], Huang et al. [82] and Farnell et al. [83]; two tandem flapping filaments in viscous flow by Uddin et al. [84]; inverted flapping flags in a uniform flow by Ryu et al. [85] and so on.

#### 1.2.3.1 Bi-stability and hysteresis

Eloy et al. [9] addressed the discrepancy in flapping flag instability between the large hysteresis loop found in experiments and small hysteresis loop found in numerical simulations. Different from most of the theoretical linear analysis on aeroelastic instabilities, this paper conducted a weakly nonlinear stability analysis using slender body and two-dimensional body respectively. For experimental research on both large and small aspect ratio flapping flags, hysteresis loop can be as large as 20%. Hysteresis loop is defined as $(U_c - U_d)/U_c$, where $[U_d\ U_c]$ is the velocity range of bi-stability, meaning that motionless state and self-sustained state coexist. Yadykin et al. [86] showed that elastic and inertial nonlinearities without aerodynamic nonlinearities do not lead to hysteresis. Other numerical research including inviscid flow with vortex model and viscous flows modelled with Navier–Stokes equation do identify bi-stability, but with a much smaller range, smaller than 5%. Besides direct numerical simulations, Michelin et al. [87] proved that an unsteady point vortex model for flapping flags is able to reproduce bi-stability and hysteresis characteristics.

In Eloy et al. [9], by simplifying the model equation using linear approximation, one can predict the critical velocity $U_{cr}$, above which the system is unstable. By simplifying using slender body assumption, in no case can subcritical bifurcation be established, thus no hysteresis. By simplifying using two-dimensional weakly nonlinear model, one would get a very small hysteresis loop for an always-subcritical situation, except for small mass ratio cases, which is supercritical. Later comparison with experiments showed good consistency with this two-dimensional weakly nonlinear model.

Further experiments in In Eloy et al. [9] using both flat plates and curved plates were reported. It showed that flutter amplitude and lower bound of hysteresis $U_d$ does not rely on plate curvature, but $U_c$ relies on curvature dramatically, shown in figure 1.11. Hypothesis on stiffness difference is proposed to explain this situation. When both plates are at rest, curved plate is stiffer than flat plate, resulting in a later oscillation. Once they are fluttering, the difference is “ironed out”, resulting in similar $U_d$.

#### 1.2.3.2 Bifurcation regime map

There has been a debate on whether the bifurcation of stability states is subcritical or supercritical in many papers. Basically saying, if we change the control parameters in a continuous way from different directions (either increasing or decreasing), the bifurcation might happen at different positions. If bifurcation does not depend on directions, it is called supercritical. While if bifurcation point changes with different directions, it is called subcritical. Many experiments such as Alben and Shelley [88] stated it to be subcritical, while some other papers such as Eloy et al. [89] suggested it to be supercritical. No common agreed conclusion is made to the date of [89] published.

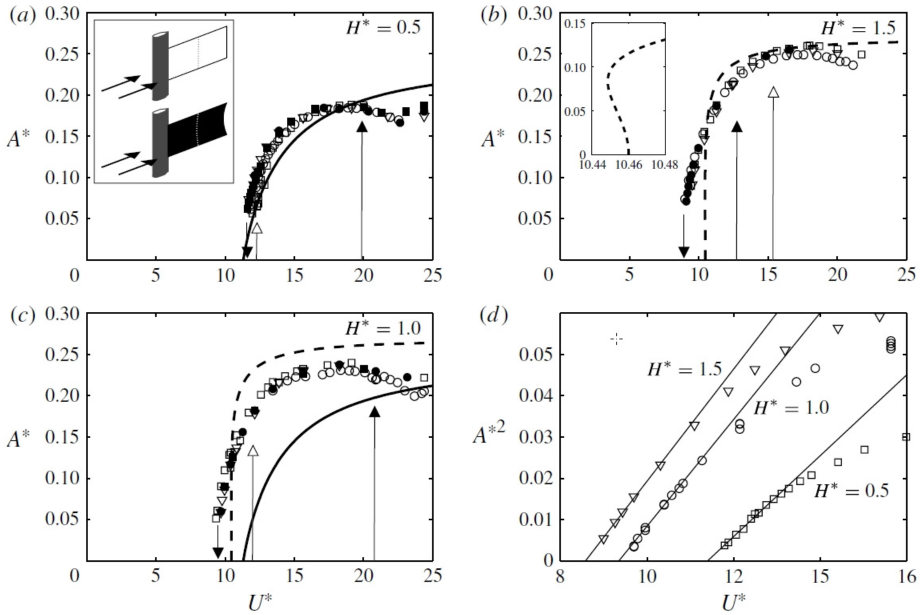



Figure 1.11: Flutter amplitude as a function of reduced velocity for three aspect ratios in In Eloy et al. [9]

Connell and Yue [10] focused on computational study of a two-dimensional flag with high extensional rigidity and low bending rigidity. By defining three dimensionless parameter — mass ratio $\mu$, Reynolds number $Re$ and bending rigidity $K_b$ — as control parameters and varying mass ratio $\mu$, three regimes of flag flapping are explored using a fluid–structure direct coupled simulation. A small $\mu$ gives a fixed-point stability, a medium range of $\mu$ gives a limit cycle stability. Further increasing $\mu$ will break the limit cycle by introducing 3/2 superharmonic mode, which leads to chaos finally. The transition from periodic oscillation state to chaos is also demonstrated in other papers such as Alben and Shelley [88].

No significant bi-stability for $K_b = 0.0001$ is reported in Connell and Yue [10], but with $K_b = 0.001$ bi-stability is observed. Another thing to mention is that for the case of $\mu = 0.145$, initial condition of tail displacement of $A_0 = 0.25$ gave limit cycle stability while $A_0 = 0.001$ gave fixed-point stability. Connell and Yue [10] also mentioned the Duffing equation to help explain hysteresis. Duffing equation is an example of a dynamical system that presents resonance jump behavior in frequency response and chaotic behavior.

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma cos(\omega t) \tag{1.2}$$

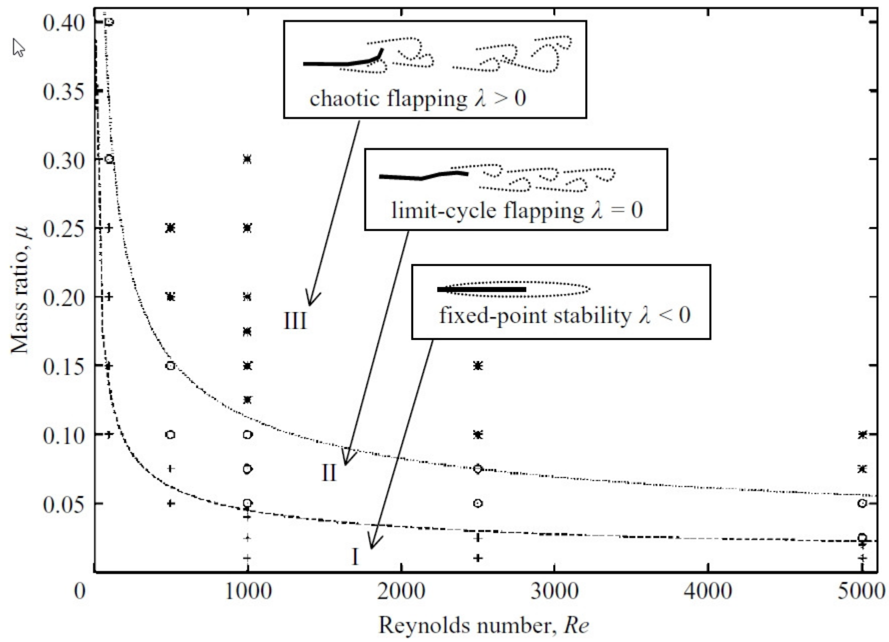



Figure 1.12: Bifurcation regime map in Connell and Yue [10]

where the nonlinear restoring force term $\beta x^3$ makes the frequency response no longer a single value function at some range of the parameters. Also a terminology called snapping is mentioned in [10], referring to the wave traveling through the flag and left the flag, causing a rapid acceleration in flag tail and later a dynamical buckling in the chaos regime. Bifurcation regime map with respect to Reynolds number and mass ratio is plotted in figure 1.12, where $\lambda$ is Lyapunov constant.

### 1.2.4 Energy harvesting

Energy harvesting is associated with fluid–structure interaction problems, for those systems that has the ability to transform fluid energy to other types of energy. By the thought to test the possibility of a membrane (eel) harvesting energy from the unsteady wake generated by flow past a bluff body, Allen and Smits [90] did experimental research using this setup in a relatively wide range of Reynolds number and conducted flow visualization using both fluorescent dye and PIV technique. When Reynolds number is low (with the order of $10^2$ to $10^3$), the membrane oscillation frequency is smaller than the natural frequency of the fluid. As Reynolds number increases to the order of $10^4$, it becomes more obvious that the membrane travels in a waveform-like pattern. The oscillation magnitude of eel increases, effective length of eel is gradually shortened and the frequency of oscillation has a trend of being coupled with the natural frequency of fluid. The final lock-in state coupled membrane oscillation frequency and magnitude with those of undisturbed vortex street behind bluff body without the membrane. Euler-Bernoulli beam equation is used to describe the motion of the membrane. Eigenmodes of the equation can be used to characterize different oscillation modes of the membrane, with every two of them orthogonal to each other.

Singh and Michelin [11] analyzed fluid–structure interaction in energy harvesting point of view using a mass-spring-damper model with two-dimensional cylinder system. In the first part of this paper, large amplitude elongated body theory by Lighthill (1971) is used to model fluid dynamics, thus no viscous effect in fluid is considered. By giving the cylinder system a small perturbation from rest and varying the nondimensional flow speed defined as $u = U_\infty (ML/K)^{1/2}$, a critical velocity $u^{cr}$ is found for Hopf bifurcation to happen. Below $u^{cr}$ a fix-point equilibrium is achieved and above that we get a self-sustained limit cycle, where the energy harvesting analysis would be interested in. In the map of $c_1$ and $c_2$ versus power harvested, there is a optimal combination of $c_1$ and $c_2$ for the most power. The boundary for bifurcation in the energy harvesting plot corresponds well with linear stability analysis of the model equation in figure 1.13. This paper further concluded that $c_1$ harvests more power than $c_2$, because the first dashpot sits closer to the origin of instability.

The second part of this paper used a drag model from the resistive force theory of Taylor (1952) to model the viscous effect of fluid. Drag force in this model is proportional to the square of body velocity, thus still different from the actual viscous term in Navier–Stokes equation. The source of instability is believed to be in the inviscid component in Singh and Michelin [11]. Further, the viscous term has a dual role in energy harvesting: its dissipative nature leads to lower power when flow speed is low, and its stabilizing nature leads to larger range of energy harvesting region when flow speed is high.

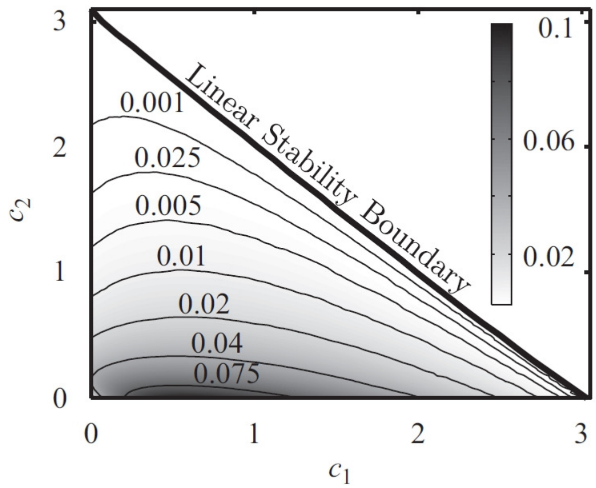



Figure 1.13: Energy harvesting map using inviscid model in Singh and Michelin [11]

## 1.3 Summary: remarks and techniques in current study

Here we discuss remarks and techniques in the current study, with comparisons to several closely related research studies addressed. Consider fluid system alone, a traditional choice for solving the saddle point system resulting from discretized Navier–Stokes equation is the fractional step method. It is known that the fractional step method is only first order accurate in time, due to the ambiguity in time level that field properties are evaluated. Perot [91] first generalized fractional step method in the form of block LU decomposition, transferring the manually specified process to a pure linear algebra problem. Taira and Colonius [27] applied this idea to the immersed boundary method and name it as the immersed boundary projection method (IBPM), where a body with prescribed motion immersed in viscous flow is used as applications. The formulation of fluid governing equations in the current study is slightly different from [27], with Navier–Stokes equation written in the vorticity form. The IBPM formulation works very well since it has solid ground in linear algebra, which is also implemented in Wang and Eldredge [13], Goza and Colonius [67] and Lācis et al. [92].

A strongly-coupled scheme is used in Wang and Eldredge [13] for simulating viscous flow interacting with rigid body systems. In this study, a relaxation scheme similar to the SOR relaxation is used in the iteration procedure for rapid convergence. Coefficients in the relaxation scheme are derived from the explicit consideration of added-mass effect. It is recognized in Causin et al. [58] that in weakly-coupled methods, the missing of added-mass effect would lead to numerical instability. In the vortex particle method based FSI study of Eldredge [12], the added-mass term is explicitly grouped together with body's inertia. It is proved in Eldredge [12] that with this formulation, the computation for zero-mass body is available. All these studies realized the role of added-mass in FSI problems, but incorporates it in an ad-hoc way. Compared to strongly-coupled study of Wang and Eldredge [13], the current study applies a monolithic coupling scheme. The added-mass term appeared in the current study rises from the use of block LU decomposition in an implicit way, and it actually modifies body's inertia term in the body dynamics as one step of block LU decomposition. Reader can refer to section 2.4 for more details.

The current method also adapts an integrating factor technique, which is based on Lattice Green's function to analytically evaluate the viscous diffusion term. This technique is proposed in Liska and Colonius [93, 94], together with the half-explicit Runge–Kutta (HERK) method for time marching. Liska and Colonius [93] used these methods to simulate body with prescribed motion in a viscous flow, while the current study applies the same methods for fluid–structure interaction problems. The techniques contribute to the monolithic feature of the current study, since the integrating factor frees the viscous term from a specified time marching scheme (usually Crank–Nicolson time discretization). Also the current study benefits from the fact that rigid body system is also differential-algebraic, which means in fully dynamically coupled FSI problems, the overall system of equations can be discretized using the same HERK time marching scheme.

The other similarity lies between the current study and Lācis et al. [92], where an implicit coupling method is used for viscous flow coupling with a single rigid body. The implicit coupling comes from the use of block LU decomposition, similar to its use for fluid system alone. In fact Lācis et al. [92] is very close to being monolithic, except that it does not use a uniform time marching scheme to handle both fluid and body systems. One important difference between [92] and the current study is in the treatment of the block matrix. Components in the block matrix between [92] and the current study are similar, however the arrangement is different. The difference in block arrangement gets revealed in the process of block LU decomposition. In [92], the blocks for field properties of fluid dynamics and body dynamics are grouped together for prediction, and the blocks of constraint for both systems are grouped together for correction step. With this way of grouping, the coupling between fluid and zero mass body is unstable, since the computation of inverse of body inertia matrix is needed. On the other hand, the current study is stable for arbitrarily low mass ratios including zero mass body. Moreover since the current study actually applies multiple connected rigid bodies instead of a single rigid body, it is found that the grouping manner in [92] would result in a bad matrix condition number if the number of rigid bodies gets large. A detailed explanation explaining this is given in section 2.3 and 2.4, here a simplified explanation is shown in figure 1.14. Operators in the block matrix are classified into fluid blocks, body blocks and interaction blocks with different colors in this figure. Different shapes are used to distinguish inertia block from constraint blocks. The added-mass term, shown with the effect of both interaction blocks and all fluid blocks, modifies body inertia in solving the body dynamics. Interaction blocks also have an effect on the modification to fluid step.

Finally we have to address the versatility of the current study, which is a result of the analytical and numerical techniques used here. The versatility mainly shows in two aspects. First, we are able to construct a variety of FSI applications with very little change in the code. Only a few lines to set up body configurations and joint types are needed for different problems. The use of spatial vectors together with the matrix form operators makes it easy for the set up. Also we can create different body-joint hierarchy and switch among different types of joint. All the example problems shown in chapter 3 use the same code with differences only in problem set up level. Second, the code developed in this study is relatively compact and open source. It is compact in the sense of computation resources needed. All examples again in chapter 3 are conducted using a normal desktop with Intel Core i7 CPU and 16 GB ram. With the grid size and time step in most examples, computation time is usually limit to only several hours. It is also open source on Github for potential users, although documentation work is still going on at the moment.

For the rest of this thesis, chapter 2 describes the methodology in details, with special attention on how we make our method monolithic, what's the benefit for making the method monolithic, and how to make our method versatile for applications. In chapter 3, multiple applications are shown, mostly to show the versatility of our algorithm meanwhile achieves good agreement with previous studies. Finally in chapter 4, conclusion is made and future work is addressed.

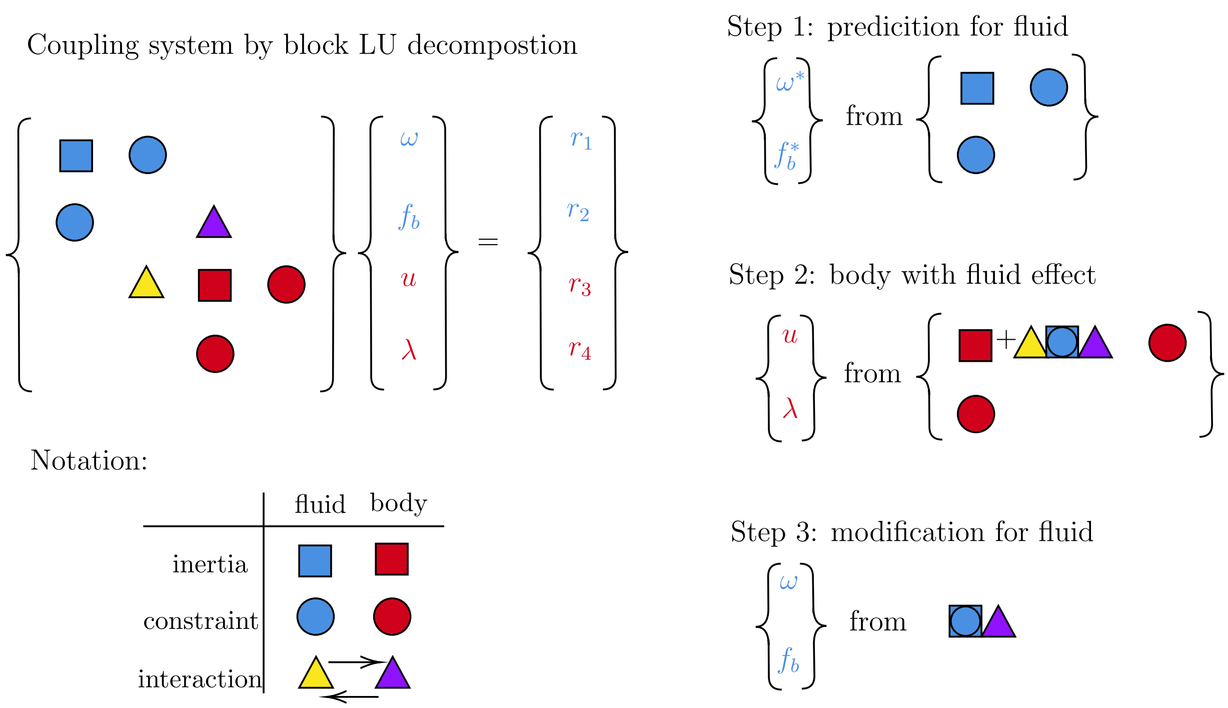



Figure 1.14: Simplified schematic sketch of the coupling scheme through block LU decomposition

# CHAPTER 2

# Methodology

The problem we are generally interested in is the interaction between incompressible viscous flow of low-to-moderate Reynolds number and rigid body systems. Starting from describing fluid system, the vorticity-streamfunction form Navier–Stokes equation is solved with the formulation of immersed boundary projection method (IBPM) by Taira and Colonius [27]. The fluid domain is a uniform Cartesian grid, with immersed boundary represented by Lagrangian points. Non-conforming mesh has great advantages in treating moving boundaries with simplicity. Regularization and interpolation operators are created out of a discrete delta function to transfer force and velocity information between fluid grid and Lagrangian points. A Lattice Green's function based integrating factor technique is applied following Liska and Colonius [93] to analytically evaluate the viscous diffusion term. Rigid body dynamics are presented in the six-dimensional spatial vector form, following Featherstone [95]. The body chain connected by joints are constrained in certain degree of freedoms, depending on the joint type. Motion constraint is enforce on these constrained degree of freedoms, together with prescribed motion.

To achieve coupling between fluid and body system, there are usually choices of monolithic coupling and partitioned coupling. In the current study, we pursue the use of monolithic coupling, because generally monolithic methods are faster and more accurate. The advantage in accuracy benefits from the ability to bypass errors introduced by explicitly tracking fluid–structure interface conditions in partitioned methods. Weakly-coupled methods are not always able to find a stable solution, and strongly-coupled methods depends on iterations for convergence. In contrast, interface properties in monolithic methods are kept implicit, which is a guarantee for stability. In order to obtain a monolithic method, different dynamical systems must be treated in the same mathematical framework to form a single closed form of multi-physics. Besides properties at interaction surface should be kept implicit, all unknowns in the system have to be solved simultaneously as well. In this sense, monolithic methods are more complicated, and their code implementation are usually specified for certain systems.

It's not usually possible to put two different dynamical systems in the same mathematical framework, but the use of Lagrangian multipliers for both system makes this possible in the continuous sense. For the fluid, the Lagrangian force on interface points formed by the immersed boundary method guarantees no-slip, no-flow-through constraint. For the rigid body system, motion constraint from either active prescribed motion or joint linkage is achieved by Lagrangian multipliers as well. Governing equations for both systems are differential-algebriac, shedding light on the creation of a monolithic method. For time marching, both fluid and rigid body system are discretized with the same half-explicit Runge–Kutta (HERK) time marching scheme. HERK method is designed to solve differential-algebriac equations in Brasey and Hairer [96]. Basically the differential equation is solved like a normal Runge–Kutta method (RK), with constraints satisfied at every stage in one time step. Using combinations of RK parameters, arbitrary time accuracy can be achieved. After time discretization, the overall saddle point system describing the fluid–body interaction is solved with block LU decomposition. As the generalization of fractional step method, block LU decomposition works in a rigorous way for the fluid system. With its application in the FSI problem, an added-mass term naturally appears to modify the body's inertia. Since added-mass is always positive, body with arbitrarily low mass ratios including zero are stable for our algorithm. We also consider the fictitious fluid inside a two-dimensional body arise from immersed boundary method into account, assuming the fictitious fluid inside body follows the same rigid body motion.

Referring to the components in fluid–structure interaction, the immersed boundary method based fluid solver is described in section 2.1. Rigid body solver designed to simulated kinematics and dynamics of linked body system is introduced in section 2.2. The monolithic coupling scheme with HERK time marching to combine the two systems is depicted in section 2.3. Finally solution for solving the saddle point system resulted from time discretization is outlined in section 2.4.

## 2.1 Fluid solver

### 2.1.1 Governing equations

We start from the Navier–Stokes equation in its null-space form (vorticity form) with immersed boundary formulation in an unbounded domain to describe the fluid dynamics and its boundary condition

$$\begin{aligned} \frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \times (\boldsymbol{\omega} \times \boldsymbol{u}) &= \frac{1}{Re}\nabla^2 \boldsymbol{\omega} + \int_\Gamma \boldsymbol{f}(\chi(s,t))\ \delta(\chi(s,t) - \boldsymbol{x}) ds \\ \int_\Omega \boldsymbol{u}(\boldsymbol{x})\delta(\boldsymbol{x} - \chi(s,t)) dx &= \boldsymbol{u}_\Gamma(\chi(s,t),t) \end{aligned} \tag{2.1}$$

where $\boldsymbol{\omega}$ is fluid vorticity, $\boldsymbol{u}$ is fluid velocity, $Re$ is the Reynolds number, $\Omega$ is fluid domain, $\Gamma$ is immersed boundary and $\chi$ is the representation of Lagrangian points of immersed body surface. The immersed surface $\chi$ is parametrized by $s$ described in body's local coordinate system and it moves with velocity $\boldsymbol{u}_\Gamma$ as a function of time. Continuity equation is not explicitly needed because the vorticity–streamfunction form of Navier–Stokes equation automatically satisfies the conservation of mass. The immersed boundary method solves for the Lagrangian multiplier $f(\chi(s,t))$ on body Lagrangian points to enforce no-slip and no-flow-through condition. Dirac delta function $\delta$ exchanges information between fluid Cartesian grid and Lagrangian points by convolution with either Lagrangian force or body velocity. The second line in equation (2.1) describes the velocity boundary conditions specified on fluid–body interface. To compute the velocity field $\boldsymbol{u}$, additional Poisson equation is solved using the Lattice Green's function for streamfunction $\psi$, then streamfunction is used to compute velocity field with an curl operation. This step is kept hidden from our governing equation and treated as an internal update step, shown in equation (2.2).

$$
\begin{aligned}
\nabla^2 \psi &= -\boldsymbol{\omega} \\
\boldsymbol{u} &= \nabla \times \psi + \boldsymbol{U_\infty}
\end{aligned}
\tag{2.2}
$$

### 2.1.2 Spatial discretization

Spatial discretization is conducted with finite difference method on a staggered grid based on references (Wang and Eldredge [13], Taira and Colonius [27]). Write continuous Navier–Stokes equation (2.1) in the discrete vorticity form,

$$
\begin{aligned}
&\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} + B_1^T \boldsymbol{f_b} = -N(\boldsymbol{\omega}, t) + \frac{1}{Re} L\boldsymbol{\omega} \\
&B_2 L^{-1} \boldsymbol{\omega} = \boldsymbol{u_b} - \boldsymbol{U_\infty}
\end{aligned}
\tag{2.3}
$$

note that $\boldsymbol{f_b}$ has been moved to the left hand side of the equation, so it is fluid force on body to be solved as Lagrange multipliers. $\boldsymbol{u_b}$ is linear velocity on Lagrangian points as boundary conditions and $\boldsymbol{U_\infty}$ is uniform flow velocity. Note that if we are solving for fluid system alone, boundary conditions $\boldsymbol{u_b}$ should be provided. While for a fluid–structure coupling problem, this equation has to be modified, as shown later. $N$ is curl of non-linear convective term and it is reduced to $\nabla \cdot (\omega \boldsymbol{u})$ for two-dimensional case, where $\omega$ can be treated as scalar. $Re$ is Reynolds number, $L$ is discrete laplacian operator, $\boldsymbol{u_b}$ is body velocity on Lagrangian points as boundary conditions. $B_2 = E_2 C$, where $E_2$ is interpolation operator from fluid grid to Lagrangian points and $C$ is discrete curl operator. $B_1^T = C^T E_1^T$, and $E_1^T$ is regularization operator from Lagrangian points to fluid grid. With stationary body, $E_1^T$ and $E_2$ are transpose of each other, and so is $B_1^T$ and $B_2$. However it is not true for moving body. Due to the time-marching scheme we choose (described in section 2.3), the momentum equation and constraint equation are evaluated at slightly different time, so $B_1^T$ and $B_2$ are not strictly transpose of each other. Interpolation and regularization matrix are constructed based on a smoothed discretization of three-point delta function following Yang et at. [97]. Note that discrete delta function in equation (2.4) is non-dimensionalized by grid size $\Delta x$, so $\boldsymbol{f_b}$ in equation (2.3) is force per unit area for two-dimensional case. The smoothed delta function used here has one-order higher derivative than the regular ones (Roma et al. [98]

for example), and is capable of suppressing the non-physical oscillations in the calculated Lagrangian force. The resulting force distributed over Lagrangian points would be more smooth.

For two-dimensional study, the finite difference is implemented with a standard second order accurate five-point stencil. However due to fact that immersed boundary method is only first-order accurate near the body boundary, this algorithm is less than second-order accurate in space in general.

$$d(r) = \begin{cases} \frac{17}{48} + \frac{\sqrt{3}\pi}{108} + \frac{|r|}{4} - \frac{r^2}{4} + \frac{1-2|r|}{16}\sqrt{-12r^2+12|r|+1} & \\ \quad - \frac{\sqrt{3}}{12}\arcsin\left(\frac{\sqrt{3}}{2}(2|r|-1)\right), & |r| \leq 1, \\ \frac{55}{48} - \frac{\sqrt{3}\pi}{108} - \frac{13|r|}{12} + \frac{r^2}{4} + \frac{2|r|-3}{48}\sqrt{-12r^2+36|r|-23} & \\ \quad + \frac{\sqrt{3}}{36}\arcsin\left(\frac{\sqrt{3}}{2}(2|r|-3)\right), & 1 \leq |r| \leq 2, \\ 0, & 2 \leq |r|. \end{cases} \tag{2.4}$$

### 2.1.3 Lattice Green's function technique

The viscous diffusion term can be represented with finite difference operator in spatial discretization step. However, there exists a more accurate way to analytically deal with elliptical differential equations on an un-bounded domain—the Fast Multipole Method (FMM), or another name the Lattice Green's function technique (LGF), or the integrating factor technique (IF). We summarize the integration factor technique developed by Liska and Colonius [99, 93] to account for the diffusion term analytically in the time-discrete level. Although Liska and Colonius applied this technique to the velocity form of Navier–Stokes equation, the approach is very similar here. The use of LGF technique equips us with the ability to avoid artificial boundary conditions and unnecessary large computation domain. To understand this technique better, we start with the typical heat diffusion equation and its spatial-discretized form.

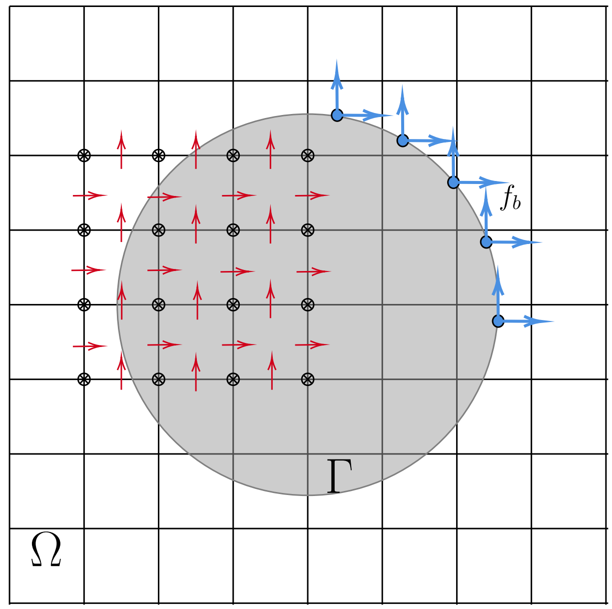



Figure 2.1: Staggered grid with immersed body boundary, vertical and horizontal red arrows represent vertical and horizontal velocities, round cross is where vorticity gets stored. Blue dots are Lagrangian points with Lagrangian multipliers depicted as blue arrows.

$$\begin{aligned} \frac{\partial h}{\partial t} &= k\nabla^2 h \\ \frac{\mathrm{d}h}{\mathrm{d}t} &= \frac{k}{\Delta x^2} L h \end{aligned} \tag{2.5}$$

where $h$ is field variable, $L$ is discrete laplacian operator. We can express $L$ with Fourier series as $L = \mathcal{F}^{-1}\sigma\mathcal{F}$ so that the matrix can be diagonalized. Define field variable $\hat{h}$ in the Fourier space and substitute in we get

$$\frac{\mathrm{d}\hat{h}}{\mathrm{d}t} = \frac{k}{\Delta x^2}\sigma\hat{h} \tag{2.6}$$

and we know the analytic solution to this ODE is

$$\hat{h} = \hat{h_\tau} \exp\left(\frac{k(t-\tau)\sigma}{\Delta x^2}\right) \tag{2.7}$$

where $\hat{h_\tau}$ is the initial condition at time $\tau$. Substitute back to the real space we get

$$h = \hat{h_\tau}\mathcal{F}^{-1} \exp\left(\frac{k(t-\tau)\sigma}{\Delta x^2}\right)\mathcal{F} = \hat{h_\tau} H(\frac{k(t-\tau)\sigma}{\Delta x^2}) \tag{2.8}$$

and this $H$ operator is called the integrating operator. Integrating factor $H(\alpha)$ has the property of

$$\begin{aligned}\frac{dH(\alpha)}{d\alpha} &= \frac{d}{d\alpha}\left[\mathcal{F}^{-1}e^{\alpha\sigma}\mathcal{F}\right] = \mathcal{F}^{-1}\sigma e^{\alpha\sigma}\mathcal{F} = \left[\mathcal{F}^{-1}\sigma\mathcal{F}\right]e^{\alpha\sigma} = LH(\alpha)\\ \frac{dH(-\alpha)}{d\alpha} &= -LH(\alpha)\end{aligned} \tag{2.9}$$

Now similarly, we can apply the integrating factor technique to discrete vorticity form of Navier–Stokes equation. Multiply both sides by $H(-\frac{t-\tau}{Re\Delta x^2})$ and define the diffused variable $v = H(\frac{t-\tau}{Re\Delta x^2})\boldsymbol{\omega} = H(-\alpha)\boldsymbol{\omega}$ to get

$$H(-\alpha)\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} + H(-\alpha)B^T\boldsymbol{f_b} = -H(-\alpha)N(\boldsymbol{\omega},t) + H(-\alpha)\frac{1}{Re}L\boldsymbol{\omega} \tag{2.10}$$

with chain rule we end up with

$$\begin{aligned}&H(-\alpha)\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} = \frac{\mathrm{d}v}{\mathrm{d}t} - \frac{\mathrm{d}H(-\alpha)}{\mathrm{d}t}\boldsymbol{\omega} = \frac{\mathrm{d}v}{\mathrm{d}t} + H(-\alpha)\frac{1}{Re}L\boldsymbol{\omega}\\ &\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} + HB_1^T\boldsymbol{f_b} = -HN(\boldsymbol{\omega},t)\end{aligned} \tag{2.11}$$

Now for simplicity, define diffused vorticity $\boldsymbol{v}$ by

$$\boldsymbol{v}(\tau,t) = H(\tau-t)\boldsymbol{\omega}(t) \tag{2.12}$$

, where $\tau$ is the end time(usually $t_0+\Delta t$) in the current time step and $t$ is the current time of the current stage($t_0 + c_i\Delta t$). By using the $H$ operator, at each stage in HERK vorticity is already diffused. Substitute into momentum equations and get rid of the diffusion term

$$\begin{aligned}&\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} + HB_1^T\boldsymbol{f_b} = -HN(\boldsymbol{\omega},t)\\ &B_2L^{-1}H^{-1}\boldsymbol{v} = \boldsymbol{u_b} - \boldsymbol{U_\infty}\end{aligned} \tag{2.13}$$

Until now, the viscous diffusion term has been folded into the integrating factor before any kind of time discretization.

### 2.1.4 Time marching

Since we are dealing with a differential-algebraic problem of index 2 after spatial discretization (equation (2.13)), the half-explicit Runge–Kutta method (HERK) developed by Brasey and Hairer [96] is naturally a good choice. HERK method, with similarity in the regular Runge–Kutta method, is easy to implement (can be seen more clearly in section 2.2) and can achieve any order of accuracy depending on the number of stages and coefficients chosen. In summary, HERK method solves for differential-algebraic system of the form

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y, z), \qquad 0 = g(y) \tag{2.14}$$

, assuming that $f$ and $g$ are sufficiently differentiable and that

$$g_y(y) f_z(y, z) \quad \text{is nonsingular} \tag{2.15}$$

The method is implemented with $s$ stages and summarized with weights at the end stage

$$Y_i = y_0 + \Delta t \sum_{j=1}^{i-1} a_{ij} f(Y_j, Z_j), \qquad i = 1, \dots, s \tag{2.16a}$$

$$0 = g(Y_i), \qquad i = 1, \dots, s \tag{2.16b}$$

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^{s} b_i f(Y_i, Z_i) \tag{2.16c}$$

$$0 = g(y^{n+1}) \tag{2.16d}$$

where $A = [a_{ij}]$ is the Runge–Kutta matrix, $b = [b_i]$ is the weight factor and $Y_i$ is the value of $y$ at each stage. At every stage inside one step, algebraic constraint is enforced together with solving the differential equation. Here we present the coefficient of a typical three-stage second-order method in Butcher table form. Further detailed discussions about RK coefficients and accuracy can be found in [96] and [93].

Now going back to our fluid equation, there are actually two choices on the variable to solve for. With the summary step in HERK,

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
1 & \frac{\sqrt{3}}{3} & \frac{3-\sqrt{3}}{3} & 0 \\
\hline
 & \frac{3+\sqrt{3}}{6} & -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6}
\end{array}
$$

Table 2.1: Butcher table of a three-stage second-order method

$$
\boldsymbol{v}_i = \boldsymbol{v}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j}\dot{\boldsymbol{v}}_j + \Delta t a_{i,i-1}\dot{\boldsymbol{v}}_{i-1} \tag{2.17}
$$

We can either choose to represent $\dot{\boldsymbol{v}}$ with the rest terms in momentum equation and solve for $\boldsymbol{v}$ (implemented in Liska and Colonius [93, 94]), or we can substitute $\boldsymbol{v}$ in the motion constraint with $\dot{\boldsymbol{v}}$ and solve for $\dot{\boldsymbol{v}}$ (in this work). Here we use the second way, because this is the more standard way of using HERK scheme, and you can directly write the discretized form from differential equations without complicated derivation. Reader can try with both derivation and see the simplicity from this choice. Since HERK scheme is implemented with the help of integrating factor technique, this is also called IF-HERK in Liska and Colonius [93]. Using IF-HERK scheme to discretize momentum equation and enforce boundary conditions (motion constraint) at the same time,

$$
\begin{bmatrix} I & H_{i-1}B_1^T \\ B_2L^{-1}H_i^{-1} & \mathbf{0} \end{bmatrix}
\begin{bmatrix} \dot{\boldsymbol{v}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \end{bmatrix}
=
\begin{bmatrix} -H_{i-1}N(\boldsymbol{\omega}_{i-1}, t) \\ -\dfrac{1}{\Delta t\ a_{i,i-1}}\left[B_2L^{-1}H_i^{-1}\left(\boldsymbol{v}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j}\dot{\boldsymbol{v}}_j\right) - \boldsymbol{u_b} + \boldsymbol{U_\infty}\right] \end{bmatrix} \tag{2.18}
$$

where $I$ is identity matrix and $\mathbf{0}$ is zero matrix. Note that by discretizing the differential equations in time, at each stage the viscous diffusion term gets accounted by applying a different integrating factor, which starts from the current time stage in IF-HERK to the end of this time step. It means that all the $\dot{\boldsymbol{v}}_{i-1}$ are computed on the same time level, so that we can add them up directly in HERK algorithm. Using the extended RK coefficients with

$2 \leq i \leq s+1$,

$$
\begin{aligned}
H_{i-1} &= H(\tau - t_{i-1}) = H(\Delta t - c_{i-1}\Delta t) = H((1-c_{i-1})\Delta t) \\
H_i^{-1} &= H((c_i - 1)\Delta t)
\end{aligned}
\tag{2.19}
$$

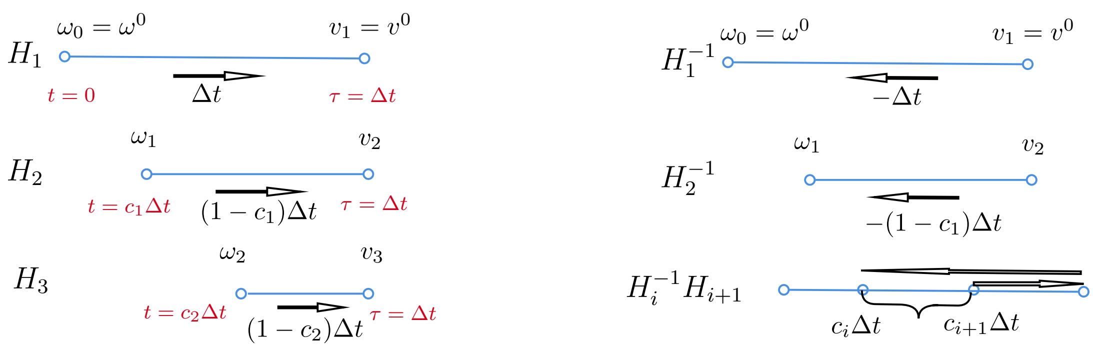



Figure 2.2: Integrating factor computation diagram

This saddle system should work in theory, but numerical issues prevent us from using this form directly. When block LU decomposition is used to solve the system, Schur complement will include $H_i^{-1}H_{i-1}$. In theory it should equal to $H_{i-1,i} = H((c_i - c_{i-1})\Delta t)$, but in numerical practice this operation will introduce errors. So we have to modify the saddle point system a little bit to make the conversion $H_i^{-1}H_{i-1} = H_{i-1,i}$ directly. We can achieve this by defining a new field acceleration term $\dot{\mathfrak{U}}$ as

$$
\dot{\mathfrak{U}}_{i-1} = H_i^{-1}\dot{\boldsymbol{v}}_{i-1}
\tag{2.20}
$$

and multiply the momentum equation of the saddle point system by $H_{i-1}^{-1}$ so the system becomes

$$
\begin{bmatrix} H_{i,i-1} & B_1^T \\ B_2L^{-1} & \mathbf{0} \end{bmatrix}
\begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \end{bmatrix}
=
\begin{bmatrix} -N(\boldsymbol{\omega}_{i-1}, t) \\ -\dfrac{1}{\Delta t\ a_{i,i-1}}\left[B_2L^{-1}\left(H_{1,i}\,\boldsymbol{\omega}_0 + \Delta t\sum_{j=1}^{i-2} a_{i,j}H_{j+1,i}\,\dot{\mathfrak{U}}_j\right) - \boldsymbol{u_b} + \boldsymbol{U_\infty}\right] \end{bmatrix}
\tag{2.21}
$$

Note that this equation is not so different from equation (2.18), but just using a diffused vorticity of different stage. For the simplicity of the equation, we can define a new variable containing all known diffused vorticity from previous stages as

$$\mathfrak{U}_{0\to i-2} = H_{1,i}\, \boldsymbol{\omega}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j} H_{j+1,i}\, \dot{\mathfrak{U}}_j \tag{2.22}$$

, where the subscript is quite self-explanatory, meaning the known quantities from time stage 0 to $i-2$. Then equation (2.21) becomes

$$\begin{bmatrix} H_{i,i-1} & B_1^T \\ B_2 L^{-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \end{bmatrix} = \begin{bmatrix} -N(\boldsymbol{\omega}_{i-1}, t) \\ -\dfrac{1}{\Delta t\ a_{i,i-1}} (B_2 L^{-1} \mathfrak{U}_{0\to i-2} - \boldsymbol{u_b} + \boldsymbol{U_\infty}) \end{bmatrix} \tag{2.23}$$

After solving for the system at each stage, collect the vorticity vector by weighted RK coefficients. We have to modify the summarization step by the integrating factor, since we solve for the diffused vorticity in the block matrix, while normal vorticity is desired.

$$\begin{aligned} \boldsymbol{\omega}_i &= H_i^{-1} \boldsymbol{v}_i = H_i^{-1}(\boldsymbol{v}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} \dot{\boldsymbol{v}}_j) \\ &= H_{1,i}\, \boldsymbol{\omega}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} H_{j+1,i}\, \dot{\mathfrak{U}}_j \end{aligned} \tag{2.24}$$

In code implementation, $H_{1,i}\, \boldsymbol{\omega}_0$ and $H_{j+1,i}\, \dot{\mathfrak{U}}_j$ can be computed using recursion by

$$\begin{aligned} & q_1 = \boldsymbol{\omega}_0 \\ & q_i = H_{i-1,i}\, q_{i-1} \quad \text{for } i \geq 2 \\ & \dot{\mathfrak{U}}_j = H_{i-1,i}\, \dot{\mathfrak{U}}_{j-1} \end{aligned} \tag{2.25}$$

## 2.2 Rigid body solver

### 2.2.1 Spatial vector algebra

Rigid body solver is described with six-dimensional (6d) spatial vector form developed by Featherstone [95]. Compared with the regular three-dimensional (3d) vectors, 6d spatial vector notation combines translational and rotational properties in the save vector, which greatly reduces the volume of algebra, simplifies the tasks of describing and explaining a dynamics algorithm. Here we present some featured aspects of spatial vector in comparison with with 3d vectors. Suppose variables are in a Cartesian coordinate with origin at $o$:

- Spatial velocity:

$$\hat{\boldsymbol{v}}_o = \begin{bmatrix} \boldsymbol{\omega}_x \\ \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z \\ \boldsymbol{v}_{ox} \\ \boldsymbol{v}_{oy} \\ \boldsymbol{v}_{oz} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ \hat{\boldsymbol{v}}_o \end{bmatrix} \tag{2.26}$$

, where $\hat{\boldsymbol{v}}_o$ is linear velocity and $\hat{\boldsymbol{\omega}}$ is angular velocity.

- Spatial force:

$$\hat{\boldsymbol{f}}_o = \begin{bmatrix} \boldsymbol{M_{ox}} \\ \boldsymbol{M_{oy}} \\ \boldsymbol{M_{oz}} \\ \boldsymbol{f_x} \\ \boldsymbol{f_y} \\ \boldsymbol{f_z} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{M}}_o \\ \hat{\boldsymbol{f}} \end{bmatrix} \tag{2.27}$$

, where $\hat{\boldsymbol{M}}_o$ is angular momentum and $\hat{\boldsymbol{f}}$ is linear force.

- Spatial inertia at body's center of mass:

$$\hat{\boldsymbol{I}}_c = \begin{bmatrix} \boldsymbol{I}_c & \boldsymbol{0} \\ \boldsymbol{0} & m\boldsymbol{1} \end{bmatrix} \tag{2.28}$$

, where $\boldsymbol{I}_c$ is rotational inertia about body's center of mass, $m$ is body mass, and $\boldsymbol{1}$ is 3×3 identity matrix. As a result, the dimension of $\hat{\boldsymbol{I}}_c$ is 6×6.

- Coordinate transform:

  Let A and B be Cartesian frames with origins at $O$ and $P$, $\boldsymbol{r} = \overrightarrow{OP}$, $E$ is rotation matrix transform in 3d vectors from A to B coordinates.

  - Transformation matrix from A to B for motion vector ${}^{B}X_{A}$:

$$\begin{aligned} {}^{B}X_{A} &= \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} \\ -\boldsymbol{r}\times & \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ -\boldsymbol{E}\boldsymbol{r}\times & \boldsymbol{E} \end{bmatrix} \\ {}^{A}X_{B} &= \left({}^{B}X_{A}\right)^{-1} \end{aligned} \tag{2.29}$$

- Transformation matrix from A to B for force vector ${}^{B}X_{A}{}^{*}$:

$$\begin{aligned} {}^{B}X_{A}{}^{*} &= \left({}^{B}X_{A}\right)^{-T} \\ {}^{A}X_{B}{}^{*} &= \left({}^{B}X_{A}{}^{*}\right)^{-1} \end{aligned} \tag{2.30}$$

- Transformation matrix from A to B for inertia matrix $\boldsymbol{I}_A$:

$$\boldsymbol{I}_B = ({}^{B}X_{A}{}^{*})\boldsymbol{I}_A({}^{A}X_{B}) \tag{2.31}$$

### 2.2.2 Governing equations

#### 2.2.2.1 Momentum equation

The dynamics we focus on is forward dynamics: given forces applied to the system, compute reaction acceleration in response. We assume the body system is generally single body or multiple bodies linked by different kinds of joints (revolute, prismatic, cylindrical etc.). Generally we assume the joints to be massless for all the derivations below. Our goal in general, is to express the body dynamics of a multiple body system with matrix operators that help reveal each body's hierarchy in the body-joint chain, also allow for the description of different joint types. We start from most general momentum equation for a single body, and then collect the system into matrices. We denote body index by $i$ and joint index by $j$ to avoid confusion. Be careful not to be confused with notation in the last section, where $i$ is used to represent time level. Start from the most general momentum equation for one body with index i

$$\boldsymbol{f}_i = \frac{\mathrm{d}(\boldsymbol{I}_i \boldsymbol{u}_i)}{\mathrm{d}t} \tag{2.32}$$

, which just indicates that the sum of forces exerted on the body equals to the time derivative of body's momentum. We can express forces in details by forces on joints $\boldsymbol{f}_{J,i}$, gravitational forces $\boldsymbol{f}_{g,i}$ and forces from the fluid $\boldsymbol{f}_{fluid,i}$. The right hand is further expanded by product rule and expressed using identities in spatial vector algebra

$$\begin{aligned} \boldsymbol{f}_{J,i} + \boldsymbol{f}_{external,i} &= \boldsymbol{I}_i \frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t} + \boldsymbol{u}_i \frac{\mathrm{d}\boldsymbol{I}_i}{\mathrm{d}t} \\ \boldsymbol{f}_{J,i} + \boldsymbol{f}_{g,i} + \boldsymbol{f}_{fluid,i} &= \boldsymbol{I}_i \frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t} + \boldsymbol{u}_i \times^{*} \boldsymbol{I}_i \boldsymbol{u}_i \end{aligned} \tag{2.33}$$

Note that here $\boldsymbol{f}_{J,i}$ is the total joint force on body $i$ applied by all the joints connected to it. Rearrange to get

$$\boldsymbol{f}_{J,i} = \boldsymbol{I}_i \frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t} + \underbrace{(\boldsymbol{u}_i \times^* \boldsymbol{I}_i \boldsymbol{u}_i - \boldsymbol{f}_{g,i} - \boldsymbol{f}_{fluid,i})}_{\boldsymbol{p}_i} \tag{2.34}$$

Here $\boldsymbol{p}_i$ is called the bias force, and it is simply the value that $\boldsymbol{f}_{J,i}$ must take in order to produce zero acceleration. We can further describe joint forces $\boldsymbol{f}_{J,i}$ by the joint force from the current body's parent, and joint force from the current body's child. Multiple parents and children are allowed.

$$\boldsymbol{f}_{J,i} = \underbrace{\sum_{j \in ch(i)} \boldsymbol{f}_{J,j}}_{\text{joint forces from } \textbf{child} \text{ joint of body i}} - \underbrace{\sum_{j \in pa(i)} \boldsymbol{f}_{J,j}}_{\text{joint forces from } \textbf{parent} \text{ joint of body i}} \tag{2.35}$$

Child joint of a body is the joint that connects the body to its child body in hierarchy, similarly for parent joint. Note that we express the body force in joint space by making joint force from child joint positive. Usually rigid body dynamics are not expressed and calculated in matrix form. Using a for loop for computing each body in the chain is sufficient for most cases. However we have the need to couple the rigid body system with fluid system and form a saddle point matrix, thus we have to collect all bodies in the chain to form a matrix. For the matrix form we can write

$$\mathbb{P}\boldsymbol{f}_J = \mathbb{M}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{p} \tag{2.36}$$

$$\mathbb{P}_{i,j} = \begin{cases} [\mathbf{1}]_{6\times 6}, \text{ if body } i \text{ is child of joint } j \\ [-\mathbf{1}]_{6\times 6}, \text{ if body } i \text{ is parent of joint } j \\ [\mathbf{0}], \text{ otherwise} \end{cases} \tag{2.37}$$

$\mathbb{M}$ is just putting inertia $\boldsymbol{I}_i$ of each body on the diagonal blocks and it remains constant all the time if each of them is described in body's local coordinate system. Now we express joint forces $\boldsymbol{f}_{J,j}$ using joint models by dividing the six degree of freedoms into joint's motion space (unconstrained space) $S$ and the constrained space $T$. Forces that show in the motion space are represented by $\boldsymbol{\tau_j}$ and forces in the constrained space are denoted by $\boldsymbol{\lambda}_j$. Basically one could understand $\boldsymbol{\lambda}_j$ as the Lagrange multiplier to enforce motion constraint. Describe in local body coordinate frame and express joint force in the motion space by spring-damper model,

$$\begin{aligned} \boldsymbol{f}_{J,j} &= S\boldsymbol{\tau_j} + T\boldsymbol{\lambda}_j \\ \boldsymbol{\tau}_j &= -k\boldsymbol{q}_J - c\boldsymbol{v}_J \end{aligned} \tag{2.38}$$

Express $\boldsymbol{f}_{J,j}$ in the inertial frame we have

$$\boldsymbol{f}_{J,j} =^{inertial} X_{joint}(S\boldsymbol{\tau_j} + T\boldsymbol{\lambda}_j) \tag{2.39}$$

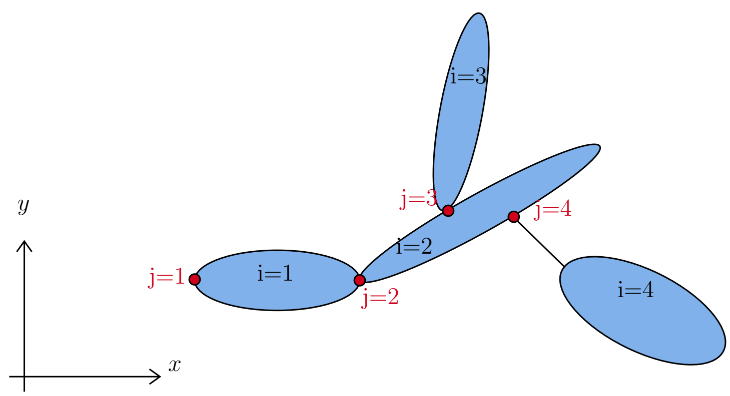



Figure 2.3: An example of linked joint-body system with planar, revolute and prismatic joints

Figure 2.3 gives an example for a linked joint-body configuration, which contains a planar joint $j = 1$ (free to translate in x and y directions, also able to rotate), two revolute joint $j = 2, 3$ (only free to rotate) and a prismatic joint $j = 4$ (only free to translate along its axis). Based on the hierarchy information, $\mathbb{P}$ matrix for figure 2.3 is shown in equation (2.40). For example the second line of matrix $\mathbb{P}$ tells us that body $i = 2$ has a parent $i = 1$, and two children $i = 3, 4$. Also motion space and constraint space for the example is shown in table 2.2.

$$\mathbb{P} = \begin{bmatrix} [\mathbf{1}] & [-\mathbf{1}] & [\mathbf{0}] & [\mathbf{0}] \\ [\mathbf{0}] & [\mathbf{1}] & [-\mathbf{1}] & [-\mathbf{1}] \\ [\mathbf{0}] & [\mathbf{0}] & [\mathbf{1}] & [\mathbf{0}] \\ [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{1}] \end{bmatrix} \tag{2.40}$$

| | revolute | prismatic | planar |
|---|---|---|---|
| $[S\|T]$ | $\left[\begin{array}{c\|ccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$ | $\left[\begin{array}{c\|ccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$ | $\left[\begin{array}{ccc\|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$ |

Table 2.2: Motion space $S$ and constraint space $T$ for revolute, prismatic and planar joint on two-dimensional plane

As a remainder, until now we haven't made rigorous statement on which coordinates is the momentum equation in. If we're solving the momentum equation for all bodies each in their own local body coordinate, then there's a benefit that the inertia matrix would stay constant for all time. Note that body local coordinate refers to a coordinate system that is attached with the body, having the same translational and rotational motion with the body. We just need to express velocity $\boldsymbol{u}$ of each body in local body coordinate as well. Collect the system for $\boldsymbol{f}_J$ we get

$$\boldsymbol{f}_J = \mathbb{A}(\mathbb{S}\boldsymbol{\tau} + \mathbb{T}\boldsymbol{\lambda}) \tag{2.41}$$

Here the $\mathbb{A}$ matrix is similar to $\mathbb{P}$ in structure, but it contains both parent-child hierarchy information and coordinate transform for forces (${}^iX_j^* = {}^jX_i{}^T$) from child body to parent body,

$$\mathbb{A}_{i,j} = \begin{cases} [\mathbf{1}], \text{ if body } i \text{ is child of joint } j \\ [\mathbf{-1}][{}^iX_j^*], \text{ if body } i \text{ is parent of joint } j \\ [\mathbf{0}], \text{ otherwise} \end{cases} \tag{2.42}$$

So again, matrix $\mathbb{A}$ applied to the example system in figure 2.3 is

$$\mathbb{A} = \begin{bmatrix} [\mathbf{1}] & [-\mathbf{1}][{}^1X_2^*] & [\mathbf{0}] & [\mathbf{0}] \\ [\mathbf{0}] & [\mathbf{1}] & [-\mathbf{1}][{}^2X_3^*] & [-\mathbf{1}][{}^2X_4^*] \\ [\mathbf{0}] & [\mathbf{0}] & [\mathbf{1}] & [\mathbf{0}] \\ [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{1}] \end{bmatrix} \tag{2.43}$$

Assembled motion space $\mathbb{S}$ and constraint space $\mathbb{T}$ are

$$\mathbb{S} = \begin{bmatrix} S_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & S_2 & & & & & \vdots \\ \vdots & & \ddots & & & & \vdots \\ \vdots & & & S_j & & & 0 \\ \vdots & & & & \ddots & & 0 \\ \vdots & & & & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & S_n \end{bmatrix}, \quad \mathbb{T} = \begin{bmatrix} T_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & T_2 & & & & & \vdots \\ \vdots & & \ddots & & & & \vdots \\ \vdots & & & T_j & & & 0 \\ \vdots & & & & \ddots & & 0 \\ \vdots & & & & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & T_n \end{bmatrix} \tag{2.44}$$

respectively, where each of $S_i$ or $T_i$ can be different. In other words, we don't necessarily need to have the same type of joints for each joint in this joint-body system. So rearrange for the matrix form momentum equations we get

$$\mathbb{M}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = [-\boldsymbol{p} + \mathbb{A}\mathbb{S}\boldsymbol{\tau}] - \mathbb{A}\mathbb{T}\boldsymbol{\lambda} \tag{2.45}$$

In this equation, body system velocity $\boldsymbol{u}$ and joint constraint force $\boldsymbol{\lambda}$ are unknowns. $\boldsymbol{\lambda}$ serves as the Lagrange multiplier to enforce desired motion constraint, which can come from either joint model, or possible active prescribed motion of the body. We need an additional motion constraint equation to form a closure.

#### 2.2.2.2 Motion constraint

In the momentum equation, we choose to solve for body velocity $\boldsymbol{u}$. However for the motion constraint, we want to express it in the joint space by $\boldsymbol{u}_J$. So we need a relation between body velocity and joint velocity,

$$\boldsymbol{u}_i = ({}^iX_{pa})\boldsymbol{u}_{pa} + \boldsymbol{u}_J \tag{2.46}$$

,where $i$ stands for current body and $pa$ stands for parent of body $i$. We can see that the velocity relation is expressed in local body coordinate of body $i$.

For a joint, among its six degrees of freedoms (dof), those constrained in $T$ are not allow to move. This is independent of active motion, just simply determined by joint type. For example revolute joints are inherently constrained in dof=$1, 2, 4, 5, 6$ except for the rotational dof around z-axis. However if we assign prescribed motion to unconstrained dof, these unconstrained dof need to be changed from unconstrained to constrained, and an extra constraint force has to be provided as Lagrange multipliers in order to sustain this active motion. So for problems involving active motion, we need to modify $S$ and $T$ from the intrinsic joint model. For example in figure 2.3, if we prescribe rotation for the revolute joint, $S$ will be empty and $T$ will be $[\mathbf{1}]_{6\times 6}$. Using the modified motion and constraint force space, we can describe motion constraint in local body coordinate in the null space of the motion space using a single equation,

$$T^T[\boldsymbol{u}_i - ({}^iX_{pa})\boldsymbol{u}_{pa}] - T^T\boldsymbol{c}_i = \mathbf{0} \tag{2.47}$$

, where $\boldsymbol{c}$ contains both prescribed (active) joint motion and constrained joint motion (no motion, thus all zeros). Collect all joints in the system and write in matrix form,

$$\mathbb{T}^T\mathbb{B}\boldsymbol{u} + \mathbb{T}^T\boldsymbol{c} = 0 \tag{2.48}$$

and we know matrix $\mathbb{B}$ is related to $\mathbb{A}$ by its inverse transpose $\mathbb{B} = (\mathbb{A})^{-T}$. $\mathbb{B}$ is written from joint view but $\mathbb{A}$ is written from body view.

$$\mathbb{B}_{j,i} = \begin{cases} [\mathbf{1}], \text{ if joint } j \text{ is parent of body } i \\ [-\mathbf{1}][{}^jX_i], \text{ if joint } j \text{ is child of body } i \\ [\mathbf{0}], \text{ otherwise} \end{cases} \tag{2.49}$$

#### 2.2.2.3 Summary

Notice that all the transform matrices are calculated by body position vector $\boldsymbol{q}$ at every time step, we need to update body position as well. So combine all the equations to full closure we have

$$
\begin{cases}
\dfrac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \boldsymbol{u} \\
\mathbb{M}\dfrac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \underbrace{[-\boldsymbol{p} + \mathbb{A}\mathbb{S}\boldsymbol{\tau}]}_{F(\boldsymbol{q},\boldsymbol{u},t)} \quad - \quad \underbrace{\mathbb{A}\mathbb{T}}_{G_1^T(\boldsymbol{q},t)} \boldsymbol{\lambda} \\
0 = \underbrace{\mathbb{T}^T\mathbb{B}}_{G_2(\boldsymbol{q},t)} \boldsymbol{u} \quad + \quad \underbrace{\mathbb{T}^T\boldsymbol{c}}_{\boldsymbol{u}_c(\boldsymbol{q},t)}
\end{cases}
\tag{2.50}
$$

Here we give assign some new symbols for the simplicity of the equation. $F$ contains the bias force and joint force in the unconstrained space. Forces from the fluid is also contained in $F$ operator. $G_1^T$ is an constraint operator for the Lagrangian force, revealing body-joint hierarchy and frame transformation information. Similarly, $G_2$ is an constraint operator for velocity in the degrees of freedom either constrained by joint type, or prescribed joint motion. The Lagrangian multiplier $\boldsymbol{\lambda}$ is used to enforce motion constraint. At last, $\boldsymbol{u}_c$ denotes the prescribed motion (if any), otherwise zeros. Also note that even the body-joint system doesn't move, $\mathbb{A}\mathbb{T}$ and $\mathbb{T}^T\mathbb{B}$ are not actually transpose of each other.

$$
\mathbb{T}^T\mathbb{B} = \mathbb{T}^T(\mathbb{A})^{-T} = (\mathbb{A}^{-1}\mathbb{T})^T \tag{2.51}
$$

### 2.2.3 Time marching

The set of governing equations we have is

$$
\begin{cases}
\dfrac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \boldsymbol{u} \\
\mathbb{M}\dfrac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = F(\boldsymbol{q},\boldsymbol{u},t) - G_1^T(\boldsymbol{q},t)\boldsymbol{\lambda} \\
0 = G_2(\boldsymbol{q},t)\boldsymbol{u} + \boldsymbol{u}_c(\boldsymbol{q},t)
\end{cases}
\tag{2.52}
$$

Observing that this is also a system of index-2 differential algebraic equations, the same half-explicit Runge–Kutta method (HERK) is implemented. At every stage among a total of s stages in a single time step, momentum equations are advanced with the satisfaction of constraints.

**Stage 1**:

$$
\boldsymbol{q}_1 = \boldsymbol{q}_0, \qquad \boldsymbol{u}_1 = \boldsymbol{u}_0 \tag{2.53}
$$

**Stage $\boldsymbol{i}(2 \leq i \leq s+1)$:**

- Calculate $F(\boldsymbol{q}_{i-1}, \boldsymbol{u}_{i-1}, t_{i-1})$ and $G_1^T(\boldsymbol{q}_{i-1}, t_{i-1})$ based on known $\boldsymbol{q}_{i-1}$ and $\boldsymbol{u}_{i-1}$ from last stage.

- Update $\boldsymbol{q}_i$ for current stage

$$\boldsymbol{q}_i = \boldsymbol{q}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} \boldsymbol{u}_j \tag{2.54}$$

- Update $G_2(\boldsymbol{q}_i, t_i)$ and $\boldsymbol{u}_c(\boldsymbol{q}_i, t_i)$ using $\boldsymbol{q}_i$ for current stage

- Define the summarization of velocity in the previous stages as

$$\boldsymbol{u}_{0\to i-2} = \boldsymbol{u}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j} \dot{\boldsymbol{u}}_j \tag{2.55}$$

  Then solve linear system for acceleration and Lagrange multipliers at stage $i$. Note that the dimension of this matrix is relatively small (less than 12 times of the number of bodies), thus can be solved with direct methods.

$$\begin{bmatrix} \mathbb{M} & G_1^T(\boldsymbol{q}_{i-1}, t_{i-1}) \\ G_2(\boldsymbol{q}_i, t_i) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} = \begin{bmatrix} F(\boldsymbol{q}_{i-1}, \boldsymbol{u}_{i-1}, t_{i-1}) \\ -\dfrac{1}{\Delta t \; a_{i,i-1}} (G_2(\boldsymbol{q}_i, t_i) \boldsymbol{u}_{0\to i-2} + \boldsymbol{u}_c(\boldsymbol{q}_i, t_i)) \end{bmatrix} \tag{2.56}$$

- Update $\boldsymbol{u}_i$ using $\dot{\boldsymbol{u}}_{i-1}$

$$\boldsymbol{u}_i = \boldsymbol{u}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} \dot{\boldsymbol{u}}_j \tag{2.57}$$

Results got from stage (s+1) is the output. RK parameters at stage $s+1$ is

$$a_{s+1,j} = b_j, \quad c_{s+1} = 1 \tag{2.58}$$

## 2.3 FSI coupling

In the fluid governing equations, $\boldsymbol{u}_b$ is supposed to be given as known values for stationary body or body with prescribed motion. Also the external fluid force in body governing equations is supposed to be given (contained in $F$ operator). In order to construct a monolithic coupling scheme, these two terms must be moved from right hand side to the left hand side as unknowns, keep them implicit and solve together with fluid field and body dynamics. There are also two more operators describing the fluid–structure interactions to be constructed here. Note that $\boldsymbol{f_b}$ computed from the fluid solver are distributed on the Lagrangian points on the fluid–body interface. In order for body solver to use it as an external force, $\boldsymbol{f_b}$ should be integrated along the immersed surface, then transform from inertia coordinate to body local coordinate. The force integral is performed using trapezoidal rule in this work. We call this operation $T_1^T$ and separates $\boldsymbol{f_b}$ from $F$ for convenience. Be aware that since body is immersed in fluid now, $F$ should also include buoyancy force for cases with two-dimensional body under gravity. Rigid body equations then become

$$\begin{bmatrix} \mathbb{M} & G_1^T(\boldsymbol{q}_{i-1}, t_{i-1}) \\ G_2(\boldsymbol{q}_i, t_i) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} = \begin{bmatrix} F(\boldsymbol{q}_{i-1}, \boldsymbol{u}_{i-1}, t_{i-1}) + T_1^T \boldsymbol{f_b} \\ -\dfrac{1}{\Delta t\ a_{i,i-1}} (G_2(\boldsymbol{q}_i, t_i)\boldsymbol{u}_{0\to i-2} + \boldsymbol{u}_c(\boldsymbol{q}_i, t_i)) \end{bmatrix} \tag{2.59}$$

Similarly, $\boldsymbol{u_b}$ should be provided on every Lagrangian point described in the inertial frame, serving as fluid boundary condition. However the body kinematics are described in the body local frame. Thus there has to be an operation that interpolates rigid body velocity over the Lagrangian points and then transform from body local coordinate to inertia frame. Velocity interpolation is performed using linear interpolation. We call this operation $T_2$. Note that $T_2$ and $T_1^T$ are only related to body points position and they should be updated in every time step with the body position in the time level that's needed. Also note that $\boldsymbol{u_b}$ serving as boundary conditions in fluid part needs to be accumulated through summation until stage $i$

$$\begin{aligned} (\boldsymbol{u_b})_i &= \underbrace{\boldsymbol{u}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j}\dot{\boldsymbol{u}}_j}_{\text{known}} + \Delta t a_{i,i-1} \underbrace{\dot{\boldsymbol{u}}_{i-1}}_{\text{to be solved}} \\ &= \boldsymbol{u}_{0\to i-2} + \Delta t a_{i,i-1}\dot{\boldsymbol{u}}_{i-1} \end{aligned} \tag{2.60}$$

The other change is that the off-diagonal blocks in fluid solver should be updated with body position accordingly. To simplify the symbols used, we rename $B_2L^{-1}$ in the fluid solver to be the new $B_2$. With these updates for monolithic coupling, fluid saddle system is modified to

$$\begin{bmatrix} H_{i,i-1} & B_1^T(\boldsymbol{q}_{i-1}) \\ B_2(\boldsymbol{q}_i) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \end{bmatrix} = \begin{bmatrix} -N(\boldsymbol{\omega}_{i-1}) \\ -\dfrac{1}{\Delta t\ a_{i,i-1}} \left[ B_2(\boldsymbol{q}_i)\mathfrak{U}_{0\to i-2} - T_2\left(\boldsymbol{u}_{0\to i-2} + \Delta t a_{i,i-1} \dot{\boldsymbol{u}}_{i-1}\right) + \boldsymbol{U}_{\boldsymbol{\infty}} \right] \end{bmatrix} \tag{2.61}$$

Now it's time to combine the two sets of equations together. Since block LU decomposition will be used to decompose the coupled system, the way of combining unknowns together does matter. Here two ways to arrange the unknowns and operation blocks are compared

$$\begin{bmatrix} H_{i,i-1} & \mathbf{0} & B_1^T & \mathbf{0} \\ \mathbf{0} & \mathbb{M} & T_1^T & G_1^T \\ B_2 & -T_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & G_2 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} H_{i,i-1} & B_1^T & \mathbf{0} & \mathbf{0} \\ B_2 & \mathbf{0} & -T_2 & \mathbf{0} \\ \mathbf{0} & -T_1^T & \mathbb{M} & G_1^T \\ \mathbf{0} & \mathbf{0} & G_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \\ \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} \tag{2.62}$$

For convenience we call the left formulation choice 1, and the right formulation choice 2. Although it's just a matter of how to place the block operators, the two ways actually result in different behaviour in block LU decomposition. When we first look at this, the first way seems better not only it combines accelerations and Lagrange multipliers from both systems together, but also the right bottom corner of the matrix shows as all zero matrices. Lācis et al. [92] actually formed their FSI system in this way. In their paper, instability for zero mass body case is reported since the prediction step in block LU decomposition requires computing the inverse of body inertia matrix, which is not possible for zero mass body. Further reasoning for not choosing the first way is found for rigid body systems with multiple bodies. Since the scale of inertia matrix for fluid and body systems are very likely to be different, the overall Schur complement might result in a bad matrix condition number. This effect is amplified as the number of bodies gets larger. Lācis et al. [92] didn't report this matrix condition issue since they only used a single rigid body as examples. With these reasons, we continue with the second choice (Reader can check for derivation with the first choice in Appendix A.1 for the differences in details). Actually the second choice successfully reveals the effect of added-mass in the block LU decomposition, as shown in the next section. With the second choice of combining both systems, our coupled fluid–structure block system becomes

$$
\begin{bmatrix}
H_{i,i-1} & B_1^T(\boldsymbol{q}_{i-1}) & \mathbf{0} & \mathbf{0} \\
B_2(\boldsymbol{q}_i) & \mathbf{0} & -T_2 & \mathbf{0} \\
\mathbf{0} & -T_1^T & \mathbb{M} & G_1^T(\boldsymbol{q}_{i-1}) \\
\mathbf{0} & \mathbf{0} & G_2(\boldsymbol{q}_i) & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\dot{\mathfrak{U}}_{i-1} \\
\boldsymbol{f_b}_{i-1} \\
\dot{\boldsymbol{u}}_{i-1} \\
\boldsymbol{\lambda}_{i-1}
\end{bmatrix}
=
\begin{bmatrix}
-N(\boldsymbol{\omega}_{i-1}) \\
-\frac{1}{\Delta t \; a_{i,i-1}} \left( B_2(\boldsymbol{q}_i) \mathfrak{U}_{0\to i-2} - T_2 \boldsymbol{u}_{0\to i-2} + \boldsymbol{U_\infty} \right) \\
F(\boldsymbol{q}_{i-1}, \boldsymbol{u}_{i-1}, t_{i-1}) \\
-\frac{1}{\Delta t \; a_{i,i-1}} \left( G_2(\boldsymbol{q}_i) \boldsymbol{u}_{0\to i-2} + \boldsymbol{u}_c(\boldsymbol{q}_i, t_i) \right)
\end{bmatrix}
=
\begin{bmatrix}
r_{11} \\
r_{21} \\
r_{12} \\
r_{22}
\end{bmatrix}
\tag{2.63}
$$

After solving for the saddle system at every stage, the assembly stage is done by

$$
\begin{cases}
\boldsymbol{\omega}_i = H_{1,i}\, \boldsymbol{\omega}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} H_{j+1,i}\, \dot{\mathfrak{U}}_j \\
\boldsymbol{u}_i = \boldsymbol{u}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} \dot{\boldsymbol{u}}_j
\end{cases}
\tag{2.64}
$$

## 2.4 Saddle point system

After obtaining the saddle point system, we solve it by block LU decomposition through a three-step manner: projection, applying constraints, and correction. Here we revisit the general steps for block LU decomposition on a block system:

$$
\begin{bmatrix}
A & C_1 \\
C_2 & D
\end{bmatrix}
\begin{bmatrix}
\boldsymbol{u} \\
\boldsymbol{\lambda}
\end{bmatrix}
=
\begin{bmatrix}
r_1 \\
r_2
\end{bmatrix}
\tag{2.65}
$$

The left side matrix is decomposed using block LU decomposition and the system is solved in 3 steps:

$$\begin{bmatrix} A & C_1 \\ C_2 & D \end{bmatrix} = \underbrace{\begin{bmatrix} A & \mathbf{0} \\ C_2 & D - C_2 A^{-1} C_1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} I & A^{-1} C_1 \\ \mathbf{0} & I \end{bmatrix}}_{U} \tag{2.66}$$

- Prediction: $A\boldsymbol{u}^* = r_1$
- Apply constraints: $(D - C_2 A^{-1} C_1)\boldsymbol{\lambda} = r_2 - C_2 \boldsymbol{u}^*$
- Correction: $\boldsymbol{u} = \boldsymbol{u}^* - A^{-1} C_1 \boldsymbol{\lambda}$

Here $(D - C_2 A^{-1} C_1)$ is called the Schur complement of the block matrix. Now we can apply the block LU decomposition to our coupled saddle point system in details.

### 2.4.1 Prediction step

$$\begin{bmatrix} H_{i,i-1} & B_1^T \\ B_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}_{i-1}^* \\ \boldsymbol{f_b}_{i-1}^* \end{bmatrix} = \begin{bmatrix} r_{11} \\ r_{21} \end{bmatrix} \tag{2.67}$$

This is very similar to fluid solver with stationary body or body with prescribed motion, only that here $B_1^T$ and $B_2$ are evaluated at slightly different time stages. We apply another step of block LU decomposition to solve this matrix. The procedure is exactly the same with the general steps only with change in symbols. The inverse of Schur complement for this matrix is stored for later use.

$$(S_f)^{-1} = \left(B_2 H_{i,i-1} B_1^T\right)^{-1} \tag{2.68}$$

For block LU decomposition on the fluid system alone, the prediction step advances the flow without immersed boundary. In the second step, the reaction force for the generation of a vortex sheet on the immersed surface to enforce no-slip and no-flow-through conditions is computed as Lagrange multipliers. In the third step, fluid is corrected by the vortex sheet. Either direct method like LU decomposition or iterative method like GMRES can be used to solve for the fluid system.

### 2.4.2 Apply constraints

Rather than calling this the "apply constraints" step, it actually makes more sense to call it "modify body inertia by fluid added-mass" step. To calculate Schur complement of the overall matrix, the inverse of the whole matrix in the prediction step is needed. This is done by the general block matrix inverse.

$$\begin{bmatrix} H_{i,i-1} & B_1^T \\ B_2 & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} H_{i-1,i} - H_{i-1,i}B_1^T(S_f)^{-1}B_2H_{i-1,i} & H_{i-1,i}B_1^T(S_f)^{-1} \\ (S_f)^{-1}B_2H_{i-1,i} & -H_{i-1,i} \end{bmatrix} \tag{2.69}$$

Substitute in to get

$$\begin{bmatrix} \mathbf{0} & -T_1^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} H_{i,i-1} & B_1^T \\ B_2 & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -T_2 & \mathbf{0} \end{bmatrix} = \begin{bmatrix} -T_1^T(S_f)^{-1}T_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.70}$$

Rearrange, now we are solving for

$$\begin{bmatrix} \mathbb{M} + T_1^T(S_f)^{-1}T_2 & G_1^T \\ G_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} = \begin{bmatrix} r_{12} + T_1^T \boldsymbol{f_b}^*_{i-1} \\ r_{22} \end{bmatrix} \tag{2.71}$$

Note that the matrix in equation (2.71) is the Schur complement of the overall fluid–body system. Also note that the dimension of this saddle system will not be large, since it only contained the number of body degree of freedom plus the number of joint constraint. So this matrix is solved through direct method with very little cost. When we look at equation (2.71) and analyze a body system immersed in fluid, the things that reveal the effect of the fluid are the fluid force on the body (on the right side with $\boldsymbol{f_b}^*$) and the term $T_1^T(S_f)^{-1}T_2$ which altered the intrinsic inertia of the body system, acting like added-mass. If we substitute in $(S_f)^{-1}$, we will find out that $T_1^T(B_2H_{i,i-1}B_1^T)^{-1}T_2$ is an operator going from body motion space to force space with the effect of viscous diffusion through the integrating factor $H$. It is also a correction force exhibited on the immersed surface to enforce the no-slip, no-flow-through boundary condition. It can be interpreted with steps as following:

- $T_2$ interpolates rigid body's acceleration onto Lagrangian points on the immersed surface

- $(B_2 H_{i,i-1} B_1^T)^{-1}$ regularized on the Cartesian grid, enforce no-slip, no-flow-through boundary condition by creating a vortex sheet around the immersed body, have that vorticity diffused, and then interpolate back to the Lagrangian points

- $T_1^T$ integrates all forces on Lagrangian points as an external force for the rigid body

This term is in the same place with body inertia, so it has the unit of mass. This is not the classical added-mass defined by inviscid flow, but it is the most physically relevant measure of added-mass in a viscous flow. The most amazing thing is that this term pop out through a LU decomposition step, derived rigorously using simple linear algebra tool, benefit from the use of monolithic coupling scheme. The added-mass term is not constructed in an ad hoc way, but it justifies the use of added-mass. With body immersed in fluid, this term will always be greater than zero. In this case, body with arbitrarily low mass ratios should work well with this coupling scheme, even if the body system has zero mass. There should be no stability problem with respect to any mass ratio, as long as grid resolution and time step are fine enough for large density ratios. The stability to mass ratios is shown in section 3.2 through examples.

### 2.4.3 Correction step

$$
\begin{aligned}
\begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \end{bmatrix} &= \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} \\ \boldsymbol{f_b}^*_{i-1} \end{bmatrix} - \begin{bmatrix} H_{i,i-1} & B_1^T \\ B_2 & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -T_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} + H_{i-1,i} B_1^T (S_f)^{-1} T_2 \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{f_b}^*_{i-1} - (S_f)^{-1} T_2 \dot{\boldsymbol{u}}_{i-1} \end{bmatrix}
\end{aligned}
\tag{2.72}
$$

Note that on the left hand side of equation (2.63), we choose to put fluid blocks on the top left corner. This choice gets revealed in the block LU decomposition, where fluid is first predicted, and rigid body system gets modified. There can be another choice of putting rigid body system on the top left corner, which would result in a formulation similar to the idea in the original immersed boundary method paper by Peskin [24]. For the alternative matrix arrangement, it would be computing body motion without fluid, modifying fluid Schur complement with body inertia, then correct body motion with the effect of fluid. Methods in this idea is offered for further studies.

### 2.4.4 Modification for two-dimensional body with fictitious fluid inside

We know that Lagrange multipliers in immersed boundary method is responsible for the jump on two sides of the Lagrangian points. When two-dimensional body is immersed in the fluid, we have to account for the momentum of the fictitious fluid inside. When the rigid body is only in translational motion (like a falling cylinder) or the rigid body has small aspect ration (like a thin ellipse), we could reasonably assume the fictitious fluid is also under the same rigid body motion. To construct the real fluid force on the body instead of jump of forces, $\mathfrak{f}$ is used instead of $\boldsymbol{f_b}$.

- For infinitely thin body, there's no fictitious fluid inside rigid body region. $\mathfrak{f} = -B_1^T \boldsymbol{f_b}$, same as before.

- For two-dimensional body,

$$\begin{aligned} \mathfrak{f} &= -B_1^T \boldsymbol{f_b} + B_1^T (T_1)^{-T} \frac{d}{dt}(\mathbb{M}_f \boldsymbol{u}) \\ &= -B_1^T \boldsymbol{f_b} + B_1^T (T_1)^{-T} \left( \mathbb{M}_f \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{u} \times^* \mathbb{M}_f \boldsymbol{u} \right) \\ &= -B_1^T \boldsymbol{f_b} + B_1^T (T_1)^{-T} \left( \mathbb{M}_f \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right) \end{aligned} \tag{2.73}$$

  Here $\mathbb{M}_f$ is the inertia matrix for fictitious fluid. Since we assume there's negligible rotation motion for the fictitious fluid, $\boldsymbol{u} \times^* \mathbb{M}_f \boldsymbol{u}$ vanishes. The coupling scheme should be modified based on this change accordingly. Note here we are using a term $(T_1)^{-T}$, which can't be constructed, but also don't need to be constructed as proved later. Remember that $T_1^T$ means an integral of fluid force over every Lagrangian point, then transform from inertia coordinate to body local coordinate. Here we need exactly the inverse operation so that the body momentum described in the body local coordinate can be transformed to inertia coordinate, then magically gets interpolated along the Lagrangian points. This operation can not be constructed, since there's infinite ways to do the interpolation. However, we just need the concept of this operation and keep it as $T_1^T$, finally it will gets cancelled in the block LU decomposition with $T_1^T$.

To involve in the modification to the system, there are also two ways to do it, as mentioned in Lācis etc. [92]. We choose the following way so that the fictitious fluid inertia term also shows at the same place with body inertia and added-mass. The coupling matrix now becomes

$$\begin{bmatrix} H_{i,i-1} & B_1^T(\boldsymbol{q}_{i-1}) & B_1^T(\boldsymbol{q}_{i-1})(T_1)^{-T}\mathbb{M}_f & \mathbf{0} \\ B_2(\boldsymbol{q}_i) & \mathbf{0} & -T_2 & \mathbf{0} \\ \mathbf{0} & -T_1^T & \mathbb{M} & G_1^T(\boldsymbol{q}_{i-1}) \\ \mathbf{0} & \mathbf{0} & G_2(\boldsymbol{q}_i) & \mathbf{0} \end{bmatrix} \tag{2.74}$$

Going through the same process of LU decomposition, the prediction step doesn't change. Equation (2.75) and (2.76) shows the change of second and third step in block LU decomposition respectively

$$\begin{bmatrix} \mathbb{M} + T_1^T(S_f)^{-1}T_2 - \mathbb{M}_f & G_1^T \\ G_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} = \begin{bmatrix} r_{12} + T_1^T \boldsymbol{f_b}^*_{i-1} \\ r_{22} \end{bmatrix} \tag{2.75}$$

$$\begin{aligned} \begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \end{bmatrix} &= \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} \\ \boldsymbol{f_b}^*_{i-1} \end{bmatrix} - \begin{bmatrix} H_{i,i-1} & B_1^T \\ B_2 & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} B_1^T(T_1)^{-T}\mathbb{M}_f & \mathbf{0} \\ -T_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} + H_{i-1,i}B_1^T(S_f)^{-1}T_2\dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{f_b}^*_{i-1} - (S_f)^{-1}T_2\dot{\boldsymbol{u}}_{i-1} + T_2\mathbb{M}_f\dot{\boldsymbol{u}}_{i-1} \end{bmatrix} \end{aligned} \tag{2.76}$$

We can see that the body inertia in equation (2.75) is not only modified by added-mass, but also by the inertia of fictitious fluid. Also on the right hand side of equation (2.76), fluid constraint force got influenced by the fictitious fluid as well.

## 2.5 Summary

We should address three main questions here, as a summary on what's novel and what's the benefit of our monolithic method compared to previous studies on FSI problems. First question is what's the advantage of a monolithic method, despite the algorithm and code construction is more complicated than a partitioned method. It is realized that weakly-coupled methods are not frequently used except for certain fields (such as aeroelasticity) due to the stability issues. Strongly-coupled methods are actually used most frequently, because they are reasonably easy to construct and have a guarantee for convergence to some extent. One can construct the fluid system and body system on its own, and then describes how the interaction works through iterations. However the iteration process is still error-prone and could grow unstable. Convergence speed is also a problem, as addressed in section 1.1. In some strongly-coupled methods like Wang and Eldredge [13], added-mass effect is used explicitly to speed up iteration convergence, while this is done in an ad-hoc way. With our monolithic solver, the iteration process is removed. By the benefit of block LU decomposition, added-mass term for viscous flow augments the body inertia in a rigorous way. This also justifies why explicitly considering added-mass could work for previous studies. Also we are able to get stable solution for arbitrarily low density ratios including neutrally-buoyant and zero mass cases, while these cases are unstable in many FSI solvers from previous studies.

The second question is that how do we make our algorithm monolithic. Monolithic refers to solving all unknown variables in the system simultaneously with a unified time marching scheme. We take the advantage of the immersed boundary description for the fluid system, where the use of Lagrange multipliers makes the problem differential-algebraic. Fluid solver uses Lagrange multipliers to enforce boundary condition, and rigid body solver uses Lagrange multipliers to enforce motion constraint. In this way, both systems have the same mathematical form so that variables can be solved simultaneously. Half-explicit Runge-

Kutta method is designed to solve differential-algebraic problems, thus it is used naturally as the common time marching scheme for both systems.

The last question is what makes our algorithm versatile. It is versatile in the sense of being able to create rigid body systems that are quite different within the same framework. This is mainly a result of our rigid body system representation. The use of joint models allows us to combine joints of different types in the rigid body system. We also systematically describe the hierarchy connection of bodies, which also contributes to the versatility. The other contribution to versatility comes from the use of Lagrange multipliers in rigid body solver to enforce active prescribed motions. So we could solve a problem that involves in both active and passive motions of bodies, meanwhile bodies of different shapes are connected in different ways by different types of joint.

# CHAPTER 3

# Results and discussion

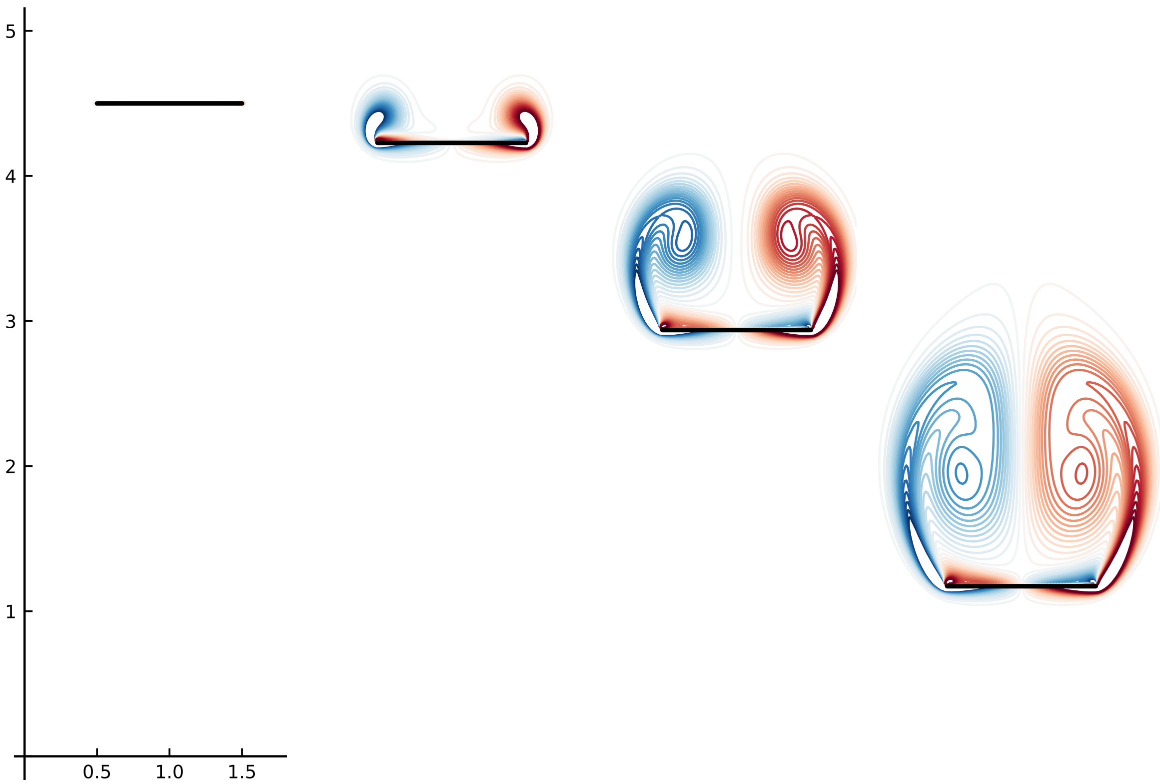
In this chapter, a variety of applications about the monolithic fluid–structure interaction solver is presented. Body configurations include cylinder, ellipse, airfoil and infinitely thin bodies connected by spring and damper. These body-joint chains are under active or passive motion, depending on the focus of the study. The versatile applications with very little modifications in code set up demonstrate the flexibility of the current method.

The first examples is the free falling of a flat plate under its own weight in a quiescent fluid, used as the convergence test for the coupling scheme. The second example discussed the free falling/rising of circular cylinders with different mass ratios, enhanced the statement in section 2.4 about stability with arbitrarily small (including zero) mass ratios. The third example involves in the study of vortex shedding by a symmetric airfoil under active pitching oscillations. The idea comes from Schnipper et al. [7], where an experimental study with the same set up is presented. Numerical results from our study show very good consistency in vortex wake forms compared with [7]. We consider this as an example of the solver being capable of handling passive and active body motion in the same framework. The fourth example describes the free swimming of a three component articulated fish, with results in good consistency compared to Eldredge [12]. Swimmers with different shapes and mass ratios are also incorporated to study for swimming effectiveness. The next example used multiple linked plates to simulate the motion of a self-excited flapping flag under gravity in viscous flow, comparing to results in Wang and Eldredge [13] and Huang et al. [82]. Special attention is given to the last example, a problem of two flat plate connected by spring and damper oscillating in a uniform flow. Similar to the case of flapping flag, the two-body linked system also has two stable states: fixed-point stability (body staying straight) and limit-cycle stability (self-sustained flapping). Critical mass ratios for flapping state regarding spring stiffness at different Reynolds number are found, with similar trend in Connell and Yue [10] for flapping flags. Critical mass ratios for flapping state regarding damping coefficient is also studied. Hysteresis behavior is investigated through the variation of equivalent reduced velocity, and it is found to be different from the bi-stability behaviour of flags in Eloy et al. [9]. Finally, the energy harvesting aspect of this model is addressed.

## 3.1 Falling flat plate

In this section, we consider an infinitely thin flat plate of unit length $L = 1$ and density twice the fluid density $\rho_b/(\rho_f L) = 2$ falling in a quiescent fluid under its own weight for convergence test and the validation for overall order of accuracy. In this two-dimensional study, fluid density is defined as mass per unit depth, while density of the infinitely thin plate is mass per unit area. The plate is released from rest and is allowed for free falling under gravity $g = -1$ for 4 time units. Reynolds number characterized by $Re = g^{1/2}L^{3/2}/\nu$ is kept at 200 and the computation domain is $[0, 2] \times [0, 5]$. The plate's degree of freedom is constrained in all degrees except for the falling y-direction. This is achieved by constructing a prismatic joint in y-direction with zero stiffness and damping coefficient. The time marching scheme for all examples is second order half-explicit Runge–Kutta (HERK) method described in table 2.1 by default if not specifically mentioned. In order to evaluate the overall order of accuracy, three sets of grid space shown in table 3.1 are used, with grid size $\Delta x = 0.01$, 0.02 and 0.04 respectively. The CFL number is fixed to be $(gL)^{1/2}\Delta t/\Delta x = 0.25$. Thus time step $\Delta t$ for the three grid resolutions are $0.02, 0.01, 0.005$ respectively.

The vorticity field for the falling plate at four instants are depicted in figure 3.1 for the most fine grid with $\Delta x = 0.01$ and $\Delta t = 0.005$. Without perturbation, the wake vortices roll up and get stretched as the plate falls. The velocity and acceleration history of the falling plate is presented in figure 3.2 for all three sets of grids. We can find out that the plate acceleration is decreasing towards 0, trying to reach the force balance but limited by the computation domain size and time. The exact values of plate initial acceleration can




(a) (b) (c) (d)

Figure 3.1: Vorticity field for falling flat plate at four instants $t = 0,\ 1,\ 2.5,\ 4$ with $\Delta x = 0.01, \Delta t = 0.005$. Vorticity contour plotted from -10 to 10 in 40 uniform increments.

be compared to the value computed from added-mass in equation (3.1), since the diffusion hasn't participated much and the initial reaction is mostly inviscid. From table 3.1 we can see that with grid resolution improving, plate initial acceleration approaches to the value calculated with equation (3.1), which is a proof for solver convergence.

$$a = -\frac{m_b}{m_b + m_{\text{added}}} = \frac{2\rho_f}{2\rho_f + 0.25\pi\rho_f L^2} = \frac{2}{2 + 0.25\pi} \approx -0.7180 \tag{3.1}$$

The evaluation of overall order of accuracy is conducted using plate terminal acceleration with Richardson extrapolation in equation (3.2). Since the spatial accuracy should be between one and two for immersed boundary method, time accuracy should be two for second order HERK scheme, the overall order of accuracy 1.22 is reasonable.

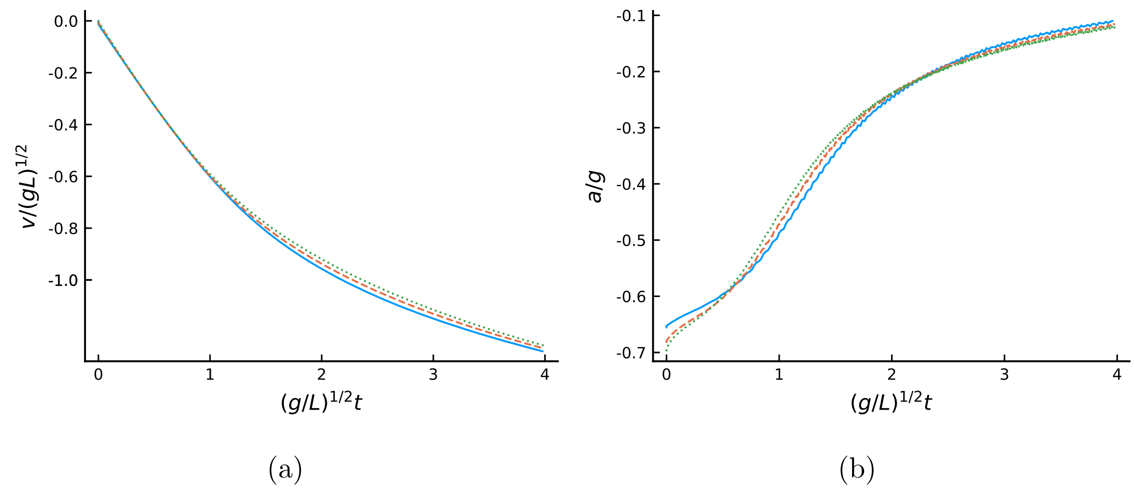



Figure 3.2: (a) Velocity and (b) acceleration of the falling flat plate with all three sets of grid, solid for case 1, dash for case 2, dot for case 3.

| case | $\Delta x$ | $\Delta t$ | $a_0$ | $a_{\text{end}}$ |
|---|---|---|---|---|
| 1 | 0.04 | 0.02 | -0.6555 | -0.1106 |
| 2 | 0.02 | 0.01 | -0.6829 | -0.1179 |
| 3 | 0.01 | 0.005 | -0.6980 | -0.1211 |

Table 3.1: Initial acceleration $a_0$ and terminal acceleration $a_{\text{end}}$ of the plate at $t = 4$

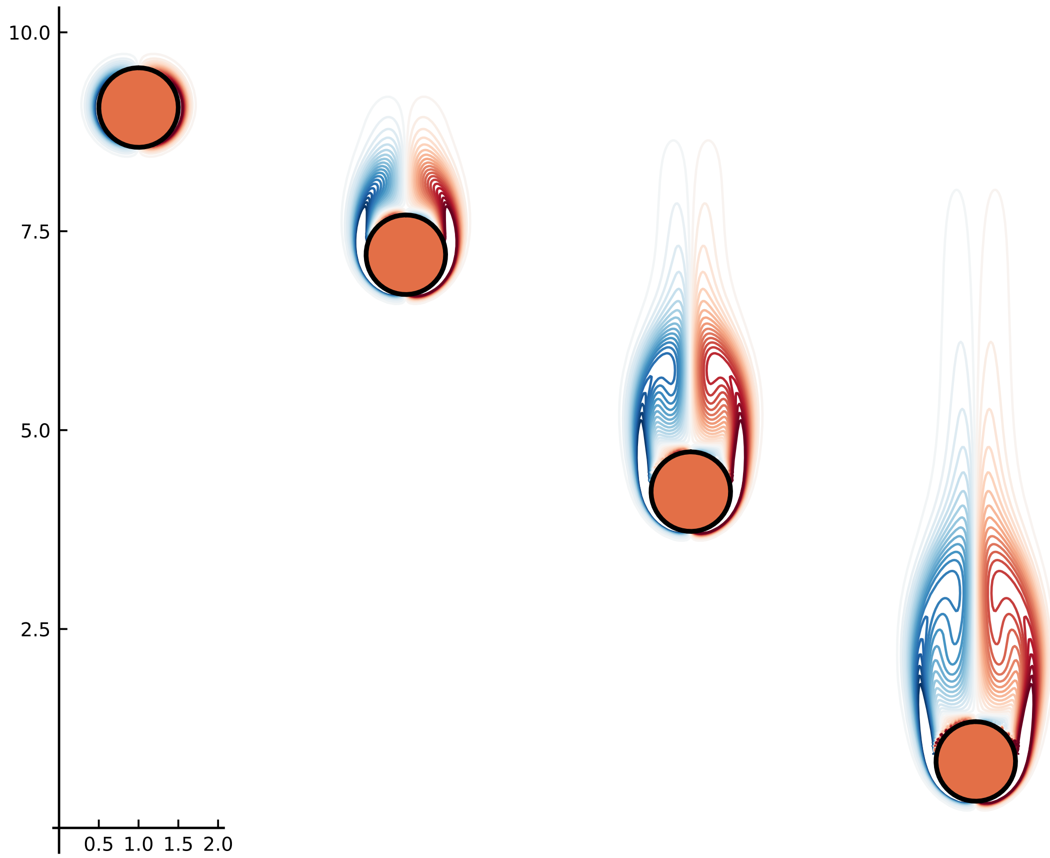
$$k = log_2 \frac{|a_{\text{end},1} - a_{\text{end},2}|}{|a_{\text{end},2} - a_{\text{end},3}|} = 1.22 \tag{3.2}$$

## 3.2 Falling/rising cylinder with different mass ratios

The free falling or rising of a circular cylinder can be viewed as a special case of the extensively studied problem — vortex induced vibrations of elastically mounted body [5] — where the spring has zero elasticity. We pick four characteristic mass ratios to show different reactions of a cylinder with diameter $D = 1$ immersed in viscous fluid with gravity and buoyancy forces. The cylinder is only allowed to move in vertical direction, with all other degrees of freedom constrained. Similar to the falling plate case, a prismatic joint with zero stiffness and damping is used to connect the cylinder and the inertia reference frame. First we pick a cylinder denser than the fluid $\rho_b = 2\rho_f$ and release it from rest. Neutrally buoyant cylinder $\rho_b = \rho_f$ is also of interest, since this special mass ratio would cause stability problem in some FSI solvers. We also choose a cylinder lighter than the fluid $\rho_b = 0.5\rho_f$, and another limiting case of $\rho_b = 0$, which is not really possible to construct for weakly-coupled or strongly-coupled FSI solvers if not manually introducing terms like added-mass to modify the body's intrinsic inertia. For all three cases of $0 \leq \rho_b/\rho_f \leq 1$, we first drag the cylinder down by imposing a linearly increasing velocity through active motion in one time unit, then release the cylinder for passive reactions. Reynolds number $Re = g^{1/2}D^{3/2}/\nu$ for all cases are kept at 200, gravity is set to be -1. Numerical parameters include grid space $\Delta x = 0.02$ and time step $\Delta t = 0.01$. The case with heavy body $\rho_b = 2\rho_f$ is on a domain of size $[0, 2] \times [0, 10]$ and computed for 10 time units. The other three cases are computed for 8 time units on a bigger domain of $[0, 4] \times [0, 12]$ to contain wider wake vortices.

For case with heavy body $\rho_b = 2\rho_f$, vorticity field at four instants are shown in figure 3.3. Time history of position, velocity, acceleration and drag coefficient is depicted in figure 3.4. We can observe from figure 3.3 that a pair of vortex with opposite sign stretch behind the body without external perturbation. Comparison in time history plots is made with Eldredge [12], where a viscous vortex particle method is applied to model the fluid force on the body. From figure 3.4 we can see that good consistency is achieved.

The vorticity filed of case $\rho_b = \rho_f$, $\rho_b = 0.5\rho_f$ and $\rho_b = 0$ are plotted side by side with the same contour levels in figure 3.5 for three instants $t = 1$, $t = 5$ and $t = 9$. Starting from time $t = 0$, Cylinder in these cases are pushed down with linearly increasing prescribed velocity until $v = -1$ is reached at $t = 1$,. As a results, it can be seen at $t = 1$, vorticity filed and body kinematics for all three cases are identical from figure 3.5 and figure 3.6. After that, the prescribed motion is removed and cylinder is set to free. With active prescribed motion removed, the cylinder started dynamically coupled phase with the fluid. The neutrally buoyant cylinder continues to fall and slow down at the meantime. Wake behind the cylinder is suppressed by the slowing down motion and becomes weaker with diffusion. Drag coefficient for this case quickly approaches zero under the balance of




(a) (b) (c) (d)

Figure 3.3: Vorticity field for circular cylinder with density $\rho_b = 2\rho_f$ falling from rest at four instants $t = 1, 4, 7, 10$. Vorticity contour plotted from -5 to 5 in 40 uniform increments.

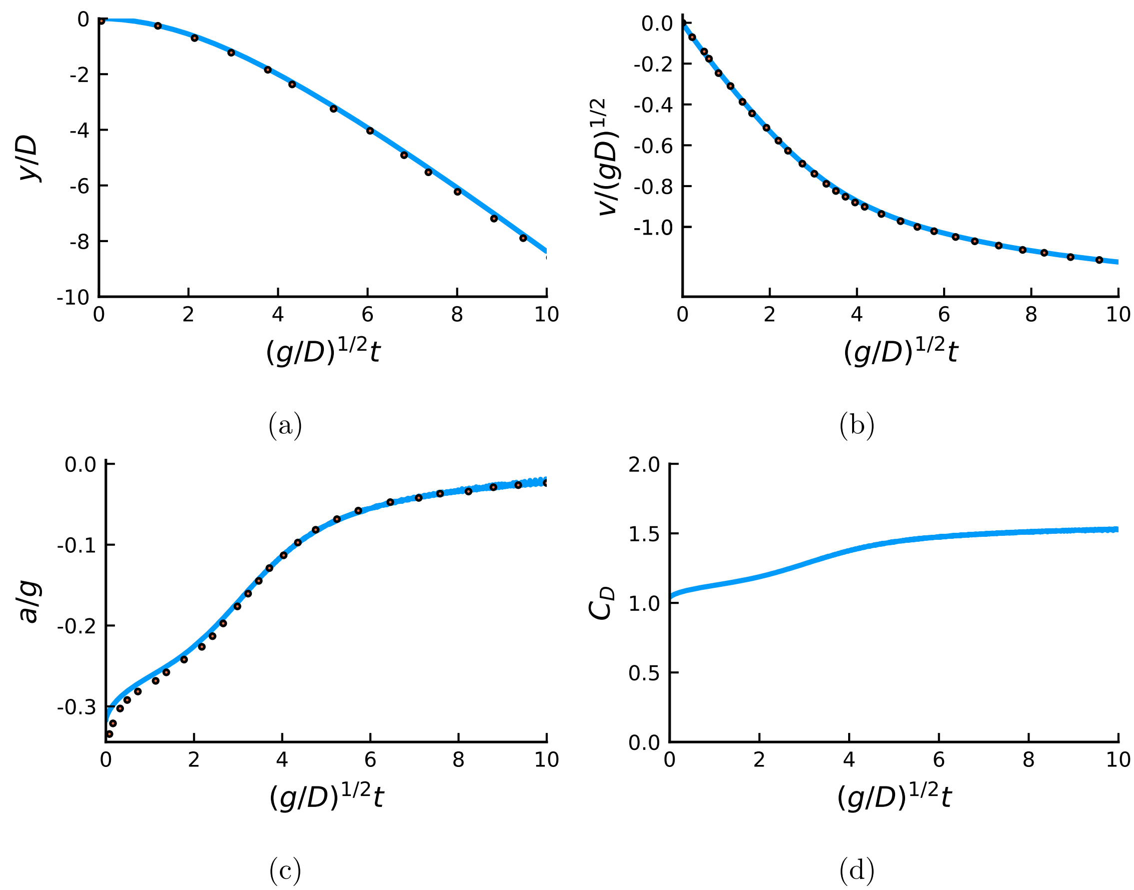



Figure 3.4: (a) Position, (b) velocity, (c) acceleration, (d) drag coefficient of the cylinder with $\rho_b = 2\rho_f$ falling from rest, blue solid; Eldredge [12], dotted.

buoyancy and gravity after the initial impact. With longer computation time and a bigger domain, we should be able to see the cylinder finally stops after the initial kinetic energy of both fluid and body being diffused by the viscous effect. On the other hand, the cases of light cylinder $\rho_b = 0.5\rho_f$ and zero mass cylinder turn to a rising attitude after the initial impact. The vortices generated in the falling phase are shed as the cylinder direction turns upward. A new pair of vortices appear and get stretched in the wake. We can see from figure 3.6 (c) that the zero mass cylinder approaches to steady state faster than the $\rho_b = 0.5\rho_f$ case, and its wake is stronger as well.

With the case of $\rho_b = 0$, we proved that the algorithm is stable for body with arbitrarily small mass ratios including zero mass case. It is also stable to large mass ratios, as long as the grid resolution is fine enough and time step is small enough under the constraint of CFL number. Again, it benefits from the formulation of the algorithm, where the added-mass term is naturally combined with body's intrinsic inertia to ensure the combined mass always positive from block LU decomposition.

## 3.3 Vortex shedding with active airfoil

The goal of this example is to provide a demonstration on the algorithm's capability to deal with active and prescribed body motion with the same mathematical framework. Both the enforcement of active motion and joint constraint are expressed through Lagrange multipliers, so they are treated in the same way. A good example comes from Schnipper et al. [7], where the rich behaviour of vortex wake of a symmetric airfoil with forced oscillation is presented by experiments. It is desired for us to get the same vortex structures from numerical simulation. The original experiment makes use of a gravity-driven, vertically flowing soap film with uniform flow velocity $U$, which functions as a two-dimensional flow. A schematic plot of the airfoil configuration is shown in figure 3.7, which consists of a circular leading edge and straight trailing edge. The ratio between the leading edge diameter $D$ and total length $C$ is $D/C = 2/11$. The airfoil is under sinusoidal forced oscillation around pivot point $O$ with peak amplitude $A$ and frequency $f$. The non-dimensional parameters for forced oscillation

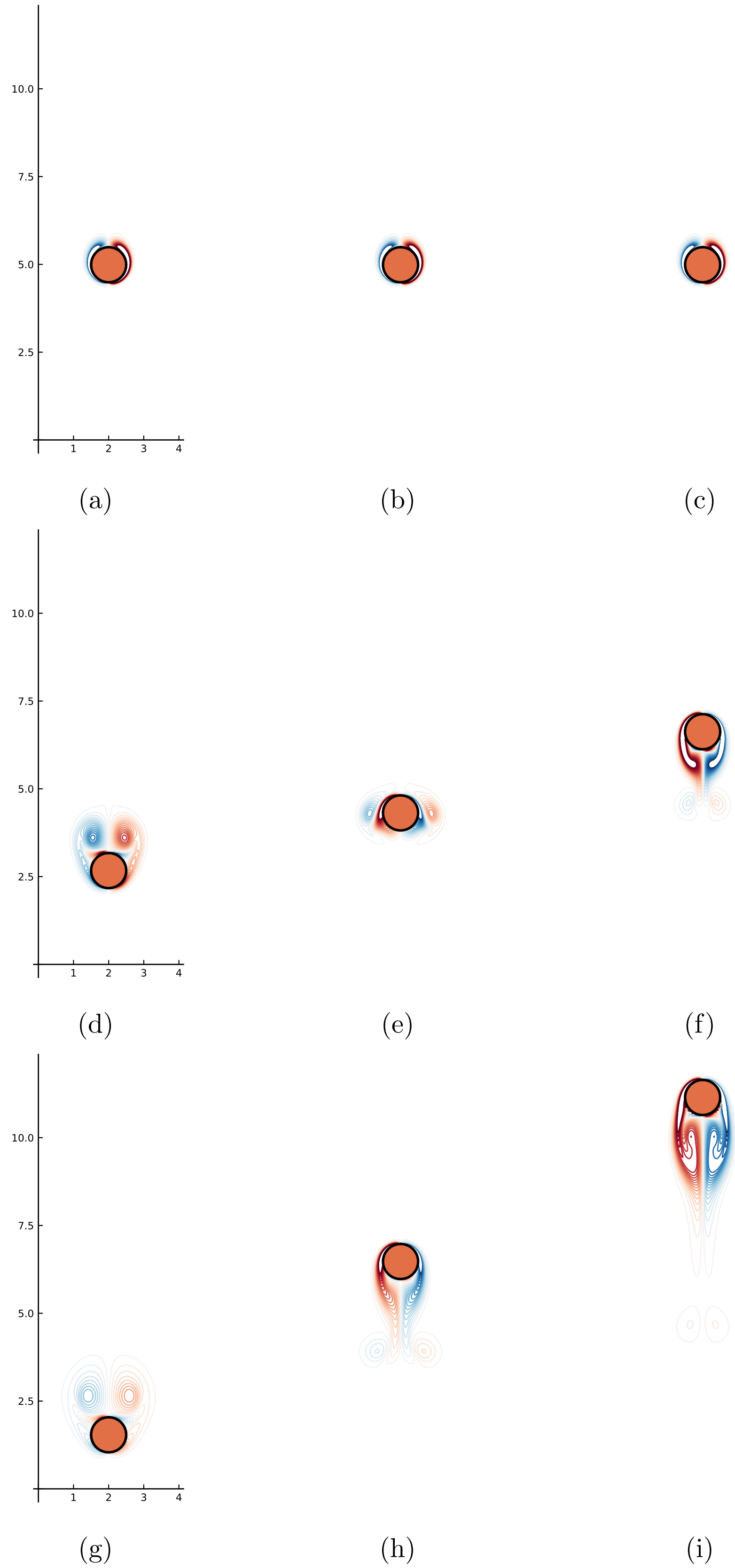
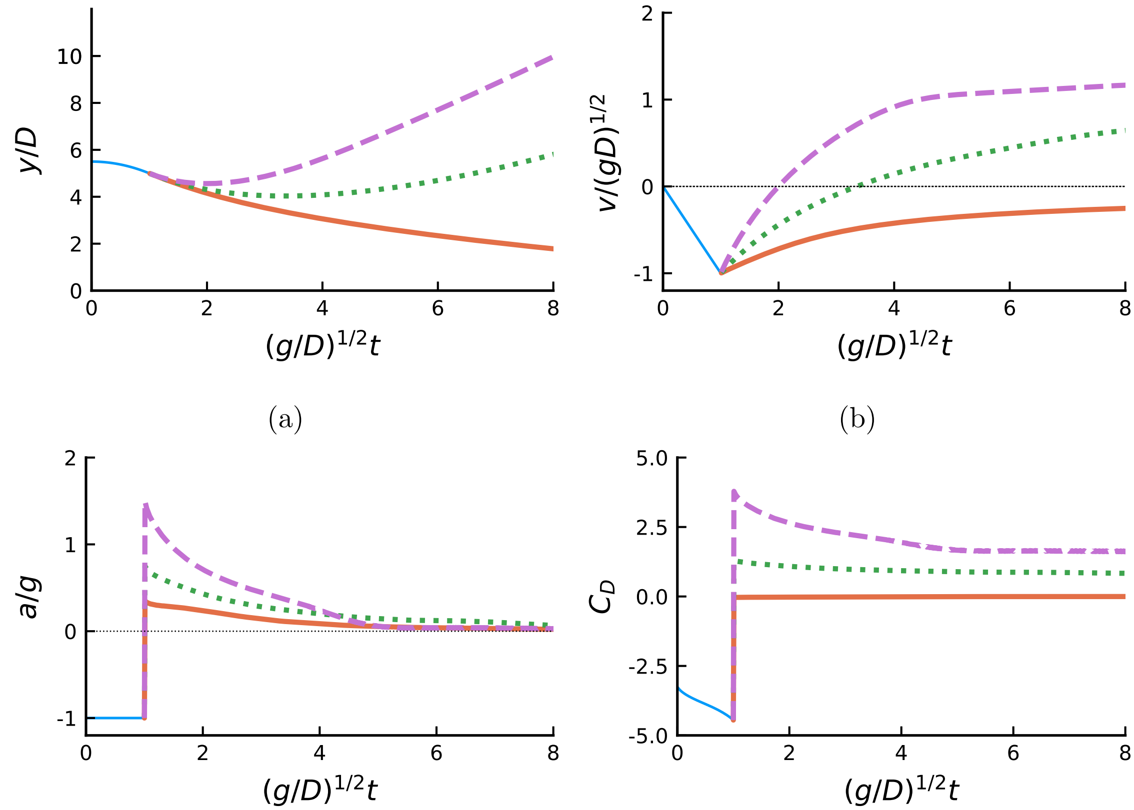



Figure 3.5: Vorticity contour plotted from -5 to 5 with 40 uniform increments for cylinder with density $\rho_b = \rho_f$, $\rho_b = 0.5\rho_f$ and $\rho_b = 0$ respectively at time (a-c) $t = 1$ when initial velocity $v_y = -1$ is achieved through prescribed motion; then release for passive reactions plotted at (d-f) $t = 5$ and (g-i) $t = 9$.




(a) (b) (c) (d)

Figure 3.6: (a) Position, (b) velocity, (c) acceleration, (d) drag coefficient of the cylinder falling with initial velocity for different density ratios, $\rho_b = \rho_f$, orange solid; $\rho_b = 0.5\rho_f$, green dotted; and $\rho_b = 0$, purple dashed.

defined in this problem are non-dimensional amplitude $A_D = 2A/D$ and Strouhal number $St_D = Df/U$. The prescribed motion is enforced through a revolute joint at the pivot point $O$. Reynolds number is defined based on leading edge diameter $D$ as $Re = DU/\nu$. The computation domain is $[0, 4] \times [0, 10]$ with grid space $\Delta x = 0.02$. Simulation for each case with different $A_D$ and $St_D$ is conducted for 30 time units with time step $\Delta t = 0.01$. We put 124 Lagrangian points on the airfoil boundary with uniform spacing same as grid space.

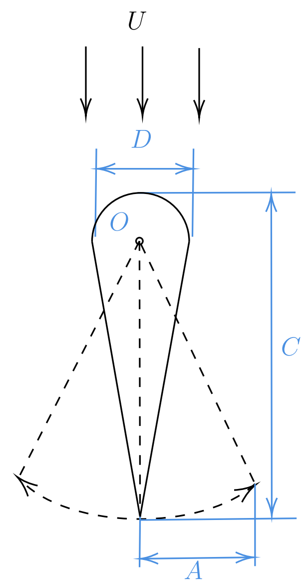



Figure 3.7: Schematic illustration of the oscillating airfoil

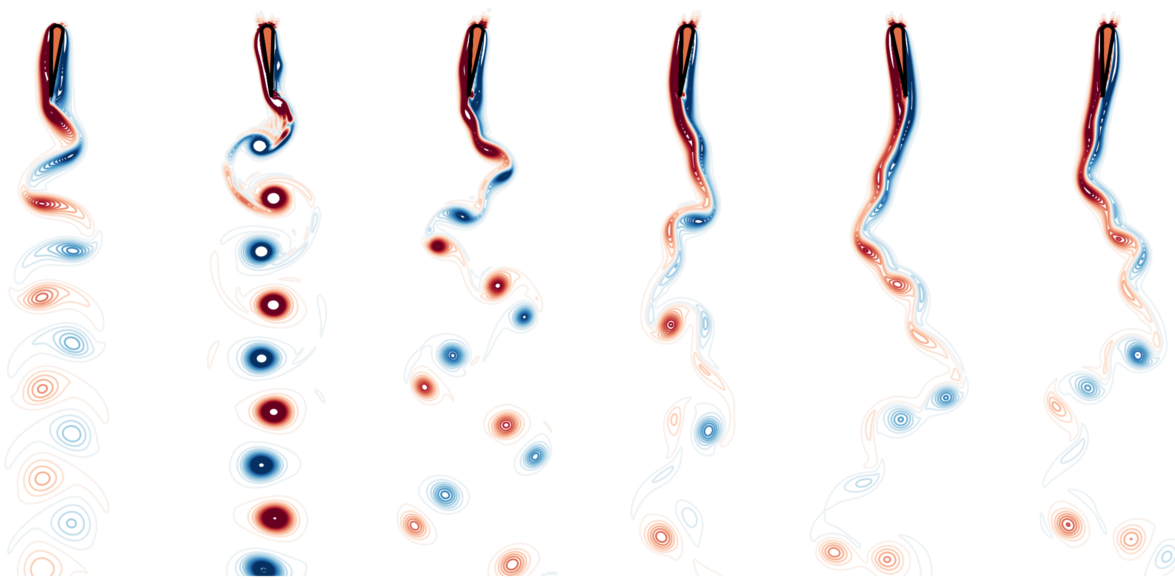
As reported in Schnipper et al. [7], the type of wake vortex generated is not sensitive to $Re$ for low to moderate Reynolds number. Thus we only need to vary oscillation amplitude $A_D$ and Strouhal number $St_D$ to set airfoil active motion for different cases. In our numerical simulation, we fix $Re = 200$ except for the first case in figure 3.8 (a) where $Re = 80$. We successfully observed very similar von Kármán (2S), inverted von Kármán, 2P, 2P+2S, 4P and 4P+2S vortex structures with the same forced oscillation amplitude and Strouhal number. The original plot from experiments in [7] is reprinted here for direct comparison in figure 3.9.

In terms of the terminology, 2S wake is just von Kármán wake, with "S" denoting a single vortex shed in half an oscillation period. Similarly, "P" in 2P refers to a pair of vortices of opposite sign shed in half a period. With $St_D = 0.12$ and $A_D = 0.98$ (figure 3.8 (a)), we observe the standard staggered von Kármán vortex street, where boundary layer separation on each side of the airfoil happens in half an oscillation period. By keeping the same Strouhal number $St_D = 0.12$ but a larger oscillation amplitude $A_D = 2.0$, the inverted von Kármán wake (figure 3.8 (b)) has stronger vortices shed aligned near the centreline. This thrust-generation related mode is discussed in more details in section 3.4 with a free swimming example. When Strouhal number is lowed down to 0.08, which is lower than side boundary layer separation frequency, more than one vortex is shed per half oscillation period. For 2P wake (figure 3.8 (c)), in each pair of vortices plotted with red and blue color, one of them is shed at airfoil oscillation peak, and the other one shed when airfoil pass the centreline at largest tip velocity. As a result two pair of vortices are shed in one period. In the regime map reported in [7], the transition from 2S mode to 2P mode can be achieved by keeping the same oscillation amplitude but lower the frequency. This is also observed with our numerical simulation (figure 3.10), where both cases have $Re = 120, A_D = 1.14$ while higher frequency $St_D = 0.12$ results in 2S mode, lower frequency $St_D = 0.08$ results in 2P mode.

With even smaller Strouhal number (figure 3.8 (d) for $St_D = 0.053$, $A_D = 1.2$, (e) for $St_D = 0.039, A_D = 1.34$, (f) for $St_D = 0.035, A_D = 1.47$), more vortices can be shed in one oscillation period. If half an oscillation period is about an integer (no less than 2) times of the interval for the vortex generated at the leading edge to advect with the uniform flow down to the tip, then the boundary layer vortex can be combined with the tip vortex to form modes like 2P and 4P. If not, modes involving 2S like 2P+2S or 4P+2S will take place. To identify more complex vortex modes like 8P reported in [7], we would need to use a bigger computation domain, which is applicable but not necessary in this study.

## 3.4 Free swimming of an articulated fish

Examples of fluid–structure interaction in section 3.1 to 3.3 all involve in the interaction between viscous fluid and a single body. In this section, we introduce a case with three bodies connected by two active joints (schematic plot in figure 3.11) interacting with viscous flow. This configuration can be interpreted as one of the simplest representation of an



(a) (b) (c) (d) (e) (f)

Figure 3.8: Vorticity field plotted from -10 to 10 in 40 uniform increments for: (a) 2S wake with $St_D = 0.12$, $A_D = 0.98$ and $Re = 80$; (b) inverted von Kármán wake with $St_D = 0.12$, $A_D = 2.0$ and $Re = 200$; (c) 2P wake with $St_D = 0.080$, $A_D = 1.4$ and $Re = 200$; (d) 2S+2P wake with $St_D = 0.053$, $A_D = 1.2$ and $Re = 200$; (e) 4P wake with $St_D = 0.039$, $A_D = 1.34$ and $Re = 200$; (e) 4P+2S wake with $St_D = 0.035$, $A_D = 1.47$ and $Re = 200$.

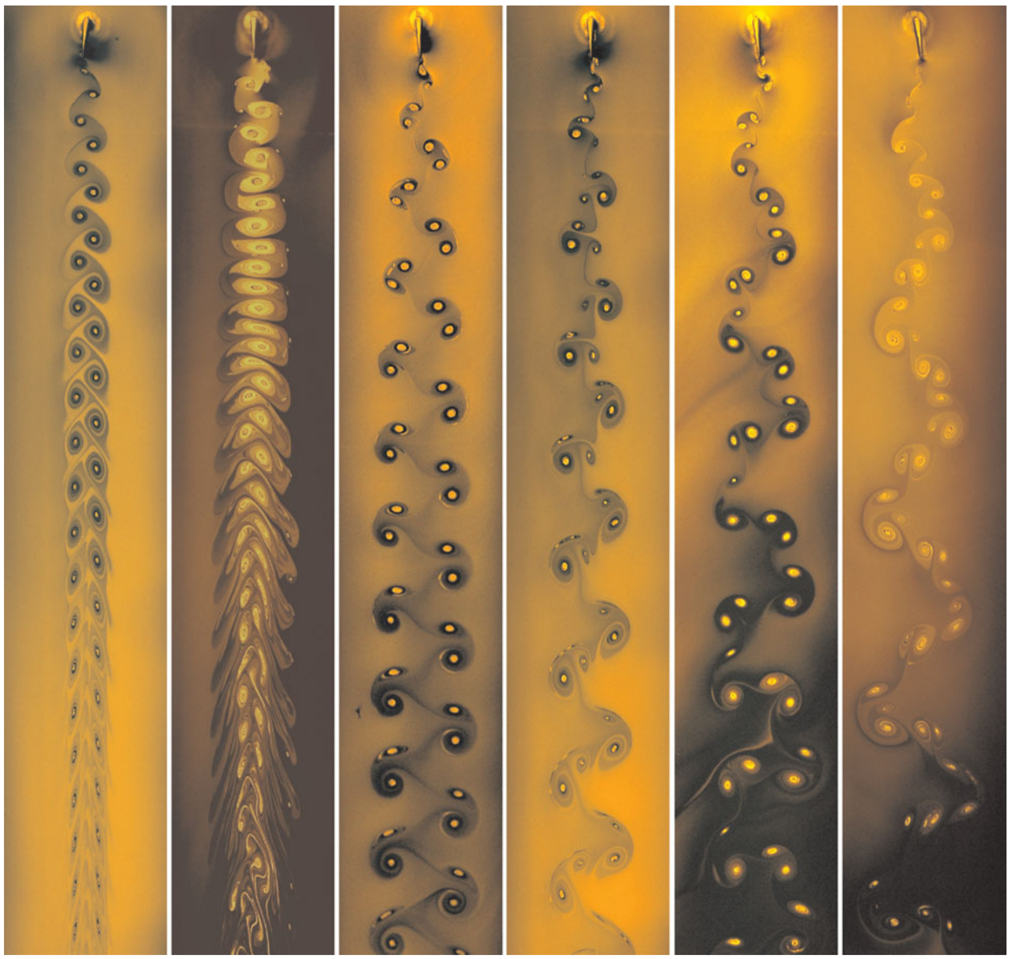
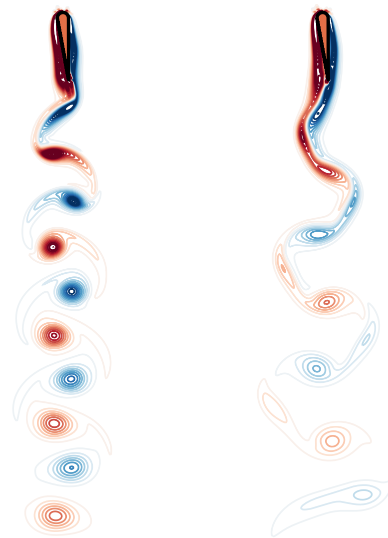


Figure 3.9: Reprint vortex shedding contour in Schnipper et al. [7], from left to right: (a) 2S wake for $St_D = 0.12, A_D = 0.98$; (b) inverted kármán vortex for $St_D = 0.12, A_D = 2.0$; (c) 2P wake for $St_D = 0.08, A_D = 1.4$; (d) 2P+2S wake for $St_D = 0.053, A_D = 1.2$; (e) 4P wake for $St_D = 0.039, A_D = 1.34$; (f) 4P+2S wake for $St_D = 0.035, A_D = 1.47$



(a) (b)

Figure 3.10: Vorticity field plotted from -10 to 10 in 40 uniform increments for: (a) 2S wake with $St_D = 0.12$, $A_D = 1.14$ and $Re = 120$; (b) 2P wake with $St_D = 0.12$, $A_D = 1.14$.

elongated body with undulatory locomotion. The schematic representation can be traced to invertebrates like nematoda or fish like eels in Lighthill [19]. This model is first brought up in Kanso et al. [100] to study self-propulsion of both zero mass and neutrally buoyant articulated rigid system in an inviscid flow. Eldredge [12] then extend the study to viscous flow using a coupled fluid–body viscous vortex particle method (VVPM) with massless bodies. Here we extend the study to direct numerical simulation with a monolithic method, applied to both zero mass and neutrally buoyant bodies with shapes of infinitely thin plates and ellipse. Discussion is made on self-propulsion effectiveness and energy cost among all cases.

To mimic the self-propulsion mechanism with desired motion, motion planning has to be chosen with care. Here we use the same prescribed motion reported in [100, 12] to actively control the second and third revolute joint as described with equation (3.3). The first joint is of planar type, allowing for rotation and translation in both x and y direction, and it is connected to the origin in hierarchy. Note that body 3 is the head and body 1 is the tail of the articulated fish. The motion of the first planar joint and the constraint forces

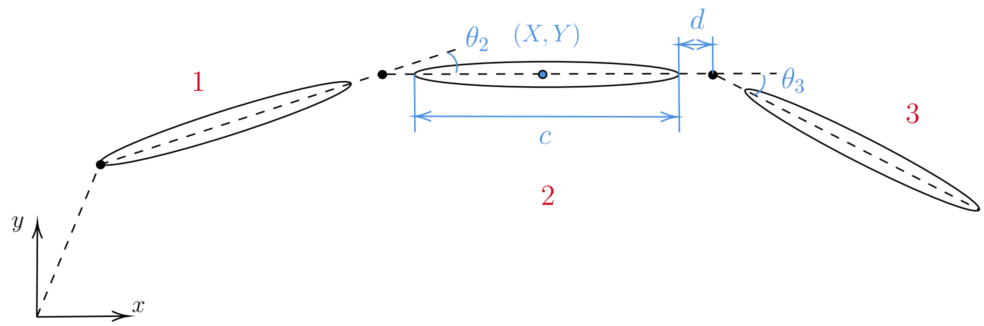



Figure 3.11: Schematic illustration of three articulated bodies

for the rest two revolute joints to maintain the prescribed motion are what we solve for. The enforcement of active motion on the revolute joint supplies zero net force in $x$ and $y$ direction, however the system succeeds in moving forward in space due to motion planning.

$$\theta_2(t) = -\cos(t - \pi/2), \qquad \theta_3(t) = -\cos(t) \tag{3.3}$$

For cases with plate shape, the plate length $c$ is 1 for each body. For ellipses, coord length $c$ for each body is set to 1, and aspect ratio of the ellipse is 10. Gaps are put between adjacent bodies for all cases, with the distance from body tip to the joint be $d = 0.1$. We choose two mass ratio ($\rho_b/\rho_f L$ for plate and $\rho_b/\rho_f$ for ellipse), 1 for neutrally buoyant case and 0 for massless case. Reynolds number $Re = (\dot{\theta}_{\max} c^2/\nu)$ is 200 for all cases. Computation domain is $[0, 8] \times [0, 3]$ with grid space $\Delta x = 0.02$. Simulation for each case is conducted for 25 time units with time step $\Delta t = 0.01$.

The vorticity contour for zero mass ellipse body at six instants is shown in figure 3.12. Both body position and vortex wake arrangement are in good agreement with Eldredge [12]. In each oscillation period, the leading edge of body 1 shed two vortex of opposite signs. Each of them is partially merged into its own boundary layer on the side, partially got cancelled by the other vortex of opposite sign shed in the last oscillation period. The tail of body 3 also sheds a pair of vortices of counter directions, but is convected and aligned to form the inverted von Kármán wake. As mentioned briefly in section 3.3, the inverted von Kármán wake is responsible for thrust generation, which is a perfect demonstration in this case since the body successfully moves forward. History of centroid position $(X, Y)$ of the body chain and rotation angle $\alpha$ of the central body with respect to horizontal direction is plotted in figure 3.13 with comparison to Eldredge [12]. The consistency is very good for rotation angle and marching distance in y-direction. Centroid position in x-direction in this study is a little smaller than results in [12]. The possibility for the difference is that the previous study used VVPM method to model the viscous fluid, which may not be as accurate as the direct simulation in this study.

We further compared vorticity contour for different body shapes and different mass ratios in figure 3.14. With history of tail position plotted in red, it's easier to compare the propulsion effectiveness for different cases. We can also compare the distance in x-direction that each case has marched forward more clearly in figure 3.15. For both plate and ellipse, it is true that smaller mass ratio has a better propulsion effectiveness. It makes sense since added fluid inertia is the same for both mass ratios as long as body shape is consistent. Driving a larger combined mass (intrinsic mass + added-mass) will lower the propulsion effectiveness. Among the four cases, massless plate prevails all while neutrally buoyant plate is the least effective regarding travelling distance in x-direction. The trailing edge vortices for both these two cases look similar, while leading edge vortices make a difference. In each oscillation period, the leading edge of the swimmer generates a pair of vortices with opposite signs. Similar to the role of leading edge vortex (LEV) for airfoils described in Eldredge and Jones [101], the LEV creates a low pressure region as the vortex develops. For massless plate, LEV swipes through the leading plate and merges nicely into the boundary layer (figure 3.16 (a)), providing a force driving the swimmer forward. On the other hand for the neutrally buoyant case (figure 3.16 (b)), the LEV gets pushed away from the body without swiping along the body much, thus produces little driving force. With similar driving force from the trailing edge but larger driving force from the leading edge, the massless plate has a longer travelling x-distance than the neutrally buoyant plate.

Table 3.2 provides an evaluation of energy cost to enforce the prescribed active motion on the second and third joint for all cases. Energy cost is calculated through equation (3.4) from $t = 0$ to $t = 25$. Here required torque for active joint number 2 and 3 ($M_2$ and $M_3$) are calculated as Lagrange multiplier of the rigid body system. Energy cost is also consistent

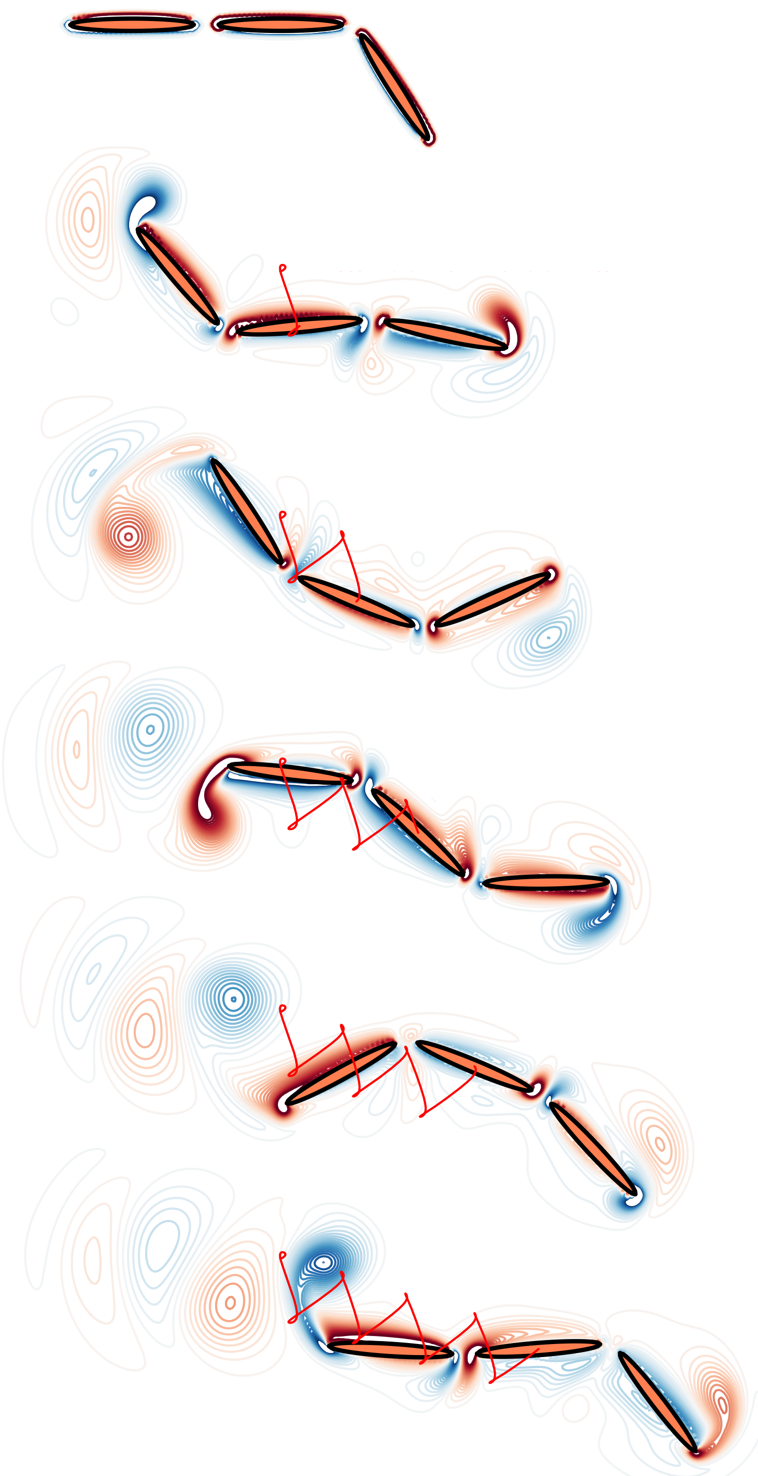


Figure 3.12: Vorticity field for free swimming articulated fish at six instants $t = 0$, $0.8T$, $1.6T$, $2.39T$, $3.18T$, $3.98T$ with history of centroid position from top to bottom, where $T$ is one oscillation period $2\pi$, with $\Delta x = 0.02, \Delta t = 0.01$. Vorticity contour plotted from -5 to 5 in 40 uniform increments.

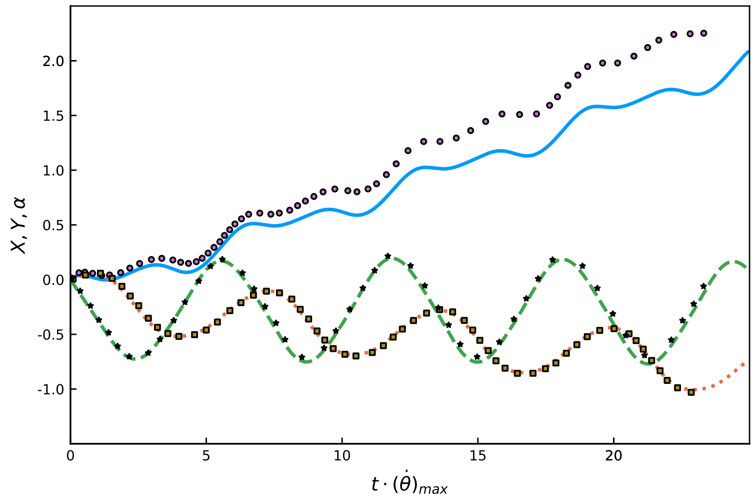
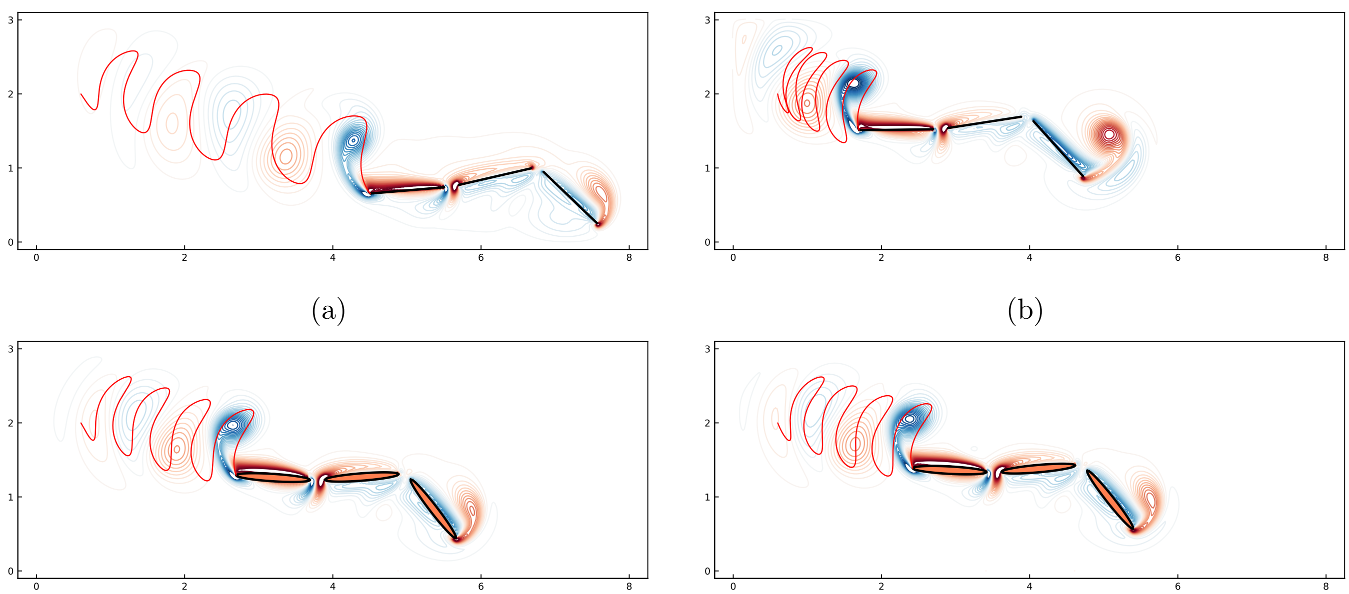



Figure 3.13: Time history of centroid position $X, Y$ and rotation angle $\alpha$ of the central body. This simulation: blue solid for $X$, orange dot for $Y$, green dashed for $\alpha$; Results in Eldredge [12]: circle for $X$, rectangle for $Y$, star for $\alpha$.



(a)

(b)

(c)

(d)

Figure 3.14: Vorticity contour plotted from -5 to 5 in 40 uniform increments at time $t = 3.98T = 25$ for (a) flat plate with mass ratio 0, (b) flat plate with mass ratio 1, (c) ellipse with mass ratio 0, (d) ellipse plate with mass ratio 1

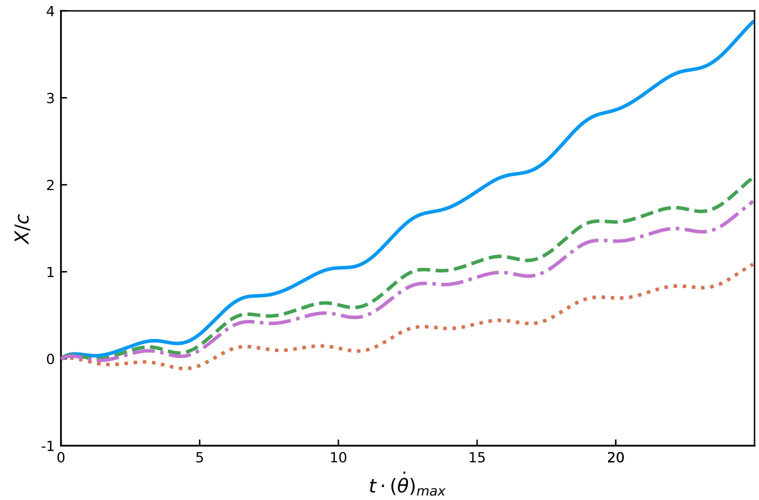
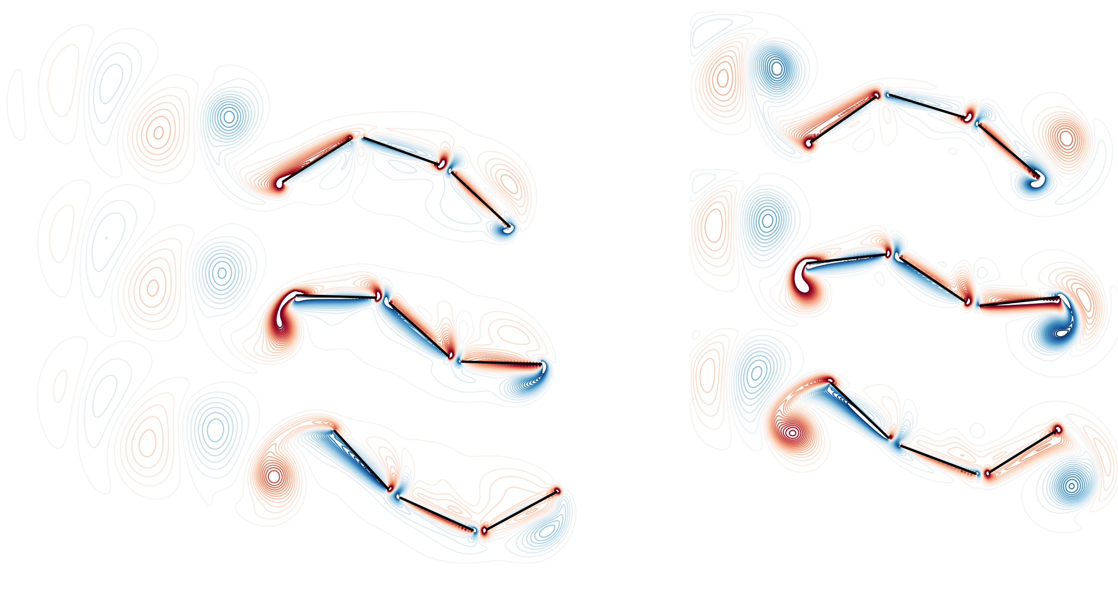



Figure 3.15: Propulsion distance in x-direction for (a) flat plate with mass ratio 0, blue solid; (b) flat plate with mass ratio 1, orange dotted; (c) ellipse with mass ratio 0, green dashed; (d) ellipse plate with mass ratio 1, purple dashdotted



(a) (b)

Figure 3.16: Vorticity contour plotted from -5 to 5 in 40 uniform increments at time $t = 20.0, 21.2, 22.4$ from top to bottom for (a) flat plate with mass ratio 0, (b) flat plate with mass ratio 1

with the conclusion of propulsion effectiveness, with plate $E_a < E_b$ and ellipse $E_c < E_d$.

$$E = \int_T \left( M_2 \dot{\theta}_2 + M_3 \dot{\theta}_3 \right) \, dt \tag{3.4}$$

| | plate $\rho = 0$ $E_a$ | plate $\rho = 1$ $E_b$ | ellipse $\rho = 0$ $E_c$ | ellipse $\rho = 1$ $E_d$ |
|---|---|---|---|---|
| $E$ | 0.7689 | 1.7516 | 0.7104 | 0.7948 |

Table 3.2: Energy cost to enforce the prescribed active motion for articulated fish, compared between different body shapes and density ratios

## 3.5 Passive flapping of an articulated flag

In section 3.4, we enforce prescribed motion for two revolute joints and solve for a planar joint. Here we introduce a case of multiple bodies connected by passive revolute joints to mimic the dynamics of a flapping flag in viscous flow. Flapping flag problems has been widely studied with continuous elastic material governed by Euler-Bernoulli beam equation [82, 67]. For a rigid body chain with number of bodies approaching infinite, it is shown that the dynamical equation converges to the continuous equation, as proved in Wang and Eldredge [13]. We apply similar set up to model the rigid body system, which has $n$ identical infinitely thin plates of total length $L$ connected by revolute joints with stiffness $k$ and zero damping coefficient (schematic plot in figure 3.17). The first body is pinned in space at $O$, only allowing for rotational motion. The system is in a uniform flow with velocity $U$ and under gravity $g$, both in $-y$ direction. In this numerical simulation, $n = 10$ and $L = 1$ so each individual plate has length $l = 0.1$. For each revolute joint, torsion stiffness $k$ should be set to $k = n\eta/L$ to reconcile with a flag of uniform bending stiffness $\eta$. So non-dimensional parameters governing the dynamics are Reynolds number $Re$, mass ratio $m$, non-dimensional torsion stiffness $k$ and Froude number $Fr$. They are given respectively as

$$Re = \frac{UL}{\nu} = 200, \quad m = \frac{\rho_b}{\rho_f L} = 1.5, \quad k = \frac{n\eta}{\rho_f U^2 L^4} = 0.01, \quad Fr = \frac{U}{\sqrt{gL}} = 1.414 \tag{3.5}$$

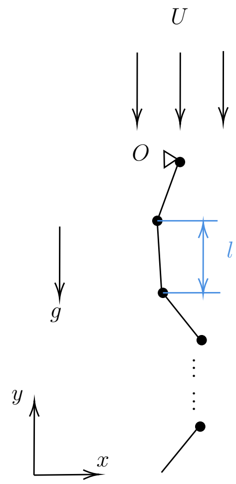
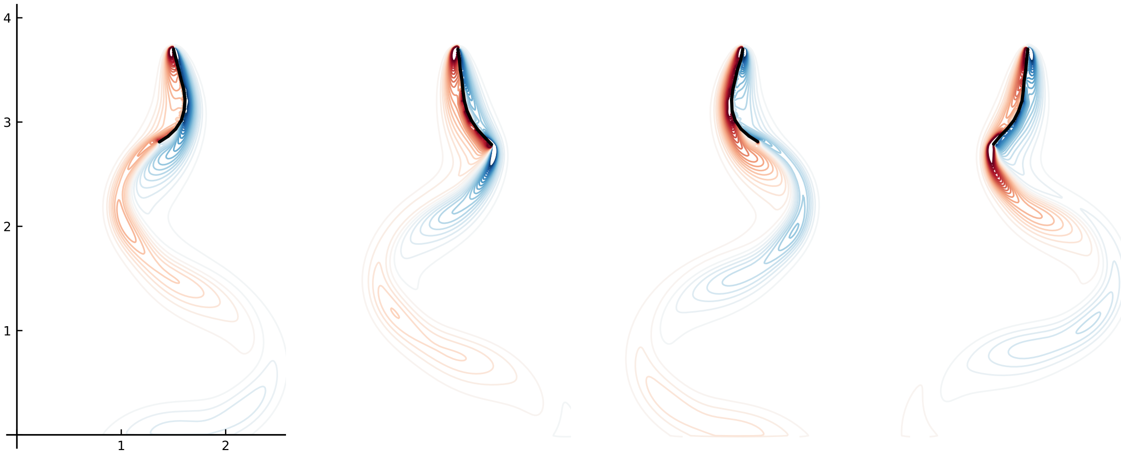



Figure 3.17: Schematic illustration of the articulated flag

Simulation is done on a domain of $[0, 3] \times [0, 4]$ with grid space $\Delta x = 0.01$ and time step $\Delta t = 0.005$ for 30 time units. Vorticity contour at four time units $t = 9.2$, 10.0, 10.8, 11.6 are plotted in figure 3.18. Vorticity contour plot is very similar to results in Huang et al. [82]. History of tail position in x direction is plotted in figure with comparison to results in Wang and Eldredge [13]. Difference is accumulating as number of oscillation period rises, but overall good consistency is achieved. The previous study applied a strongly-coupled fluid–structure interaction solver, which requires iteration between fluid and body interface. This may result in slightly accumulating error compared to this study. Here we introduce this simulation case only to show what the solver is capable of, without too much in depth discussion on the physical behaviour.

## 3.6 Dynamics of a two-linked body system

In the previous section, we demonstrated the capability of using rigid body systems to approximate a continuously deforming structure. Here we focus on a two-linked rigid body system, which inherits many characteristic from other types of vortex induced vibrations (airfoil, flag, elastically-mounted cylinder etc.). The problem set up is similar to the articulated flag, but with bodies and uniform flow placed on horizontal direction and gravity ignored (schematic plot in figure 3.20). We also allow for non-zero damping coefficient in the




(a) (b) (c) (d)

Figure 3.18: Vorticity field for articulated flag at four instants (a) $t = 9.2$, (b) $t = 10.0$, (c) $t = 10.8$, (d) $t = 11.6$ with $\Delta x = 0.01, \Delta t = 0.005$. Vorticity contour plotted from -15 to 15 in 40 uniform increments.

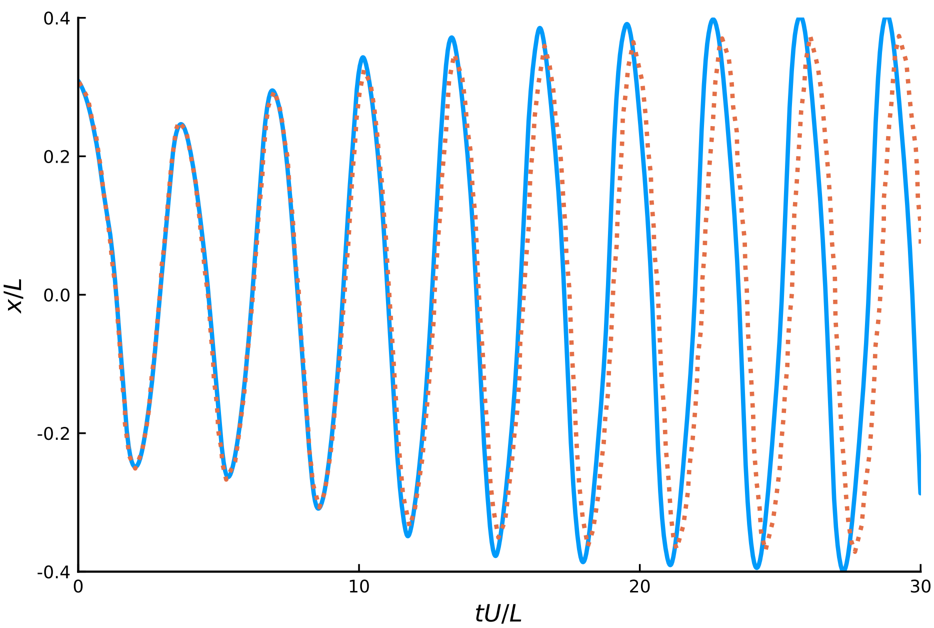



Figure 3.19: Tail position in x direction for articulated flag. This simulation, blue solid; Wang and Eldredge [13], orange dot.

joint to investigate energy harvesting. It is believed that the simplicity of two-linked rigid body system will shrink the number of control variables and computation resources needed while keeping the main physical features revealed. For simplicity, we consider the two rigid bodies to be the same infinitely thin flat plate, connected by revolute joints. The first body is pinned in space at the location of the first joint $O$ with a revolute joint. There's no gap between the two bodies. Here the non-dimensional parameters for the system are Reynolds number $Re$, mass ratio $m$, non-dimensional torsion stiffness $k$, non-dimensional damping coefficient $c$. They are given respectively as

$$Re = \frac{UL}{\nu}, \quad m = \frac{\rho_b}{\rho_f L}, \quad k = \frac{k'}{\rho_f U^2 L^2}, \quad c = \frac{c'}{\rho_f U^2 L^3} \tag{3.6}$$

where each of the infinitely thin body has length $L/2$ , $U$ is uniform flow velocity, $\nu$ is fluid kinematic viscosity, $k'$ is dimensional torsion stiffness to a two-dimensional torque, $c'$ is dimensional damping coefficient to a two-dimensional torque. From this point on, stiffness refers to $k$ and damping refers to $c$ for short.

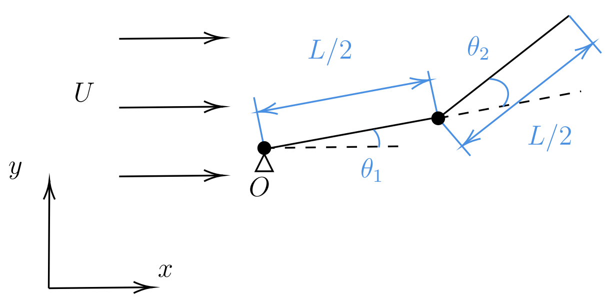



Figure 3.20: Schematic sketch of the two-linked rigid body system

Vortex induced vibrations can generally be divided into free vibration and forced vibration. In free vibration case, fluid–structure coupling is more involved, and often physical features are richer. Characteristics like hysteresis mostly occurs with free vibration. Energy transfer between fluid an structure happens in a more compact way. On the other hand, forced vibration can manipulate parameters in a wider range, which means more types pf vortex wake shed can be observed for example. Common features of vortex induced vibrations include bi-stability in Connell and Yue [10], hysteresis in Eloy et al. [9], lock-in in

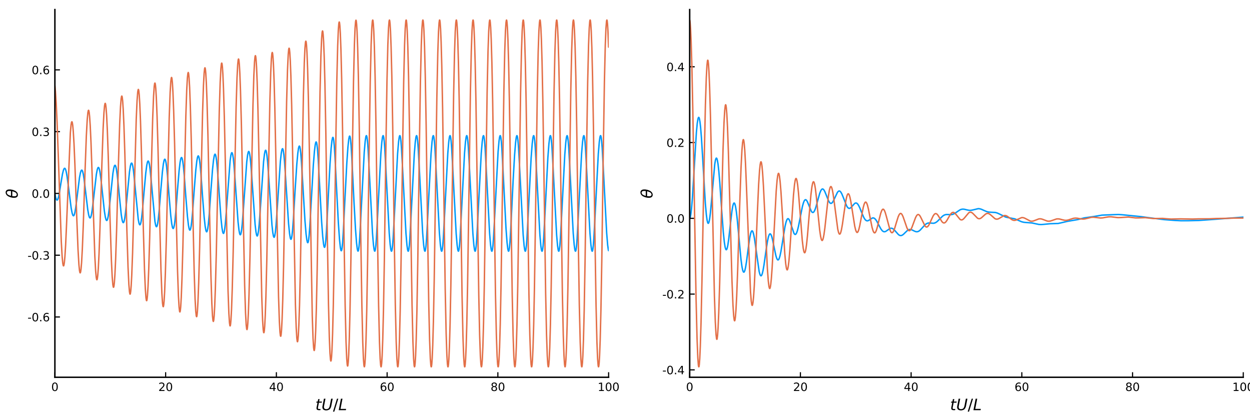
Williamson and Govardhan [5] etc. In this study, we focus our attention on the free vibration by leaving the body motion passive. In previous work on bluff bodies, airfoils and flapping flags, it is common to vary flow velocity or its combination with mass ratio (reduced velocity) to explore the dynamical response. Torsion stiffness is also varied to explore its influence as in Carmo et al. [6]. However, damping coefficient has rarely been considered to have possible influence on stability state of the system. In this section, we explore the role of damping coefficient in the dynamical response besides its role in energy harvesting.

### 3.6.1 Effect of non-zero uniform flow

A simple comparison of zero and non-zero uniform flow coupled with the same two-linked body system is shown in the first place to demonstrate the effect of uniform fluid. By choosing the same parameters $Re = 200$, $m = 1$, $k = 0.05$, $c = 0$ and the same initial condition $\theta_1 = 0$, $\dot{\theta}_1 = 0$, $\theta_2 = \pi/6$, $\dot{\theta}_2 = 0$, figure 3.21 shows two cases, one with non-zero uniform flow velocity $U = 1$ and the other with zero flow velocity. Note that the same initial condition applies to all the cases in this section if not specifically mentioned. Without uniform flow, initial spring potential energy is damped through dashpots and finally reach a straight configuration. In contrast, the periodic oscillation of fluid interacting with body system revealed the underlying fact that energy transfer from uniform flow to body system through an oscillating cycle is positive. Also we observe small wiggles of body oscillation in the no flow case, indicating a second frequency. It is shown that uniform flow has a smoothing effect on the dynamical response that washed out this second frequency.

### 3.6.2 Stability boundary for oscillation

The rigid body system in this section can actually be viewed as a flag which has only two segments. Similar to the flapping flag in uniform flow, the rigid body system has two stability states as well, i.e. fixed-point stability (stationary or straight) and limit-cycle stability (self-sustained oscillation). Unlike the flapping flag which also has an "approaching to chaos" state, the two-linked body system can never go to chaos. An example of the two different




(a) (b)

Figure 3.21: Comparison on oscillation angle of (a) Re=200 case and (b) no flow velocity case plotted with $\theta_1$ in blue for body 1, $\theta_2$ in orange for body 2.

stability states are shown in figure 3.22 with vorticity field and phase space of of the second joint $(\theta_2, \dot{\theta}_2)$. Common governing parameters for both cases are $Re = 200$, $k = 0.05$ and $c = 0$. The difference is in mass ratio, where $m = 1$ leads to self-sustained oscillation and $m = 0.5$ leads to stationary state. In the large parametric space, we would like to identify the boundary between stationary state and self-sustained oscillation state, as a function of $Re$, $m$, $k$ and $c$. Figure 3.23 shows the critical mass ratio for oscillation as a function of sprint stiffness $k$ in the range of $[0, 0.1]$ for three Reynolds number $Re = 200$, 600 and 1000, with damping coefficient $c$ fixed at 0. The critical mass according to each spring stiffness is found by several numerical simulations to identify the boundary between stationary state and oscillating motion. For each curve, self-sustained oscillation is achieved above the line (region I), while stationary state is blow the line (region II). Generally speaking, the relation between $k$ and $m$ for each fixed $Re$ is nearly exponential. The resistance brought by larger stiffness would need larger body inertia to overcome for oscillation stability. At higher Reynolds number, the impact of fluid inertial effect also requires larger mass ratio to surmount. The trend in figure 3.23 is similar to the study of critical mass ratio for continuously deforming flag in Connell and Yue [10]. It is also worth mentioning for one feature different from flag study. At $Re = 200$, if the stiffness $k$ is larger than a critical value around 0.082, it's not possible to achieve oscillation state regardless $m$ chosen.

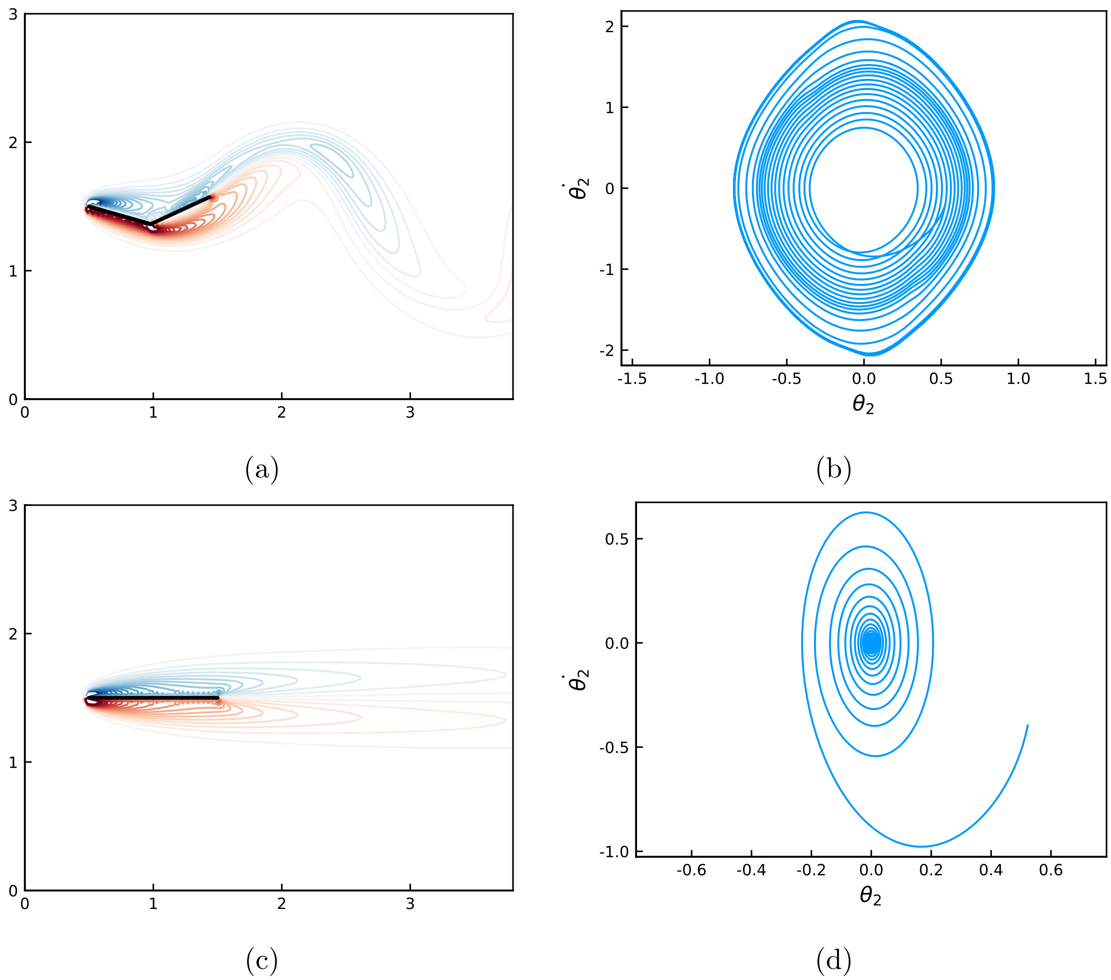



Figure 3.22: Vorticity field at $t = 100$ for (a) $m = 1$ and (b) $m = 0.5$ with contour levels from -15 to 15 with 40 uniform increments; Phase space plot for the second joint $(\theta_2, \dot{\theta}_2)$ for (c) $m = 1$ and (d) $m = 0.5$ in radians.

Figure 3.24 shows the critical mass ratio for oscillation as a function of damping coefficient $c$ in the range of $[0, 0.01]$ at three Reynolds number $Re = 200$, 600 and 1000, with spring stiffness $k$ fixed at 0.05. Similar to figure 3.23, larger mass ratio is required for self-sustained oscillation state with larger damping coefficient. The relation between $c$ and $m$ for each fixed $Re$ is nearly linearly.

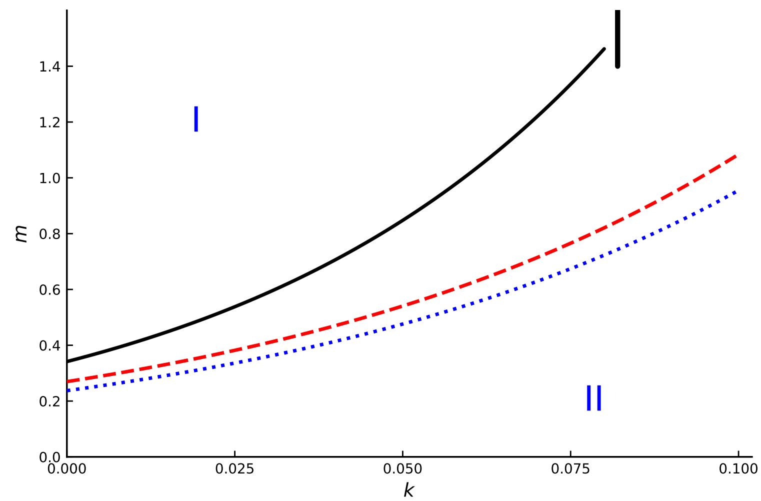



Figure 3.23: Critical mass ratio for self-sustained oscillation with respect to spring torsion stiffness $k$ for $Re = 200$, black solid; $Re = 600$, red dashed; $Re = 1000$, blue dotted. Damping coefficient is fixed at $c = 0$. Notation I for self-sustained oscillation region and II for stationary state region.

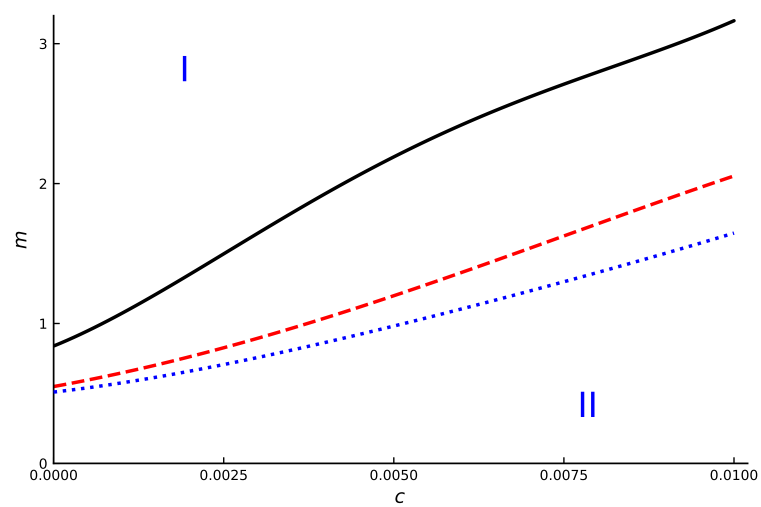



Figure 3.24: Critical mass ratio for self-sustained oscillation with respect to damping coefficient $c$ for $Re = 200$, black solid; $Re = 600$, red dashed; $Re = 1000$, blue dotted. Spring stiffness is fixed at $k = 0.05$. Notation I for self-sustained oscillation region and II for stationary state region

### 3.6.3 Bifurcation, bi-stability and hysteresis

Bifurcation in vortex induced vibration problems refers to the change of stability state between fixed-point stability and limit cycle stability. When the bifurcation is approached by varying certain control parameters from different directions, different behaviours in stability state could occur. If the bifurcation point does not depend on the direction, it is called supercritical bifurcation. If the bifurcation is not consistent when varying control parameters in different directions, it is called subcritical bifurcation. For subcritical bifurcation, we define the hysteresis region to be regions ranged from forward bifurcation point to backward bifurcation point. For flags, there are a lot of debates on whether bifurcation is supercritical or subcritical, and there are discrepancies between experimental results and numerical results. The usual practice is to choose reduced flow velocity $U^*$ as the varying parameter, which is the natural choice and also easy to implement in experiments. For hysteresis study, we choose to convert our non-dimensional parameters into the set used in a study for flag hysteresis by Eloy et al [9] since there are internal connections between flags and the two-linked body system. The comparison of non-dimensional parameters between current study and [9] is presented in table 3.3.

| | Eloy et al. [9] | Current study |
|---|---|---|
| mass ratio $m^*$ | $\dfrac{\rho_f L}{m_b}$ | $\dfrac{1}{m}$ |
| reduced flow velocity $U^*$ | $UL\sqrt{\dfrac{m_b}{D}}$ | $\sqrt{\dfrac{m}{k}}$ |
| fluid viscosity $\nu^*$ | $\dfrac{\nu m_b^{3/2}}{\rho_f^2 D^{1/2}}$ | $\dfrac{m^{3/2}}{k^{1/2}} \cdot \dfrac{1}{Re}$ |
| structure damping $\mu^*$ | $\dfrac{\mu \rho_f^2}{m_b^{5/2} D^{1/2}}$ | $\dfrac{c}{m^{5/2} \cdot k^{1/2}}$ |

Table 3.3: Non-dimensional parameter comparison between current study and [9].

From table 3.3 we can see that $m$ in this current study needs to be fixed to keep mass ratio unchanged. At the same time, $c$ and $k^{1/2}$ need to vary at the same time to keep structural damping in the Euler-Bernoulli equation constant. Also Reynolds number $Re$ in this study needs to be varied together with $k$ and $m$ to keep fluid viscosity constant. The constant parameters are chosen similar to Eloy et al. [9] with $m^* = 0.6$, $\nu^* = 0.07$ and $\mu^* = 0.01$, considering the difference between flag bending stiffness and spring torsion stiffness. Range of reduced velocity $U^*$ is set to $[5, 25]$. Results with figure 3.25 first shows that the bifurcation is supercritical, with increasing $U^*$ (forward pass plotted with blue circle) and decreasing $U^*$ (backward pass plotted with red cross) trigger bifurcation at the same reduced velocity. This feature is different from flags, where forward pass and backward pass result in different bifurcation point (thus subcritical). However the supercritical bifurcation for the two-linked body case still leads to a hysteresis loop, which is formed by different self-sustained oscillation amplitude. We can still call it bi-stability, since the coexistence of two limit-cycle of different amplitude is demonstrated from the hysteresis loop. Meanwhile we have to differ this feature from the bi-stability in flags, which is defined as the coexistence of fixed-point stability and limit-cycle stability. The coexistence of two limit-cycle is illustrated more clear in figure 3.26, where history of the second joint deflection angle $\theta_2$ and phase space plot of the second joint $(\theta_2, \dot{\theta}_2)$ are plotted at the same reduced velocity $U^* = 15.4$, with one in forward pass (blue) and one in backward pass (orange). Both cases reach self-sustained state on their own, however the peak values from backward pass is larger.

### 3.6.4 Energy harvesting analysis

Energy harvested from fluid by joint damper per unit time is calculated following equation (3.7) (also used in Singh and Michelin [11]). For a rigid body system with two identical dashpots

$$P = \frac{1}{T} \int_0^T \left( c\, \dot{\theta}_1^{\,2} + c\, \dot{\theta}_2^{\,2} \right) dt \tag{3.7}$$

, where $T$ is period of oscillation in the self-sustained state, and $\dot{\theta}_1, \dot{\theta}_2$ are angular velocity of the two joints respectively. In this study, we choose to fix the other control parameters at $Re = 600$, $m = 2.0$ and $k = 0.05$, while the range of $c$ limited at $[0.001, 0.009]$. We can see

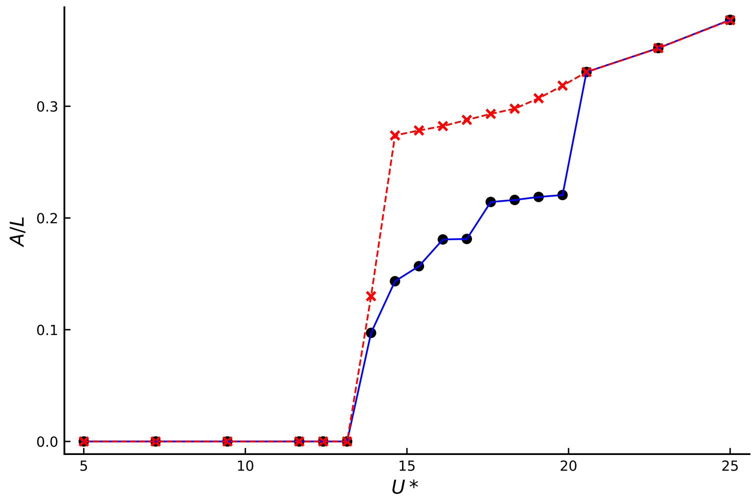



Figure 3.25: Hysteresis loop of tail y-position $A$ for two-linked rigid body system with varying reduced flow velocity $U^*$. Increasing $U^*$ (forward pass), blue circle; decreasing $U^*$ (backward pass), red cross.

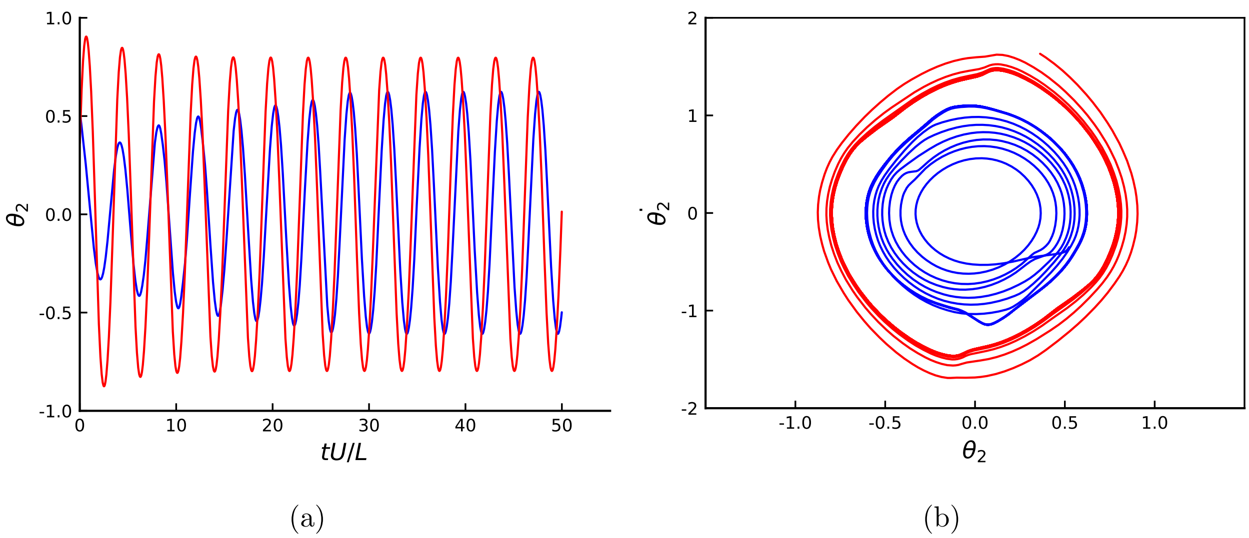



Figure 3.26: (a) History of the second joint's deflection angle $\theta_2$ and (b) phase space plot of $(\theta_2, \dot{\theta}_2)$ at the same reduced velocity $U^* = 15.4$. Forward pass in blue and backward pass in red.

that with the set of parameters chosen, oscillation state is guaranteed to be reached from figure 3.24. Beyond $c = 0.009$, no self-sustained oscillation will be achieved, meaning there will be zero oscillation amplitude, zero period and zero energy harvested. Figure 3.27 depicts peak oscillation amplitude of both joints in (a), oscillation period in (b), energy harvested period unit time in (c), and energy harvested per oscillation period in (d). We can see that oscillation amplitude of both joints decreases as damping effect increases, meantime the second joint has a larger oscillation amplitude than the first one. Both bodies oscillates with a common period with certain phase lag, and the period increases with damping coefficient $c$. Energy harvested per unit time also increases with $c$. From equation (3.7) we can observe that although oscillation amplitude decreases and period increases, the contribution from larger damping coefficient rules. This indicates that as long as it's still in the self-sustained stability range, increasing damping coefficient will result in a better energy harvesting performance.

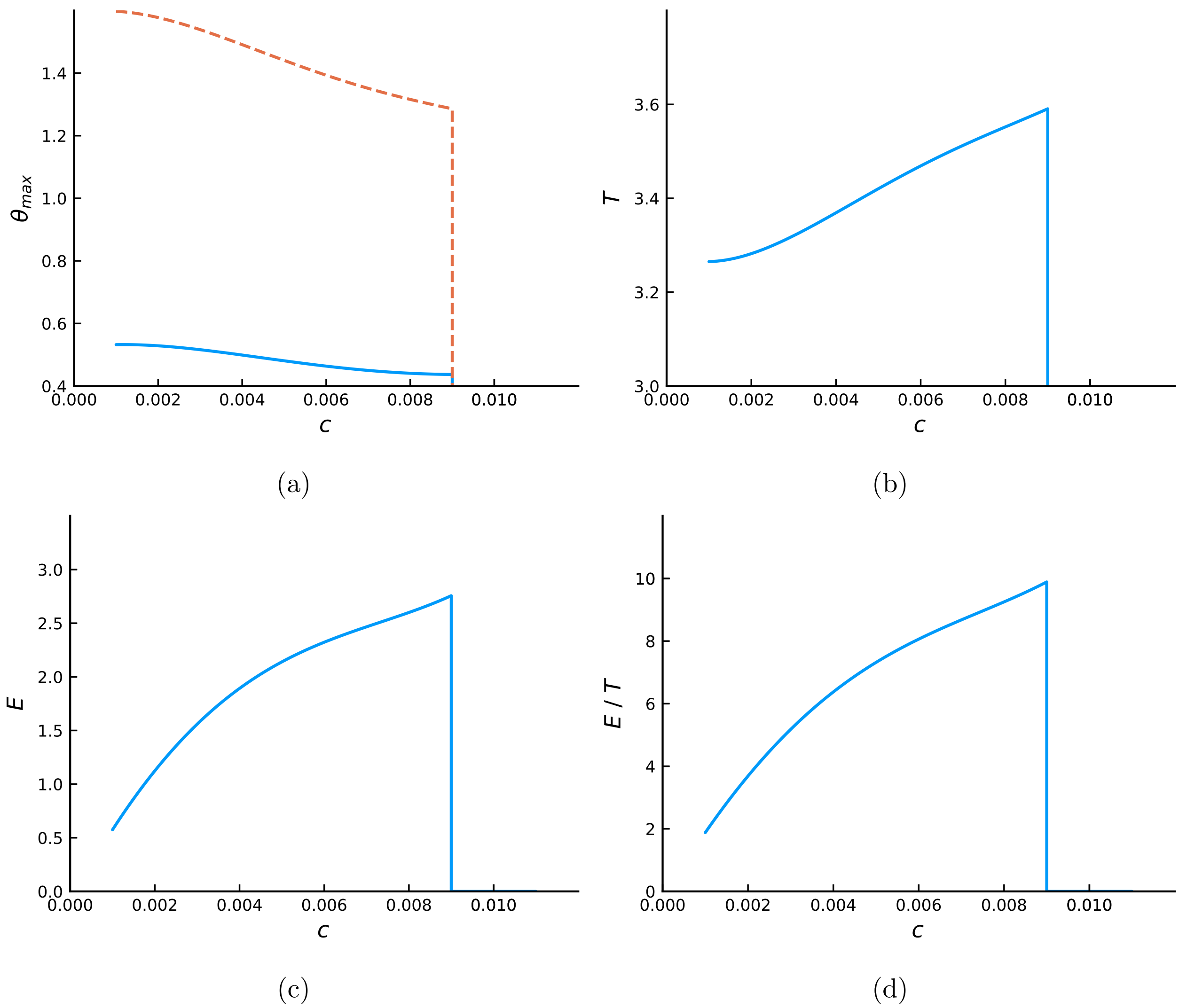



Figure 3.27: (a) Maximum oscillation angles of first joint (blue solid) and second joint (orange dashed) in radians, (b) oscillation period, (c) energy harvested per unit time, (d) energy harvested in one oscillation period.

# CHAPTER 4

# Conclusions and Future Works

## 4.1 Conclusions

In the current study, we conducted a numerical investigation on fluid–structure interaction problems by developing a monolithic coupling algorithm between viscous flow on an unbounded domain and rigid body systems. The fluid system is constructed with immersed boundary projection method in the vorticity form of Navier–Stokes equation on a uniform Cartesian grid. The Lattice Green's function based integrating factor technique developed by Liska and Colonius [93] is used to analytically evaluate the viscous diffusion term on time discrete level. Rigid body system is represented with six-dimensional spatial vector formulation and joint model representation. Unified treatment of both fluid and body constraints are enforced through the use of Lagrange multipliers, including the enforcement of boundary condition, joint constraint and body's prescribed motion (if any). In the monolithic coupling approach, fluid–body interaction force on the immersed surface and boundary conditions are kept implicit and solved simultaneously. Half-explicit Runge–Kutta method is applied for both fluid and rigid body system in time marching, taking the advantage that both systems are differential-algebraic. Different ways to arrange the saddle point system and their influence are discussed with comparison to previous studies. Due to the novel approach of matrix block arrangement, our algorithm is proved to be stable for arbitrary low density ratios including zero mass bodies. By block LU decomposition on the fluid–body saddle point system, an added-mass term for viscous flow shows naturally from linear algebra techniques to augment inertia of the rigid body. In this way, our algorithm justifies the use of added-mass to change body inertia in previous studies by partitioned method. Also no iteration process or parameter tuning is needed in the solving procedure. For two-dimensional body with finite volume, correction is also made by assuming rigid body motion of the fictitious fluid inside body regarding immersed boundary method. Within the same mathematical framework, both active and passive motions of rigid body system are allowed, since joint constraint and active motions are enforced in the same way as Lagrange multipliers.

Multiple applications are constructed for verifications and physical investigations. Among these applications, solver convergence is demonstrated using a falling flat plate under gravity in a quiescent fluid. Stability regarding arbitrary low to zero mass ratios is demonstrated through the falling/rising of a circular cylinder under its own weight. An investigation of vortex wakes from forced oscillation of a symmetric airfoil is illustrated compared to experimental results in other studies. We also show examples of an articulated fish swimming in viscous flow and self-excited oscillation of an articulated flag. Finally bifurcation, hysteresis and energy harvesting behaviours are studied using a two-component linked plates model. The algorithm is shown to be versatile for all aforementioned applications with very little change in the code, as a result of the choice of rigid body systems representation. This is considered as a new feature compared to previous FSI studies. Also the code developed in this study is relatively compact and the requirement for computation resources is kept at a relatively low level.

## 4.2 Future works

There is always space for improvement for any research work, so it the current study. First, all the examples shown in this work is two-dimensional. The monolithic coupling scheme should work exactly the same for two-dimensional examples and three-dimensional examples, however one needs time and effort to make it work for three-dimensional examples. The current rigid body solver is ready for three-dimensional construction, it actually can simulate the dynamics of a two-dimensional rigid body system in a three-dimensional space without fluid. The fluid solver needs to be updated to allow for computation in the three-dimensional space, including using velocity form Navier–Stokes equation instead of the vorticity form (problems from streamfunction in three-dimensional space), stencil change for all differential operators, and change in data types to contain three directions etc. The ensuing problem is obvious, computation load increase by another dimension would be large. The current solver does not incorporate a parallel computing module because the computation load is quite durable for two-dimensional problems. Upgrading the parallel computing part is definitely not trivial and requires effort.

One of the problems in immersed boundary method is that the computed force on the immersed surface is not continuous, as addressed in Dorschner and Colonius [42]. Although a smoother discrete delta function in Yang et al. [97] is applied to smooth the spurious force, the non-physical oscillation still persists, only smoothed to a certain degree. The spurious oscillation results in undesired behaviour in force-related properties, lift coefficient for example. This could also be important in FSI problems since the interaction force is solved implicitly, with influence on other field properties. We did not observe significant influence of the spurious force in the current study, however there should be possibilities for improvement on this issue.

The other concern is in the treatment for fictitious fluid inside a two-dimensional body with finite volume. The current correction takes the fictitious fluid into account, however the assumption is that the fictitious fluid undergoes the same rigid body motion. This is reasonable for a rigid body under translation, however it is not adequate for body with rotation, except for every slim bodies. In the example of articulated fish swimming, we assume that the fictitious fluid has the same rigid body motion with the slim ellipse since two-dimensional fluid motion inside the thin body is mainly constrained. If it's a cylinder instead of ellipse, problems would rise. One example is the experimental study in Horowitz and Williamson [102] regarding transverse motion of a falling/rising cylinder. In this study, if mass ratio of the cylinder is in certain range and cylinder is allowed to move both vertically and horizontally, it would experience large oscillatory transverse motion. The transverse motion is very likely achieved through cylinder rotation. With our current treatment of the fictitious fluid, the solver would go unstable if we do not constrain the transverse degree of freedom with the same set up. We believe that if correct fluid motion can be modelled inside the body, the current algorithm should be able to conduct a study comparable to Horowitz and Williamson [102]. Possible solutions include applying immersed domain method instead of immersed boundary method, or creating a Heavyside mask function to correctly compensate for the jump across the immersed boundary.

There are some studies that can be done with the current solver, but not presented in this work due to the limit of time. First, it could be a good idea to compare stability and accuracy of a strongly-coupled method and a monolithic method, like the study in Blom [68]. It is actually relatively easy to construct another strongly-coupled method similar to Wang and Eldredge [13]. It would be neat to compare side by side on the difference of the coupling schemes. For the study of two plates linked by joints in section 3.6, a larger parametric space can be applied. Also energy harvesting behaviour could be explored with more variations. Finally, there are some types of simulation case that can be constructed, only with a little more effort. For example the case of two flags in parallel position in Huang et al. [82]. The solver at this moment is not able to place multiple bodies not connected in hierarchy, since assumption for hierarchy connection is made in the code implementation. However there is no profound difficulty from extending the code to satisfy a slightly different body configuration. This would be a minor issue should there be more time allowed.

# APPENDIX A

# Appendix

## A.1 The first way in equation (2.62) to form the saddle point system

$$\begin{bmatrix} H_{i,i-1} & \mathbf{0} & B_1^T & \mathbf{0} \\ \mathbf{0} & \mathbb{M} & T_1^T & G_1^T \\ B_2 & -T_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & G_2 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \dot{\boldsymbol{u}}_{i-1} \\ \boldsymbol{f_b}_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix}$$

$$= \begin{bmatrix} \left.\begin{matrix} -N(\boldsymbol{\omega}_{i-1}) \\ F(\boldsymbol{q}_{i-1}, \boldsymbol{u}_{i-1}) \end{matrix}\right\} \quad r_1 \\ \left.\begin{matrix} -\dfrac{1}{\Delta t\ a_{i,i-1}} \{B_2(H_{1,i}\,\boldsymbol{\omega}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j} H_{j+1,i}\,\dot{\mathfrak{U}}_j) - T_2(\boldsymbol{u}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j} \dot{\boldsymbol{u}}_j) + \boldsymbol{u}_\infty\} \\ -\dfrac{1}{\Delta t\ a_{i,i-1}} \{G_2(\boldsymbol{v}_0 + \Delta t \sum_{j=1}^{i-2} a_{i,j} \dot{\boldsymbol{u}}_j) + gt(t_i)\} \end{matrix}\right\} \quad r_2 \end{bmatrix}$$

After solving for the saddle system at every stage, the assembly stage is done by

$$\begin{cases} \boldsymbol{\omega}_i = H_{1,i}\,\boldsymbol{\omega}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} H_{j+1,i}\,\dot{\mathfrak{U}}_j \\ \boldsymbol{u}_i = \boldsymbol{u}_0 + \Delta t \sum_{j=1}^{i-1} a_{i,j} \dot{\boldsymbol{u}}_j \end{cases}$$

### A.1.1 Prediction step

$$\begin{bmatrix} H_{i,i-1} & \mathbf{0} \\ \mathbf{0} & \mathbb{M} \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} \\ \dot{\boldsymbol{u}}^*_{i-1} \end{bmatrix} = \begin{bmatrix} -N(\boldsymbol{\omega}_{i-1}) \\ F(\boldsymbol{q}_{i-1}, \boldsymbol{v}_{i-1}) \end{bmatrix}$$

$$\dot{\mathfrak{U}}^*_{i-1} = H_{i-1,i}\left(-n(\boldsymbol{\omega}_{i-1})\right)$$

$$\mathbb{M}\dot{\boldsymbol{u}}^*_{i-1} = F(\boldsymbol{q}_{i-1}, \boldsymbol{v}_{i-1})$$

### A.1.2 Apply constraints

$$\begin{bmatrix} BL^{-1} & -T \\ \mathbf{0} & G(\boldsymbol{q}_i) \end{bmatrix} \begin{bmatrix} H_{i-1,i} & \mathbf{0} \\ \mathbf{0} & \mathbb{M}^{-1} \end{bmatrix} \begin{bmatrix} B^T & \mathbf{0} \\ T^T & G^T(\boldsymbol{q}_{i-1}) \end{bmatrix} \begin{bmatrix} (\boldsymbol{f_b})_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} = \begin{bmatrix} BL^{-1} & -T \\ \mathbf{0} & G(\boldsymbol{q}_i) \end{bmatrix} \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} \\ \dot{\boldsymbol{u}}^*_{i-1} \end{bmatrix} - \begin{bmatrix} r_{2,1} \\ r_{2,2} \end{bmatrix}$$

Rearrange to get

$$\begin{bmatrix} BL^{-1}H_{i-1,i}B^T - T\mathbb{M}^{-1}T^T & -T\mathbb{M}^{-1}G^T(\boldsymbol{q}_{i-1}) \\ G(\boldsymbol{q}_i)\mathbb{M}^{-1}T^T & G(\boldsymbol{q}_i)\mathbb{M}^{-1}G^T(\boldsymbol{q}_{i-1}) \end{bmatrix} \begin{bmatrix} (\boldsymbol{f_b})_{i-1} \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix} = \begin{bmatrix} -r_{2,1} - T\dot{\boldsymbol{u}}^*_{i-1} + BL^{-1}\dot{\mathfrak{U}}^*_{i-1} \\ -r_{2,2} + G(\boldsymbol{q}_i)\dot{\boldsymbol{u}}^*_{i-1} \end{bmatrix}$$

Note that the dimension of this saddle system will not be large, since it only contained the number of body points plus the number of joint constraint. From the linear system, we can find that the body point force $\boldsymbol{f_b}$ has two roles, both for enforcing fluid incompressiblity and correction on body velocity in body dynamics as external force. The difference of these two effects, together with the joint constraint forces, are responsible for correcting fluid velocity. For the motion on joints (second row of the linear system), joint motion resulted from fluid forces plus joint motion resulted from constraint joint forces should equal to prescribed joint motion.

### A.1.3 Correction step

$$\begin{bmatrix} \dot{\mathfrak{U}}_{i-1} \\ \\ \dot{\boldsymbol{u}}_{i-1} \end{bmatrix} = \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} \\ \\ \dot{\boldsymbol{u}}^*_{i-1} \end{bmatrix} - \begin{bmatrix} H_{i-1,i} & \mathbf{0} \\ \\ \mathbf{0} & \mathbb{M}^{-1} \end{bmatrix} \begin{bmatrix} B^T & \mathbf{0} \\ \\ T^T & G^T(\boldsymbol{q}_{i-1}) \end{bmatrix} \begin{bmatrix} (\boldsymbol{f_b})_{i-1} \\ \\ \boldsymbol{\lambda}_{i-1} \end{bmatrix}$$
$$= \begin{bmatrix} \dot{\mathfrak{U}}^*_{i-1} - H_{i-1,i} B^T (\boldsymbol{f_b})_{i-1} \\ \\ \dot{\boldsymbol{u}}^*_{i-1} - \left( \mathbb{M}^{-1} T^T (\boldsymbol{f_b})_{i-1} + \mathbb{M}^{-1} G^T(\boldsymbol{q}_{i-1}) \boldsymbol{\lambda}_{i-1} \right) \end{bmatrix}$$

## A.2 Code reference

The code used for the current study is open source, written in language Julia. Julia is designed for scientific computing, with object-oriented feature like Python and fast computation speed comparable to Fortran and C++. There are three contributing packages developed to support FSI study in the current research, listed below.

- Rigid body dynamics solver: Dyn3d.jl, https://github.com/ruizhi92/Dyn3d.jl
- Viscous flow solver: ViscousFlow.jl, https://github.com/JuliaIBPM/ViscousFlow.jl
- FSI coupling solver: FSI.jl, https://github.com/ruizhi92/FSI.jl

Package configurations and url address can possibly change in the future, reader should be able to trace this with packages under user ruizhi92, jdeldre or organization JuliaIBPM.

REFERENCES


[1] Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interaction—a review. *Communications in Computational Physics*, 12(2):337–377, 2012.

[2] CHK Williamson and A Roshko. Vortex formation in the wake of an oscillating cylinder. *Journal of fluids and structures*, 2(4):355–381, 1988.

[3] D Brika and A Laneville. Vortex-induced vibrations of a long flexible circular cylinder. *Journal of Fluid Mechanics*, 250:481–481, 1993.

[4] A Khalak and CHK Williamson. Motions, forces and mode transitions in vortex-induced vibrations at low mass-damping. *Journal of fluids and Structures*, 13(7-8):813–851, 1999.

[5] CHK Williamson and R Govardhan. Vortex-induced vibrations. *Annu. Rev. Fluid Mech.*, 36:413–455, 2004.

[6] BS Carmo, SJ Sherwin, PW Bearman, and RHJ Willden. Flow-induced vibration of a circular cylinder subjected to wake interference at low reynolds number. *Journal of Fluids and Structures*, 27(4):503–522, 2011.

[7] Teis Schnipper, Anders Andersen, and Tomas Bohr. Vortex wakes of a flapping foil. *Journal of Fluid Mechanics*, 633:411–423, 2009.

[8] Jun Zhang, Stephen Childress, Albert Libchaber, and Michael Shelley. Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind. *Nature*, 408(6814):835–839, 2000.

[9] Christophe Eloy, Nicolas Kofman, and Lionel Schouveiler. The origin of hysteresis in the flag instability. *Journal of Fluid Mechanics*, 691:583–593, 2012.

[10] Benjamin SH Connell and Dick KP Yue. Flapping dynamics of a flag in a uniform stream. *Journal of fluid mechanics*, 581:33–67, 2007.

[11] Kiran Singh, Sébastien Michelin, and Emmanuel De Langre. Energy harvesting from axial fluid-elastic instabilities of a cylinder. *Journal of Fluids and Structures*, 30:159–172, 2012.

[12] Jeff D Eldredge. Dynamically coupled fluid–body interactions in vorticity-based numerical simulations. *Journal of Computational Physics*, 227(21):9170–9194, 2008.

[13] Chengjie Wang and Jeff D Eldredge. Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method. *Journal of Computational Physics*, 295:87–113, 2015.

[14] Henri J-P Morand and Roger Ohayon. *Fluid structure interaction-Applied numerical methods*. Number BOOK. Wiley, 1995.

[15] AS Tijsseling. Fluid-structure interaction in liquid-filled pipe systems: a review. *Journal of Fluids and Structures*, 10(2):109–146, 1996.

[16] Matthias Heil and Andrew L Hazel. Fluid-structure interaction in internal physiological flows. *Annual review of fluid mechanics*, 43:141–162, 2011.

[17] David N Ku. Blood flow in arteries. *Annual review of fluid mechanics*, 29(1):399–434, 1997.

[18] Charles S Peskin. Flow patterns around heart valves: a numerical method. *Journal of computational physics*, 10(2):252–271, 1972.

[19] MJ Lighthill. Hydromechanics of aquatic animal propulsion. *Annual review of fluid mechanics*, 1(1):413–446, 1969.

[20] Toshiyuki Nakata and Hao Liu. A fluid–structure interaction model of insect flight with flexible wings. *Journal of Computational Physics*, 231(4):1822–1847, 2012.

[21] Ming-Chen Hsu and Yuri Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50(6):821–833, 2012.

[22] Wing Kam Liu, Yaling Liu, David Farrell, Lucy Zhang, X Sheldon Wang, Yoshio Fukui, Neelesh Patankar, Yongjie Zhang, Chandrajit Bajaj, Junghoon Lee, et al. Immersed finite element method and its applications to biological systems. *Computer methods in applied mechanics and engineering*, 195(13-16):1722–1749, 2006.

[23] Peter J Mucha, Shang-You Tee, David A Weitz, Boris I Shraiman, and Michael P Brenner. A model for velocity fluctuations in sedimentation. *Journal of fluid mechanics*, 501:71–104, 2004.

[24] Charles S Peskin. Numerical analysis of blood flow in the heart. *Journal of computational physics*, 25(3):220–252, 1977.

[25] J Mohd-Yusof. Combined immersed boundaries/b-splines methods for simulations of flows in complex geometries. ctr annual research briefs, nasa ames, 1997.

[26] Long Lee and Randall J LeVeque. An immersed interface method for incompressible navier–stokes equations. *SIAM Journal on Scientific Computing*, 25(3):832–856, 2003.

[27] Kunihiko Taira and Tim Colonius. The immersed boundary method: a projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.

[28] Earl H Dowell and Kenneth C Hall. Modeling of fluid-structure interaction. *Annual review of fluid mechanics*, 33(1):445–490, 2001.

[29] Jeff D Eldredge. *Mathematical Modeling of Unsteady Inviscid Flows*. Springer, 2019.

[30] Kunihiko Taira, Steven L Brunton, Scott TM Dawson, Clarence W Rowley, Tim Colonius, Beverley J McKeon, Oliver T Schmidt, Stanislav Gordeyev, Vassilios Theofilis, and Lawrence S Ukeiley. Modal analysis of fluid flows: An overview. *Aiaa Journal*, pages 4013–4041, 2017.

[31] Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

[32] S Brown and S Brown. Displacement extrapolations for cfd+ csm aeroelastic analysis. In *38th Structures, Structural Dynamics, and Materials Conference*, page 1090, 1997.

[33] Ryoichi Onishi, Toshiya Kimura, Zhihong Guo, and Toshiyuki Iwamiya. Coupled aero-structural model-approach and application to high aspect-ratio wing-box structures. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 4837, 1998.

[34] Juan Raul Cebral and Rainald Lohner. Conservative load projection and tracking for fluid-structure problems. *AIAA journal*, 35(4):687–692, 1997.

[35] Gene Hou and Arunkumar Satyanarayana. Analytical sensitivity analysis of a static aeroelastic wing. In *8th Symposium on Multidisciplinary Analysis and Optimization*, page 4824, 2000.

[36] Jean Donea, Antonio Huerta, Jean-Philippe Ponthot, and Antonio Rodríguez-Ferran. Arbitrary l agrangian–e ulerian methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–23, 2017.

[37] Josep Sarrate, Antonio Huerta, and Jean Donea. Arbitrary lagrangian–eulerian formulation for fluid–rigid body interaction. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3171–3188, 2001.

[38] Wing Kam Liu and HG Chang. Efficient computational procedures for long-time duration fluid-structure interaction problems. 1984.

[39] Wagdi G Habashi, Julien Dompierre, Yves Bourgault, Djaffar Ait-Ali-Yahia, Michel Fortin, and Marie-Gabrielle Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent cfd. part i: general principles. *International Journal for Numerical Methods in Fluids*, 32(6):725–744, 2000.

[40] Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The discrete geometric conservation law and the nonlinear stability of ale schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.

[41] JC Newman Iii, PA Newman, AC Taylor Iii, and GJ-W Hou. Efficient nonlinear static aeroelastic wing analysis. *Computers & fluids*, 28(4-5):615–628, 1999.

[42] Benedikt Dorschner and Tim Colonius. Immersed boundary projection methods. In *Immersed Boundary Method*, pages 3–43. Springer, 2020.

[43] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

[44] Richard P Beyer and Randall J LeVeque. Analysis of a one-dimensional model for the immersed boundary method. *SIAM Journal on Numerical Analysis*, 29(2):332–364, 1992.

[45] Ming-Chih Lai and Charles S Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of computational Physics*, 160(2):705–719, 2000.

[46] John M Stockie and Brian R Wetton. Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. *Journal of Computational Physics*, 154(1):41–64, 1999.

[47] D Goldstein, R Handler, and L Sirovich. Modeling a no-slip flow boundary with an external force field. *Journal of computational physics*, 105(2):354–366, 1993.

[48] Gianluca Iaccarino and Roberto Verzicco. Immersed boundary technique for turbulent flow simulations. *Appl. Mech. Rev.*, 56(3):331–347, 2003.

[49] Roberto Verzicco, Jamaludin Mohd-Yusof, Paolo Orlandi, and Daniel Haworth. Large eddy simulation in complex geometric configurations using boundary body forces. *AIAA journal*, 38(3):427–433, 2000.

[50] Sekhar Majumdar, Gianluca Iaccarino, and Paul Durbin. Rans solvers with adaptive structured boundary non-conforming grids. *Annual Research Briefs*, 1, 2001.

[51] Joel H Ferziger, Milovan Perić, and Robert L Street. *Computational methods for fluid dynamics*, volume 3. Springer, 2002.

[52] Tim Colonius and Kunihiko Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2131–2146, 2008.

[53] Roland Glowinski, Tsorng-Whay Pan, and Jacques Periaux. Distributed lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Computer methods in applied mechanics and engineering*, 151(1-2):181–194, 1998.

[54] Roland Glowinski, T-W Pan, Todd I Hesla, and Daniel D Joseph. A distributed lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999.

[55] David B Stein, Robert D Guy, and Becca Thomases. Immersed boundary smooth extension (ibse): a high-order method for solving incompressible flows in arbitrary smooth domains. *Journal of Computational Physics*, 335:155–178, 2017.

[56] Charbel Farhat, Kristoffer G Van der Zee, and Philippe Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer methods in applied mechanics and engineering*, 195(17-18):1973–2001, 2006.

[57] Carlos A Felippa, Kwang-Chun Park, and Charbel Farhat. Partitioned analysis of coupled mechanical systems. *Computer methods in applied mechanics and engineering*, 190(24-25):3247–3270, 2001.

[58] Paola Causin, Jean-Frédéric Gerbeau, and Fabio Nobile. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer methods in applied mechanics and engineering*, 194(42-44):4506–4527, 2005.

[59] Christiane Förster, Wolfgang A Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer methods in applied mechanics and engineering*, 196(7):1278–1293, 2007.

[60] Joris Degroote. Partitioned simulation of fluid-structure interaction. *Archives of computational methods in engineering*, 20(3):185–238, 2013.

[61] Matthias Heil. An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1-2):1–23, 2004.

[62] Fang-Bao Tian, Hu Dai, Haoxiang Luo, James F Doyle, and Bernard Rousseau. Fluid–structure interaction involving large deformations: 3d simulations and applications to biological systems. *Journal of computational physics*, 258:451–469, 2014.

[63] Hermann G Matthies and Jan Steindorf. Partitioned but strongly coupled iteration schemes for nonlinear fluid–structure interaction. *Computers & structures*, 80(27-30):1991–1999, 2002.

[64] J Yang, S Preidikman, and E Balaras. A strongly coupled, embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies. *Journal of Fluids and Structures*, 24(2):167–182, 2008.

[65] Charbel Habchi, Serge Russeil, Daniel Bougeard, Jean-Luc Harion, Thierry Lemenand, Akram Ghanem, Dominique Della Valle, and Hassan Peerhossaini. Partitioned solver for strongly coupled fluid–structure interaction. *Computers & Fluids*, 71:306–319, 2013.

[66] Jan Vierendeels, Kris Dumont, and PR Verdonck. A partitioned strongly coupled fluid-structure interaction method to model heart valve dynamics. *Journal of Computational and Applied Mathematics*, 215(2):602–609, 2008.

[67] Andres Goza and Tim Colonius. A strongly-coupled immersed-boundary formulation for thin elastic structures. *Journal of Computational Physics*, 336:401–411, 2017.

[68] Frederic J Blom. A monolithical fluid-structure interaction algorithm applied to the piston problem. *Computer methods in applied mechanics and engineering*, 167(3-4):369–391, 1998.

[69] Björn Hübner, Elmar Walhorn, and Dieter Dinkler. A monolithic approach to fluid–structure interaction using space–time finite elements. *Computer methods in applied mechanics and engineering*, 193(23-26):2087–2104, 2004.

[70] PB Ryzhakov, Riccardo Rossi, SR Idelsohn, and Eugenio Oñate. A monolithic lagrangian approach for fluid–structure interaction problems. *Computational mechanics*, 46(6):883–899, 2010.

[71] Peter W Bearman. Vortex shedding from oscillating bluff bodies. *Annual review of fluid mechanics*, 16(1):195–222, 1984.

[72] R King and DJ Johns. Wake interaction experiments with two flexible circular cylinders in flowing water. *Journal of Sound and Vibration*, 45(2):259–283, 1976.

[73] P Sváček, M Feistauer, and J Horáček. Numerical simulation of flow induced airfoil vibrations with large amplitudes. *Journal of fluids and structures*, 23(3):391–411, 2007.

[74] John Young and Joseph C S. Lai. Oscillation frequency and amplitude effects on the wake of a plunging airfoil. *AIAA journal*, 42(10):2042–2052, 2004.

[75] Serhiy Yarusevych and Michael S H. Boutilier. Vortex shedding of an airfoil at low reynolds numbers. *AIAA journal*, 49(10):2221–2227, 2011.

[76] DJJ Farnell, T David, and DC Barton. Numerical simulations of a filament in a flowing soap film. *International Journal for Numerical Methods in Fluids*, 44(3):313–330, 2004.

[77] Luoding Zhu and Charles S Peskin. Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. *Journal of Computational Physics*, 179(2):452–468, 2002.

[78] Michael Shelley, Nicolas Vandenberghe, and Jun Zhang. Heavy flags undergo spontaneous oscillations in flowing water. *Physical review letters*, 94(9):094302, 2005.

[79] Sankha Banerjee, Benjamin SH Connell, and Dick KP Yue. Three-dimensional effects on flag flapping dynamics. *Journal of Fluid Mechanics*, 783:103–136, 2015.

[80] Christophe Eloy, Claire Souilliez, and Lionel Schouveiler. Flutter of a rectangular plate. *Journal of Fluids and Structures*, 23(6):904–919, 2007.

[81] Luoding Zhu and Charles S Peskin. Interaction of two flapping filaments in a flowing soap film. *Physics of fluids*, 15(7):1954–1960, 2003.

[82] Wei-Xi Huang, Soo Jai Shin, and Hyung Jin Sung. Simulation of flexible filaments in a uniform flow by the immersed boundary method. *Journal of Computational Physics*, 226(2):2206–2228, 2007.

[83] DJJ Farnell, T David, and DC Barton. Coupled states of flapping flags. *Journal of fluids and structures*, 19(1):29–36, 2004.

[84] Emad Uddin, Wei-Xi Huang, and Hyung Jin Sung. Actively flapping tandem flexible flags in a viscous flow. *Journal of Fluid Mechanics*, 780:120–142, 2015.

[85] Jaeha Ryu, Sung Goon Park, Boyoung Kim, and Hyung Jin Sung. Flapping dynamics of an inverted flag in a uniform flow. *Journal of Fluids and Structures*, 57:159–169, 2015.

[86] Y Yadykin, V Tenetov, and D Levin. The flow-induced vibration of a flexible strip hanging vertically in a parallel flow part 1: Temporal aeroelastic instability. *Journal of Fluids and Structures*, 15(8):1167–1185, 2001.

[87] Sebastien Michelin, Stefan G Llewellyn Smith, and Beverley J Glover. Vortex shedding model of a flapping flag. *Journal of Fluid Mechanics*, 617:1–10, 2008.

[88] Silas Alben and Michael J Shelley. Flapping states of a flag in an inviscid fluid: bistability and the transition to chaos. *Physical review letters*, 100(7):074301, 2008.

[89] Christophe Eloy, Romain Lagrange, Claire Souilliez, and Lionel Schouveiler. Aeroelastic instability of cantilevered flexible plates in uniform flow. *Journal of Fluid Mechanics*, 611:97–106, 2008.

[90] JJ Allen and AJ Smits. Energy harvesting eel. *Journal of fluids and structures*, 15(3-4):629–640, 2001.

[91] J Blair Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108(1):51–58, 1993.

[92] Uğis Lācis, Kunihiko Taira, and Shervin Bagheri. A stable fluid–structure-interaction solver for low-density rigid bodies using the immersed boundary projection method. *Journal of Computational Physics*, 305:300–318, 2016.

[93] Sebastian Liska and Tim Colonius. A fast lattice green's function method for solving viscous incompressible flows on unbounded domains. *Journal of Computational Physics*, 316:360–384, 2016.

[94] Sebastian Liska and Tim Colonius. A fast immersed boundary method for external incompressible viscous flows using lattice green's functions. *Journal of Computational Physics*, 331:257–279, 2017.

[95] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[96] Valérie Brasey and Ernst Hairer. Half-explicit runge–kutta methods for differential-algebraic systems of index 2. *SIAM Journal on Numerical Analysis*, 30(2):538–552, 1993.

[97] Xiaolei Yang, Xing Zhang, Zhilin Li, and Guo-Wei He. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, 228(20):7821–7836, 2009.

[98] Alexandre M Roma, Charles S Peskin, and Marsha J Berger. An adaptive version of the immersed boundary method. *Journal of computational physics*, 153(2):509–534, 1999.

[99] Sebastian Liska and Tim Colonius. A parallel fast multipole method for elliptic difference equations. *Journal of Computational Physics*, 278:76–91, 2014.

[100] Eva Kanso, Jerrold E Marsden, Clarence W Rowley, and Juan B Melli-Huber. Locomotion of articulated bodies in a perfect fluid. *Journal of Nonlinear Science*, 15(4):255–289, 2005.

[101] Jeff D Eldredge and Anya R Jones. Leading-edge vortices: mechanics and modeling. *Annual Review of Fluid Mechanics*, 51:75–104, 2019.

[102] M Horowitz and CHK Williamson. Dynamics of a rising and falling cylinder. *Journal of fluids and structures*, 22(6-7):837–843, 2006.