Numerical Investigation of Solitary Flexible Cylinders in Axial Flow

MT54035 MSc Thesis

Zhihong Zeng

Delft University of Technology

TU Delft

Numerical Investigation of Solitary Flexible Cylinders in Axial Flow

by

Zhihong Zeng

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday May 22, 2025 at 09:30 AM.

Student number:	5800366	
Project duration:	February 28, 2024 – May 22, 2025	
Thesis committee:	Prof. dr. G. D. Weymouth,	TU Delft, Supervisor, Chair
	Dr. A. Grammatikopoulos,	TU Delft
	Dr. M. Lauber,	TU Delft

Cover:	Bubbles from deep sea gas reservoirs (modified)[11].
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

The instability of a solitary flexible cylinder in three-dimensional axial flow with various boundary conditions, lengths, and Reynolds numbers has been investigated by CFD (computational fluid dynamics) simulation. A new CFD code is modified from an existing code for a two-dimensional FSI (fluid-structure interaction) problem. The fluid field is solved by the BDIM (boundary data immersion boundary method). The solid structure's displacement is solved by the IGA (isogeometric analysis). The coupling effect is solved by the partitioned and implicit method combined with the IQN (interface Quasi-Newton) method. The simulated results of a clamped-free cylinder can match the experimental results quantitatively. The investigation for cylinders with different parameters indicates that the instability transition as cylinder length increases is similar to that of a clamped-free cylinder as flow velocity increases, small vibrations around the non-zero neutral position are observed when cylinder is in divergence, the deflection amplitude during the transition range reduces, an extra blunt downstream end can reduce the flutter amplitude, and increasing the fluid viscosity in the viscous force exemption range has no stabilizing effect.

Preface

First, I want to express my appreciation to Prof. G. D. Weymouth. He gave me the opportunity to do this challenging thesis project and directed me to do the scientific research. I want to thank Dr. A. Grammatikopoulos for attending as my committee member. Next, I want to express my gratitude to Dr. M. Lauber. He developed the original two-dimensional flow-structure interaction (FSI) code and taught me much about methodology and coding techniques. I also want to express my appreciation to the company Optics11. They addressed the fundamental problem and provided the introduction. They were kind and patient while waiting for me to finish the whole project.

I want to thank all the people who supported me continuously: my parents, my friends, and, of course, TUDelft. I also appreciate you, the people who are reading this thesis report. Thank you for spending time reading my work.

This project is a long journey. Initially, I requested a challenging master's thesis project to prepare for a possible PhD program. The difficulty was actually beyond my expectations. For example, I knew nothing about coding before this project. It took me a week to install Julia and fix the malfunction. Thankfully, I had the fundamental knowledge of hydrodynamics. Of course, my understanding was insufficient initially, but at least I could learned from literature and be taught by others.

During the project, I encountered a lot of problems. Prof. G. D. Weymouth and Dr. M. Lauber taught me to solve issues patiently. The company never urged me to speed up. And my parents kept funding me. I was appreciated because I could focus on learning and investigating without worry. I dived deep into the research and felt like an actual researcher.

I never gave up. I made the code work in the end and validated it successfully. Due to my ability limit, the investigation results were only qualitative. It was hard to tell if this thesis project proved that I could be a PhD student. However, it was a leap from nothing to something for my person. I learned much about each step of solving an FSI problem numerically, including coding.

I realized TUDelft always asks its students to go beyond their limits. I used to think the project was impossible to complete, but I finally made it. When I look back, I feel the work was not that difficult. If someone has coding experience, this project could be easy. As for me, I have put all my effort into it.

However, you may ask if this project is so simple that everyone can do it? I cannot agree with that.

Zhihong Zeng
Delft, May 2025

Contents

Abstract i

Preface ii

Nomenclature v

Figure list viii

Table list xii

1 Introduction 1
- 1.1 Research background . . . 1
- 1.2 Research question and report structure . . . 3

2 Literature review and research matrix 5
- 2.1 Model simplification . . . 5
- 2.2 Literature review . . . 7
  - 2.2.1 Analytical research . . . 8
  - 2.2.2 Experiment . . . 12
  - 2.2.3 Numerical simulation . . . 16
- 2.3 Numerical tools . . . 17
  - 2.3.1 B-spline Curve . . . 17
  - 2.3.2 Finite element analysis . . . 18
  - 2.3.3 Isogeometric analysis . . . 18
  - 2.3.4 Gauss quadrature . . . 21
  - 2.3.5 Finite volume method . . . 22
  - 2.3.6 Boundary Data Immersion Method . . . 22
  - 2.3.7 FSI method . . . 24
- 2.4 Research matrix . . . 26
- 2.5 Research approach determination . . . 26
- 2.6 Contribution . . . 27

3 Solid structure solver 28
- 3.1 Boundary conditions . . . 28
- 3.2 Beam model . . . 28
- 3.3 Methodology . . . 30
  - 3.3.1 B-spline curve . . . 30
  - 3.3.2 Isogeometric analysis . . . 30
  - 3.3.3 Gauss integration . . . 32
- 3.4 Validation . . . 32
- 3.5 Discussion . . . 34

4 Parametric geometry 35
- 4.1 Actual experiment . . . 35
- 4.2 Numerical setting . . . 37
- 4.3 Validation and verification of the parametric code . . . 39
- 4.4 Discussion . . . 40

5 FSI solver 43
- 5.1 Reference and scaling . . . 43
- 5.2 Pressure force integrator . . . 45
  - 5.2.1 Body generation . . . 45
  - 5.2.2 Methodology . . . 47

5.2.3 Validation . . . 50
5.3 FSI solver validation . . . 53
5.3.1 Methodology . . . 53
5.3.2 Validation . . . 55
5.4 Discussion . . . 60
5.4.1 Pressure force integrator . . . 60
5.4.2 FSI solver . . . 60

6 Investigation **61**
6.1 Length of the cylinder . . . 61
6.1.1 Numerical simulation . . . 61
6.1.2 Discussion . . . 67
6.2 Upper boundary condition . . . 68
6.3 Reynolds number . . . 68
6.4 Downstream end shape . . . 69
6.4.1 Extra blunt end . . . 69
6.4.2 Parachute . . . 70
6.4.3 Discussion . . . 71

7 Conclusion **72**

References **74**

A Conference paper outline **77**

B Solid structure solver validation code **79**

C Parametric body validation code **81**

D Name of the fluid domain boundaries **84**

E Velocity and pressure fields **85**

F FSI validation code **87**

G Pressure integrator validation **90**

H Integrator validation result **99**

I Cylinder shapes in FSI validation **102**

J Code for investigation **106**

K Pinned-free cylinder shapes **112**

L Change log **114**

Nomenclature

Abbreviations

Abbreviation	Definition
2D	Two dimensional
3D	Three dimensional
B.C.	Boundary condition
BDIM	Boundary data immersion method
CFD	Computational fluid dynamics
FEA	Finite element analysis
FEM	Finite element method
FFT	Fast Fourier transformation
FVM	Finite volume method
FSI	Fluid-Structure interaction
HTA	Hydrophone towed array
IGA	Iso-geometric analysis
IQN	Interface Quasi-Newton (method)
NURBS	Non-uniform rational B-spline (curve)
SD	Signed distance
SDF	Signed distance function
SNR	Signal-to-noise ratio

Symbols

Essential variables are provided. The variables used in the local description are not listed. All variables have definitions in the report.

Symbol	Definition
L_{arr}	Array cable length
D_{arr}	Array cable diameter
L_C	Critical length in the long model
D_{cyl}	Cylinder diameter in the cylinder experiment
L_{cyl}	Cylinder length in the cylinder experiment
U_{exp}	Far-field flow velocity in the cylinder experiment
w_{max}	Possible maximum lateral displacement in the cylinder experiment
f	Correction coefficient of the slender body theory
N_i	Basis function of in IGA
U_{real}	The flow velocity of the array cable's normal working condition
d_k, $\mathbf{d}_k$	Displacement vector used in IQN method
J_k	Jacobian matrix in the classical Newton method
f_k	External force vector used in IQN method
$R(d_k)$	Residual of d_k
$R(f_k)$	Residual of f_k
$R(k)$	Overall residual of iteration step k in IQN method
E_{sim}	Young's modulus used in the numerical simulation
I_{sim}	Moment of inertial of the numerical cylinder cross-section
w_y, w_z	Displacements of the central axis in y and z direction
q_y, q_z	External forces per unit length in y and z direction

Symbol	Definition
G_j	Gauss weight on Gauss point j
Err_y, Err_z	Accumulated error in the solid structure solver validation
EI	Bending stiffness of the analytical result
EI_y, EI_z	Bending stiffness in y and z directions
$C_{D,exp}$	Total drag coefficient in the airship experiment
$vol.$	Volume of the model airship in the airship experiment
Re_{exp}	Reynolds number determined by $vol.^{1/3}$
v_{exp}	Far-field velocity of the airship experiment
$C_{D,wet}$	Experiment's total drag coefficient determined by the wetted surface area
$C_{D,p,wet}$	Experiment's pressure drag coefficient determined by the wetted surface area
$C_{D,f,wet}$	Experiment's friction drag coefficient determined by the wetted surface area
L_{tip}	The distance from the central axis downstream end to the fluid field boundary
L_{nose}	The distance from the central axis upper end to the fluid field boundary
D_{width}	Half width of the numerical fluid domain
L_{sim}	Simulation's characteristic length and the length of the rotationally symmetric body's central axis
D_{sim}	Body diameter in the simulation
D_{sim1}	Cylinder diameter in the parachute model
D_{sim2}	Diameter on the end of the taper part in the parachute model
D_{sim3}	Parachute diameter
U_∞	Far-field flow velocity in the simulation
$vol.sim$	Volume of the numerical airship
Ca_{cyl}	Cauchy number of the cylinder experiment
E_{cyl}	Young's modulus used in the cylinder experiment
$E_{sim}I_{sim}$	Bending stiffness in the flow solver
$E_{para}I_{para}$	Parametric bending stiffness in the solid structure solver
D_{end}	Body's diameter on the central axis end
L_{end}	Length of the taper end in the cylinder experiment
$D(\xi)$	Diameter function based on ξ
$\mathbf{X}_{ci}$	Coordinates of Sampling points on the edge of the cross-section
$\mathbf{k}$	Rotation axis
S	Operator of the solid structure solver
F	Operator of the flow solver
f_n	Known force vector at the time step n
d_n	Known displacement vector at the time step n
Ca	Cauchy number used in the simulation
Re	Reynolds number used in the flexible cylinder simulation
u	Dimensionless far-field flow velocity in the cylinder experiment
L_l	Simulation fluid domain length
L_w	Simulation fluid domain width
L_h	Simulation fluid domain height
s_y,s_z	Dimensionless central axis' displacements in y and z directions
t_{conv}	Dimensionless convective time in the simulation
$u_{cd,sim}$	Critical velocity of divergence instability in the simulation
$u_{cf,sim}$	Critical velocity of flutter instability in the simulation
AR	Aspect ratio of the cylinder part in the simulation
AR_{cd}	Critical aspect ratio of divergence instability in the simulation
AR_{cf}	Critical aspect ratio of flutter instability in the simulation
$numE$	Number of the elements used in the solid structure solver
n_g	Quantity of the Gauss points used in the solid structure solver
P	Degree of the B-spline curve in the code
n_I	Number of control points in the element I
$\mathbf{n}_i$	Force's normal direction of a point on the cross-section's edge
n_{cr}	Quantity of the sampling points on the cross-section edge

Symbol	Definition
d_A	The lateral displacement at an arbitrary point A on the central axis of the cylinder
l_A	The longitudinal position and moment lever length of point A
f_{gy}, f_{gz}	Analytical pressure forces per unit length on Gauss points in the pressure force integrator validation
$f_{gy,sim}, f_{gz,sim}$	Numerical pressure forces per unit length on Gauss points in the pressure force integrator validation
Ξ	Knot vector of the B-spline curve
ρ_{sea}	Sea water density
μ_{sea}	Sea water dynamic viscosity
Ω_f	Incompressible viscous fluid domain
Ω_b	Solid or deforming body domain
ϵ	Slenderness in the experiment
μ_{exp}	Air dynamic viscosity
ρ_{exp}	Air density of the airship experiment
δ_v	Viscous sublayer thickness
ρ_{wat}	Water density
μ_{wat}	Water dynamic viscosity
ξ	Parametric coordinate in the longitudinal direction on the central axis
ξ_{cyl}	Parametric cylinder length
ξ_{sim}	Parametric central axis length (cylinder with different end)
ξ_{cone}	Cylinder with a cone's parametric length in the parachute model
ξ_{rope}	Cylinder with a cone and rope's parametric length in the parachute model
ξ_g	Parametric coordinate of a Gauss point
Ω_0	Basic cross-section
Ω_1	Rotated basic cross-section
Ω_c	Actual cross-section which passes a Gauss point
γ	Rotation angle of the plane
α	Slope angle of the taper part
μ_{sim}	Fluid dynamic viscosity in the simulation
ρ_{sim}	Fluid density in the simulation
β_{sim}	Mass ratio of solid and fluid in the simulation

Figure list

Figure	Description
Figure 1.1	Hydrophone Towed Array system[55].
Figure 1.2	The working condition of the HTA system[50][55].
Figure 1.3	Conditions of the hydrophone array cable[55].
Figure 1.4	Schematic of the HTA system (side view)[55].
Figure 1.5	Schematic of the array cable with different parts[55].
Figure 2.1	Boundary conditions of the array cable with all constraints. The downstream end shape is omitted.
Figure 2.2	Aerial refueling tanker[21].
Figure 2.3	Simplified boundary conditions of array cable. The downstream end shape is omitted.
Figure 2.4	Lateral displacements in different directions.
Figure 2.5	Instabilities of a clamped-free cylinder in axial flow.
Figure 2.6	Instabilities of a pinned-free cylinder in axial flow.
Figure 2.7	A cantilevered cylinder in axial flow[45].
Figure 2.8	Stability diagram for long cylinders, $l \geq 1$, from computations at $l = 10$, depending on the parameter f relative to the shape of the downstream end. —, Divergence limit; - - -, flutter limit; . . . , exact divergence limit adapted from the solution of Triantafyllou & Chryssostomidis[60][25].
Figure 2.9	(a) Stability diagram for a cylinder in axial flow modeled as a string, in terms of the dimensionless length l and flow velocity v; S denotes stability, and F denotes flutter. (b) The critical flow velocities for divergence and flutter for $f = 0.8$ and variable l; the full line shows results obtained by the full theory, while the dashed lines represent 'short cylinder' and long cylinder' approximations[25].
Figure 2.10	Model shape of the unstable modes as a function of the cylinder length with a tapered end and the clamped-free boundary condition. Flutter motion over a cycle of oscillations for cylinders with different lengths (modified)[25].
Figure 2.11	A flexible cylinder subjected to axial flow and supported only at the upstream end by a translational and a rotational spring, the stiffnesses of which are represented by k_0 and c_0. In this paper, k_0 is assumed to be infinite, and $c_0 = 0$, i.e., the pinned-free cylinder[22].
Figure 2.12	Variation of critical flow velocity for static and dynamic instabilities of a flexible, neutrally buoyant pinned-free cylinder as a function of εC_f. The numerical results obtained via Galerkin's method are Δ for $f = 0.7$ and $\circ$ for $f = 1.0$.The solid line shows the results obtained analytically via the Adomian Decomposition Method; the dashed line shows the linear interpolation of the numerical results. εC_f is the dimensionless length of the cylinder. f represents the shape of the cylinder's downstream end. A larger f indicates a more streamlined end[22].
Figure 2.13	Schematic diagram of experimental apparatus[43].
Figure 2.14	Instabilities of a clamped-free cylinder in axial flow by Païdoussis[43].
Figure 2.15	Comparison of critical velocities between analytical predictions and experiments. The symbol "I" represents the interval of critical velocities[43][45].
Figure 2.16	The effect of the slenderness ratio ϵ on the instability of clamped-free cylinders. u_{cb} is the critical velocity of buckling, and u_{co} is the critical velocity of second-mode oscillation[43].
Figure 2.17	Schematic diagram of the blowdown facility and photographic equipment[36].
Figure 2.18	Time-averaged cylinder position in the test section at three velocities. Note the greatly expanded vertical scale compared to the horizontal scale[36].

Figure	Description
Figure 2.19	Photographs illustrating (a) static divergence (yawing/buckling), (b) second-mode flutter, and (c) third-mode flutter of a horizontal pinned-free cylinder in axial flow[44].
Figure 2.20	Piecewise cubic B-Spline curve (solid line) and its control net (dotted)[8].
Figure 2.21	Global load-displacement matrix equation. u_i and f_j indicate the deflection at the ith node and the force at the jth node[53].
Figure 2.22	Schematic of a wavelike cantilever beam displacement by different methods. Black represents the original position. Red represents the displacement. f_y is the load. The Blue dashed line is the actual shape of the beam.
Figure 2.23	Basis functions as components of the interpolation function. Superpose basis functions to obtain the shape function or the displacement function[18][8].
Figure 2.24	Smoothing across the immersed boundary. The equations valid in each domain are convolved with a radius ϵ kernel and added together. The kernel at a point (marked by a dot) that belongs to the boundary region is represented[33].
Figure 2.25	Energy spectrum of turbulence in the function of wave number k, indicating the range of application of the DNS, LES, and RANS models. The length scales l_T and l_I are associated with the LES and RANS approximations[18][14].
Figure 3.1	Lateral pressure force at a point on the cylinder in the global system, where $\mathbf{q}_{sim}$ is the distribution load. The cylinder is simplified to a curve for better illustration. The curve can be regarded as the central axis of the cylinder.
Figure 3.2	Side view of a bending cylinder in the xoz plane. The cylinder is simplified to a curve, and the displacement is amplified for better illustration. Blue represents the original position, while black represents deformation. The long-dashed curve is the central axis of the cylinder. B point is on the axis and has no displacement in the x direction after bending. The cross-section past the point can rotate. The slight axial extension is neglected based on the small deflection assumption.
Figure 3.3	Lateral pressure force decomposition in the xoz plane. The cylinder is simplified to a curve, and the displacement is amplified for better illustration.
Figure 3.4	Schematic of an element in different IGA solvers. Simple flow charts of solvers are also provided. The dashed line is the control net. The solid line represents the beam. Black dots represent control points. Blue and red represent displacements and forces.
Figure 3.5	Global stiffness matrix of a cantilever beam. $EI = 10000$, $L = 10$. Two ghost columns and rows for boundary conditions are omitted.
Figure 3.6	Loads on the cantilever beam.
Figure 3.7	Solid structure solver validation. Numerical results are shown on 13 sampling points.
Figure 3.8	Flow chart of the improved solid structure solver.
Figure 4.1	Schematic of the experiment setting[17].
Figure 4.2	2D half contour of the airship based on the table 4.1.
Figure 4.3	Generation of the 3D body from 2D B-spline curve by mapping.
Figure 4.4	Schematic of the fluid domain and the airship. The green arrow is the far-field flow direction: x direction.
Figure 4.5	2D schematic of the domain verification. The blue arrows represent the streamline. The red line indicates an inappropriate domain. The black line is the appropriate domain that remains to be found.
Figure 4.6	Verification of the fluid domain.
Figure 4.7	Validation and verification of the parametric code. L variation of the pressure drag coefficient.
Figure 4.8	Viscous drag coefficient $C_{D,f}$ under various characteristic lengths.
Figure 4.9	Velocity profile near the solid boundary. The black curve is the velocity profile. y_1 represents the grid thickness. The shadowed rectangle represents the airship's surface. The numerical approximation is $tan\ \theta_1 = (u(y_1) - u(0))/y_1$. The actual velocity gradient is $tan\ \theta_v = \partial u(0)/\partial y$. It is apparent that $tan\ \theta_1 < tan\ \theta_v$. The numerical gradient is smaller than the actual one.

Figure	Description
Figure 4.10	Flow velocity profile around the airship. The black line represents the profile of the front sampling line, and the red line represents the profile of the back one. The profile near the physical boundary is smooth since we apply a kernel in BDIM.
Figure 4.11	Side schematic of a cylinder in the code. The red line represents the surface of the cylinder. The blue line is the central axis generated by the B-spline curve. The dashed line is the signed distance from the point to the axis.
Figure 5.1	Side view of dimensions of tested cylinder with a tapered end in different reference spaces. The solid black line represents the shape in real space. The dashed black line divides various segments of the whole body. Blue and red dashed lines are the central axis generated by the B-spline curve. $\xi_{sim} = 1$ represents the whole length.
Figure 5.2	Side view of cylinder dimensions with a parachute at the free end in the simulation and parametric space. The dashed black line divides the body into various segments: a cylinder, a cone, a rope, and a parachute. Blue and red dashed lines are the central axis generated by the B-spline curve. $\xi_{sim} = 1$ represents the whole length.
Figure 5.3	Schematic of two coordinate systems. Black arrows form the global system. Blue arrows form the local system. The dashed blue line is the displacement of the A point in the local system. The red line represents the axis of the bending cylinder. The axes of the two systems are parallel to each other.
Figure 5.4	Schematic of an inclined cylinder with an extra blunt (taper) end ($D_{end} > D_{sim}$) in the simulation. The central axis is in the xoy plane. $h_\xi = 0.3$ in this figure. The dashed red line is the central axis. The blue coordinate system is local (obtained by multiplying L_{sim} by the parametric system).
Figure 5.5	Schematic of an inclined cylinder with a parachute at the end. The central axis of the body is in the xoz plane. The central axis is straight. $h_\xi = 0.3$ in this figure. The green dashed line separates different segments.
Figure 5.6	Schematic of a semicircle cylinder with a constant diameter. $D_{sim} = 6$ in this figure. The dashed red central axis of the cylinder lies in the xoy plane. The blue coordinate system is local but not a parametric system.
Figure 5.7	Schematic of a 3D cylinder. The red line represents a cross-section. The red dot is the Gauss point in the cross-section. The long-dashed line is the central axis.
Figure 5.8	Schematic of the force integration in the cross-section. The red dot is the Gauss point.
Figure 5.9	Procedure of applying Rodrigues' rotation formula. G denotes the Gauss point.
Figure 5.10	Schematic of the slope angle α of different end shapes.
Figure 5.11	A cross-section contains the Gauss point in the local coordinate system.
Figure 5.12	Schematic of the solar system of the semicircle cylinder. The black dot represents the sampling points.
Figure 5.13	Comparison between analytical and numerical results of the semicircle cylinder. $\beta = \pi - \theta_{arc}$. The red line is the analytical results with $D_{sim} = 6$; the black boxes are its numerical approximation. The green line is the analytical results with $D_{sim} = 9$; the blue triangles are its approximation. The numerical force's z component is at $O(10^{-14})$.
Figure 5.14	Comparison between analytical and numerical results of the cylinder with a tapered end. $L_{sim} = 64$. $\xi_{cyl} = 0.8$. $D_{sim} = 6$. The red line is the analytical results with $D_{end} = 12$ and $h_\xi = 0.2$; the black boxes are its numerical approximation. The green line is the analytical results with $D_{end} = 0$ and $h_\xi = 0.3$; the blue boxes are its approximation. $x_{\xi L}$ is the longitudinal coordinate of the sampling point. The numerical force's z component is at $O(10^{-14})$.

Figure	Description
Figure 5.15	Comparison between analytical and numerical results of the cylinder with a parachute end. $L_{sim} = 64$. $\xi_{cyl} = 0.7$, $\xi_{cone} = 0.8$, and $\xi_{rope} = 0.9$. $D_{sim1} = 8$, $D_{sim2} = 2$, and $D_{sim3} = 12$. The red line is the analytical results with $h_\xi = 0.2$; the black boxes are its numerical approximation. The green line is the analytical results with $h_\xi = 0.3$; the blue triangles are its approximation. The numerical force's y component is at $O(10^{-14})$.
Figure 5.16	Flow chart of the coupling solver in the FSI code at time step $n + 1$.
Figure 5.17	Side view of the fluid field for the validation. The red dashed line represents the central axis generated by the B-spline curve. The ghost grids for the boundary condition are not shown in the figure.
Figure 5.18	Velocity fields of cylinder in different instabilities.
Figure 5.19	Displacements diagram over t_{conv}. The dashed and solid lines are the midpoint and endpoint displacements.
Figure 5.20	Absolute displacement ranges at the downstream end when $t_{conv} > 20$ under various u. The cylinder has unclear instability when $u = 2.0$ and has flutter instability when $u = 5.0$.
Figure 5.21	Ranges of divergence and flutter critical velocities. Green and orange horizontal I are divergence ranges from the experiment and the simulation. The flutter ranges from the experiment and the simulation are blue and red horizontal I.
Figure 6.1	Time-varying displacements of the clamped-free cylinder with a taper end and various aspect ratios in axial flow. The solid and dashed lines represent the displacements at the downstream end and midpoint.
Figure 6.2	Shapes of the clamped-free cylinder's central axis in xoy and xoz planes. $AR = 30$. Symbols are sampling points.
Figure 6.3	Time-varying displacements of the pinned-free cylinder with a taper end and various aspect ratios in axial flow. The solid and dashed lines represent the displacements on the downstream end and the midpoint.
Figure 6.4	Absolute displacement ranges at the downstream end when $t_{conv} > 20$ with various AR. The downstream end oscillates around the zero position when $AR = 40$.
Figure 6.5	Shapes of the pinned-free cylinder's central axis in xoy and xoz planes. $AR = 40$. Symbols are sampling points.
Figure 6.6	Velocity sampling point and the comparison diagram between the flutter and the vortex-shedding frequencies.
Figure 6.7	Method of predicting the critical velocity line. u_x is the far-field flow velocity.
Figure 6.8	Time-varying displacements of the pinned-free cylinder with a taper end and $Re = 6489$. The solid and dashed lines represent the displacement at the downstream end and midpoint.
Figure 6.9	Time-varying displacements of the pinned-free cylinder with a taper end and $Re = 648.9$. The solid and dashed lines represent the displacement at the downstream end and the midpoint.
Figure 6.10	Fluid field schematic in the xoy plane. $Re = 648.9$. $t_{conv} = 4$.
Figure 6.11	Time-varying displacements of the clamped-free cylinder with an extra blunt end. The solid and dashed lines represent the displacement at the downstream end and the midpoint.
Figure 6.12	Time-varying displacements of the pinned-free cylinder with an extra blunt end. The solid and dashed lines represent the displacements at the downstream end and the midpoint.
Figure 6.13	Time-varying displacements of the pinned-free cylinder with a parachute end. $AR = 40.00$.
Figure 6.14	Bending cylinder with a parachute end in the xoy plane. $t_{conv} = 30$.
Figure 7.1	Modified array cable.

Table list

Table	Description
Table 1.1	Dimension of hydrophone towed array system.
Table 4.1	Dimensions of N.P.L. airship models in inches.
Table 5.1	Parameters of the FSI validation cases.
Table 6.1	Parameters of length effect investigation cases.

1

Introduction

1.1. Research background

Due to the low visibility in deep water, using acoustic methods to investigate the deep ocean environment is becoming increasingly popular. Based on this requirement, the hydrophone was developed. This equipment is able to filter out background noise, including noise from the sea surface and hull vibration[55].

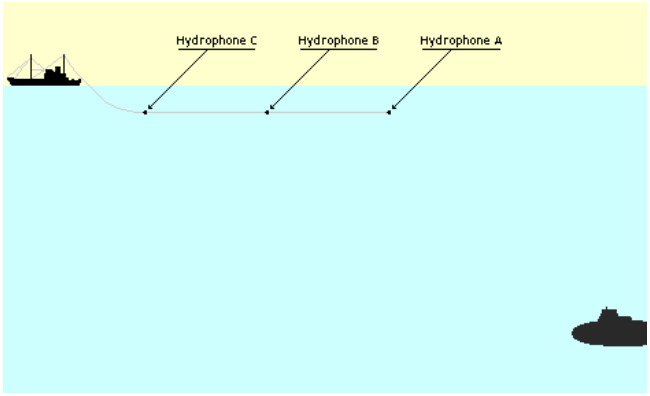
A single acoustic sensor can only obtain limited audio information, and its sensitivity is relatively low. Considering the limitation, the hydrophone towed array (HTA) system has been designed to localize the acoustic source and improve SNR (signal-to-noise ratio), as shown in Figure 1.1.

(a) Single hydrophone.

(b) Hydrophone array.

Figure 1.1: Hydrophone Towed Array system[55].

HTA contains a large number of hydrophones, which are installed in a flexible tube. Each hydrophone is arranged along the center line of the tube, which makes the flexible tube significantly longer. A long hydrophone array can also bring about better sensitivity and angle resolution. This cable-like hydrophone array is towed behind a vessel by a cable. The hydrophone array is positioned far from the boat to minimize the impact of the noise from the ship. Multiple hydrophone arrays can be set in parallel to improve acoustic resolution further. The working condition of the HTA system is shown in Figure 1.2.

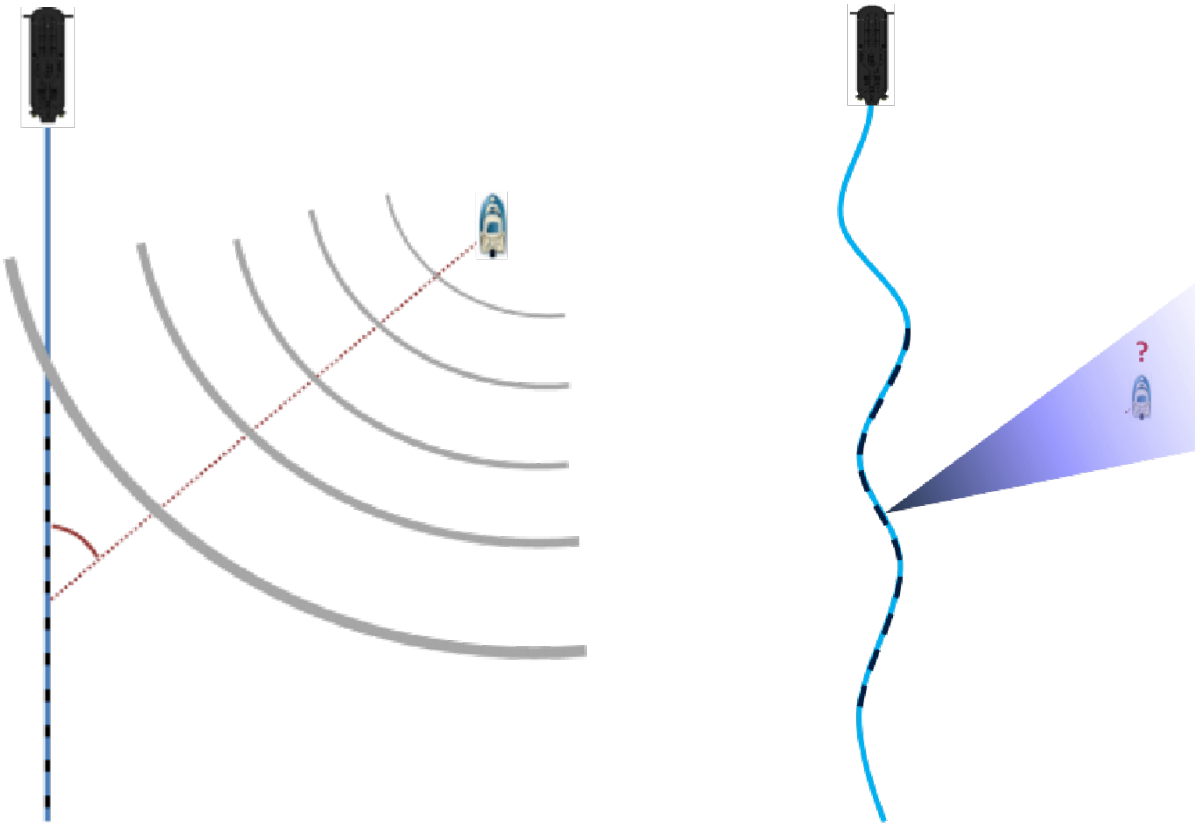


(a) Side-view of HTAs[50].

(b) Parallel setting of multiple hydrophone arrays[55].

Figure 1.2: The working condition of the HTA system.

The hydrophone towed array is designed to have neutral buoyancy and is towed at a constant velocity. In ideal conditions, the hydrophone array cable should be perfectly straight, as shown in Figure 1.3a, which is an assumption of beamforming algorithms. If the array cable distorts from its straight shape, lateral displacement and an inclined angle can cause significant performance degradation. The localization of acoustic sources will be influenced, as shown in Figure 1.3b.



(a) The ideal condition.

(b) The deformed condition.

Figure 1.3: Conditions of the hydrophone array cable[55].

The typical structure of the HTA system is given by the company Optics11. Figure 1.4 shows the geometry of the whole HTA system with the vessel. Figure 1.5 shows that the so-called array cable contains the array and connectors. The cable connectors are installed on the upstream and downstream ends

of the array cable, which are smooth and only a few millimeters larger in diameter than the array. The array cable is towed by a tow cable on the upstream side, and a drag rope is attached to the downstream side. The tow cable is allowed to bend. The array cable has neutral buoyancy and is supposed to be perfectly straight and horizontal.

In more detail, the array cable should be parallel to the steady water plane and ship velocity vector (i.e., tow direction). The drag rope is much thinner than the array cable, and its end is free. The purpose of the drag rope is to provide additional drag, making the array cable straight. The selection of drag rope is semi-random[55].

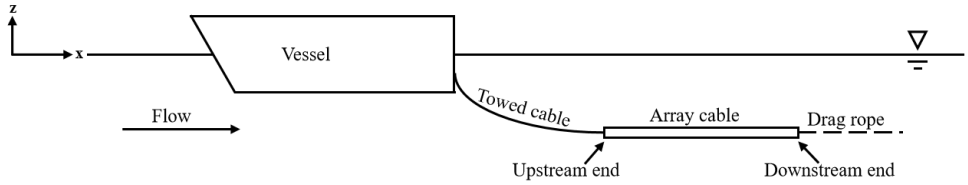


Figure 1.4: Schematic of HTA system (side view)[55].

The dimension of the system is given in Table 1.1. The aspect ratio of a typical array cable (L_{arr}/D_{arr}) is enormous. The standard working depth is $100\ m$, and the towing speed is 1 to $10\ knots$. The array cable can be regarded as a flexible (allows bending) but inextensible cylinder with various upstream and downstream ends. The deep water assumption is applicable, so there is no effect from the water surface.

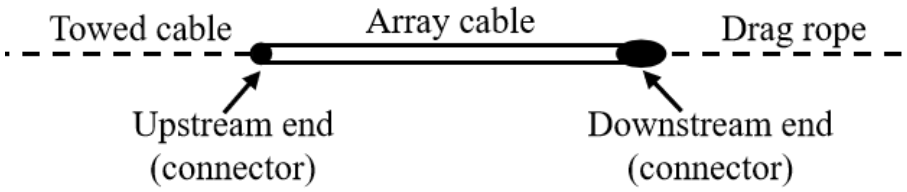


Figure 1.5: Schematic of the array cable with different parts[55].

Table 1.1: Dimension of hydrophone towed array system[55].

Tow cable length	l_{tow}	500	m	Tow cable diameter	d_{tow}	0.02	m
Array cable length	L_{arr}	100	m	Array cable diameter	D_{arr}	0.03	m

The hydrodynamics of the array cable are critical for the operational performance. The overall objective is to optimize the hydrodynamics of the array cable to maximize its performance. The acoustic performance is mainly affected by (1) distortions in the shape of the array cable and (2) flow noise[55]. This project will focus on the shape distortion effect of a solitary array cable in flow.

1.2. Research question and report structure

To reduce the shape deformation of the array cable - a flexible cylinder, the fundamental cause of the phenomenon should be found. Apparently, the flexible cylinder will deform if the flow field is not axis-symmetric along its axis. Intuitively, making the flow field around the cylinder axially symmetric could suppress the deformation of the cylinder. However, self-sustaining buckling and unstable oscillation of a flexible cylinder have been observed even when the cylinder is in axial flow[43], which is defined as the instability of the flexible cylinder. The regular axial flow with constant flow speed is the ideal working condition of the HTA system. If the array cable deforms even in this perfect condition, the situation will be worse in the disturbed flow field. In other words, to reduce the shape deformation of a flexible cylinder in a complex flow field, it must first be stable in an ideal condition, which means no buckling or oscillation in axial flow. This introduces the research question of this project:

Instabilities of a solitary flexible cylinder in axial flow.

The axial flow is the far-field flow (initial condition). The flow direction is aligned with the cylinder's initial central axis. There is a coupling effect between the flexible cylinder and the flow, which means the cylinder disturbs the flow field, and the fluid simultaneously deforms the cylinder. The research question is a fluid-structure interaction (FSI) problem.

Previous research indicates that multiple reasons cause different types of instabilities. A literature review is required to define the research matrix and select an appropriate research method. These will be provided in Chapter 2. A solid structure solver, a code for the parametric body, and a coupling solver are required to solve an FSI problem. Chapters 3, 4, and 5 provide their methodologies and validations. Once the research method has been validated, the investigation of the research question will follow in Chapter 6. The details of the experimental setting and the results analysis will be shown in this chapter. The overall conclusion and the future plan will be provided in Chapter 7. All coordinate systems in this project obey the right-hand rule. Gravity is neglected because the array cable is neutrally buoyant.

2

Literature review and research matrix

Previous research has employed multiple approaches to investigate the instability of a cylinder in axial flow, including analytical research, numerical simulations, and experiments. Different factors affect cylinder instability. This chapter will briefly introduce existing results from various research methods and then select appropriate parameters for this project to explore. After providing reasons, a clear research matrix will be obtained. The research approach of this thesis will also be determined. The main contribution of this thesis project is shown at the end of the chapter.

2.1. Model simplification

The boundary conditions of the array cable are complex. This section introduces the simplification of boundary conditions.

Based on the observation of the actual array cable, the complete boundary conditions at the upstream end and downstream end can be obtained. The upstream end of the array cable is connected to the towed cable. The upstream connector is smooth and only a few millimeters larger in diameter than the array, as shown in Figure 1.4, so this connector can be regarded as a half-sphere with the cylinder diameter. The upstream end is neither wholly free nor fixed. Inspired by Figure 2.11[22], as shown in Figure 2.1, it should have elastic constraints in the x, y, and z directions and elastic rotational constraints in all rotation directions.

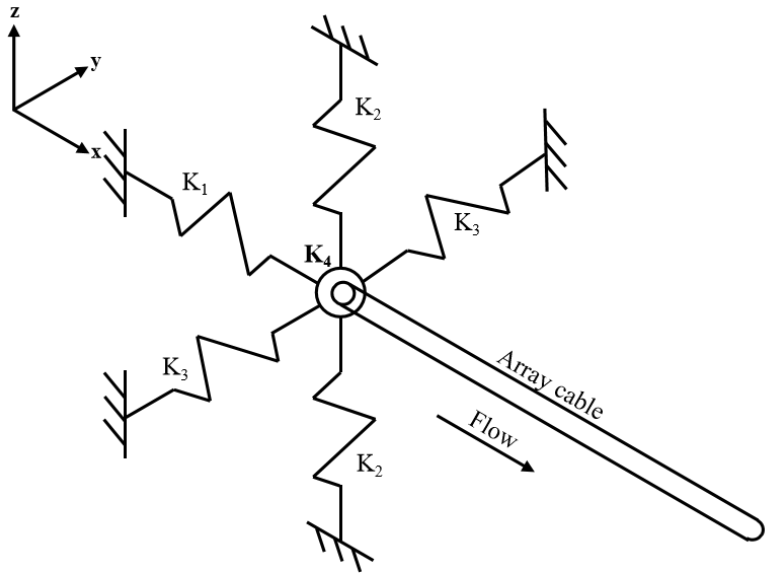


Figure 2.1: Boundary conditions of the array cable with all constraints. The downstream end shape is omitted.

The parameters K_1, K_2, K_3 are translational stiffness, and $\mathbf{K}_4$ is rotational stiffness[22]. The stiffness could be nonlinear. The downstream end is considered to be completely free. Such a physical model is complex due to the multiple degrees of freedom. A further simplification is possible.

The upstream end is also the end of the towed cable. The working condition of the towed cable end is similar to that of the end of the fuel pipe installed on an aerial refueling tanker. The fuel pipe is also flexible and has bending stiffness. Based on the observation of reality, the fuel pipe end is relatively stable. It has tiny displacements in all directions, as shown in Figure 2.2.

Figure 2.2: Aerial refueling tanker[21].

The motion of the fuel pipe end indicates that, in our case, the displacements at the towed cable end (the upstream end of the array cable) should be small. The translational stiffness K_1, K_2, and K_3 can be assumed to be infinite, so there are no displacements at the upstream end. An experiment pointed out that the oscillations of a clamped-free cylinder in axial flow are orbital[43], so the upstream end of the array cable should have freedom of rotation.

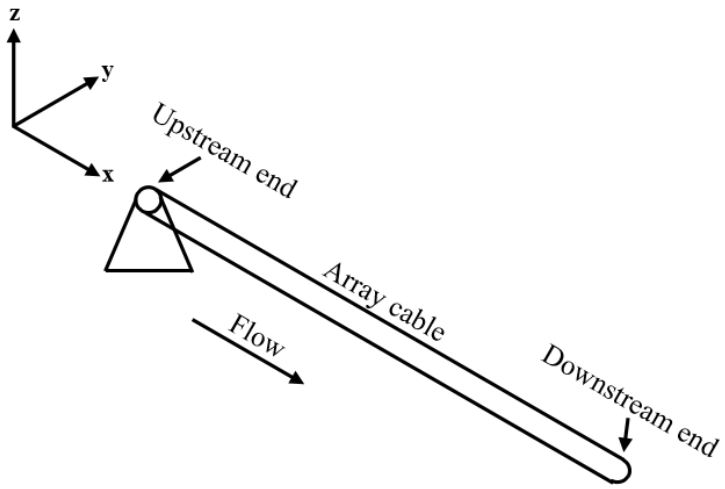


Figure 2.3: Simplified boundary conditions of array cable. The downstream end shape is omitted.

Considering the above reasons, the rotational stiffness $\mathbf{K}_4$ at the upstream end is assumed to be zero, so there is no rotational constraint. In summary, the simplified boundary conditions of the array cable will be pinned at the upstream end and completely free at the downstream end, as shown in Figure 2.3.

The drag rope after the downstream end is much thinner than the array cable. The tangential viscous drag is negligible; the reason is provided in section 3.2. The drag rope can be removed from the

simplified model. The downstream end now has no constraints, so it is free to move. According to the later literature review, the shape of the downstream end should be considered.

2.2. Literature review

An unstable cylinder cannot maintain its straight shape and original direction in axial flow. In other words, an arbitrary point on the unstable cylinder will have non-vanishing lateral displacements, as shown in Figure 2.4. The solid bending curve represents the central axis of the flexible cylinder. The rotation angles are derivatives of displacements.

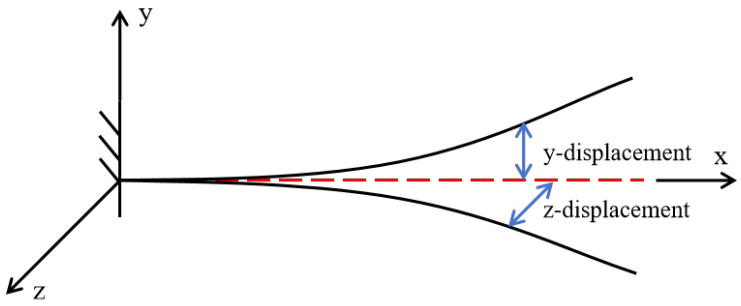
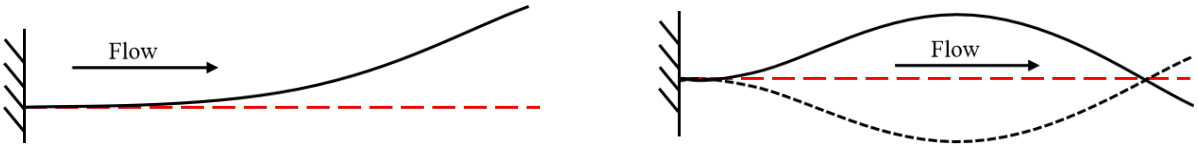


Figure 2.4: Lateral displacements in different directions.

There are two basic types of instability of a flexible cylinder[44][46]:

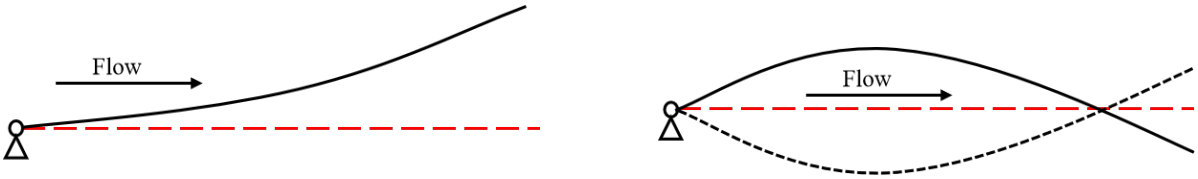
- **Divergence (buckling/yawing)**: Static buckling or yawing of the cylinder. The points on the cylinder deviate from the original axis to the same side and develop a new neutral position, as shown in Figure 2.6a. For a clamped-free (the upstream end is fixed, and the downstream end is free) cylinder, the shape is similar to a beam deflected in its first mode[43][51].
- **Flutter**: Oscillation of the cylinder. The points on the cylinder oscillate at different sides from the original axis, as shown in Figure 2.6b. For a clamped-free cylinder, the flutter can be the second mode or higher[44][51]. The dashed curve indicates the flutter cylinder at a different time step.



(a) Divergence instability (buckling). **(b)** Flutter instability.

Figure 2.5: Instabilities of a clamped-free cylinder in axial flow.

The different types of instability of a pinned-free (the upstream end is articulated, and the downstream end is free) cylinder are shown in Figure 2.6.



(a) Divergence instability (yawing). **(b)** Flutter instability.

Figure 2.6: Instabilities of a pinned-free cylinder in axial flow.

Various factors affecting instability have been tested in previous research using different approaches.

2.2.1. Analytical research

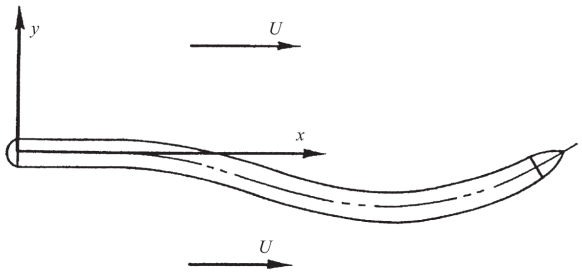


Figure 2.7: A cantilevered cylinder in axial flow[45].

The analytical method determines the instability by solving essential physical equations with boundary conditions. The physical and mathematical models are usually simplified to make them solvable. For example, Païdoussis[44] used a two-dimensional linear model to predict the instability of a "not too long" flexible cylinder in axial flow. He used force balance in the global coordinate system to build motion equations 2.1 and 2.2 in x- and y-directions:

$$\frac{\partial}{\partial x}(T+pA)+\left(mg-\frac{\partial p}{\partial x}A\right)+F_L+(F_N+F_A)\frac{\partial y}{\partial x}=0\ , \tag{2.1}$$

$$\frac{\partial Q}{\partial x}-F_N-F_A+F_L\frac{\partial y}{\partial x}+\frac{\partial}{\partial x}\left[(T+pA)\frac{\partial y}{\partial x}\right]-m\frac{\partial^2 y}{\partial t^2}=0\ , \tag{2.2}$$

where Q is the lateral shear force, T the axial tension, and m the mass of the cylinder per unit length[44]. Q is determined by the elementary Euler-Bernoulli beam model, as shown in equation 2.3, where E^* is the viscoelastic constant[44]. If E^* is negligible, the equation 2.3 is the same as the Euler-Bernoulli beam equation with small bending deflection simplification[38].

$$Q=-\frac{\partial}{\partial x}\left(EI\frac{\partial^2 y}{\partial x^2}\right)-E^*\frac{\partial}{\partial x}\left(I\frac{\partial^3 y}{\partial x^2\partial t}\right) \tag{2.3}$$

F_A is the inviscid force determined by the slender body theory, as shown in equation 2.4[31][44].

$$F_A=\left(\frac{\partial}{\partial t}+U\frac{\partial}{\partial x}\right)\left[M\left(\frac{\partial y}{\partial t}+U\frac{\partial y}{\partial x}\right)\right] \tag{2.4}$$

F_N and F_L are viscous forces in the normal and tangential directions. Païdoussis applied Taylor's formulas[56], which have coefficients based on solid or fluid characteristics, to determine the viscous force in different directions, as shown in Equations 2.5 and 2.6[44]:

$$F_N=\frac{1}{2}\rho DU^2(C_f sin\ i+C_{Dp}sin^2\ i)\ , \tag{2.5}$$

$$F_L=\frac{1}{2}\rho DU^2C_f sin\ i\ , \tag{2.6}$$

where D is the cylinder diameter, C_{Dp} and C_f are the coefficients associated with form and friction drag.

Païdoussis' analytical model indicates that the shape of the cylinder's downstream end is one of the critical effects on the instability. The downstream (free) end shape is a boundary condition. The elementary

Euler-Bernoulli beam model and the slender body theory are also applied. The primary modification is the inviscid force term F_A, which is corrected by a parameter f, as shown in equation 2.7[44]:

$$\int_{L-l}^{L} \frac{\partial Q}{\partial x} dx - f \int_{L-l}^{L} \left(\frac{\partial}{\partial t} + U \frac{\partial}{\partial x} \right) [M(x)v] dx - \int_{L-l}^{L} m(x) \frac{\partial^2 y}{\partial t^2} dx = 0 \ , \tag{2.7}$$

where l is the free end length, and v is the resultant relative velocity[31]. The range of f is from 0 to 1. If f is closer to 0, the end is more blunt; if f is closer to 1, the end is close to streamlined[44].

The correction is required because the theoretical lateral force at the free end may not be fully realized as a result of (I) the lateral flow not being truly two-dimensional since the fluid has the opportunity to pass around rather than over the tapered end[35], and (II) boundary-layer effects[15].

The analytical model predicts no instability if $f = 0$. Decreasing f should have a stabilizing effect. The results are validated qualitatively by experiments[42][44][46].

For a long cantilevered cylinder with clamped-free boundary conditions and neutral buoyancy, a modification of Païdoussis $et\ al.$'s linear analytical model was developed by Langre $et\ al.$[25]. A short description of this model is given below.

The cylinder is modeled as a beam, which means its flexural rigidity is non-vanishing. The equation governing the lateral motion $Y(X,T)$ reads[25]:

$$EI \frac{\partial^4 Y}{\partial X^4} - \frac{\partial}{\partial x} (\Theta \frac{\partial Y}{\partial X}) + m \frac{\partial^2 Y}{\partial T^2} = F_F \ , \tag{2.8}$$

where Y is the lateral position, x is the axial position, EI is the flexural rigidity, $\Theta(X)$ is the local axial tension, m is the mass per unit length, and F_F is the transverse fluid force per unit length acting on the beam. A uniform axial flow is assumed. The fluid loading resulting from the motion may be modeled as the sum of the following: the inviscid force, the drag force, the base drag force, and the lift force exerted at the downstream end[25].

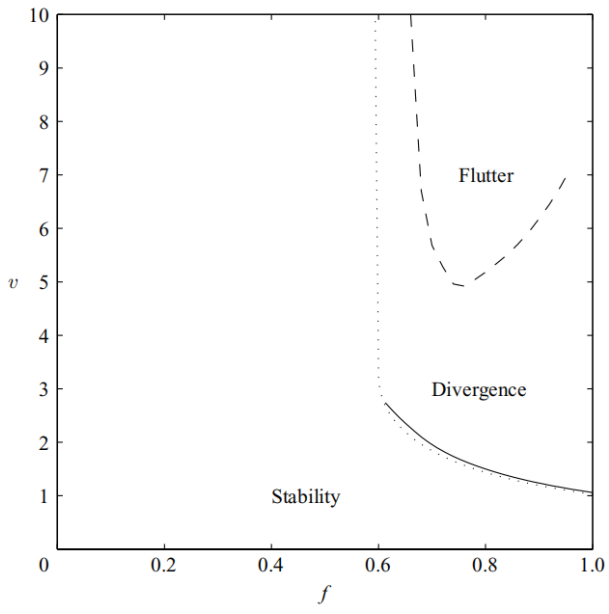


Figure 2.8: Stability diagram for long cylinders, $l \geq 1$, from computations at $l = 10$, depending on the parameter f relative to the shape of the downstream end. —, Divergence limit; - - -, flutter limit; . . . , exact divergence limit adapted from the solution of Triantafyllou & Chryssostomidis[60][25].

The major modification of this model is using the neutral point. The position of the neutral point is defined by the critical length L_C:

$$L_C = \frac{D(\pi - 2C_b)}{2C_T}, \tag{2.9}$$

where D is the diameter of the cylinder, C_b is the base drag coefficient, and C_T is the tangential coefficient. The coordinate system's origin is at the cylinder's downstream end. Downstream of the neutral point, $-L_C < X \leq 0$, the beam is in compression, and upstream of this point, $X < -L_C$, is in tension. With all points downstream of the neutral point being in compression, no stiffness exists in this cylinder segment other than that due to flexural rigidity. It should be noted that the position of this neutral point does not depend on the flow velocity. The critical length L_C is used in the scaling in place of the cylinder L, as in Doare & Langre[12] and Lemaitre $et\ al.$'s[30] papers to analyze the effect of length on stability more appropriately. The dimensionless length is $l = L/L_C$, where L is the cylinder length[25].

Substitute the fluid loading in equation 2.8 and apply the boundary conditions; the partial differential equation can be solved numerically to obtain the lateral motion of the long cylinder in axial flow. The results of critical flow velocities are shown in figures 2.8 and 2.9.

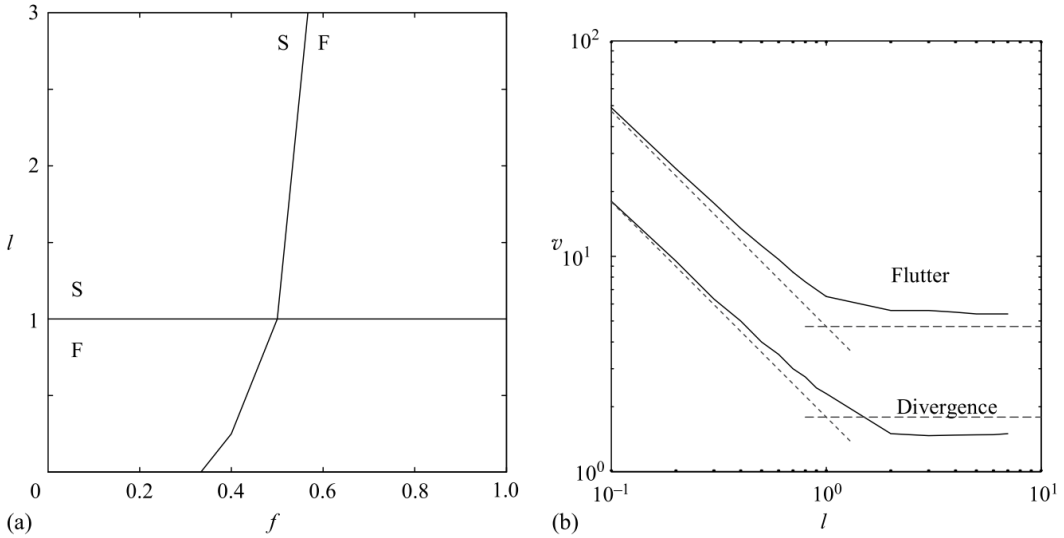


Figure 2.9: (a) Stability diagram for a cylinder in axial flow modeled as a string, in terms of the dimensionless length l and flow velocity v; S denotes stability, and F denotes flutter. (b) The critical flow velocities for divergence and flutter for $f = 0.8$ and variable l; the full line shows results obtained by the full theory, while the dashed lines represent 'short cylinder' and long cylinder' approximations[25].

Langre $et\ al.$'s analytical model successfully predicts the existence of instability, especially the flutter, of a long cylinder with a sufficiently streamlined downstream end in axial flow[25]. Figure 2.10 shows the model shape of the unstable modes.

Figure 2.10: Model shape of the unstable modes as a function of the cylinder length with a tapered end and the clamped-free boundary condition. Flutter motion over a cycle of oscillations for cylinders with different lengths (modified)[25].

Compared to Ni & Hansen's experiment[36] ($l = 5$), the critical velocities are of the same order of magnitude as shown in figure 2.9, but the predicted f has higher values. Considerable work remains

to achieve the same accuracy in experimental comparisons for long cylinders as is available for short ones[25].

In our case, the upstream end of the array cable is connected to a movable towed cable. A fixed boundary condition may be unsuitable at this position. Removing a constraint could reduce the model error, for example, by considering the boundary condition as ”pinned-free.”

For a pinned-free cylinder in axial flow, Kheiri and Païdoussis[22] developed a two-dimensional analytical model to predict the instability of the pinned-free cylinder in axial flow. The extended Hamilton's principle is used to obtain the linear equation of motion and boundary conditions for the cylinder. The schematic of the pinned-free is shown in Figure 2.11, and the predicted critical velocity is shown in Figure 2.12. The results show that the shape of the downstream end affects the instability of the flexible cylinder in axial flow. A more blunt downstream end provides a larger stable region in Figure 2.12. εC_f is the dimensionless length scale of the pinned-free cylinder. Figure 2.12 also indicates that increasing length causes the cylinder to become more unstable before the length reaches a critical value. The results are validated qualitatively by experiments conducted in 1965 by Païdoussis[22].

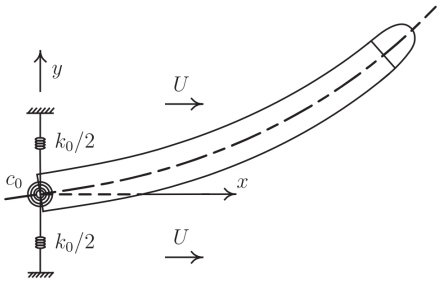


Figure 2.11: A flexible cylinder subjected to axial flow and supported only at the upstream end by a translational and a rotational spring, the stiffnesses of which are represented by k_0 and c_0. In this paper, k_0 is assumed to be infinite, and $c_0 = 0$, i.e., the pinned-free cylinder[22].

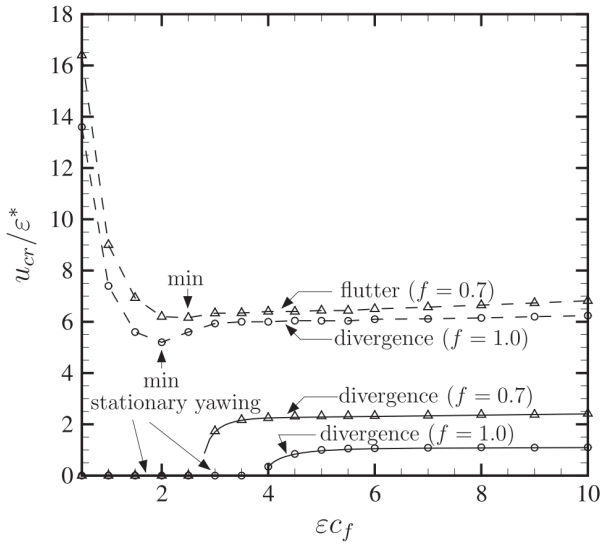


Figure 2.12: Variation of critical flow velocity for static and dynamic instabilities of a flexible, neutrally buoyant pinned-free cylinder as a function of εC_f. The numerical results obtained via Galerkin's method are Δ for $f = 0.7$ and $\circ$ for $f = 1.0$.The solid line shows the results obtained analytically via the Adomian Decomposition Method; the dashed line shows the linear interpolation of the numerical results. εC_f is the dimensionless length of the cylinder. f represents the shape of the cylinder's downstream end. A larger f indicates a more streamlined end[22].

Although analytical models can qualitatively predict a cylinder's instability in axial flow, they have a fatal weakness: the model error is vast. In an FSI problem, cross-flow and coupling effects play a critical role.

However, in analytical models, the cross-flow effect and hydraulic forces are determined by coefficients based on the characteristics of solids or fluids.

Both Reynolds and Cauchy numbers are critical for a problem with a flexible cylinder in axial flow, while analytical models focus on Cauchy numbers. Flow similarity is not guaranteed if there is no flow field, so the fluid-structure coupling cannot be simulated. Furthermore, when predicting the instability of a cylinder with different lengths, the model requires modification, which causes some inconvenience.

Still, high efficiency makes the analytical model a powerful research approach. The model can predict the instability of a cylinder with a significant length. The analytical model also indicates that the Euler-Bernoulli beam theory is applicable.

2.2.2. Experiment

To validate his analytical model, Païdoussis did multiple experiments to investigate the motion of a solitary cantilever cylinder in axial flow. The effect of the downstream end shape on the instability has been tested. The experiment apparatus of a horizontal cylinder is shown in Figure 2.13. The diameter of the cylinder D_{cyl} was $0.653\ inch$, and the diameter of the cylindrical flow field was $2.000\ inch$. The cylinder was aligned along the central axis of the flow field so that the possible maximum lateral displacement w_{max} of a point on the cylinder was less than twice the cylinder diameter D_{cyl}. Considering the cylinder length L_{cyl}, which was $15.400\ inch$[43], a relation can be easily obtained: $w_{max}/L_{cyl} < 0.1$, which indicates that the small deflection assumption is suitable for the solid mechanism of this project[58].

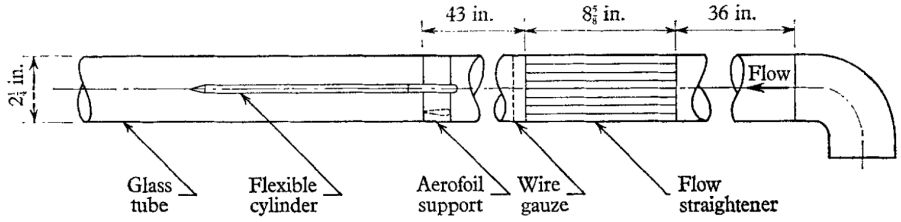
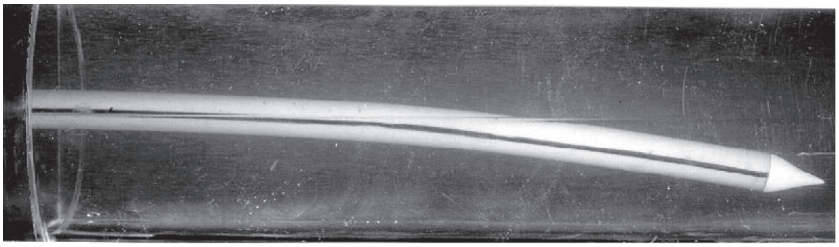
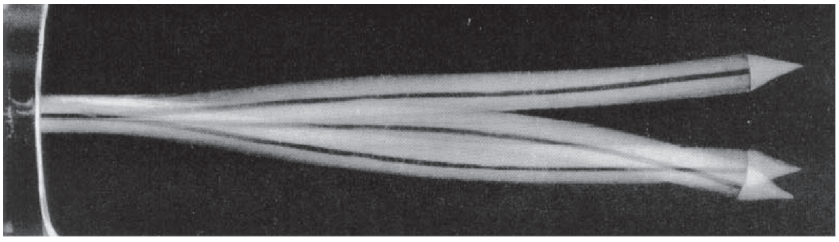


Figure 2.13: Schematic diagram of experimental apparatus[43].

A metal strip was embedded in the cylinder to prevent sagging since the cylinder was not neutrally buoyant. Various shapes of the downstream end, which were represented by the parameter f, had been tested. The critical velocities were obtained by continuously increasing the flow velocity. The cylinder oscillations are generally three-dimensional (orbital)[43]. Different types of instabilities of a clamped-free cylinder in axial flow are shown in Figure 2.14.



(a) Photographs of a flexible clamped-free cylinder in axial flow in divergence (buckling).

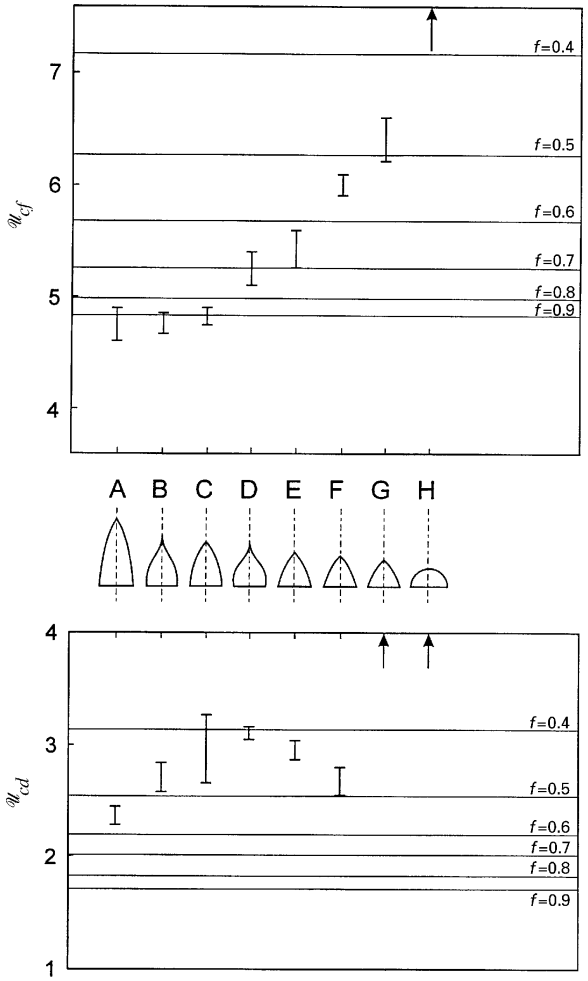


(b) Photographs of a flexible clamped-free cylinder in axial flow undergoing second-mode flutter.

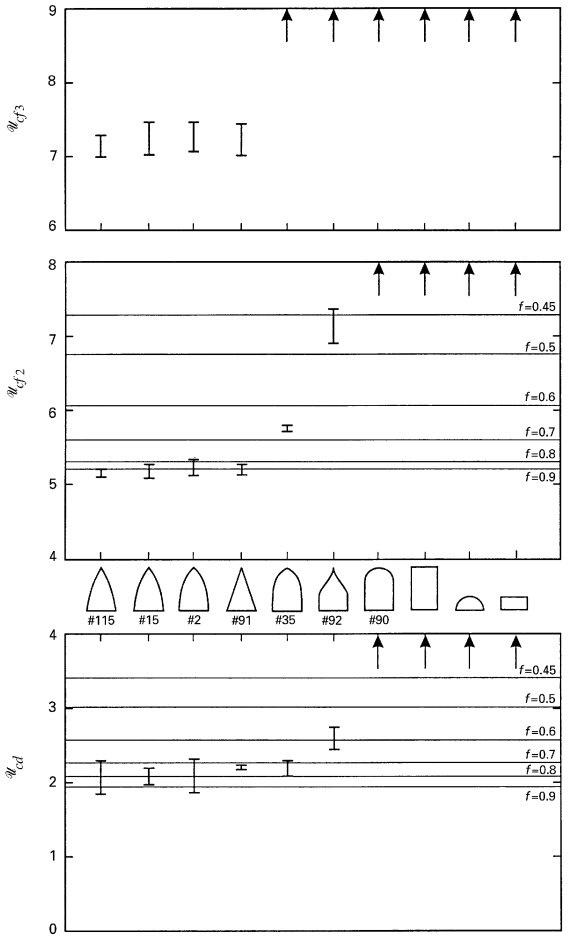
Figure 2.14: Instabilities of a clamped-free cylinder in axial flow by Païdoussis[43].

The results of critical velocities for the horizontal clamped-free cylinder are shown in Figure 2.15a. The same test for a vertical, clamped-free cylinder was also completed (with different cylinder parameters) [46], and the results are given in Figure 2.15b. At small flow velocities, small random vibrations were always damped. When f was not too small nor the slenderness ϵ too large, buckling developed slowly with increasing flow velocity. The transition from buckling to second-mode instability involved a gradual return of the cylinder to its position of rest along the x-axis before the further increase in the flow velocity resulted in unstable oscillation. Increasing the velocity after this point caused the cylinder to flutter in the second mode, then in higher mode[43]. Notice that increasing flow velocity not only changed the Reynolds number but also changed the Cauchy number.

The experiments qualitatively confirm the prediction from the analytical method. For a "not too long" cylinder, the streamlined downstream end (smoothly tapered, $f \to 1$) makes the cylinder more unstable, while the blunt end ($f \to 0$) has a stabilizing effect. If the flow velocity is sufficiently small, small random vibrations are always damped[43], no matter the downstream end shape.



(a) The critical velocities for divergence, u_{cd}, and second-mode flutter, u_{cf}, as functions of the free-end shape for the horizontal system: I, experimental data; -, theoretical results from varying f.[43]



(b) The critical velocities for divergence, u_{cd}, and second-and third-mode flutter, u_{cf2} and u_{cf3}, respectively, as functions of the free-end shape for the more recent experiments with vertical cylinders (without metal strip): I, experimental data; -, theoretical results from varying f.[45]

Figure 2.15: Comparison of critical velocities between analytical predictions and experiments. The symbol "I" represents the interval of critical velocities.

Païdoussis also experimented to test the effect of cylinder length, in other words, the slenderness (aspect ratio) $\epsilon = L_{exp}/D_{exp}$. The length was successively reduced by cutting off pieces from the free end. The same tapered end-piece was used in all tests. The results are shown in Figure 2.16.

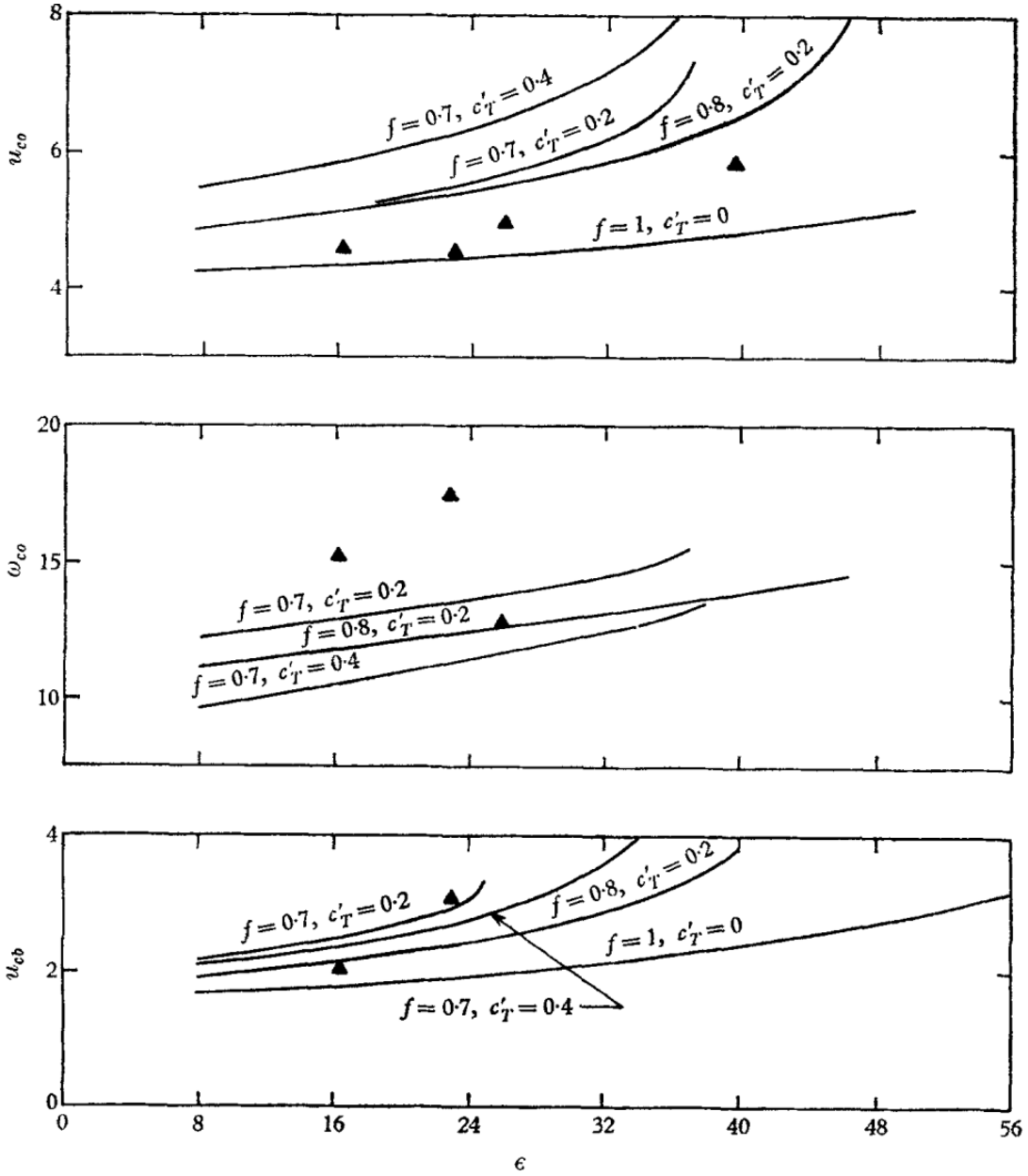


Figure 2.16: The effect of the slenderness ratio ϵ on the instability of clamped-free cylinders. u_{cb} is the critical velocity of buckling, and u_{co} is the critical velocity of second-mode oscillation[43].

When $\epsilon > 24$, no static buckling developed. At $\epsilon = 39.6$, the amplitude and frequency of oscillation were erratic, the latter vanishing occasionally while the cylinder retained an S-shape. When $\epsilon > 40$, there was no second mode instability, at least with the maximum attainable flow velocity[43]. The experiment shows that expanding the length in a specific range could stabilize the clamped-free cylinder. The critical velocity of divergence and flutter may increase. The transition of instability as the cylinder slenderness changes remains to be investigated.

It is possible to experiment with a model-scale cylinder that has a relatively large aspect ratio. As shown in Figure 2.17, Ni and Hansen experimented as in[36]. They set up an experiment to monitor the flow-induced motion of a cylinder in axial flow. The boundary condition of the cylinder was clamped-free. The tested cylinder was 7.92 meters long and had a diameter of 0.0159 meters. The aspect ratio was 498.11, which was relatively large but still could not reach the level of a thousand, the typical aspect ratio of an array cable.

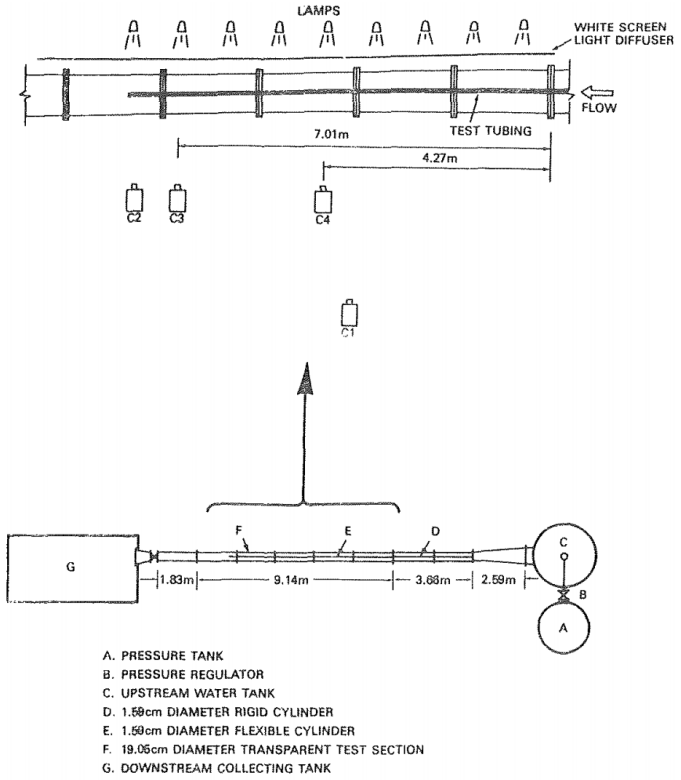


Figure 2.17: Schematic diagram of the blowdown facility and photographic equipment[36].

The time-averaged positions of the cylinder are shown in Figure 2.18. The minimal displacement of the mean cable position from the centerline of the test section is entirely due to the slight deviations from neutral buoyancy. The flags on each data point in the figure indicate the maximum lateral excursion, which means the mean displacement is not a static divergence effect. The cylinder vibrates at different frequencies. The experimental result deviates from the analytical prediction. Ni and Hansen suggested using the length at which static divergence actually occurred rather than the total length of the cylinder in Païdoussis' analytical model[43] to obtain a better agreement between the analytical model and the experiment[36].

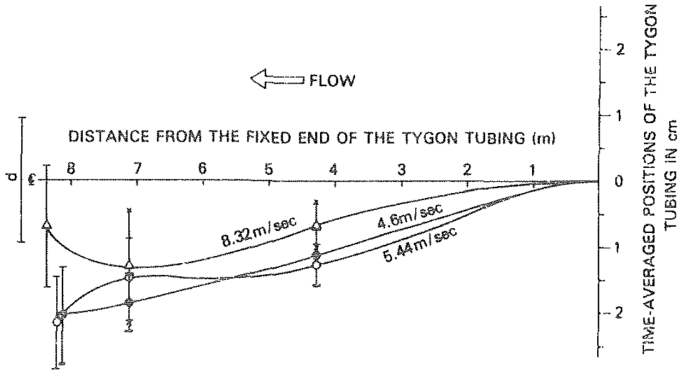


Figure 2.18: Time-averaged cylinder position in the test section at three velocities. Note the greatly expanded vertical scale compared to the horizontal scale[36].

Païdoussis also experimentally tested the instability of a pinned-free cylinder in axial flow[44], as shown in Figure 2.19. Only one set of parameters has been tested. The motion occurred in two ways: rotation about the pin or bending the steel strip. The flow field was thin, and the cylinder deflection was slight. The result validates the analytical model mentioned above for a pinned-free cylinder[22].

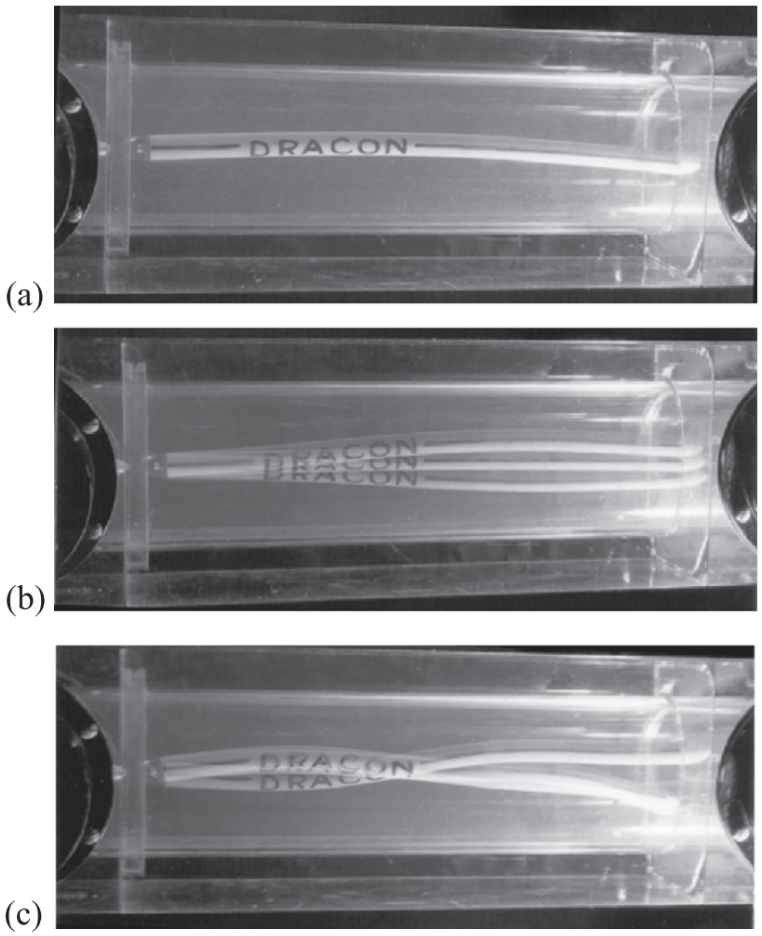


Figure 2.19: Photographs illustrating (a) static divergence (yawing/buckling), (b) second-mode flutter, and (c) third-mode flutter of a horizontal pinned-free cylinder in axial flow[44].

In conclusion, the experiments mentioned above qualitatively validate analytical models, which means that some assumptions in analytical models can be applied to this project.

The cylinder's displacement in axial flow is small, only about once or twice the cylinder's diameter or less, regardless of length. Thus, researchers can investigate the cylinder's instability using a thin fluid field. The results of the experiments also prove the feasibility of the small deflection assumption in the beam model.

The experimental research method usually has the least model error. On the other hand, it has disadvantages such as high cost and time-consuming.

2.2.3. Numerical simulation

Numerical methods can be used to investigate the instability of a flexible cylinder in axial flow. Liu $et\ al.$ simulated the fluid-structure interaction for an elastic clamped-clamped (upstream and downstream ends are fixed) cylinder subjected to tubular fluid flow. They applied the ALE Navier-Stokes equations with large eddy simulation to model the turbulent flow, and the Euler-Bernoulli beam dynamic equation was solved for the elastic cylinder vibration. The numerical method they used for the solid mechanism was the finite element method (FEM)[32]. De Ridder $et\ al.$ simulated the fluid forces and fluid-elastic instabilities of a clamped-clamped cylinder in turbulent axial flow[10]. LaBarbera ran a two-dimensional numerical simulation to test the instability of a short cylinder with a streamlined end[24].

The numerical simulation has several advantages, including low cost and a relatively low model error. Changing and testing various parameters in a numerical model is convenient, while an analytical model usually needs modification when parameters change.

Based on a review of the dynamics of cylindrical structures in axial flow by Païdoussis in 2021[41], the three-dimensional numerical research about a pinned-free or clamped-free cylinder was rare. These two sets of boundary conditions share some similarities with the array cable. Thus, creating a general numerical model for the cylinder mentioned above is meaningful, which makes it easier to predict the instability of an array cable and for researchers to test various conditions.

2.3. Numerical tools

Some numerical methods could be helpful. Introductions are provided in this section.

2.3.1. B-spline Curve

The B-spline curve is defined by a knot vector Ξ and basis functions. The knot vector is a set of non-decreasing real numbers representing coordinates in the parametric space of the curve:

$$\Xi = \{\zeta_1, ..., \zeta_{n+p+1}\}, \tag{2.10}$$

where P is the order of the B-spline, and n is the number of basis functions (and control points) necessary to describe it. ζ is the parametric coordinate along the whole curve. Knot spans subdivide the domain into "elements"[8].

Based on a knot vector, B-spline basis functions are defined recursively starting with $P = 0$ (piecewise constants)[8]:

$$N_{i,0}(\zeta) = \begin{cases} 1 & \zeta_i \leq \zeta \leq \zeta_{i+1} \\ 0 & otherwise. \end{cases} \tag{2.11}$$

For $p > 1$:

$$N_{i,p}(\zeta) = \frac{\zeta - \zeta_i}{\zeta_{i+p} - \zeta_i} N_{i,p-1}(\zeta) + \frac{\zeta_{i+p+1} - \zeta}{\zeta_{i+p+1} - \zeta i + 1} N_{i+1,p-1}(\zeta). \tag{2.12}$$

Given a set of n control points, the piecewise polynomial B-spline curve $\mathbf{C}(\zeta)$ of order P is obtained by taking a linear combination of basis functions and control points:

$$\mathbf{C}(\zeta) = \sum_{i=1}^{n} N_{i,P}(\zeta)\mathbf{B}_i, \tag{2.13}$$

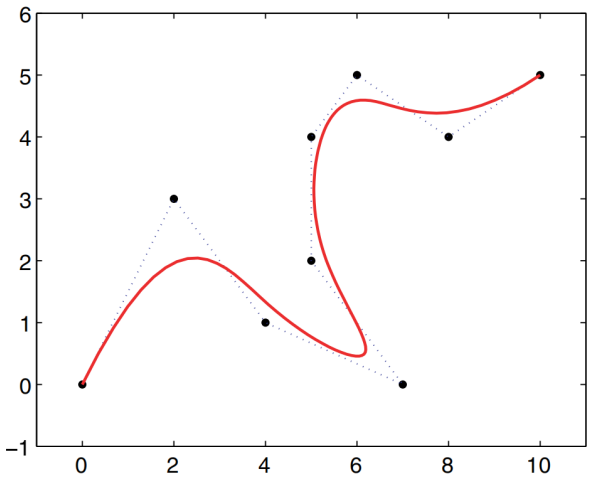


Figure 2.20: Piecewise cubic B-Spline curve (solid line) and its control net (dotted)[8].

An example curve is shown in Figure 2.20. It is a cubic B-spline curve. The basis function is a cubic polynomial, which requires four control points to define. The curve generation process is as follows: pick the first four control points $\mathbf{B}_1$ to $\mathbf{B}_4$ with basis functions $N_{1,3}(\zeta)$ to $N_{4,3}(\zeta)$ to generate the first piece of the curve; use control points $\mathbf{B}_2$ to $\mathbf{B}_5$ with $N_{2,3}(\zeta)$ to $N_{5,3}(\zeta)$ to generate the second piece of the curve; the rest of curve piece follow the same rule until the last piece reach the final control point. Hence, nine control points will generate five segments of the curve. Connect all the segments, and then we can have the whole curve. There is an overlap region between adjacent curve segments, which is why the continuity is high: the second curve segment considers the shape information from the previous segment by sharing several control points.

The control points are usually not on the curve, as shown in Figure 2.20. An essential property of the B-spline curve is affine covariance, which states that an affine transformation of the curve is obtained by applying the transformation to its control points[8].

In our case, the flexible cylinder's displacement is small, with several orders of unstable modes. The cylinder shape is continuous and smooth. Thus, the B-spline curve is sufficient for defining the shape.

2.3.2. Finite element analysis

The finite element analysis determines the cylinder's displacement under the hydro-pressure load. In FEA, the finite element method (FEM) is first applied to discretize the space in contiguous elements. Differential equations based on the material's mechanism are used for each element individually. After the space discretization, the results from FEM are interpolated, and a system with linear algebraic equations can be constructed and solved[1][18][53]:

$$\mathbf{K}_{ij}\mathbf{u}_j = \mathbf{f}_i \ , \tag{2.14}$$

where **u** and **f** are the displacements and loads, which are externally applied at the nodal points. The equations can be written in a matrix form, as shown in Figure 2.21.

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{Bmatrix}$$

Figure 2.21: Global load-displacement matrix equation. u_i and f_j indicate the deflection at the ith node and the force at the jth node[53].

For example, consider an element with two nodes on the nonlinear Euler-Bernoulli beam in the two-dimensional plane xoy. The rotation angle is the derivative of the displacement, and the torque in the axial direction is neglected, so the governing equations only have force and displacement terms without rotation angle and moment. Each node has two degrees of freedom.

Assume the number of nodes on the beam is N, and the x axis is the beam's original central axis. The node values are determined by integrating the governing equation along elements. The dimension of the global stiffness matrix is $2N \times 2N$. The stiffness matrix is sparse.

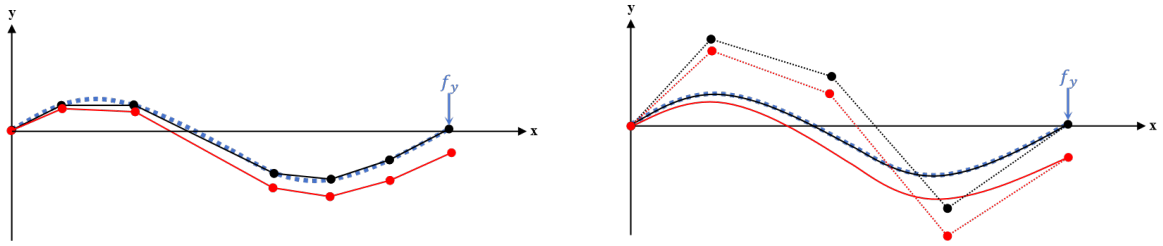
FEA can solve problems with complicated stress[53]. However, "in classical FEA, the basis chosen to approximate the unknown solution fields is then used to approximate known geometry"[7]. The interpolation function between nodes is linear, so a beam element is always straight. A large number of elements are required to represent an original bent curve. The known geometry information is a waste.

2.3.3. Isogeometric analysis

The isogeometric analysis turns the idea of FEA around, selects a basis capable of exactly representing the known geometry, and uses it for the fields we wish to approximate[7].

"A fundamental tenet of isogeometric analysis is to represent geometry as accurately as possible"[8]. For the qualitative illustration, consider a wavelike cantilever beam under a single force applied to the

tip in two-dimensional space. Assume it is a pure-bending linear Euler-Bernoulli beam. The shape function of the beam is known.



(a) Displacement determined by classical FEA. The straight lines with dots are elements in the classical FEA.

(b) Displacement determined by IGA. The dots are control points in the B-spline curve. The tiny dashed line represents the control net.

Figure 2.22: Schematic of a wavelike cantilever beam displacement by different methods. Black represents the original position. Red represents the displacement. f_y is the load. The Blue dashed line is the actual shape of the beam.

As shown in Figure 2.22a, the elements in the classical FEA are straight. The reason is that the displacement between two nodes in an element is determined by applying a piecewise linear interpolation function with the values on nodes in the local coordinate system[18]. Thus, more elements are required to describe a known wavelike beam. There are six elements and seven nodes in the figure; considering only one degree of freedom, the dimension of the stiffness matrix of the whole system is 7×7. A system of seven linear algebraic equations must be solved to determine the displacement, which requires a higher computational cost. The error is also more significant due to the straight element, as shown in the figure. Although more elements are used for refinement, the shape is inaccurate.

IGA can use the known geometry information, as shown in Figure 2.22b. It is possible to generate wavelike elements using the known shape function before applying the force. The significant improvement of IGA is that it applies a nonlinear interpolation function based on the known geometry within an element, allowing the element to have a complex shape that fits the actual shape better. Hence, fewer elements can accurately describe the beam. The figure contains five control points corresponding to two third-degree elements in IGA. The stiffness matrix's dimension is 5×5, less than the dimension of classical FEA. The error is also minor. In a linear system, the determination process is similar to generating and bending wavelike elements. In other words, the method determines the displacement of a straight cantilever beam (reference space), distributes it to the scaled wavelike beam (parametric space), and finally transfers the result to the actual physical space.

The mathematics of IGA is briefly introduced[27].

To apply IGA, knot spans subdivide the domain into "elements"[8]. We define the element as a "curve segment" mentioned in the subsection 2.3.2 in the B-spline curve. If the degree is three, four control points define a curve segment.

We need to start from the FEA principle to illustrate IGA. Consider a cantilever beam under hydropressure force in the y direction in the physical space, which is denoted by q_y. The physical longitudinal coordinate x ranges from zero to one, representing the beginning and end of an element I. Assume the beam is a pure bending, isotropic, linear Euler-Bernoulli beam. Integrate the static governing equation along the element[7], and we have:

$$\int_0^1 EI_y \frac{\partial^4 w_y(x)}{\partial x^4} dx = \int_0^1 q_y(x) dx \ , \tag{2.15}$$

where $w_y(x)$ is the displacement and $q_y(x)$ is the distribution load. The boundary conditions are:

$$w_y(0) = 0 \ \ and \ \ \frac{\partial w_y(0)}{\partial x} = 0 \ . \tag{2.16}$$

Define a test function: $m(x)$, which fulfill the boundary conditions[62]:

$$m(0) = 0 \ \ and \ \ \frac{\partial m(0)}{\partial x} = 0 \ . \tag{2.17}$$

Hence, $\partial^2 m(0)/\partial x^2 = 0$. The test function also has to fulfill the following:

$$\int_0^1 m(x) \cdot R(\widetilde{w_y}) dx = 0 \ , \tag{2.18}$$

where $R(\widetilde{w_y})$ is the residual between approximated solution $\widetilde{w_y}$ and exact solution w_y[18]:

$$R(\widetilde{w_y}) = EI_y \frac{\partial^4 \widetilde{w_y}}{\partial x^4} - q_y \ , \tag{2.19}$$

Multiply the test function on each side of equation 2.15:

$$EI_y \int_0^1 m(x) \frac{\partial^4 w_y(x)}{\partial x^4} d\zeta = \int_0^1 m(x) q_y(x) dx \ . \tag{2.20}$$

The left side of equation 2.20 can be expressed as:

$$EI_y \int_0^1 m(x) \frac{\partial^4 w_y(x)}{\partial x^4} dx = EI_y \left[\frac{\partial^2}{\partial x^2} \left(m(x) \frac{\partial^2 w_y(x)}{\partial x^2} \right) \right]_0^1 - EI_y \int_0^1 \frac{\partial^2 m(x)}{\partial x^2} \frac{\partial^2 w_y(x)}{x^2} dx \tag{2.21}$$

The first term on the right-hand side of equation 2.21 is zero because[49]:

$$m(0) = 0 \ \ and \ \ \frac{\partial^2 w_y(1)}{\partial x^2} = 0 \tag{2.22}$$

Substitute simplified equation 2.21 in equation 2.20 to obtain the weak form of PDE[62]:

$$-EI_y \int_0^1 \frac{\partial^2 m(x)}{\partial x^2} \frac{\partial^2 w_y(x)}{x^2} dx = \int_0^1 m(x) q_y(x) dx \ . \tag{2.23}$$

Now, we can apply the Galerkin method to the equation[18]. Express original variables by known basis functions and unknown control variables:

$$w_y(x) = \sum_{i=1}^{n_I} N_i(x) W_i \ , \ m(x) = \sum_{i=1}^{n_I} N_i(x) M_i \ . \tag{2.24}$$

Use N_i as the abbreviation of basis functions. The equation 2.23 becomes:

$$-EI_y \int_0^1 \frac{\partial^2}{\partial x^2} \left(\sum_{i=1}^{n_I} N_i M_i \right) \frac{\partial^2}{\partial x^2} \left(\sum_{i=1}^{n_I} N_i W_i \right) dx = \int_0^1 \left(\sum_{i=1}^{n_I} N_i M_i \right) q_y(x) dx \ . \tag{2.25}$$

Here, we can apply IGA. n_I is the number of control variables in an element, N_i is the basis function from the B-spline curve, M_i is the control variable of the test function, and W_i is the displacement of the control point that defines the curve shape. Divide M_i on both sides of equation 2.25 since the equation holds for any M_i. Extract the control variable out of the integration:

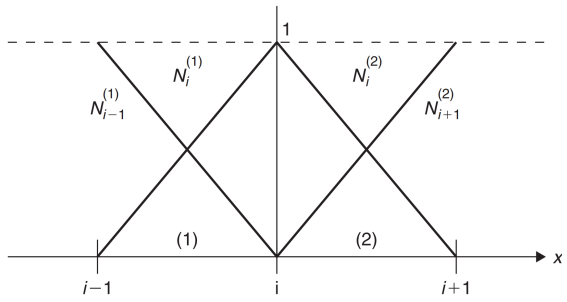
$$-EI_y \sum_{i=1}^{n_I} W_i \int_0^1 \frac{\partial^2 N_i}{\partial x^2} \frac{\partial^2 N_i}{\partial x^2} dx = \sum_{i=1}^{n_I} \int_0^1 N_i q_y(x) dx \ . \tag{2.26}$$

The second derivative of the basis function N_i exists and is known because the degree of basis functions is at least cubic in IGA, as shown in Figure 2.23b. Thus, we can integrate the second derivative instead of the fourth. The PDE is simplified, making the integral equation more straightforward to solve.

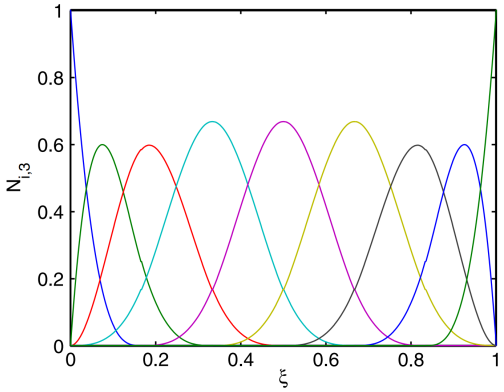
The interpolation function in the classical FEA is different. N_i is linear, as shown in Figure 2.23a. w_y becomes[18]:

$$w_y(x) = \sum_{i=1}^{2} N_i(x) w_{y,i} \ , \ N_1(\xi) = 1 - \xi \ , \ N_2(\xi) = \xi \ , \ \xi = \frac{x - x_{i-1}}{\Delta x_i} \ . \tag{2.27}$$

Simplifying PDE as equation 2.26 before integrating is impossible since the basis functions are linear: the derivatives become zero from the second one. We can only start integrating PDE from the fourth derivative, which is more complex. The geometry information is wholly wasted.



(a) Linear one-dimensional element and shape functions. (1)(2) denote the element. i denotes the node[18]. The vertical axis is the value of the basis function.



(b) Cubic basis functions formed from the open knot vector $\Xi = \{0, 0, 0, 0, 1/6, 1/3, 1/2, 2/3, 5/6, 1, 1, 1.\ \zeta$ is the parametric coordinate in an element[8].

Figure 2.23: Basis functions as components of the interpolation function. Superpose basis functions to obtain the shape function or the displacement function.

In conclusion, IGA is applicable for the solid structure solver in our case because (1) the analytical signed-distance function of the cylinder can be obtained from WaterLily, the geometry information is apparent; (2) the cylinder, in our case, is a linear Euler-Bernoulli beam, the deflection is slight and smooth; (3) the external force is continuous and smooth; (4) IGA has higher efficiency due to fewer elements; (5) IGA define the curve better since the basis functions are nonlinear; (6) IGA is robust for oscillatory behavior of the solutions[62].

The element load-displacement matrix equation can be constructed by integrating terms on both sides of equation 2.26 and its similar form in the z direction along an element. The integration is numerically approximated by Gaussian quadrature. We can assemble the global stiffness matrix with the local stiffness matrix[7].

2.3.4. Gauss quadrature

Integrating the differential governing equation as 2.26 along the beam by the analytical method with two integral limits could be complex. The analytical function of the distribution load is also unknown. Measuring the unit length force on Gauss points is the only way to approximate the integration of external forces. Hence, the code uses the numerical method of approximating the integral as a weighted sum[61]. The typical integration domain in the Gauss quadrature is $[-1, 1]$. We can apply a map to transfer the domain to the physical one. For example, consider the integration of the force term in the y direction in equation 2.26:

$$\int_0^1 N_i q_y(x) dx = \sum_{j=1}^{n_g} G_j N_{i,j} q_{y,j} \ . \tag{2.28}$$

Here, i denotes the control variable, and j denotes the Gauss integration point. n_g is the number of the Gauss point, $q_{y,j}$ is the unit length force on the beam's Gauss point, and G_j is the pre-defined Gauss weight. $N_{i,j}$ is the value of the basis function i on the Gauss point j.

Apply Gauss integration to the left side of equation 2.26, and we have:

$$\int_0^1 \frac{\partial^2 N_i}{\partial \zeta^2} \frac{\partial^2 N_i}{\partial \zeta^2} d\zeta = \sum_{j=1}^{n_g} G_j \frac{\partial^2 N_{i,j}}{\partial \zeta^2} \frac{\partial^2 N_{i,j}}{\partial \zeta^2} \ . \tag{2.29}$$

2.3.5. Finite volume method

Computational fluid dynamics has developed dramatically in this century. Numerical simulation has become a popular method for solving fluid problems. It is highly efficient because it does not require actual customized equipment and uses computers to solve complex equations. Running numerical simulations could result in a low model error as their principle is to simulate experiments in digital environments.

The Finite volume method is one of the known numerical methods for solving fluid dynamics problems. It has an advantage due to its direct connection to physical flow properties. The basis of the method relies on the direct discretization of the integral form of the conservation law. The FVM requires setting up the following steps[18].

- Subdivide the mesh, obtained from the space discretization, into finite (small) volumes, one control volume being associated with each mesh point;
- Apply the integral conservation law to each of these finite volumes.

It is possible to use FVM to solve the FSI problem. Slone $et\ al.$[54] and Chijioke $et\ al.$[6] provide examples of the usage of FVM. OpenFOAM and Waterlily are two known open-source CFD codes based on the finite volume method (FVM). Compared to OpenFOAM, Waterlily has advantages: it is more efficient and can be combined with machine learning[63].

It is theoretically possible to solve a fluid dynamic problem with a large geometry scale by using the finite volume method. However, computational cost is a firm limit for FVM. For the problem with large geometry, the flow solver must solve a considerable number of meshes. In our case, the cylinder (cable) length is 100 meters; a full-scale simulation may require several months unless a supercomputer is used, which is not accessible in this project.

Due to the limit of computational cost, researchers usually run their simulations at a model scale while maintaining flow similarity. A typical way to achieve flow similarity for a structure immersed in the fluid is to keep the Reynolds number of the models the same as that of the real structures. In our case, the typical diameter of the structure is $D_{arr} = 0.03$ meters. The flow speed is $U_{real} = 5\ knots$ (i.e., $2.572\ m/s$). The density and dynamic viscosity of seawater are $\rho_{sea} = 1026\ kg/m^3$ and $\mu_{sea} = 1.22 \times 10^{-3}\ Pa \cdot s$. The Reynolds number of the structure can be determined by equation 2.30.

$$Re = \frac{\rho_{sea} U_{real} D_{arr}}{\mu_{sea}} \tag{2.30}$$

The Reynolds number is 6.49×10^4, and the Reynolds number of the related model should be the same. If we use the structure length L_{arr}, the value will reach 2.16×10^8. The required fluid field should also be enormous to contain a full-size model due to the aspect ratio. Many grids are necessary for the numerical simulation with the actual aspect ratio.

2.3.6. Boundary Data Immersion Method

The open-source code WaterLily was developed purely by Julia. It is a powerful tool because it is convenient to create and import expansion packages to the mainframe. The FVM code Waterlily uses the boundary data immersion method (BDIM) to deal with the body boundary immersed in the fluid. BDIM is a type of immersion boundary method. The basic principles of the immersion boundary method and reasons why that is unsuitable for high Reynolds flow are given below.

For an FSI problem, use Ω_f to denote the incompressible viscous fluid domain and Ω_b to denote the solid or deforming body domain with prescribed velocity $\vec{V}(\vec{x},t)$[33]. The governing equation in the solid body is

$$\vec{u} = \vec{V} \ . \tag{2.31}$$

The incompressible Navier-Stokes equation governs the fluid:

$$\frac{\partial \vec{u}}{\partial t} + \left(\vec{u}\cdot\vec{\nabla}\right)\vec{u} + \frac{1}{\rho}\vec{\nabla}p - \nu\nabla^2\vec{u} = 0 \ . \tag{2.32}$$

The coefficients ρ and μ are the constant fluid density and viscosity, respectively. Integrate the equation 2.32 over a time step Δt, the fluid and body equations can be written as:

$$\begin{cases} \vec{u} = \vec{b} \ , & for\ \vec{x} \in \Omega_b \\ \vec{u} = \vec{f}(\vec{u}) \ , & for\ \vec{x} \in \Omega_f \end{cases} \tag{2.33}$$

with

$$\vec{b} = \vec{V} \ , \tag{2.34}$$

$$\vec{f}(\vec{u}, t_0+\Delta t) = \vec{u}(t_0) + \int_{t_0}^{t_0+\Delta t} \left[-\left(\vec{u}\cdot\vec{\nabla}\right)\vec{u} + \nu\nabla^2\vec{u}\right]\, dt - \int_{t_0}^{t_0+\Delta t} \frac{1}{\rho}\vec{\nabla}p\, dt \ . \tag{2.35}$$

The second integration in the right-hand side of the equation 2.35 is the non-pressure terms, and the last integration is the pressure impulse over Δt[33].

In an immersion boundary method, equation 2.32 will be discretized on a non-boundary conforming Cartesian grid, and the boundary condition will be imposed indirectly through modifications of equation 2.33. Grid generation is much easier using the immersion boundary method because a body does not necessarily have to fit and conform to a Cartesian grid[16].

There are two categories of immersion boundary methods. The first category is the continuous forcing approach. A significant drawback of this approach is that the smoothing of the forcing function leads to an inability to provide a sharp representation of the immersed boundary. The methods in this category are not helpful for high Reynolds number flows. Besides, continuous approaches require solving the governing equations within the immersed body. With increasing Reynolds numbers, the proportion of grid points inside the immersed boundary also increases[16], which requires more computational cost.

The second category is the direct forcing approach. This approach is better suited for higher Reynolds numbers because it imposes the velocity boundary conditions at the immersed boundary[16]. However, the challenge of high Reynolds number flows remains. The higher the Reynolds number, the larger the jump in the velocity derivative, which exacerbates this problem and requires special techniques for accurate simulation[33].

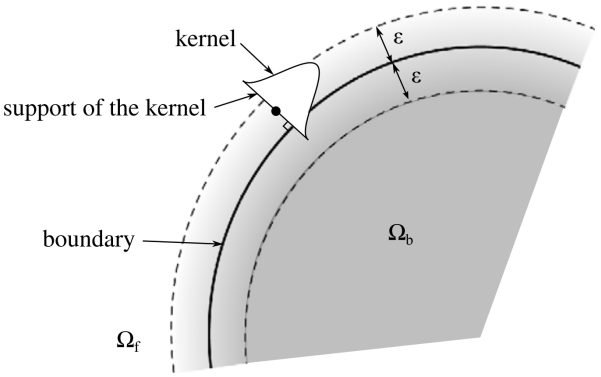


Figure 2.24: Smoothing across the immersed boundary. The equations valid in each domain are convolved with a radius ϵ kernel and added together. The kernel at a point (marked by a dot) that belongs to the boundary region is represented[33].

The BDIM is a new, robust, and accurate Cartesian-grid treatment for the immersion of solid bodies within a fluid with general boundary conditions. It is ”derived based on a general integration kernel formulation which allows the field equations of each domain and the interfacial conditions to be combined analytically”[64]. The kernel is shown in Figure 2.24. The kernel smoothly distributes the value on the physical boundary to a region with a thickness of 2ϵ.

A paper published in 2014 developed the second-order BDIM. The authors modified the original method, which makes it more suitable for high Reynolds number flows. The paper points out that the computational cost limits the application of second-order BDIM to $Re \leq 10^5$[33]. Even considering the hardware advancements nowadays, using BDIM to solve the flow with a Reynolds number of 10^8 is impossible.

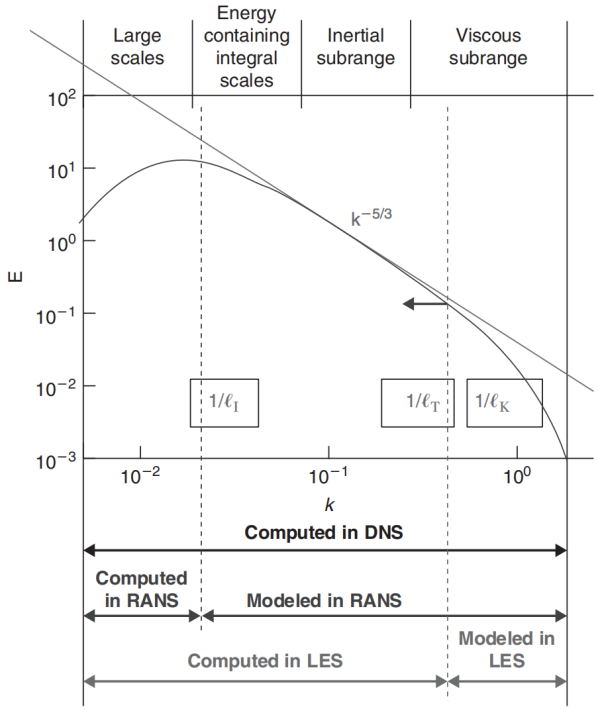


Figure 2.25: Energy spectrum of turbulence in the function of wave number k, indicating the range of application of the DNS, LES, and RANS models. The length scales l_T and l_I are associated with the LES and RANS approximations[18][14].

Besides the drawback of immersion boundary methods, large eddy simulation (LES) for determining turbulent flows in Waterlily also limits the applicable Reynolds number. LES is an intermediate technique between directly simulating turbulent flows and solving Reynolds-averaged equations[47]. In LES, the contribution of the large, energy-carrying structures to momentum and energy transfer is computed exactly, and only the effect of the smallest scales of turbulence is modeled, as shown in Figure 2.25. The need for grid resolution close to solid walls is the main limitation to applying LES to high-Reynolds flows[20]. The use of LES brings additional requirements in terms of computational cost.

In summary, we cannot simulate the full-scale array cable of our case even by Waterlily directly.

2.3.7. FSI method

The instability of a flexible cylinder in axial flow is an FSI problem. The flow field applies loads to the cylinder, including hydrodynamic and hydrostatic forces. The cylinder deforms and changes the flow field simultaneously. The coupling effect must be determined. This subsection discusses different approaches to connecting the flow solver and the solid model.

There are two approaches to solving the interaction between the fluid and structure: the monolithic and partitioned approaches. In a monolithic method, fluid-structure interaction at the mutual interface is treated synchronously. Monolithic schemes appear unconditionally stable and considerably more

accurate than the partitioned method. However, monolithic schemes remain computationally more expensive per time step[34].

In a partitioned method, the fluid and the structure equations are alternately integrated in time, and the interface conditions are enforced asynchronously. Typically, partitioned methods are based on the following sequential process[34]:

- transfer the motion of the structural boundary to the fluid;
- update the position of the moving fluid mesh;
- advance the fluid system in time and compute the new pressure;
- convert the new fluid pressure into a structural load;
- advance the structural system in time under the fluid-induced load.

Partitioned methods require less computational cost because they require only one fluid and structure solution per time step, which can be considered a single fluid-structure interaction. The major drawback of partitioned methods is the loss of conservation properties of the continuum fluid-structure system due to the time lag between the time integration of fluid and structure. The error of partitioned methods is more significant than that of a monolithic method[34].

Considering the limited computational cost and time required for this project, the partitioned method is selected for the investigation. In a partitioned scheme, two methods exist for solving the time integration, explicit and implicit.

An explicit time integration has no approximation of the interface Jacobian. The resulting system can be simplified to the sequential evaluation of a fluid operator F, which, given a structural deformation $\mathbf{d}_\Gamma$, updates the fluid variables and computes the fluid traction vector λ_f and a similar operator S. The solution to the coupled problem is then reduced to the evaluation of[26]

$$\begin{aligned} \lambda_f^{n+1} &= F(\mathbf{d}_\Gamma^{n+1}), \\ \mathbf{d}_\Gamma^{n+1} &= S(\lambda_f^{n+1}). \end{aligned} \tag{2.36}$$

The subscript f represents the flow field, and Γ represents the fluid-structure interface. Because the fluid is computed at $t^{n+1} = t^n + \Delta t$ with the structural position at t^n, the kinematic boundary condition is never adequately enforced at the end of each time step[26]. Energy conservation is not guaranteed[48]. The energy balance at the interface is more substantial with small mass ratios[5], which is the condition in our case, making the explicit method unsuitable for our FSI system.

The implicit methods eliminate the energy error due to the lack of enforcement of the boundary condition by imposing the Dirichlet condition (no-slip and penetration) and the Neumann condition (stress continuity) at the fluid-structure interface at the end of every time step. This results in a fixed-point operator for the interface displacement of the fluid-structure coupled system[26]

$$H(\mathbf{d}_\Gamma^k) = S \circ F(\mathbf{d}_\Gamma^k) = \tilde{\mathbf{d}}_\Gamma^k, \tag{2.37}$$

where the notation $S \circ F$ indicates that the result of the function F is given as an argument to the function S. Finding the converged solution to the Dirichlet-Neumann coupling procedure is thus equivalent to finding the fixed point of the FSI problem[26]

$$\mathbf{d}_\Gamma^* = H(\mathbf{d}_\Gamma^*). \tag{2.38}$$

Compared to explicit methods, implicit methods are more suitable for our case.

Interface Quasi-Newton (IQN) methods are methods to solve the aforementioned FSI problem. They have theoretical fast convergence rates and the reduced cost associated with the interface Jacobian. The methods avoid difficulties due to the solution to the approximate tangent problem of Newton-Raphson methods[26]. The difference between the classical and the IQN methods is briefly introduced.

Take the displacement vector $\mathbf{d}_k$ as an example. k denotes the iteration step. In the classical Newton method, we have:

$$\mathbf{d}_{k+1} = \mathbf{d}_k + \Delta\mathbf{d}_k = \mathbf{d}_k - J_k \backslash R(\mathbf{d}_k) \ . \tag{2.39}$$

$J_k = \frac{\partial R}{\partial \mathbf{d}}|_k$ is the $m \times m$ Jacobian of the residual operator at step k, m is the number of degrees of freedom of the displacement on the interface. The IQN method uses a sequence of $n \ll m$ examples of previous displacement updates and residuals:

$$\mathbf{W}_k = [\Delta\mathbf{d}_k, \Delta\mathbf{d}_{k-1}, ..., \Delta\mathbf{d}_{k-n+1}] \ , \ \mathbf{V}_k = [R(\mathbf{d}_k), R(\mathbf{d}_{k-1}), ..., R(\mathbf{d}_{k-n+1})] \ . \tag{2.40}$$

Then, we approximate the Jacobian as:

$$\Delta\mathbf{d}_k = -J_k \backslash R(\mathbf{d}_k) \approx \mathbf{W}_k(\mathbf{V}_k^T\mathbf{V}_k)\backslash\mathbf{V}_k^T R(\mathbf{d}_k) \ . \tag{2.41}$$

The new equation is solved with an $n \times n$ matrix, which is much easier than using the Jacobian matrix[28].

2.4. Research matrix

Previous research indicates that a cylinder's instability in axial flow is related to the boundary conditions of the upstream and downstream ends, the cylinder's length (aspect ratio), and the properties of the flow field. However, although experiments tested those parameters, the effect of a single parameter was unclear because multiple parameters were changed in different experiments.

According to the reasons mentioned above, this thesis project will test the neutral buoyant flexible cylinder with various parameters in axial flow:

- **Shape of the downstream end**: Previous research indicates that the downstream end shape plays a vital role in the instability. Whether some shapes that have never been tested before can stabilize or destabilize the cylinder remains to be explored. Various cylinder shapes will be tested.
- **Boundary condition of the upstream end**: The test cases with pinned-free and clamped-free cylinders will be compared to investigate the effect of the upstream boundary condition.
- **Length of the cylinder**: This project will test various cylinder lengths (aspect ratios) to investigate how the instability changes. The aspect ratio of an actual array cable is more than 3000. It is impossible to run a real or numerical experiment. However, improving the existing analytical model and then predicting the instability of the typical array cable using the new model is possible. We can also observe the instability transition as the cylinder length changes.
- **Reynolds number**: Although the viscous load on the cylinder is negligible due to the high Reynolds number, reducing the Reynolds number in the viscosity exemption range could alter the instability by changing the flow field. A pinned-free cylinder with various Reynolds numbers will be tested in flow fields.

Chapters 3, 4, and 5 will provide the research details. This project aims to predict the cylinder's instabilities under various conditions. The precise amplitude of the buckling or flutter will not be considered because the unstable mode of an array cable is relatively useless in reality. The tendency of the altered amplitude as parameters change is considered.

2.5. Research approach determination

Considering the cost and model error, this project will use the **numerical simulation** to investigate the instability of a solitary cylinder in axial flow. Using CFD code to test cases with many variables is convenient and cost-saving. The Julia code to solve a two-dimensional FSI problem is available. The solid structure solver, FSI solver, and parametric body codes should be modified and validated first to suit the three-dimensional research problem.

The CFD code, WaterLily with BDIM and LES, is selected to solve the flow field. Weymouth $et\ al.$[64][63] has validated this flow solver.

The IGA with the linear Euler-Bernoulli beam model is selected to solve the deformation of the cylinder. The feasibility of IGA is proved by Verhelst and Cottrell $et\ al.$[62][8].

A partitioned approach, an implicit method, and the IQN method are selected to determine the FSI effect. Lauber shows that the method applies to an FSI problem with a flexible structure[26].

2.6. Contribution

An open-source code to numerically predict the instability of a solitary cylinder with various parameters in three-dimensional axial flow has been developed and validated. The new code is modified from an existing code for a two-dimensional FSI problem. Based on the numerical result, the instability of a typical array cable is predicted. The effects of cylinder length, boundary conditions, and the fluid field Reynolds number on the instability have been investigated. A thesis report has been completed to introduce the whole research. Instability predictions and suggestions are provided to the company Optics11. The code will be uploaded for general testing and use. The results of this project are sufficient to launch a conference paper. **Appendix A** provides the outline of the paper. The author plans to finish the paper soon.

3

Solid structure solver

The hydraulic force is applied to the flexible cylinder. The solid structure solver determines the cylinder deformation based on the load. This chapter introduces the simplified solid structure model, the solver's methodology, and its validation.

3.1. Boundary conditions

A clamped-free cylinder validates the solid structure solver and FSI solver. The boundary conditions are fixed upstream and free downstream. Hence, we apply the Neumann and Dirichlet conditions on the cylinder's upstream end. The displacement and rotational angle are constant zero[2]. There is no Neumann condition at the pinned-free cylinder.

3.2. Beam model

The flexible cylinder is neutrally buoyant, so the force of gravity is negligible. Its material is considered even and isotropic, with no self-damping.

A cylinder in axial flow is subject to hydro-pressure and viscous forces. The Reynolds number of the actual array cable is determined by equation 2.30. Substitute the typical working conditions[55][59]: $\rho_{sea} = 1026\ kg \cdot m^3$, $U_{real} = 5\ knots$, $\mu_{sea} = 0.00122\ Pa \cdot s$, and $D_{arr} = 0.03\ m$; so $Re = 6.49 \times 10^4$.

$$Re = \frac{\rho_{sea} U_{real} D_{arr}}{\mu_{sea}}$$

We use the diameter to determine the Reynolds number since lateral displacement is mainly of concern, and the lateral force is expected to make a significant contribution. Hence, the diameter should be the characteristic length. The range of Reynolds numbers in previous experiments was $10^4 < Re_{exp} < 10^5$[43], also determined by the diameter. According to the typical Reynolds number, the viscous force is negligible.

Figure 3.1: Lateral pressure force at a point on the cylinder in the global system, where $\mathbf{q}_{sim}$ is the distribution load. The cylinder is simplified to a curve for better illustration. The curve can be regarded as the central axis of the cylinder.

Hydro-pressure can create lateral and axial forces. The lateral force at a point on the cylinder is per-

pendicular to the cylinder's central axis (longitudinal axis). The axial force's direction is the same as the local axis's tangential direction. Notice that if the cylinder is bent, the cylinder axis may not be aligned with the global coordinate axis, as shown in Figure 3.1. According to the characteristics of an actual array cable, the cylinder is inextensible and homogeneous, so the axial force caused by pressure is neglected. In summary, no axial external force is considered.

The literature review allows the small deflection assumption to be applied. The coupling effect between lateral displacements in different directions is neglected. The analytical model[44] does not consider the axial torque on the cylinder, but it can still predict the instability, so the author infers that it can be ignored. No torque on the cylinder is considered in the solid structure solver.

After the simplification above, the flexible cylinder is only subject to the lateral force caused by pressure. Considering the slenderness and continuous hydro-pressure load, shear deformation is neglected. Thus, when the cylinder is bent, its cross-section is further assumed to be rigid and perpendicular to the central axis. The cylinder is only subject to bending moments. In conclusion, the pure bending assumption is applied, followed by the Euler-Bernoulli beam theory[38][3].

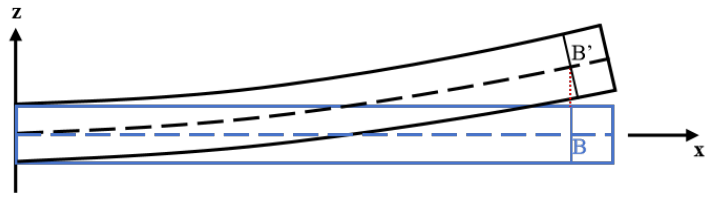


Figure 3.2: Side view of a bending cylinder in the xoz plane. The cylinder is simplified to a curve, and the displacement is amplified for better illustration. Blue represents the original position, while black represents deformation. The long-dashed curve is the central axis of the cylinder. B point is on the axis and has no displacement in the x direction after bending. The cross-section past the point can rotate. The slight axial extension is neglected based on the small deflection assumption.

Notice that the slight deflection and inextensible assumptions indicate that an arbitrary point on the cylinder's central axis only has displacements in the y or z direction (the small extension is neglected), as shown in Figure 3.2. The solid structure solver only determines the displacement of the axis. Then, the cylinder is created based on the axis and its diameter.

The model can be simplified further. Set the x axis of the global system as the origin position (no deformation) of the cylinder's central axis, and place the cylinder's upstream end at the origin of the coordinate system. This setting will be used in later chapters. The lateral force caused by pressure can be decomposed into forces in the directions of x, y, and z. For example, pick the force in the xoz plane, as shown in Figure 3.3. The force in the x direction can only contribute to the displacement in the z direction by creating a bending moment along the y axis to rotate the beam. Thus, we focus on comparing moments.

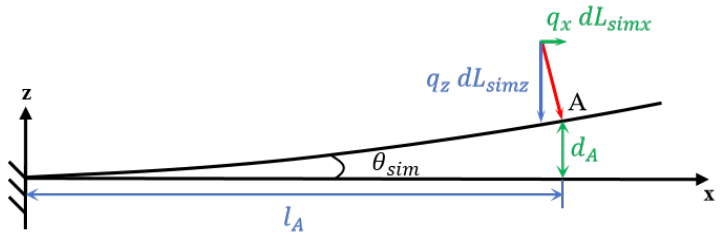


Figure 3.3: Lateral pressure force decomposition in the xoz plane. The cylinder is simplified to a curve, and the displacement is amplified for better illustration.

$q_x dL_{simx}$ is the x-component of the lateral force on point A; $q_z dL_{simz}$ is the z-component. d_A and l_A are levers that correspond to force components. d_A is also the lateral displacement at point A. l_A is also the longitudinal position of point A. θ_{sim} is the rotation angle. Because the deflection is small, there is a relation: $tan(\theta_{sim}) \approx d_A/l_A \sim 1/10$. Then, apply the scaling method:

$$M_x = q_x dL_{simx} dA,\ M_z = q_z dL_{simz} l_A,$$
$$\frac{M_x}{M_z} = \frac{q_x dL_{simx}}{q_z dL_{simz}} \frac{d_A}{l_A} = \frac{d_A}{l_A} tan(\theta_{sim}) = (\frac{d_A}{l_A})^2 \sim \frac{1}{10^2}\ . \tag{3.1}$$

M_x is the bending moment due to the force x component; M_z is the moment due to the z component. The relation 3.1 indicates that the M_x is two orders of magnitude smaller than M_z. A similar relation between M_x and M_y can be obtained in the xoy plane. On the downstream end, the x component of the lateral force could be more prominent due to the streamlined shape. The $tan(\theta_{sim})$ relation may disappear, but we still have $M_x/M_z \sim 1/10$ because of the lever relation. Considering the tiny proportion of the downstream end in the whole cylinder, the force in the x direction is also neglected on the downstream end.

Based on the relations above, the x component of the lateral pressure force can be neglected for a pure bending cylinder with a slight deflection. The solid structure solver will only consider the pressure forces in the direction of z and y.

In summary, the linear differential solid mechanism governing equations of the pure bending cylinder are[38]:

$$\begin{cases} E_{sim} I_{sim} \dfrac{d^4 w_y}{dx^4} = q_y(x), \\ E_{sim} I_{sim} \dfrac{d^4 w_z}{dx^4} = q_z(x). \end{cases} \tag{3.2}$$

The cylinder is regarded as a linear Euler-Bernoulli beam[62]. E_{sim} is Young's modulus used in the numerical simulation, and I_{sim} is the moment of inertia. q_y and q_z are the distribution load in y and z directions. w_y and w_z are local displacements in y and z directions. x is the longitudinal location on the cylinder.

A pressure integrator obtains the external load on the cylinder's central axis from the pressure field. Chapter 5 provides details about the integrator. According to the external load, the displacements are determined separately by the solid structure solver and then substituted into the FSI solver to get the resultant displacement in three-dimensional space.

3.3. Methodology

3.3.1. B-spline curve

The solid structure solver generates the geometry using the B-spline curve. The generated curve, which has no thickness, represents the central axis of the tested cylinder. The Julia code for a B-spline curve in three-dimensional space is available and does not require modification.

We regard $\mathbf{C}(\zeta)$ in equation 2.13 as the curve shape function in the parametric space, in other words, the displacement of a beam's central axis under load. The segment curve mentioned in the subsection 2.3.1 is the element in IGA, which will be discussed in the subsection 3.3.2.

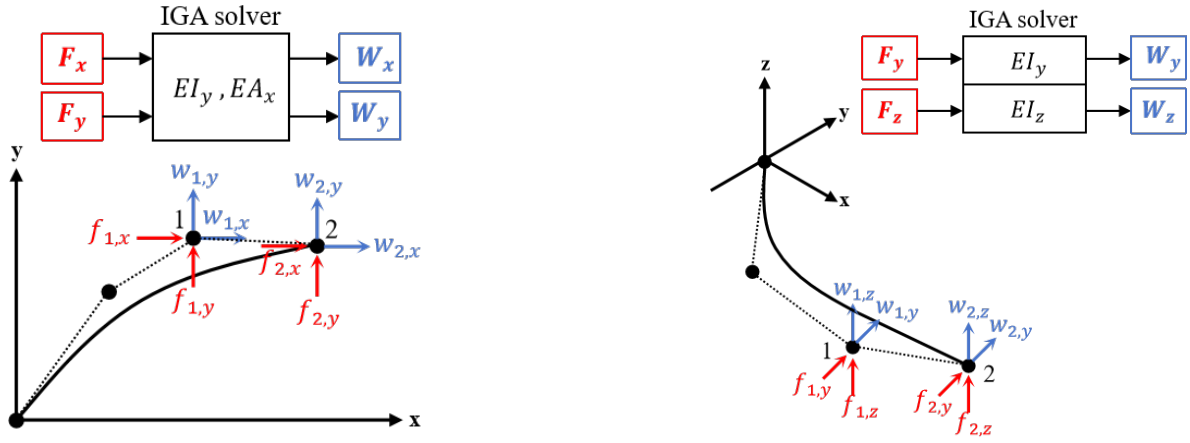
3.3.2. Isogeometric analysis

The solid structure solver uses isogeometric analysis (IGA) with the B-spline curve to improve code efficiency. IGA constructs the load-displacement matrix equation on the control points of the B-spline curve.

An IGA solver is available. However, the original solver is developed to determine a non-linear Euler-Bernoulli beam's two-dimensional displacements in x and y directions. Assume we have n control points. Each control point has two degrees of freedom, so the global stiffness matrix's dimension is $2n \times 2n$.

Although the cylinder, in our case, is three-dimensional, the displacements and external forces are limited in y and z directions due to simplifications. Hence, the dimension of the global stiffness matrix

is still $2n \times 2n$. The original stiffness matrix in the IGA solver has sufficient dimensions to solve the problem with displacements and loads in two directions.



(a) Element in the original IGA solver.

(b) Element in the modified IGA solver.

Figure 3.4: Schematic of an element in different IGA solvers. Simple flow charts of solvers are also provided. The dashed line is the control net. The solid line represents the beam. Black dots represent control points. Blue and red represent displacements and forces.

The existing IGA solver can be applied to our case as a solid structure solver to solve for displacements in the y and z directions by modifying the local stiffness matrix. Then, the global matrix can be constructed. The principle is to replace the governing equations in the x direction with the linear Euler-Bernoulli equations in the z direction and set the off-diagonal blocks to zero.

The difference between the original and modified solvers is shown in Figure 3.4. We can build the integral formulation of our case's IGA element I, refer to equations 2.26, 2.28, and 2.29:

$$\begin{cases} -EI_y \sum_{i=1}^{n_I} W_{y,i} \sum_{j=1}^{n_g} G_j \frac{\partial^2 N_{i,j}}{\partial x^2} \frac{\partial^2 N_{i,j}}{\partial x^2} = \sum_{i=1}^{n_I} \sum_{j=1}^{n_g} G_j N_{i,j} q_{y,j} \ , \\ -EI_z \sum_{i=1}^{n_I} W_{z,i} \sum_{j=1}^{n_g} G_j \frac{\partial^2 N_{i,j}}{\partial x^2} \frac{\partial^2 N_{i,j}}{\partial x^2} = \sum_{i=1}^{n_I} \sum_{j=1}^{n_g} G_j N_{i,j} q_{z,j} \ . \end{cases} \tag{3.3}$$

Equation system 3.3 can be decomposed by linear algebra:

$$\begin{cases} \left(-EI_y \sum_{j=1}^{n_g} G_j \frac{\partial^2 N_{1,j}}{\partial x^2} \frac{\partial^2 N_{1,j}}{\partial x^2} \right) W_{y,1} = \sum_{j=1}^{n_g} G_j N_{1,j} q_{y,j} \ , \\ \vdots \\ \left(-EI_y \sum_{j=1}^{n_g} G_j \frac{\partial^2 N_{n_I,j}}{\partial x^2} \frac{\partial^2 N_{n_I,j}}{\partial x^2} \right) W_{y,n_I} = \sum_{j=1}^{n_g} G_j N_{n_I,j} q_{y,j} \ , \\ \left(-EI_z \sum_{j=1}^{n_g} G_j \frac{\partial^2 N_{1,j}}{\partial x^2} \frac{\partial^2 N_{1,j}}{\partial x^2} \right) W_{z,1} = \sum_{j=1}^{n_g} G_j N_{1,j} q_{z,j} \ , \\ \vdots \\ \left(-EI_z \sum_{j=1}^{n_g} G_j \frac{\partial^2 N_{n_I,j}}{\partial x^2} \frac{\partial^2 N_{n_I,j}}{\partial x^2} \right) W_{z,n_I} = \sum_{j=1}^{n_g} G_j N_{n_I,j} q_{z,j} \ . \end{cases} \tag{3.4}$$

The dimension of the element stiffness matrix is $2n_I \times 2n_I$. With element matrices, a global stiffness matrix can be constructed. Consider a cantilever beam generated by two second-degree elements. Figure 3.5 shows the global stiffness matrix. The matrix is fuller than a global stiffness in the classical FEA because a control point influences other elements more in IGA. However, FEA cannot accurately describe a cantilever beam with only two elements like IGA. For example, a distributed load can be applied to the IGA elements perpendicularly. The displacement error in IGA is tiny ($O(10^{-20})$). Thus, we believe IGA still has higher efficiency.

```
     1.92e6    -480000.0  -240000.0     120000.0     0.0        0.0          0.0        0.0          0.0     0.521782
-480000.0       480000.0  -480000.0     240000.0     0.0        0.0          0.0        0.0          0.0     0.0
-240000.0      -480000.0        1.92e6     -1.32e6   0.0        0.0          0.0        0.0          0.0     0.0
 120000.0       240000.0     -1.32e6   960000.0     0.0        0.0          0.0        0.0          0.0     0.0
      0.0            0.0        0.0          0.0     1.0        0.0          0.0        0.0          0.0     0.0
      0.0            0.0        0.0          0.0     0.0        1.92e6    -480000.0  -240000.0     120000.0     0.0
      0.0            0.0        0.0          0.0     0.0     -480000.0     480000.0  -480000.0     240000.0     0.0
      0.0            0.0        0.0          0.0     0.0     -240000.0    -480000.0     1.92e6     -1.32e6   0.0
      0.0            0.0        0.0          0.0     0.0      120000.0     240000.0    -1.32e6   960000.0     0.0
      0.521782       0.0        0.0          0.0     0.0        0.0          0.0        0.0          0.0     0.0
```

Figure 3.5: Global stiffness matrix of a cantilever beam. $EI = 10000$, $L = 10$. Two ghost columns and rows for boundary conditions are omitted.

In the solid structure solver, the new positions of the control points are first determined to generate the curve under load, which is the flexible cylinder's central axis. Then, displacements of arbitrary points on the axis are measured. Once the central axis' displacement is obtained, the position of a point on the cylinder surface can be determined based on the cylinder's diameter.

The original IGA solver can apply Dirichlet (function value is constant) and Neumann (function's derivative is constant) boundary conditions[2] to the curve beginning and end. It can also solve the dynamic system using the GeneralizedAlpha method, which requires the mass ratio between the fluid and the solid structure. Once the modification of the global stiffness matrix has been applied, with correct forces in y and z directions, displacements can be solved.

3.3.3. Gauss integration

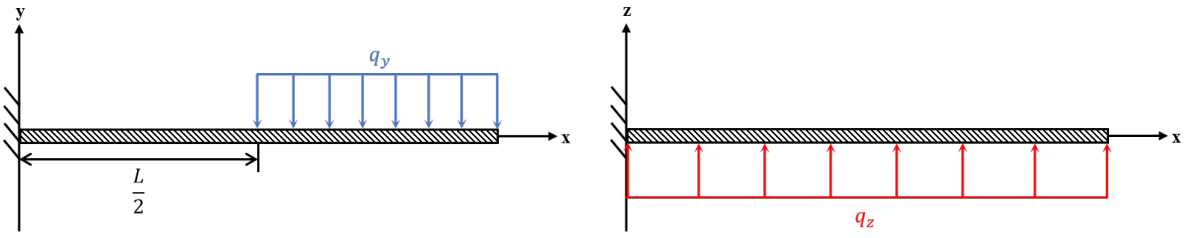
The Gauss integration method, also known as Gauss quadrature, has been applied to the code to determine the stiffness matrix and the external force vector as in equation 3.4. Nothing is required to be modified.

The unit length load on the Gauss point, instead of the control point, is measured from the flow field. The detail is provided in Chapter 5.

3.4. Validation

The modified FEA solver, used as the solid structure solver in this project, has been validated by comparing its numerical results with the existing analytical results of a cantilever beam. The existing validation code for the original FEA solver is available for validating the modified solver after providing forces in the y and z directions and specifying the boundary conditions. The code is provided in Appendix B.

The beam is subject to a constant distributed load in the y direction along the rear half of the beam and a constant distributed load in the z direction along the whole beam, as shown in Figure 3.6.



(a) Load in the y direction.

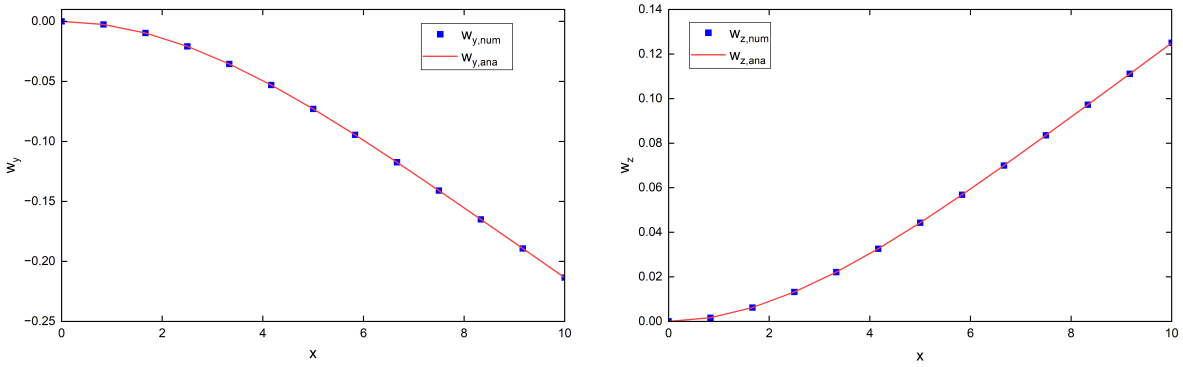
(b) Load in the z direction.

Figure 3.6: Loads on the cantilever beam.

The numerical experiment setting is the following. **All the values in the code are dimensionless.** The number of elements in IGA is $numE = 12$, and the degree of the B-spline curve is $P = 3$. The bending stiffness is $EI = 1000$. The beam length is $L = 10$. The distribution loads in y and z directions are $q_y = -2$ and $q_z = 1$. There are 48 Gauss points in IGA. Apply the load directly on Gauss points: all points are subject to the load in the z direction, while the last 24 points are subject to the load in the y direction. The quantity of the Gauss point in the code is defined as

$$n_g = numE \times (P + 1) \ . \tag{3.5}$$

The comparisons between numerical and analytical results in two directions are provided in Figure 3.7a and 3.7b. $w_{y,num}$ and $w_{z,num}$ are numerical displacements. The numerical displacements fit the analytical results very well.



(a) Displacement result comparison in the y direction.

(b) Displacement result comparison in the z direction.

Figure 3.7: Solid structure solver validation. Numerical results are shown on 13 sampling points.

The analytical results of the beam displacement in the physical space are given in 3.6 and 3.7[49]:

$$w_{y,ana} = \begin{cases} \left[\dfrac{3q_y}{16EI/L^3}\left(\dfrac{x}{L}\right)^2 - \dfrac{q_y}{12EI/L^3}\left(\dfrac{x}{L}\right)^3\right] L \ , & 0 \le x \le L/2; \\ \left[\dfrac{q_y}{24EI/L^3}\left(6\left(\dfrac{x}{L}\right)^2 - 4\left(\dfrac{x}{L}\right)^3 + \left(\dfrac{x}{L}\right)^4\right) - \dfrac{q_y}{128EI/L^3} - \dfrac{q_y}{48EI/L^3}\left(\left(\dfrac{x}{L}\right) - 0.5\right)\right] L \ , \\ L/2 < x \le L. \end{cases} \tag{3.6}$$

$$w_{z,ana} = \left[\frac{q_z}{24EI/L^3}\left(6\left(\frac{x}{L}\right)^2 - 4\left(\frac{x}{L}\right)^3 + \left(\frac{x}{L}\right)^4\right)\right] L \ , \quad 0 \le x \le L. \tag{3.7}$$

The polynomials in square brackets in equations are parametric displacement functions. x/L is the parametric coordinate ζ in IGA. L^4 is multiplied to transfer the value from parametric to physical space.

There are 13 displacement sampling points. The norm of the error vector determines the accumulated error:

$$Err_y = \sqrt{\sum_{i=1}^{13}(w_{y,i,num} - w_{y,i,ana})^2} \ , \quad Err_z = \sqrt{\sum_{i=1}^{13}(w_{z,i,num} - w_{z,i,ana})^2}. \tag{3.8}$$

$Err_y = 2.79 \times 10^{-13}$ and $Err_z = 4.39 \times 10^{-13}$ indicate the tiny error in the solver. We can conclude that the solid structure solver is validated for our qualitative research.

3.5. Discussion

The solid structure solver only considers the forces and displacements in the y and z directions. Although this is reasonable, improvements can be made in future work. The principle is to reduce the model error by considering the x component of load and displacement.

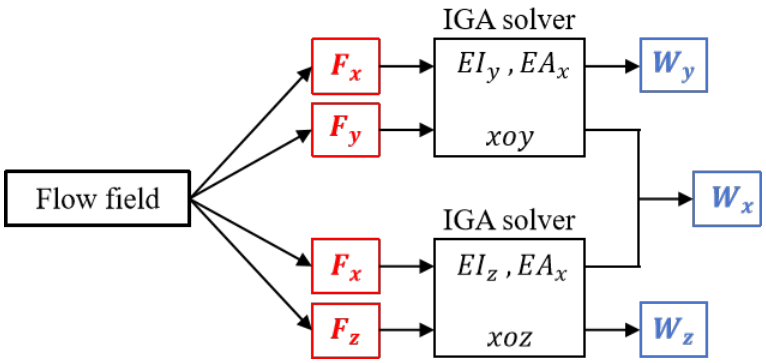


Figure 3.8: Flow chart of the improved solid structure solver.

The limitation of the original IGA solver is a potential reason for the simplification. The solver only has two channels for the load-displacement balances, so the simplest way to achieve our goal is to select the two most essential directions. The original plan was to use the existing IGA solver twice to determine displacements in three directions, as shown in Figure 3.8. Complex IGA and FSI solver modifications are required to achieve this plan. Modifying the original solver to be three-dimensional has similar problems. Hence, the most straightforward modification is applied.

4

Parametric geometry

The B-spline curve defines the central axis of our flexible cylinder in the solid structure solver. The flow solver should be able to construct and accurately measure the parametric body. WaterLily uses the signed distance function (SDF) to define the geometry in the flow field. A code is developed to generate the body using the B-spline curve combined with SDF. This chapter provides validation and Verification of the so-called parametric body code.

4.1. Actual experiment

An experiment measuring the drag of a scaled airship aligned with the far-field flow direction[17] is selected as the validation reference. The experiment equipment is shown in Figure 4.1. The airship had no motion.

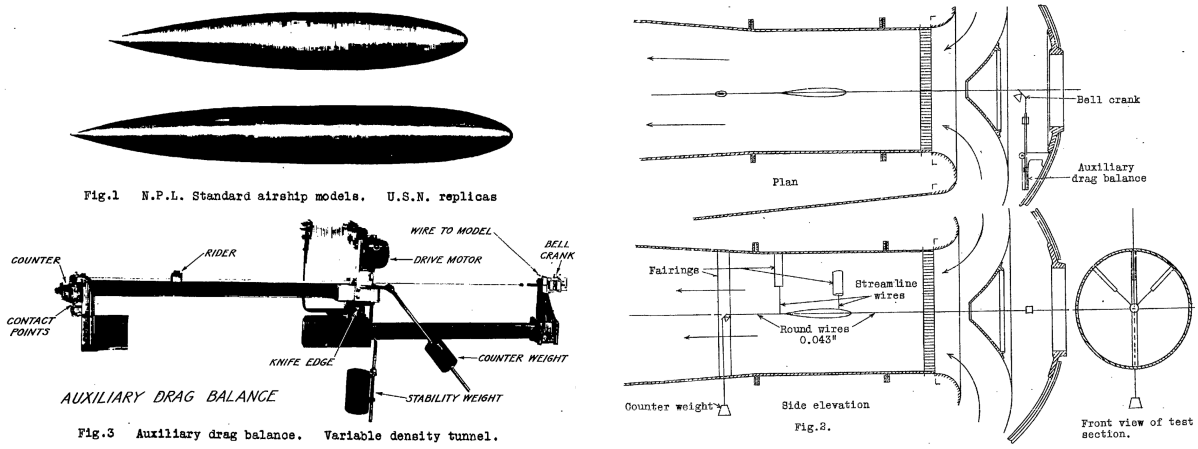


Figure 4.1: Schematic of the experiment setting[17].

The longitudinal axis of the airship was aligned with the central axis of the flow field. The slender airship shares some similarities with our case. The cylinder with a half-sphere leading edge and various shapes downstream end can be considered a slim body. Hence, the code is validated if the numerical results for the airship under the same conditions match the experimental results. The hydrodynamic drag is only considered since the experiment and our case neglect the gravity effect.

The experiment tested the airship's total drag coefficient for various Reynolds numbers Re_{exp}. The Reynolds number of the flow field was changed by altering the atmospheric pressure. The total drag coefficient was determined by[17]:

$$C_{D,exp} = \frac{Drag}{0.5\rho_{exp}(vol.)^{2/3}} , \tag{4.1}$$

where $Drag$ is the measured drag, ρ_{exp} is the air density, and $vol.$ is the airship volume. The Reynolds number was determined by:

$$Re_{exp} = \frac{\rho_{exp} v_{exp}}{\mu_{exp}} (vol.)^{1/3} . \tag{4.2}$$

Here, v_{exp} and μ_{exp} are far-field air velocity and dynamic viscosity. The velocity is constant $v_{exp} = 50\ miles/h$. We select a pair of data for the validation. When $Re_{exp} = 108500$, $C_{D,exp} = 0.0341$. The dimensions of the tested airship are given in Table 4.1. $Sta.$ and $Diam.$ are the station number and local diameter.

Table 4.1: Dimensions of N.P.L. airship models in inches[17].

$Sta.$	$Diam.$	$Sta.$	$Diam.$
0.000	0.000	12.000	4.184
0.500	1.294	13.000	4.158
1.000	1.849	14.000	4.101
1.500	2.266	15.000	4.010
2.000	2.580	16.000	3.889
2.500	2.847	17.000	3.724
3.000	3.073	18.000	3.532
3.500	3.268	19.000	3.326
4.000	3.439	20.000	3.098
4.500	3.585	21.000	2.845
5.000	3.711	22.000	2.554
6.000	3.916	23.000	2.236
7.000	4.059	24.000	1.883
8.000	4.150	25.000	1.502
9.000	4.188	26.000	1.068
10.000	4.196	27.000	0.592
11.000	4.195	27.953	0.000

The viscous force in our case is negligible due to the extremely high Reynolds number. The pressure force can be validated independently. A statistical formula for an airship can extract the experimental dynamic pressure drag coefficient[19]:

$$\frac{C_{D,wet}}{C_f} = 1 + 1.5\left(\frac{d}{l}\right)^{3/2} + 7\left(\frac{d}{l}\right)^{3} . \tag{4.3}$$

$C_{D,wet}$ and C_f are the total drag and skin friction drag coefficients based on the wetted area. l and d are the length and maximum diameter of the rotationally symmetric body. This formula applies to a rotationally symmetric body with a small d/l ratio and Reynolds number between 10^5 and 10^6[19], which is the case of the tested airship.

Substituting $l = 27.953\ inch$ and $d = 4.196\ inch$ to equation 4.3, we can obtain the ratio between the dynamic pressure drag $C_{D,p,exp}$ and total drag coefficients $C_{D,exp}$:

$$\frac{C_{D,p,exp}}{C_{D,exp}} = 1 - \frac{C_{f,exp}}{C_{D,exp}} = 0.09984 \tag{4.4}$$

Substitute the experimental total drag coefficient, and $C_{D,p,exp} = 0.00340$. It is used for validation.

4.2. Numerical setting

Appendix C shows the parametric body validation code. Based on its offset table, the two-dimensional half contour of the airship can be built. A fitted curve is generated manually by tuning the control points of the B-spline curve in the parametric code. The degree is four. The control points' positions in two-dimensional (2D) space are parameterized, so various sizes of conformal shapes can be rebuilt with characteristic lengths.

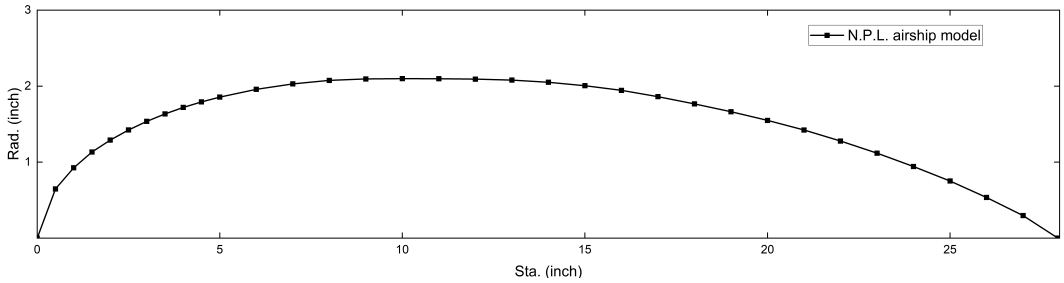
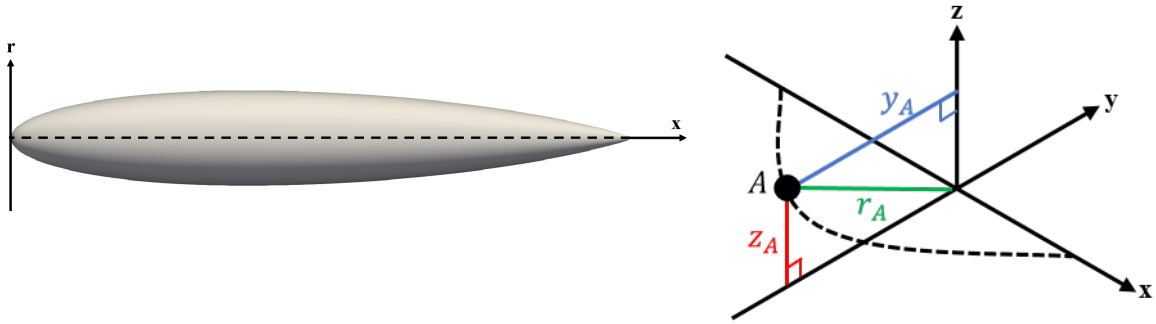


Figure 4.2: 2D half contour of the airship based on the table 4.1.

The code also obtains the SDF corresponding to the B-spine curve. The 2D B-spline curve is spun along the central axis of the airship by applying a map function to generate the 3D surface, as shown in Figure 4.3a. We set the central axis to be aligned with the x-axis in the local coordinate system. The map function is used to determine arbitrary points in the 3D space.

Assume the SDF of the 2D curve is $S_{2D}(x, r)$, x is the longitudinal direction. r is determined by $r = \sqrt{y^2 + z^2}$, as shown in Figure 4.3b. The SDF in 3D space is $S_{3D}(x, r(y, z))$. $r(y, z)$ is the so-called map function. The SDF in the global coordinate system then becomes $S(x + x_0, r(y + y_0, z + z_0))$, where (x_0, y_0, z_0) are coordinates of the local system's origin in the global system.

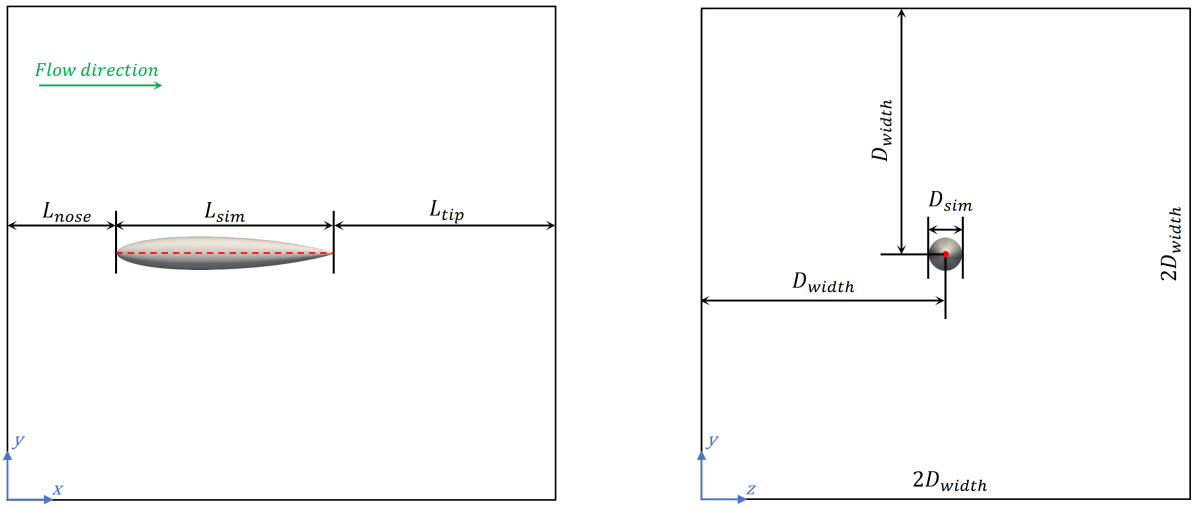


(a) 3D numerical airship model. The dashed line represents the hidden x axis.

(b) Schematic of the map function. The dashed line represents the spatial curve.

Figure 4.3: Generation of the 3D body from 2D B-spline curve by mapping.

The fluid domain should be defined after the body generation. The names of the fluid domain boundaries are shown in Appendix D. Figure 4.4 shows the dimensions of the fluid domain and the airship's position. L_{sim} is the characteristic length in the numerical simulation, representing the number of grid points along the airship length. Increasing L_{sim} means using more grids to describe a geometry, increasing the mesh resolution. D_{sim} is the max airship diameter in the code, which maintains the actual aspect ratio: $L_{sim}/D_{sim} = l/d = 27.953/4.196$. As shown in Figure 4.4a, L_{nose} is the longitudinal distance between the airship nose and the flow inlet boundary, and L_{tip} is the distance between the airship tip and the outlet boundary defined by the characteristic length with a magnification n_L. We have: $L_{tip} = n_L L_{sim}$. Figure 4.4b shows that D_{width} is the half width of the flow domain, which is defined by the airship diameter with a magnification n_D. We have: $D_{width} = n_D D_{sim}$.



(a) View of the fluid domain towards the negative z direction.

(b) View of the fluid domain towards the positive x direction.

Figure 4.4: Schematic of the fluid domain and the airship. The green arrow is the far-field flow direction: x direction.

The boundary conditions on the airship surface are no-penetration and no-slip conditions. The fluid domain simulates the infinite field, considering the deep water assumption. On the fluid domain's front, back, upper, and lower boundaries, so-called side boundaries, the fluid velocity field should fulfill: (1) the velocity component perpendicular to the boundary is constant zero (Dirichlet B.C.); (2) the spatial derivatives of all velocity components are zero (Neumann B.C.). The parallel velocity component should be zero on the inlet and outlet boundaries, and the Neumann boundary condition is the same as before. The pressure field is solved by the fractional step method[18] based on the velocity field.

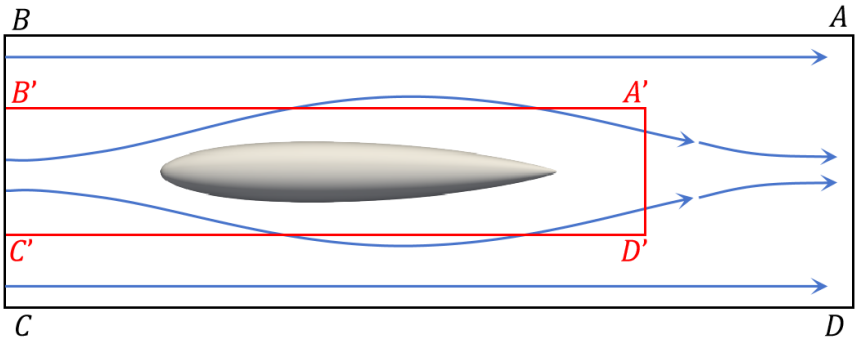


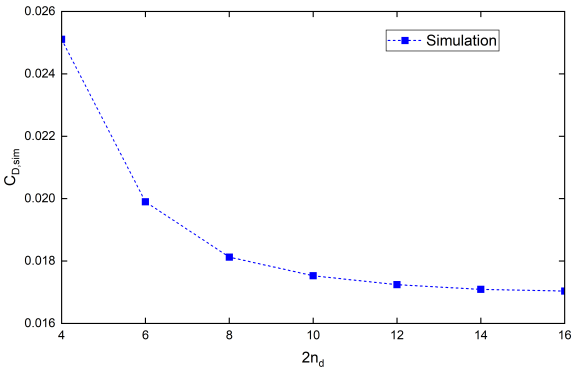
Figure 4.5: 2D schematic of the domain verification. The blue arrows represent the streamline. The red line indicates an inappropriate domain. The black line is the appropriate domain that remains to be found.

If the fluid boundaries are too close to the airship, the flow around the body will violate the boundary conditions, as shown in Figure 4.5. The condition that the streamline flows out and back to the domain after the development should be prevented. Hence, an appropriate fluid domain is verified by adjusting n_D and n_L and monitoring the convergence of the total drag coefficient.

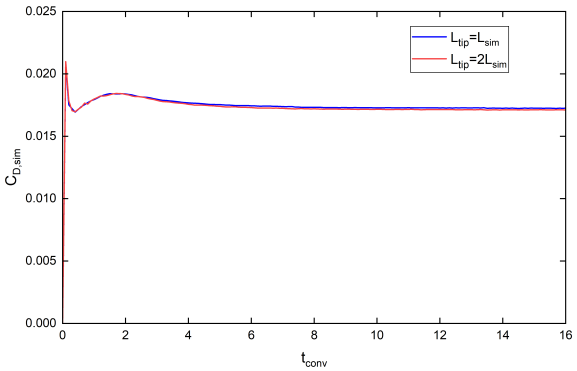
The fluid domain's dimensions in WaterLily should be even numbers. The characteristic length is constant $L_{sim} = 64$, but the domain width can only be found as a close even number due to the aspect ratio. We still denote the case by n_D. $L_{nose} = 0.5L_{sim}$ is pre-defined without testing since the flow on the inlet boundary has no development. It maintains the initial flow settings, which are the boundary conditions.

The fluid domain width with the magnification $2n_D$ is first verified, as shown in Figure 4.6a. The integer n_D ranges from 2 to 8. The simulation's dimensionless far-field flow velocity is $U_\infty = 1$. The simulation

duration is 16 convective time units. The convective time is $t_{convv} = tU_\infty/L_{sim}$, where t is the dimensionless time scale in the simulation. The drag coefficient is stable after a short period, as shown in Figure 4.10b. The average of the instantaneous drag coefficients in the last two convective time units is the data point in Figure 4.6a.



(a) n_D variation of the total drag coefficient. $C_{D,sim}$ is the total drag coefficient from the numerical simulation.



(b) Time variations of drag coefficients with different L_{tip}. t_{conv} is the convective time in simulations. $2D_{width} = 12D_{sim}$.

Figure 4.6: Verification of the fluid domain.

The drag coefficient converges after the width reaches $12D$. Thus, we conclude that the domain width should be larger than $12D$ for the drag coefficient to converge. Then we use $2D_{width} = 12D$ to verify the L_{tip}, as shown in Figure 4.10b. The result between $L_{tip} = L_{sim}$ and $L_{tip} = 2L_{sim}$ has a tiny difference, so we select $L_{tip} = L_{sim}$ to reduce the computational cost.

4.3. Validation and verification of the parametric code

After finding an appropriate fluid domain for testing, we can now validate the parametric code. Due to the problem caused by the irregular airship aspect ratio, redefine the domain width $2D_{width}$ directly to the characteristic length L_{sim}:

$$2D_{width} = 12 \times \frac{3}{16} L_{sim} \ , \tag{4.5}$$

$$2D_{width} = 2.25 \times \frac{27.953}{4.196} D_{sim} \approx 15 D_{sim} > 12 D_{sim} \ .$$

L_{sim} is set to be a multiple of 16, so the domain width is always an even number. We keep $L_{nose} = 0.5L_{sim}$ and $L_{tip} = L_{sim}$. The Reynolds number is the same as in the experiment. The far-field flow speed is $U_\infty = 1$. The fluid density in the code is $\rho_{sim} = 1$. Hence, the dynamic viscosity used in the simulation is determined by:

$$\mu_{sim} = \frac{\rho_{sim} U_\infty}{Re_{exp}} (vol._{sim})^{1/3} \ . \tag{4.6}$$

Here, $vol._{sim}$ is the numerical airship volume, which can be determined by the scaling method with known geometry relation:

$$(vol._{sim})^{1/3} = \left(\frac{vol.}{l^3} \right)^{1/3} L_{sim} \ . \tag{4.7}$$

The parametric code is validated and verified by increasing the characteristic length and comparing the resultant pressure drag coefficient $C_{D,p,sim}$ with the experiment data $C_{D,p,exp}$, as shown in Figure 4.7. The simulation duration is 8 convective time units. We still take the average of instantaneous pressure

drag coefficients in the last two time units as the data point in Figure 4.7. The schematics of the velocity and pressure fields at $t_{conv} = 8$ are shown in Appendix E. The numerical result of the pressure drag coefficient is converged when $L_{sim} = 256$. The value is $C_{D,p,sim} = 0.00328$, and the error rate is 3.5% compared to $C_{D,p,exp} = 0.00340$. The error is sufficiently small for our research.

Hence, we conclude that the parametric body code is validated and verified for a problem dominated by the pressure force. The code can accurately describe a slender body in axial flow.

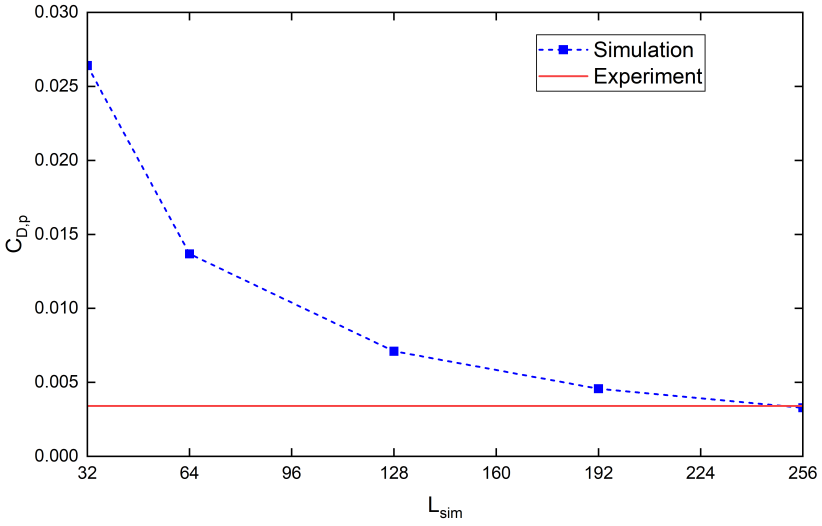


Figure 4.7: Validation and verification of the parametric code. L variation of the pressure drag coefficient.

4.4. Discussion

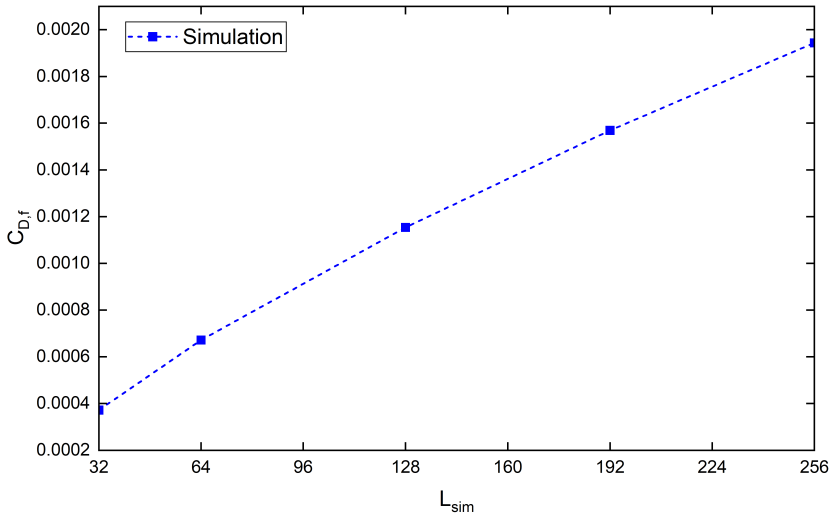


Figure 4.8: Viscous drag coefficient $C_{D,f}$ under various characteristic lengths.

Using a statistical formula to extract the pressure drag coefficient from the experimental total drag coefficient is not a typical approach. The regular process uses the total coefficient for the validation. The viscous drag of our case cannot be converged even with the largest L_{sim}, as shown in Figure 4.8. The post-process is the same as the method used for the pressure drag.

The first possible reason is the insufficient mesh resolution. Assume the velocity u is a function of

y. The local spatial gradient $\partial u(y)/\partial y$ on the body surface should be determined accurately to obtain the friction by Newton's law of viscosity. For FVM with the staggered grid, this indicates that the grid adjacent to the body surface should be thinner than the viscous sublayer thickness. Use the turbulent channel flow's velocity profile to approximate the sublayer thickness of the actual airship[37], as shown in equation 4.8:

$$\delta_v = 5\frac{\nu}{u^*}\ , \tag{4.8}$$

where δ_v corresponds to $y+ = yu^*/\nu = 5$. Substitute the kinematic viscosity at $15°C$ under a standard atmosphere pressure $\nu = 1.47 \times 10^{-5}\ m^2/s$[13]. The corresponding $u*$ is solved by the wall bounded friction law equation[37] with $C_{D,f,exp} = C_{D,exp} - C_{D,p,exp}$:

$$C_{D,f,exp} = 2\left(\frac{u^*}{v_{exp}}\right)^2 \tag{4.9}$$

The ratio between sublayer thickness δ_v and the actual airship's max diameter d is compared with the ratio between a single grid and the total number of grids describing the numerical diameter:

$$\frac{\delta_v}{d} = 2.5 \times 10^{-4}\ ,\ \frac{1}{L_{sim} \times (d/l)} = 2.6 \times 10^{-2} \tag{4.10}$$

Although we apply the max tested characteristic length $L_{sim} = 256$, the numerical ratio is much larger than the actual ratio. The grid is too thick to approximate the velocity gradient on the body surface accurately, as shown in the schematic figure 4.9.

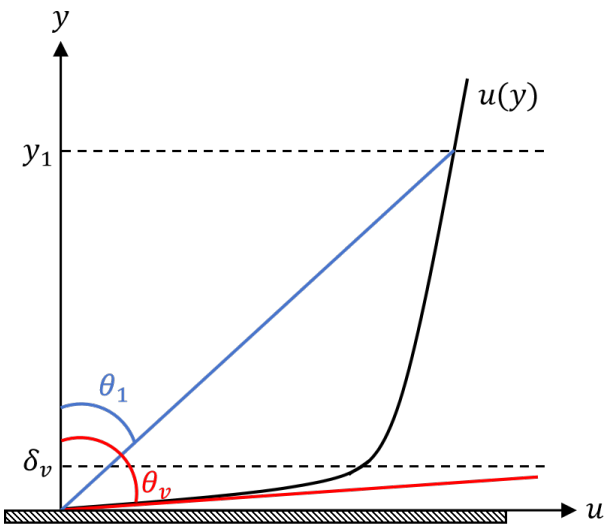


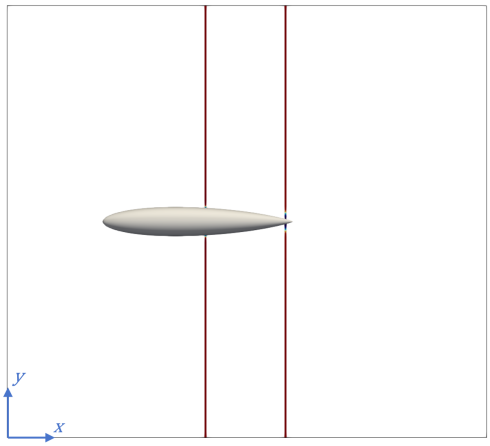
Figure 4.9: Velocity profile near the solid boundary. The black curve is the velocity profile. y_1 represents the grid thickness. The shadowed rectangle represents the airship's surface. The numerical approximation is $tan\ \theta_1 = (u(y_1) - u(0))/y_1$. The actual velocity gradient is $tan\ \theta_v = \partial u(0)/\partial y$. It is apparent that $tan\ \theta_1 < tan\ \theta_v$. The numerical gradient is smaller than the actual one.

Increasing L_{sim} leads to reducing y_1 and increasing θ_1, and then the numerical gradient grows. The numerical viscous drag increases, which confirms the tendency shown in Figure 4.8.

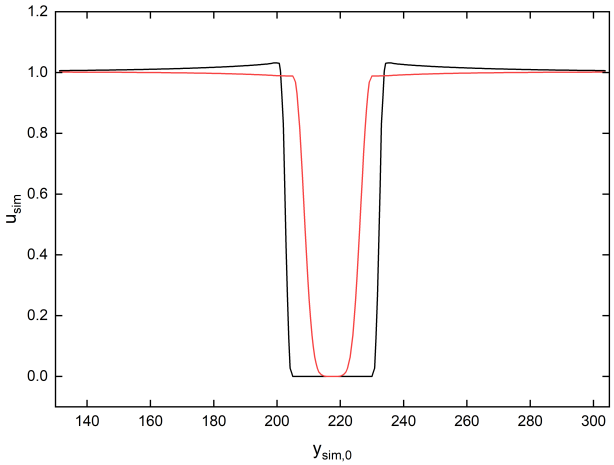
Furthermore, the airship's shape is perfectly streamlined and rotationally symmetric, reducing axial pressure drag and increasing the ratio of viscous to total drag. In our case, we consider the lateral unit length force on the cylinder. The body is slender but not streamlined. Rotational symmetry does not contribute to the lateral force. The pressure and viscous force share the same characteristic length (diameter). Thus, we can regard the Reynolds number as the ratio between two forces.

We believe increasing the characteristic length further could help the viscous drag coefficient converge. However, the Reynolds number of our case is enormous, indicating the low viscous effect. In conclusion, increasing the characteristic length at the cost of more computational consumption is not worth it, since we have validated that the code can obtain a converged and correct pressure drag coefficient.

The second possible reason is that the numerical simulation lacks roughness and support structures, as in the actual experiment. A rough surface and extra structures could make the flow field more turbulent despite researchers trying to maintain a laminar flow. However, the simulation's fluid field is perfectly laminar, as shown in Figure 4.10 and Appendix E. The velocity profile is stable out of the boundary layer region. Hence, we can apply the statistical formula 4.3.



(a) Positions of sampling line. The lines parallel the y axis and pass the airship's rotational axis.



(b) Velocity in x direction on grid points on the sampling line. $y_{sim,0}$ is the global coordinate, including the ghost cell for boundary conditions.

Figure 4.10: Flow velocity profile around the airship. The black line represents the profile of the front sampling line, and the red line represents the profile of the back one. The profile near the physical boundary is smooth since we apply a kernel in BDIM.

Although using the same parametric body code, the way the code generates an airship differs from the method of generating a cylinder. To create a cylinder, the central axis is first defined by the B-spline curve and then given a thickness (diameter) to determine its surface, which utilizes the SDF feature. Measure the distance from an arbitrary point in the space to the central axis. If this distance exceeds the half-thickness, the spatial point is outside the cylinder. On the contrary, it is the opposite. The spatial point is on the surface if the distance equals half the thickness. So, the 3D cylinder always has a half-sphere cap at both ends, as shown in Figure 4.11.

The airship experiment is the best case we found for validating the code. Considering all the above reasons, we still conclude that the parametric body code is validated.

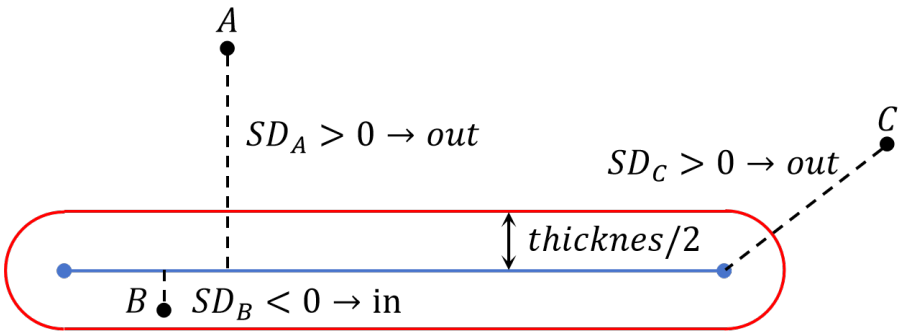


Figure 4.11: Side schematic of a cylinder in the code. The red line represents the surface of the cylinder. The blue line is the central axis generated by the B-spline curve. The dashed line is the signed distance from the point to the axis.

5

FSI solver

The FSI solver determines the coupling effect between the structure and fluid. A pressure force integrator is developed to measure the load on the flexible cylinder from the fluid field. The original FSI solver applies to a 2D problem, so it is modified and validated to be suitable for our 3D case.

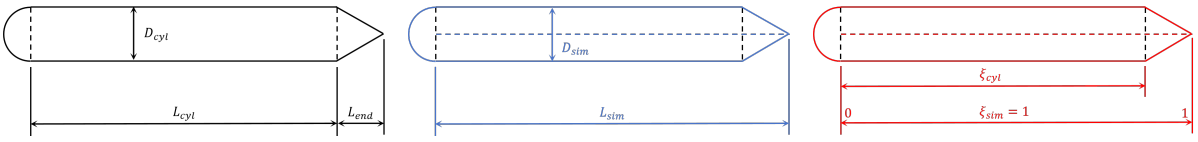
5.1. Reference and scaling

Before developing the pressure force integrator, we must determine the relationship between various reference spaces.

We have a tested cylinder with a tapered downstream end from an actual experiment[43]. The reference system is called the **physical space**, as shown in Figure 5.1a. The dimensionless geometry is defined in this space: the aspect ratio of the cylinder (L_{cyl}/D_{cyl}) and the ratio of the tapered end (L_{end}/D_{cyl}). Two similarity criteria, the Reynolds number and the Cauchy number, are determined in this space:

$$Re = \frac{\rho_{wat} U_{exp} D_{cyl}}{\mu_{wat}} \ , \ Ca_{cyl} = \frac{E_{cyl}}{\rho_{wat} U_{exp}^2} \ . \tag{5.1}$$

L_{cyl} and L_{cyl} are the lengths of the cylinder segment and the tapered end. D_{cyl} is the cylinder diameter. U_{exp} is the far-field flow velocity in the experiment, and E_{cyl} is Young's modulus of the tested cylinder. The water property is selected as $15°C$ pure water. The density is $\rho_{wat} = 1000\ kg/m^3$, and the dynamic viscosity is $\mu_{wat} = 1.1375 \times 10^{-3}\ Pa \cdot s$[40].



(a) Actual cylinder[43] in the physical space. L_{cyl} is the cylinder length. L_{end} is the tapered end length.

(b) Cylinder in the simulation space. L_{sim} is the length of the central axis, which is also the characteristic length of the simulation.

(c) Cylinder in the parametric space used in the solid structure solver. $\xi = x_{sim}/L_{sim}$, where x_{sim} is the simulation coordinate.

Figure 5.1: Side view of dimensions of tested cylinder with a tapered end in different reference spaces. The solid black line represents the shape in real space. The dashed black line divides various segments of the whole body. Blue and red dashed lines are the central axis generated by the B-spline curve. $\xi_{sim} = 1$ represents the whole length.

We use WaterLily as the flow solver to simulate the cylinder in axial flow. As shown in Figure 5.1b, the properties used in the solver are defined in the so-called global **simulation space**, which includes L_{sim} and D_{sim}. L_{sim} is the characteristic length of the simulation, which is a longitudinal length, and it is also the total length of the central axis of the inextensible cylinder with various ends (differs from "cylinder" in the physical space) since the deflection is slight. The simulation space should maintain the same

similarity criteria Re and Ca_{cyl}. Thus, Young's modulus and the dynamic viscosity of the simulation are determined by:

$$\mu_{sim} = \frac{\rho_{sim} U_\infty D_{sim}}{Re} \ , \ E_{sim} = Ca_{cyl} \rho_{sim} U_\infty^2 \ . \tag{5.2}$$

All properties in the code are dimensionless. The fluid density and far-field flow velocity are $\rho_{sim} = 1$ and $U_\infty = 1$. We also have the geometry relation:

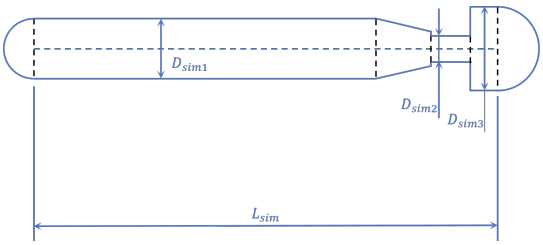
$$\frac{L_{cyl}}{D_{cyl}} = \frac{L_{sim}}{D_{sim}} \ . \tag{5.3}$$

The solid structure solver generates the body and solves the displacement in the local **parametric space**. The characteristic length in this space is $\xi_{sim} = 1$, as shown in Figure 5.1c. The geometry relation between simulation and parametric space is $\xi_{sim}/L_{sim} = 1/L_{sim}$. The relation between physical and parametric space is $L_{cyl}/(L_{cyl} + L_{end}) = \xi_{cyl}/\xi_{sim}$, where ξ_{cyl} is the parametric length of the cylinder segment of the whole body.

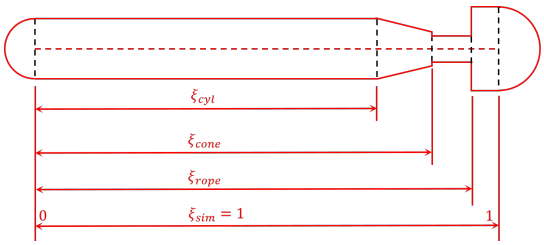
The FSI code delivers the distribution load measured in the simulation space to the solid structure solver. The unit length load requires no change because it has no length unit. The bending stiffness $E_{sim} I_{sim}$ requires scaling to the parametric space since it contains a third-order length unit. The relation between parametric and simulation spaces is:

$$E_{para} I_{para} = E_{sim} I_{sim} \cdot \left(\frac{1}{L_{sim}} \right)^3 \tag{5.4}$$

In Chapter 3, we have a simplification that all the points on the cylinder's central axis have no displacement in the axial direction - the global x direction. The x coordinates of these points never change, so we use the local parametric coordinates in the axial direction $\xi = x_{sim}/L_{sim}$ to denote the longitudinal position of arbitrary points on the central axis. Once we have the central axis, the rotationally symmetric body can be generated based on the diameter ("thickness" in the code). The original code can only accept a constant diameter, so a modification is developed: the diameter in the simulation space becomes a function $D(\xi)$ of parametric coordinate (in the code, we define the diameter function in the parametric space, the flow solver will multiply a L_{sim} automatically to obtain the D_{sim}). With the diameter function, we can generate the cylinder with a tapered downstream end or even a more complex body, as shown in Figure 5.2. The details are provided in the section 5.2. The caps at the cylinder's beginning and end are negligible since they only take a small part of the body. We cannot remove the caps due to the SDF feature. However, we can obtain a cone if the local diameter is zero, as shown in Figure 5.1. We call the cone and blunt end the "tapered end".



(a) A cylinder with a parachute at the downstream end in the simulation space.



(b) The cylinder with a parachute in the parametric space used in the solid structure solver.

Figure 5.2: Side view of cylinder dimensions with a parachute at the free end in the simulation and parametric space. The dashed black line divides the body into various segments: a cylinder, a cone, a rope, and a parachute. Blue and red dashed lines are the central axis generated by the B-spline curve. $\xi_{sim} = 1$ represents the whole length.

The solid structure solver returns the central axis' displacement in the parametric space, so the FSI code always multiplies an L_{sim} for transformation. This process cannot be changed since it relates to the basic coupling solver.

A further description of the coordinate system is given. The flow solver uses the global coordinate system. In this system, the initial central axis of the cylinder is set to be parallel to the global x direction, which is also the direction of the far-field flow. The solid structure solver uses a local parametric coordinate system; the origin is located at the beginning of the cylinder's central axis, generated by the B-spline curve. The initial central axis lies on the system's x_ξ axis. The parametric coordinate indicates the position on the central axis. Global coordination can be obtained from this position, as shown in Figure 5.3.

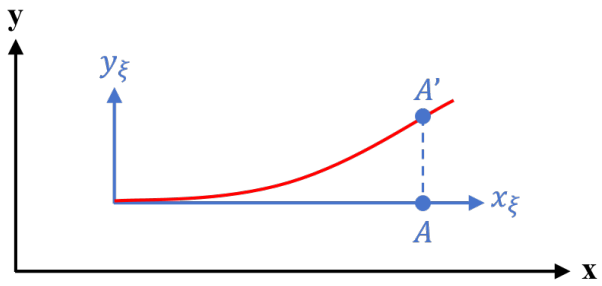


Figure 5.3: Schematic of two coordinate systems. Black arrows form the global system. Blue arrows form the local system. The dashed blue line is the displacement of the A point in the local system. The red line represents the axis of the bending cylinder. The axes of the two systems are parallel to each other.

5.2. Pressure force integrator

We developed the FSI code using an old version of WaterLily. After this project, an upgrade is required to make the code generally available. The pressure force integrator is developed in the frame of the parametric body code. The code can measure the unit length force on the Gauss point from the pressure field given by the flow solver. The validation code is provided in Appendix G.

5.2.1. Body generation

Before we develop the integrator, we should be able to generate a cylinder with different downstream ends. Since the body is rotationally symmetric, we first generate the central axis using the B-spline (NURBS) curve. The coordinate system used to create the geometry is local.

The control points are in the parametric space; the actual position is obtained by multiplying the characteristic length L_{sim} and applying translation. We use three control points to generate a straight axis in the xoy plane for validation. The coordinates are: $(0, 0, 0)$, $(0.5, h_\xi/2, 0)$, and $(1, h_\xi, 0)$. The inclined angle of the axis can be shifted by adjusting h_ξ, as shown in Figure 5.4. If h_ξ is large, the central axis length no longer equals L_{sim}. In our case, we have more control points and tiny deflections. Multiply L_{sim} by the parametric coordinates and apply the translation, then we can have the global control points (in the black coordinate system in Figure 5.4).

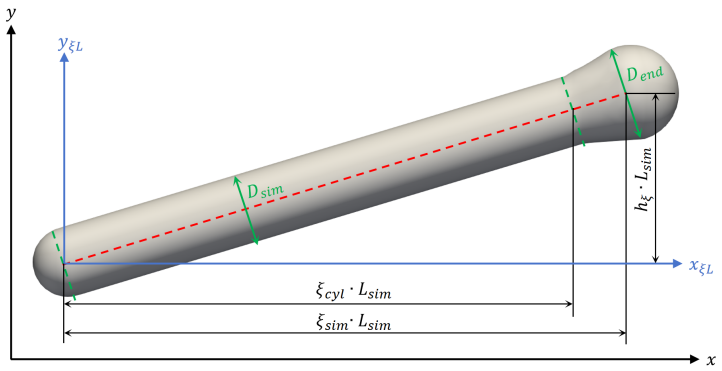


Figure 5.4: Schematic of an inclined cylinder with an extra blunt (taper) end ($D_{end} > D_{sim}$) in the simulation. The central axis is in the xoy plane. $h_\xi = 0.3$ in this figure. The dashed red line is the central axis. The blue coordinate system is local (obtained by multiplying L_{sim} by the parametric system).

Then, we apply the diameter function. For a cylinder with a tapered end, we have:

$$D(\xi) = \begin{cases} D_{sim} \ , 0 \leq \xi \leq \xi_{cyl} \ ; \\ \left(\dfrac{D_{end}/2 - D_{sim}/2}{1 - \xi_{cyl}} \xi + \dfrac{D_{end}}{2} - \dfrac{D_{end}/2 - D_{sim}/2}{1 - \xi_{cyl}} \right) \times 2 \ , \xi_{cyl} < \xi \leq 1 \ . \end{cases} \tag{5.5}$$

D_{end} is the body diameter at the end of the central axis, as shown in Figure 5.4. ξ_{cyl} is the parametric separation point of two segments: the cylindrical part and the tapered end. The tapered end becomes a cone if $D_{end} = 0$. The equation 5.5 provides the diameter in the global system. It only uses the parametric coordinate ξ to define the longitudinal position on the central axis.

We define the central axis of the cylinder with a parachute end in the xoz plane for validation. The control points are $(0, 0, 0)$, $(0.5, 0, h_\xi/2,)$, and $(1, 0, h_\xi)$. The diameter function is:

$$D(\xi) = \begin{cases} D_{sim1} \ , 0 \leq \xi \leq \xi_{cyl} \ ; \\ \left(\dfrac{D_{sim2}/2 - D_{sim1}/2}{1 - \xi_{cyl}} \xi + \dfrac{D_{sim2}}{2} - \dfrac{D_{sim2}/2 - D_{sim1}/2}{1 - \xi_{cyl}} \right) \times 2 \ , \xi_{cyl} < \xi \leq \xi_{cone} \ ; \\ D_{sim2} \ , \xi_{cone} < \xi \leq \xi_{rope} \ ; \\ D_{sim3} \ , \xi_{rope} < \xi \leq 1 \ . \end{cases} \tag{5.6}$$

D_{sim1}, D_{sim2}, and D_{sim3} are diameters of different segments, as shown in Figure 5.2a. ξ_{cyl}, ξ_{cone}, and ξ_{rope} is parametric lengths of various segments. Diameters have abruptions at $\xi = \xi_{cone}$ and $\xi = \xi_{rope}$, as shown in Figure 5.2b. The generated body is provided in Figure 5.5. The geometry is imperfect due to the mesh resolution, and the ”parachute” is solid due to code limitations. We believe it is sufficient for qualitative research.

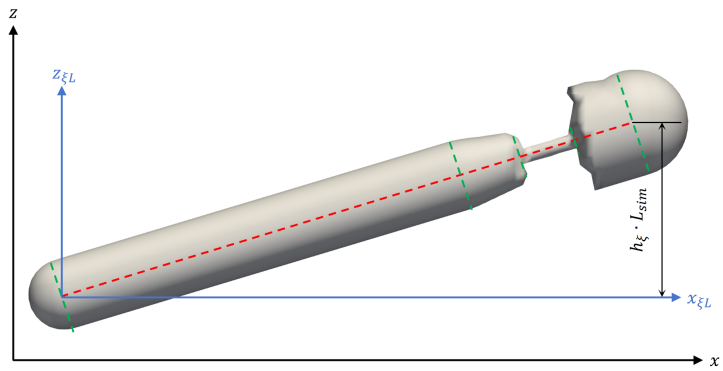


Figure 5.5: Schematic of an inclined cylinder with a parachute at the end. The central axis of the body is in the xoz plane. The central axis is straight. $h_\xi = 0.3$ in this figure. The green dashed line separates different segments.

Although we apply the variable diameters, the bending stiffness in the solid structure solver is constant. Applying variable bending stiffness requires additional code modifications. Since the tapered end or parachute only takes a small part of the body, we still use the bending stiffness of the cylindrical part on the tapered end. In our validation cases, the tapered end may look giant. However, it will be tiny compared to the whole body when we simulate the investigation cases in Chapter 6, which is similar to the previous experiment[43].

We can also generate the central axis as a semicircle. For later validation, define the axis in the xoy plane by the NURBS curve. The NURBS curve is the non-uniform rational B-spline curve[52]. A semicircle’s parametric control points, weights, and knot vector are available in an existing example case:

$$\begin{aligned} \Xi =& [0,0,0,1/2,1/2,1,1,1] \ , \\ wieghts =& [1,\sqrt{2}/2,1,\sqrt{2}/2,1] \ , \\ cps =& (-1,0,0),(-1,1,0),(0,1,0),(1,1,0),(1,0,0) \end{aligned} \tag{5.7}$$

The diameter of the semicircle is $2L_{sim}$, as shown in Figure 5.6. The diameter of the cylinder is set to be a constant D_{sim}, and no specific shape is applied to the end.

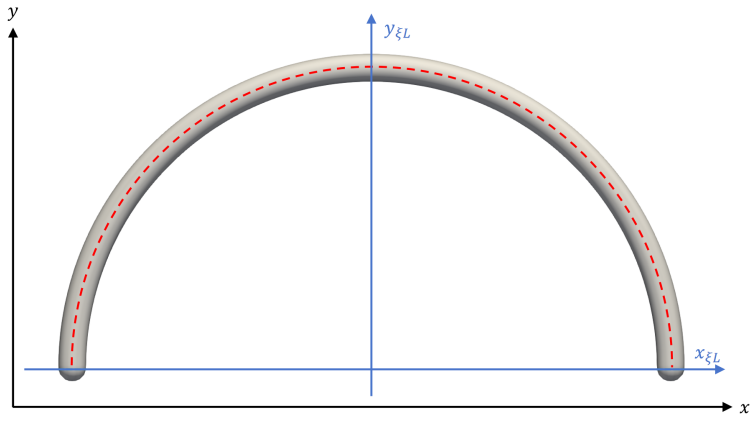


Figure 5.6: Schematic of a semicircle cylinder with a constant diameter. $D_{sim} = 6$ in this figure. The dashed red central axis of the cylinder lies in the xoy plane. The blue coordinate system is local but not a parametric system.

5.2.2. Methodology

The solid structure solver requires the unit-length force on the Gauss point, which is on the body's central axis (rotational axis). We integrate the pressure force on the edge of the cross-section that passes the Gauss point and is perpendicular to the local axis's tangential direction to represent the local load distribution on the Gauss point, as shown in Figure 5.7. The unit is force per length.

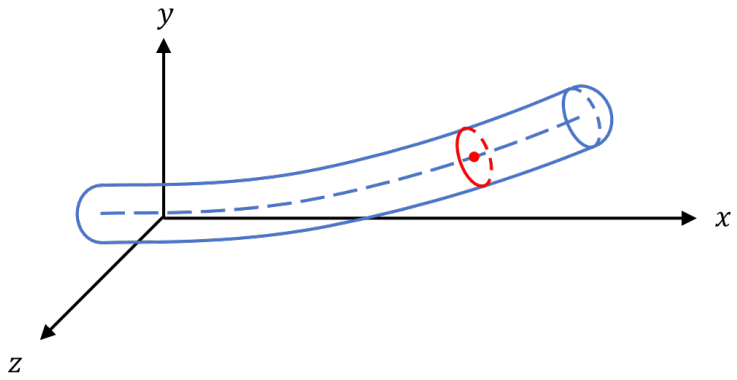
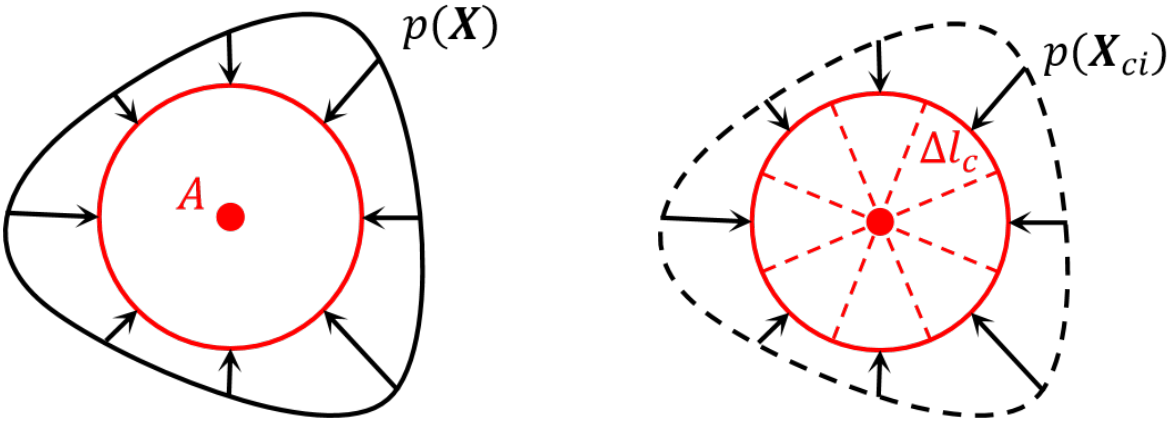


Figure 5.7: Schematic of a 3D cylinder. The red line represents a cross-section. The red dot is the Gauss point in the cross-section. The long-dashed line is the central axis.

Focus on the cross-section. Figure 5.8a shows the pressure force integration along the section edge. **X** is the global coordinate of a point on the cross-section edge, and p(**X**) is the pressure on the point. Determining the analytical expression of the pressure function is complex, so we use the feature of FVM to approximate the force integration:

$$\oint_{\Omega_c} p(\mathbf{X}) \cdot \mathbf{n} dl = \sum_{i=1}^{n_{cr}} p(\mathbf{X}_{ci}) \cdot \mathbf{n}_i \Delta l_c \ . \tag{5.8}$$

Ω_c denotes the cross-section. **n** is the local normal direction. We evenly divide the circle into n_{cr} segments, so each segment has the length $\Delta l_c = \pi D(\xi_g)/n_{cr}$, where ξ_g is the parametric longitudinal coordinate of the Gauss point. $p(\mathbf{X}_{ci})$ is the pressure on the sampling point on the circle, as shown in Figure 5.8b. The sampling point is at the midpoint of each arc segment. Given n_{cr} sampling points, the solar coordinate of the point in the cross-section plane is $((i-1)\Delta\theta_c\ ,\ D(\xi_g)/2)$, where $\Delta\theta_c = 2\pi/n_{cr}$ and integer $i \in [1, n_{cr}]$.



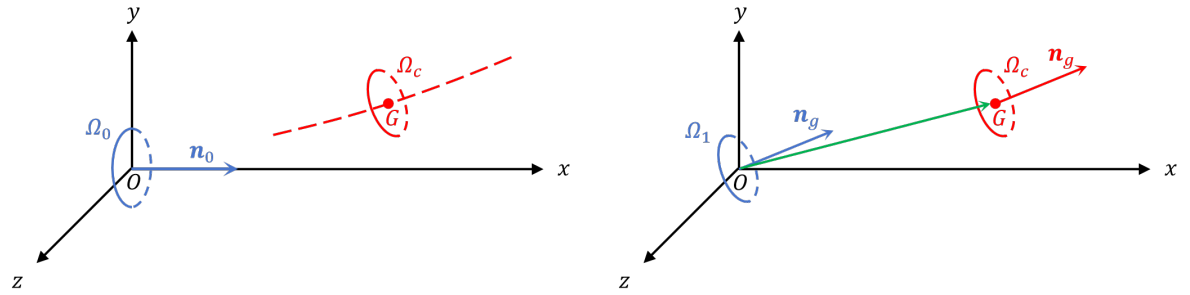
(a) Original force integration along the edge.

(b) Approximation of the force integration.

Figure 5.8: Schematic of the force integration in the cross-section. The red dot is the Gauss point.

With the approximation equation 5.8, we only need to determine the global coordinates of sampling points $\mathbf{X}_{ci}$ and the corresponding force normal directions $\mathbf{n}_i$. Deliver the coordinates to the flow solver, and then we can automatically obtain the local pressure from the numerical pressure field. Determining the coordinates and normal direction requires using Rodrigues' rotation formula[9][23][65].

We first define a basic cross-section in the yoz plane of the global system, with the center at the system origin, as shown in Figure 5.9a. All processes are done in the global system.



(a) Basic cross-section in the global yoz plane.

(b) Intermediate cross-section with a rotation angle.

Figure 5.9: Procedure of applying Rodrigues' rotation formula. G denotes the Gauss point.

For the cylindrical part, on the edge of the cross-section Ω_0, the sampling point's coordinate is defined as

$$(x_0, y_0, z_0) = \begin{cases} x_0 = 0\ ; \\ y_0 = \dfrac{D(\xi_g)}{2} cos[(i-1)\Delta\theta_c]\ ,\ i \in [1, n_{cr}] \cap \mathbb{N}\ ; \\ z_0 = \dfrac{D(\xi_g)}{2} sin[(i-1)\Delta\theta_c]\ ,\ i \in [1, n_{cr}] \cap \mathbb{N}\ . \end{cases} \tag{5.9}$$

The plane's normal direction of cross-section Ω_0 is $\mathbf{n}_0 = [1, 0, 0]$. Given an arbitrary sampling point B on the edge of the basic cross-section. Denote the origin of the coordinate system as O. Normalize

vector $\vec{OB}$ first for better explanation, then we can have the point's normal direction on the edge in the yoz plane (**only for the cylindrical part**; in the code we actually rotate the vector first, then normalize the point's normal direction $\mathbf{n}_{B1}$ in the equation 5.15, $\mathbf{n}_{B0} = \vec{OB}$; in the code we do not need to multiply the radius in the equation 5.14 since we do not normalize $\mathbf{n}_{B1}$ in this equation):

$$\mathbf{n}_{B0} = \frac{\vec{OB}}{|\vec{OB}|} \ . \tag{5.10}$$

We have $\mathbf{n}_{B0} \perp \mathbf{n}_0$. With known ξ_g, we can obtain the normalized tangential direction $\mathbf{n}_g$ of the cylinder's central axis on the Gauss point from the parametric body code. $\mathbf{n}_g$ is the plane's normal direction of actual cross-section Ω_g. The rotation angle between $\mathbf{n}_0$ and $\mathbf{n}_g$ is determined by:

$$\gamma = arcos\left(\frac{\mathbf{n}_0 \cdot \mathbf{n}_g}{|\mathbf{n}_0||\mathbf{n}_g|}\right) \ . \tag{5.11}$$

We define the rotation axis by the cross product between the two planes' normal directions and normalize it:

$$\mathbf{k} = \frac{\mathbf{n}_0 \times \mathbf{n}_g}{|\mathbf{n}_0 \times \mathbf{n}_g|} \ . \tag{5.12}$$

If $\mathbf{n}_0 \times \mathbf{n}_g = 0$, we directly return $\mathbf{k} = [0, 0, 0]$ in the code to avoid dividing zero. The spatial relation between an arbitrary point's normal direction $\mathbf{n}_{B0}$ and the plane's normal direction $\mathbf{n}_0$ is fixed, so they share the same rotation axis $\mathbf{k}$ and angle γ. Apply the Rodrigues' rotation formula to rotate the cross-section Ω_0 to Ω_1, and make $\mathbf{n}_0$ parallel to $\mathbf{n}_g$, as shown in Figure 5.9b. The point's normal direction after the rotation is[23][9][65]:

$$\mathbf{n}_{B1} = cos(\gamma)\mathbf{n}_{B0} + [1 - cos(\gamma)](\mathbf{k} \cdot \mathbf{n}_{B0})\mathbf{k} + sin(\gamma)(\mathbf{k} \times \mathbf{n}_{B0}) \ . \tag{5.13}$$

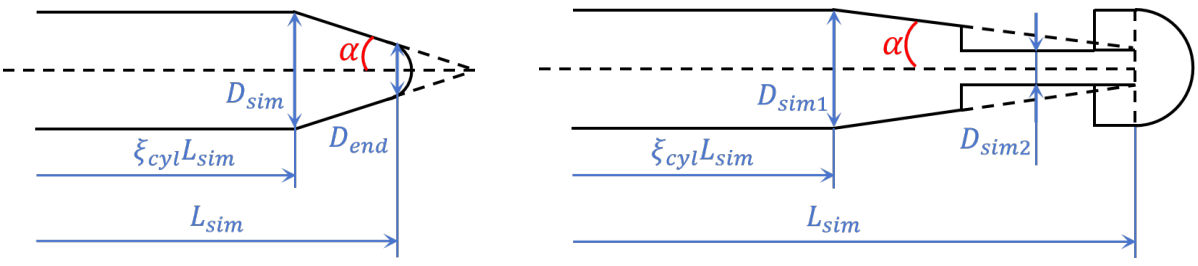
The sampling point's coordinate on the section Ω_g is equal to the corresponding value of the vector:

$$\mathbf{X}_{ci} = (x_i, y_i, z_i) = \frac{D(\xi_g)}{2}\mathbf{n}_{B1} + \vec{OG} \ . \tag{5.14}$$

On the body surface, the point's normal direction $\mathbf{n}_{B1}$ is towards the outside, while the force's normal direction is towards the inside. We have the force's normal direction:

$$\mathbf{n}_i = -\mathbf{n}_{B1} \ . \tag{5.15}$$

On the downstream end (tapered end or parachute), D_ξ is a linear function of ξ on the taper segment: $\xi \in (\xi_{cyl}, 1]$ or $\xi \in (\xi_{cyl}, \xi_{rope}]$. We first need to determine the slope angle α, as shown in Figure 5.10.



(a) Taper angle of a tapered end.

(b) Taper angle of a parachute end.

Figure 5.10: Schematic of the slope angle α of different end shapes.

The slope angle α is determined by:

$$\begin{aligned} For\ a\ tapered\ end:\ \alpha &= arctan\left(\frac{D_{end}-D_{sim}}{L_{sim}-\xi_{cyl}L_{sim}}\right)\ , \\ For\ a\ parachute\ end:\ \alpha &= arctan\left(\frac{D_{sim2}-D_{sim1}}{L_{sim}-\xi_{cyl}L_{sim}}\right)\ . \end{aligned} \tag{5.16}$$

The point's normal direction differs in the basic cross-section Ω_0. Given a new symbol $\mathbf{n}_{B01}$, it becomes:

$$\mathbf{n}_{B01} = \begin{bmatrix} sin(\alpha) \\ cos[(i-1)\Delta\theta_c]cos(\alpha) \\ sin[(i-1)\Delta\theta_c]cos(\alpha) \end{bmatrix}\ ,\ i \in [1, n_{cr}] \cap \mathbb{N}\ . \tag{5.17}$$

The force in the x direction has been neglected, so we apply a modification to the force's normal direction (operator $\odot$ is the element-wise product):

$$\mathbf{n}_i = -\mathbf{n}_{B1} \odot \begin{bmatrix} 0 \\ cos(\alpha) \\ cos(\alpha) \end{bmatrix}\ . \tag{5.18}$$

Since we never use it, we can multiply anything by the x component, for example, zero. We can still use $\mathbf{n}_{B1}$ to determine the point's coordinate on the actual cross-section's edge in the simulation. The range of α is $[-\pi/2, 0) \cup (0, \pi/2]$, so no sign issue.

In summary, we can obtain an arbitrary sampling point and the corresponding force's normal direction with known n_{cr}, ξ_g, and $D(\xi)$. The local pressure $p(\mathbf{X}_{ci})$ is obtained by averaging the pressures in adjacent grids since the sampling points may not be precisely on a grid point in the simulation.

5.2.3. Validation

To validate the pressure force integrator, we manually define a hydrostatic pressure field in the y or z direction with $\rho_{sim} = 1$ and $g_{sim} = 1$. We first define the right-hand side of the pressure Poisson equation[18]:

$$\frac{1}{\rho_{sim}}\Delta p_{sim} = -\vec{\nabla}\cdot(\vec{u}\cdot\vec{\nabla})\vec{u}\ . \tag{5.19}$$

Then, we let the flow solver solve a hydrostatic pressure field instead of defining it by assigning a value directly, which makes the field closer to an actual simulated field. The cylinder with a tapered end and the semicircle cylinder have a pressure field: $p(\mathbf{X}) = y$. The cylinder with a parachute end has the following pressure field: $p(\mathbf{X}) = z$. The pressure force integrator should be able to measure the correct unit-length force in the y and z directions at the Gauss point.

Before we run the validation code, the analytical result must be determined. Consider the cylinder with a tapered end, for example. The central axis of the cylinder with a tapered end is straight. The axis is in the xoy plane. The analytical pressure field is $p(\mathbf{X}) = y$. Figure 5.11 shows a cross-section in the coordinate system at the Gauss point G.

Inspired by the parametric sphere function[57], the point's coordinate on the cross-section's edge is determined by the expression 5.20. β is the cross-section's inclined angle along z axis in the xoy plane. η is the solar coordinate in the cross-section's plane.

$$(x, y, z) = \begin{cases} x = \dfrac{D(\xi_g)}{2}sin(\eta)cos(\beta) \\ y = \dfrac{D(\xi_g)}{2}sin(\eta)sin(\beta) \\ z = \dfrac{D(\xi_g)}{2}cos(\eta) \end{cases}\ . \tag{5.20}$$

In the cylindrical part, the y component of the point's normal direction on the cross-section edge is $n_y = y$.

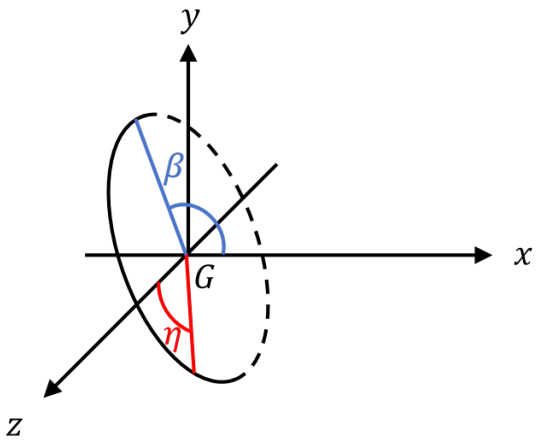


Figure 5.11: A cross-section contains the Gauss point in the local coordinate system.

Because the pressure gradient is only in the y direction and the cylinder is symmetric with respect to the xoy plane, the force integration's z component should be zero. On the contrary, it is the opposite. The unit length force f_{gy} in the global system is determined by:

$$f_{gy}(\xi_g) = -\int_0^{2\pi} \frac{D(\xi_g)}{2} \left[sin(\eta)sin(\beta)\right]^2 \frac{D(\xi_g)}{2} d\eta = -\frac{\pi D^2(\xi_g)}{4} sin^2(\beta), \xi_g \in [0, \xi_{cyl}] \; or \; (\xi_{rope}, 1]. \quad (5.21)$$

The pressure force is towards the negative direction, which makes sense. For the taper segment, we only need to multiply a $cos(\alpha)$ in the equation:

$$f_{gy}(\xi_g) = -\frac{\pi D^2(\xi_g)}{4} sin^2(\beta)cos(\alpha) \; , \; \xi_g \in (\xi_{cyl}, 1] \; or \; (\xi_{cyl}, \xi_{rope}] \; . \quad (5.22)$$

The same method can determine the unit length force under the pressure field $p(\mathbf{X}) = z$. In that case, the central axis is in the global xoz plane, and β is the inclined angle in the same plane. β is easy to obtain since the geometry is known. With β, we can get the analytical unit length force, including the semicircle cylinder. Now, we can run the validation code. The number of sampling points on the cross-section edge is $n_{cr} = 64$.

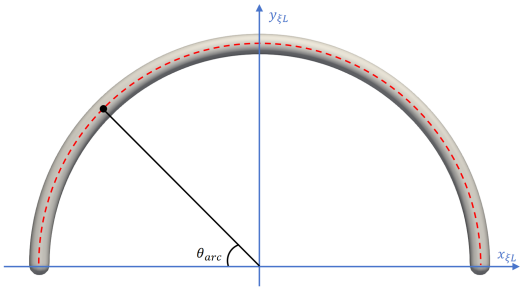


Figure 5.12: Schematic of the solar system of the semicircle cylinder. The black dot represents the sampling points.

For the semicircle cylinder, we define 15 sampling points on the central axis to analog Gauss points, evenly dividing the entire body into 16 segments. The beginning and end of the central axis are not

considered, since the Gauss integration never uses the values of the integration limits. The central axis is in the xoy plane. We use the angle in the solar coordinate system to denote the sampling points, so $\theta_{arc} = n\pi/16$ is the n^{th} sampling point's coordinate, as shown in Figure 5.12. The pressure field is $p(\mathbf{X}) = y$, and $L_{sim} = 64$. Appendix H provides the points' coordinates and numerical and analytical results. The comparison diagram is shown in Figure 5.13.

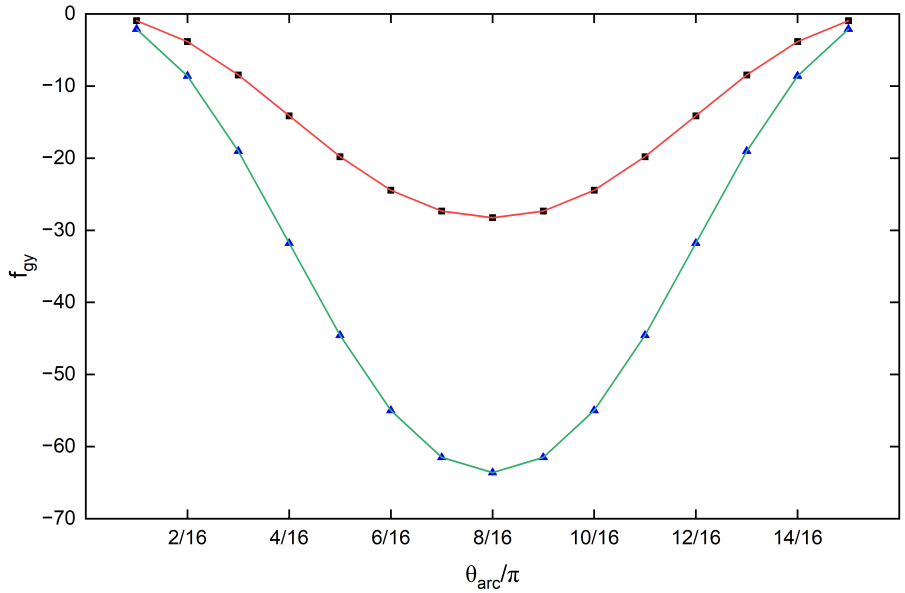


Figure 5.13: Comparison between analytical and numerical results of the semicircle cylinder. $\beta = \pi - \theta_{arc}$. The red line is the analytical results with $D_{sim} = 6$; the black boxes are its numerical approximation. The green line is the analytical results with $D_{sim} = 9$; the blue triangles are its approximation. The numerical force's z component is at $O(10^{-14})$.

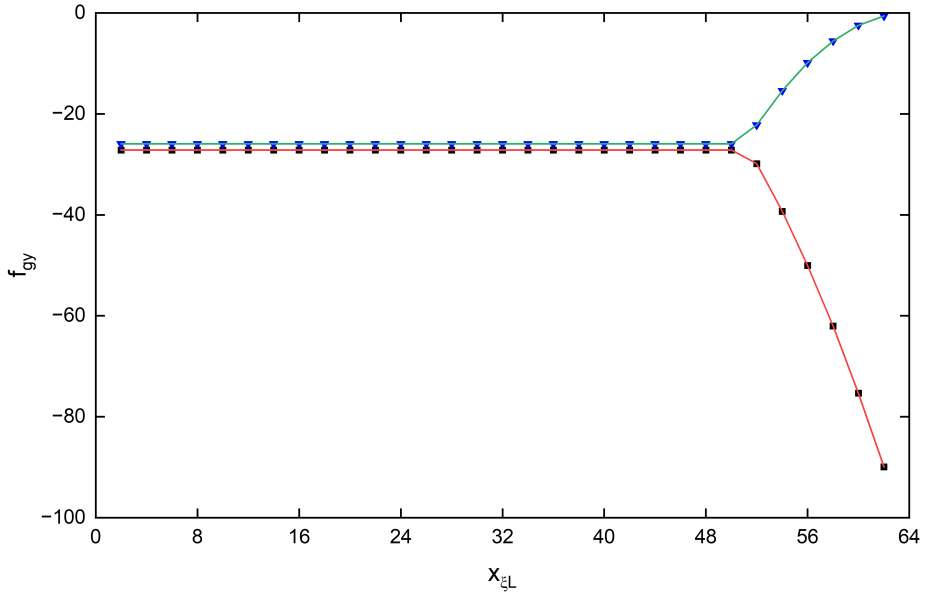


Figure 5.14: Comparison between analytical and numerical results of the cylinder with a tapered end. $L_{sim} = 64$. $\xi_{cyl} = 0.8$. $D_{sim} = 6$. The red line is the analytical results with $D_{end} = 12$ and $h_\xi = 0.2$; the black boxes are its numerical approximation. The green line is the analytical results with $D_{end} = 0$ and $h_\xi = 0.3$; the blue boxes are its approximation. $x_{\xi L}$ is the longitudinal coordinate of the sampling point. The numerical force's z component is at $O(10^{-14})$.

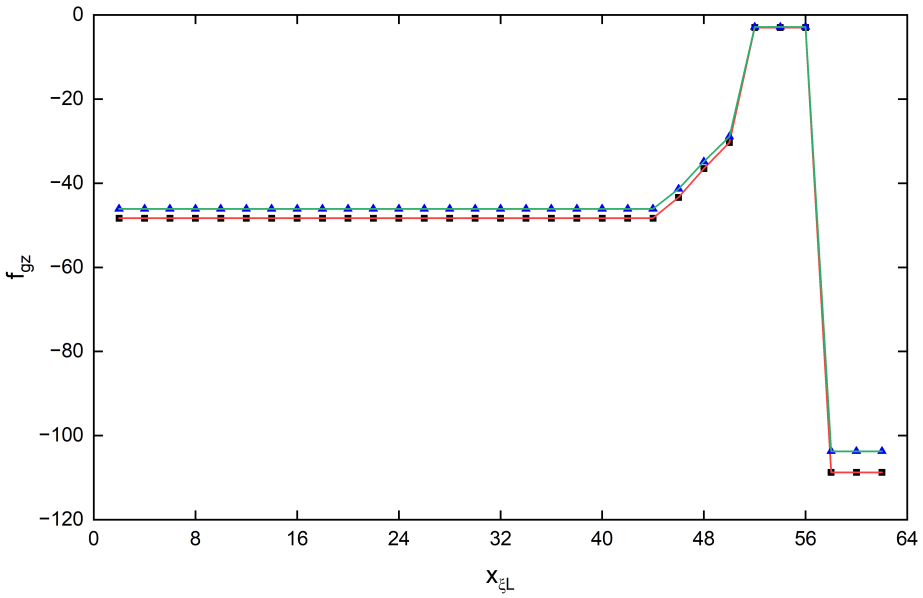


Figure 5.15: Comparison between analytical and numerical results of the cylinder with a parachute end. $L_{sim} = 64$. $\xi_{cyl} = 0.7$, $\xi_{cone} = 0.8$, and $\xi_{rope} = 0.9$. $D_{sim1} = 8$, $D_{sim2} = 2$, and $D_{sim3} = 12$. The red line is the analytical results with $h_\xi = 0.2$; the black boxes are its numerical approximation. The green line is the analytical results with $h_\xi = 0.3$; the blue triangles are its approximation. The numerical force's y component is at $O(10^{-14})$.

We define 31 sampling points on the central axis for the cylinder with a tapered end. The straight central axis is in the xoy plane. The pressure field is $p(\mathbf{X}) = y$. The characteristic length $L_{sim} = 64$. $\beta = \pi - arctan(1/h)$. Two cylinders with various parameters and h_ξ are tested. The sampling points' coordinates in the local system in Figure 5.4 (blue one) and force results are provided in Appendix H. The comparison diagram is shown in Figure 5.14.

For the cylinder with a parachute end, we define 31 sampling points. The straight central axis is in the xoz plane. The pressure field is $p(\mathbf{X}) = z$. The characteristic length remains the same. $\beta = \pi - arctan(1/h)$. Two cylinders are tested with different h_ξ. Appendix H provides the details of the results. The comparison diagram is shown in Figure 5.15.

The maximum error rate of all cases is 0.037%, corresponding to a point on the cylinder with the tapered end and $D_{end} = 0$, determined by $(f_{gy,sim} - f_{gy})/f_{gy}$. We then combined the code into the frame of the FSI code, which can be found in Appendix G. A simple test is also conducted to ensure the code can use actual Gauss points. In conclusion, the pressure force integrator is validated.

5.3. FSI solver validation

5.3.1. Methodology

The FSI solver bridges the flow solver and the solid structure solver. A solver is available for a 2D case. The coupling solver based on the Interface Quasi-Newton method is combined in the code. The flow solver receives and delivers data with three dimensions, while the solid structure requires two-dimensional data in the y and z directions. Hence, modification is necessary.

Some vectors and matrices' dimensions in the original FSI solver are first extended to three dimensions, allowing it to connect with the 3D flow solver. Our pressure integrator measures the forces in the y and z directions on Gauss points from the fluid field that the flow solver determines. The external forces are in the global coordinate system and delivered directly to the solid structure solver. The solid structure solver determines the displacement of the parametric control point in the corresponding directions. The solver returns the global displacement after multiplying L_{sim} by the parametric one. The new displacements in the y and z directions are combined with a zero x component to form a 3D vector. With $D(\xi)$, the parametric body code can generate a spatial curve using the B-spline curve.

The FSI coupling is solved by the implicit method. Define an operator $S(f_k)$ representing our solid structure solver, which receives the external pressure force f_k and returns the control point displacement d_k. Define an operator $F(d_k)$ representing the flow solver combined with the pressure force integrator, which receives the structure displacement d_k, generates a new fluid field, and returns the hydrodynamic force F_k. d_k and f_k are vectors. In the coupling process, the equation system 5.23 should be fulfilled simultaneously.

$$\begin{cases} S(f_k) = d_k \\ F(d_k) = f_k \end{cases} . \tag{5.23}$$

k is the number of iterations at each time step. The system can be treated as a fixed point problem[26]: $S(F(d_k)) = d_k$. We can use the Newton method to find the root of the equation by iterations. Define a residual function in an iteration step: $R(d_k) = S(F(d_k)) - d_k$. The goal is to find the root to make $R(d_k) = 0$.

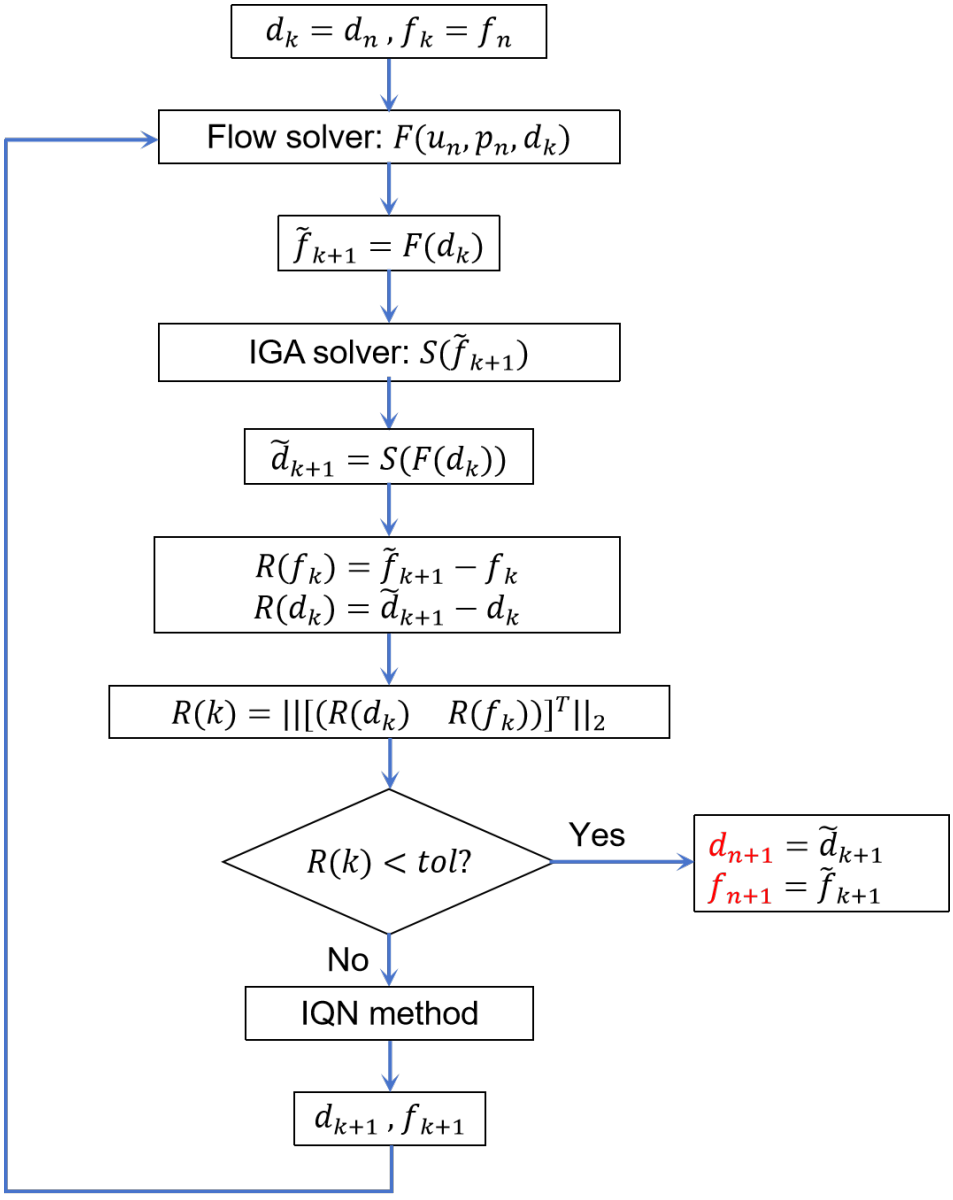


Figure 5.16: Flow chart of the coupling solver in the FSI code at time step $n + 1$.

Figure 5.16 provides the flow chart of the solving process at the iteration step k. Assume we are solving d_{n+1} and f_{n+1}. d_n and f_n are known. n denotes the time step. Use $d_k = d_n$ and $f_k = f_n$ as the initial values. Next, substitute d_k into the flow solver to obtain the external force $\tilde{f}_{k+1} = F(d_k)$. Then, substitute $\tilde{f}_{k+1}$ into the solid structure solver to obtain the displacement $\tilde{d}_{k+1} = S(F(d_k))$. The force and displacement residuals are

$$R(f_k) = \tilde{f}_{k+1} - f_k \ , \tag{5.24}$$

$$R(d_k) = \tilde{d}_{k+1} - d_k \ . \tag{5.25}$$

The overall residual of iteration k is determined by the norm of vector residual with $R(d_k)$ and $R(f_k)$:

$$R(k) = || \left[R(d_k) \quad R(f_k) \right]^T ||_2 \ . \tag{5.26}$$

All displacement and force residuals are in the square brackets; the dimension is the sum of the quantities of the two residuals. Define a tolerance tol. If $R(k) < tol$, the solutions converge. $\tilde{d}_{k+1}$ and $\tilde{f}_{k+1}$ are the precise values on the time step $n+1$. If $R(k+1) > tol$, the solutions are incorrect. Thus, use the IQN method to determine a new d_{k+1} and f_{k+1} for the next iteration step. Then, substitute d_{k+1} and f_{k+1} into the flow solver again to start the iteration step $k+1$. The fluid velocity and pressure fields revert to u_n and p_n. The difference between the classical and the Interface Quasi-Newton methods is provided in the subsection 2.3.7. .

5.3.2. Validation

We select the experiment of a clamped-free cylinder with a tapered end, conducted by Païdoussis[43], as the reference to validate the FSI solver. The FSI solver imports the flow solver and the solid structure solver. It is a complete code for determining the instability of a flexible cylinder in axial flow, taking into account flow-structure coupling. We expect to obtain various unstable modes of the cylinder, and the results should agree with the experiment.

We first introduce the experiment setting. The shape A in Figure 2.15a is selected as the tested tapered end. We regard it as a cone, which the parametric body code can generate. The length of the tapered end is $L_{end} = 31 D_{cyl}/17$, as measured from the figure. The experiment's ratio determines the mass ratio of the simulation β_{sim}:

$$\frac{\rho_{wat}}{\rho_{wat} + \rho_{cyl}} = 0.46 \Rightarrow \beta_{sim} = \frac{\rho_s}{\rho_{sim}} = \frac{\rho_{cyl}}{\rho_{wat}} = 1.157 \ . \tag{5.27}$$

ρ_s and ρ_{sim} are the cylinder and water density in the simulation. $\rho_{sim} = 1$. ρ_{cyl} and ρ_{wat} are those densities of the experiment. $\rho_{wat} = 1000 \ kg/m^3$. The mass ratio β_{sim} is used in the dynamic solver of the solid structure solver. We still neglect gravity and consider only hydrodynamic force.

The experiment's cylinder length is $L_{cyl} = 0.391 \ m$, and the diameter is $D_{cyl} = 0.166 \ m$. The bending stiffness is $EI = 6.39 \times 10^{-3} \ N \ m^2$. The unit length cylinder mass is $m = 0.25 \ kg/m$[44]. In the code, experiment parameters are converted from imperial units.

The critical velocity in the instability diagram 2.15a is expressed by the dimensionless velocity u:

$$u = \left(\frac{\rho_{wat} \pi {D_{cyl}}^4 / 4}{EI} \right)^{1/2} U_{exp} L_{cyl} \ . \tag{5.28}$$

Each tested case's experiment far-field velocity U_{exp} can be obtained to determine the corresponding Reynolds number Re:

$$Re = \frac{\rho_{wat} U_{exp} D_{cyl}}{\mu_{wat}} \ .$$

We can also obtain the Cauchy number by EI directly:

$$Ca = \frac{EI}{\rho_{wat} U_{exp}^2} / \frac{\pi {D_{cyl}}^4}{64} \ . \tag{5.29}$$

The similarity criteria the simulation should obey are Ca and Re. We substitute them into the code and determine μ_{sim} and $E_{sim}I_{sim}$:

$$\mu_{sim} = \frac{\rho_{sim} U_\infty^2 D_{sim}}{Re} \ , \ E_{sim}I_{sim} = (Ca \cdot \rho_{sim} U_\infty^2) \frac{\pi {D_{sim}}^4}{64} \ . \tag{5.30}$$

In the code we have: $\rho_{sim} = 1$ and $U_\infty = 1$. The far-field axial flow velocity U_∞ is in the x direction in the simulation. The bending stiffness for the solid structure solver is $E_{para}I_{para} = E_{sim}I_{sim}/L_{sim}^3$, where $L_{sim} = 192$ is the characteristic length of the simulation and central axis length of the cylinder with a tapered end. For a cylinder with a parachute end, replace D_{sim} with D_{sim1}. We have $D_{sim} = D_{sim1} = 7.56$. We determine ξ_{cyl} before obtaining D_{sim}:

$$\xi_{cyl} = \frac{L_{cyl}/D_{cyl}}{L_{cyl}/D_{cyl} + L_{end}/D_{cyl}} \ . \tag{5.31}$$

$D_{end} = 0$ in the validation case. The diameter of the cylindrical part is:

$$D_{sim} = \xi_{cyl} L_{sim} \times \frac{D_{cyl}}{L_{cyl}} \ . \tag{5.32}$$

With D_{sim} and ξ_{cyl}, we can define the diameter function $D(\xi)$ to generate the cylinder with a tapered end. Figure 5.17 shows the numerical fluid field.

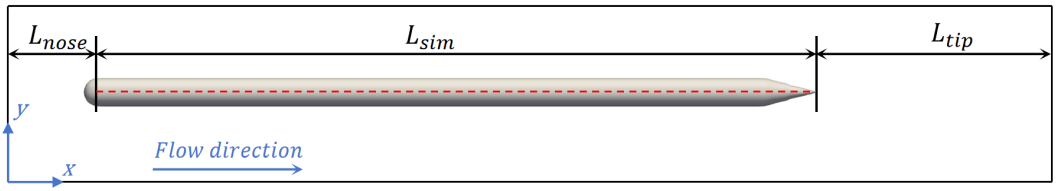
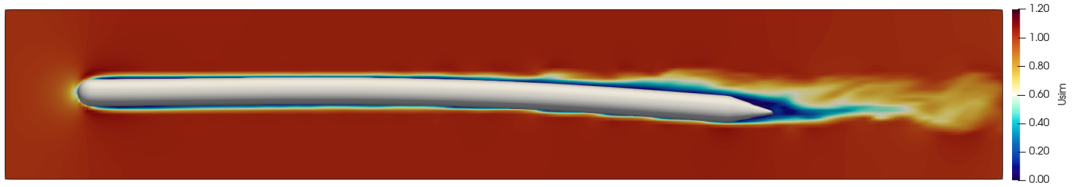
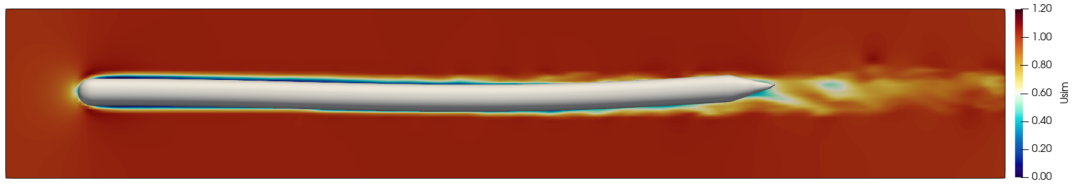


Figure 5.17: Side view of the fluid field for the validation. The red dashed line represents the central axis generated by the B-spline curve. The ghost grids for the boundary condition are not shown in the figure.

The fluid field is a cuboid. The length, width, and height are $L_l = 280$ and $L_w = L_h = 48$. The central axis of the cylinder with a tapered end is straight and parallel to the x direction. The coordinate of the axis' nose is $(x, y, z) = (24.5, 24.5, 24.5)$ to prevent the singularity. Thus, $L_{nose} = 24.5$, and $L_{tip} = L_l - L_{nose} - L_{sim}$. The far-field flow direction is the x direction. Then, we use a third-degree B-spline curve with $numE = 12$ elements and $n_g = 48$ Gauss points to generate the parametric mesh on the solid structure.



(a) Fluid field with a cylinder in divergence in the xoy plane. $u = 4$. $t_{conv} = 30.0$.



(b) Fluid field with a flutter cylinder in the xoy plane. $u = 5$. $t_{conv} = 28.8$.

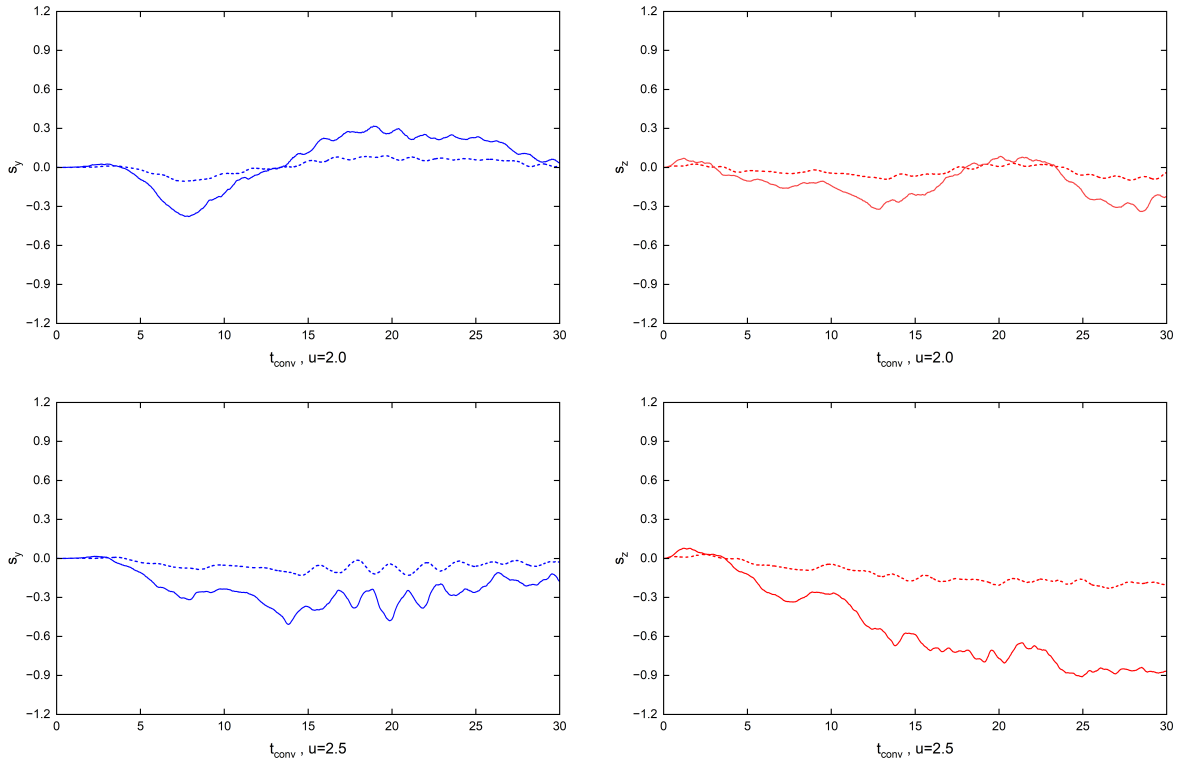
Figure 5.18: Velocity fields of cylinder in different instabilities.

Six dimensionless velocities $u \in [2.0, 2.5, 3.0, 4.0, 4.5, 5.0]$ have been tested. A half sphere end ($\xi_{cyl} = 1$), the H shape shown in Figure 2.15a, has been tested additionally when $u = 5$. The simulation duration is 30 convective time units, ensuring the fluid and structural displacements fully develop. We gave a poke $q_p = 0.01$ in the z direction when $t_{conv} \leq 0.2$ to trigger the initial instability and prevent the critical stability. The initial condition should not affect the system's stability. The velocity fields of divergence (buckling) and flutter cylinders are shown in Figure 5.18a and 5.18b. The flow solver automatically determines the boundary layer. The boundary layer grows slowly and smoothly in the upper part of the cylinder, while it develops significantly close to the downstream end.

Define s_y and s_z as the relative displacements ($displacement/D_{sim}$) in two directions of a point on the body's central axis. We monitor the time-varying displacements at the midpoint ($\xi = 0.5$) and the downstream end ($\xi = 1.0$). With two monitoring points and the upper boundary, we can infer the instantaneous cylinder shape to distinguish different types of instabilities. We mainly observe the displacement when $t_{conv} > 20$, ensuring the instability fully develops. We expect to obtain an approximately periodic state of divergence or flutter. The group of figures 5.19 provides the resultant diagrams. The validation code can be found in Appendix F. Appendix I provides more figures about the cylinder shape. The parameters of each test case are shown in Table 5.1. We can finally infer the range of the critical velocities of divergence and flutter to generate a figure similar to the figure 2.15a.

Table 5.1: Parameters of validation cases.

$L_{sim} = 192$, $D_{sim} = 7.557$, $\xi_{cyl} = 0.928$				
u	Re	Ca	$E_{sim}I_{sim}$	$E_{para}I_{para}$
2.0	12816	2225	356148	0.05032
2.5	16020	1424	227935	0.03220
3.0	19224	989	158288	0.02236
4.0	25632	556	89037	0.01258
4.5	28837	439	70350	0.00994
5.0	32041	356	56984	0.00805



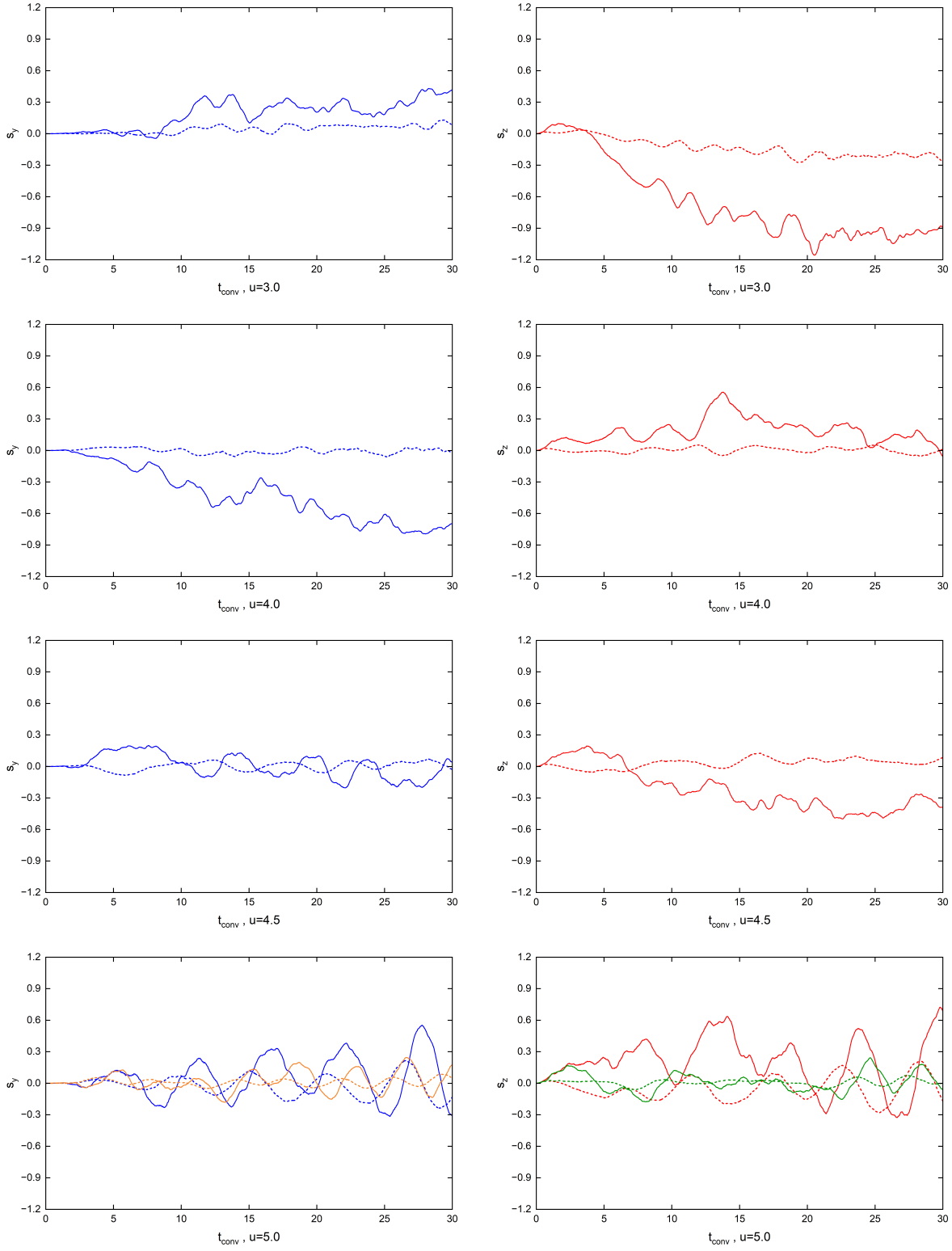


Figure 5.19: Displacements diagram over t_{conv}. The dashed and solid lines are the midpoint and endpoint displacements.

When $u = 2.0$, there is no clear indication of static divergence or flutter instability, as defined in Figure 2.5. In the y direction, the displacement gradually reduces to zero. In the z direction, the displacement's neutral position shifts from zero over a small distance. The phases of displacement at the midpoint and downstream are similar. The overall deflection is tiny. We conclude that the critical velocity of the

divergence instability is $u_{cd,sim} \geq 2$ (the cylinder always diverges first, then flutters[43]).

When $u = 2.5$, a clear divergence (buckling) instability is observed in the simulation animation. In the time variation diagram, deflections in two directions develop a non-zero neutral position and fluctuate slightly around it. In the z direction, the deflection of the neutral position is around a D_{sim}, which matches the experimental observation[43]. In each plane xoz and xoy, the displacements on the mid and end points deflect in the same direction (positive n_y and negative n_z), indicating the cylinder shape is similar to a first-mode bending, which is confirmed by Figure I.1. The critical velocity of the divergence should be $u_{cd,sim} < 2.5$.

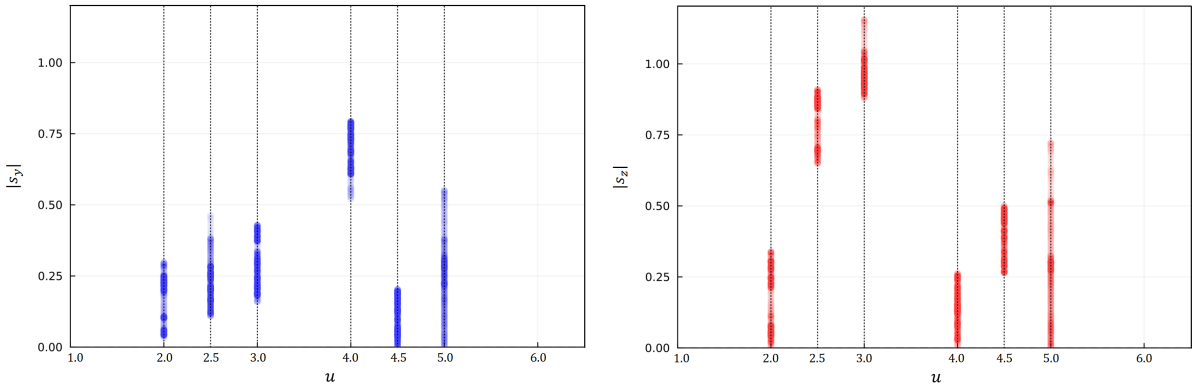
When $u = 3.0$, the cylinder buckles in two directions and maintains non-zero neutral positions, which indicates the divergence instability.

When $u = 4.0$, we can still observe a clear static divergence, especially in the xoy plane. The amplitude (deflection of the neutral position on the end) of the buckling reduces in the z direction, which matches the experiment[43]: the transition from divergence to flutter "involved a gradual return of the cylinder to its position of rest along the x-axis, before further increase in the flow velocity resulted in unstable oscillation"[43].

When $u = 4.5$, the cylinder starts to slightly flutter in the y direction but maintains divergence in the z direction. We infer that the instability of the cylinder transforms at this point. Hence, the critical velocity of the flutter instability is $u_{cf,sim} \geq 4.5$.

When $u = 5.0$, the cylinder is fluttering. The neutral position of deflection shifts back to zero. Figure 5.19 shows that the phases of displacements on the mid and end points are different. At the same t_{conv}, we can find a positive displacement on the mid-point but a negative one on the end, which indicates that the flutter mode should be at least the second. Thus, the critical velocity of flutter is $u_{cf,sim} < 5.0$.

The absolute displacement ranges of the downstream end in the y and z directions when $t_{conv} > 20$ under various u are provided in Figure 5.20. The dots representing the magnitudes of displacements form the blue and red bands. Each band contains all resultant displacements in the time duration mentioned before. It is similar to compressing curves in Figure 5.19 from 2D to 1D. The denser part of the band indicates that the downstream end reaches such displacements frequently. We can regard the midpoint of the band as the approximated neutral position's magnitude when the cylinder is in divergence instability. Figure 5.20 indicates that the downstream end has small vibrations around the non-zero neutral position when the cylinder is diverged. The downstream end oscillates around the zero neutral position when the cylinder flutters. The maximum displacement indicates that the cylinder fulfills the small deflection assumption mentioned in Section 3.2.



(a) Displacement range in the y direction.

(b) Displacement range in the z direction.

Figure 5.20: Absolute displacement ranges at the downstream end when $t_{conv} > 20$ under various u. The cylinder has unclear instability when $u = 2.0$ and has flutter instability when $u = 5.0$.

With corresponding relations between instabilities and u, we can draw a stability diagram 5.21 corresponding to u, similar to Figure 2.15a. The critical velocity ranges determined by the simulation can match the experimental results.

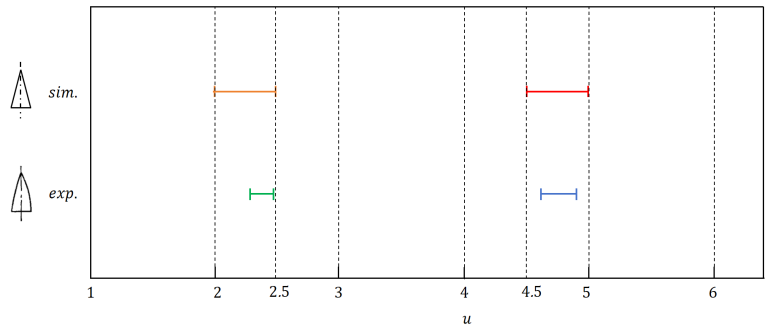


Figure 5.21: Ranges of divergence and flutter critical velocities. Green and orange horizontal I are divergence ranges from the experiment and the simulation. The flutter ranges from the experiment and the simulation are blue and red horizontal I.

In summary, the FSI solver is quantitatively validated. The code is sufficient for our research.

5.4. Discussion

5.4.1. Pressure force integrator

In the pressure force integrator, we use adjacent grid points around the sampling point on the edge of the cross-section to approximate the pressure. It is possible to pick up a point in the body where the pressure field is not solved. Thus, we develop a code to make the sampling point move a small distance δ outside the body surface. Equation 5.14 becomes:

$$\mathbf{X}_{ci,1} = \left(\frac{D(\xi_g)}{2} + \delta \right) \mathbf{n}_{B1} + \vec{OG} \, . \tag{5.33}$$

Some δ has been tested. We provide a cylinder with a parachute and $\delta = 0.01$ in Appendix H. The error is generally more significant than the case with $\delta = 0$, so we set $\delta = 0$ in the version code used. A possible reason is that the BDIM has a boundary region that prevents selecting a grid point in the body. Besides, $\delta / D(\xi_g)$ becomes significant when D_{ξ_g} is small. We conclude that δ is not necessary.

5.4.2. FSI solver

We observe a tiny flutter when the downstream end is blunt, as shown in the last row of the figures group 5.19. The blue and red represent the cylinder results with a tapered end. The brown and green represent the results of the cylinder with a half-sphere (blunt) end. The experimental result shows no flutter at this flow velocity. It could be because of the insufficient grid resolution. We only use no more than eight grids to represent the cylinder diameter. However, the flutter amplitude is significantly smaller than the amplitude of the cylinder with a tapered end. We still conclude that the code can distinguish the effects of different end shapes on the instability.

The original 2D FSI solver was developed based on an old version of WaterLily. The original code cannot use a GPU to accelerate the computation. Our code is modified from the original, so we can only run it on a CPU. Although we use a 16-core AMD 7945HX CPU, a single validation case takes about a day to execute. We set L_w, D_{sim}, and D_{sim1} as large as we can, considering the time consumption. We have $L_w / D_{sim} > 6$, indicating that our field is wider than the experiment's tube shown in Figure 2.13. Hence, further development to apply GPU support to our code in the future is meaningful. We can also upgrade a more complex $D(\xi)$ to fit the streamlined end.

When we validate the parametric body code, the airship requires the fluid domain width to be 12 times the diameter. You may find that the fluid field width is about 6 times the cylinder's diameter for FSI validation. One reason is the computational cost limitation. Another reason is that the cylinder is more slender with less cross flow. The code has been validated with a thinner flow field in the end.

6

Investigation

The FSI solver has been validated. The investigation can proceed following the research matrix. We test the effect of various parameters on the instability of the flexible cylinder in axial flow.

6.1. Length of the cylinder

6.1.1. Numerical simulation

The parameters from Optics 11 are used to determine the simulation parameters. Constant Re and Ca can be determined by equations 2.30 and 5.29 with $U_{real} = 5\ knots = 2.572\ m/s$, $\rho_{sea} = 1026\ kg/m^3$, $\mu_{sea} = 1.22 \times 10^{-3}\ Pa \cdot s$, $E_{arr} = 1.2 \times 10^7\ Pa$, and $D_{arr} = 0.03\ m$[55][59]. μ_{sim} and $E_{sim}I_{sim}$ can be determined by equations 5.30. The shape of the array cable is a cylinder with a tapered end, so $D_{end} = 0$. The cylinder diameter is constant $D_{sim} = 7.557$. The length of the taper end is still $L_{end} = 31D_{cyl}/17$. The buoyancy is neutral, so $\beta_{sim} = 1$. The upper boundary condition is clamped or pinned. The only changed parameter during simulations is the aspect ratio AR, which is defined as

$$AR = \frac{L_{sim}}{D_{sim}}\ . \tag{6.1}$$

Then, ξ_{cyl} is determined with AR:

$$\xi_{cyl} = \frac{AR}{AR + 31/17}\ . \tag{6.2}$$

The characteristic length of the simulation and the body's central axis length are:

$$L_{sim} = AR \cdot D_{sim}/\xi_{cyl}\ . \tag{6.3}$$

The degree of the B-spline curve is three. The number of elements is determined by (define an operator Int to find the closest integer):

$$numE = Int\left(\frac{12AR}{L_{cyl}/D_{cyl}}\right)\ . \tag{6.4}$$

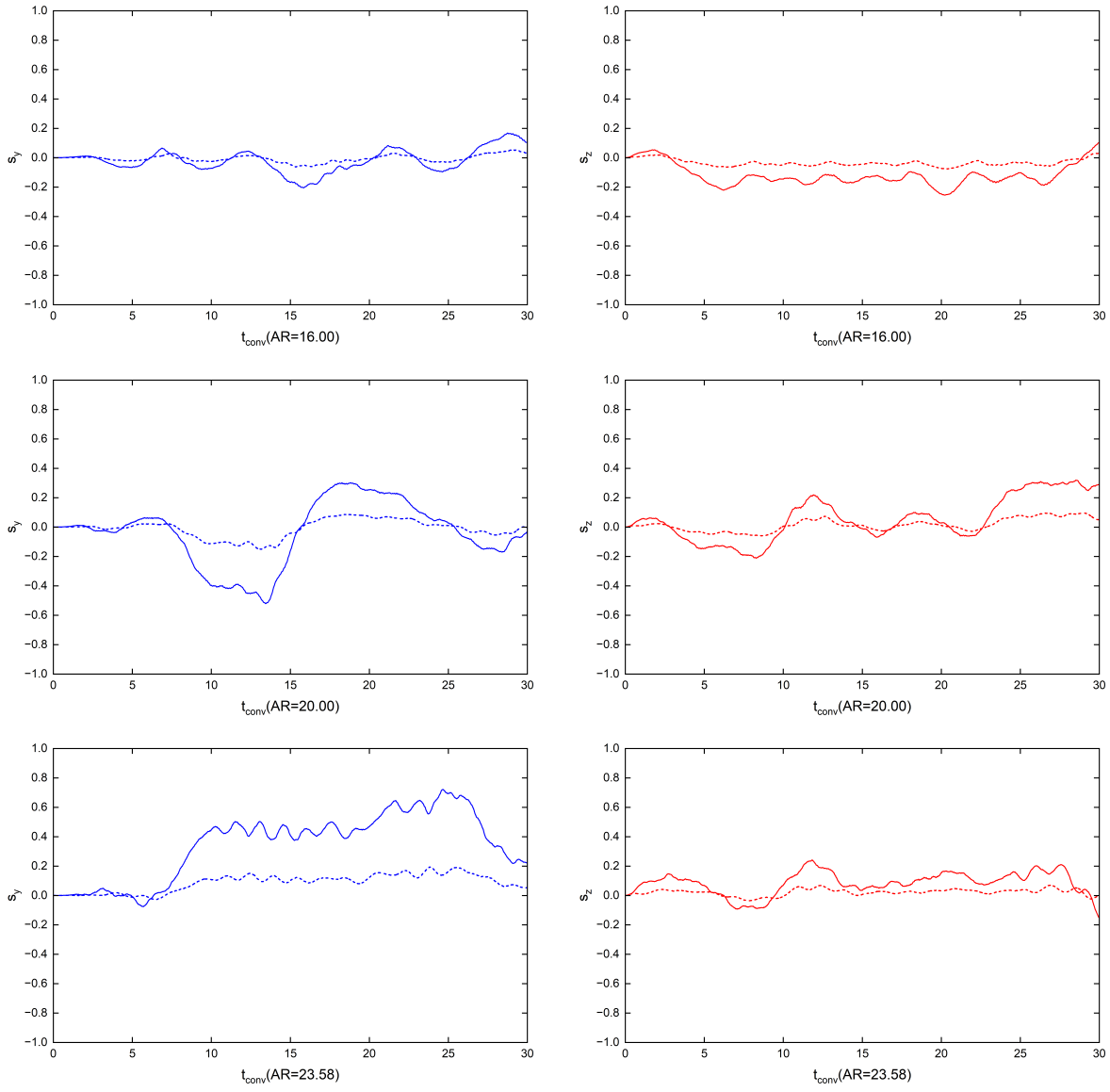
L_{cyl} and D_{cyl} are the cylinder length and diameter used in FSI validation. Equation 6.4 means a longer cylinder has more elements. The actual aspect ratio L_{arr}/D_{arr} is too enormous to simulate. Hence, we extend AR as long as possible to find the change pattern of the cylinder's instability with changing length. The parameters of each test case are shown in Table 6.1. Figure 5.17 shows the fluid field setting. We set $L_{tip} = 64$, and then determine the $L_l = L_{nose} + L_{sim} + L_{tip}$. L_l is the closest even number or multiple of 16. ($AR = 23.58$ represents the ratio from the validation case; decimals are

omitted. The corresponding $L_{sim} = 192$.) The other field dimensions remain the same. We still give an initial poke in the z direction when $t_{conv} \leq 0.2$.

Table 6.1: Parameters of length effect investigation cases.

Re=64890, Ca=1768				
AR	$numE$	ξ_{cyl}	L_{sim}	$E_{para}I_{para}$
16.00	8	0.89769	134.69197	0.11583
20.00	10	0.91644	164.91987	0.06310
23.58	12	0.92822	192.00000	0.04001
30.00	15	0.94270	240.48963	0.02035
40.00	20	0.95640	316.05938	0.00896
50.00	25	0.96481	391.62913	0.00471

The time-varying displacements s_y and s_z on the midpoint and downstream end, as used in the subsection 5.3.2, are shown in Figures 6.1 and 6.3. The code can be found in Appendix J. The figures of the clamped-free cylinder are first plotted.



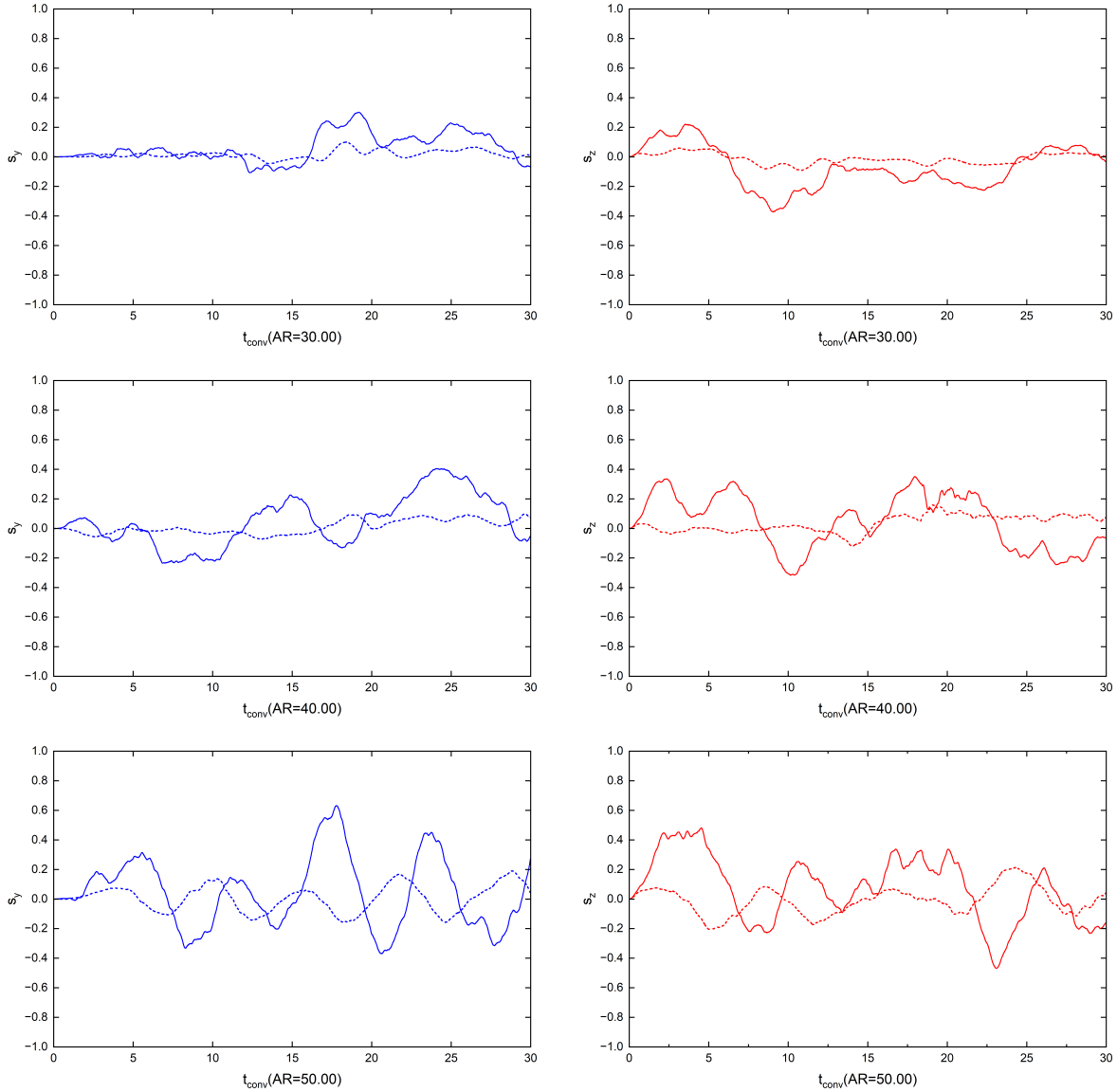


Figure 6.1: Time-varying displacements of the clamped-free cylinder with a taper end and various aspect ratios in axial flow. The solid and dashed lines represent the displacements at the downstream end and midpoint.

As shown in Figure 6.1, the deflection before AR reaches 50 is slight and 3D, making it challenging to observe the motion directly from the animation. Therefore, we use the time variation diagram for analysis. The clamped-free cylinder has some random and slight vibration when $AR = 16, 20$. This should be the stage before the static divergence (buckling). If we increase the cylinder length further, the cylinder will be in divergence instability when $AR = 23.58$: the midpoint and downstream end deflect to the same side simultaneously. Hence, the critical aspect ratio of divergence should be $AR_{cd} < 23.58$. When the aspect ratio reaches 30, the static buckling amplitude reduces to around zero. When $AR = 40$, the cylinder starts to flutter, so the critical aspect ratio of flutter should be $23.58 < AR_{cf} < 40$. When $AR = 50$, a clear pattern of flutter instability is observed, which is confirmed by the simulation animation.

The instability transition as we increase the cylinder length is similar to the transition as increasing the flow velocity around a clamped-free cylinder with a constant length[43][44]: the cylinder first diverges to a non-zero neutral position, then gradually moves back to the original zero neutral position ($y = 0$ and $z = 0$) as cylinder length increases, as shown in $AR = 23.58$ and $AR = 30$ in Figures 6.1; the flutter happens after that if we keep increasing the length. $AR = 30$ could be the transition point. Figure 6.2 shows the axis shape at different t_{conv}. The cylinder has a tiny vibration at this point in its first mode.

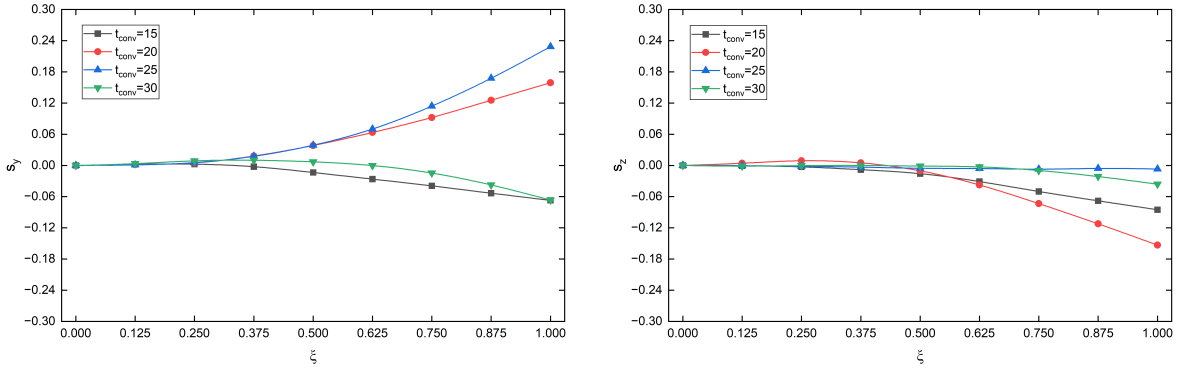
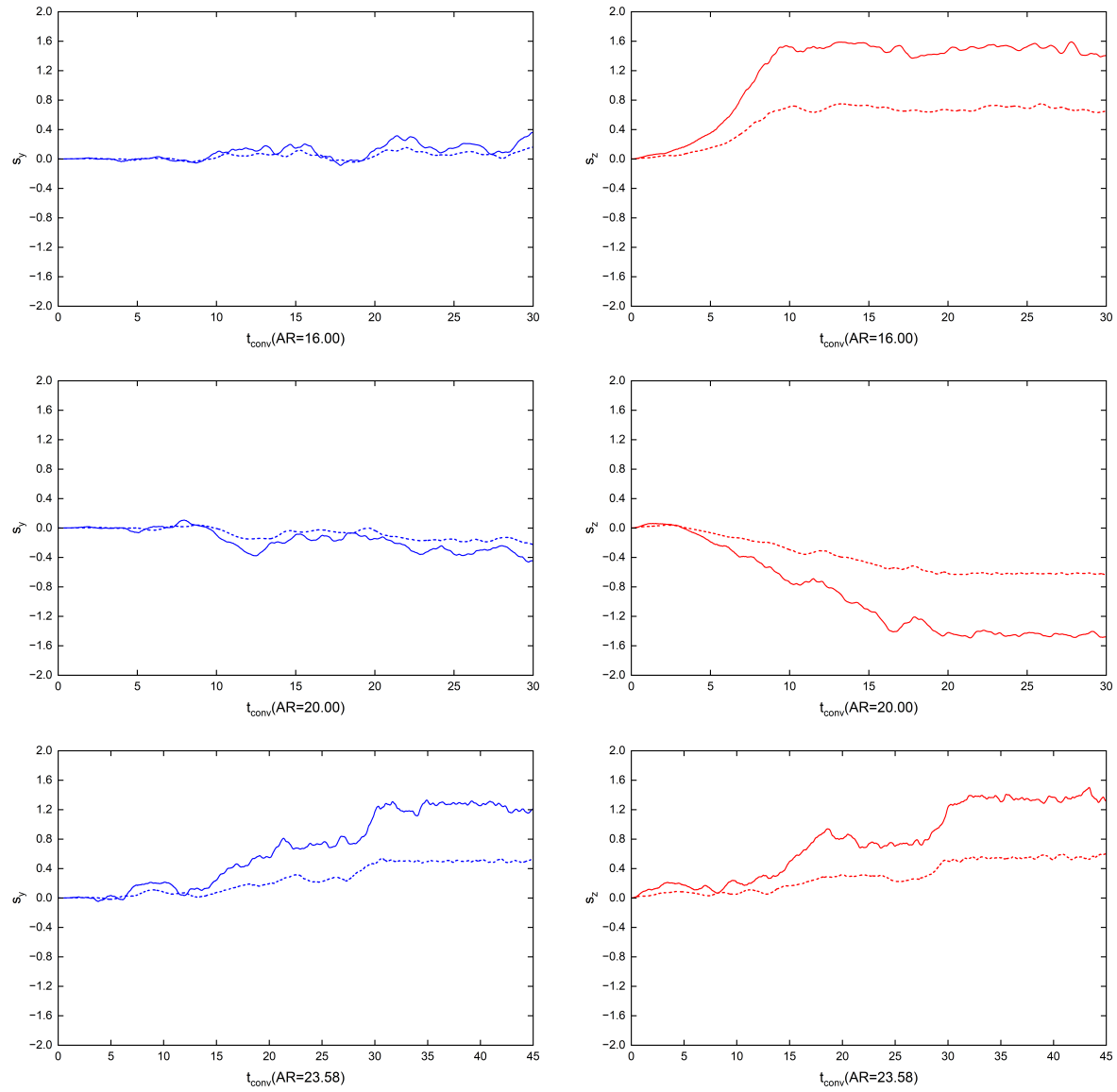


Figure 6.2: Shapes of the clamped-free cylinder's central axis in xoy and xoz planes. $AR = 30$. Symbols are sampling points.

The time-varying displacement figures of the pinned-free cylinder are plotted next. A pinned-free cylinder with $AR = 50$ is not tested since the cylinder has already fluttered when $AR = 40$.



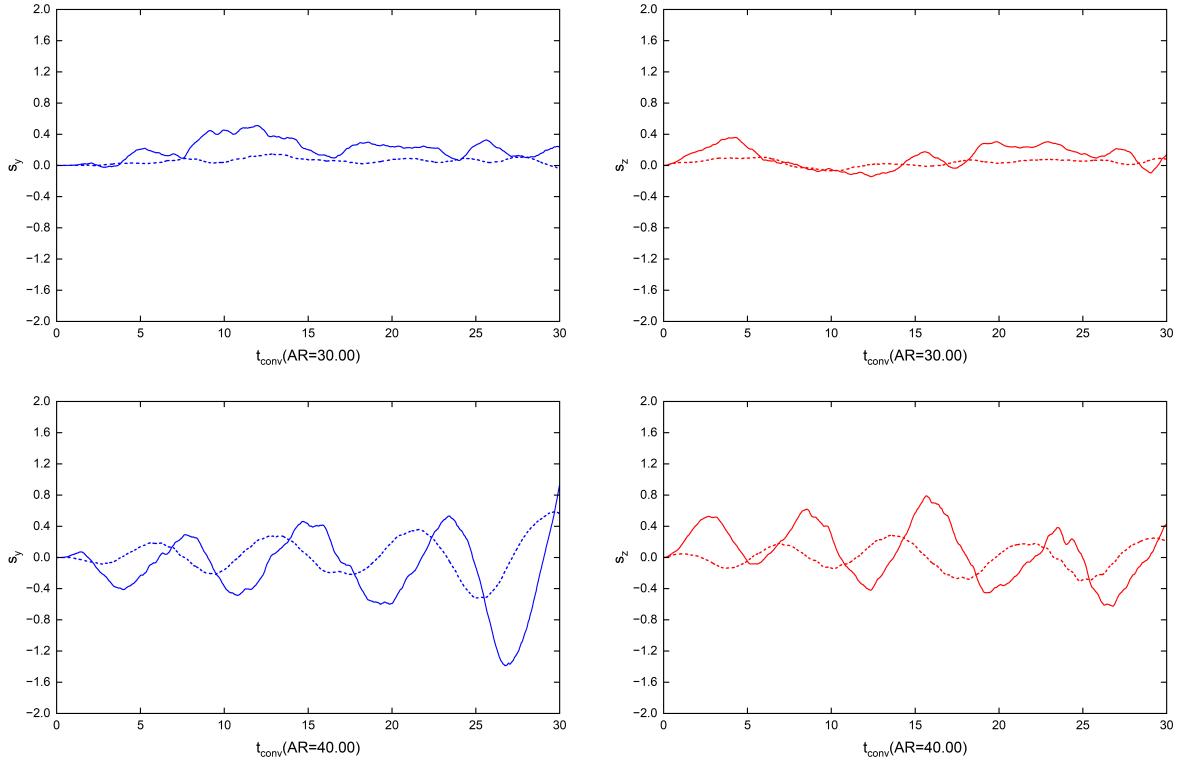
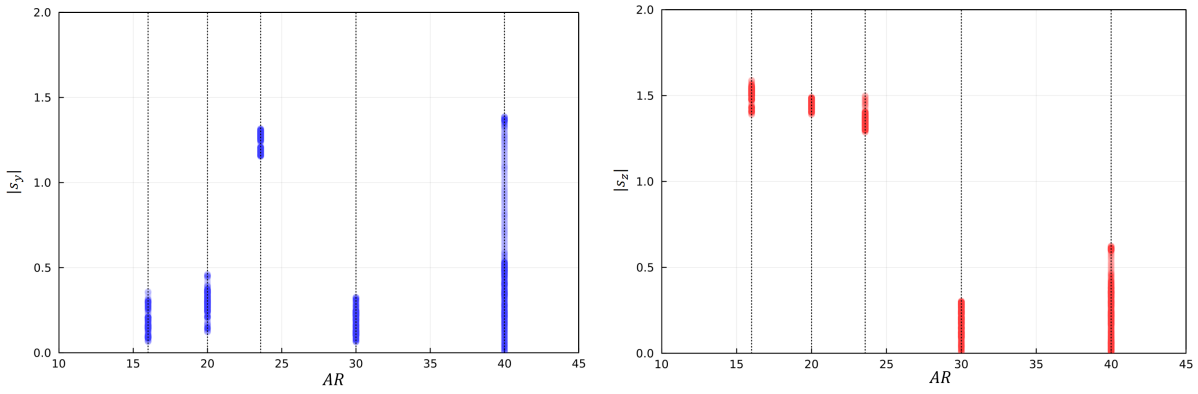


Figure 6.3: Time-varying displacements of the pinned-free cylinder with a taper end and various aspect ratios in axial flow. The solid and dashed lines represent the displacements on the downstream end and the midpoint.



(a) Displacement range in the y direction.

(b) Displacement range in the z direction.

Figure 6.4: Absolute displacement ranges at the downstream end in the last 10 convective time units. The downstream end oscillates around the zero position when $AR = 40$.

To find the steady state, the case with $AR = 23.58$ is simulated for 45 convective time units, while the other case uses 30 time units. We mainly observe the displacement in the last 10 convective time units, ensuring the instability fully develops.

The maximum amplitudes of the clamped and pinned-free cylinder displacements are smaller than $0.8D_{sim}$ and $1.6D_{sim}$, respectively. An arbitrary point A on the central axis of the cylinder still has the relation $d_A/l_A < 1/10$, which fulfills the small deflection assumption mentioned in Section 3.2.

Similar to the figure 5.20, Figure 6.4 provides absolute displacement ranges of the downstream end in the y and z directions when $t_{conv} > 20$ with various aspect ratios. We can regard the midpoint of the color band as the approximated neutral position's magnitude when $AR < 40$. The denser part of

the color band indicates that the downstream end has small vibrations around the non-zero neutral position, if the cylinder is diverged. According to the figure 6.3, the cylinder flutters when $AR = 40$, so the neutral position here is zero.

The figures 6.3, 6.4, and K.1 show that despite the amplitude being tiny in the y direction, the pinned-free cylinder is in clear divergence (yawing) when $AR = 16$. The instability in the z direction develops immediately with a large amplitude, possibly because of the initial poke in this direction. We still can infer that $AR_{cd} < 16$. An apparent divergence instability develops in the y direction when $AR = 23.58$. $AR = 30$ could be the transition point between divergence and flutter. The deflection amplitude reduces to around zero. The cylinder completely flutters in its second mode when $AR = 40$, so we have $23.58 < AR_{cf} < 40$. The axis shapes at various t_{conv} are shown in Figure 6.5. The central axis shapes of other cylinders with different AR are provided in Appendix K. The instability transition is similar to the behavior of the clamped-free cylinder, and it is more apparent:

- The magnitude of the neutral position of the divergence instability first increases, then reduces as the length increases, as shown in $AR = 16, 20, 23.58, 30$ in Figure 6.4a. It gradually returns to the original zero neutral position before the flutter instability, as shown in Figure 6.4b. There is no increasing phase in the z direction, possibly because the initial poke makes the pinned-free cylinder skip the instability development in this direction.
- If the cylinder length increases further, the flutter develops along the zero neutral position (cylinder's original central axis). The flutter amplitude grows from the tiny vibration around the zero position.

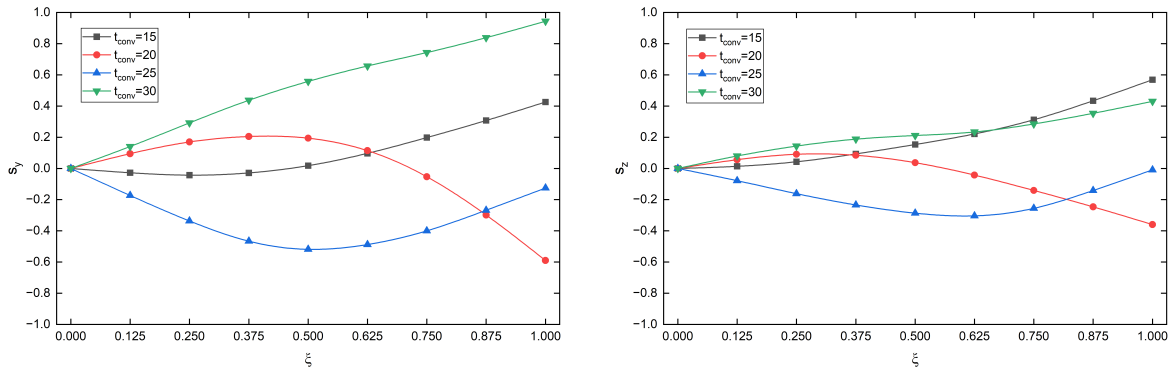
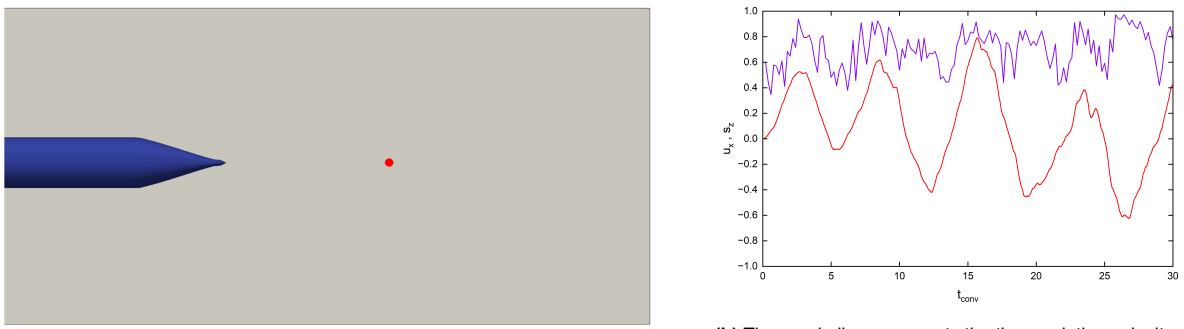


Figure 6.5: Shapes of the pinned-free cylinder's central axis in xoy and xoz planes. $AR = 40$. Symbols are sampling points.



(a) Position of the velocity sampling point (red dot).

(b) The purple line represents the time variation velocity u_x on the sampling point. The red line represents s_z.

Figure 6.6: Velocity sampling point and the comparison diagram between the flutter and the vortex-shedding frequencies.

We can also qualitatively compare the flutter frequency and the vortex-shedding frequency. Behind the cylinder's downstream end, we select a spatial point in the fluid field and monitor the time-varying

flow velocity in the x direction at this point. When a vortex passes the spatial point, the velocity fluctuates from the far-field velocity $U_\infty = 1$. We regard the fluctuating frequency as the vortex-shedding frequency. The velocity sampling interval is 0.2 convective time units. The Nyquist frequency is $5.0\ Hz$, so the maximum shedding frequency we can capture by FFT is lower than $2.5\ Hz$[29].

We select s_z on the pinned-free cylinder's downstream end ($\xi = 1$) when $AR = 40.00$ for comparison. The sampling rate of s_z is much higher, depending on the CFL condition. Thus, the band limit is not a problem. Figure 6.6b is the comparison diagram.

The Fourier transformation is the appropriate method for obtaining the power density diagram over frequencies[66][4] to find the dominant frequency of a periodic signal. The author attempted to generate the diagram using OriginLab[39]. The resultant diagram is meaningless. A possible reason is that the sampling rate of the velocity is too low. An improvement is required in the future: predefining the coordinate of the sampling point instead of randomly selecting a point from Paraview. And record the velocity data with the cylinder's displacement to share the same sampling rate.

We can still conclude that the flutter frequency is higher than the vortex-shedding frequency from Figure 6.6b.

6.1.2. Discussion

Increasing the cylinder length will cause the cylinder to diverge and then flutter. However, the deflections of pinned-free and clamped-free cylinders significantly reduce during the transition range between two instabilities, for example, $23.58 < AR < 40.00$. The cylinder is still unstable, but the oscillation amplitude is relatively small when $AR = 30.00$, which makes the structure's overall shape close to straight. Some industrial applications may benefit from the features mentioned above.

The cylinder becomes more unstable as the length increases, which is apparent. Combining the simulation and the experiment[36] observation, we conclude that increasing length (aspect ratio) will destabilize the structure after the transition point. Once the cylinder exhibits apparent fluttering, increasing the length cannot reduce the oscillation mode or stabilize the cylinder. Hence, the long array cable should be unstable under its working conditions.

In the literature review, we found that a cylinder with an intermediate length could be stable under various flow velocities. We should determine the critical velocities of cylinders with multiple aspect ratios to confirm this hypothesis. It is not easy to achieve this goal since the code can only run on the CPU at the moment. The case with $AR = 50$ took a week to execute. Many test cases should be taken to predict the critical velocity line, as shown in Figure 6.7. The time consumption is unacceptable.

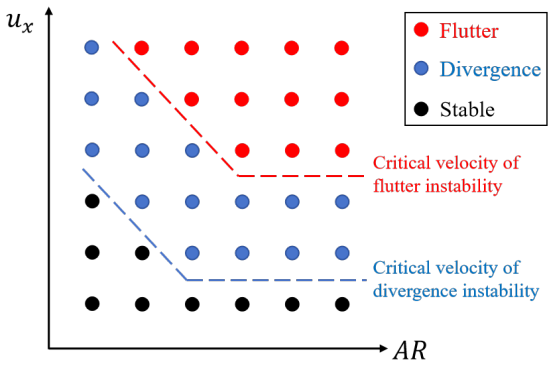


Figure 6.7: Method of predicting the critical velocity line. u_x is the far-field flow velocity.

The tested cylinders are much shorter than the actual array cable. However, we still infer that the array cable will flutter under its working condition since there is no indication of the stabilizing effect of increasing the cylinder length (aspect ratio) during our numerical investigation.

Increasing the length could lead to a higher mode flutter instability, since a longer cylinder is more similar to a string instead of a beam, and the constraint from the upper boundary becomes weaker at

the downstream end. The flutter amplitude could have a limited increase. The reason is that a larger amplitude indicates a larger angle between the far-field flow and the cylinder's tangential directions. This brings a larger restoring force caused by hydro-pressure and structural elasticity to prevent the amplitude from further increasing. The flutter cylinder shares some similarity with a forced spring-mass system.

In reality, the array cable should be in tension along its whole structure when it is in axial flow, because the vortex shedding behind the downstream end could create a lower pressure zone than that in the front, and the viscous force mainly contributes to the direction parallel to the axial flow. The situation differs from the counterintuitive assumption in the analytical model for a long and fluttered cylinder developed by Langre $et\ al.$[25]. Hence, the company can use experimental or numerical methods for intuitive investigation.

6.2. Upper boundary condition

In Chapter 6.1, we tested the cylinder with clamped and pinned upper boundary conditions. The clamped boundary has Neumann and Dirichlet conditions, and the pinned boundary only has the Dirichlet condition. Comparing the figures 6.1 and 6.3, it is apparent that the pinned-free cylinder is more unstable, which has a larger deflection amplitude and lower critical velocities. The rotational constraint on the upper boundary plays a vital role in the instability. Hence, the recommendation for the company is to strengthen the upper boundary, which is the cable connector between the tow and array cables.

The code can now only use the two boundary conditions mentioned before. It is possible to apply a spring boundary shown in Figure 2.1 in the future to make the simulation more general.

6.3. Reynolds number

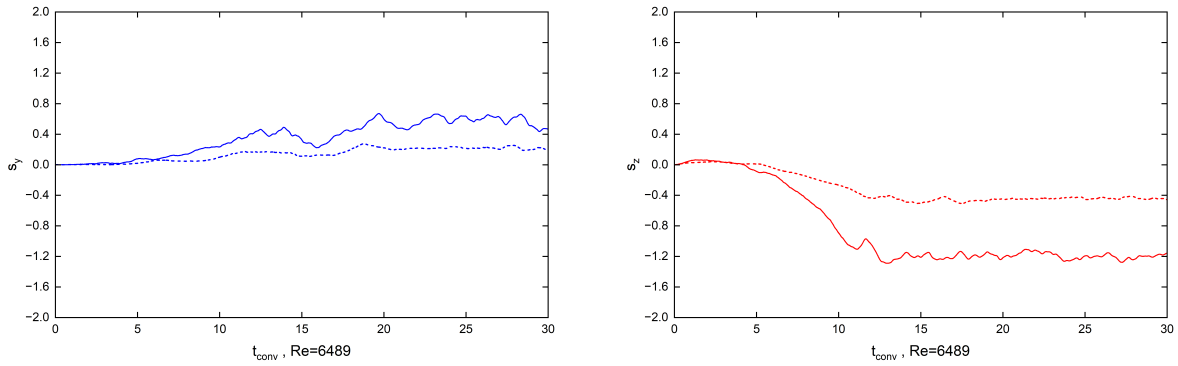


Figure 6.8: Time-varying displacements of the pinned-free cylinder with a taper end and $Re = 6489$. The solid and dashed lines represent the displacement at the downstream end and midpoint.

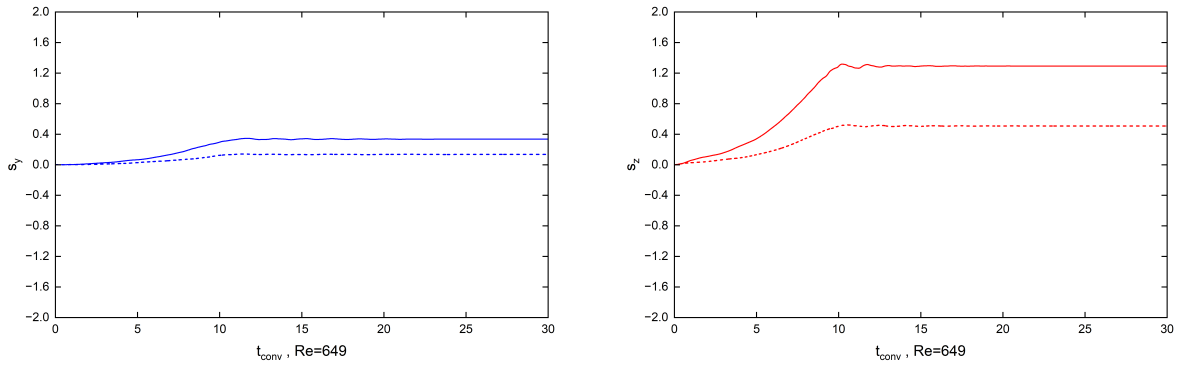


Figure 6.9: Time-varying displacements of the pinned-free cylinder with a taper end and $Re = 648.9$. The solid and dashed lines represent the displacement at the downstream end and the midpoint.

We use a pinned-free cylinder with $AR = 23.58$ to test the effect of the Reynolds number on the cylinder's instability. All the other settings are the same as the case shown in Chapter 6.1 and Table 6.1. The original Reynolds number is 64890, so two additional numbers, $Re = 6489$ and 648.9, are tested, and the viscous force on the cylinder remains negligible. We only change the Reynolds number, and the Cauchy number remains unchanged. It is similar to increasing the fluid viscosity, so the fluid field becomes more laminar. The time variation diagrams of s_y and s_z are shown in Figures 6.8 and 6.9.

Compared to Figure 6.3, the cylinder in the fluid field with lower Reynolds numbers is still in divergence. The amplitudes of the shifted neutral positions are of the same magnitude. The deflection direction is changed when $Re = 6489$. The slight vibration around the neutral position disappears when $Re = 648.9$ because the fluid field is very laminar, as shown in Figure 6.10.

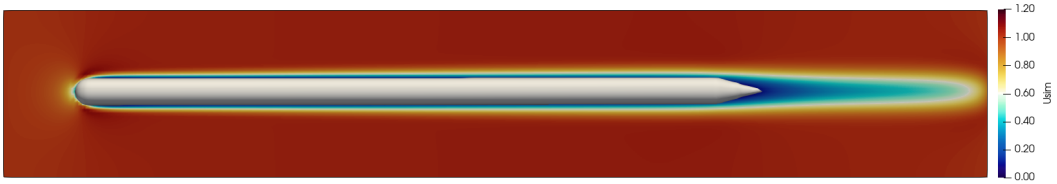


Figure 6.10: Fluid field schematic in the xoy plane. $Re = 648.9$. $t_{conv} = 4$.

Thus, we conclude that reducing the Reynolds number by increasing the fluid viscosity cannot stabilize the overall structure. However, it can reduce the vibration around the non-zero neutral position when the cylinder is in divergence instability. We believe this vibration reduction effect can also happen on a long cylinder that diverges.

We infer that increasing the Reynolds number of a very long cylinder that flutters by decreasing the viscosity will not change the flutter amplitude or instability mode, since Figure 6.6b indicates that the cylinder's unstable motion correlated weakly with the vortex shedding that relates to the Reynolds number. Increasing the cylinder length to alter the Reynolds number is different because the Cauchy number changes simultaneously in this way.

6.4. Downstream end shape

6.4.1. Extra blunt end

Our code simulates two downstream end shapes that have never been tested before. We first apply an extra blunt end to the validation case with $u = 5$. The extra blunt end has $D_{end} > D_{sim}$, as shown in Figure 5.4. We set $D_{end} = 11$, and other parameters remain unchanged, as shown in Table 5.1. The time variation diagrams of s_y and s_z are provided in Figure 6.11. Compared to the last row in the group of figures 5.19, the flutter amplitude reduces significantly. The shape of the cylinder is close to straight.

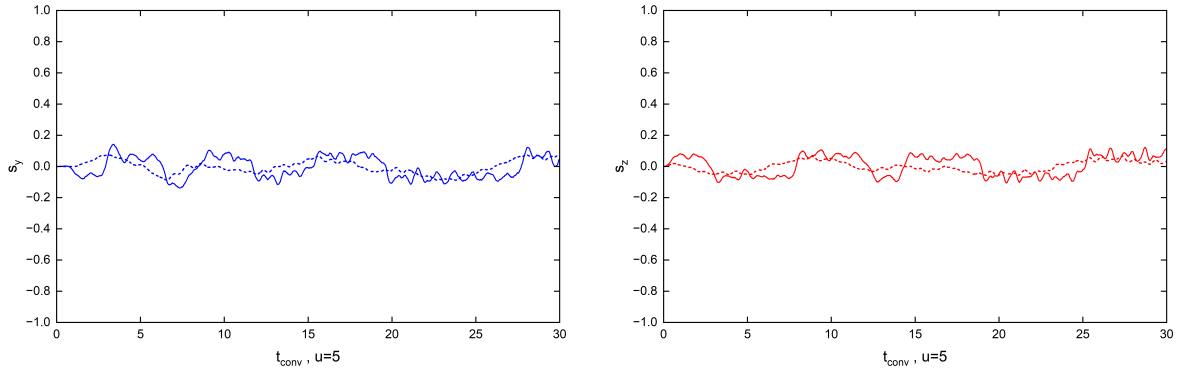


Figure 6.11: Time-varying displacements of the clamped-free cylinder with an extra blunt end. The solid and dashed lines represent the displacement at the downstream end and the midpoint.

Then, we apply the extra blunt end to a pinned-free cylinder with $AR = 40.00$. We also set $D_{end} = 11$, and other parameters remain unchanged, as shown in Table 6.1. The pinned-free cylinder is more similar to the array cable, making the simulation more meaningful. The time variation diagrams of s_y and s_z are provided in Figure 6.12. Compared to the last row in the group of figures 6.3, the flutter amplitude reduces significantly. The instability still exists since the displacement is never damped.

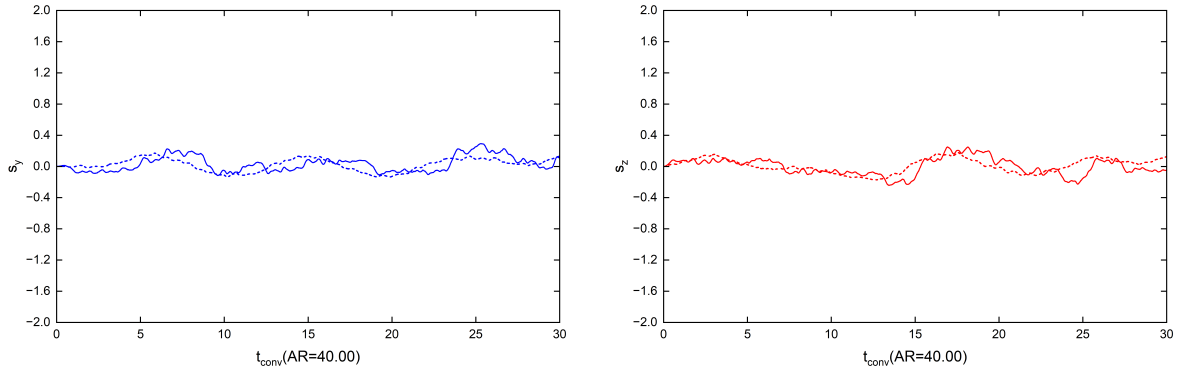


Figure 6.12: Time-varying displacements of the pinned-free cylinder with an extra blunt end. The solid and dashed lines represent the displacements at the downstream end and the midpoint.

The simulation results indicate that the extra blunt end can effectively reduce flutter amplitude. A possible reason is that the extra blunt end reduces the lift force on the downstream end, which is a driving force of the instability. Considering the analytical theory for the flutter of a long cylinder[25] also predicts that a sufficiently blunt end can stabilize the cylinder, we conclude that the extra blunt end stabilizes a relatively long cylinder.

6.4.2. Parachute

Applying a parachute to the cable end is a widely used method that helps stabilize the structure, as shown in Figure 2.2. Therefore, we try to apply a parachute to the downstream end of a pinned-free cylinder. We still use the cylinder with $AR = 40$. The structure is shown in Figure 5.2. Given new parameters: $D_{sim1} = 7.557$, $D_{sim2} = 2$, $D_{sim3} = 11$, $\xi_{cyl} = 0.91275$, $\xi_{cone} = 0.95436$, $\xi_{rope} = 0.98174$, $L_{sim} = 331.17333$, and $E_{para}I_{para} = 0.00779$. Other settings remain unchanged. The time variation diagrams of s_y and s_z are provided in Figure 6.13. The solid and dashed lines represent the displacement at the downstream end and the midpoint.

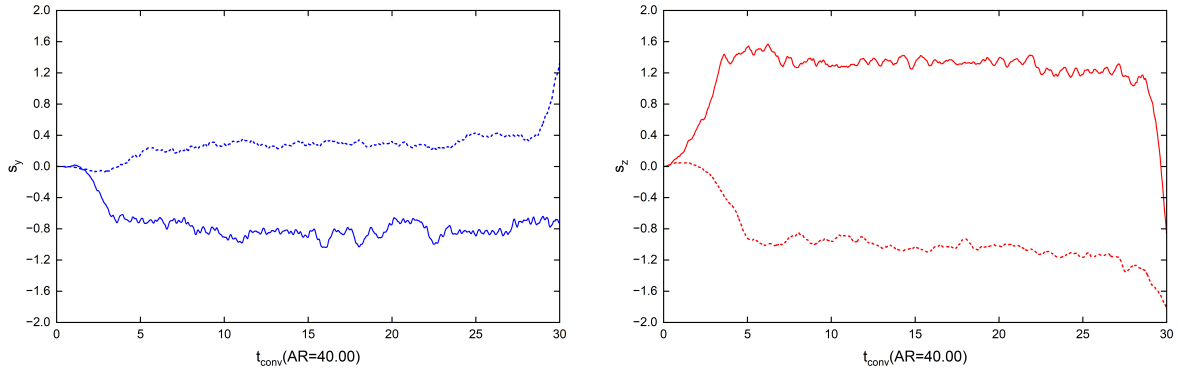


Figure 6.13: Time-varying displacements of the pinned-free cylinder with a parachute end. $AR = 40.00$.

The result seems incorrect. A pinned-free cylinder can't maintain static in axial flow with such significant second-mode bending. The distance between the solid body and flow boundary becomes too short to ensure the fluid boundary conditions, as shown in Figure 6.14.

Figure 6.14: Bending cylinder with a parachute end in the xoy plane. $t_{conv} = 30$.

The pressure force in the x direction is an essential restoring force on a parachute. Our simplification neglects this force, making the numerical model unsuitable. The parachute's geometry is also imperfect.

To apply the parachute, we should consider the force in the x direction in our solid structure solver. We must also use a variable bending stiffness since the rope is a string. The parametric body code also requires a modification to generate a real parachute instead of a solid block. In the end, we should extend the fluid field.

6.4.3. Discussion

This subsection provides a hypothesis of the mechanism of the destabilizing and stabilizing effects of the downstream end. The downstream end with various shapes creates a lift force, according to the research from Langre $et\ al.$[25]. The lift force at the downstream end is time-varying with some periodicity due to the vortex shedding behind. The local lift force triggers the initial instability at the adjacent part of the cylinder. The downstream end deflects a distance, and the nearby cylindrical part is pulled a distance deviating from the original position. The overall structure is now slightly bent, creating a lift force along the entire body like flaps. The overall lift force is sufficiently large to reach an equilibrium with the structural elastic restoring force before the cylinder flutters, which makes the cylinder statically diverge (buckle) or yaw with a non-zero neutral position. The local lift force at the downstream end cannot affect the cylinder's overall instability but can create vibration around the neutral position.

The lift force caused by the bending cylinder is fragile, since the cylinder is not an actual hydrofoil. If we keep increasing the flow velocity or the cylinder length, the lift force becomes smaller. The part of the hydro-pressure force that contributes to pushing the cylinder back to the zero neutral position grows. The smaller lift force cannot counter the elastic restoring force if the cylinder maintains the previous deflection (curvature). The cylinder then falls back to a neutral position closer to the zero-neutral position, like stalling, keeping a smaller static buckling or yawing. The lift force can still encounter the restoring force at this stage. The slight vibration around the neutral position always happens.

Increasing flow velocity or cylinder length further, the cylinder will keep deflecting to another side due to inertia and elasticity after it reaches the zero neutral position. The hydro-pressure force lifts the cylinder when the deflection is slight, and pushes the cylinder back when the deflection is large. The structural internal force and external hydro-pressure force form a system similar to the forced spring-mass system, and the hydro-pressure force inputs energy into the system continuously. The overall FSI effect leads to the flutter instability.

The blunt downstream end has a smaller (shorter) projected area in the lateral direction. It cannot create a sufficiently large local deflection to generate the lift force along the entire body and trigger the overall instability. If the overall instability has never been triggered, we can only observe the vibration caused by the vortex shedding behind the downstream end, as shown in the figures 6.11 and 6.12. That could be how an extra blunt end stabilizes the fluttered cylinder.

The mechanism requires more investigation to determine. The author records the pressure forces per unit length on Gauss points, but the integrated forces are not recorded explicitly. On some points, the unit length loads have the trend mentioned before. However, the data is still insufficient. In future research, we should monitor more physical variables, such as the integrated force along the cylinder and the thickness of the boundary layer.

7

Conclusion

After modifications, the solid structure solver, the parametric body code, and the FSI solver have been validated. The original 2D FSI code can now solve an FSI problem with a 3D fluid field. The code can qualitatively predict the instability of a solitary flexible cylinder with a taper or blunt end in axial flow. The effects of various parameters on the instability have been tested by the new numerical tool. From the observation, we have the following conclusions.

- The behavior of the instability transition of clamped-free and pinned-free cylinders as their lengths increase is similar to the transition of a clamped-free cylinder as increasing flow velocity: the cylinder in static divergence gradually moves back to the original zero neutral position, then develops the flutter instability. The behavior of the pinned-free cylinder is more apparent.
- The cylinder slightly vibrates around the non-zero neutral position when it is in the divergence instability.
- In the transition range of the length (aspect ratio), the cylinder is unstable, but the deflection amplitude is small. The transition range is the range between divergence and flutter instabilities.
- Increasing the length beyond the transition range will lead to the apparent flutter instability.
- An extra blunt downstream end can significantly reduce the flutter amplitude.
- The pinned-free cylinder is more unstable than a clamped-free cylinder.
- Increasing the fluid viscosity in the viscous force exemption range has no stabilizing effect.

Based on the conclusions, we predict that the array cable will flutter even under ideal working conditions (axial flow and no disturbance). We have suggestions for the company to stabilize the array cable: (1) Strengthening the upper boundary of the array cable to reduce displacement and rotation at the upper end, for example, by applying a PID system or structural reinforcement at the connectors. (2) Applying an extra blunt end to the downstream end, as shown in Figure 7.1. The investigated cylinders in this project are much shorter (with lower aspect ratios) than the actual array cable. We recommend that the company run actual experiments to test the stabilizing effects of the structures mentioned above.

Figure 7.1: Modified array cable.

The author plans to summarize the results of this project as a conference paper soon. Appendix A provides the outline of the paper.

The FSI code in this project is open-source. The author will try to upgrade the code soon to make GPU acceleration available and then upload it for open use. The code can be updated further to monitor more physical parameters. Analyzing more data can reveal the mechanism behind the cylinder's stability. It is possible to launch a journal paper after further investigation into the mechanism in the future.

The code is recommended for quick, qualitative predictions before formal calculations, simulations, or experiments. It requires more validation before it can be used generally and industrially.

References

[1] Ansys. *What is Finite Element Analysis (FEA)?* URL: https://www.ansys.com/simulation-topics/what-is-finite-element-analysis.

[2] W. Arendt and M. Warma. “Dirichlet and Neumann boundary conditions: What is in between?” In: *Nonlinear Evolution Equations and Related Topics: Dedicated to Philippe Bénilan* (2004), pp. 119–135.

[3] O. A. Bauchau and J. I. Craig. *Euler-Bernoulli beam theory*. Dordrecht: Springer, 2009, pp. 173–221.

[4] R. N. Bracewell. “The fourier transform”. In: *Scientific American* 260.6 (1989), pp. 86–95.

[5] P. Causin, J.-F. Gerbeau, and F. Nobile. “Added-mass effect in the design of partitioned algorithms for fluid–structure problems.” In: *Computer methods in applied mechanics and engineering* 194.42-44 (2005), pp. 4506–4527.

[6] C. Chijioke et al. “FSI of a Cantilever Beam: FVM-FEM and Neural Network Analysis.” In: *Fluids Engineering Division Summer Meeting* 85833 (2022).

[7] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[8] J. A. Cottrell et al. “Isogeometric analysis of structural vibrations”. In: *Computer Methods in Applied Mechanics and Engineering* 195.41 (2006). John H. Argyris Memorial Issue. Part II, pp. 5257–5296. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2005.09.027. URL: https://www.sciencedirect.com/science/article/pii/S0045782505005451.

[9] J. S. Dai. “Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections”. In: *Mechanism and Machine Theory* 92 (2015), pp. 144–152.

[10] J. De Ridder et al. “Simulating the fluid forces and fluid-elastic instabilities of a clamped–clamped cylinder in turbulent axial flow”. In: *Journal of Fluids and Structures* 55 (2015), pp. 139–154. ISSN: 0889-9746.

[11] The Marine Diaries. *Discovering the Value of the Invaluable Deep Sea*. URL: https://www.themarinediaries.com/tmd-blog/discovering-the-value-of-the-invaluable-deep-sea.

[12] O. Doaré and E. de Langre. “The flow-induced instability of long hanging pipes.” In: *European Journal of Mechanics-A/Solids* 21.5 (2002), pp. 857–867.

[13] Engineers Edge. *Viscosity of air, dynamic, and kinematic*. URL: https://www.engineersedge.com/physics/viscosity_of_air_dynamic_and_kinematic_14483.htm (visited on 03/29/2025).

[14] C. Fureby. “Large eddy simulation modelling of combustion for propulsion applications.” In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367.1899 (2009), pp. 2957–2969.

[15] W. R. Hawthorne. “The early development of the Dracone flexible barge.” In: *Proceedings of the institution of mechanical engineers* 175.1 (1961), pp. 52–83.

[16] B. Henry. “Immersed boundary methods.” In: (2010).

[17] G. J. Higgins. “Tests of the N.P.L. airship model in the variable density wind tunnel”. In: *Technical notes National advisory committee for aeronautics* 264 (1927).

[18] C. Hirth. *Numerical computation of internal and external flows.* Elsevier, 2007.

[19] S. F. Hoerner. *Fluid-dynamic drag*. Sighard F. Hoerner, 1965, pp. 6-16–6-17.

[20] G. Iaccarino and R. Verzicco. “Immersed boundary technique for turbulent flow simulations.” In: *Appl. Mech. Rev.* 56.3 (2003), pp. 331–347.

[21] R. Jenson. *Helicopter aerial refueling*. URL: http://www.afsoc.af.mil/shared/media/photodb/photos/050925-F-3983J-019.jpg.

[22] M. Kheiri and M. P. Païdoussis. “Dynamics and stability of a flexible pinned-free cylinder in axial flow.” In: *Journal of Fluids and Structures* 55 (2015), pp. 204–217.

[23] D. Koks. *Explorations in mathematical physics: the concepts behind an elegant language*. Springer, 2006, pp. 151–154.

[24] N. A. Labarbera. “A Comparison of Modeling Approaches of a High Aspect Cylinder in Axial Flow.” In: (2015).

[25] E. de Langre et al. “Flutter of long flexible cylinders in axial flow.” In: *Journal of Fluid Mechanics* 571 (2007), pp. 371–389.

[26] M. Lauber. “Computational fluid-structure Interaction of membranes and shells with application to bat flight.” In: *Doctoral dissertation, University of Southampton* (2023).

[27] M. Lauber. *Lecture note: Heat equation and discretization*.

[28] M. Lauber, G. D. Weymouth, and G. Limbert. “Immersed-Boundary Fluid-Structure Interaction of Membranes and Shells”. In: *Journal of Physics: Conference Series* 2647.5 (2024), p. 052002.

[29] J. W. Leis. *Digital signal processing using MATLAB for students and researchers*. John Wiley & Sons, 2011.

[30] C. Lemaitre, P. Hémon, and E. de Langre. “The flow-induced instability of long hanging pipes.” In: *Journal of Fluids and Structures* 20.7 (2005), pp. 913–925.

[31] M. J. Lighthill. “Note on the swimming of slender fish.” In: *Journal of fluid Mechanics* 9.2 (1960), pp. 305–317.

[32] Z.G. Liu, Y. Liu, and J. Lu. “Numerical simulation of the fluid–structure interaction for an elastic cylinder subjected to tubular fluid flow”. In: *Computers Fluids* 68 (2012), pp. 192–202. ISSN: 0045-7930.

[33] A. P. Maertens and G. D. Weymouth. “Accurate Cartesian-grid simulations of near-body flows at intermediate Reynolds numbers.” In: *Computer Methods in Applied Mechanics and Engineering* 283 (2015), pp. 106–129.

[34] C. Michler et al. “A monolithic approach to fluid–structure interaction.” In: *Computers and fluids* 33.5-6 (2004), pp. 839–848.

[35] M. M. Munk. “The aerodynamic forces on airship hulls.” In: *NACA Report* 184 (1924).

[36] C. C. Ni and R. J. Hansen. “An experimental study of the flow-induced motions of a flexible cylinder in axial flow.” In: *ASME. J. Fluids Eng.* 100.4 (1978), pp. 389–394.

[37] F. T. M. Nieuwstadt, J. Westerweel, and B. J. Boersma. *Turbulence: introduction to theory and applications of turbulent flows*. Springer, 2016, pp. 87–100.

[38] A. Öchsner. *Euler–Bernoulli Beam Theory*. Cham: Springer International Publishing, 2021, pp. 7–66.

[39] OriginLab. *Fast Fourier Transform (FFT)*. URL: https://www.originlab.com/doc/Origin-Help/FFT.

[40] Anton paar. *Viscosity of water*. URL: https://wiki.anton-paar.com/en/water/ (visited on 03/30/2025).

[41] M. P. Païdoussis. “Dynamics of cylindrical structures in axial flow: A review”. In: *Journal of Fluids and Structures* 107 (2021), p. 103374.

[42] M. P. Païdoussis. “Dynamics of cylindrical structures subjected to axial flow.” In: *J. Sound Vib.* 29.3 (1973), pp. 365–385.

[43] M. P. Païdoussis. “Dynamics of flexible slender cylinders in axial flow. Part 2: experiments.” In: *Journal of Fluid Mechanics* 26.4 (1966), pp. 737–751.

[44] M. P. Païdoussis. *Fluid-Structure Interactions: Volume 2 Slender Structures and Axial Flow.* Second Edition. Québec, Canada: Elsevier, 2016, pp. 145–231.

[45] M. P. Païdoussis. “Linear and nonlinear dynamics of cantilevered cylinders in axial flow. Part 1: physical dynamics”. In: *Journal of Fluids and Structures* 16.6 (2002), pp. 691–713.

[46] M. P. Païdoussis et al. “Linear and nonlinear dynamics of cantilevered cylinders in axial flow. Part 1: physical dynamics.” In: *Journal of fluids and structures* 16.6 (2002), pp. 691–713.

[47] U. Piomelli. “Large-eddy simulation: achievements and challenges.” In: *Progress in aerospace sciences* 35.4 (1999), pp. 335–362.

[48] S. Piperno and C. Farhat. “Partitioned procedures for the transient solution of coupled aeroelastic problems–Part II: energy transfer analysis and three-dimensional applications.” In: *Computer methods in applied mechanics and engineering* 190.24-25 (2001), pp. 3147–3170.

[49] Y. Yin Q. Fan and J. Tang. *Mechanics of Materials*. Beijing: Tsinghua University Press, 2014, pp. 174–185.

[50] University of Rhode Island and Inner Space Center. *Hydrophone Arrays*. 2021. URL: https://dosits.org/galleries/technology-gallery/basic-technology/hydrophone-arrays/ (visited on 05/18/2024).

[51] A. Rodriguez. “Vibrational Control of a Cantilever Beam Using Sensor and Actuator Averaging Methods”. In: *Bachelor thesis, Massachusetts Institute of Technology* (2024), p. 22.

[52] D. F. Rogers. *An introduction to NURBS: with historical perspective*. Elsevier, 2000.

[53] D. Roylance. “Finite element analysis”. In: *Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge* (2001).

[54] A. K. Slone et al. “Dynamic fluid–structure interaction using finite volume unstructured mesh procedures.” In: *Computers & structures* 80.5-6 (2002), pp. 371–390.

[55] P. Stajanca. *OptiArray Technology introduction*. 2024. URL: www.optics11.com.

[56] G. I. Taylor. “Analysis of the swimming of long and narrow animals.” In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 214.1117 (1952), pp. 158–183.

[57] testbook. *Equation of Sphere: Circumference, Surface Area Volume Formula*. URL: https://testbook.com/maths/equation-of-sphere.

[58] S. Timoshenko. *History of strength of materials: with a brief account of the history of theory of elasticity and theory of structures*. Courier Corporation, 1983, pp. 25–36.

[59] The Engineering Toolbox. *Seawater - Properties*. URL: https://www.engineeringtoolbox.com/sea-water-properties-d_840.html.

[60] G. S. Triantafyllou and C. Chryssostomidis. “Analytic Determination of the Buckling Speed of Towed Slender Cylindrical Beams.” In: *ASME. J. Energy Resour. Technol.* 106.2 (1984), pp. 246–249.

[61] S.P. Venkateshan and Prasanna Swaminathan. *Computational Methods in Engineering*. Boston: Academic Press, 2014, pp. 317–373. ISBN: 978-0-12-416702-5. DOI: https://doi.org/10.1016/B978-0-12-416702-5.50009-0. URL: https://www.sciencedirect.com/science/article/pii/B9780124167025500090.

[62] H.M. Verhelst. “Modeling wrinkling behavior of large floating thin offshore structures: an application of isogeometric structural analysis for post-buckling analyses”. In: *Delft University of Technology* (2019), pp. 35–43.

[63] G. D. Weymouth and B. Font. “WaterLily. jl: A differentiable fluid simulator in Julia with fast heterogeneous execution.” In: *arXiv* 2304.08159 (2023).

[64] G. D. Weymouth and D. K. Yue. “Boundary data immersion method for Cartesian-grid simulations of fluid-body interaction problems.” In: *Journal of Computational Physics* 230.16 (2011), pp. 6233–6247.

[65] Wikipedia. *Rodrigues' rotation formula*. URL: https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula (visited on 04/01/2025).

[66] R. N. Youngworth, B. B. Gallagher, and B. L. Stamper. “An overview of power spectral density (PSD) calculations”. In: *Optical manufacturing and testing VI* 5869 (2005), pp. 206–216.

A

Conference paper outline

Numerical investigations of solitary clamped-free and pinned-free flexible cylinders in axial flow

Abstract

The instability of a solitary flexible cylinder in three-dimensional axial flow with various boundary conditions, lengths, and Reynolds numbers has been investigated by CFD (computational fluid dynamics) simulation. A new CFD code is modified from an existing code for a two-dimensional FSI (fluid-structure interaction) problem. The fluid field is solved by the BDIM (boundary data immersion boundary method). The solid structure's displacement is solved by the IGA (isogeometric analysis). The coupling effect is solved by the partitioned and implicit method combined with the IQN (interface Quasi-Newton) method. The simulated results of a clamped-free cylinder can match the experimental results quantitatively. The investigation for cylinders with different parameters indicates that the instability transition as cylinder length increases is similar to that of a clamped-free cylinder as flow velocity increases, small vibrations around the non-zero neutral position are observed when cylinder is in divergence, the deflection amplitude during the transition range reduces, an extra blunt downstream end can reduce the flutter amplitude, and increasing the fluid viscosity in the viscous force exemption range has no stabilizing effect.

Introduction

Briefly introduce the array cable. Describe the research question. Illustrate the definitions of the boundary conditions and types of instabilities. Provide a short literature review of the previous research.

- There are validated analytical models that do not consider the FSI effect.
- There are numerical studies for a pinned-pinned or clamped-clamped cylinder in the 3D fluid field, while the numerical study for a clamped-free or pinned-free cylinder is rare.
- Introduce the experimental studies.

Then, provide the research matrix and the outline of the paper.

Flow solver

Introduce the used solver: WaterLily. Provide the principle of BDIM and governing equations. Since other researchers have validated the flow solver, this paper omits the validation.

Solid structure solver

Introduce the principle of IGA and the difference from FEA. Illustrate the Gauss interaction method. Provide the governing equations. Compare the numerical and analytical results of a cantilever beam under distributed loads in two directions.

Parametric body code

Illustrate how to use the parametric code to generate the cylinder. Mention the principle of the B-spline curve and SDF. Introduce the airship case used for validation. Validate the code by comparing the drag coefficient with that measured in the airship experiment.

FSI solver

Geometry

Introduce the geometry of the cylinder with a tapered end. Describe the coordinate systems that are used in the code.

Pressure integrator

Illustrate how to measure the hydro-pressure force per unit length on Gauss points. Validate the pressure integrator by comparing the numerical approximation with the analytical results.

FSI solver validation

Introduce the IQN method. Provide the experiment's parameters used for validation. Provide the settings for the numerical simulation. Validate the code by comparing numerical and experimental results.

Investigation

Introduce the test cases' parameters and geometries. Test a clamped-free cylinder with multiple lengths (aspect ratios). Simulate a pinned-free cylinder with various lengths, downstream ends, and Reynolds numbers of the fluid field. Provides the time-varying displacements and the displacement range. Analyze and discuss the results. Remove the investigation of the cylinder with a parachute.

Conclusion

Make conclusions. Discuss the limits of this research and the future plans.

Reference

...

B

Solid structure solver validation code

Splines.jl\Splines.jl\examples\test2.jl

```
using Splines
using Plots

numElem=12;degP=3 # number of elements, degree of the B-spline curve

println("Testing on fixed-fixed beam with UDL:")
println(" numElem: ", numElem)
println(" degP: ", degP)
# Material properties and mesh
ptLeft = 0.0 # define the parametric space lower limit
ptRight = 1.0  # define the parametric space uper limit
EI = 10000 # bending stiffness
EA = 1.0 # tensile stiffness (never use)
qy = 1.0 # distribution load in y-direction
qz = 1.0 # distribution load in z-direction
L = 10 # actual length of the beam
# "x" is the parametric coordinate
exact_sol_z(x,qz) = qz/(24EI).*(6*x.^2 .- 4*x.^3 .+ x.^4)*L^4 # cantilever beam, analytical solution in physical space.
exact_sol_y1(x,qy) = (3qy/(16EI)*x.^2-qy/(12EI)*x.^3)*L^4 # analytical solution of the first half beam
exact_sol_y2(x,qy) = ((qy/(24EI).*(6x.^2 .- 4x.^3 .+ x.^4) .-1/(128EI)*(qy) .- 1/(48EI)*qy*(x.-0.5)))*L^4 # analytical
solution of the rest part

mesh, gauss_rule = Mesh1D(ptLeft, ptRight, numElem, degP)

# boundary conditions
Dirichlet_BC = [
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=1), # Boundary1D(type of B.C., position applied B.C., apply on which
variable)
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=2)
]

```

```
Neumann_BC = [
    Boundary1D("Neumann", ptLeft, 0.0; comp=1),
    Boundary1D("Neumann", ptLeft, 0.0; comp=2)
]

# make a problem
operator = DynamicFEOperator(mesh, gauss_rule, EI, EA, Dirichlet_BC, Neumann_BC)

# uniform external loading at integration points
force = zeros(2,length(uv_integration(operator))); force[1,25:48] .= -2qy; force[2,:] .= qz # apply force on gauss points

# update the jacobian, the residual and the external force
# linearized residuals
integrate!(operator, zero(operator.resid), force)

# compute the residuals
operator.resid .= - operator.ext

# apply BC
applyBC!(operator)

# solve the system and return
result = -operator.jacob\operator.resid

# Postprocessing
name = "Fea_vali"
v0 = result[1:mesh.numBasis]
w0 = result[mesh.numBasis+1:2mesh.numBasis]
x = LinRange(ptLeft, ptRight, numElem+1)
x1 = LinRange(ptLeft, 0.5, round(Int,numElem/2)+1)
x2 = LinRange(0.5, ptRight, round(Int,numElem/2)+1)
v = getSol(mesh, v0, 1) *L^4 # transfer the result in parametric space to physical space
w = getSol(mesh, w0, 1) *L^4
```

```

ve1 = exact_sol_y1(x1,-2qy)
ve2 = exact_sol_y2(x2,-2qy)
ve = vcat(ve1[1:round(Int,numElem/2)+1],ve2[2:round(Int,numElem/2)+1])
println("Error: ", norm(v .- ve)) # print the error
plot(x, v, label="Sol")
plot!(x, ve, label="Exact")
savefig(name*"_y"*".png")

we = exact_sol_z(x,qz)
println("Error: ", norm(w .- we)) # print the error
plot(x, w, label="Sol")
plot!(x, we, label="Exact")
savefig(name*"_z"*".png")
```

C

Parametric body validation code

ParametricBodies.jl\example\3DAirship_v4.jl

```
###
using Pkg
Pkg.develop(path="E:/Graduation_Assignment/waterlily/WaterLily.jl")
using WaterLily
println(pathof(WaterLily))
###
using ParametricBodies
using StaticArrays
using CUDA
using WriteVTK
using Plots

using WaterLily: @loop,inside,inside_u,nds,∇²u
function pressure_force(sim,t=WaterLily.time(sim.flow),T=promote_type(Float64,eltype(sim.flow.p)))
    sim.flow.f .= zero(eltype(sim.flow.p))
    @loop sim.flow.f[I,:] .= sim.flow.p[I]*nds(sim.body,loc(0,I,T),t) over I ∈ inside(sim.flow.p)
    sum(T,sim.flow.f,dims=ntuple(i->i,ndims(sim.flow.p)))[:] |> Array
end
function viscous_force(sim,t=WaterLily.time(sim.flow),T=promote_type(Float64,eltype(sim.flow.u)))
    sim.flow.f .= zero(eltype(sim.flow.u))
    @loop sim.flow.f[I,:] .= -sim.flow.v*∇²u(I,sim.flow.u)*nds(sim.body,loc(0,I,T),t) over I ∈ inside_u(sim.flow.u)
    sum(T,sim.flow.f,dims=ntuple(i->i,ndims(sim.flow.u)-1))[:] |> Array
end

function characteristic_length(L) # f<9pi/(6*400)
        f = 0.00391/(27.953*0.0254)^3
        @assert f<9pi/(6*400)
    return f^(1/3)*L
end

```

```
function airship(L; Re_exp=108500,U=1,T=Float32,mem=Array)#CUDA.CuArray)
    length = characteristic_length(L)
    @assert L%32==0 # ensures your L is sufficiently big pow of 2
    D = 3L÷16 # safe over estimate
    cps = SA{T}[1.0  0.819232  0.38  0.15   0.0    0.0
                0.0  0.054     0.092 0.079  0.03   0.0 ] * L
    curve = BSplineCurve(cps; degree=4)
    function map(x::SVector{3},t) # Map the 3D space into a 2D curve
        y = x - SA[L÷2,6D,6D]
        r = √(y[2]^2 + y[3]^2)
        return SA[y[1],r]
    end
    Body = ParametricBody(curve; map,scale=1)
    Simulation((5L÷2,12D,12D),(U,0,0),L; ν=U*length/Re_exp,body=Body,T=T,mem=mem,exitBC=true)
end

velocity(a::Simulation) = a.flow.u |> Array;
pressure(a::Simulation) = a.flow.p |> Array;
#NDS(a::Simulation) = a.flow.f ./ a.flow.p |> Array;
vforce(a::Simulation) = a.flow.f |> Array;
body(a::Simulation) = (measure_sdf!(a.flow.σ, a.body);
                       a.flow.σ |> Array;)
custom_attrib = Dict(
    "Velocity" => velocity,
    "Pressure" => pressure,
    #"NDS" => NDS,
    "vforce" => vforce,
    "Body" => body
)# this maps what to write to the name in the file


function run(L,duration,tstep)

```

```
    A = characteristic_length(L)^2
    sim = airship(L)

    WaterLily.apply!(x->x[1], sim.flow.p)
    V = pressure_force(sim)[1]
    #@assert abs(A-V^(2/3))<10
    sim.flow.p .= 0

    name = "3DAirshp_$(L)_108500_v4"
    wr = vtkWriter(name; attrib=custom_attrib,dir="E:/Graduation_Assignment/waterlily/ParametricBodies.jl/vtk_data")
    pforce,vforce,drag = [],[],[]
    ratio = []

    for tᵢ in range(0,duration;step=tstep)
        sim_step!(sim,tᵢ,remeasure=false)
        cp = 2*pressure_force(sim)[1]/A
        cv = 2*viscous_force(sim)[1]/A
        push!(pforce,cp)
        push!(vforce,cv)
        push!(drag,cp+cv)
        push!(ratio,cv/cp)
        println("3D: tU/L=",round(tᵢ,digits=4),", Δt=",round(sim.flow.Δt[end],digits=3))
        write!(wr, sim)
    end
    close(wr)

    @show(-drag)
    @show(-pforce)
    @show(-vforce)
    @show(ratio)
    @show(A)
    @show(V^(2/3))
    plot(range(0,duration;step=tstep),-drag,label="Cd",xlabel="tU/L",ylabel="C_force")
```

```
    plot!(range(0,duration;step=tstep),-vforce,label="Cv")
    plot!(range(0,duration;step=tstep),-pforce,label="Cp")
    savefig(name*".png")
    plot(range(0,duration;step=tstep),-vforce,label="Cv",xlabel="tU/L",ylabel="C_force")
    savefig(name*"_vforce"*".png")
    plot(range(0,duration;step=tstep),-pforce,label="Cp",xlabel="tU/L",ylabel="C_force")
    savefig(name*"_pforce"*".png")
    plot(range(0,duration;step=tstep),ratio,label="Cv/Cp",xlabel="tU/L",ylabel="Cv/Cp")
    savefig(name*"_CvCp"*".png")
end

run(2^5,8,0.1)

#for L in [2^5,2^6,3*2^5,2^7,3*2^6]
    #run(L)
#end
```

D

Name of the fluid domain boundaries

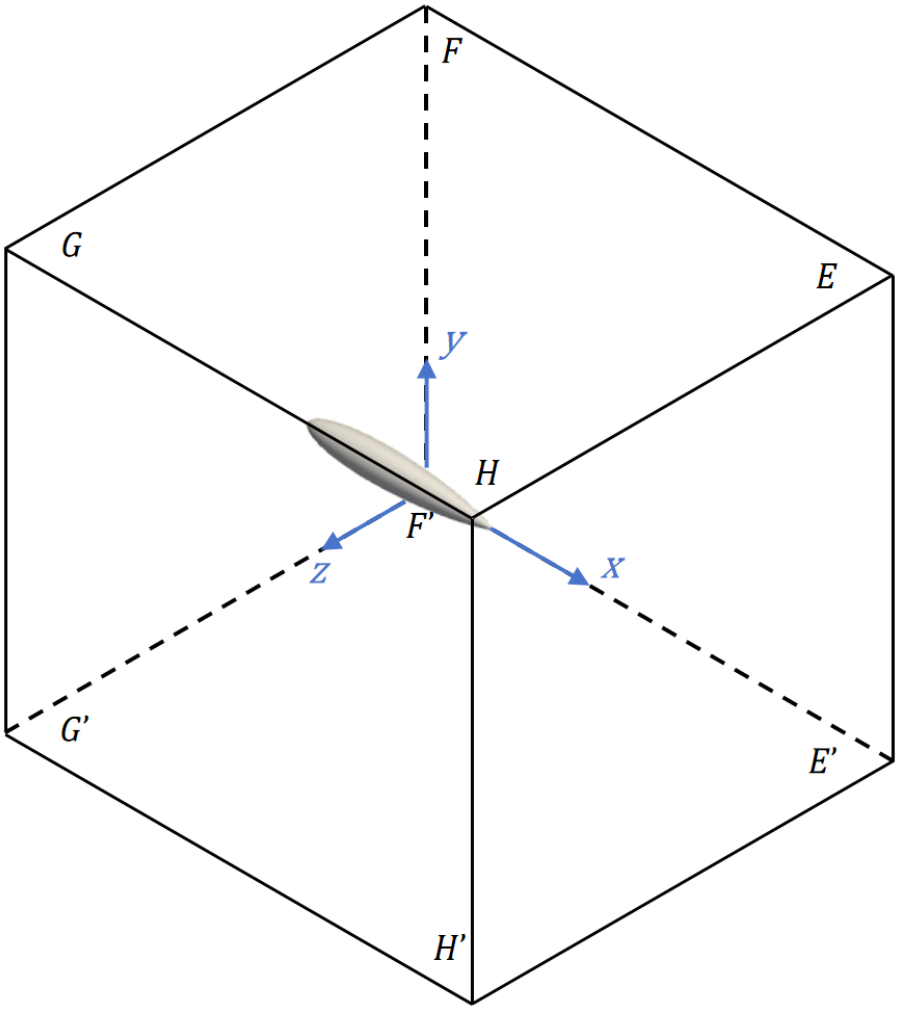


Figure D.1: Three-dimensional view of the fluid domain with the airship. $EFGH$ is the upper boundary. $E'F'G'H'$ is the lower boundary. $FGG'F'$ is the (flow) inlet boundary. $EHH'E'$ is the (flow) outlet boundary. $HGG'H'$ is the front boundary. $EFF'E'$ is the back boundary. The flow is in the x direction.

E

Velocity and pressure fields

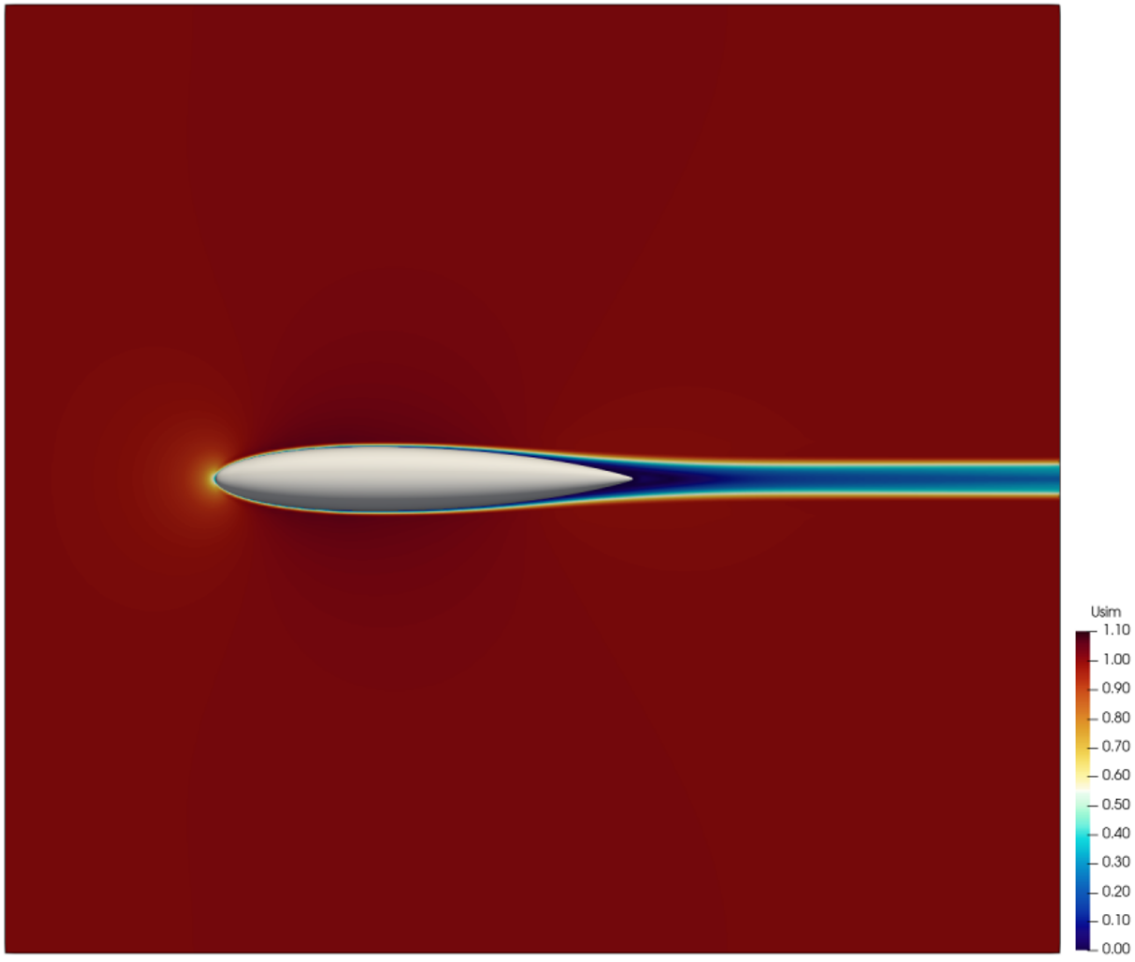


Figure E.1: Velocity field in the local xoy plane. $L = 192$.

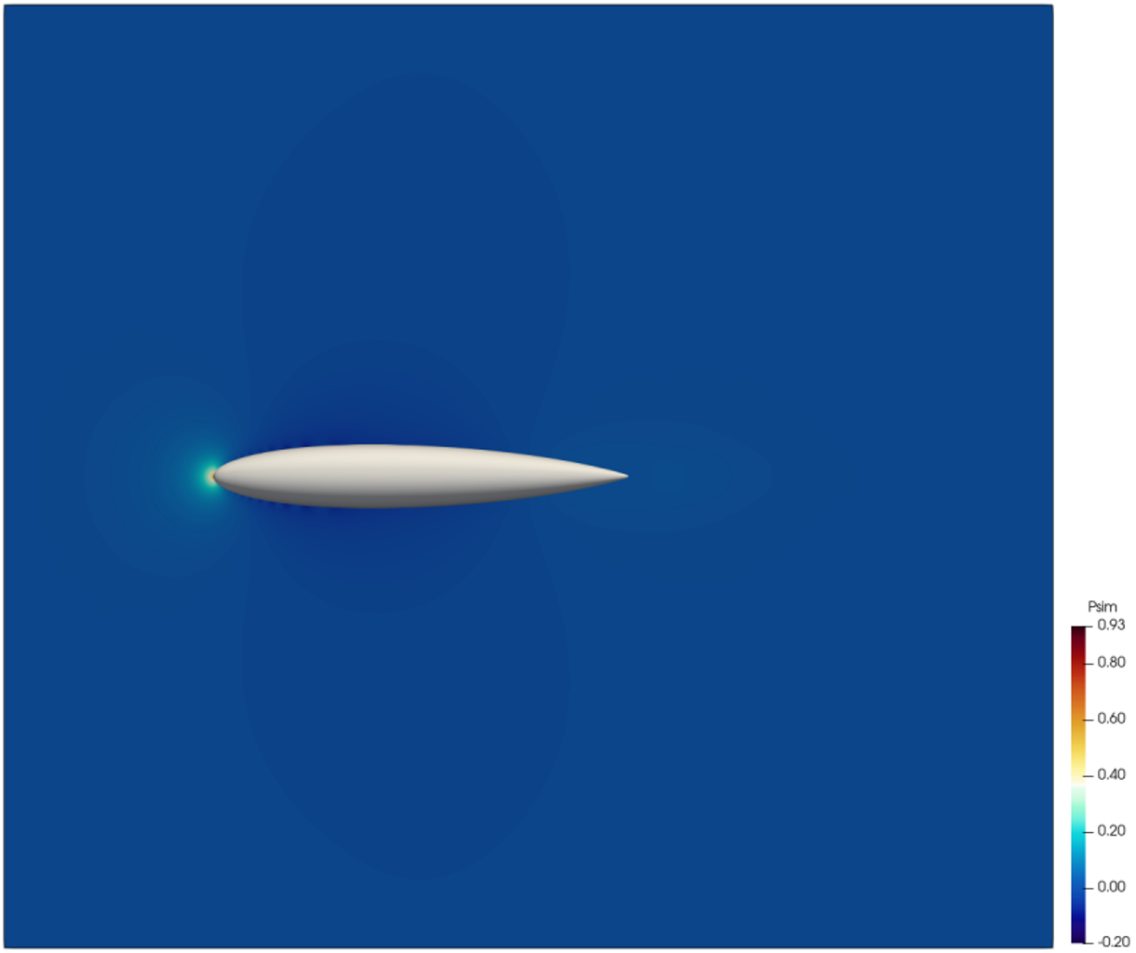


Figure E.2: Pressure field in the local xoy plane. $L = 192$.

F

FSI validation code

Filament.jl\code\src\FlappingFillament_tapper_end4.jl

```
using WaterLily
using ParametricBodies
using Splines
using StaticArrays
using LinearAlgebra
using WriteVTK
using CSV,Tables,DataFrames

include("../ext/vis.jl")
include("Coupling.jl")
include("utils.jl")
# Material properties and mesh
L = 192
ll = round(Int,L/4)
LL = round(Int,L)
numElem = 12
numE = round(Int,numElem)
degP = 3
ptLeft = 0.0
ptRight = 1.0

# simulation para
AR = 15.4 / 0.653 # aspect ratio of cylinder
tapper_leng = 31 / 17 ## dimension_less tapper length
tl = "31_17"
tapper = AR / (AR + tapper_leng)
TL = L * (ptRight - ptLeft) # total length with tapper end in numerical space
thk1 = TL*tapper/AR
thk2 = 0
thkk2 = round(Int,thk2)

# exp data
rho_exp = 1000
μ_exp = 1.1375 * 10^(-3) # 15 ceils

u_exp = 5.0 ###
D_exp = 0.653 * 0.0254 # m
L_exp = 15.4 * 0.0254 # m

```

```
# m_expp = 0.014
# M_expp = m_exp * (0.46/0.54) # lb/inch
m_exp = 0.25 # kg/m
M_exp = rho_exp * D_exp^2 * π / 4 * 0.055997 # lb/in
β = (M_exp/0.055997) / (m_exp + (M_exp/0.055997)) # mass ratio from literature

EI_exp = 859 # lb*inch^3/sec^2
U_exp = u_exp / ((L_exp/0.0254) * (M_exp/EI_exp)^0.5) * 0.0254 # m/s
Re_exp = rho_exp * D_exp * U_exp / μ_exp
Ca_exp = EI_exp * 7.433*10^(-6) / (π*D_exp^4/64) / (rho_exp*U_exp^2)
# Ca_test = EI_exp * 7.433*10^(-6) / (rho_exp*U_exp^2*(L_exp/tapper)^4)

# parameters
U = 1 * (ptRight - ptLeft)
E_sim = Ca_exp * (1*U^2)
EI_sim = E_sim * (π*(thk1)^4) / 64
# EI_sim_test = Ca_test * (1*U^2*L^4)
# EU_test = EI_sim_test / TL^3
EI = EI_sim / TL^3 # EI for fea solver
EA = 100_000.0  # make inextensible
density(ξ) = m_exp / (π*D_exp^2/4) * (1/rho_exp) # no gravity force, desnity of the solid structure in numerical space
β_rho = 1 / (1 + density(1))

# mesh
mesh, gauss_rule = Mesh1D(ptLeft, ptRight, numElem, degP)

# boundary conditions
Dirichlet_BC = [
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=1),
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=2)
]
Neumann_BC = [
    Boundary1D("Neumann", ptLeft, 0.0; comp=1),
    Boundary1D("Neumann", ptLeft, 0.0; comp=2)
]

# make a structure
struc = DynamicFEOperator(mesh, gauss_rule, EI, EA,
                          Dirichlet_BC, Neumann_BC, ρ=density; ρ∞=0.0)

## Simulation parameters
Re=Re_exp
```

```
ϵ=0.5

# construct from mesh, this can be tidy
u⁰ = SMatrix{3,size(mesh.controlPoints,2)}(mesh.controlPoints[1:3,:].*L.+[24,24,24].+0.5) #controlPoints
half_doma = 24.5

# flow sim
function thickness(u,thick1,thick2,tapper)
    u<=tapper ? thick1 : ((((thick2)/2-(thick1)/2)/(1-tapper))*u + ((thick2)/2-((thick2)/2-(thick1)/2)/(1-tapper)))*2
end
body = DynamicNurbsBody(NurbsCurve(u⁰,mesh.knots,mesh.weights);thk=(u)->thickness(u,thk1,thk2,tapper),boundary=false)

# force function
integration_points = uv_integration(struc)/(ptRight-ptLeft) ## utils.jl

# make a coupled sim
sim = CoupledSimulation((280,48,48),(U,0,0),L,body,struc,IQNCoupling;
                        U,ν=U*(thk1)/Re,ϵ,T=Float64,relax=0.05,maxCol=6)

q_poke = 0.01

# sim time
tₒ = round(sim_time(sim)); duration = 30; step = 0.2
# write para_view
velocity(a::CoupledSimulation) = a.flow.u |> Array;
pressure(a::CoupledSimulation) = a.flow.p |> Array;
_body(a::CoupledSimulation) = (measure_sdf!(a.flow.σ, a.body, WaterLily.time(a.flow));
                        a.flow.σ |> Array;)
custom_attrib = Dict(
    "u" => velocity, "p" => pressure, "d" => _body)
uu = round(Int,10*u_exp) # for the file name
name = "3D_cylinder_withTapper_u$(uu)_L$(LL)_Gpnts$(numE)_thk2$(thkk2)_Validation_"*tl
wr = vtkWriter(name; attrib=custom_attrib)

disy_1 = []
disz_1 = []
disy_2 = []
disz_2 = []
disy_3 = []
disz_3 = []
disy_4 = []
disz_4 = []
```

```
disy_5 = []
disz_5 = []
disy_6 = []
disz_6 = []
disy_7 = []
disz_7 = []
disy_8 = []
disz_8 = []
forces_y = []
forces_z = []
ttime = []
# time loop
@time for tᵢ in range(t₀,t₀+duration;step)

    # update until time tᵢ in the background
    t = sum(sim.flow.Δt[1:end-1])

    while t < tᵢ*sim.L/sim.U

        println("  tᵢ=$tᵢ, t=$(round(t,digits=2)), Δt=$(round(sim.flow.Δt[end],digits=2))")

        # save at start of iterations
        store!(sim); iter=1;

        # iterative loop
        while true

            #  integrate once in time
            solve_step!(sim.struc, sim.forces[2:3,:], sim.flow.Δt[end]/sim.L) # apply FEA solver

            # update flow, this requires scaling the displacements
            sim.body = ParametricBodies.update!(sim.body,u⁰.+ L*sim.pnts,sim.flow.Δt[end]) ###!!!
            measure!(sim,t); mom_step!(sim.flow,sim.pois) # body and flow from flow solver.

            # apply!(x->x[2],sim.flow.p)
            # get new coupling variable
            sim.pnts[2:3,:] = points(sim.struc) # use FEA solver to obtain the displacement of control points.
            sim.forces = force_fea(sim,degP+1,(thk1)/2,(thk2)/2,numElem,sim.L,tapper)
            if tᵢ <= 0.2
                sim.forces[3,:] .+= q_poke
            end
```

```
            # accelerate coupling
            print("    iteration: ",iter)
            converged = update!(sim.cpl, sim.pnts, sim.forces; atol=1e-2) # coupling.jl

            # check for convengence
            (converged || iter+1 > 50) && break

            # if we have not converged, we must revert
            revert!(sim); iter += 1
        end
        t += sim.flow.Δt[end]
        push!(disy_1, sim.body.curve.(0.125)[2] .- half_doma)
        push!(disz_1, sim.body.curve.(0.125)[3] .- half_doma)
        push!(disy_2, sim.body.curve.(0.25)[2] .- half_doma)
        push!(disz_2, sim.body.curve.(0.25)[3] .- half_doma)
        push!(disy_3, sim.body.curve.(0.375)[2] .- half_doma)
        push!(disz_3, sim.body.curve.(0.375)[3] .- half_doma)
        push!(disy_4, sim.body.curve.(0.5)[2] .- half_doma)
        push!(disz_4, sim.body.curve.(0.5)[3] .- half_doma)
        push!(disy_5, sim.body.curve.(0.625)[2] .- half_doma)
        push!(disz_5, sim.body.curve.(0.625)[3] .- half_doma)
        push!(disy_6, sim.body.curve.(0.75)[2] .- half_doma)
        push!(disz_6, sim.body.curve.(0.75)[3] .- half_doma)
        push!(disy_7, sim.body.curve.(0.875)[2] .- half_doma)
        push!(disz_7, sim.body.curve.(0.875)[3] .- half_doma)
        push!(disy_8, sim.body.curve.(1.0)[2] .- half_doma)
        push!(disz_8, sim.body.curve.(1.0)[3] .- half_doma)
        push!(forces_y, sim.forces[2,:])
        push!(forces_z, sim.forces[3,:])
        push!(ttime,t*sim.U/sim.L)
    end
    write!(wr, sim)
    CSV.write(name*"_displacement_"*".csv",
Tables.table(hcat(ttime,disy_1,disz_1,disy_2,disz_2,disy_3,disz_3,disy_4,disz_4,disy_5,disz_5,disy_6,disz_6,disy_7,disz_7,disy_8,disz_8,forces_y,forces_z)),
writeheader=false)
end
close(wr)
```

G

Pressure integrator validation

ParametricBodies.jl\example\force_integrator_vali_line5_extraBluntEnd.jl

```
using WaterLily
using StaticArrays
using WriteVTK
using ParametricBodies
using LinearAlgebra: cross,dot,norm
using ForwardDiff
import WaterLily: interp
using WaterLily: @loop,inside,inside_u,nds,∇²u
function pressure_force(sim,t=WaterLily.time(sim.flow),T=promote_type(Float64,eltype(sim.flow.p)))
    sim.flow.f .= zero(eltype(sim.flow.p))
    @loop sim.flow.f[I,:] .= sim.flow.p[I]*nds(sim.body,loc(0,I,T),t) over I ∈ inside(sim.flow.p)
    sum(T,sim.flow.f,dims=ntuple(i->i,ndims(sim.flow.p)))[:] |> Array
end
function pressure_force2(sim::Simulation,n,R1,R2,L,tapper;T=promote_type(Float64,eltype(sim.flow.p)))
    forces = zeros(n-1,3)
    ds = 1/n
    global beta = zeros(n-1,2) # for validation
    for i in 1:n-1
        xp = zeros(3)
        s = i*ds
        p = sim.body.curve.(s,0)
        tan = ForwardDiff.derivative(s->sim.body.curve(s),s)
        tann = tan/(norm(tan))

        ### for validation
        rp = p - [L,L,16]
        nn0 = [1,0,0]
        slope = acos(dot(rp,nn0)/(norm(rp)*norm(nn0)))
        slope = π/2-slope
        global beta[i,1] = sin(π-slope)
```

```
        global beta[i,2] = cos(π-slope)
        ### for validation

        n0 = [1f0,0,0]
        k = cross(n0,tann)/norm(cross(n0,tann))
        any(isnan.(k)) && (k=[0,0,0])

        γ = acos(dot(n0,tann)/(norm(n0)*norm(tann)))
        # any(isnan.(γ)) && (γ=0)

        if s > tapper
            RR = ((R2-R1)/(1-tapper))*s + (R2-(R2-R1)/(1-tapper))
            α = atan((R1-R2)/((1-tapper)*L))
            cosα = cos(α)
            sinα = sin(α)

            # println("analytical_x:")
            # println(-(RR^2*(beta[1,1]*beta[1,2])*sinα*π))
            # println("analytical_y:")
            println(-(RR^2*(beta[1,1])^2*cosα*π)) # for validation, analytical solution
            # println(-RR^2*π)
        else RR = R1; cosα = 1; sinα=1; println(-RR^2*π*(beta[1,1])^2); #println(-RR^2*π*(beta[1,1]*beta[1,2]))
        end
        nn = 64
        dθ = 2π/nn
        dl = dθ*RR
        pf1,pf2,pf3 = 0,0,0
        for j in 1:nn
            θ = (j-1)*dθ
            xp[1] = 0
            xp[2] = (RR)*cos(θ)
            xp[3] = (RR)*sin(θ)
            xpp = cos(γ)*xp + (1-cos(γ))*(dot(k,xp))*k + sin(γ)*cross(k,xp) # Rodrigues' rotation formula

            Nds = -xpp/(norm(xpp)) .* [sinα,cosα,cosα]  # normal direction, consider the tapper end
            xppp = (xpp.+p.+1.5) .+ 0.00*Nds
            pres = interp(SA[xppp...],sim.flow.p)
            c = pres.*Nds*dl
            pf1 += c[1]
            pf2 += c[2]
            pf3 += c[3]
        end
        forces[i,1] = pf1
        forces[i,2] = pf2
        forces[i,3] = pf3
    end
    return forces
end
function make_sim(L,thk1,thk2,h,tapper;Re=250,U=1,mem=Array)
    cps = SA_F32[0  0.5 1
                 0  h/2 h
                 0  0.0 0]*L .+ [L,L,16]
    weights = SA_F32[1,1,1]
    knots = SA_F32[0,0,0,1,1,1]
    curve = NurbsCurve(cps,knots,weights)
    function thickness(u,thick1,thick2,tapper)
        u<=tapper ? thick1 : (((thick2/2-thick1/2)/(1-tapper))*u + (thick2/2-(thick2/2-thick1/2)/(1-tapper)))*2
    end
    body = DynamicNurbsBody(curve;thk=(u)->thickness(u,thk1,thk2,tapper),boundary=false)
    Simulation((4L,2L,32),(U,0,0),L;U,v=U/Re,body,T=Float64,mem)
end
function run(L,thk1,thk2,n,h,tapper)
    sim = make_sim(L,thk1,thk2,h,tapper)
    # @inside sim.flow.p[I] = WaterLily.μ₀(sdf(sim.body,loc(0,I),0).+1.0,1)*loc(0,I)[2]
    @inside sim.pois.z[I] = WaterLily.∂(2,I,sim.pois.L) ###
    WaterLily.update!(sim.pois)
    solver!(sim.pois)
```

```
97
98      velocity(a::Simulation) = a.flow.u |> Array;
99      pressure(a::Simulation) = a.flow.p |> Array;
100     _body(a::Simulation) = (measure_sdf!(a.flow.σ, a.body, WaterLily.time(a.flow));
101                             a.flow.σ |> Array;)
102     _nds0(a::Simulation) = (WaterLily.@loop a.flow.f[I,:] .= WaterLily.nds(a.body,loc(0,I),0.0) over I ∈
inside(a.flow.p);
103                             a.flow.f |> Array;)
104
105     custom_attrib = Dict(
106         "u" => velocity, "p" => pressure, "d" => _body, "nds0" => _nds0#, "nds1" => _nds1, "nds2" => _nds2, "nds3" =>
_nds3, "nds4" => _nds4, "cy1" => _sdf1, "cy2" => _sdf2, "cy3" => _sdf3, "cy4" => _sdf4
107     )
108     wr = vtkWriter("ThreeD_cylinder_validation_straight_extraBlunt"; attrib=custom_attrib)
109     write!(wr, sim)
110     close(wr)
111     global p = pressure_force2(sim,n,(thk1)/2,(thk2)/2,L,tapper)
112     println("simulation-y:")
113     for i in 1:n-1
114         println(p[i,2])
115     end
116     println("simulation-z:")
117     for i in 1:n-1
118         println(p[i,3])
119     end
120     println("coordinate")
121     intx=L/n
122     inty=h*L/n
123     for i in 1:n-1
124         println("(",i*intx,",",round(i*inty,digits=3),")")
125     end
126 end
127 @time run(64,6,12,32,0.2,0.8)
```

Figure G.1: Pressure integrator for the cylinder with a taper end.

ParametricBodies.jl\example\force_integrator_vali_line5_parachute.jl

```
1   using WaterLily
2   using StaticArrays
3   using WriteVTK
4   using ParametricBodies
5   using LinearAlgebra: cross,dot,norm
6   using ForwardDiff
7   import WaterLily: interp
8   using WaterLily: @loop,inside,inside_u,nds,∇²u
9   function pressure_force(sim,t=WaterLily.time(sim.flow),T=promote_type(Float64,eltype(sim.flow.p)))
10      sim.flow.f .= zero(eltype(sim.flow.p))
11      @loop sim.flow.f[I,:] .= sim.flow.p[I]*nds(sim.body,loc(0,I,T),t) over I ∈ inside(sim.flow.p)
12      sum(T,sim.flow.f,dims=ntuple(i->i,ndims(sim.flow.p)))[:] |> Array
13  end
14  function pressure_force2(sim::Simulation,n,R1,R2,R3,L,tapper,rope,parachute;T=promote_type(Float64,eltype(sim.flow.p)))
15      forces = zeros(n-1,3)
16      ds = 1/n
17      global beta = zeros(n-1,2) # for validation
18      for i in 1:n-1
19          xp = zeros(3)
20          s = i*ds
21          p = sim.body.curve.(s,0)
22          tan = ForwardDiff.derivative(s->sim.body.curve(s),s)
23          tann = tan/(norm(tan))
24
25          ### for validation
26          rp = p - [L,16,L]
27          nn0 = [1,0,0]
28          slope = acos(dot(rp,nn0)/(norm(rp)*norm(nn0)))
29          slope = π/2-slope
30          global beta[i,1] = sin(π-slope)
```

```
        global beta[i,2] = cos(π-slope)
        ### for validation

        n0 = [1f0,0,0]
        k = cross(n0,tann)/norm(cross(n0,tann))
        any(isnan.(k)) && (k=[0,0,0])

        γ = acos(dot(n0,tann)/(norm(n0)*norm(tann)))
        # any(isnan.(γ)) && (γ=0)

        if s > tapper && s<=rope
            RR = ((R2-R1)/(1-tapper))*s + (R2-(R2-R1)/(1-tapper))
            α = atan((R1-R2)/((1-tapper)*L))
            cosα = cos(α)
            sinα = sin(α)
            println(-(RR^2*(beta[1,1])^2*cosα*π)) # for validation, analytical solution
        elseif s>rope && s<=parachute
            RR = R2; cosα = 1; sinα=1; println(-RR^2*π*(beta[1,1])^2 )
        elseif s>parachute
            RR = R3; cosα = 1; sinα=1; println(-RR^2*π*(beta[1,1])^2 )
        else
            RR = R1; cosα = 1; sinα=1; println(-RR^2*π*(beta[1,1])^2)
        end

        nn = 64
        dθ = 2π/nn
        dl = dθ*RR
        pf1,pf2,pf3 = 0,0,0
        for j in 1:nn
            θ = (j-1)*dθ
            xp[1] = 0
            xp[2] = (RR)*cos(θ)
            xp[3] = (RR)*sin(θ)



            xpp = cos(γ)*xp + (1-cos(γ))*(dot(k,xp))*k + sin(γ)*cross(k,xp) # Rodrigues' rotation formula
            Nds = -xpp/(norm(xpp)) .* [sinα,cosα,cosα]  # normal direction, consider the tapper end
            xppp = (xpp.+p.+1.5) .+ 0.00*Nds
            pres = interp(SA[xppp...],sim.flow.p)
            c = pres.*Nds*dl

            pf1 += c[1] # on the taper end, force in x-direction is not accurate.
            pf2 += c[2]
            pf3 += c[3]
        end
        if s > rope
            forces[i,1] = pf1 #* (R1/R2)^4 # compensate the reuduced EI
            forces[i,2] = pf2 #* (R1/R2)^4
            forces[i,3] = pf3 #* (R1/R2)^4
        else
            forces[i,1] = pf1
            forces[i,2] = pf2
            forces[i,3] = pf3
        end
    end
    return forces
end
function make_sim(L,thk1,thk2,thk3,h,tapper,rope,parachute;Re=250,U=1,mem=Array)
    cps = SA_F32[0  0.5 1
                 0  0.0 0
                 0  h/2 h]*L .+ [L,16,L]
    weights = SA_F32[1,1,1]
    knots = SA_F32[0,0,0,1,1,1]
    curve = NurbsCurve(cps,knots,weights)
    function thickness(u,thick1,thick2,thick3,tapper,rope,parachute)
        # u<=tapper ? thick1 : ((((thick2)/2-(thick1)/2)/(1-tapper))*u + ((thick2)/2-((thick2)/2-(thick1)/2)/(1-
tapper)))*2
```

```
        if u <= tapper
            return thick1
        elseif u > tapper && u <= rope
            return ((((thick2)/2-(thick1)/2)/(1-tapper))*u + ((thick2)/2-((thick2)/2-(thick1)/2)/(1-tapper)))*2
        elseif u > rope && u <= parachute
            return thick2
        else return thick3
        end
    end
    body = DynamicNurbsBody(curve;thk=(u)->thickness(u,thk1,thk2,thk3,tapper,rope,parachute),boundary=false)
    Simulation((4L,32,2L),(U,0,0),L;U,v=U/Re,body,T=Float64,mem)
end
function run(L,thk1,thk2,thk3,n,h,tapper,rope,parachute)
    sim = make_sim(L,thk1,thk2,thk3,n,h,tapper,rope,parachute)
    # @inside sim.flow.p[I] = WaterLily.μ₀(sdf(sim.body,loc(0,I),0).+1.0,1)*loc(0,I)[2]
    @inside sim.pois.z[I] = WaterLily.∂(3,I,sim.pois.L) ###
    WaterLily.update!(sim.pois)
    solver!(sim.pois)

    velocity(a::Simulation) = a.flow.u |> Array;
    pressure(a::Simulation) = a.flow.p |> Array;
    _body(a::Simulation) = (measure_sdf!(a.flow.σ, a.body, WaterLily.time(a.flow));
                            a.flow.σ |> Array;)
    _nds0(a::Simulation) = (WaterLily.@loop a.flow.f[I,:] .= WaterLily.nds(a.body,loc(0,I),0.0) over I ∈
inside(a.flow.p);
                            a.flow.f |> Array;)

    custom_attrib = Dict(
        "u" => velocity, "p" => pressure, "d" => _body, "nds0" => _nds0#, "nds1" => _nds1, "nds2" => _nds2, "nds3" =>
_nds3, "nds4" => _nds4, "cy1" => _sdf1, "cy2" => _sdf2, "cy3" => _sdf3, "cy4" => _sdf4
    )
    wr = vtkWriter("ThreeD_cylinder_validation_straight_parachute"; attrib=custom_attrib)
    write!(wr, sim)
```

```
    close(wr)

    global p = pressure_force2(sim,n,(thk1)/2,(thk2)/2,(thk3)/2,L,tapper,rope,parachute)
    println("simulation-y:")
    for i in 1:n-1
        println(p[i,2])
    end
    println("simulation-z:")
    for i in 1:n-1
        println(p[i,3])
    end
    println("coordinate")
    intx=L/n
    inty=h*L/n
    for i in 1:n-1
        println("(",i*intx,",",round(i*inty,digits=3),")")
    end
end
@time run(64,8,2,12,32,0.3,0.7,0.8,0.9)
```

Figure G.2: Pressure integrator for the cylinder with a parachute.

ParametricBodies.jl\example\force_integrator_vali_arc5.jl

```
using WaterLily
using StaticArrays
using WriteVTK
using ParametricBodies
using LinearAlgebra: cross,dot,norm
using ForwardDiff
import WaterLily: interp
using WaterLily: @loop,inside,inside_u,nds,∇²u
function pressure_force(sim,t=WaterLily.time(sim.flow),T=promote_type(Float64,eltype(sim.flow.p)))
    sim.flow.f .= zero(eltype(sim.flow.p))
    @loop sim.flow.f[I,:] .= sim.flow.p[I]*nds(sim.body,loc(0,I,T),t) over I ∈ inside(sim.flow.p)
    sum(T,sim.flow.f,dims=ntuple(i->i,ndims(sim.flow.p)))[:] |> Array
end
function pressure_force2(sim::Simulation,n,R,L;T=promote_type(Float64,eltype(sim.flow.p)))
    forces = zeros(n-1,3)
    ds = 1/n
    global beta = zeros(n-1,2) # for validation
    for i in 1:n-1
        xp = zeros(3)
        s = i*ds
        p = sim.body.curve.(s,0)
        tan = ForwardDiff.derivative(s->sim.body.curve(s),s)
        tann = tan/(norm(tan))

        rp = p - [2L,32,16]
        nn0 = [-1,0,0]
        slope = acos(dot(rp,nn0)/(norm(rp)*norm(nn0)))
        global beta[i,1] = sin(π-slope)
        global beta[i,2] = cos(π-slope)


        n0 = [1f0,0,0]
        if norm(cross(n0,tann)) == 0
            k = cross(n0,tann)
        else k = cross(n0,tann)/norm(cross(n0,tann))
        end
        γ = acos(dot(n0,tann)/(norm(n0)*norm(tann)))

        nn = 64
        dθ = 2π/nn
        dl = dθ*R
        pf1,pf2,pf3 = 0,0,0
        for j in 1:nn
            θ = (j-1)*dθ
            xp[1] = 0
            xp[2] = (R)*cos(θ)
            xp[3] = (R)*sin(θ)

            xpp = cos(γ)*xp + (1-cos(γ))*(dot(xp,k))*k + cross((sin(γ)*k),xp)
            Nds = -xpp/(norm(xpp)) #.* [sinα,cosα,cosα]  # normal direction, consider the tapper end
            xppp = (xpp.+p.+1.5) .+ 0.00*Nds
            pres = interp(SA[xppp...],sim.flow.p)
            c = pres.*Nds*dl

            pf1 += c[1]
            pf2 += c[2]
            pf3 += c[3]
        end
        forces[i,1] = pf1
        forces[i,2] = pf2
        forces[i,3] = pf3
    end
    return forces
end
```

```
function make_sim(L,thk;Re=250,U=1,mem=Array)
    cps = SA_F32[-1 -1 0  1  1
                 0   1 1  1  0
                 0   0 0  0  0]*L .+ [2L,32,16]
    weights = SA_F32[1,√2/2,1,√2/2,1]
    knots =   SA_F32[0,0,0,1/2,1/2,1,1,1]
    curve = NurbsCurve(cps,knots,weights)
    body = DynamicNurbsBody(curve;thk=(u)->thk,boundary=false)
    Simulation((4L,2L,32),(U,0,0),L;U,ν=U/Re,body,T=Float64,mem)
end
function run(L,thk,n)
    sim = make_sim(L,thk)
    # @inside sim.flow.p[I] = WaterLily.μ₀(sdf(sim.body,loc(0,I),0).+2.0,1)*loc(0,I)[2] ###
    # apply!(x->x[3],sim.flow.p)
    @inside sim.pois.z[I] = WaterLily.∂(2,I,sim.pois.L)
    WaterLily.update!(sim.pois)
    solver!(sim.pois)

    velocity(a::Simulation) = a.flow.u |> Array;
    pressure(a::Simulation) = a.flow.p |> Array;
    _body(a::Simulation) = (measure_sdf!(a.flow.σ, a.body, WaterLily.time(a.flow));
                            a.flow.σ |> Array;)
    _nds0(a::Simulation) = (WaterLily.@loop a.flow.f[I,:] .= WaterLily.nds(a.body,loc(0,I),0.0) over I ∈
inside(a.flow.p);
                            a.flow.f |> Array;)

    custom_attrib = Dict(
        "u" => velocity, "p" => pressure, "d" => _body, "nds0" => _nds0
    )
    wr = vtkWriter("ThreeD_cylinder_validation"; attrib=custom_attrib)
    write!(wr, sim)
    close(wr)

    @time global p = pressure_force2(sim,n,thk/2,L)
    @show(pressure_force(sim)[2])
    @show((thk/2)^2*π*L*π+4/3*(thk/2)^3*π)

    println("analytical_x:")
    for k in 1:n-1
        println(-beta[k,1]*beta[k,2]*π*((thk)/2)^2)
    end
    println("analytical_y:")
    for k in 1:n-1
        println(-beta[k,1]*beta[k,1]*π*((thk)/2)^2)
    end
    println("simulation-y:")
    for i in 1:n-1
        println(p[i,2])
    end
    println("simulation-z:")
    for i in 1:n-1
        println(p[i,3])
    end
end
run(64,6,16)
```

Figure G.3: Pressure integrator for the semi-circle cylinder.

~\Desktop\utils.jl

```
using ParametricBodies: _pforce, _vforce
using WaterLily
using StaticArrays
using WriteVTK
using Splines
using LinearAlgebra
using ForwardDiff
import WaterLily: interp


function force_fea(sim::AbstractSimulation,N,R1,R2,numElem,L,tapper;T=promote_type(Float64,eltype(sim.flow.p)))
    n = N*numElem
    forces = zeros(n,3)

    for i in 1:n
        xp = zeros(3)
        s = integration_points[i]
        p = sim.body.curve.(s,0)
        tan = ForwardDiff.derivative(s->sim.body.curve(s),s)
        tann = tan/norm(tan)

        n0 = [1f0,0,0]
        k = cross(n0,tann)/norm(cross(n0,tann))
        any(isnan.(k)) && (k=[0,0,0])

        γ = acos(dot(n0,tann)/(norm(n0)*norm(tann)))

        if s > tapper
            RR = ((R2-R1)/(1-tapper))*s + (R2-(R2-R1)/(1-tapper))
            α = atan((R1-R2)/((1-tapper)*L))
            cosα = cos(α)
            sinα = sin(α)
        else RR = R1; cosα = 1; sinα = 1
        end

        nn = 64
        dθ = 2π/nn
        dl = dθ*RR
        pf1,pf2,pf3 = 0,0,0
        for j in 1:nn
            θ = (j-1)*dθ
            xp[1] = 0 # make the code cant measure the x-force on the tapper end
            xp[2] = (RR)*cos(θ)
            xp[3] = (RR)*sin(θ)

            xpp = cos(γ)*xp + (1-cos(γ))*(dot(k,xp))*k + sin(γ)*cross(k,xp) # Rodrigues' rotation formula
            Nds = -xpp/(norm(xpp)) .* [sinα,cosα,cosα]
            xppp = (xpp.+p.+1.5) .+0.00*Nds ### pickup the points outside the body
            pres = interp(SA[xppp...],sim.flow.p)
            c = pres.*Nds*dl

            pf1 += c[1] # on the taper end, the force in x-direction is not correct.
            pf2 += c[2]
            pf3 += c[3]
        end
        forces[i,1] = pf1
        forces[i,2] = pf2
        forces[i,3] = pf3
    end
    return (forces')
end

function
force_fea_parachute(sim::AbstractSimulation,N,R1,R2,R3,numElem,L,tapper,rope,parachute;T=promote_type(Float64,eltype(sim.flow
    n = N*numElem
    forces = zeros(n,3)

    for i in 1:n
        xp = zeros(3)
        s = integration_points[i]
```

```
        p = sim.body.curve.(s,0)
        tan = ForwardDiff.derivative(s->sim.body.curve(s),s)
        tann = tan/norm(tan)

        n0 = [1f0,0,0]
        k = cross(n0,tann)/norm(cross(n0,tann))
        any(isnan.(k)) && (k=[0,0,0])

        γ = acos(dot(n0,tann)/(norm(n0)*norm(tann)))

        if s > tapper && s <= rope # taper end
            RR = ((R2-R1)/(1-tapper))*s + (R2-(R2-R1)/(1-tapper))
            α = atan((R1-R2)/((1-tapper)*L))
            cosα = cos(α)
            sinα = sin(α)
        elseif s > rope && s <= parachute # rope part
            RR = R2; cosα = 1; sinα = 1
        elseif s > parachute # parachute part
            RR = R3; cosα = 1; sinα = 1
        else
            RR = R1; cosα = 1; sinα = 1 # cylinder part
        end

        nn = 64
        dθ = 2π/nn
        dl = dθ*RR
        pf1,pf2,pf3 = 0,0,0
        for j in 1:nn
            θ = (j-1)*dθ
            xp[1] = 0 # make the code cant measure the x-force on the tapper end
            xp[2] = (RR)*cos(θ)
            xp[3] = (RR)*sin(θ)

            xpp = cos(γ)*xp + (1-cos(γ))*(dot(k,xp))*k + sin(γ)*cross(k,xp) # Rodrigues' rotation formula
            Nds = -xpp/(norm(xpp)) .* [sinα,cosα,cosα]
            xppp = (xpp.+p.+1.5) .+0.00*Nds ### pickup the points outside the body
            pres = interp(SA[xppp...],sim.flow.p)
            c = pres.*Nds*dl

            pf1 += c[1] # on the taper end, the force in x-direction is not correct.
            pf2 += c[2]
            pf3 += c[3]
        end
        if s > rope
            forces[i,1] = pf1 #* (R1/R2)^4 # compensate the reuduced EI
            forces[i,2] = pf2 #* (R1/R2)^4
            forces[i,3] = pf3 #* (R1/R2)^4
        else
            forces[i,1] = pf1
            forces[i,2] = pf2
            forces[i,3] = pf3
        end
    end
    return (forces')
end
```

Figure G.4: Force integrator in FSI solver.

H

Integrator validation result

f_{sim} and $f_{sim,1}$ are the numerical results with $\delta = 0$ and $\delta = 0.01$. f_{ana} is the analytical result.

Table H.1: y component of the force on the cylinder with a taper end, $L_{sim} = 64$, $n_{pnt} = 31$.

$D_{sim} = 6$, $D_{end} = 12$, $h_\xi = 0.2$			$D_{sim} = 6$, $D_{end} = 0$, $h_\xi = 0.3$		
(x, y)	f_{sim}	f_{ana}	(x, y)	f_{sim}	f_{ana}
(2.0,0.4)	-27.18616	-27.18686	(2.0,0.6)	-25.93182	-25.93976
(4.0,0.8)	-27.18608	-27.18686	(4.0,1.2)	-25.93117	-25.93976
(6.0,1.2)	-27.18604	-27.18686	(6.0,1.8)	-25.93093	-25.93976
(8.0,1.6)	-27.18601	-27.18686	(8.0,2.4)	-25.93091	-25.93976
(10.0,2.0)	-27.18600	-27.18686	(10.0,3.0)	-25.93101	-25.93976
(12.0,2.4)	-27.18599	-27.18686	(12.0,3.6)	-25.93108	-25.93976
(14.0,2.8)	-27.18597	-27.18686	(14.0,4.2)	-25.93110	-25.93976
(16.0,3.2)	-27.18596	-27.18686	(16.0,4.8)	-25.93096	-25.93976
(18.0,3.6)	-27.18596	-27.18686	(18.0,5.4)	-25.93077	-25.93976
(20.0,4.0)	-27.18595	-27.18686	(20.0,6.0)	-25.93053	-25.93976
(22.0,4.4)	-27.18593	-27.18686	(22.0,6.6)	-25.93039	-25.93976
(24.0,4.8)	-27.18590	-27.18686	(24.0,7.2)	-25.93028	-25.93976
(26.0,5.2)	-27.18587	-27.18686	(26.0,7.8)	-25.93030	-25.93976
(28.0,5.6)	-27.18585	-27.18686	(28.0,8.4)	-25.93036	-25.93976
(30.0,6.0)	-27.18583	-27.18686	(30.0,9.0)	-25.93054	-25.93976
(32.0,6.4)	-27.18580	-27.18686	(32.0,9.6)	-25.93072	-25.93976
(34.0,6.8)	-27.18577	-27.18686	(34.0,10.2)	-25.93099	-25.93976
(36.0,7.2)	-27.18576	-27.18686	(36.0,10.8)	-25.93135	-25.93976
(38.0,7.6)	-27.18576	-27.18686	(38.0,11.4)	-25.93166	-25.93976
(40.0,8.0)	-27.18576	-27.18686	(40.0,12.0)	-25.93178	-25.93976
(42.0,8.4)	-27.18575	-27.18686	(42.0,12.6)	-25.93184	-25.93976
(44.0,8.8)	-27.18575	-27.18686	(44.0,13.2)	-25.93177	-25.93976
(46.0,9.2)	-27.18576	-27.18686	(46.0,13.8)	-25.93172	-25.93976
(48.0,9.6)	-27.18576	-27.18686	(48.0,14.4)	-25.93187	-25.93976
(50.0,10.0)	-27.18576	-27.18686	(50.0,15.0)	-25.93229	-25.93976
(52.0,10.4)	-29.88012	-29.88166	(52.0,15.6)	-22.19099	-22.19710
(54.0,10.8)	-39.31510	-39.31662	(54.0,16.2)	-15.41036	-15.41465
(56.0,11.2)	-50.04224	-50.04403	(56.0,16.8)	-9.86256	-9.86538
(58.0,11.6)	-62.06168	-62.06390	(58.0,17.4)	-5.54809	-5.54928
(60.0,12.0)	-75.37379	-75.37623	(60.0,18.0)	-2.46591	-2.46634
(62.0,12.4)	-89.97815	-89.98103	(62.0,18.6)	-0.61652	-0.61659

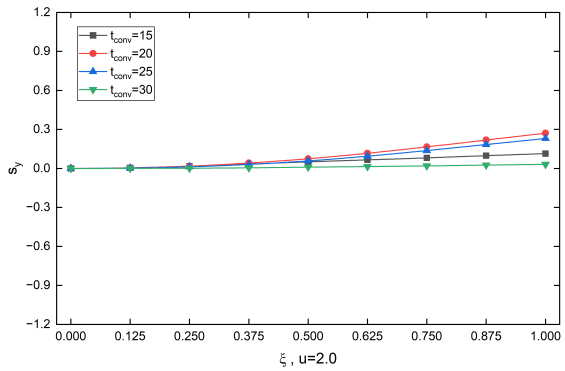
Table H.2: z component of the force on the cylinder with a parachute, $L_{sim} = 64$, $n_{pnt} = 31$.

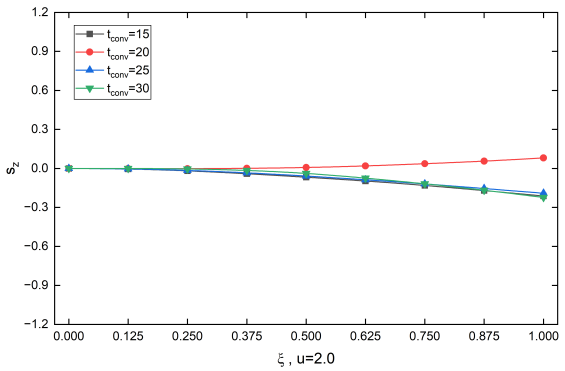
$h_\xi = 0.2$			$h_\xi = 0.3$			
(x, z)	f_{sim}	f_{ana}	(x, z)	f_{sim}	$f_{sim,1}$	f_{ana}
(2.0,0.4)	-48.33125	-48.33219	(2.0,0.6)	-46.11435	-45.99952	-46.11512
(4.0,0.8)	-48.33102	-48.33219	(4.0,1.2)	-46.11429	-45.99901	-46.11512
(6.0,1.2)	-48.33090	-48.33219	(6.0,1.8)	-46.11429	-45.99900	-46.11512
(8.0,1.6)	-48.33099	-48.33219	(8.0,2.4)	-46.11430	-45.99982	-46.11512
(10.0,2.0)	-48.33086	-48.33219	(10.0,3.0)	-46.11423	-45.99894	-46.11512
(12.0,2.4)	-48.33092	-48.33219	(12.0,3.6)	-46.11423	-45.99975	-46.11512
(14.0,2.8)	-48.33078	-48.33219	(14.0,4.2)	-46.11417	-45.99888	-46.11512
(16.0,3.2)	-48.33081	-48.33219	(16.0,4.8)	-46.11414	-45.99885	-46.11512
(18.0,3.6)	-48.33093	-48.33219	(18.0,5.4)	-46.11415	-45.99934	-46.11512
(20.0,4.0)	-48.33073	-48.33219	(20.0,6.0)	-46.11411	-45.99882	-46.11512
(22.0,4.4)	-48.33087	-48.33219	(22.0,6.6)	-46.11409	-45.99928	-46.11512
(24.0,4.8)	-48.33068	-48.33219	(24.0,7.2)	-46.11403	-45.99874	-46.11512
(26.0,5.2)	-48.33057	-48.33219	(26.0,7.8)	-46.11403	-45.99874	-46.11512
(28.0,5.6)	-48.33064	-48.33219	(28.0,8.4)	-46.11405	-45.99957	-46.11512
(30.0,6.0)	-48.33050	-48.33219	(30.0,9.0)	-46.11402	-45.99874	-46.11512
(32.0,6.4)	-48.33056	-48.33219	(32.0,9.6)	-46.11406	-45.99958	-46.11512
(34.0,6.8)	-48.33042	-48.33219	(34.0,10.2)	-46.11402	-45.99873	-46.11512
(36.0,7.2)	-48.33047	-48.33219	(36.0,10.8)	-46.11399	-45.99870	-46.11512
(38.0,7.6)	-48.33061	-48.33219	(38.0,11.4)	-46.11399	-45.99917	-46.11512
(40.0,8.0)	-48.33044	-48.33219	(40.0,12.0)	-46.11396	-45.99867	-46.11512
(42.0,8.4)	-48.33063	-48.33219	(42.0,12.6)	-46.11396	-45.99914	-46.11512
(44.0,8.8)	-48.33051	-48.33219	(44.0,13.2)	-46.11396	-45.99867	-46.11512
(46.0,9.2)	-43.37935	-43.38089	(46.0,13.8)	-41.38989	-41.28262	-41.39094
(48.0,9.6)	-36.55943	-36.56073	(48.0,14.4)	-34.88285	-34.78438	-34.88363
(50.0,10.0)	-30.32241	-30.32349	(50.0,15.0)	-28.93186	-28.84218	-28.93250
(52.0,10.4)	-3.02065	-3.02076	(52.0,15.6)	-2.88211	-2.85329	-2.88220
(54.0,10.8)	-3.02064	-3.02076	(54.0,16.2)	-2.88211	-2.85329	-2.88220
(56.0,11.2)	-3.02065	-3.02076	(56.0,16.8)	-2.88209	-2.85327	-2.88220
(58.0,11.6)	-108.74438	-108.74744	(58.0,17.4)	-103.75697	-103.58404	-103.75903
(60.0,12.0)	-108.74399	-108.74744	(60.0,18.0)	-103.75682	-103.58389	-103.75903
(62.0,12.4)	-108.74394	-108.74744	(62.0,18.6)	-103.75682	-103.58389	-103.75903

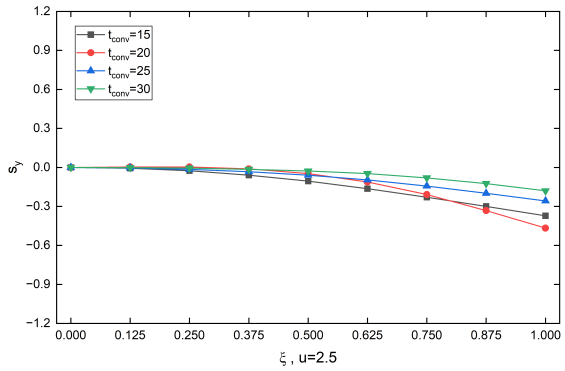
Table H.3: y component of the force on the semi-circle cylinder, $L_{sim} = 64$, $n_{pnt} = 15$.

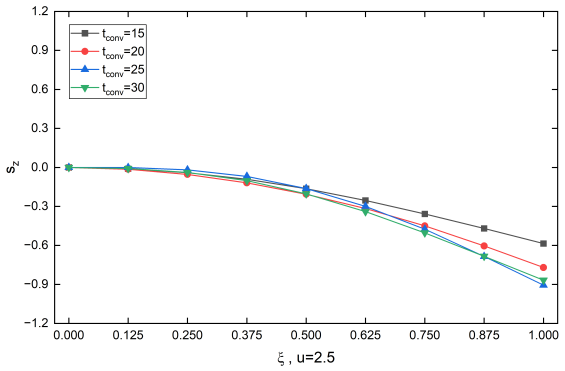
$D_{sim} = 6$			$D_{sim} = 9$		
θ_ξ	f_{sim}	f_{ana}	θ_ξ	f_{sim}	f_{ana}
1/16	-0.93598	-0.93618	1/16	-2.10564	-2.10640
2/16	-3.83054	-3.83099	2/16	-8.61808	-8.61974
3/16	-8.46569	-8.46642	3/16	-19.04698	-19.04946
4/16	-14.13625	-14.13717	4/16	-31.80602	-31.80863
5/16	-19.80697	-19.80791	5/16	-44.56477	-44.56780
6/16	-24.44249	-24.44334	6/16	-54.99474	-54.99751
7/16	-27.33736	-27.33816	7/16	-61.50820	-61.51085
8/16	-28.27359	-28.27433	8/16	-63.61457	-63.61725
9/16	-27.33736	-27.33816	9/16	-61.50820	-61.51085
10/16	-24.44249	-24.44334	10/16	-54.99474	-54.99751
11/16	-19.80697	-19.80791	11/16	-44.56477	-44.56780
12/16	-14.13625	-14.13717	12/16	-31.80602	-31.80863
13/16	-8.46569	-8.46642	13/16	-19.04698	-19.04946
14/16	-3.83054	-3.83099	14/16	-8.61808	-8.61974
15/16	-0.93598	-0.93618	15/16	-2.10564	-2.10640

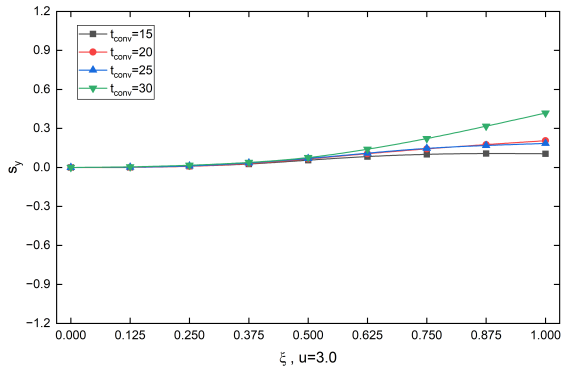
Cylinder shapes in FSI validation

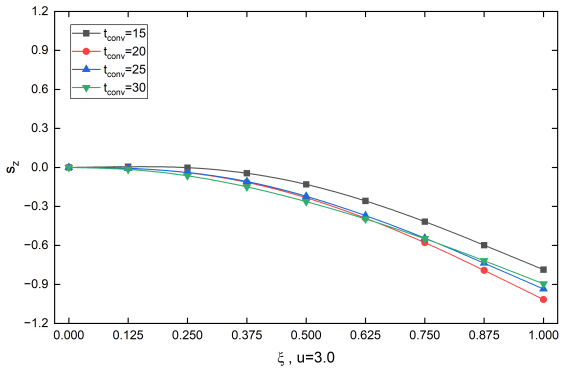












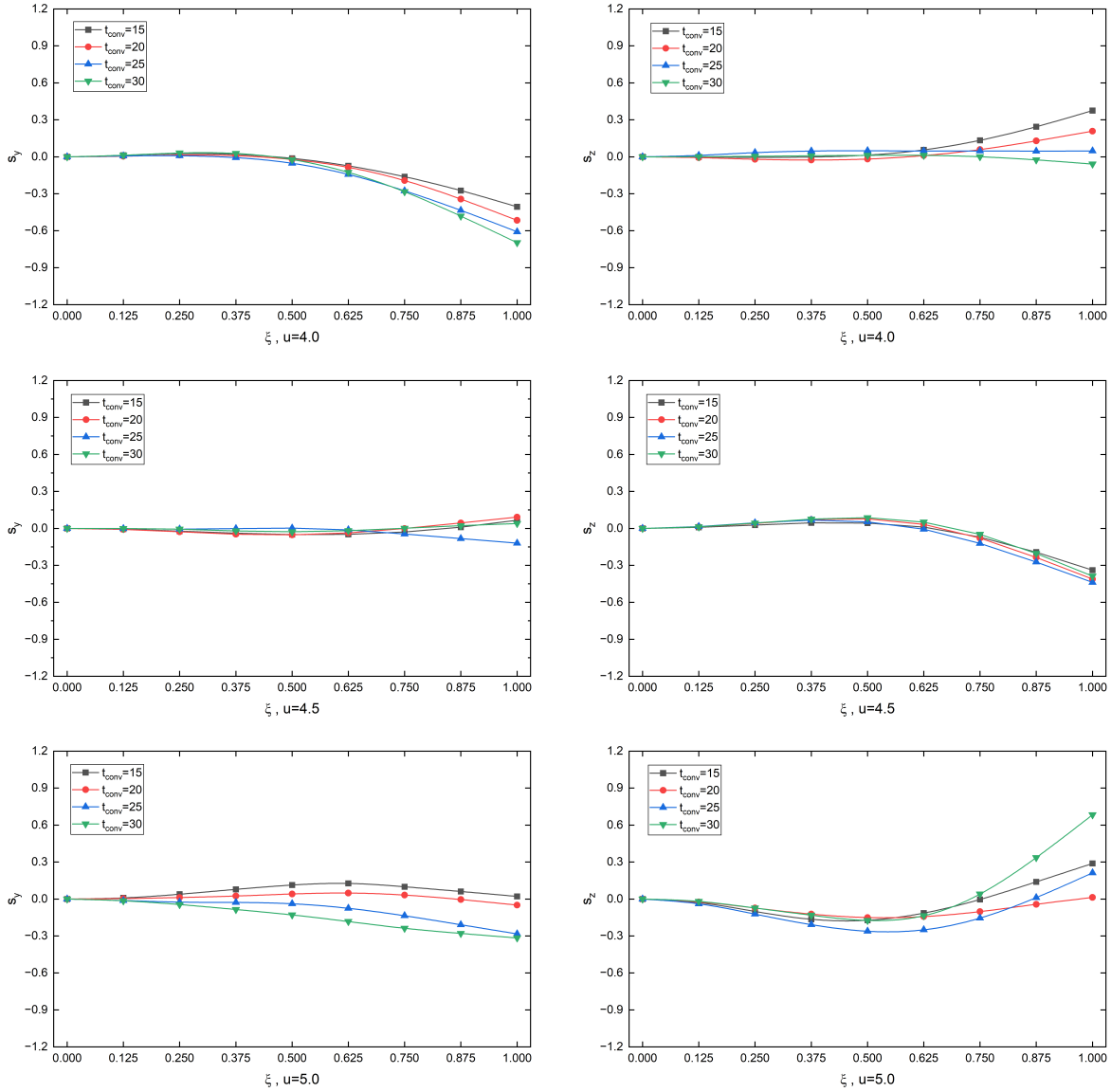
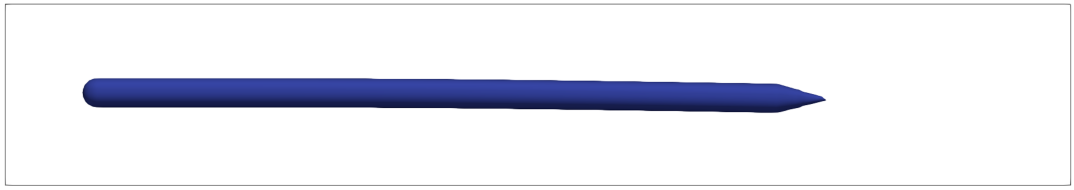


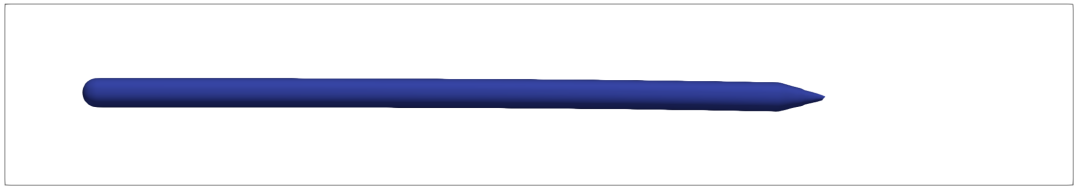
Figure I.1: Clamped-free cylinder's central axis shapes at various t_{conv}.

(a) xoy view. $t_{conv} = 30.0$.

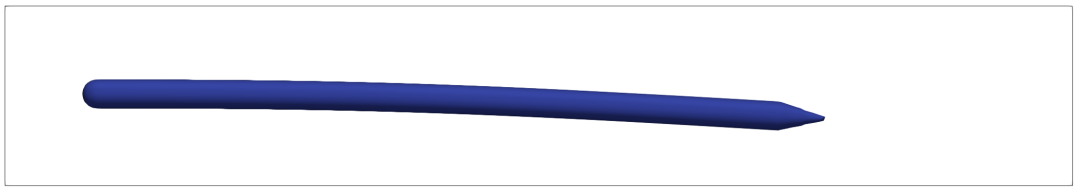


(b) xoz view. $t_{conv} = 30.0$.

Figure I.2: $u = 2.0$.

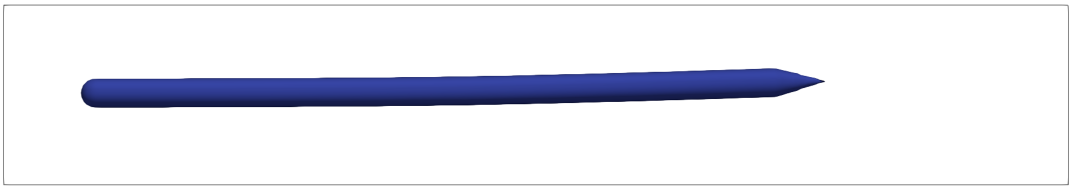


(a) xoy view. $t_{conv} = 30.0$.

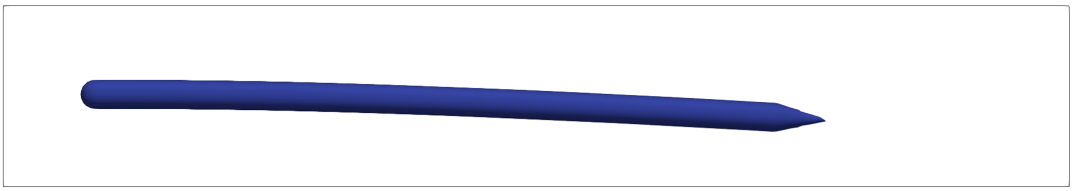


(b) xoz view. $t_{conv} = 30.0$.

Figure I.3: $u = 2.5$.

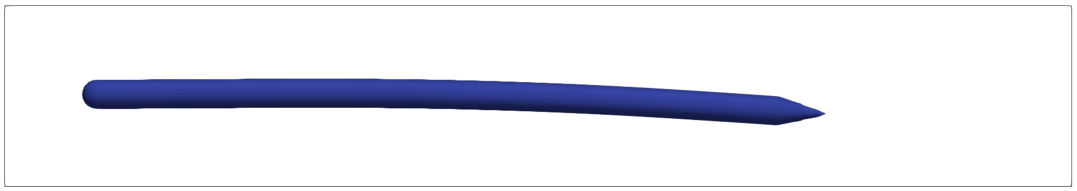


(a) xoy view. $t_{conv} = 30.0$.

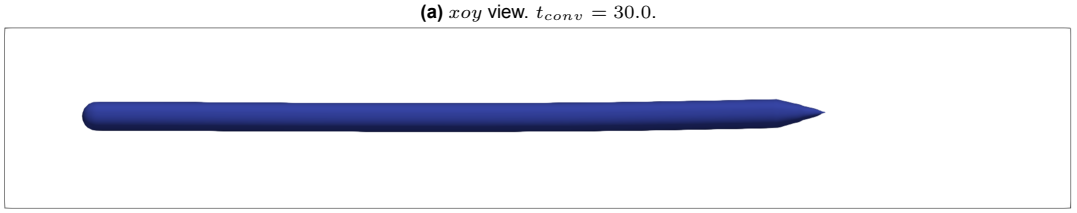


(b) xoz view. $t_{conv} = 30.0$.

Figure I.4: $u = 3.0$.

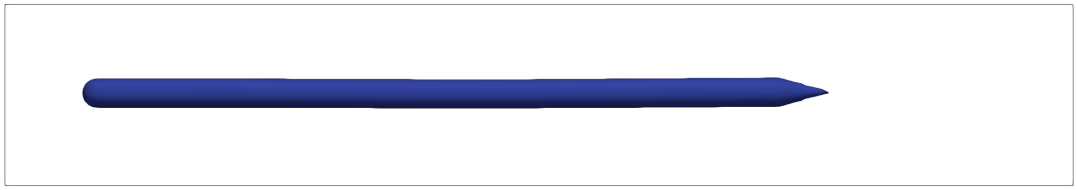


(a) xoy view. $t_{conv} = 30.0$.

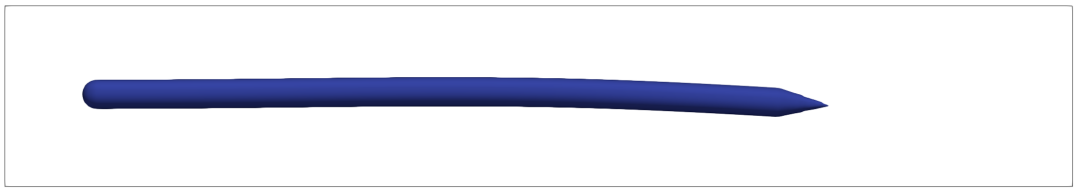


(b) xoz view. $t_{conv} = 28.6$.

Figure I.5: $u = 4.0$.

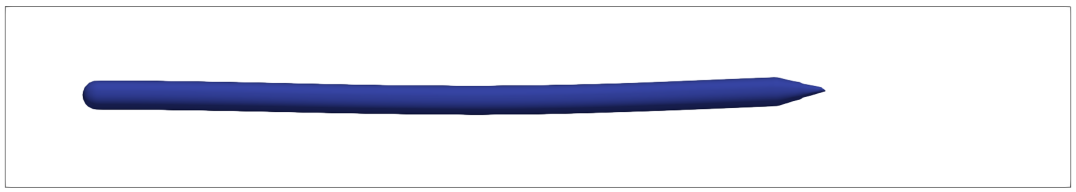


(a) xoy view. $t_{conv} = 30.0$.

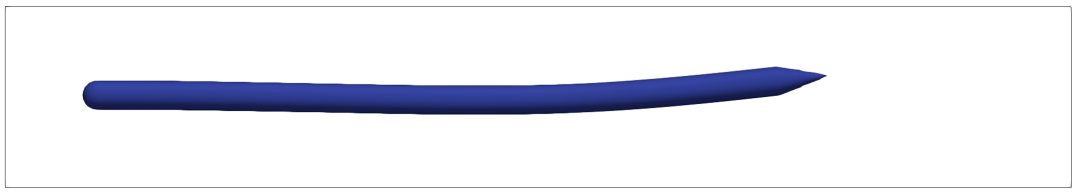


(b) xoz view. $t_{conv} = 30.0$.

Figure I.6: $u = 4.5$.



(a) xoy view. $t_{conv} = 23.2$.



(b) xoz view. $t_{conv} = 30.0$.

Figure I.7: $u = 5.0$.

J

Code for investigation

Filament.jl\code\src\FlappingFillament_tapper_realCase.jl

```
using WaterLily
using ParametricBodies
using Splines
using StaticArrays
using LinearAlgebra
using WriteVTK
using CSV,Tables,DataFrames
include("../ext/vis.jl")
include("Coupling.jl")
include("utils.jl")

ptLeft = 0.0
ptRight = 1.0

# Geometry
AR = 15.4 / 0.653 # aspect ratio of cylinder #####
ARR = round(Int,AR)

thk1 = 7.556975354821782
thk2 = 0
thkk2 = round(Int,thk2)

tapper_leng = 31 / 17 # dimension_less tapper length
tapper = AR / (AR + tapper_leng)
tl = "31_17"

L = thk1 * AR / tapper
TL = L * (ptRight - ptLeft) # total length with tapper end in numerical space
LL = round(Int,L)

# Material properties and mesh
numElem = round(Int, 12*AR/(15.4/0.653))
numE = round(Int,numElem)
degP = 3

# exp data
rho_exp = 1026 # sea water
μ_exp = 1.22 * 10^(-3) # 15 ceils, sea water

```

```
D_exp = 0.03 # m
L_exp = AR * D_exp # m
I_exp = π*D_exp^4/64

U_exp = 5*0.5144 # m/s #####
uu = round(Int,10*U_exp) # for the file name
E_exp = 12*10^6 # Pa
EI_exp = E_exp*I_exp
Re_exp = rho_exp * D_exp * U_exp / μ_exp ## Reynolds number
Ca_exp = EI_exp / I_exp / (rho_exp*U_exp^2) ## Cauchy number
Ca_exp_expand = EI_exp / (rho_exp*U_exp^2*L_exp^4)

# simulation parameters
U = 1 * (ptRight - ptLeft)
E_sim = Ca_exp * (1*U^2)
EI_sim = E_sim * (π*(thk1)^4) / 64
EI_sim_test = Ca_exp_expand * (1*U^2*(TL*tapper)^4)
EI = EI_sim / TL^3 # EI for FEA solver
EA = 100_000.0  # make inextensible
density(ξ) = 1 # assume nutral buoyancy, density of solid structure
# mesh
mesh, gauss_rule = Mesh1D(ptLeft, ptRight, numElem, degP)

# boundary conditions
Dirichlet_BC = [
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=1),
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=2)
]
Neumann_BC = [
    Boundary1D("Neumann", ptLeft, 0.0; comp=1),
    Boundary1D("Neumann", ptLeft, 0.0; comp=2)
]

# make a structure
struc = DynamicFEOperator(mesh, gauss_rule, EI, EA,
                          Dirichlet_BC, Neumann_BC, ρ=density; ρ∞=0.0)

## Simulation parameters
Re=Re_exp
# Re=Re/10 # change Re only if you need
ϵ=0.5


```

```
# construct from mesh, this can be tidy
u⁰ = SMatrix{3,size(mesh.controlPoints,2)}(mesh.controlPoints[1:3,:].*L.+[24,24,24].+0.5) #controlPoints
half_doma = 24.5

# flow sim
function thickness(u,thick1,thick2,tapper)
    u<=tapper ? thick1 : ((((thick2)/2-(thick1)/2)/(1-tapper))*u + ((thick2)/2-((thick2)/2-(thick1)/2)/(1-tapper)))*2
end
body = DynamicNurbsBody(NurbsCurve(u⁰,mesh.knots,mesh.weights);thk=(u)->thickness(u,thk1,thk2,tapper),boundary=false)

# force function
integration_points = uv_integration(struc)/(ptRight-ptLeft) ## utils.jl

# make a coupled sim
sim = CoupledSimulation((280,48,48),(U,0,0),L,body,struc,IQNCoupling;
                        U,ν=U*(thk1)/Re,ϵ,T=Float64,relax=0.05,maxCol=6)

q_poke = 0.01

# sim time
t₀ = round(sim_time(sim)); duration = 30; step = 0.2
# write para_view
velocity(a::CoupledSimulation) = a.flow.u |> Array;
pressure(a::CoupledSimulation) = a.flow.p |> Array;
_body(a::CoupledSimulation) = (measure_sdf!(a.flow.σ, a.body, WaterLily.time(a.flow));
                        a.flow.σ |> Array;)
custom_attrib = Dict(
    "u" => velocity, "p" => pressure, "d" => _body)

name = "ExpU$(uu)_L$(LL)_Gpnts$(numE)_thk2$(thkk2)_AR$(ARR)_CF_RealCase_truncErrTest123"*tl
wr = vtkWriter(name; attrib=custom_attrib)

disy_1 = []
disz_1 = []
disy_2 = []
disz_2 = []
disy_3 = []
disz_3 = []
disy_4 = []
disz_4 = []
disy_5 = []
disz_5 = []
```

```
disy_6 = []
disz_6 = []
disy_7 = []
disz_7 = []
disy_8 = []
disz_8 = []
forces_y = []
forces_z = []
ttime = []
# time loop
@time for tᵢ in range(t₀,t₀+duration;step)

    # update until time tᵢ in the background
    t = sum(sim.flow.Δt[1:end-1])

    while t < tᵢ*sim.L/sim.U

        println("  tᵢ=$tᵢ, t=$(round(t,digits=2)), Δt=$(round(sim.flow.Δt[end],digits=2))")

        # save at start of iterations
        store!(sim); iter=1;

        # iterative loop
        while true

            #  integrate once in time
            solve_step!(sim.struc, sim.forces[2:3,:], sim.flow.Δt[end]/sim.L) # apply FEA solver

            # update flow, this requires scaling the displacements
            sim.body = ParametricBodies.update!(sim.body,u⁰.+ L*sim.pnts,sim.flow.Δt[end]) ###!!!
            measure!(sim,t); mom_step!(sim.flow,sim.pois) # body and flow from flow solver.

            # apply!(x->x[2],sim.flow.p)
            # get new coupling variable
            sim.pnts[2:3,:] = points(sim.struc) # use FEA solver to obtain the displacement of control points. Here is the IGA.
            sim.forces = force_fea(sim,degP+1,(thk1)/2,(thk2)/2,numElem,sim.L,tapper)
            if tᵢ <= 0.2
                sim.forces[3,:] .+= q_poke
            end

            # accelerate coupling
            print("    iteration: ",iter)

            converged = update!(sim.cpl, sim.pnts, sim.forces; atol=1e-2) # coupling.jl

            # check for convengence
            (converged || iter+1 > 50) && break

            # if we have not converged, we must revert
            revert!(sim); iter += 1
        end
        t += sim.flow.Δt[end]
        push!(disy_1, sim.body.curve.(0.125)[2] .- half_doma)
        push!(disz_1, sim.body.curve.(0.125)[3] .- half_doma)
        push!(disy_2, sim.body.curve.(0.25)[2] .- half_doma)
        push!(disz_2, sim.body.curve.(0.25)[3] .- half_doma)
        push!(disy_3, sim.body.curve.(0.375)[2] .- half_doma)
        push!(disz_3, sim.body.curve.(0.375)[3] .- half_doma)
        push!(disy_4, sim.body.curve.(0.5)[2] .- half_doma)
        push!(disz_4, sim.body.curve.(0.5)[3] .- half_doma)
        push!(disy_5, sim.body.curve.(0.625)[2] .- half_doma)
        push!(disz_5, sim.body.curve.(0.625)[3] .- half_doma)
        push!(disy_6, sim.body.curve.(0.75)[2] .- half_doma)
        push!(disz_6, sim.body.curve.(0.75)[3] .- half_doma)
        push!(disy_7, sim.body.curve.(0.875)[2] .- half_doma)
        push!(disz_7, sim.body.curve.(0.875)[3] .- half_doma)
        push!(disy_8, sim.body.curve.(1.0)[2] .- half_doma)
        push!(disz_8, sim.body.curve.(1.0)[3] .- half_doma)
        push!(forces_y, sim.forces[2,:])
        push!(forces_z, sim.forces[3,:])
        push!(ttime,t*sim.U/sim.L)
    end
    write!(wr, sim)
    CSV.write(name*"_displacement_"*".csv",
Tables.table(hcat(ttime,disy_1,disz_1,disy_2,disz_2,disy_3,disz_3,disy_4,disz_4,disy_5,disz_5,disy_6,disz_6,disy_7,disz_7,disy_8,disz_8,forces_y,forces_z)),
writeheader=false)
end
close(wr)

# uu=integration_points[48]
# thkk = (((thk2)/2-(thk1)/2)/(1-tapper))*uu + ((thk2)/2-((thk2)/2-(thk1)/2)/(1-tapper))
# α = atan(((thk1)/2-(thk2)/2)/((1-tapper)*L))
# dV = -thkk^2*π * cos(α)
# @show(sim.forces[:,48])
# @show(dV)
```

Figure J.1: Code for the cylinder with a taper end.

Filament.jl\code\src\FlappingFillament_tapper_realCase_parachute.jl

```
using WaterLily
using ParametricBodies
using Splines
using StaticArrays
using LinearAlgebra
using WriteVTK
using CSV,Tables,DataFrames
include("../ext/vis.jl")
include("Coupling.jl")
include("utils.jl")

ptLeft = 0.0
ptRight = 1.0

# Geometry
AR = 40 # aspect ratio of cylinder #####
ARR = round(Int,AR)

thk1 = 7.556975354821782
thk2 = 2
thkk2 = round(Int,thk2)
thk3 = 11
thkk3 = round(Int,thk3)

tapper_leng = 31 / 17 # dimension_less tapper length
rope_leng = 1.2 # dimension_less rope length
parachute_leng = 0.8 # dimension_less parachute length
tapper = AR / (AR + tapper_leng + parachute_leng + rope_leng)
rope = tapper +  tapper_leng / (AR + tapper_leng + parachute_leng +rope_leng)
parachute = rope + rope_leng / (AR + tapper_leng + parachute_leng +rope_leng)
# test = parachute + parachute_leng / (AR + tapper_leng + parachute_leng +rope_leng)
tl = "31_17"

L = thk1 * AR / tapper
TL = L * (ptRight - ptLeft) # total length with tapper end in numerical space
LL = round(Int,L)

# Material properties and mesh
numElem = round(Int, 12*AR/(15.4/0.653))

numE = round(Int,numElem)
degP = 3

# exp data
rho_exp = 1026 # sea water
μ_exp = 1.22 * 10^(-3) # 15 ceils, sea water

D_exp = 0.03 # m
L_exp = AR * D_exp # m
I_exp = π*D_exp^4/64

U_exp = 5*0.5144 # m/s #####
uu = round(Int,10*U_exp) # for the file name
E_exp = 12*10^6 # Pa
EI_exp = E_exp*I_exp
Re_exp = rho_exp * D_exp * U_exp / μ_exp ## Reynolds number
Ca_exp = EI_exp / I_exp / (rho_exp*U_exp^2) ## Cauchy number
Ca_exp_expand = EI_exp / (rho_exp*U_exp^2*L_exp^4)

# simulation parameters
U = 1 * (ptRight - ptLeft)
E_sim = Ca_exp * (1*U^2)
EI_sim = E_sim * (π*(thk1)^4) / 64
EI_sim_test = Ca_exp_expand * (1*U^2*(TL*tapper)^4)
EI = EI_sim / TL^3 # EI for FEA solver
EA = 100_000.0  # make inextensible
density(ξ) = 1 # assume nutral buoyancy, density of solid structure

# mesh
mesh, gauss_rule = Mesh1D(ptLeft, ptRight, numElem, degP)

# boundary conditions
Dirichlet_BC = [
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=1),
    Boundary1D("Dirichlet", ptLeft, 0.0; comp=2)
]
Neumann_BC = [
    # Boundary1D("Neumann", ptLeft, 0.0; comp=1),
    # Boundary1D("Neumann", ptLeft, 0.0; comp=2)
]

# make a structure
```

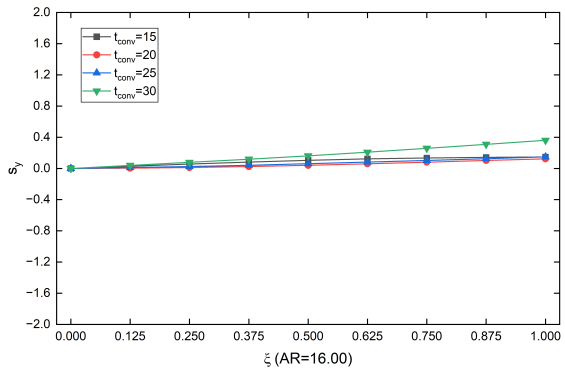
```
struc = DynamicFEOperator(mesh, gauss_rule, EI, EA,
                          Dirichlet_BC, Neumann_BC, ρ=density; ρ∞=0.0)

## Simulation parameters
Re=Re_exp
ϵ=0.5

# construct from mesh, this can be tidy
u⁰ = SMatrix{3,size(mesh.controlPoints,2)}(mesh.controlPoints[1:3,:].*L.+[24,24,24].+0.5) #controlPoints
half_doma = 24.5

# flow sim
function thickness(u,thick1,thick2,thick3,tapper,rope,parachute)
    # u<=tapper ? thick1 : ((((thick2)/2-(thick1)/2)/(1-tapper))*u + ((thick2)/2-((thick2)/2-(thick1)/2)/(1-tapper)))*2
    if u <= tapper
        return thick1
    elseif u > tapper && u <= rope
        return ((((thick2)/2-(thick1)/2)/(1-tapper))*u + ((thick2)/2-((thick2)/2-(thick1)/2)/(1-tapper)))*2
    elseif u > rope && u <= parachute
        return thick2
    else return thick3
    end
end
body = DynamicNurbsBody(NurbsCurve(u⁰,mesh.knots,mesh.weights);thk=(u)->thickness(u,thk1,thk2,thk3,tapper,rope,parachute),boundary=false)

# force function
integration_points = uv_integration(struc)/(ptRight-ptLeft) ## utils.jl

# make a coupled sim
sim = CoupledSimulation((420,48,48),(U,0,0),L,body,struc,IQNCoupling;
                        U,v=U*(thk1)/Re,ϵ,T=Float64,relax=0.05,maxCol=6)

q_poke = 0.01

# sim time
tₒ = round(sim_time(sim)); duration = 30; step = 0.2
# write para_view
velocity(a::CoupledSimulation) = a.flow.u |> Array;
pressure(a::CoupledSimulation) = a.flow.p |> Array;
_body(a::CoupledSimulation) = (measure_sdf!(a.flow.σ, a.body, WaterLily.time(a.flow));
                       a.flow.σ |> Array;)
custom_attrib = Dict(
    "u" => velocity, "p" => pressure, "d" => _body)

name = "ExpU$(uu)_L$(LL)_Gpnts$(numE)_thk2$(thkk2)_thk3$(thkk3)_AR$(ARR)_PinnedFree_RealCase_parachute_truncTest123"*tl
wr = vtkWriter(name; attrib=custom_attrib)

disy_1 = []
disz_1 = []
disy_2 = []
disz_2 = []
disy_3 = []
disz_3 = []
disy_4 = []
disz_4 = []
disy_5 = []
disz_5 = []
disy_6 = []
disz_6 = []
disy_7 = []
disz_7 = []
disy_8 = []
disz_8 = []
forces_y = []
forces_z = []
ttime = []
# time loop
@time for tᵢ in range(tₒ,tₒ+duration;step)

    # update until time tᵢ in the background
    t = sum(sim.flow.Δt[1:end-1])

    while t < tᵢ*sim.L/sim.U

        println("  tᵢ=$tᵢ, t=$(round(t,digits=2)), Δt=$(round(sim.flow.Δt[end],digits=2))")

        # save at start of iterations
        store!(sim); iter=1;

        # iterative loop
        while true

            #  integrate once in time
            solve_step!(sim.struc, sim.forces[2:3,:], sim.flow.Δt[end]/sim.L) # apply FEA solver
```

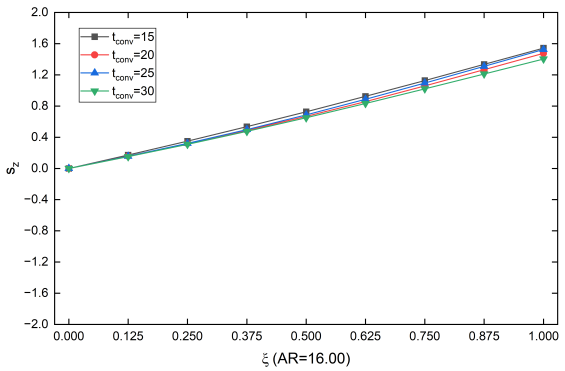
```
166
167             # update flow, this requires scaling the displacements
168             sim.body = ParametricBodies.update!(sim.body,u⁰.+ L*sim.pnts,sim.flow.Δt[end]) ###!!!
169             measure!(sim,t); mom_step!(sim.flow,sim.pois) # body and flow from flow solver.
170
171             # apply!(x->x[2],sim.flow.p)
172             # get new coupling variable
173             sim.pnts[2:3,:] = points(sim.struc) # use FEA solver to obtain the displacement of control points. Here is the IGA.
174             sim.forces = force_fea_parachute(sim,degP+1,(thk1)/2,(thk2)/2,(thk3)/2,numElem,sim.L,tapper,rope,parachute)
175             if tᵢ <= 0.2
176                 sim.forces[3,:] .+= q_poke
177             end
178
179             # accelerate coupling
180             print("    iteration: ",iter)
181             converged = update!(sim.cpl, sim.pnts, sim.forces; atol=1e-2) # coupling.jl
182
183             # check for convengence
184             (converged || iter+1 > 50) && break
185
186             # if we have not converged, we must revert
187             revert!(sim); iter += 1
188         end
189         t += sim.flow.Δt[end]
190         push!(disy_1, sim.body.curve.(0.125)[2] .- half_doma)
191         push!(disz_1, sim.body.curve.(0.125)[3] .- half_doma)
192         push!(disy_2, sim.body.curve.(0.25)[2] .- half_doma)
193         push!(disz_2, sim.body.curve.(0.25)[3] .- half_doma)
194         push!(disy_3, sim.body.curve.(0.375)[2] .- half_doma)
195         push!(disz_3, sim.body.curve.(0.375)[3] .- half_doma)
196         push!(disy_4, sim.body.curve.(0.5)[2] .- half_doma)
197         push!(disz_4, sim.body.curve.(0.5)[3] .- half_doma)
198         push!(disy_5, sim.body.curve.(0.625)[2] .- half_doma)
199         push!(disz_5, sim.body.curve.(0.625)[3] .- half_doma)
200         push!(disy_6, sim.body.curve.(0.75)[2] .- half_doma)
201         push!(disz_6, sim.body.curve.(0.75)[3] .- half_doma)
202         push!(disy_7, sim.body.curve.(0.875)[2] .- half_doma)
203         push!(disz_7, sim.body.curve.(0.875)[3] .- half_doma)
204         push!(disy_8, sim.body.curve.(1.0)[2] .- half_doma)
205         push!(disz_8, sim.body.curve.(1.0)[3] .- half_doma)
206         push!(forces_y, sim.forces[2,:])
207         push!(forces_z, sim.forces[3,:])
208         push!(ttime,t*sim.U/sim.L)
209     end
210     write!(wr, sim)
211     CSV.write(name*"_displacement_"*".csv",
    Tables.table(hcat(ttime,disy_1,disz_1,disy_2,disz_2,disy_3,disz_3,disy_4,disz_4,disy_5,disz_5,disy_6,disz_6,disy_7,disz_7,disy_8,disz_8,forces_y,forces_z)),
    writeheader=false)
212 end
213 close(wr)
214
215 # uu=integration_points[78]
216 # thkk = (((thk2)/2-(thk1)/2)/(1-tapper))*uu + ((thk2)/2-((thk2)/2-(thk1)/2)/(1-tapper))
217 # α = atan(((thk1)/2-(thk2)/2)/((1-tapper)*L))
218 # @show(sim.forces[:,78])
219 # dV = -thkk^2*π * cos(α)
220 # @show(dV)
221
```

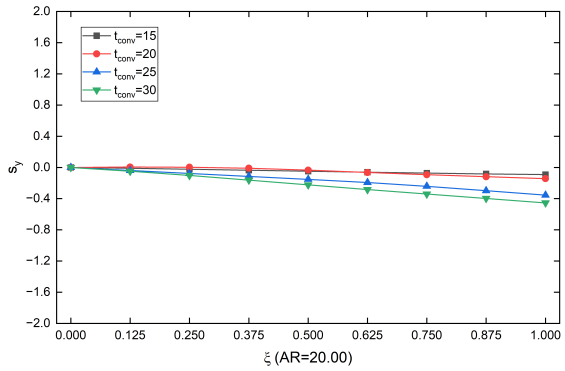
Figure J.2: Code for the cylinder with a parachute end.

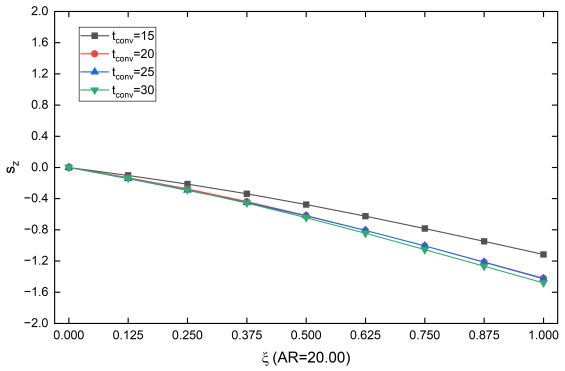
K

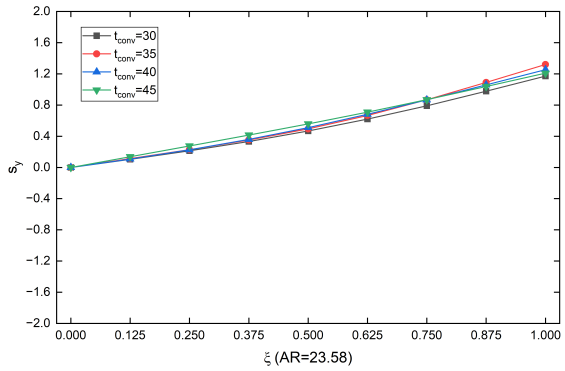
Pinned-free cylinder shapes

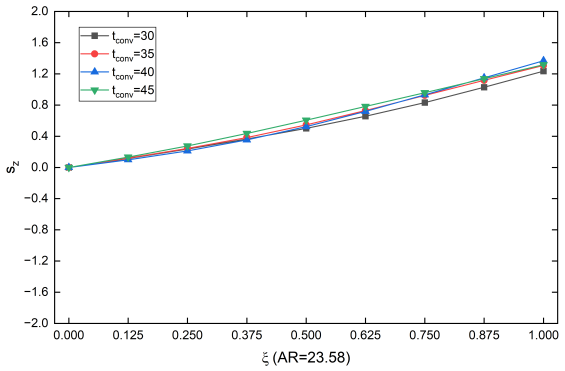












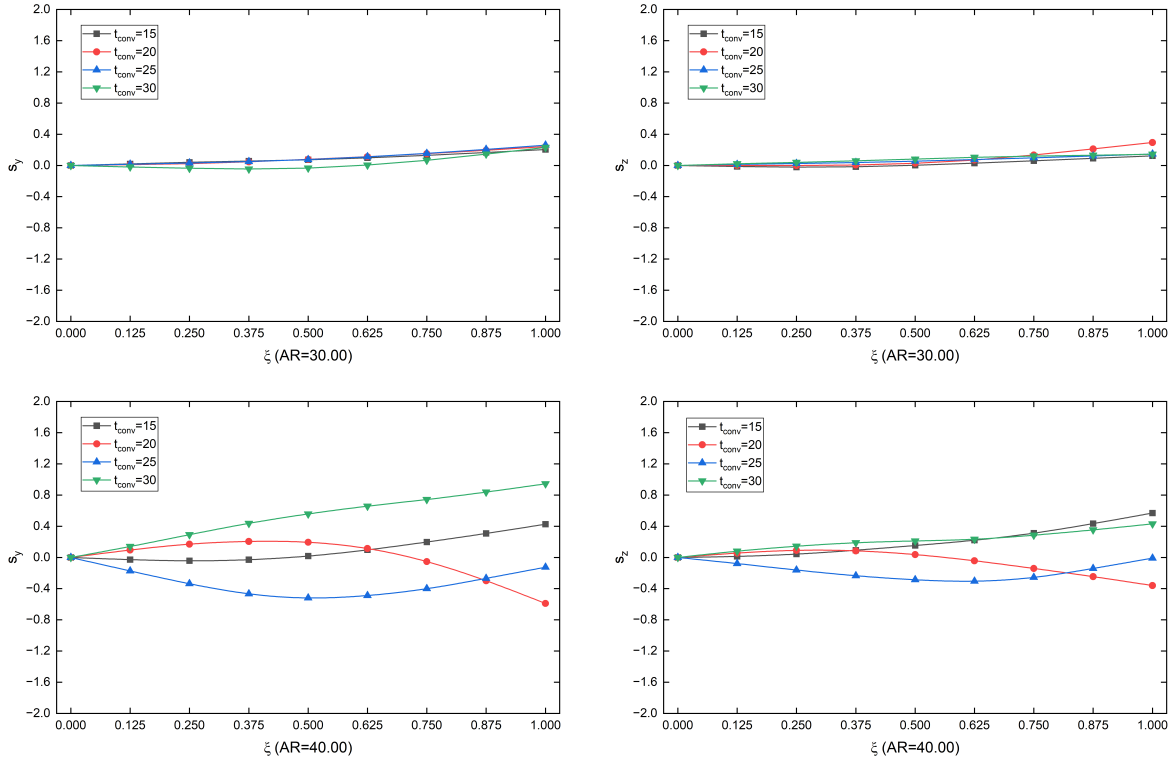


Figure K.1: Pinned-free cylinder's central axis shapes at various t_{conv}.

L

Change log

Last modified date: 11/05/2025

Added the outline of the conference paper. Appendix A.

How does the boundary layer change? Page 57, first paragraph.

Added two additional validation cases. Page 57-60. The FSI is now quantitatively validated.

Added the ranges of the displacements at the downstream end to help analyze. Figure 5.20, and Figure 6.4.

Extended the simulation time of the case with a pinned-free cylinder and $AR = 23.58$ to ensure the instability fully develops. Page 64-65.

The displacements of longer cylinders still fulfill the small deflection assumption: page 59, the second last paragraph; page 65, the second paragraph.

Added the instability prediction for a very long cylinder. Page 68, first paragraph.

The array cable is in tension or compression? Page 68, paragraph 2. The assumption in the analytical model is counterintuitive, so we no longer recommend using the model for real cases.

What will happen on a fluttered pinned-free cylinder if we increase the fluid field's Reynolds number further? Page 69, paragraph 4.

Added the hypothesis of the mechanism of a cylinder's instabilities in axial flow. Page 71, subsection 6.4.3.

Added a conclusion: The cylinder slightly vibrates around the non-zero neutral position when it is in the divergence instability. Page 72, first paragraph.

How to strengthen the array cable's upper boundary? Page 72, paragraph 2.

Added the pinned-free cylinder's central axis shape with various lengths. Appendix K.

The author also improved the thesis's typography and fixed grammar mistakes. More descriptions were added in the nomenclature and figure list.